

Tutorials & FAQs

In this section of the documentation we would like to give you some tutorials, with which the support of your Shopware Shops becomes even easier. Here you will find for example information on debugging error messages in Shopware, helpful tricks for the analysis in the database but also everything worth knowing about the GDPR.

We also take a look at the FAQ from the Shopware community to answer the frequently asked questions and give tips.

Making changes in the template

Goal

The goal of this tutorial is to learn how to make small changes in your template. With the help of a few examples we will show you how to achieve this. The first examples are described step-by-step, the following examples won't show each step separately, because the procedure is always the same.

Procedure

As procedure we take the following steps as a basis:

- Definition of the adjustment
- Searching the template file, which is responsible for the needed feature by default
- Deduce the adjustment in the own template
- Testing

The Bare Theme and the Responsive Theme are either default templates, which never should be changed, because adjustments could be deleted when this file will be updated in Shopware! Because of this you have to deduce the template correctly.

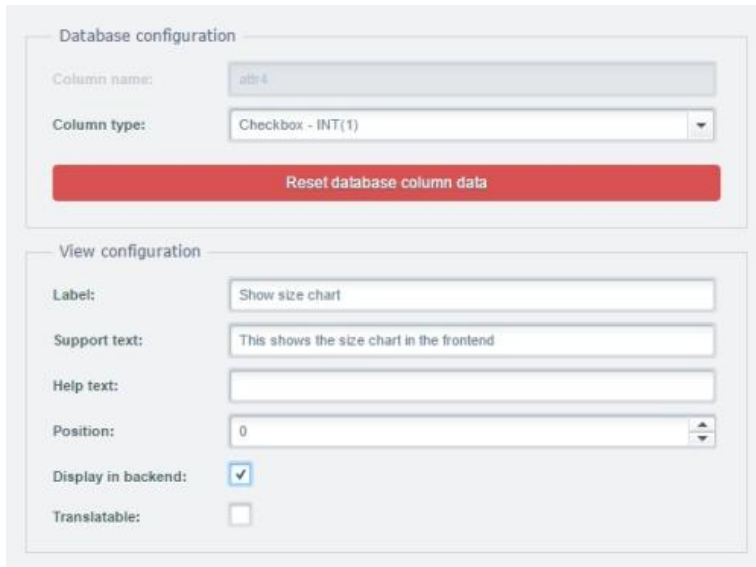
If not yet made, create a new theme in the [Theme Manager](#) which inherits from the default Responsive Theme. Shopware creates the complete directory structure for adjusting your template in the /themes directory, but don't create any files, because these are adopted from the default template, if no file exists. (More Information [here](#).)

Size chart on detail page

Goal is to display a size chart below the order number on the item detail page.

Create / manage free text field

Because the link should appear only at defined items, it makes sense to use [free text fields](#) for this. So create a new item free text field of the type "checkbox", for example we use "attr4":



The image shows a configuration interface with two main sections: "Database configuration" and "View configuration".

Database configuration:

- Column name: attr4
- Column type: Checkbox - INT(1)
- Reset database column data (red button)

View configuration:

- Label: Show size chart
- Support text: This shows the size chart in the frontend
- Help text: (empty)
- Position: 0
- Display in backend:
- Translatable:

The free text field is now available in the backend and can be set for each desired item.

Adjust the template

To spot the place, where the default template build the order number, you have to open the theme directory of the Bare Theme, because the Bare Theme builds the whole structure of Shopware. Because our desired change should be made on the detail page, navigate to the directory `/themes/Frontend/Bare/frontend/detail`, here you find all template files, which are responsible for the detail page.

Because our adjustment should be made below the order number, we need the code, which builds the order number, you can search e.g. by IDE or editors like "Notepad++" this filed for keywords to find the right place, so you should search the "detail"-directory for the string "order number".

Because "order number" is contained in multiple files, you have to navigate through the results and find the right one, which accomplishes the task, in this case it's the `index.tpl` because here the order number blocks and the attribute 1 and 2 blocks are standing among themselves. The block for the order number is named `frontend_detail_data_ordernumber` here. Now we know, that the `index.tpl` and the block `frontend_detail_data_ordernumber` is responsible for the function we want to adjust.

Inherit adjustments correctly

Now create in `/themes/Frontend/YOURTHEME/frontend/detail` a new file named `index.tpl`. At first we have to deduce from the default theme, the first line must be:

```
{extends file="parent:frontend/detail/index.tpl" }
```

Now we extend the block. Because the size chart should appear under the order number, the block has to be extended with `smarty.block.parent` ([Explanation](#)). The extended block now looks like this:

```
{* Append Article SKU with link to size chart *}
{block name='frontend_detail_data_ordernumber'}
  {smarty.block.parent}
  {if $$Article.attr4 == true}
    <li class="base-info--entry entry--sku">
      {* Size chart - Label *}
      {block name='frontend_detail_data_sizechart_label'}
        <strong class="entry--label">
          {s name="SizeChartLabel" namespace="frontend/detail/data"}{/s}
        </strong>
      {/block}
      {* Size chart - Link *}
      {block name='frontend_detail_data_sizechart_link'}
        <strong class="entry--content">
          {s name="SizeChartLink" namespace="frontend/detail/data"}{/s}
        </strong>
      {/block}
    </li>
  {/if}
{/block}
```

If your theme gets compiled now, the new code will be applied and it should be displayed in the template, but the defined snippets are still empty.

Filling snippets

The snippets "SiteChartLabel" and "SizeChartLink" are currently empty and must be filled by using the [Snippet management](#), just open the snippet manager and search for the snippets by name or open the defined namespace "*frontend/detail/data*". As example we set the snippets to "Size advice?" and a link to the size chart itself with the following HTML code:

```
<a href="/size-chart" title="Size chart" target="_blank">Size chart</a>
```

The link to `/size-chart` expects, that you defined this page yet. This can be done e.g. by using [Shop pages](#).

Further description below the item listing

In some cases it may be helpful to display another description text below the category listing, goal of this tutorial is to display this description. which should look like the category description, if the description is filled.

Create / manage free text field

Because the description should appear only in categories which have the description filled, it makes sense to use [free text fields](#) for this. So create a new text field of the type "HTML" for the categories, in this example we use "attribute1":

The image shows two sections of a configuration interface. The top section, titled "Database configuration", contains a text input field for "Column name" with the value "attribute1", a dropdown menu for "Column type" set to "HTML editor - MEDIUMTEXT", and a red button labeled "Reset database column data". The bottom section, titled "View configuration", contains several fields: "Label" with "Additional description", "Support text" with "This description is displayed beyond the item listing.", "Help text" (empty), "Position" with "0", "Display in backend" with a checked checkbox, and "Translatable" with an unchecked checkbox.

The free text field is available now in the backend and can be filled.

Adjust the template

To spot the place, where the default template builds the listing, you have to open the theme directory of the Bare Theme, because the Bare Theme builds the whole structure of Shopware. Because our desired change should be made on the listing, navigate to the directory `/themes/Frontend/Bare/frontend/listing`, here you find all template files, which are responsible for the listing.

IN this case we don't know where to start, so we take a look in the `index.tpl` first, which is the basic structure of each component, there we see the included listing and below the tag `could`. So we know that the `index.tpl` is the file we need to deduce, the Block is called `frontend_listing_index_listing`.

Spot the variables

To spot the attribute, you can use the default template naming for search:

First take a look which variables are used in the file you are working on, in the listing this is `$$sCategoryContent`, so you can use your deduced template file to get all available variables by using

```
<pre>{ $$sCategoryContent | print_r }</pre>
```

This shows you all available variables, so you can search for your attributes variable.

Inherit adjustments correctly

Then create a new file called `index.tpl` in the directory `/themes/Frontend/YOURTHEME/frontend/listing`. At first we have to deduce from the default theme, the first line must be:

```
{ extends file="parent:frontend/listing/index.tpl" }
```

The block "frontend_listing_index_listing" will be extended by using smarty.block.parent to give the content in the template after the listing is shown:

```
{* Listing *}
{* Extend listing with additional description *}
{block name="frontend_listing_index_listing"}
    {smarty.block.parent}
    {if $sCategoryContent.attribute.attribute1}
        {$sCategoryContent.attribute.attribute1}
    {/if}
{/block}
```

Because this only displays the attribute itself, this don't look nice:







>Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed diam nonummy eirmod tempor invidunt ut labore et dolore magna aliquyam erat, sed diam voluptua. At vero eos et accusam et justo duo dolores et ea rebum. Stet clita kasd gubergren, no sea takimata sanctus est Lorem ipsum dolor sit amet. Lorem ipsum dolor sit amet, consetetur sadipscing elitr, sed diam nonummy eirmod tempor invidunt ut labore et dolore magna aliquyam erat, sed diam voluptua. At vero eos et accusam et justo duo dolores et ea rebum. Stet clita kasd gubergren, no sea takimata sanctus est Lorem ipsum dolor sit amet. Lorem ipsum dolor sit amet, consetetur sadipscing elitr, sed diam nonummy eirmod tempor invidunt ut labore et dolore magna aliquyam erat, sed diam voluptua. At vero eos et accusam et justo duo dolores et ea rebum. Stet clita kasd gubergren, no sea takimata sanctus est Lorem ipsum dolor sit amet.

So we change it another time and get the styling of the category description to use it for our additional description too, that it looks the same. As described in "Find the place which is responsible for the function", we search for the category description, which is built in the text.tpl and adopt this to our description box:

```
{block name="frontend_listing_index_listing"}
    {smarty.block.parent}
    <div class="hero-unit category--teaser panel has--border is--rounded">
        <div class="hero--text panel--body is--wide">
            {if $sCategoryContent.attribute.attribute1}
                <div class="teaser--text-long">
                    {$sCategoryContent.attribute.attribute1}
                </div>
                <div class="teaser--text-short is--hidden">
                    {$sCategoryContent.attribute.attribute1|strip_tags|truncate:200}
                    <a href="#" title="{s namespace="frontend/listing/listing" name="ListingActionsOpenOffCanvas" s namespace="frontend/listing/listing" name="ListingActionsOpenOffCanvas"}" >
                        </a>
                </div>
                <div class="teaser--text-offcanvas is--hidden">
                    <a href="#" title="{s namespace="frontend/listing/listing" name="ListingActionsOpenOffCanvas" s namespace="frontend/listing/listing" name="ListingActionsOpenOffCanvas"}" >
                        <i class="icon--arrow-left"></i> {s namespace="frontend/listing/listing" name="ListingActionsOpenOffCanvas"}
                    </a>
                    <div class="offcanvas--content">
                        <div class="content--title">{$sCategoryContent.cmsheadline}</div>
                        {$sCategoryContent.attribute.attribute1}
                    </div>
                </div>
            {/if}
        </div>
    </div>
{/block}
```

The result looks like this:

Desktop:

 ROCKWELL £569.00 *	 PROVOKE £399.00 *	 A 10/30 £750.00 * £799.00	 HILL BILL £649.00 *
---	--	---	--

Lorem ipsum dolor sit amet, consetetur sadipscing elitr, sed diam nonumy eirmod tempor invidunt ut labore et dolore magna aliquyam erat, sed diam voluptua. At vero eos et accusam et justo duo dolores et ea rebum. Stet clita kasd gubergren, no sea takimata sanctus est Lorem ipsum dolor sit amet. Lorem ipsum dolor sit amet, consetetur sadipscing elitr, sed diam nonumy eirmod tempor invidunt ut labore et dolore magna aliquyam erat, sed diam voluptua. At vero eos et accusam et justo duo dolores et ea rebum. Stet clita kasd gubergren, no sea takimata sanctus est Lorem ipsum dolor sit amet. Lorem ipsum dolor sit amet, consetetur sadipscing elitr, sed diam nonumy eirmod tempor invidunt ut labore et dolore magna aliquyam erat, sed diam voluptua. At vero eos et accusam et justo duo dolores et ea rebum. Stet clita kasd gubergren, no sea takimata sanctus est Lorem ipsum dolor sit amet.

Duis autem vel eum inure dolor in hendrerit in vulputate velit esse molestie consequat, vel illum dolore eu feugiat nulla facilisis at vero eros et accumsan et justo odio dignissim qui blandit praesent luptatum zzril delenit augue duis dolore te feugait nulla facilisi. Lorem ipsum dolor sit amet, consetetur adipscing elit, sed diam nonumy nibh euismod tincidunt ut laoreet dolore magna aliquam erat volutpat.

Mobile with opened off-canvas box:

£519.00 * Close window

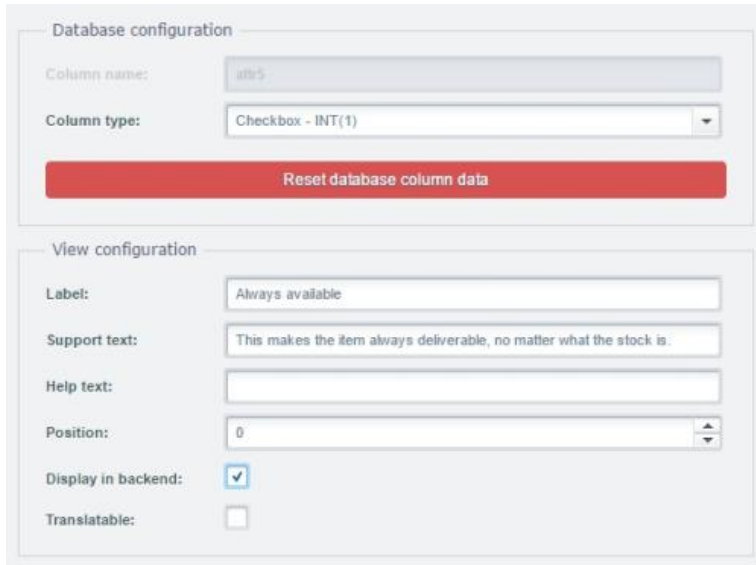
 ROCKWELL £569.00 *	Lorem ipsum dolor sit amet, consetetur sadipscing elitr, sed diam nonumy eirmod tempor invidunt ut labore et dolore magna aliquyam erat, sed diam voluptua. At vero eos et accusam et justo duo dolores et ea rebum. Stet clita kasd gubergren, no sea takimata sanctus est Lorem ipsum dolor sit amet. Lorem ipsum dolor sit amet, consetetur sadipscing elitr, sed diam nonumy eirmod tempor invidunt ut labore et dolore magna aliquyam erat, sed diam voluptua. At vero eos et accusam et justo duo dolores et ea rebum. Stet clita kasd gubergren, no sea takimata sanctus est Lorem ipsum dolor sit amet. Lorem ipsum dolor sit amet, consetetur sadipscing elitr, sed diam nonumy eirmod tempor invidunt ut labore et dolore magna aliquyam erat, sed diam voluptua. At vero eos et accusam et justo duo dolores et ea rebum. Stet clita kasd gubergren, no sea takimata sanctus est Lorem ipsum dolor sit amet.
 A 10/30 £750.00 * £799.00	Lorem ipsum dolor sit amet, consetetur sadipscing elitr, sed diam nonumy eirmod tempor invidunt ut labore et dolore magna aliquyam erat, sed diam voluptua. At vero eos et accusam et justo duo dolores et ea rebum. Stet clita kasd gubergren, no sea takimata sanctus est Lorem ipsum dolor sit amet. Lorem ipsum dolor sit amet, consetetur sadipscing elitr, sed diam nonumy eirmod tempor invidunt ut labore et dolore magna aliquyam erat, sed diam voluptua. At vero eos et accusam et justo duo dolores et ea rebum. Stet clita kasd gubergren, no sea takimata sanctus est Lorem ipsum dolor sit amet.
Lorem ipsum dolor sed diam nonumy e dolore magna aliqu eos et accusam et ji.	Lorem ipsum dolor sit amet, consetetur sadipscing elitr, sed diam nonumy eirmod tempor invidunt ut labore et dolore magna aliquyam erat, sed diam voluptua. At vero eos et accusam et justo duo dolores et ea rebum. Stet clita kasd gubergren, no sea takimata sanctus est Lorem ipsum dolor sit amet.

Mark defined items as always deliverable

Goal of this tutorial is, that defined items are displayed as "Ready to ship today" no matter what the stock the item in Shopware has.

Create / manage free text field

Because this feature should also be usable for every item itself, we use [free text fields](#) also here. Create a new free text field of the type "Checkbox" for items, in our example we use "attr5":



The image shows a configuration interface with two main sections: 'Database configuration' and 'View configuration'. In the 'Database configuration' section, there is a text input field for 'Column name' containing 'attr5', a dropdown menu for 'Column type' set to 'Checkbox - INT(1)', and a red button labeled 'Reset database column data'. The 'View configuration' section includes a 'Label' field with 'Always available', a 'Support text' field with 'This makes the item always deliverable, no matter what the stock is', an empty 'Help text' field, a 'Position' field with '0', a checked 'Display in backend' checkbox, and an unchecked 'Translatable' checkbox.

The free text field now is available for use.

Adjust the template

The availability is displayed at multiple places in Shopware, because of this we search the whole Bare Theme folder for keywords using our IDE or "Notepad++" to find their usage. Trying this we see that "delivery" is used in the `delivery_informations.tpl` which again is used at the relevant places in the frontend, so we seem to be right. Now we know, which file is responsible for the availability, because this file has only 1 block, this is clear.

Inherit adjustments correctly

Now create a new file named `delivery_informations.tpl` in `/themes/Frontend/DEINTHEME/frontend/plugins`. At first we have to deduce from the default theme, the first line must be:

```
{extends file="parent:frontend/plugins/delivery_informations.tpl"}
```

Now we change the block this way, that we retrieve our defined attribute and then display the right delivery status. The "Ready to ship today" state can be deduced by its name: "DetailDataInfoInStock" - Alternatively you also can search the snippets by name, this also works fine. Now we change the file:

```
{block name='frontend_widgets_delivery_infos'}
  {if $$Article.attr5 == true || $$BasketItem.additional_details.attr5 == true}
    <div class="product--delivery">
      <link href="http://schema.org/InStock" itemprop="availability">
        <p class="delivery--information">
          <span class="delivery--text delivery--text-available">
            <i class="delivery--status-icon delivery--status-available"></i>
              {s name="DetailDataInfoInStock"}}{/s}
          </span>
        </p>
      </div>
    {else}
```

```
{/if} {$smarty.block.parent}
{/block}
```

This way the item will look like "Ready to ship today" no matter what stock it is.

Deleting something from the theme

If you want to delete something from the frontend, you can do this easily by overwriting the desired block empty, this deletes the content in the frontend. An example:

```
{extends file="parent:PARENT"}
{block name='NAME'}
{/block}
```

Keep in mind, that this requires the same steps as described before, only deduce the theme in your own one, never change the default theme!

Changing the robots.txt

Because Shopware generates the robots.txt also over the theme, you can also change this easily. The path to the template file, which is responsible for generating the robots.txt, is /frontend/robots_txt/index.tpl.

Here you can deduce this file in your own theme like described before to individualize your robots.txt file.

Display divergent shipping address on the invoice

Documents are also template files which can also be individualized. The default document template files can be found in /Bare/documents, the files have the following responsibilities:

File	Responsible for
index.tpl	Invoice
index_gs.tpl	Credit
index_ls.tpl	Delivery note
index_sr.tpl	Cancellation

For example we add the divergent shipping address to the invoice, when it differs to the billing address.


```

{block name="document_index_info_dispatch"}
  {if $Order._dispatch.name}
    <div style="font-size:11px;color:#333;">
      {s name="DocumentIndexSelectedDispatch"}{/s}
      {$Order._dispatch.name}
    </div>
  {/if}
  {assign var="shippingaddress" value="shipping"}
  {assign var="shippingArr" value=$User.$shippingaddress}
  {assign var="billingArr" value=$User.$address}
  {$billingString = $billingArr['salutation']|cat:$billingArr['title']|cat:$billingArr["firstnam
  {$shippingString = $shippingArr['salutation']|cat:$shippingArr['title']|cat:$shippingArr["fir
  {if $shippingString != $billingString}
    {s name="DocumentIndexSelectedShippingaddress"}{/s}
    <div style="margin-left: 20px;">
      {if $User.$shippingaddress}
        {$User.$shippingaddress.salutation|salutation}
        {if {config name="displayprofiletitle"}}
          {$User.$shippingaddress.title}<br/>
        {/if}
        {$User.$shippingaddress.firstname} {$User.$shippingaddress.lastname}
        {$User.$shippingaddress.street}
        {if {config name=showAdditionAddressLine1}}
          {$User.$shippingaddress.additional_address_line1}
        {/if}
        {if {config name=showAdditionAddressLine2}}
          {$User.$shippingaddress.additional_address_line2}
        {/if}
        {if {config name=showZipBeforeCity}}
          {$User.$shippingaddress.zipcode} {$User.$shippingaddress.city}
        {else}
          {$User.$shippingaddress.city} {$User.$shippingaddress.zipcode}
        {/if}
      {/if}
    </div>
  {/if}
{/block}

```

Add further country flags for the language change

In the default setting, the flags for Germany, Great Britain, the Netherlands, France, Spain and Italy are stored for the language change. These are stored in the file `/themes/Frontend/Responsive/frontend/_public/src/less/_components/flags.less`. To provide further flags for selection, a theme adjustment is necessary.

Adjustments

flags.less

The `flags.less` file contains the information on the localisations and flags to be used. Each localisation receives its own entry that begins with the ISO code. You can find this code in the basic settings under **Shop settings > Localisations**.

In general, you can create your own flags using CSS.

This is particularly useful for flags that consist of horizontal/vertical stripes, for example.

The flags stored in the standard can also be used as a guide.

More complex flags can also be created in this way. However, since several layers have to be worked with here, this may require advanced CSS knowledge.

As a simpler alternative, it is possible to integrate a ready-made flag as a base64. To do this, an image is reformatted into a character string and this character string is stored in the file.

Further details on Base64 and a converter can be found at <https://www.base64-image.de/>.

In your theme, save the customised file under:

```
/themes/Frontend/YOURTHEME/frontend/_public/src/less/_components/flags.less
```

Example

Creating a flag as CSS

The extension for Hungary is listed here as an example

```
&.hu_HU { background-position: 0 0; background-image: linear-gradient(to bottom, #DD0000 33%,
```

The entry in flags.less begins with the ISO code of the localisation, in this case hu_HU.

This is followed by the information on the creation of the flag. The beginning

```
background-position: 0 0; background-image:
```

is identical for all flags created using CSS. The part

```
linear-gradient(to bottom, #DD0000 33%, white 33%, white 66%, #336333 66%);
```

contains the information for creating the flag. In this case, the flag is created from **top to bottom** ("linear-gradient(to bottom)"). The first part takes up **33%** of the area and receives the colour with the HEX value **DD0000**. The second colour area starts at 33% and ends at 66% with the colour white. The last area starts at 66% and receives the colour value 336333).

The complete file then looks like this:

```
/*
Language Flags
=====
Displays a country flag the size of 14px x 11px used for language selections purposes.
inspired by: https://github.com/dhanishgajjar/css-flags
en_GB: we need to use svg for en_GB, cause it's smaller than the CSS-Version
*/

.language--flag {
    .unitize-height(11);
    .unitize-width(14);
    display: inline-block;
    text-indent: 100%;
    white-space: nowrap;
    overflow: hidden;

    &.de_DE { background-position: 0 0; background-image: linear-gradient(to bottom, black 33%
    &.en_GB { background-position: 0 0; background-size: 100%; background-repeat: no-repeat; b
    &.nl_NL { background-position: 0 0; background-image: linear-gradient(to bottom, #DD0000 3
    &.fr_FR { background-position: 0 0; background-image: linear-gradient(to right, #21468B 33
    &.es_ES { background-position: 0 0; background-image: linear-gradient(#DD0000 25%, #FFCE00
    &.it_IT { background-position: 0 0; background-image: linear-gradient(to right, #009246 33
```

```
} &.hu_HU { background-position: 0 0; background-image: linear-gradient(to bottom, #DD0000 33%, #DD0000 33%, #DD0000 33%); background-size: 100%; background-repeat: no-repeat; background-attachment: scroll; }
```

Embed flag as Base64

The extension for Sweden is listed here as an example.

First find an image file of the flag you want to use and convert it to Base64 code.

```
&.sv_SE { background-position: 0 0; background-size: 100%; background-repeat: no-repeat; background-attachment: scroll; }
```

The entry in **flags.less** begins with the ISO abbreviation of the localisation. This is followed by a part that is generally valid for the inclusion of Base64 code

```
background-position: 0 0; background-size: 100%; background-repeat: no-repeat; background-attachment: scroll; }
```

The individual code is then attached to it.

The complete file then looks like this:

```
/*
Language Flags
=====
Displays a country flag the size of 14px x 11px used for language selections purposes.
inspired by: https://github.com/dhanishgajjar/css-flags
en_GB: we need to use svg for en_GB, cause it's smaller than the CSS-Version
*/

.language--flag {
  .unitize-height(11);
  .unitize-width(14);
  display: inline-block;
  text-indent: 100%;
  white-space: nowrap;
  overflow: hidden;

  &.de_DE { background-position: 0 0; background-image: linear-gradient(to bottom, black 33%, black 33%, #DD0000 33%); background-size: 100%; background-repeat: no-repeat; background-attachment: scroll; }
  &.en_GB { background-position: 0 0; background-size: 100%; background-repeat: no-repeat; background-attachment: scroll; }
  &.nl_NL { background-position: 0 0; background-image: linear-gradient(to bottom, #DD0000 33%, #DD0000 33%, #DD0000 33%); background-size: 100%; background-repeat: no-repeat; background-attachment: scroll; }
  &.fr_FR { background-position: 0 0; background-image: linear-gradient(to right, #21468B 33%, #21468B 33%, #21468B 33%); background-size: 100%; background-repeat: no-repeat; background-attachment: scroll; }
  &.es_ES { background-position: 0 0; background-image: linear-gradient(#DD0000 25%, #FFCE00 25%, #FFCE00 25%, #FFCE00 25%, #FFCE00 25%, #FFCE00 25%); background-size: 100%; background-repeat: no-repeat; background-attachment: scroll; }
  &.it_IT { background-position: 0 0; background-image: linear-gradient(to right, #009246 33%, #009246 33%, #009246 33%); background-size: 100%; background-repeat: no-repeat; background-attachment: scroll; }
  &.sv_SE { background-position: 0 0; background-size: 100%; background-repeat: no-repeat; background-attachment: scroll; }
}
```

Display the payment status of orders in the frontend

Here we describe how to add the payment status to your own order or payment state and sending emails by using it.

Implement the status in the theme

The easiest way to provide the payment status in the frontend is to add it below the orderstatus. To do this, add the "order_item.tpl" in your own theme in the following path:
/themes/Frontend/YOURTHEMENAME/frontend/account/.

There you can extend the following block with the payment status:

```
{extends file="parent:frontend/account/order_item.tpl"}
{block name="frontend_account_order_item_status_value"}
{$smarty.block.parent}
<div class="column--value">
  <span class="order--status-icon status--{$offerPosition.cleared}"></span>
  {if $offerPosition.cleared==9}
  {s name="partially_invoiced"}Partially invoiced{/s}
  {elseif $offerPosition.cleared==10}
  {s name="completely_invoiced"}Completely invoiced{/s}
  {elseif $offerPosition.cleared==11}
  {s name="partially_paid"}Partially paid{/s}
  {elseif $offerPosition.cleared==12}
  {s name="completely_paid"}Completely paid{/s}
  {elseif $offerPosition.cleared==13}
  {s name="1st_reminder"}1st Reminder{/s}
  {elseif $offerPosition.cleared==14}
  {s name="2nd_reminder"}2nd Reminder{/s}
  {elseif $offerPosition.cleared==15}
  {s name="3rd_reminder"}3rd Reminder{/s}
  {elseif $offerPosition.cleared==16}
  {s name="encashment"}Encashment{/s}
  {elseif $offerPosition.cleared==17}
  {s name="open"}Open{/s}
  {elseif $offerPosition.cleared==18}
  {s name="reserved"}Reserved{/s}
  {elseif $offerPosition.cleared==19}
  {s name="delayed"}Delayed{/s}
  {elseif $offerPosition.cleared==20}
  {s name="re_crediting"}Re-Crediting{/s}
  {elseif $offerPosition.cleared==21}
  {s name="review_necessary"}Review necessary{/s}
  {/if}
</div>
{/block}
```

Add the correct Icon

To display the corresponding icon next to the status and you already know about LESS-Styling you could add the following code. If you're not so good with LESS at this moment you could create your own "all.less" file. The LESS code will be compiled into normal CSS by the LESS compiler. Create your all.less in the following directory: /themes/Frontend/YOURTHEMENAME/frontend/_public/src/less Please enter the following code:

```
.order--status-icon {
  &.status--9, //partially_invoiced
  &.status--10, //completely_invoiced
  &.status--11, //partially_paid
  &.status--13, //1st_reminder
  &.status--14, //2nd_reminder
  &.status--15, //3rd Reminder
  &.status--17, //open
  &.status--18, //reserved
  &.status--19 { //delayed
    background: @highlight-info;
  }

  &.status--12, //completely_paid
  &.status--20 { //re_crediting
    background: @highlight-success;
  }

  &.status--16, //encashment
  &.status--21 { //review_necessary
    background: @highlight-error;
  }
}
```

If you clear the cache after your adjustment and re-compile your theme, the payment status is displayed as shown in the frontend:

Orders

Here you can find your orders sorted according to date.

Date	Order number	Shipping type	Order status	Special offers
7 Jun 2017 09:35:00 AM	20010	Englischer Versand	■ Your order has not been processed yet. ■ Open	View
7 Jun 2017 09:34:00 AM	20009	Englischer Versand	■ Your order has not been processed yet. ■ Partially paid	View
7 Jun 2017 09:34:00 AM	20008	Englischer Versand	■ Your order has not been processed yet. ■ Encashment	View
7 Jun 2017 09:30:00 AM	20007	Englischer Versand	■ Your order has not been processed yet. ■ Completely paid	View

Remove Shopware logo and lettering from the footer

Remove logo

You can remove the Shopware logo from the footer via a template adjustment.



Example

First you have to connect to your FTP server. In the Shopware directory under "**themes > Frontend > YOUR THEME > frontend > index**" you have the option of including a file that removes the logo.

To do this, create the file "footer.tpl" with the following content in the named path:

```
{extends file="parent:frontend/index/footer.tpl"}  
{block name="frontend_index_shopware_footer_logo"}  
{/block}
```

Please note that this is an example and that you may have to make further changes to the code. You should also make sure to create your own theme, which is derived from the original theme, so that your changes are update-safe.

After you have created the file, recompiled the theme and cleared the cache, the logo should be removed.

Remove lettering

If you want to change or remove the text "Realised with Shopware", you can do this in the snippet management under **Settings > Snippets** in the backend. Here you can simply search for the text module "IndexCopyright" and remove the value.

AGPLv3

Significance for Developers and Customers

The Affero General Public License v3 is a derivative of the GPLv3 and grants anyone the non-exclusive right to use the software. As of Shopware 4.0, the software is free (open source) and may be copied, circulated or adapted to meet individual needs. The AGPLv3 contains a copyleft clause, which means all modified versions of the software must also be licensed under AGPLv3. Developers are free to access the source code and may redistribute a modified form of the software in accordance with these license terms. Customers may use Shopware without a license fee, but must meet the license obligations of AGPLv3. Access to the source code ensures the sustainability of the Community Edition and that all necessary adjustments are in line with AGPLv3.

Templates Modifications

Portions of the standard template of the Shopware CE are made available under the new BSD license. This allows individuals to develop their own frontend templates, since a block system inherits portions of the standard template. This makes it possible to adapt the template to meet individual needs. It is important to take note this does not apply for Shopware's backend templates. These are licensed under AGPLv3; therefore any customized versions of the standard backend templated much be relicensed under AGPLv3.

Plugins

We want to give developers the freedom to create plugins for diverse business models without license requirements. For this reason, we allow an exception to the license terms of AGPLv3 for self-written plugins. These plugins have separate license terms - even when the plugin is derived from the class "Shopware_Components_Plugin_Bootstrap" - and do not have to be approved as open source software. For this purpose, the AGPLv3 license text contains the following add-on: Additional permission under AGPLv3, Sec. 7 If you modify this Program, or any covered work, by combining it with Plugins entirely written by you, the licensors of this Program grant you additional permission to convey those Plugins under terms of your choice, provided that those terms do not place additional restrictions on the Program.

Core Modifications

Modifications and extensions of the source code, that have no effect on the frontend template or plugin system, are considered modifications of the AGPLv3 license. These must be licensed under AGPLv3. The same applies for override classes.

Changes to the Shopware CE must be licensed under AGPLv3 when they are used by third parties or offered in the Shopware Community Store. Any propagated copies of the modified Shopware version are considered "conveyed" work, and therefore must be made available following paragraphs 4 & 5 of the AGPLv3. The mere operation of an online shop means that the software is being used. This is referred to in paragraph 13 of the AGPLv3 as "interacting with it remotely through a computer network" and therefore requires a release of changes.

Publication of Changes

Paragraphs 5 & 6 of the AGPLv3 specify license obligations when Shopware CE is changed. Changes must be identified and provided with a date. It's best to provide this information as a header in the source code, i.e.: * 31.07.2012: install method changed, © 2012 Max Muster * This program is free software: you can redistribute it and/or modify * it under the terms of the GNU Affero General Public License as * published by the Free Software Foundation, either version 3 of the * License, or (at your option) any later version. * / The complete amended source code must be made available and readable by machines. The AGPLv3 allows several variants. If the source code is only used or distributed online, the easiest way is to directly supply the code with the source information. Should the source code be hosted on a different server, there should be a download link placed in close proximity to ensure the object code can be easily accessible. The details can be found in the AGPLv3 license conditions.

Differences between the GPL and AGPLv3

The GNU General Public License, version 3 (GPLv3) contains a weaker copyleft agreement than the AGPLv3. Under the GPLv3 license, modifications to the software only have to be relicensed in conventional cases of software distribution, which does not include SaaS or cloud services (i.e. when the software runs on a server). We wanted to be sure that all changes to the source code were made available, therefore opted for the AGPLv3 license. According to paragraph 13 of AGPLv3, those who use a modified version of the Community Edition must make changes to the source code available. Apart from that, the GPLv3 and AGPLv3 have only minor differences.

Trademark Law

No. "Shopware" is a registered trademark of shopware AG. Therefore no variations of the name (i.e. "MyShopware") may be used. Our customers should be able to easily identify which Shopware versions come from us. This not only applies to the name under which a modified version is distributed, but also for all names within the software. However, modified versions of the Community Edition may be used under a different name.

Support for Open Source Plugins / Templates

No. If you offer your plugins or templates under open source licenses, you are not obligated to provide support. However, if plugins or templates are commercially sold, support is generally required due to laws regarding sales warranties.

Compatibility of AGPLv3

AGPLv3 is compatible with the following licenses: GPLv3, GPLv2 (or any later version), LGPLv3, LGPLv2.1 (or any later version), BSD license (2-clause and 3-clause), MIT license and Apache License 2.0. "Compatible" means that code used under these other licenses may be licensed with the Shopware Community Edition under AGPLv3.

Implementation in your Software Environment

Implementation is only possible if the other open source license is compatible with AGPLv3. When compatible, it is permitted to use the modified Shopware version under AGPLv3. This is to ensure that additional license obligations (not already included in the AGPLv3 – see paragraph 7) are listed separately

in the source file. Additionally, the license text of the implemented code must be listed separately, along with the program from which the code originates.

Reset the backend password

Introduction

In case of losing your password you can reset your password via database, for this, your still used password will be overwritten. Keep in mind, that all passwords in shopware are saved encrypted.

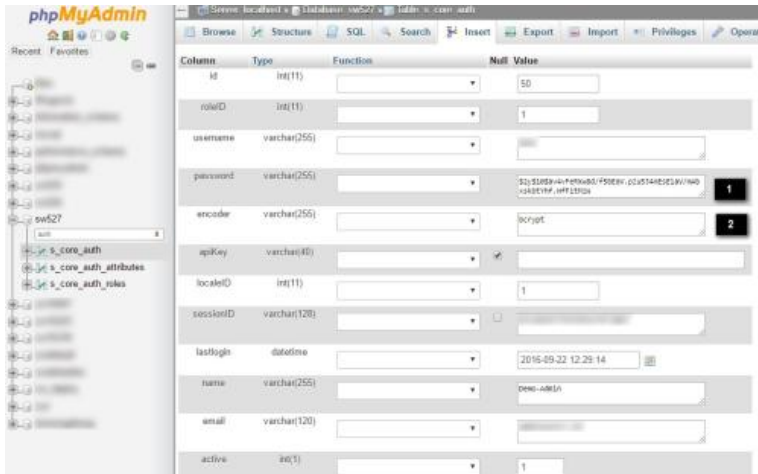
Changing password

With database

To change your password to "demo", open phpMyAdmin and open the table **s_core_auth** in your database, here you have to edit the entry for your desired user with the following data:

```
password: $2y$10$av4vFeRXw8d/f5bEaV.pzu5J4nEsEiaV/mAbxskbtYhf.HfFitMzW  
encoder: bcrypt
```

After this, you can login to your shop with the new password.



With SSH/CLI

If you can access your shop with SSH, you can create a new Admin user with the following CLI command:

```
php bin/console sw:admin:create
```


You can find more information about the Shopware CLI here: [Shopware CLI documentation](#)

```
php bin/console sw:admin:create --email=$EMAIL --username=$USERNAME --name=$NAME --locale=$LOCALE
```

User rights for plugins without ACL support

Introduction

Sometimes it occurs, that a plugin from the Shopware Store is not compatible with our user management and don't provide an ACL resource. In some cases you might still not want all users to have access to the menu item of the plugin. In this tutorial we explain how you can hide the menu item through a manual configuration in the backend.

Identification of the controller

First you need the controller that is used in the backend when a user clicks the menu item. This you can identify with a tool like Firebug (Firefox Add-On) or the Chrome-debugger (F12). In our example we use the plugin "Business Essentials" as example. The **Business Essentials** controller we need is the GET-Parameter **/backend/BusinessEssentials?...**

Firebug

In Firebug you can either see the url in the tab **Console** or the Tab **Network**. You should clear the view complete to get a clear list of requests, then click on the menu item in the backend. The first GET-parameter should be the correct URL for the menu item.

Chrome-Debugger

The Chrome-debugger is similar to this. Here you can also find the information in the tab **Network**.

Creation of the user rights

With the URL you know the exact controller-name. In the case of Business Essentials this is **BusinessEssentials**. The parameters that occur after the "?" in the GET call can be ignored. Please keep in mind that the call can differ from the plugin name! Knowing the controller name you can create a new rule & permission at **Configuration > User administration**. To do this, open the menu item "*Edit rules & permissions*" and chose a role there, afterwards you can add a new permission for Business Essentials using "*Add resource*". For the naming of the resource it is important to exactly type the lowercase of the controller name.

In our example this is businessessentials. Create a resource with the name businessessentials and add the privilege **read**. All roles that shall be able to see the menu item have to get the privilege **read**. For the

roles, who shall not see the menu item, the whole resource has to be deactivated. All further functions in the user administration have to be individually implemented in the pluign and are not possible without programming.

Hiding the main menu

It is also possible to hide the main menu entries. For this you have to read the controller name from the `s_core_menu` for the menu entry and create it in lower-case as resource as described above.

Calculation of the Cart

Introduction

The heart of a shop is the shopping cart. A comprehensible and correct calculation of the sums, positions and taxes is essential in order not to disturb the confidence of the customers in the shop. For this reason, this article should once clarify the calculation basis of the shopping cart and give you opportunities to check the calculation yourself. In addition to numerous basic functions of shopware such as price groups, discounts and surcharges, shipping costs, vouchers and promotions, there are of course also countless plugins that take part in the calculation of the shopping cart. Therefore it is important that you also get an idea of which extensions may influence the calculation and what effects this may have.

Basics

In Germany there are two valid methods for calculating VAT. The **vertical** and the **horizontal** method. Shopware uses the **vertical** method, where the tax is calculated at item level, rounded and finally added together. In contrast, the **horizontal** method first calculates the sum of the VAT contained in all products and then rounds it. The use of the **vertical** method enables the tax contained in each item to be shown on the /checkout/confirm page. This is not possible with the **horizontal** procedure.

Shopware also shows two decimal places at all points of the frontend. So that your customers can always follow the calculation, the value rounded to 2 digits is always used.

Calculation Examples

Net Price

Vertical (used by shopware)

Quantity	Tax Rate	Net	Gross	plus VAT		Net Total	plus VAT	Gross Total
----------	----------	-----	-------	----------	--	-----------	----------	-------------

5	7	1,49	1,59	0,10		5*1,49	5*0,10	7,45+0,50
						7,45	0,50	7,95

Horizontal

Quantity	Tax Rate	Net	Gross		Net Total	plus VAT	Gross Total
5	7	1,49	-		5*1,49	7,45*0,07	7,45+0,52
					7,45	0,52	7,97

Gross Price

Vertical (used by shopware)

Quantity	Tax Rate	Net	Gross	plus VAT		Net Total	plus VAT	Gross Total
5	7	1,49	1,59	0,10		7,95-0,50	5*0,10	5*1,59
						7,45	0,50	7,95

Horizontal

Quantity	Tax Rate	Net	Gross		Net Total	plus VAT	Gross Total
5	7	-	1,59		7,95/1,07	7,43-7,95	5*1,59
					7,43	0,52	7,95

Explanation

The calculation differences arise here on the basis of the calculation method used. In online shops, the **vertical** calculation is the more common method, since it is important for many shop operators to be able to show the VAT contained at item level. This would not be possible with the **horizontal** calculation, since the rounded item values can never result in the total value.

Debugging

Shopware does not have a central place for calculating the shopping basket. The calculations in the front- and backend are basically separated from each other. Therefore, you should always follow the complete path in your shop if you suspect an error here. Best you proceed as follows, if a customer contacts you:

- **Reproduce shopping cart in front end**
- **Take a screenshot of the order completion page (/checkout/finish)**
- **Compare totals in the order overview with the screenshot**
- **Compare totals in the PDF invoice with the screenshot**
- **Check calculated shopping cart with [Excel spreadsheet](#)**

If a problem is confirmed here, you should first check whether one of the plugins used interferes with the calculation of the shopping cart. For this you can use the security mode or set up a test shop that does not use any plugins.

Remove defective plugin

Under certain circumstances, a plugin may no longer function properly and can not be deactivated or uninstalled correctly in the Plugin Manager. In these cases, it is possible to remove the plugin and the associated plugin data manually.

Note that you will be removing data from the database and files from your server. So make sure you have a backup of the database and your data so that you can restore everything in case of an emergency.

Deactivate functions

First, you should deactivate the functions of the affected plugin. For example, if it is a payment plugin or a shipping plugin, deactivate the respective payment method or shipping method so that it can no longer be used by customers. In general, the functions provided by the defective plugin should no longer be used so that problems do not arise when the plugin is removed.

Remove database entry

If you only want to deactivate the plugin, it is sufficient to set the entry in the **active** column of the respective plugin to 0 in the database table **s_core_plugins**. This can be sufficient to solve the problems caused by the defective plugin.

If you want to remove it completely because it is still causing problems or even because it is to be completely reinstalled, you can remove the data record of the plugin in the database table **s_core_plugins**.

Remove plugin data

The plugin may also have created a plugin directory that should be physically removed from your server. You will usually find this in the Shopware main directory under **engine/Shopware/Plugins/**. This directory has the three subfolders **Community**, **Default** and **Local**, which each have the three subfolders **Backend**, **Core** and **Frontend**. The directory of the plug-in can usually be found in one of these subfolders. If it cannot be found here, you should ask the manufacturer of the plugin whether and where the physical data is stored.

General Data Protection Regulation (GDPR)

On 25 May 2018, the European General Data Protection Regulation (GDPR) will come into force and replace national regulations. In addition, the GDPR will apply to all companies and institutions operating within the EU that work with personal data such as names, addresses, bank details and dates of birth. The standardised processing of personal data is primarily aimed at better transparency in handling consumer data and the improved protection of this data. This wiki article is intended to provide you with a brief summary regarding the processing of personal data in Shopware so that you can prepare the required documentation (for example the list of processing activities, or LPA for short) or prepare the data protection declaration for a Shopware instance better. Since implementing the GDPR differs very considerably from one company to another, when implementing it in your shop you should always consult your legal adviser in order to meet all the requirements here. You can obtain a general overview of the contents of the GDPR from our [whitepaper](#).

What personal data is processed in Shopware?

At this point we would like to show you what personal data is processed directly in Shopware. As an eCommerce system it is of course absolutely necessary to process data from the customer, to record supplier addresses, for example. Here you should note that not only the software itself, but also the underlying hardware is involved in the processing. Since your host has a very individual set-up, you should clarify with him, if necessary, the extent to which communication is taking place there (e.g. with a separate database server, where the application communicates with the database via a network). Here it should only be about processing this data in Shopware on the software side.

This point is deliberately kept very general for now. Shopware itself stores a wide variety of data that is either directly related to the user (= personal) or not directly related to the user (anonymised). Anonymised data is not personal data and is not covered by the GDPR, which means that no special measures are required here. N.B.: Pseudonymised data that can be assigned to a person, for example by using an indicator, (e.g. via bank details, a customer number or an order number) counts as personal data again, even if it is not immediately apparent who the person is. The personal data is always required when Shopware accepts an entry from the customer or the shop owner has to work with this data. This may involve the registration or assessment function in the frontend, or the processing of orders via the backend. Of course, personal data can also be picked up via the RestAPI with authorisation. Anonymised data can of course also be recorded, for example to provide article recommendations ("Customers also purchased",

"Customers also looked at", etc.) and statistics in the backend. Since it is easy to lose the overview due to Shopware's innumerable functions, we would like to list for you the data that is stored in Shopware.

This list may be incomplete. Plug-ins can extend data storage and thus expand extensive areas in Shopware. Whether you process other personal data is something you must assess individually in your shop.

In Article 4 (1) of the GDPR, personal data is defined as follows: "Any information relating to an identified or identifiable natural person; an identifiable natural person is one who can be identified, directly or indirectly, in particular by reference to an identifier such as a name, an identification number, location data, an online identifier or to one or more factors specific to the physical, physiological, genetic, mental, economic, cultural or social identity of that natural person."

Customer data

In order to carry out an order and other actions in the shop, the customer requires a customer account. The contents include the address data, but also additional personal information, depending on what you have configured in your settings. Queries regarding date of birth or the details of the company name are both possible. Here you should specifically check the registration of your shop in order to obtain an overview of what data you are collecting from the customer. In the Shopware database, all the tables contain the customer data with the abbreviation "s_user*". But you can of course also consult this data in the backend by opening the customer in question at Customer > Customer.

Orders

Of course, the checkout is the heart of every online shop. In order to issue an order in Shopware, the customer first requires a customer account, of course. All this information is also stored together in the order. Then the products that the customer purchases in your shop are added, along with the customer's IP address. In addition, the referrer - that is, the source used by the customer to reach your shop - is also stored there. You can see all the relevant information via the backend at Customer > Orders, and in the database in the tables with the abbreviation "s_order". Further information for processing IP addresses can be found at: [Are Ip-addresses saved in Shopware?](#)

Bank details (SEPA)

Shopware's own SEPA function is seldom used today. Most shop owners use an external solution that does not store this data in Shopware, and only transfers the successful payment to Shopware (for example, PayPal Plus). If you use Shopware's own SEPA function, this also amounts to processing personal data, of course. Our SEPA function is explained here: [SEPA-Configuration](#)
The SEPA data entered by the customer is kept in the table s_core_payment_data.

IP addresses

As mentioned above, the IP addresses for individual orders are stored. In addition, a list of places where an IP address is stored is provided in this document: [Are Ip-addresses saved in Shopware?](#)

Newsletter

Like many other systems, Shopware provides a function for Newsletter registration. For example, the form can be found in the shop footer, or an expanded form is available via /newsletter. What data is collected here depends largely on the basic setting "Check extended fields in newsletter registration". Depending on the definition, additional information is collected here, such as the customer's first name and last name. Documents regarding the Newsletter can be found here: [Newsletter](#)

All the information that the customer stores in the frontend can be found in the administration interface in the "Marketing > Newsletter > Administration > Receiver" area. In the database, the tables `s_campaigns_mailaddresses` and `s_campaigns_maildata` are used to store the data. In addition, you can also define whether the double opt-in procedure is to be used when registering for the newsletter; for this purpose the basic setting "Double opt-in for newsletter subscriptions" is used. The opt-in data is stored in the table `s_core_optin`.

Forms

The forms in Shopware send an e-mail with the data entered by the customer to the address stored on the form. The shop owner can decide what data is requested there and this can vary and be customized. Thus you should review independently what data you process and how you do so in the backend under Content > Forms. The documents regarding the form function can be found at: [Forms](#)

Reviews

On the article details page, Shopware provides the opportunity to give a product review. This must be confirmed via the double opt-in if the basic setting "Double opt-in for customer reviews". The table `s_core_optin` is responsible for the opt-in procedure. The confirmed reviews can then be viewed, deleted or approved in the backend under Items > Customer reviews. Here you can find documents regarding the ratings function in Shopware: [Customer reviews](#)

Backend

The Administration interface is a bit special in this area, since you can of course view and set up customer records there, as well as administration users who are also linked by an e-mail address. Thus the backend not only processes the customers' data in your shop, but also the data of your staff, to a certain extent. You can configure the user administration of the backend under Configuration > User administration. It is also possible to restrict individual areas of the backend or to allow read-only access. Documents regarding the rights and roles in the backend can be found here: [User administration](#)

RestAPI

If you have connected additional systems, such as an ERP, this normally communicates with the shop using the RestAPI. The users of the RestAPI are set up via the backend as described above. Here you can find a list of the data that you can obtain via the RestAPI: [RestAPI](#)

Which data is stored anonymously?

Shopware uses anonymised content to provide your customers in the shop with individual cross-selling proposals. This concerns, for example, the functions "Customers also bought" and "Customers also

viewed". There the behaviour of the customers in your shop is analysed anonymously and used to display recommendations. You can manage these functions via the performance module in the backend. Additional documents regarding this area can be found at: [Cross selling](#)

Statistics

In the backend you can also consult statistics on the purchasing behaviour, orders and much more. Documents regarding the statistics can also be found in our documentation: [Analysis](#)

You can also manage the recording of statistics via the performance module, and the option to exclude individual IP addresses from documentation is also available in the basic settings. This data is retained from all the data recorded in the Shopware database.

What information is transferred in encrypted form?

The encrypted transfer of data is normally achieved via the HTTPs protocol. To be able to use this in your shop, you first require a valid SSL certificate that must be set up on your server. Then you can activate SSL in your shop by using these instructions: [How to configure and activate an SSL-certificate](#)

Here a general distinction must be made between encrypted transfer of data and encrypted storage. We always recommend operating all areas of the shop via HTTPs so that all personal data is transferred via HTTPs in encrypted form. This data is then stored in the relevant database tables. The data itself is only encrypted when this is required, e.g. the user password.

What information is stored in the user's browser?

Whether setting cookies that are technically not required for operating the shop makes an opt-in necessary or not is disputed among legal experts. We recommend that you implement one of the established solutions for information on data protection law regarding cookies. More information can be found at <https://www.cookiechoices.org>; please consult your legal adviser on this issue. Shopware itself already provides the opportunity to activate the cookie reference as default.

Cookies (session, SLT, CSRF)

Shopware stores cookies in the visitor's browser in order to guarantee the basic settings of the shop. Using the cookies enables, for example, the content of the shopping cart, the login status and also the CSRF protection. Shopware cannot be used without permitting cookies in the browser. IMPORTANT: Shopware only ever stores IDs in the customer's browser, the relevant information is assigned in the application area.

Session

By using the session cookie, Shopware decides whether the relevant user has an active shopping cart and whether the user is logged in. It therefore serves as identification between browser and server. No additional information is stored in the browser apart from the session ID. The sessions are handled and controlled via PHP on the server side and this must be seen as independent of Shopware.

CSRF

In addition, Shopware produces an individual CSRF cookie when the shop is visited so that the customer can navigate the individual areas of the shop. Here information on CSRF protection can be found: [CSRF protection hint](#)

SLT

One of the new features in 5.3 is the SLT cookie that enables the shop to recognise customers when they return to the shop, even if the session has already expired. All information on the SLT cookie can be found here: [Shopware login token](#)

The SLT cookie can also be deactivated in the basic settings.

Notepad

If a customer places a product on the notepad, a cookie with the name "sUniqueID" is created for this purpose in order to save the content of the notepad. The saved products are stored in the table s_order_notes.

Last viewed articles

In addition, the information on the "last viewed articles" is stored in the browser's local storage. You can find the documentation for this function here: [Last viewed](#)

Cookies overview

In this table you will find relevant cookies for the use of Shopware:

Cookie name	Function	Storage period
__csrf_token-1	Required for the validation of client details.	Is bound to the browser session.
session-1	Identifies the current session, the user and their shopping cart.	Is bound to the browser session.
cookiePreferences	The information about which cookies the customer wants is stored in a serialised string.	The information is stored for half a year.
slt	Allows the customer to be recognised when returning to the shop, even if the session has already expired.	Stored for one year or deleted by manual logout of the client.

sUniqueID	Is responsible for the allocation of the watch list and is used when using the watch list.	Stored for one year.
x-ua-device	Serves to determine the end device used, e.g. for the correct display of the shop.	Is bound to the browser session.
allowCookie	Saves the cookie settings of the shop customer.	Does not expire.
timezone	The time zone is saved, for example, for time information in e-mails.	
PHPSESSID	PHP session, which are used to assign the current session (shopping cart, etc.).	Is bound to the browser session.
csrf	All CSRF cookies are used to validate form entries in order to prevent XSS attacks.	Is bound to the browser session.
sw-cache-hash	Technical cookie for the cache functionality.	
cookie-preference	For the client's cookie preferences if these are configured in the offcanvas.	The information is stored for half a year.
sw-states	Technical cookie for the cache functionality.	

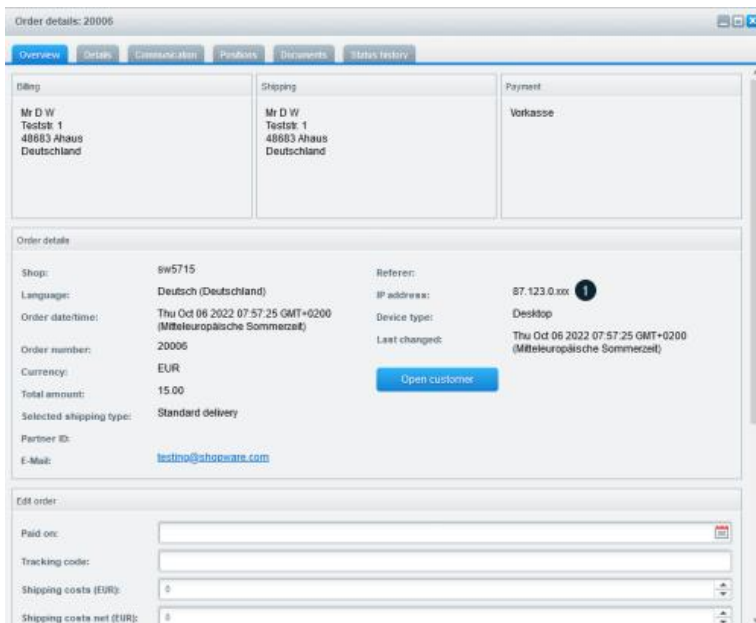
Which ip-addresses are saved in Shopware?

IP-addresses will be saved in a regular Shopware-installation. These are used to identify users in your shop in four different cases.

s_order

The ip-address of an order can be found in the order details. Therefore you open an order at **Customers > Orders**.

The ip-address is shown on the **Order Overview (1)**. The last three digits of the ip-address are anonymised.



The full ip-address can be found in the database table "s_order" in the column "remote_addr".

s_core_log

The database table "s_core_log" collects all activity by backend users. This table can be used to recreate, when specific changes were made. Based on the ip-address you might be able to specify, who made these changes.

s_statistics_pool

This data is used in the statistics, to calculate the visitors per day.

s_statistics_currentusers

The table s_statistics_currentusers is used to temporarily save data for the "Online visitors"-widget and is cleared every three minutes.

How can I integrate the data protection declaration into my shop?

Websites require a data protection declaration under the GDPR as well. Thus information must be provided on the underlying legal bases and on the requirement to conclude a contract or whether there is another obligation to provide the data. It is also necessary for you to find out in the data protection declaration how you are to handle personal data. The data protection declaration may be a shop page that you link to the relevant areas in your shop, for example. Here it is important that you provide this information "upon collection" of the personal data. Thus the data protection declaration should normally be accessible with only one click. This is typically performed in the footer and also in the checkout area. Here the documents regarding the Shop pages in Shopware can be found: [Shop pages](#)

You can normally adapt individual texts in the checkout or wherever you process personal data, by using the snippet management. The snippet management can be found here: [Snippets](#)

The forms also provide the possibility of integrating an additional checkbox: [Checkbox example](#)

We wish to point out that serious sanctions apply in cases of non-compliance with these duties to provide information. If there is no data protection declaration, processing is normally against the law. You may also be liable to pay a fine, and damages or injunctive relief may be claimed if the duties to provide information are fulfilled incorrectly. Consumer protection associations and data protection groups may also instigate proceedings against a website operator without a data protection declaration. Finally, competitors may also issue warnings.

How can I link the data protection declaration in the registration by checkbox?

You can simply go to Basic settings > Frontend > Login/registration to activate the option "Data protection conditions must be accepted via checkbox". In the registration process you will then find a checkbox that has an additional link to the data protection provisions. You can adapt the text and link it to the relevant shop page by using the snippet "RegisterLabelDataCheckbox". However, a separate checkbox is not absolutely necessary. It is also sufficient to refer to the data protection information in the ordering process (e.g. "I have read the data protection information."), if this is linked to the data protection notes.

Is there an option to integrate a data protection reference at all relevant places in the shop?

Due to the GDPR launch Shopware can display privacy notes at all relevant places in the frontend. As of Shopware 5.4.3 you'll find the related setting in the basic settings in Additional settings > Privacy.

What personal data is transferred to third parties by Shopware?

Shopware does not transfer any information to third parties by default. This may of course be done in addition through add-ons. For example, if you set up PayPal in your shop, data from the shop (the delivery address, the order amount and the shopping cart) will be transferred to PayPal. There are of course also a number of other service providers that process data from Shopware further if you set up an add-on of this kind in your shop. Some prominent examples here could be payment providers, ERP systems as well as newsletter service providers. The best way to find out what data is transferred to third parties here is to ask the manufacturer of the respective add-on.

How can I have personal data output in structured form?

Import/Export

The GDPR states that the shop owner must provide a customer with the data relating to him in structured form on request. Here Shopware provides the import/export function with which you can export all the relevant data of a customer. The module provides the possibility of setting up a customer stream for the relevant profiles (orders, customers, newsletter recipients). Here you can set up a customer stream in

advance with the customers involved and produce an export, for example. Please note that depending on the customer's registration (quick buyer or customer account) there may also be more than one customer account. Documents regarding the export possibilities (CSV and XML) can be found here: [Import/Export](#)

Database

The database also provides the option of providing the information to a customer via an SQL query.

How can I delete all personal data from my shop?

All the data defined in this document can normally be deleted conveniently via the backend. So if a customer asks you to delete all his personal data, you can simply delete this data by using the ordering and customer module in the backend. This automatically removes the various links as well. You can also use the newsletter module to remove the recipient from the list of recipients. Shopware thus provides the option of removing all personal data via the backend on the customer's request without problems. Whether and when you are obliged to delete customer data is a legal issue that you should clarify with your legal adviser. It is unfortunately impossible to make generalised statements on this point.

FAQs

To what extent is Shopware prepared for the new GDPR?

For some time now, we at Shopware have been working with well-known certification bodies to ensure that the system meets the requirements of the GDPR, which comes into force in May. It has been found that according to our current information, Shopware already provides shop owners with the necessary functions they need to create the settings required by the rules of the GDPR. Thus, for example, Shopware already provides all the required tools in the regular end user documentation, such as for removing personal data from the system, which is a core requirement of the new General Data Protection Regulation. No special plug-in/update is planned in relation to the GDPR.

Are the relevant functions installed automatically with software updates?

If it turns out to be necessary to adapt our software, we will of course provide the appropriate adaptation in the form of an update.

Starting at Shopware 5.4.3, GDPR related tutorials are no longer needed, as they have been integrated into the core of Shopware. You can find these options in your basic settings under "privacy".

How am I able to display a privacy policy note for newsletter registrations?

To insert a reference into your theme, you can use this tutorial. Before you do that, you should have the basic knowledge of making adjustments in the template. [Tutorial - Making changes in the template](#)

Implementation

In this file you have to make the first adjustment.

/themes/Frontend/YourTheme/frontend/index/footer-navigation.tpl

There you have to add the following code:

```
{extends file="parent:frontend/index/footer-navigation.tpl"}
{block name="frontend_index_footer_column_newsletter_form"}
  {smarty.block.parent}
  {s_name="IndexFooterNewsletterPrivacy"}I have read the <a title="data protection information"
{/block}
```

The second adjustment must be added in this file:

/themes/Frontend/YourTheme/frontend/newsletter/index.tpl

There you have to add the following code:

```
{extends file="parent:frontend/newsletter/index.tpl"}
{block name="frontend_newsletter_form_submit"}\
  {s_name="IndexFooterNewsletterPrivacy"}I have read the <a title="data protection information"
  {smarty.block.parent}
{/block}
```

For displaying the adjustments in the frontend you must compile your theme.

If the file is not yet included in your theme, you must create it yourself in the attached path.

This adjustment creates a new snippet **IndexFooterNewsletterPrivacy**, which can also be edited in the backend. The link in the snippet refers to the standard shop page "Privacy". If you use your own shop page, the link must be adjusted.

Frontend after the adjustment

Newsletter registration in the footer



The added reference **(1)** below the newsletter registration in the footer.

First name
Last name
Street and number
Zipcode City
The fields marked with * are required
I have read the data protection information ⓘ
Save >

Complete newsletter form

The reference **(1)** at the end of the newsletter form.

How am I able to display a privacy policy note for registrations?

You can use this tutorial to insert a note into your theme. Before you do that, you should have the basic knowledge of making adjustments in the template. [Tutorial - Making changes in the template](#)

Implementation

In the file: **/themes/Frontend/YourTheme/frontend/register/index.tpl** you have to add the following code. If the file does not exist in your theme, you have to create it manually.

```
{extends file="parent:frontend/register/index.tpl"}  
{block name='frontend_register_index_form_submit'}  
  <div class="register--privacy">  
    {s name="RegisterPrivacy"}I have read the <a title="data protection information" href="{url  
</div>  
    {$smarty.block.parent}  
  {/block}
```

Alternatively you can activate the checkbox for the data protection provisions under **Configuration > basic settings > Additional settings > Privacy > Data protection conditions must be accepted via checknox:**

This adjustment creates a new snippet **RegisterPrivacy**, which can also be edited in the backend. The link in the snippet refers to the standard shop page "Privacy". If you use your own shop page, the link must be adjusted.

Frontend after the adjustment

First name*
 Last name*
 Your email address*
 Your password*

Your password must contain at least 8 characters.
 The password is case sensitive.

Your address

Street and number*
 Zipcode* City*
 Country*

The shipping address does not match the billing address

The fields marked with * are required

I have read the [data protection information](#) **1**

The note **(1)** at the end of the registration.

How am I able to display a privacy policy note for evaluations?

You can use this tutorial to insert a note into your theme. Before you do that, you should have the basic knowledge of making adjustments in the template. [Tutorial - Making changes in the template](#)

Snippet

In the file: **/themes/Frontend/YourTheme/frontend/detail/comment/form.tpl** you have to add the following code. If the file does not exist in your theme, you have to create it manually.

```

{extends file="parent:frontend/detail/comment/form.tpl"}
{block name='frontend_detail_comment_input_actions'}
  <p>{s name="CommentPrivacy"}I have read <a title="data protection information" href="{url
    { $smarty.block.parent }
  }{/block}

```

For displaying the adjustments in the frontend you must compile your theme.


This adjustment creates a new snippet **CommentPrivacy**, which can also be edited in the backend. The link in the snippet refers to the standard shop page "Privacy". If you use your own shop page, the link must be adjusted.

Frontend after the Adjustment

Description Evaluations **0**

Customer evaluation for "Hauptartikel mit Grundpreisberechnung"

Write an evaluation

 Evaluations will be activated after verification.

Your name

Your email address*

Summary*

10 excellent

Your opinion

Please enter these characters in the following text field.

The fields marked with * are required.

I have read the [data protection information](#)

Save

Checkbox

You also can insert a checkbox. For that you have to add the following code in this file:
/themes/Frontend/YourTheme/frontend/detail/comment/form.tpl

```
{extends file="parent:frontend/detail/comment/form.tpl"}
{block name='frontend_detail_comment_input_actions'}
  <p>
    <input name="comment-checkbox" type="checkbox" id="commentcheckbox" required="required" />
    <label for="commentcheckbox" class="chklable">{s name="CommentPrivacy"}I have read the
  </p>
  {&#x7B;smarty.block.parent}
{/block}
```

Frontend after the adjustment

Description Evaluations 0

Customer evaluation for "Hauptartikel mit Grundpreisberechnung"

Write an evaluation

Evaluations will be activated after verification.

Your name

Your email address*

Summary*

10 excellent

Your opinion

Please enter these characters in the following text field.

The fields marked with * are required.

I have read the [data protection information](#)

Save

Tips and tricks

Under certain circumstances, it may not be necessary to collect individual personal data. If, for example, you do not want to save the customer's e-mail address, you can disable the Double-Opt-In procedure. You can disable this in the basic settings under **Storefront > E-Mail Settings > Double-Opt-In for article ratings**.

How am I able to display a privacy policy note for my forms?

You can use this tutorial to insert a note into your theme. Before you do that, you should have the basic knowledge of making adjustments in the template. [Tutorial - Making changes in the template](#)

Implementation

In the file: **/themes/Frontend/YourTheme/frontend/forms/form-elements.tpl** you have to add the following code. If the file does not exist in your theme, you have to create it manually.

```
{extends file="parent:frontend/forms/form-elements.tpl"}
{block name='frontend_forms_form_elements_form_submit'}
  {s name="SupportPrivacy"}I have read the <a title="data protection information" href="{url c
  {$smarty.block.parent}
{/block}
```

For displaying the adjustments in the frontend you must compile your theme.

This adjustment creates a new snippet **SupportPrivacy**, which can also be edited in the backend. The link in the snippet refers to the standard shop page "Privacy". If you use your own shop page, the link must be adjusted.

Frontend after the adjustment

Information

- About us
- Contact
- Help / Support
- Members login
- Newsletter
- Payment / Dispatch
- Privacy

Contact

Contact

Title*

First name*

Last name*

eMail-Address*

Phone

Subject*

Comment*

Please enter these characters in the following text field.

The fields marked with * are required.

I have read the [data protection information](#) **1**

Send >

The added note **(1)**.

Checkbox instead of a note

Alternatively you can insert a checkbox which must be checked to send out the form. Proceed as follows.

Create the **field (1)** in the form.

Edit form

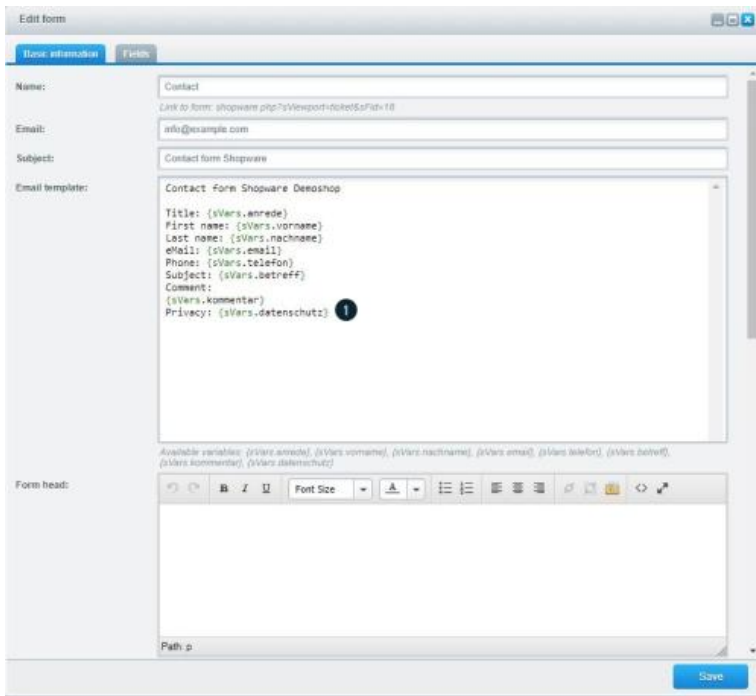
Kontaktverwaltung Fields

Add field You can sort the fields via drag & drop

Name	Description	Type	View	Options	Comment	Error message	Required
anrede	Title	select	normal	Ms.Mr			true
vorname	First name	text	normal				true
nachname	Last name	text	normal				true
email	eMail-Address	text	normal				true
telefon	Phone	text	normal				false
betreff	Subject	text	normal				true
komentar	Comment	textarea	normal				true
datenschutz	privacy	Checkbox	normal				false 1

Save Cancel

If you want to deactivate the checkbox initially, insert a "0" in the field "Options". For confirmation you can **output (1)** your input-value in the email.



Adjustment (1) in the frontend.

How am I able to display a privacy policy note for blog comments?

You can use this tutorial to insert a note into your theme. Before you do that, you should have the basic knowledge of making adjustments in the template. [Tutorial - Making changes in the template](#)

Snippet

In the file: **/themes/Frontend/YourTheme/frontend/blog/comment/form.tpl** you have to add the following code. If the file does not exist in your theme, you have to create it manually.

```
{extends file="parent:frontend/blog/comment/form.tpl"}
{block name='frontend_blog_comments_input_submit'}
  <p>{s name="BlogPrivacy"}I have read the <a title="data protection information" href="{url
    {Smarty.block.parent}
  {/block}
```

For displaying the adjustments in the frontend you must compile your theme.

This adjustment creates a new snippet **BlogPrivacy**, which can also be edited in the backend. The link in the snippet refers to the standard shop page "Privacy". If you use your own shop page, the link must be adjusted.

Frontend after the adjustment

Checkbox

You also can insert a checkbox. For that, add the following code in this file:
/themes/Frontend/YourTheme/frontend/blog/comment/form.tpl

```
{extends file="parent:frontend/blog/comment/form.tpl"}
{block name='frontend_blog_comments_input_submit'}
  <p>
    <input name="blog-checkbox" type="checkbox" id="blogcheckbox" required="required" aria-re
    <label for="blogcheckbox" class="chklabel">{s name="CommentPrivacy"}I have read the <a ti
  </p>
  {Smarty.block.parent}
{/block}
```

Frontend after the adjustment

Tips and tricks

Under certain circumstances, it may not be necessary to collect individual personal data. If, for example, you do not want to save the customer's e-mail address with the blog comments, you can disable the Double-Opt-In procedure. You can disable this in the basic settings under **Storefront > E-Mail Settings > Double-Opt-In for article ratings**.

How am I able to display a privacy policy note for email notifications?

You can use this tutorial to insert a note into your theme. Before you do that, you should have the basic knowledge of making adjustments in the template. [Tutorial - Making changes in the template](#)

Snippet

In the file: **/themes/Frontend/YourTheme/frontend/plugins/notification/index.tpl** you have to add the following code. If the file does not exist in your theme, you have to create it manually.

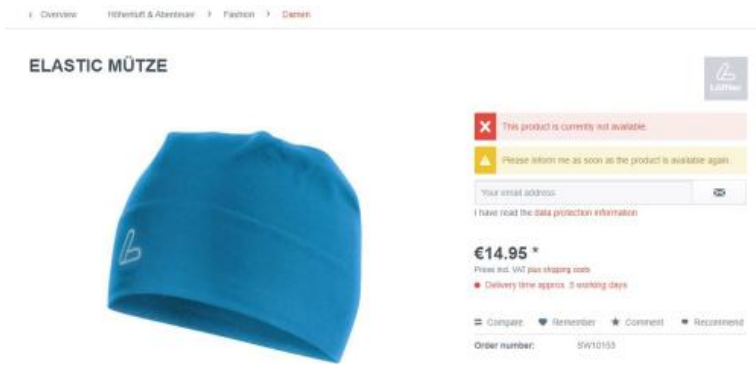
```
{extends file="parent:frontend/plugins/notification/index.tpl"}
{block name='frontend_detail_index_notification_button'}

{$smarty.block.parent}
  <p>{s name="NotificationPrivacy"}I have read the <a title="data protection information" href="#">
{/block}
```

For displaying the adjustments in the frontend you must compile your theme.

This adjustment creates a new snippet **NotificationPrivacy**, which can also be edited in the backend. The link in the snippet refers to the standard shop page "Privacy". If you use your own shop page, the link must be adjusted.

Frontend after the Adjustment

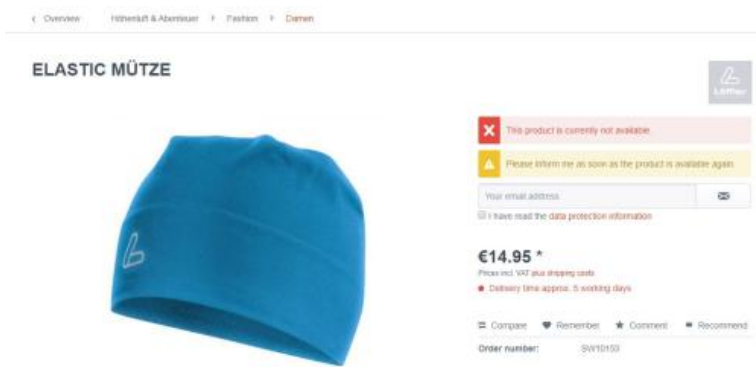


Checkbox

You also can insert a checkbox. For that you have to add the following code in this file:
/themes/Frontend/YourTheme/frontend/detail/comment/form.tpl

```
{extends file="parent:frontend/plugins/notification/index.tpl"}
{block name='frontend_detail_index_notification_button'}
{$smarty.block.parent}
<p>
  <input name="notification-checkbox" type="checkbox" id="notificationcheckbox" required="r
  <label for="notificationcheckbox" class="chklable">{s name="NotificationPrivacy"}I have r
</p>
{/block}
```

Frontend after the adjustment



Tips and tricks

Under certain circumstances, it may not be necessary to collect individual personal data. If, for example, you do not want to save the customer's e-mail address, you can disable the Double-Opt-In procedure. You can set this options in the basic settings under **Storefront> E-Mail Settings**.

Debugging error messages in Shopware

On this page you will find some useful information that will help you understand error messages and determine the cause.

How do I find the error message?

Errors can occur in many places for a variety of reasons, but sometimes they can not be found in the log file. So how do you find out where the error is caused?

If a serious error has occurred, first remember when the error occurred! Then see what steps you can take to reproduce the error and consider whether the error could come from Shopware itself or a plugin. Then you search for the message.

Frontend Error Reporter

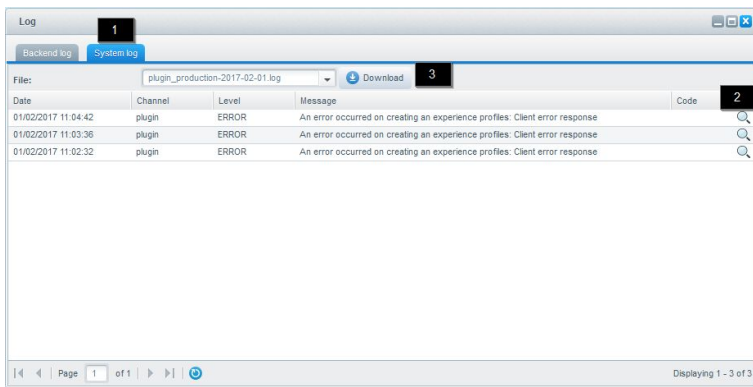
The Frontend Error Reporter is a possibility to display serious errors directly in the frontend instead of the "An error has occurred" page. Since the template displays the messages directly and thus also displays them to your customers, the use is only recommended if there is little traffic on the shop or you are working in a development environment. How to activate this advanced error debugging is explained under [Enhanced Error Debugging](#).

Log Files

Because the error reporter is disabled by default, the error messages will be written in a logfile on your server which you can find in `/var/log/`, for example **core_production-2016-02-01.log**. Depending on the environment you use (production, development) or from where the message comes (core, plugin) the file might be named different. **Note** that the file will be created for each day on which an error happens. When your error comes from a plugin, the logfile should be named "plugin_production-DATE.log", if it came out of shopware itself, the file should be named "core_production-DATUM.log". Just open the file from the date you search for and take a look for the exact timestamp, when your error happened. Here you should see an error message which describes the problem. How to read error messages we'll explain later in this article.

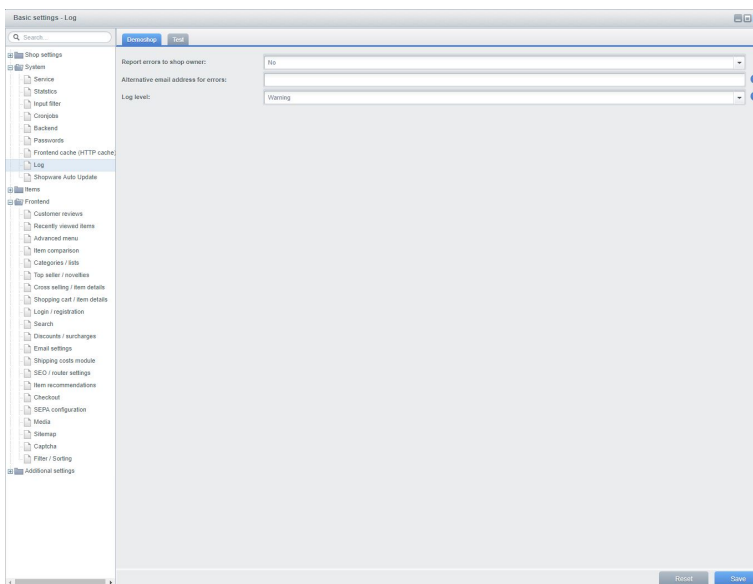
System-Log im Backend

Since the Shopware version 5.2.13 you can also see the messages of the system log in backend. You will find the new tab called "System log" (1) at Configuration > Logfile. Clicking on the magnifying glass (2) gives detailed information about the individual messages. In the "File" area (3) you can switch between the individual log files and also download log files, if you want to forward them, for example.



Sending errors via email

In the basic settings in *System > Log* you have the possibility to let shopware send you all the error messages.



How to read error messages?

In this passage we'll show you how to read error messages and how to extract the reason for the error.

Area	Notes for analysis
Stack trace	Here you can find informations of the most recently accessed functions. So you can find out at what point a function is called, which led to the error. Also plug-in names are often listed. This plugin should be the first place to start when you go into the error analysis.
Time	This information provides the option to further localize whether the error happens regularly or occur at regularly scheduled operations to this behavior.

Area	Notes for analysis
uri	Here, the referring URL will be displayed. With a manual call to the URL, you can test, for example, whether "only" the call of a no (longer) existent URL, for example, by a bot, the error has caused. Otherwise, the place of occurrence can be determined in detail.
query	Provides further informations whether the error happened in the frontend / backend , which controller has been addressed, etc.
shopId / shopName	If language- and / or subshops are used, you can see in which shop the error occurred.

PHP error messages

At PHP errors, you'll not see your shop, instead you see a blank error message in your browser like the default 503 or 500 error message or a blank white site. Debugging PHP errors is a bit more different than the others. To know, whats going wrong, you have 2 possibilities:

- Take a look into your servers error log folder, where it stores the php errors. The servers error log is not always stored at the same location, so ask your hoster where you can find that. If the error log provides not all messages you can ask your Hoster to increase the log-level, so that more errors will be logged.
- If you know, what you're doing, you can also jump to the considering function in the code, where the error occurs and insert a "die" with an error output to see, whats happening, But don't forget to reset the file to it's default after debugging!

Template errors

Template errors appear according to the error in the error reporter in the frontend or in the logfile and will look like this:

```
[2016-02-22 09:43:20] core.ERROR: exception 'SmartyException' with message 'Unable to load tem
Stack trace:
#0 /home/shopware/www/sw513/engine/Library/Smarty/sysplugins/smarty_internal_template.php(286)
#1 /home/shopware/www/sw513/var/cache/production_201602150844/templates/frontend_Responsive_de
#2 /home/shopware/www/sw513/engine/Library/Smarty/sysplugins/smarty_internal_templatebase.php(
#3 /home/shopware/www/sw513/engine/Library/Enlight/View/Default.php(274): Smarty_Internal_Temp
#4 /home/shopware/www/sw513/engine/Library/Enlight/Controller/Plugins/ViewRenderer/Bootstrap.p
#5 /home/shopware/www/sw513/engine/Library/Enlight/Controller/Plugins/ViewRenderer/Bootstrap.p
#6 /home/shopware/www/sw513/engine/Library/Enlight/Controller/Plugins/ViewRenderer/Bootstrap.p
#7 [internal function]: Enlight_Controller_Plugins_ViewRenderer_Bootstrap->onPostDispatch(Obj
#8 /home/shopware/www/sw513/engine/Library/Enlight/Event/Handler/Default.php(91): call_user_fu
#9 /home/shopware/www/sw513/engine/Library/Enlight/Event/EventManager.php(210): Enlight_Event_
#10 /home/shopware/www/sw513/engine/Library/Enlight/Controller/Action.php(201): Enlight_Event_
#11 /home/shopware/www/sw513/engine/Library/Enlight/Controller/Dispatcher/Default.php(523): En
#12 /home/shopware/www/sw513/engine/Library/Enlight/Controller/Front.php(226): Enlight_Control
#13 /home/shopware/www/sw513/engine/Shopware/Kernel.php(153): Enlight_Controller_Front->dispat
#14 /home/shopware/www/sw513/vendor/symfony/http-kernel/HttpCache/HttpCache.php(492): Shopware
#15 /home/shopware/www/sw513/engine/Shopware/Components/HttpCache/AppCache.php(255): Symfony\C
#16 /home/shopware/www/sw513/vendor/symfony/http-kernel/HttpCache/HttpCache.php(449): Shopware
#17 /home/shopware/www/sw513/vendor/symfony/http-kernel/HttpCache/HttpCache.php(349): Symfony\
#18 /home/shopware/www/sw513/engine/Shopware/Components/HttpCache/AppCache.php(178): Symfony\C
#19 /home/shopware/www/sw513/vendor/symfony/http-kernel/HttpCache/HttpCache.php(213): Shopware
```

```
#20 /home/shopware/www/sw513/engine/Shopware/Components/HttpCache/AppCache.php(114): Symfony\C
#21 /home/shopware/www/sw513/shopware.php(101): Shopware\Components\HttpCache\AppCache->handle
#22 {main} [] {"uid": "51a7455"}
```

The message says, that a template snippet could not be loaded. In 'frontend/index' the template tried to call 'frontend/index/headers.tpl' which is not existent or has insufficient permissions. So the first step is to look in the path to make sure, that the file is existent and owns the needed permissions. If the file is not existent, you have 2 possibilities: *Is the file missing?* or *Will the wrong file be called?* You can find out that by searching all files under '/frontend/index' for this call (in our example 'frontend/index/index.tpl') and compare this file with the same file from the default responsive theme. If the default theme calls the same file, you have this file missing. If the default theme calls another file, you did a mistake and called a wrong file. In our example we misspelled *headers.tpl* because it must be named *header.tpl*. You have to change this and save your file, after saving you should not get this error again.

MySQL errors

MySQL errors can appear at multiple actions and cause multiple reasons. For example: 2 backend users do the exact same action in the backend to the exact same time e.g. saving an item or a setting. This can cause for example the following error:

```
exception 'PDOException' with message 'SQLSTATE[23000]: Integrity constraint violation: 1062 D
```

Duplicate entry '185-38' in means that you try to save something, which already exists. This can be a setting or a value, according to what you currently do. As a possible solution you can try it another time or take a look in the database and delete the value which occurs the error.

If you make changes on the database, backup your environment first! Changing the database can cause in errors, so do your changes only if you know what you're doing!

Known error messages and their cause

In this section you will find error messages that are known to us from support requests and the community. There are also possible solutions and causes.

Common or "Oops! An error has occurred"

DemoShop

Home

Oops! An error has occurred!

We have been informed about the problem and try to solve it. Please try again within a short time.

"Oops! An error has occurred" is the default error page in the frontend, when a so called "503 - Service unavailable" error occurs. This is a script error, which can be caused by many reasons. In many cases, this are plugins or inconsistent data in your database. In this case it's necessary to turn on the [error debugging](#).

The provided X-CSRF-Token is invalid

DemoShop

Home

Ups! Ein Fehler ist aufgetreten!

Die nachfolgenden Hinweise sollten Ihnen weiterhelfen.

The provided X-CSRF-Token for path "/5220/account/login/Target/account/sTargetAction/index" is invalid. Please go back, reload the page and try again. In engine/Shopware/Components/CSRFTokenValidator.php on line 158

Stack trace:

```
#0 [internal function]: Shopware\Components\CSRFTokenValidator->checkFrontendTokenValidation(Object(Enlight_Controller_ActionEventArgs))
#1 engine/Library/Enlight/Event/Handler/Default.php(91): call_user_func(Array, Object(Enlight_Controller_ActionEventArgs))
#2 engine/Library/Enlight/EventManager.php(244): Enlight_Event_Handler_Default->execute(Object(Enlight_Controller_ActionEventArgs))
#3 engine/Library/Controller/Action.php(143): Enlight_Event_Manager->notify('Enlight_Controller_ActionEventArgs')
#4 engine/Library/Controller/Dispatcher/Default.php(52): Enlight_Controller_Action->dispatch('loginAction')
#5 engine/Library/Enlight/Controller/Front.php(23): Enlight_Controller_Dispatcher_Default->dispatch(Object(Enlight_Controller_Request_RequestHttp))
#6 engine/Shopware/Kernel.php(189): Enlight_Controller_Front->dispatch()
#7 vendor/symfony/http-kernel/HttpCache/HttpCache.php(88): Shopware\Kernel->handle(Object(Symfony\Component\HttpFoundation\Request), 1, true)
#8 engine/Shopware/Components/HttpCache/AppCache.php(25): Symfony\Component\HttpKernel\HttpCache->forward(Object(Symfony\Component\HttpFoundation\Request))
#9 vendor/symfony/http-kernel/HttpCache/HttpCache.php(238): Shopware\Components\HttpCache\AppCache->forward(Object(Symfony\Component\HttpFoundation\Request))
#10 vendor/symfony/http-kernel/HttpCache/HttpCache.php(275): Symfony\Component\HttpKernel\HttpCache\HttpCache->pass(Object(Symfony\Component\HttpFoundation\Request))
#11 engine/Shopware/Components/HttpCache/AppCache.php(133): Symfony\Component\HttpKernel\HttpCache\HttpCache->invalidate(Object(Symfony\Component\HttpFoundation\Request))
#12 vendor/symfony/http-kernel/HttpCache/HttpCache.php(286): Shopware\Components\HttpCache\AppCache->invalidate(Object(Symfony\Component\HttpFoundation\Request))
#13 engine/Shopware/Components/HttpCache/AppCache.php(114): Symfony\Component\HttpKernel\HttpCache\HttpCache->handle(Object(Symfony\Component\HttpFoundation\Request))
#14 shopware.php(117): Shopware\Components\HttpCache\AppCache->handle(Object(Symfony\Component\HttpFoundation\Request))
#15 {main}
```

exception 'Shopware\Components\CSRFTokenValidationException' with message 'The provided X-CSRF-Token is invalid. Please go back, reload the page and try again. In engine/Shopware/Components/CSRFTokenValidator.php on line 158'

Stack trace:
#0 [internal function]: Shopware\Components\CSRFTokenValidator->checkFrontendTokenValidation(Object(Enlight_Controller_ActionEventArgs))
#1 /var/www/vhosts/domain.com/httpdocs/shopware/engine/Library/Enlight/Event/Handler/Default.php(91): call_user_func(Array, Object(Enlight_Controller_ActionEventArgs))
#2 /var/www/vhosts/domain.com/httpdocs/shopware/engine/Library/Enlight/EventManager.php(244): Enlight_Event_Handler_Default->execute(Object(Enlight_Controller_ActionEventArgs))
#3 /var/www/vhosts/domain.com/httpdocs/shopware/engine/Library/Enlight/Controller/Action.php(143): Enlight_Event_Manager->notify('Enlight_Controller_ActionEventArgs')
#4 /var/www/vhosts/domain.com/httpdocs/shopware/engine/Library/Enlight/Controller/Dispatcher/Default.php(52): Enlight_Controller_Action->dispatch('loginAction')
#5 /var/www/vhosts/domain.com/httpdocs/shopware/engine/Library/Enlight/Controller/Front.php(23): Enlight_Controller_Dispatcher_Default->dispatch(Object(Enlight_Controller_Request_RequestHttp))
#6 /var/www/vhosts/domain.com/httpdocs/shopware/engine/Shopware/Kernel.php(176): Enlight_Controller_Front->dispatch()
#7 /var/www/vhosts/domain.com/httpdocs/shopware/vendor/symfony/http-kernel/HttpCache/HttpCache.php(88): Shopware\Kernel->handle(Object(Symfony\Component\HttpFoundation\Request), 1, true)
#8 /var/www/vhosts/domain.com/httpdocs/shopware/engine/Shopware/Components/HttpCache/AppCache.php(25): Symfony\Component\HttpKernel\HttpCache->forward(Object(Symfony\Component\HttpFoundation\Request))
#9 /var/www/vhosts/domain.com/httpdocs/shopware/vendor/symfony/http-kernel/HttpCache/HttpCache.php(238): Shopware\Components\HttpCache\AppCache->forward(Object(Symfony\Component\HttpFoundation\Request))
#10 /var/www/vhosts/domain.com/httpdocs/shopware/vendor/symfony/http-kernel/HttpCache/HttpCache.php(275): Symfony\Component\HttpKernel\HttpCache\HttpCache->pass(Object(Symfony\Component\HttpFoundation\Request))
#11 /var/www/vhosts/domain.com/httpdocs/shopware/engine/Shopware/Components/HttpCache/AppCache.php(133): Symfony\Component\HttpKernel\HttpCache\HttpCache->invalidate(Object(Symfony\Component\HttpFoundation\Request))
#12 /var/www/vhosts/domain.com/httpdocs/shopware/vendor/symfony/http-kernel/HttpCache/HttpCache.php(286): Shopware\Components\HttpCache\AppCache->invalidate(Object(Symfony\Component\HttpFoundation\Request))
#13 /var/www/vhosts/domain.com/httpdocs/shopware/engine/Shopware/Components/HttpCache/AppCache.php(114): Symfony\Component\HttpKernel\HttpCache\HttpCache->handle(Object(Symfony\Component\HttpFoundation\Request))
#14 /var/www/vhosts/domain.com/httpdocs/shopware/shopware.php(101): Shopware\Components\HttpCache\AppCache->handle(Object(Symfony\Component\HttpFoundation\Request))
#15 {main}

What does it mean?

Since Shopware 5.2.0 we use the so called X-CSRF-Token (cross sight request forgery). Using this token, we try to prevent actions, that are not done deliberately by the customer. When processing actions or exchanging data, we compare this token. Was the token not generated correctly or the token doesn't match, the error message occurs. In many cases it's hard to find the reason for the invalid token, because you need a reproducible workflow, where you can provoke the error, only using a reproducible workflow you can prove, that the error is really gone and not caused by coincidence. It also might happen, that the CSRF token is the cause, but not the error message, so we recommend to check the following things at debugging:

- **Individual theme:** Check if you also get the error message in the default responsive theme. Changes at data exchange points may cause invalid tokens.
- **Plugins:** Deactivate all Plugins and try to reproduce the error. If it's not reproducible activate all plugins step by step. If the error appears after activating a plugin, mostly the plugin causes the error.

problem.

- **Updates:** It's recommended to use the latest shopware version because we improve this security feature continuously.

Deactivating the X-CSRF-Token

When you're not able to find the errors cause, you can deactivate the whole feature temporarily. In some cases other error messages are hidden by the CSRF error message, so you can debug them after you know them. Deactivating this feature does not mean, that your shop is vulnerable. Shopware has security functions, which are working even then, but for security reasons you should solve the problem to reactivate the csrf protection. To deactivate the csrf protection you find the code snippet [here](#).

Integrity constraint violation

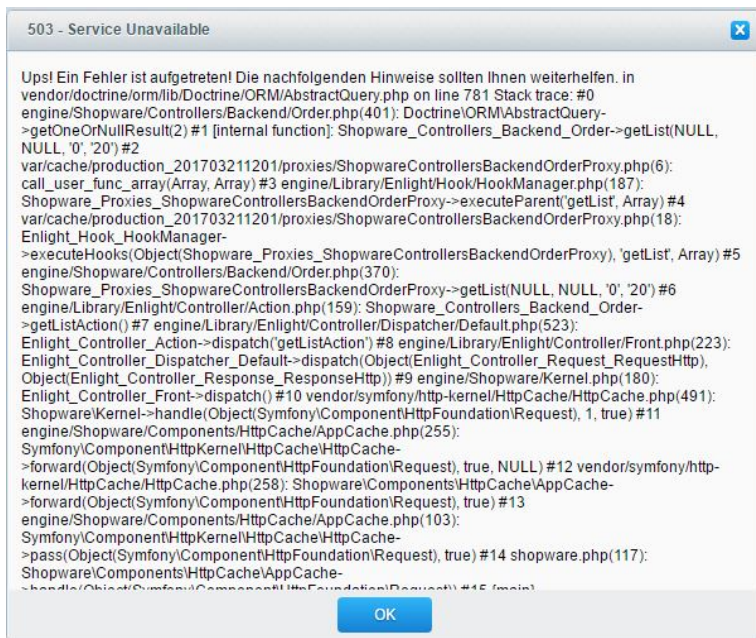
What does it mean?

SQL error messages can be caused by inconsistent or broken database records, caused by broken **Foreign Keys** or **duplicate entries**.

What can I do to solve it?

Try to repair the foreign keys in your database. Our [foreign key reparation tutorial](#) should help you. This fixes the realtions between the different Shopware core tables - but keep in mind, that plugin tables will not be repaired. You should also check duplicate entries in the mentioned table and delete them.

Order list cannot be opened



Oops! An error occurred! The following notes should help you. in vendor/doctrine/orm/lib/Doctr
#0 engine/Shopware/Controllers/Backend/Order.php(1038): Doctrine\ORM\AbstractQuery->getOneOrNu
#1 var/cache/production_201704210836/proxies/ShopwareControllersBackendOrderProxy.php(6): Shop

```
#2 engine/Library/Enlight/Hook/HookManager.php(186): Shopware_Proxies_ShopwareControllersBackend
#3 var/cache/production_201704210836/proxies/ShopwareControllersBackendOrderProxy.php(38): Enl
#4 engine/Shopware/Controllers/Backend/Order.php(271): Shopware_Proxies_ShopwareControllersBac
#5 engine/Library/Enlight/Controller/Action.php(159): Shopware_Controllers_Backend_Order->getL
#6 engine/Library/Enlight/Controller/Dispatcher/Default.php(523): Enlight_Controller_Action->d
#7 engine/Library/Enlight/Controller/Front.php(223): Enlight_Controller_Dispatcher_Default->di
#8 engine/Shopware/Kernel.php(182): Enlight_Controller_Front->dispatch()
#9 vendor/symfony/http-kernel/HttpCache/HttpCache.php(491): Shopware\Kernel->handle(Object(Enl
#10 engine/Shopware/Components/HttpCache/AppCache.php(266): Symfony\Component\HttpKernel\HttpC
#11 vendor/symfony/http-kernel/HttpCache/HttpCache.php(258): Shopware\Components\HttpCache\App
#12 engine/Shopware/Components/HttpCache/AppCache.php(103): Symfony\Component\HttpKernel\HttpC
#13 shopware.php(117): Shopware\Components\HttpCache\AppCache->handle(Object(Symfony\Component
```

The error message

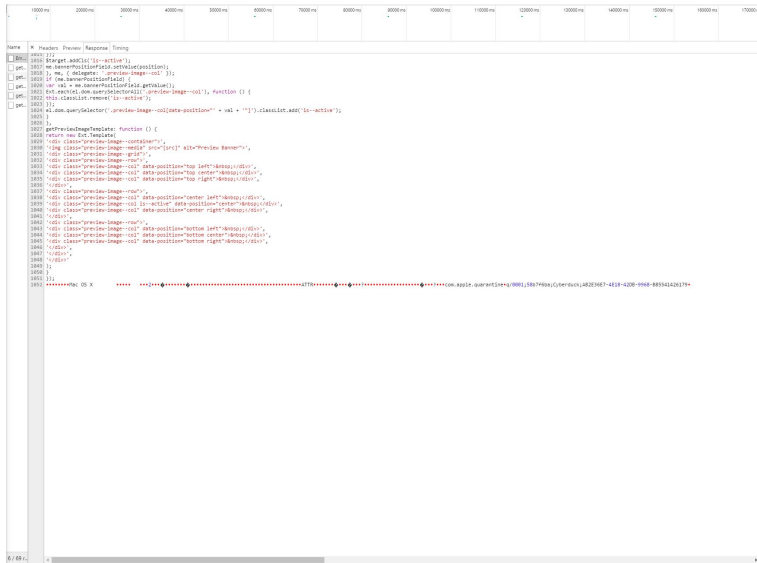
In most cases it's caused by duplicate order numbers - Shopware doesn't allow duplicate order numbers and by default and Shopware will never generate duplicate order numbers. Interfaces to your ERP or plugins can leverage this logic, in that case you should contact the vendor of the interface.

The solution

You can use our [example script](#) to find orders with duplicate order numbers.

Shopping worlds module cannot be opened

The error message



Open the developer tools in google chrome and open the shopping world module. Under "Network" you should find the XHR Request (Emotion?=XXXX). Go to preview and check, if there is the following code in the last line:

```
Mac OS X 2&#65533;&#65533;ATTR&#65533;&#65533;?&#65533;?com.apple.quarantineq/0001;58b
```

Reason / how to fix it

A not signed ftp client app on your Mac could be the reason. The os added the line to all files that was uploaded, because the app is in quarantine. Try to reupload the files with another signed program.

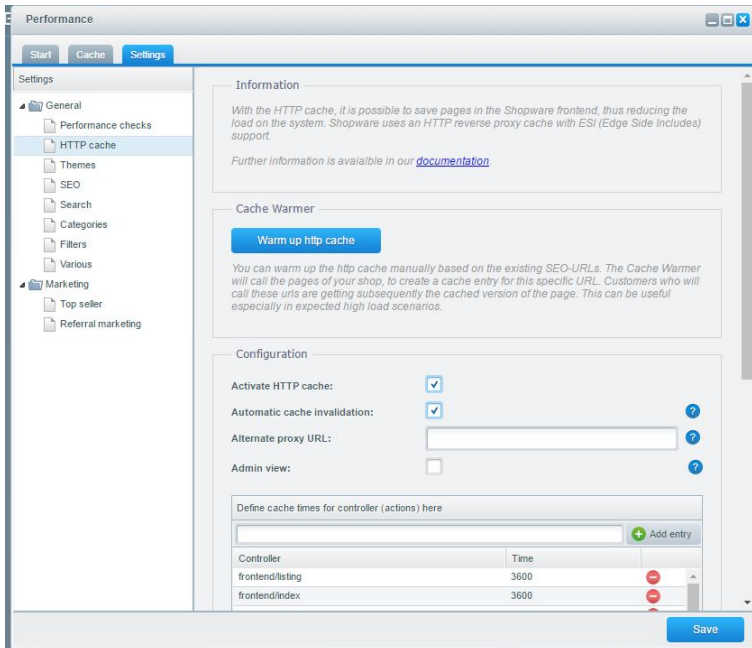
Error when saving the article: undefined

The meaning

When you get the error message "undefined" when saving an article in backend, there is probably a problem with invalidating the cache over the reverse proxy. You will see the following message in your server error log:

```
Fatal error: Call to a member function getId() on null in  
/var/www/html/engine/Shopware/Components/DependencyInjection/Bridge/Session.php on line 59
```

The meaning



First you have to note that this is caused by the server configuration of your hoster. Most of the time it's necessary to enter the "Alternate proxy URL" in **Configuration > Cache/Performance > Settings > HTTP Cache** which should be your domain. In some cases you have to enter a different proxy url, so please contact your hoster if you have questions.

Tweaks for the config.php

The config.php is the first configuration to load Shopware during execution. By adding new configuration snippets you can unlock many useful tools that will help you with debugging, among other things. You can view all available configurations in the "Default Config" at [Github](#).

Enhanced error debugging

You activate the Error Reporter in config.php by adding the following lines:

```
'front' => array(
    'showException' => true,
    'noErrorHandler' => false,
),
//Low-Level PHP-Fehler ab Shopware 5.2.0
'phpsettings' => [
    'display_errors' => 1,
]
```

Your config.php will look something like this:

```
<?php return array (
    'db' =>
    array (
        'host' => 'HOST',
        'port' => 'PORT',
        'username' => 'USER',
        'password' => 'PASSWORD',
        'dbname' => 'DBNAME',
    ),
    'front' => array(
        'showException' => true,
        'noErrorHandler' => false,
    ),
    //Low-Level PHP-Fehler ab Shopware 5.2.0
    'phpsettings' => [
        'display_errors' => 1,
    ]
);
```

After you have saved the file, the errors are displayed directly in the frontend.

Activating HTTP-Cache Debugging

Sometimes you want to check if the pages you call are really in the cache. Here you can use the following snippet, which activates the Symfony debug cache.

```
// Http-Cache
'httpCache' => [
    'enabled' => true, // true or false
    'debug' => true,
],
```

If you now open the [developer console](#) of your browser and look at the answer of the initial document request under "Network", you will see the following:

```
X-Content-Digest:enldac22cb10b9a66cb6590a7819384d54fcb637e838f65c40b655b07c85cd5f7d
X-Shopware-Allow-NoCache:price-1
X-Shopware-Cache-Id:/c3;
X-Symfony-Cache: GET /: miss, store;
    GET /?action=shopMenu&controller=index&module=widgets: fresh;
    GET /?controller=compare&module=widgets: miss, store;
    GET /?action=menu&controller=index&group=gLeft&module=widgets: miss, store;
    GET /?action=info&controller=checkout&module=widgets: miss, store
```

- **miss** = no Cache files available
- **fresh** = Controller / Page liegt in cache

Deactivate the CSRF-protection

If there are invalid tokens in the frontend, it may be helpful to temporarily disable this function.

Add the following lines of code to config.php:

```
'csrfProtection' => [
    'frontend' => false,
    'backend' => false
]
```

Save mails sent from Shopware as a file on the server

If you have problems sending mails, you don't know where to start at first. By activating the Mail-File Config you make sure that all mails are saved as a file.

```
'mail' => [
    'type' => 'file'
],
```

Set target path:

```
'mail' => [
    'type' => 'file',
    'path' => $this->DocPath().'mails'
],
```

Example configuration for developer

Here we deactivate all caches as well as the template cache. Thus changes are directly visible, without having to configure the theme.

Do not use this configuration in your production system, as it will massively affect performance.

```
'front' => [
    'showException' => true,
    'throwExceptions' => true,
    'noErrorHandler' => false,
],
//Zeige Low-Level PHP-Fehler
'phpsettings' => [
    'display_errors' => 1,
],
// Template-Cache
'template' => [
    'forceCompile' => true,
],
// Backend-Cache
'cache' => [
    'backend' => 'Black-Hole',
    'backendOptions' => [],
    'frontendOptions' => [
        'write_control' => false
    ],
],
```

```
// Model-Cache
'model' => [
    'cacheProvider' => 'Array' // supports Apc, Array, Wincache and Xcache
],
// Http-Cache
'httpCache' => [
    'enabled' => true, // true or false
    'debug' => true,
],
```

I need support with finding the reason / solution

Shopware provides many different ways to get help. First of all we offer a qualified support with our commercial version, with an active [software subscription](#) we support you in our technical department, where you'll get your answer within your promised reaction time. Our community [forum](#) is another way to get help by other users - please keep in mind that there is no guarantee to get an answer. The third way is to consult a [Shopware partner](#) which can help you debugging your problem.

Repair foreign keys

In certain constellations and also only with some hosters, it can happen that the foreign keys that shopware needs in the database are lost. The model generation of the attribute models needs these keys so that the attributes can be created correctly.

This can lead to error messages similar to this one:

```
[Semantical Error] line 0, col 114 near 'billing WHERE': Error: Class Shopware\Models\Attribut
Stack trace: ...
```

Please do the following steps:

- 1. Make a backup of the database.**
- 2. Download the file under "Available Downloads" on this page.**
- 3. Unzip the downloaded file on your local system**
- 4. Select your shopware database in PHPMysqlAdmin On the right side you will find the tab "Import". With this function you can select a "file to import". Select the downloaded file from your local computer. Confirm this process with (Okay) on the page.**
- 5. Delete the cache via FTP by selecting the folder "production_xxxx". (x = date of the used build) in the "var/cache" folder.**
- 6. Check the file rights of the folders. These need full read and write access**
- 7. Call up the front end in the browser again.**

Done

Error handling

...CONSTRAINT FAILED... FOREIGN KEY (`basketID`)

If you receive an error message similar to this one when executing the SQL file, there is corrupt data in the database:

MySQL reports:

```
#1452 - Cannot add or update a child row: a foreign key constraint fails (`result 2 when expla
```

You can remove them with the following statement. Execute the statement in phpMyAdmin:

```
DELETE FROM s_order_basket_attributes WHERE basketID NOT IN (  
    SELECT id FROM s_order_basket  
);
```

Now restart at step 4 and execute the SQL file again.

...CONSTRAINT FAILED... FOREIGN KEY (`categoryID`)

If you receive an error message similar to this one when executing the SQL file, there is corrupt data in the database:

MySql reports:

```
#1452 - Cannot add or update a child row: a foreign key constraint fails (`****`. `#sql-ac5_2d1
```

You can remove them with the following statement. Execute the statement in phpMyAdmin:

```
DELETE ca -- SELECT ca.*  
FROM `s_categories_attributes` ca  
LEFT JOIN s_categories c  
ON c.id = ca.categoryID  
WHERE c.id IS NULL
```

If you get the messages with another table, you have to adjust the query.

Example:

```
DELETE ca -- SELECT ca.*  
FROM `s_MY_TABLE_attributes` ca  
LEFT JOIN s_MY_TABLE c  
ON c.id = ca.categoryID  
WHERE c.id IS NULL
```

...CONSTRAINED FAILED... FOREIGN KEY ('articleID')

mySql reports:

```
#1452 - Cannot add or update a child row:  
a foreign key constraint fails
```

```
(`xxxxxx`.`xxxxxxxxxxx`,  
CONSTRAINT `#xxxxxxxxxxxxx`  
FOREIGN KEY (`articleID`)  
REFERENCES `s_articles` (`id`)  
ON DELETE CASCADE ON UPDATE NO ACTION)
```

Solution:

```
DELETE ca -- SELECT ca.*  
FROM `s_articles_attributes` ca  
LEFT JOIN s_articles c  
ON c.id = ca.articleID  
WHERE c.id IS NULL
```

Available downloads

A backup of the database should be created before the repair.
The current file is compatible with Shopware 5.6.2 and higher. Make sure that you also use the older Foreign Key repair for an older version.

Current Foreign Keys repair

[Download Foreign Keys from Shopware 5.6.2](#)

Foreign Key repair for older Shopware versions

[Download Foreign Keys till 5.2.13](#)

[Download Download Foreign Keys from 5.2.13](#)

[Download Foreign Keys from Shopware 5.5](#)

Installation / update via shell

Shopware can not only be installed via browser, it can also be installed and updated by shell to install and update your instances automatically or scripted. In the following article we describe how that works.

Installation

Upload and unzip installation package

At first, the install package must be uploaded via FTP, unzip it afterwards. The Database has to be created as well, so make sure, that you also have a usable database. If not, just create one.

Installation via wizard

The installation begins with the following command (assuming, that all commands are executed from the Shopware root directory!):

```
// Start installation
php recovery/install/index.php

// Getting help
php recovery/install/index.php --help
```

Press ENTER to start the installation:

At first, lets setup the database, the following data is needed:

Press Enter to use the recommendations (in Clamps), otherwise provide your desired values. If needed you can create a new database here, if you have the permissions to.

After providing all values, the database install starts:

After the database is set up, the basic informations like localization, host, path, name, email & currency are needed:

After this it goes on with the backend admin data:

After the admin user is created, you'll be asked for the license. You can choose between "ce" for installing a CE or "cm" for a commercial license, in this case you have to provide your license to continue:

After you provided the license, the installation has finished.

One line installation

To accelerate the installation or for scripting, a wizard is surely not useful, therefore we show you how you can install Shopware with one command:

```
php recovery/install/index.php --no-interaction --quiet --no-skip-import --db-host="HOST" --db
```

"--no-interaction" means no interaction, the installer won't ask you anything, it only uses the provided data. "--quiet" installs Shopware in the background. "--no-skip-import" imports the database no matter if the database is still stuffed.

For localisation you can choose between the following:

- de_DE => Germany
- en_GB => United Kingdom

At currencies, the following are available:

- EUR => Euro

- USD => US Dollar
- GBP => British Pound

Update

Upload and unzip package

Just upload the update package and unzip it, don't forget to overwrite all files.

Update via wizard

The wizard update can be started with the following command:

```
php recovery/update/index.php
```

The updater starts:

With ENTER you start the update:

After the update has finished, the updater tells you to delete the /update-assets folder to remove the maintenance mode:

You can delete the /update-assets folder with the following command:

```
// update-assets löschen  
rm -r update-assets/
```

One line update

Also the update can be processed by one command to script it. Therefore the following attributes can be used:

- --quiet => Processing in the background
- --no-interaction => No interaction

The command to update in one step is:

```
php recovery/update/index.php --no-interaction --quiet && rm -r update-assets/
```

Internationalization

Internationalization is an important thing to reach the target in different countries. This guide helps you to internationalize your shop by showing, of what you have to take care.

The way i wanna go / What do i have to consider?

Before making the first steps into internationalisation you should bother, where you wanna go and what is especially important to get there. A few examples:

- Which payment methods are preferred in foreign countries, do I need a new payment interface?
- How is the VAT situation in foreign countries, are there some special cases to take care of, e.g. rebated VAT for books and food in Germany or taxfree billing to Switzerland.
- Do i have to use different currencies and can my current payment interfaces handle them?
- Translation: Can I implement existent solutions or do I have to translate myself?
- Do I want to serve my foreign country customers different prices?
- Can i ship in foreign countries or do I need a new shipping interface?
- How much should the shipping cost for different foreign countries?

Additionally you can find the best tools to sell internationally here: [Shopware Store - The best tools to sell internationally](#)

Let's internationalise!

Activate foreign countries

The first step is [activating foreign countries](#), that customers from this countries can register in your shop. Take care, that the countries are assigned to the right area, this will be necessary in the VAT rules later.

Create categories

You can translate the categories directly for your languages in the backend under **Articles > Categories** using the globe.

Create currencies

To sell in foreign countries, you may need different currencies, how to create and configure them you can find out [here](#).

VAT

Because foreign countries may use different VAT rates, you have to configure the VAT rules for the desired countries, how you can do that, you can read [here](#).

Create the shop

Depending on the decision whether you want to create a language shop or a subshop with a different URL, you have to create the shop. You'll find the How-To here:

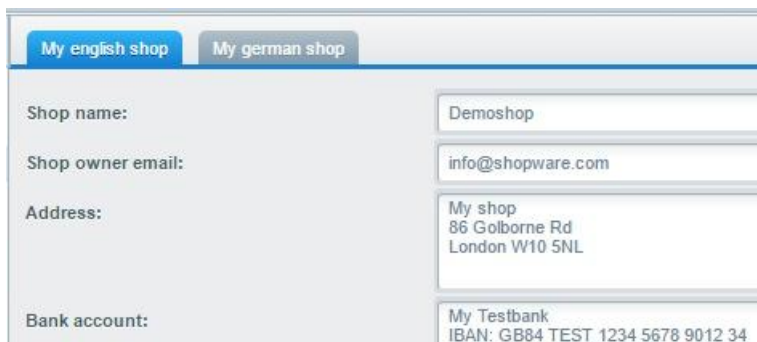
[Tutorial for creating Subshops / Language Shops for Shopware 4 - Shopware 5.1.x](#)

Pay attention to set the right localisation and as the circumstances require "Adopt translation" and also the configuration of the correct category.

Different configuration for your shop

Eventually you want to configure the shop another way your main shop is, e.g. different address formats or if your payment method needs other login data for the new country. Also this has to be considered.

The different configuration is available, when the new shop is created and the Backend is reloaded. You'll notice it on the separate tabs in the different configuration modules:



My english shop | My german shop

Shop name: Demoshop

Shop owner email: info@shopware.com

Address: My shop
86 Golborne Rd
London W10 5NL

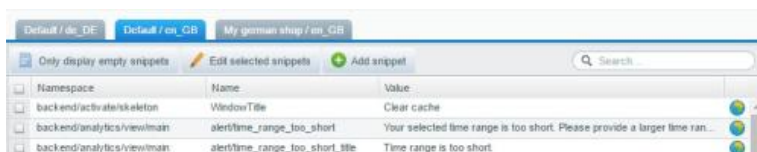
Bank account: My Testbank
IBAN: GB84 TEST 1234 5678 9012 34

Configurations from Dropdown fields can be inherited from the main shop or configured different.

Translations

Snippets

By default Shopware delivers Snippets for german and english. If needed, you can add other languages anytime to Shopware by translating yourself or use a [translation plugin](#) from our store.

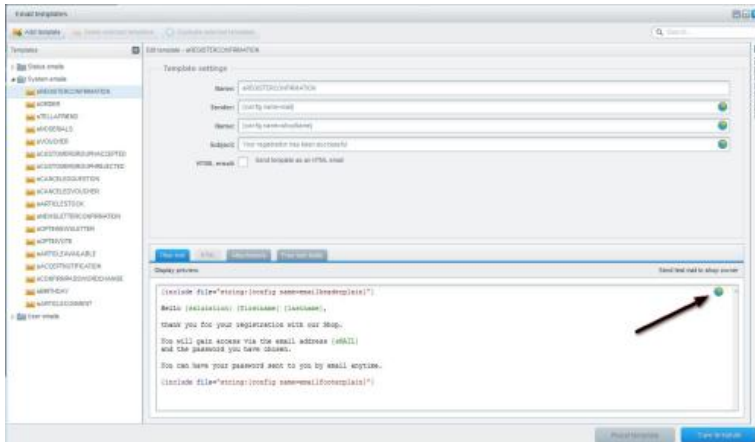


Namespace	Name	Value
backend/activate/skeleton	WindowTitle	Clear cache
backend/analytics/view/main	alerftime_range_too_short	Your selected time range is too short. Please provide a larger time ran...
backend/analytics/view/main	alerftime_range_too_short_title	Time range is too short

Did you know, that you can suggest snippet changes and translate Shopware using the localisation management platform crowdin? Just visit us: [Shopware at crowdin](#)

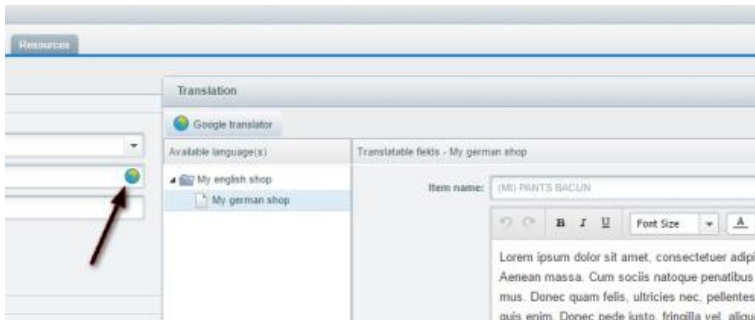
Emails

To internationalise your shop successful, you also need to translate your emails. You can do this in the email template module in "Configuration > Email templates" by choosing your email and the desired text (plaintext or HTML-text) and click on the blue globe, hereby you can translate your emails as known from all other modules.



Translating items

Beside snippets you also have to translate your items, you can do this by clicking the blue globe in the upper right corner of the necessary fields.



The following popup let you translate the content for every language- or subshop. Dont forget to save your changes!

The following item based data is translatable:

- Item name
- Description
- Meta-Information
- Free text fields (if used)
- Property sets / groups / options
- Meta Information of images
- Variant groups / options

Testing

If all necessary things are configured, you can start to test your internationalisation, here you can test, if everything is translated and especially if all of your payment- and shipping interfaces work the right way. If your test passes, you have successfully internationalised your shop!

Subsequent change of VAT rate

We provide the plugin VAT. Change / Reduction in our Community Store, which you can use to make the necessary adjustments for the VAT change. This can be found [here](#).

If you want to make the adjustments manually, this tutorial will explain what you should consider when changing the VAT rate.

Whenever you make changes in this area, we generally recommend that you first create a backup of the database! This provides you the possibility to return to the current status at any time.

Adjusting the tax rate

In the first step, you should be aware that the price of the item will change if you adjust the tax rate. You can now decide whether you want to adjust the **gross price** and the **net price** of the item remains the same, or whether you want to adjust the net price so that the gross price remains the same for the customer.

In the basic settings under Items > Other VAT rates you can define your own VAT rates for vouchers and discounts.

Adjust gross price (variant 1)

If you want the net price of the item to stay the same, and you want to adjust the gross price, you just have to change the tax rate in the backend. To do this, open the menu item **Configuration > Basic settings > Shop setting**.



The screenshot shows a 'Details' form for configuring a VAT rate. It includes a 'Name' field with the value '19%', a 'Default tax' field with the value '19', and a 'Tax regulations' section. The 'Tax regulations' section has a '+ Add entry' button and a dropdown menu set to 'Default'. Below this is a table with columns for 'Name', 'Area', 'Country', 'State', and 'Tax'. The table currently shows 'No entries'.

Name	Area	Country	State	Tax
No entries				

Change the default tax rate to the desired tax level (Ex. 16). Since Shopware stores the net prices in the database, the new gross prices are now automatically displayed and calculated in the frontend and backend.

In the order overview, the name of the selected tax rate is displayed in the column Taxes on the tab Items. If you change the name of the tax rate, the name of the tax rate as it is now called will also be displayed in older orders under Taxes. (For example 16%)

Adjust gross price (variant 2)

Another way to adjust the taxes so that the net price is maintained is to create a completely new tax rate. You can do this in the backend under **Configuration > Basic Settings > Shop Settings > Taxes**. Afterwards you can use the following SQL query to ensure that the new tax rate is assigned to the desired articles.

```
UPDATE `s_articles` SET s_articles.`taxID` = '2' WHERE `s_articles`.taxID='1';
```

In this example the tax rate with TaxID 2 is deposited for all articles, which use the tax rate with TaxID 1. You can find the exact TaxID in the database table: s_core_tax.

If you have stored fixed tax rates in a shipping method or in a voucher, you should also replace them with the new tax rate.

Additionally, you should check your plug-ins to see if they use the default tax rate. If necessary adjust the tax rate here as well.

Adjust net price

In Shopware the net prices are stored in the database. If you want to keep the gross price for the customer and thus adjust the net prices, changes in the database are necessary.

Since the gross price remains the same here, the net price is changed here. If you have customer groups for whom the net price is displayed in the front end, this price changes accordingly.

If you have stored tax rules for other countries in your standard tax rate, you should note that these rules refer to the net price of the items. If you have set up a tax rule of 20% for Austria, these will be calculated on the changed net price. The gross price for Austria will then change in your shop.

First, adjust the respective tax rate in the backend under **Configuration > Basic Settings > Shop Settings > Taxes** as explained in the step adjust gross price.

If you only have items with one tax rate, you can use the following examples as a guide. First divide by the new tax rate and then multiply by the old tax rate.

```
UPDATE s_articles_prices SET price = price/1.new_tax_rate*1.old_tax_rate
```

Examples

Tax reduction

As an example, the old tax rate of 19% is changed to the new tax rate of 16%.

```
UPDATE s_articles_prices SET price = price/1.16*1.19
```

Tax increase

This query is reversed for the increase of the tax rate from 16% to 19%:

```
UPDATE s_articles_prices SET price = price/1.19*1.16
```

Consider different tax rates

If you have items from more than one tax rate, but you only want to adjust one of the tax rates, execute this SQL query instead:

```
UPDATE s_articles_prices, s_articles SET s_articles_prices.price = s_articles_prices.price/1.19*1.16 WHERE (`s_articles`.id = `s_articles_prices`.articleID AND `s_articles`.taxID='3')
```

In this example the tax rate is changed from 19% to 16%. This happens for all articles that use the tax rate with TaxID 3. The exact TaxID can be found in the database table: `s_core_tax`.

How not to change net retailer prices

If you have maintained net prices for the retailer customer group in addition to the normal shop customer prices and do not want to change them, you can perform an additional check for the "pricegroup" in the SQL query.

A query that only changes the prices for customer group "EK" could look like this:

```
UPDATE s_articles_prices, s_articles SET s_articles_prices.price = s_articles_prices.price/1.19 WHERE (`s_articles`.id = `s_articles_prices`.articleID AND `s_articles`.taxID='3' AND `s_articles`.customerGroup='EK')
```

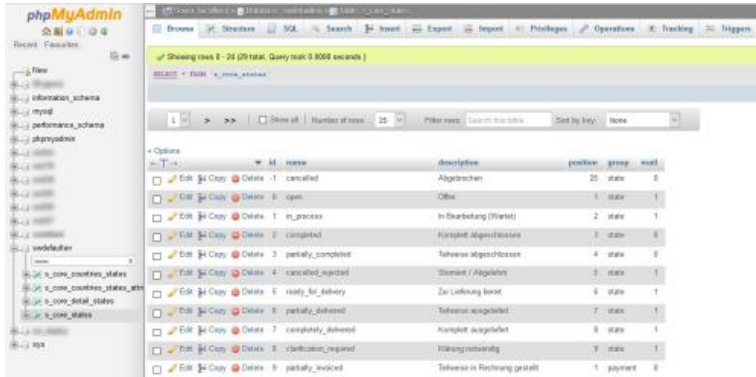
This query can be adjusted accordingly to change the price for other customer groups. You can find an overview of the existing customer groups in the backend in the basic settings under Shop Settings > Customer Groups. Alternatively, you can also view the existing customer groups in the database in the table `s_core_customergroups`.

Create new order and payment status

In this article we describe how to create your own order or payment state and sending emails by using it.

Creating a status

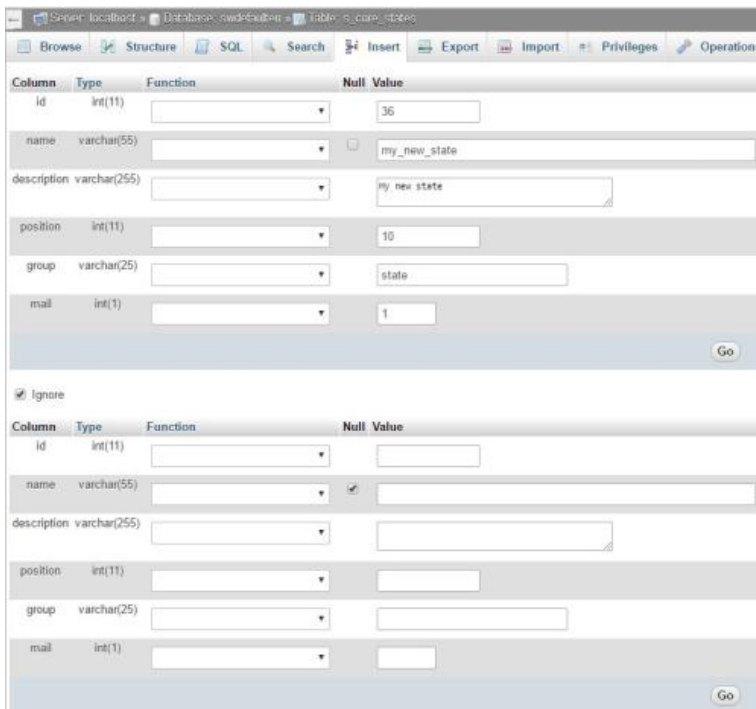
At first you have to create the state, therefore open your phpMyAdmin and your database table "s_core_states":



The screenshot shows the phpMyAdmin interface for the 's_core_states' table. The table structure is as follows:

ID	name	Description	position	group	mail
1	cancelled	Abgebrochen	25	state	0
2	open	Offen	1	state	1
3	in_process	In Bearbeitung (Wartet)	2	state	1
4	completed	Komplett abgeschlossen	3	state	0
5	partially_completed	Teilweise abgeschlossen	4	state	0
6	cancelled_expected	Storniert / Abgelehnt	5	state	1
7	ready_for_delivery	Zur Lieferung bereit	6	state	1
8	partially_delivered	Teilweise ausgeliefert	7	state	1
9	completely_delivered	Komplett ausgeliefert	8	state	1
10	clarification_required	Klärung notwendig	9	state	1
11	partially_involved	Teilweise in Rechnung gestellt	1	payment	0

Now have a look, which ID you can use, by default "36" is the first free ID which you can use for your state and then input a new state:



The screenshot shows the phpMyAdmin interface for the 's_core_states' table. The table structure is as follows:

Column	Type	Function	Null	Value
id	int(11)			36
name	varchar(55)		<input type="checkbox"/>	my_new_state
description	varchar(255)		<input type="checkbox"/>	My new state
position	int(11)			10
group	varchar(25)			state
mail	int(1)			1

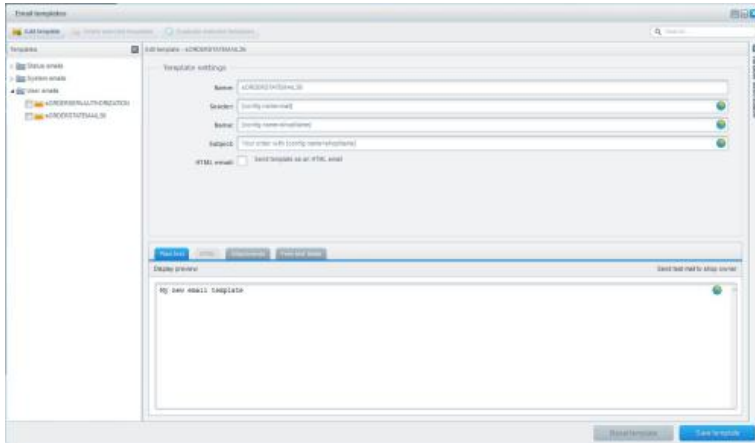
Below the main form, there is an 'Ignore' section with the following fields:

Column	Type	Function	Null	Value
id	int(11)			
name	varchar(55)		<input checked="" type="checkbox"/>	
description	varchar(255)		<input type="checkbox"/>	
position	int(11)			
group	varchar(25)			
mail	int(1)			

- **name** is an internal name
- **description** is the official name of the state which the user will also see.
- **position** describes the position of the state in the list of the backend module.
- **group** chooses the affinity of the state, "state" for an order state and "payment" for a payment state.
- **mail** defines, if a mail will be sent when this state will be used for an order (1) or not (0).

Creating email template

The status is created, now you just have to create the email template for it, To do this, open the email template module and create a new mail with the name "sORDERSTATEMAIL"+ "ID", in our example "sORDERSTATEMAIL36" and fill the mail with the desired content:



Now clear the cache and reload the backend.

Provide status name in the backend

First you open the snippet management module and create a new snippet in the "namespace backend/static/payment_status" for payment status or "backend/static/order_status" for order status.

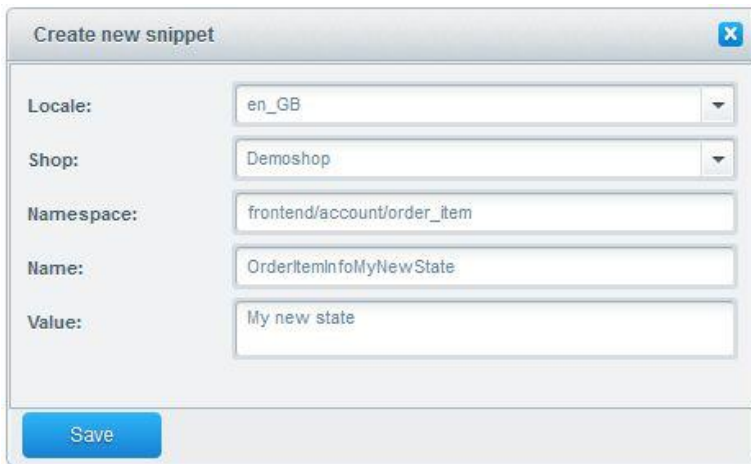
For example "MyNewStatus" as the technical name of the status in the database with your wanted name in the value:

Locale:	de_DE
Shop:	Demoshop
Namespace:	backend/static/payment_status
Name:	MyNewStatus
Value:	My New Status

Implement the state in the theme

At first, open the snippet management module and create a new snippet in the namespace "frontend/account/order_item", e.g. "OrderItemInfoMyNewState" which you have to fill with your desired

text:



Now you can provide the snippet in the frontend. To do this, create a template file named "order_item.tpl" in the directory /themes/Frontend/YOUIROWNTHEMENAME/frontend/account/ on the server. Insert the following code to integrate your own state:

```
{extends file='parent:frontend/account/order_item.tpl'}  
{block name="frontend_account_order_item_status_value"  
<div class="column--value">  
<span class="order--status-icon status--{$offerPosition.status}"></span>  
{if $offerPosition.status==0}  
{s name="OrderItemInfoNotProcessed"}{/s}  
{elseif $offerPosition.status==1}  
{s name="OrderItemInfoInProgress"}{/s}  
{elseif $offerPosition.status==2}  
{s name="OrderItemInfoCompleted"}{/s}  
{elseif $offerPosition.status==3}  
{s name="OrderItemInfoPartiallyCompleted"}{/s}  
{elseif $offerPosition.status==4}  
{s name="OrderItemInfoCanceled"}{/s}  
{elseif $offerPosition.status==5}  
{s name="OrderItemInfoReadyForShipping"}{/s}  
{elseif $offerPosition.status==6}  
{s name="OrderItemInfoPartiallyShipped"}{/s}  
{elseif $offerPosition.status==7}  
{s name="OrderItemInfoShipped"}{/s}  
{elseif $offerPosition.status==8}  
{s name="OrderItemInfoClarificationNeeded"}{/s}  
{elseif $offerPosition.status==36}  
{s name="OrderItemInfoMyNewState"}{/s}  
{/if}  
</div>  
{/block}
```

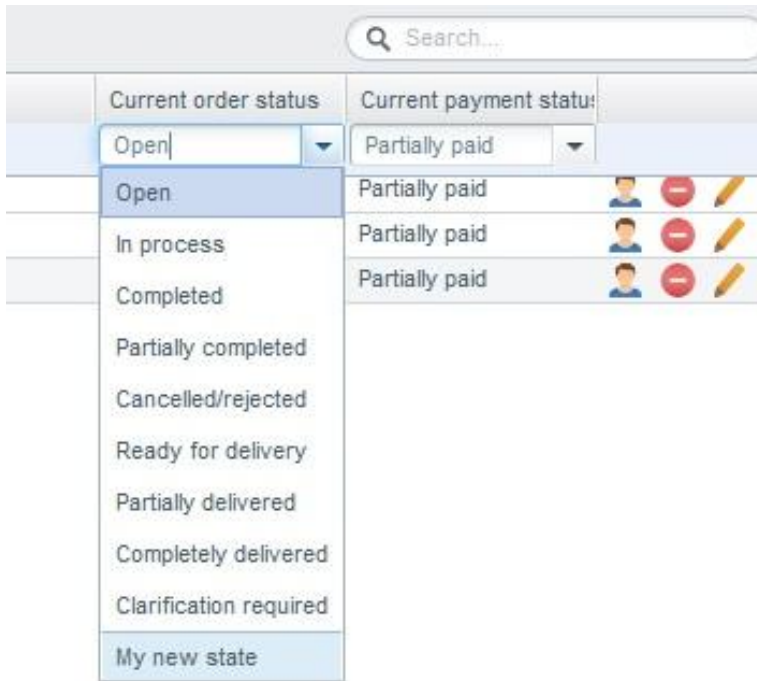
To display the corresponding icon next to the status it's necessary to create your own "all.less" file. The LESS code will be compiled into normal CSS by the LESS compiler. Create your all.less in the following directory: /themes/Frontend/EIGENERTHEMENAME/frontend/_public/src/less

Please enter the following code:

```
.order--status-icon {  
  position: static;  
  
  &.status--36 { //My New State  
    background: @highlight-info;  
  }  
}
```

Testing

After this you can start testing your new state:



And also in the frontend:

Orders

Here you can find your orders sorted according to date.

Date	Order number	Shipping type	Order status	Special offers
4 Oct 2016 03:24:00 PM	20005	Standard shipping	My new state	View

Page 1 From 1

Performance Tips

General tips and tricks

subject	standard	optimum	info
----------------	-----------------	----------------	-------------

PHP version	-	7.4.x	PHP 7.4.x, depending on the circumstances, this version offers a possible performance increase of up to 25-30% (PHP stack). Furthermore, the Memory_Limit - depending on the number of plugins and data sets used - should be at least 512MB.
Bytecode cache	APC	ZendOpcache + APCu	Installing and configuring a Bytecode cache is more than recommended - it will likely increase PHP stack execution by an additional up to 25%.
MySQL	MySQL 5.7.x	MySQL 5.7.x or MariaDB derivate	If possible you should make use of the latest major productive MySQL version available. This will positively affect the execution of joins and subselects and read accesses. MySQL should be used in the current major version released for production purposes, as there have been significant improvements in the execution of joins, subselects and read accesses to the database.
webserver	Apache 2.2 or 2.4	Nginx	If your shop generates particular large amounts of traffic, you might consider an alternate webserver such as e.g. Nginx. Make sure, that the mod_rewrite module is enabled and that there is a possibility to change webserver settings in .htaccess.
Debugging / profiling	-	-	Make sure there are no profiling tools like xDebug or XhProf active and running on your server. Those will negatively affect PHP execution. The same is true for any Shopware debugging & benchmark plugin. So you should never use them in a productive environment.

With every new iteration of Shopware there are performance improvements to be expected. Always keeping your software up-to-date should be your admin's first priority.

Choosing an adequate hoster

A hoster should resemble the aspired range of expertise in its offer. There should be an equally fitting upgrade path available, so that hosters can alter their offers according to the development and growth of your shop. You should consider the following topics:

- Shared hosting vs. vServer vs. dedicated server
- Software stack resembles the above mentioned recommendations?
- Any prior Shopware experience at hand?

- Are there any Shopware based reference shops available?

Generally we recommend accepting services from one of our certified web hosters. A current list of hosters can be found following this link: [Shopware hosting partner](#).

How to measure performance?

When measuring a shop's performance, loading times can only be an indicator for an expected level of performance. Characteristics present in your shop, ranging from item count to internal structure or momentary capacity utilisation will always have a significant influence on palpable loading times.

You should perform the following analysis steps in a test system - you can create this in a subfolder using these instructions: <https://docs.shopware.com/en/shopware-5-en/tutorials-and-faq/shop-transfer-live-system-test-environment>

- Deactivate all third-party extensions without exception
- Assign Standard Responsive Theme
- Warm up cache
- Check Performance again

If these steps have led to an improvement, the factors must be excluded step-by-step. If there is still unsatisfactory performance, it is advisable to determine the following benchmark values:

- How many articles does the shop have? How many variants?
- How many categories does the shop have? How many category assignments?
- How many simultaneous accesses are there?
- Where is the poor performance noticeable?
- Can a time be defined here at which the problems occur?

The cronjob "Cleanup" is elementary for a cleanup of the system - the following database tables are cleaned up here:

- *_s_emarketing_lastarticles* - article history older than 30 days
- *_s_statistics_search* - Similar search terms older than 30 days
- *_s_core_log* - backend log entries older than 30 days
- *_s_order_notes* - watch lists which are older than one year and have no user assignment
- *_s_statistics_referrer* - Referrers older than 90 days (default)
- *_s_statistics_article_impression* - article views older than 90 days (default)

Please also check the number of records in the following database tables:

- *_s articles_categories_ro*
- *_s articles_details*
- *_s articles_prices*
- *_s articles_also_shown_ro*

Measuring performance with Firebug

To generate comparable measurements, we recommend to install the Shopware demo data plugin before running tests in your shop. Firebug is a browser extension which can be installed directly via your browser. If installed it will display load times for every page component in your browser's networking tab.

Particularly relevant are the entire webpage's load times - most likely the first entry of the list. To gather relevant measurements you should load the landing page, listing pages and details pages several times and note all loading times. By calculating the means you will receive meaningful statistics on your shop's performance.

Measuring performance with Apache benchmark

Call the benchmark using a linux shell:

```
ab -n 10 -c 1 http://HOMEPAGE_URL / CATEGORY / DETAILPAGE
```

Here you should go for the median of all connection times (= total time).

Analysing performance

task	contact person	info
Identify problematic component	customer / hoster	Try and localize the source of the performance drops. Possibly this could be your webserver, the database server, your PHP stack and so on.
Analysing slow SQL queries	hoster	What database queries affect the overall performance of your shop?
Analysing the hosting platform	hoster / customer	Do you have a hosting environment suitable for the size and the aggregated traffic of your particular shop? Do you use current and recommended server components? Do you employ dedicated servers instead of vhostings or webspace packages? Do you run any other applications apart from Shopware on your server?

Accessibility	customer	Is there an ERP system clogging up the connection with large amounts of data? Are there regular tasks running on your shop server, potentially lowering the performance.
---------------	----------	--

Optimizing the MySQL configuration

The standard configuration of many distributions is not ideally suited for productive environments. But don't worry, there is a script for that! "Mysqltuner" provides hints towards suboptimal configuration and parameter settings.

```
wget http://mysqltuner.com/mysqltuner.pl
chmod +x mysqltuner.pl
./mysqltuner.pl
```

Many properties and performance drops

Set the attribute `optimizer_search_depth = 0` in your MySQL configuration. [Read more.](#)

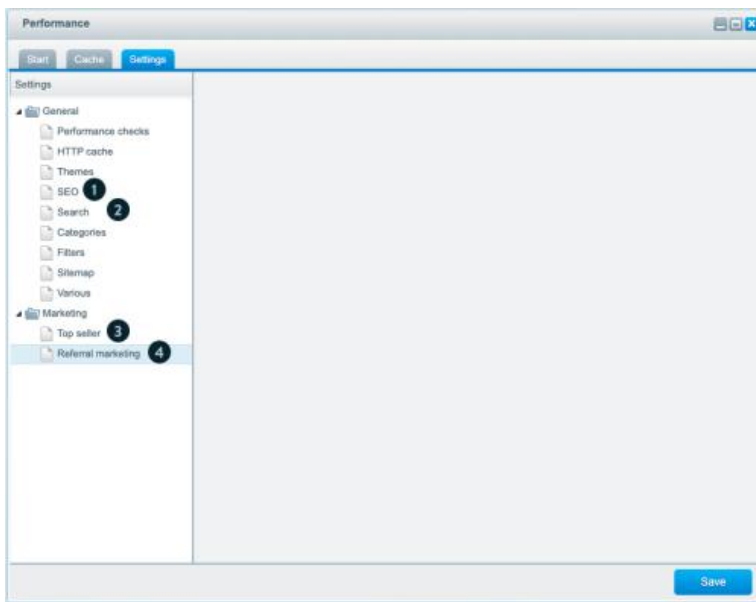
Tips & Tricks

Standard sorting for category lists

In case you have defined your own sorting criteria here, you should make sure there is an index in your database for each of the table's columns.

Switching your update strategy

There are several different processes active when one of your shop's pages is accessed. For example SEO links are generated, the search engine's cache is generated as well as the necessary data for displaying top selling items or other items of interest based on similar customer behaviour. Thoroughly check if and how using this function may benefit your enterprise and consider switching the update strategy from "live" (per page request) to "cronjob" (cumulative, time-controlled). To do so navigate to **Configuration > Cache/performance > Cache/performance > Settings (tab)**.



Look for the entries **SEO (1)**, **Search (2)**, **Top seller (3)** and **Referral Marketing (4)** to make your changes there. To make use of cronjob functionalities, you first will have to install two custom plugins, "Cron" and "CronRefresh". With these plugins you can time control server requests and should schedule them, most likely towards night time. If you are not sure how that works, kindly refer to this wiki article: [Cronjobs](#).

Activating HTTP cache

With Shopware you can make use of caching instead of live fetching all the data for each user from the database. As each database access uses up scarce web resources, consider activating http-caches. This will create a cache file, that incorporates the most important information from the article's listing and its details page, therefore preventing unnecessary database queries. This will also make browsing your shop a bliss as it will improve the performance greatly. For keeping prices up-to-date, this cache is invalidated when prices change!

With the cronjob we created above, we made sure the cache file doesn't grow too big, as it is deleted regularly (standard value: every 24h). This has proven to be quite practical as the http-cache might grow by 1-2GB each day if frequently used. Activate the http cache via **Configuration > Cache/performance > Cache/performance > Settings (tab) > Http cache** and check the plugin and all changes made for compatibility with your shop. It may be wise to make a test run before going live with the changes you made.

Inspecting plugins

There are not only a few cases in which the use of plugins severely impacted a shop's performance. To see if that might be the case in your shop, too, it might be wise to deactivate all third-party plugins and compare the frontend performances with and without activated plugins.

An easy way of doing this is to conduct an SQL query in a test environment. Both queries shown hereunder can be run in phpMyAdmin and will deactivate a plugins that are not part of the default Shopware configuration.

```
UPDATE `s_core_plugins` SET `active`= 0 WHERE `source` <> 'Default' AND author <> 'shopware AG'
```

Step-by-step instructions are to be found in the following wiki article: [Shopware SQL](#).

Alternatively you may want to use the [Plugin safe mode](#) provided via the Plugin manager from Shopware 5.3 onwards.

Once the shop runs smoothly again, you can try and make out the plugin impacting your shop's performance by reactivating the third party plugins step-by-step. You can also re-evaluate the need for each individual plugin and tidy up your shops backend a little.

"Debug" plugin

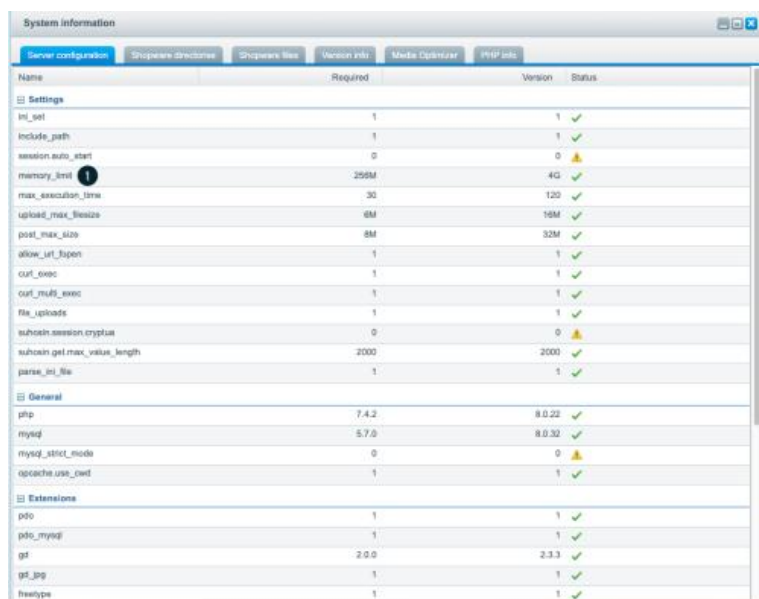
Shopware out-of-the-box comes with a "Debug" plugin, that is able to monitor and benchmark individual processes on a server. It will provide you with sufficient information on e.g. the duration of page requests and it will hint at processes e.g. database queries, controller etc. that might slow down individual requests.

Optimizing servers

Have you ever heard about extended caching, namely ZendOP and APCu? Either way, you should consider installing and activating both in your shop to boost performance even more. To evaluate the current performance levels of a shop, use the tools found in **Configuration > Caches/performance > Caches/performance > Settings > Performance checks**

System check

Shopware gathers information on your system configuration and lists them in **Configuration > System info**.



Name	Required	Version	Status
Settings			
int_set	1	1	✓
include_path	1	1	✓
session_auto_start	0	0	⚠
memory_limit	256M	4G	✓
max_execution_time	30	120	✓
upload_max_filesize	8M	16M	✓
post_max_size	8M	32M	✓
allow_url_fopen	1	1	✓
curl_exec	1	1	✓
curl_multi_exec	1	1	✓
file_uploads	1	1	✓
suhosin.session.cryptua	0	0	⚠
suhosin.get.max_value_length	2000	2000	✓
parse_ini_file	1	1	✓
General			
php	7.4.2	8.0.22	✓
mysql	5.7.0	8.0.32	✓
mysqlstrictmode	0	0	⚠
opcache.use_cwd	1	1	✓
Extensions			
pdo	1	1	✓
pdo_mysql	1	1	✓
gd	2.0.0	2.3.3	✓
gd_bmp	1	1	✓
imagick	1	1	✓

Here you find entries like "memory_limit" **(1)**. Exceeding the memory limit e.g. will negatively impact performance. Be aware that memory limits can be tweaked and e.g. virtually enlarged, making a thorough examination of your system worthwhile.

Compiler Caching

Visiting **Configuration > Theme manager > Settings** you have the opportunity to deactivate compiler caching. This is generally advised during the normal productive shop cycle. Because compiler caching greatly slows down a shop, it should only be used when you are developing and tweaking on the infrastructure anyway.

Plugin Licenses - Questions and Answers

Licences FAQ

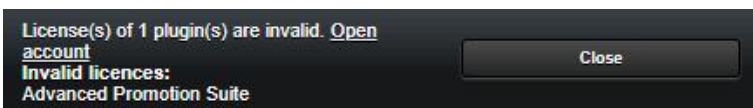
What does the "Plugin Warning" window mean when I open my backend?

If you are using a plugin for which you do not have a valid licence, a corresponding message will appear when you open the backend. At least one licence of the plugins you are using has expired.



How do I find out which plugins are not licensed?

In addition to the warning box (see above), another, smaller warning box is displayed in the Shopware backend. This shows you the name of the affected plugin. Shopware makes no difference here whether the plugin has the status "Activated", "Deactivated" or "Uninstalled" in the plugin manager. **Unlicensed plugins must be completely deleted and may no longer appear in the plugin manager of your shop!** In this example, the unlicensed plugin "Advanced Promotion Suite" still appears in the plugin manager. It must be completely deleted or the required licence must be purchased via the Shopware Store.

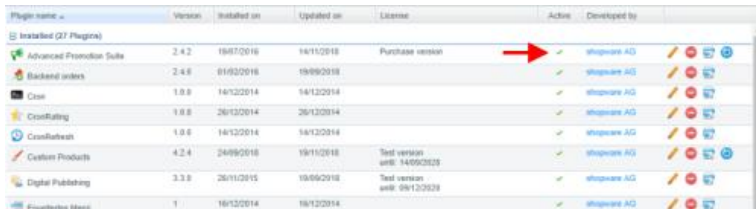


Note: If this message does not appear in your backend, delete all Shopware cookies in your browser and reload the backend.

How do I delete the affected plugin?

To remedy the licence violation, the plugin must be completely deleted from the plugin manager. It is not sufficient that it is only deactivated, as the plugin can then be reactivated at any time.

Step 1: In the backend of your shop, open the plugin manager and select the "Installed" tab. Then deactivate the plugin in the "Active" area (if still active) by clicking on the green tick:



Plugin name	Version	Installed on	Updated on	License	Active	Developed by
Installed (27 Plugins)						
Advanced Promotion Suite	2.4.2	19/07/2016	14/11/2016	Purchase version	✔	shopware AG
Backend orders	2.4.6	01/02/2016	19/09/2016		✔	shopware AG
Case	1.0.0	14/12/2014	14/12/2014		✔	shopware AG
CrossRating	1.0.0	26/12/2014	26/12/2014		✔	shopware AG
CrossRefcard	1.0.0	14/12/2014	14/12/2014		✔	shopware AG
Custom Products	4.2.4	24/09/2016	19/11/2016	Test version url: 14/09/2016	✔	shopware AG
Digital Publishing	3.3.0	26/11/2015	19/09/2016	Test version url: 09/12/2015	✔	shopware AG
Flowerbox Basic	1	16/12/2014	16/12/2014		✔	shopware AG

Step 2: The plugin now appears under "Deactivated". Click on the button "Install/Uninstall".



Plugin name	Version	Installed on	Updated on	License	Active	Developed by
Inactive (3 Plugins)						
Advanced Promotion Suite	2.4.2	19/07/2016	14/11/2016	Purchase version	✘	shopware AG
Debug	1.0.0	22/09/2016	22/09/2016		✘	shopware AG
...	3.8.1	21/12/2016	14/11/2016		✘	shopware AG

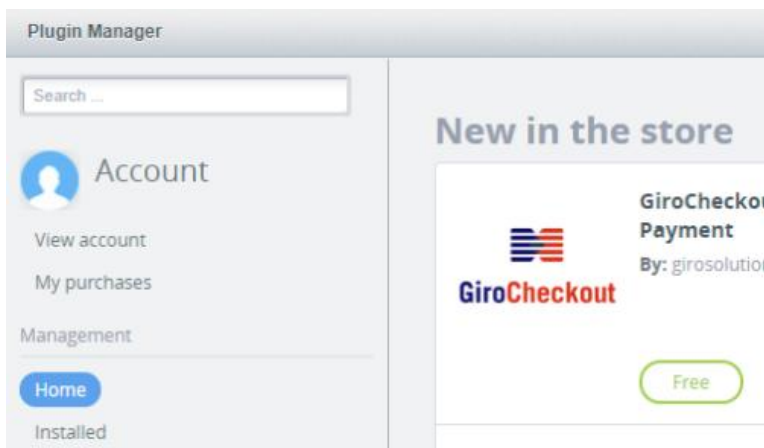
Step 3: The plugin is now listed under "Uninstalled". Click on the "Delete" button there. This removes the plugin permanently from the plugin manager.



Plugin name	Version	Installed on	Updated on	License	Active	Developed by
Uninstalled (7 Plugins)						
Advanced Promotion Suite	2.4.2		14/11/2016	Purchase version	✘	shopware AG
CrossRating	1				✘	shopware AG
...	1.0.0				✘	shopware AG

Note: If the plugin cannot be deleted, in most cases it can be deleted directly from the database. If there are complications, contact the plugin manufacturer, who can offer you support.

Step 4: Log in to the plugin manager under "My purchases" with your account access data so that the deletion of the plugin is displayed to us.



Attention: If there is an installation in the subfolder or a local installation under the domain, the plugin must also be completely deleted there.

Why does the message "Trial licence expired" still appear in the backend, even though I have a valid rental licence?

In this case, it will be the case that the test licence for the affected plugin is still stored in the licence manager in the Shopware backend in addition to the rental licence.

Simply delete the test licence in the licence manager in the backend. After that, the message "Test licence expired" should no longer appear for the affected plugin.

Subscriptions FAQ

The information in this article relates to existing subscriptions. New subscriptions for Shopware 5 plugins can no longer be taken out. Further information on the end of life of Shopware 5 can be found [here](#).

What is a plugin subscription?

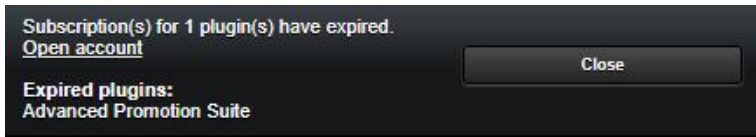
With a valid plugin subscription, you ensure that you are provided with plugin updates by the plugin manufacturer. Updates can include new plugin functions, optimizations, compatibility with new Shopware versions or the elimination of security gaps.

Note: Not all plugins have a subscription check. Free plugins, such as the Shopware PayPal plugin, can be updated without having to book a subscription.

Why is my plugin subscription no longer valid?

The subscriptions are limited in time and must be renewed by you via your Shopware account. The purchase of a plugin normally includes a subscription valid for 1 year. This means: You will be provided with

plugin updates by the plugin manufacturer for 1 year. If you do not renew the plugin subscription, you will not be able to update the plugin and a corresponding message will appear when you open your backend:



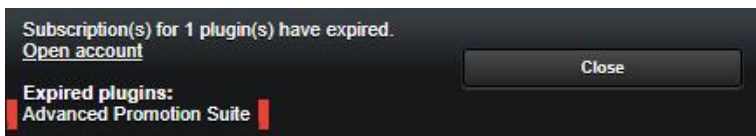
Note: The subscription check checks all plugins listed in the plugin manager. The check does not distinguish whether the plugin is "Active", "Deactivated" or "Uninstalled". All plugins listed in the plugin manager are checked.

What are the consequences of using a plugin without a valid subscription?

A missing plugin subscription has no effect on the functionality of Shopware or the plugin concerned. However, you should bear in mind that in this case you can no longer benefit from improvements and compatibility on the part of the plugin manufacturer.

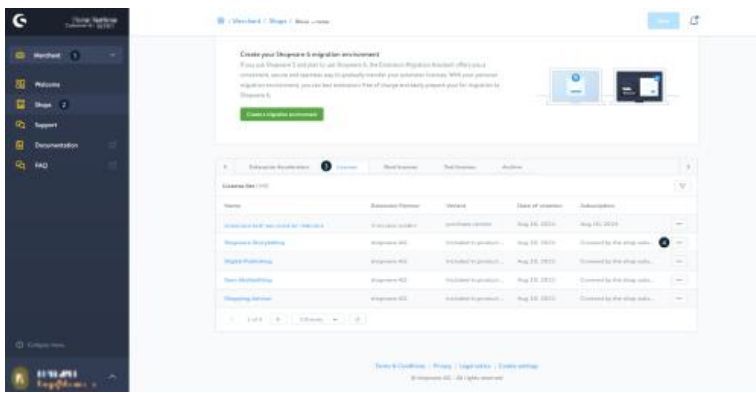
How do I find out which of my plugins listed in the Plugin Manager in the backend no longer has a subscription?

The message in the backend shows you which plugin does not have a valid subscription. In the example, the two plugins "Advanced Promotion Suite" and "Payment Icons/Payment Types for Header, Footer & Checkout" no longer have a valid subscription and can therefore no longer be updated.

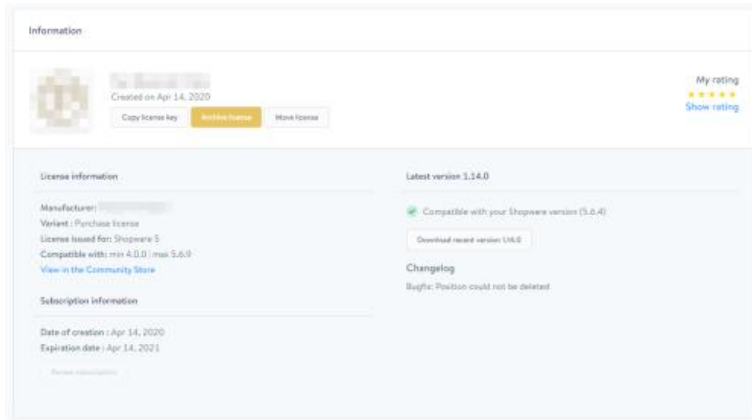


How can I extend a plugin subscription?

In your [Shopware account](#) you will find the menu item **Merchant (1)** on the left-hand side. After you have selected this, you can click on the sub-option **Shops (2)**. All domains that are linked to your Shopware account will be displayed there. Then click on the desired domain so that the **licences (3)** can be displayed. Within the list you can see which plugins are licensed for your domain and for which plugins a subscription is active. If you click on the **... button (4)**, a context menu opens in which you can then click on **Details**.



After clicking on **Details**, an overview of the selected plugin is displayed. Here you have the option to extend the plugin subscription, but only 4 weeks before the subscription expires.

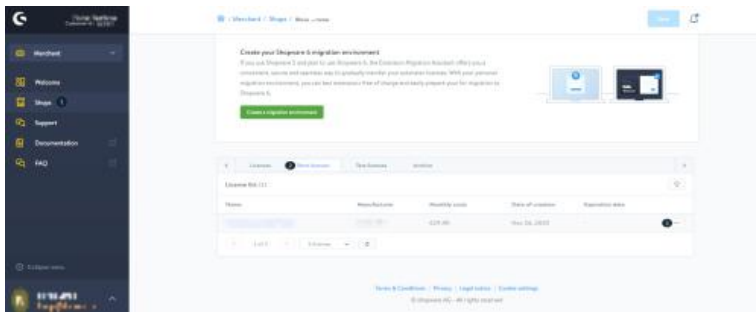


Note: Without a valid plugin subscription, no plugin updates are available to you from the manufacturer.

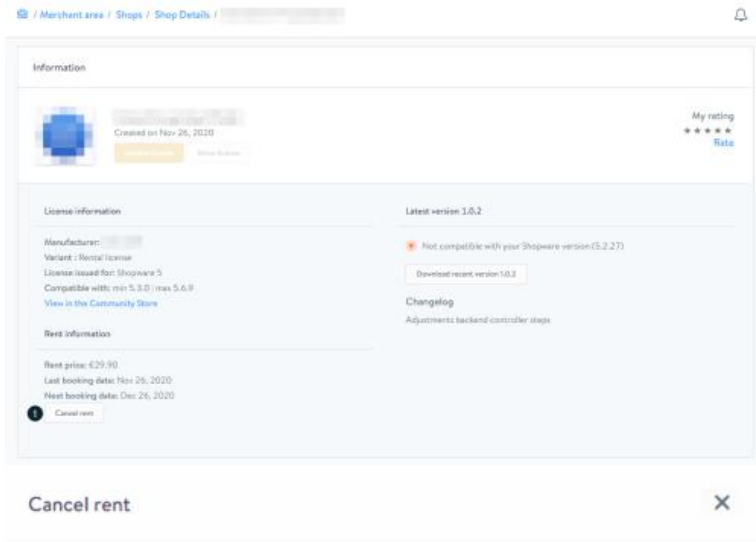
We recommend that you keep the plugins up to date. Otherwise you will miss out on improvements, bug fixes and possible security updates from the plugin manufacturer.

How can I unsubscribe from a rental licence?

If you have rented one or more plugins and would like to cancel this rental, you must log into your Shopware account at <https://account.shopware.com/> to do so. There you will find your current shops in the menu under **Merchant > Shops (1)**. As soon as you have clicked on the desired shop, you will receive an overview of your licences. You will also find the **rental licences (2)**. To cancel your rental licence, you must click on the **... button (3)** to open the context menu. In this context menu, click on **Details**.



Clicking on **Details** takes you to the overview page of the selected plugin. You can now cancel the rental licence by clicking on the **Cancel rental button (1)**.



A cancellation of the rent of the plugin "Interaktiver Produktberater" is to the 26.12.2020 possible. As of the date of termination, the use of the plugin is prohibited immediately and must be completely removed from the shop following the instructions outlined in our [documentation](#). This will be checked in the upcoming update and maintenance processes.

I confirm that the plugin will not be used in the shop after the cancellation date.

After you have ticked the box, you will be asked for the reason for the termination of the rental and then the rental can be terminated. The status of the termination is then displayed directly on the overview page of the plugin.

Contact for queries about the account, licences or subscriptions

If you have any questions, please contact us at store@shopware.com or call +49 (0) 2555 928850.

Open source in Shopware 5.5

Shopware is open source since 5.2.0, but plugins were encrypted even longer. With Shopware 5.5 this era ends now. Here we explain how you can run your shop completely open source.

Check plugins

If plugins want to be compatible with Shopware 5.5, they must no longer be encrypted.

Check all plugins purchased in the store for compatibility with Shopware 5.5, you should make sure that all plugins are compatible before updating Shopware!

If you use individual third-party plug-ins that are not available in the store, ask the manufacturer if the plug-ins are encrypted and request an unencrypted version that is compatible with 5.5.

How do I check the plugins at all?

You can easily check plugins for encryption by opening the bootstrap.php of the respective plugin.

An open source bootstrap looks like this:

```
1 <?php
2
3 class Shopware_Plugins_Frontend_SwagDemoDataDE_Bootstrap extends
4     Shopware_Components_Plugin_Bootstrap
5 {
6     public function getCapabilities()
7     {
8         return array(
9             'install' => true,
10            'update' => true,
11            'enable' => true,
12            'secureUninstall' => false
13        );
14    }
15    public function install()
16    {
17        if (!$this->assertMinimumVersion('5.4.0')) {
18            return false;
19        }
20    }
21}
```

An encrypted one, on the other hand, looks like this:

```
1 <?php //00535
2 // Copyright shopware AG (www.shopware.com) [l0ldad2]
3 //
4 if(!extension_loaded('ionCube
* Loader')){$_oc=strtolower(substr(PHP_UNAME(),0,3));$_ln='ioncube_lo
* ader_'.$_oc.'_'.substr(PHP_VERSION(),0,3).(($_oc=='win')?''.dll':'.so
* ');if(function_exists('dl')){@dl($_ln);if(function_exists('_il_exec
* ')){return
* _il_exec();}$_ln='/ioncube/'.$_ln;$_oid=$_id=realpath(ini_get('ex
* tension_dir'));$_here=dirname(__FILE__);if(strlen($_id)>1&&$_id[1]
* ==:'){$_id=str_replace('\\','/',substr($_id,2));$_here=str_replac
* e('\\','/',substr($_here,2));}$_rd=str_repeat('../',substr_count($_
* _id,'/'));$_here.='';$_i=strlen($_rd);while($_i--){if($_rd[$_i]
* ==/'){$_lp=substr($_rd,0,$_i).$_ln;if(file_exists($_oid.$_lp))
* {$_ln=$_lp;break;}}if(function_exists('dl')){@dl($_ln);}}else{die
* ('The file '.__FILE__.' is
* corrupted.\n");if(function_exists('_il_exec')){return
* _il_exec();}echo("Site error: the
* ".(PHP_SAPI_NAME()=='cli'? 'ionCube': '<a
* href="http://www.ioncube.com">ionCube</a>')." PHP Loader needs to be
* installed. This is a widely used PHP extension for running ionCube
```

Update shopware

The second step is to update Shopware. If your Shopware installation runs in version 5.5.0 or higher, you can download plugins from the store only unencrypted.

Update plugins

After the update, you should upgrade all store plug-ins to a 5.5-compatible version so that all store plug-ins are unencrypted on your system.

Encrypted plugins not distributed via the store must be replaced by unencrypted versions at this point at the latest!

Remove the license manager

After all plugins are open source, you can delete the license manager and remove it from the system.

The license manager will also run in Shopware 5.5 for compatibility reasons, but it will no longer be developed, so that the use of modern PHP versions from 7.1 with installed license manager is also not possible, we therefore strongly recommend to remove the license manager at the end!

Shop transfer live system / test environment

In this article we'll show you how to transfer your test environment to your productive environment and vice versa. Please note, that your server may have different versions of e.g. phpMyAdmin or FTP client, so they may look different.

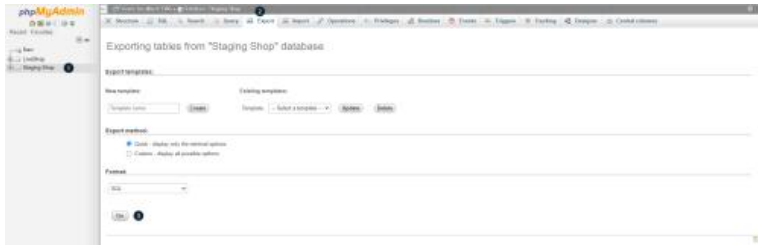
From Test System to Live System

This shows you how to transfer your test shop into your live system, e.g. after developing.

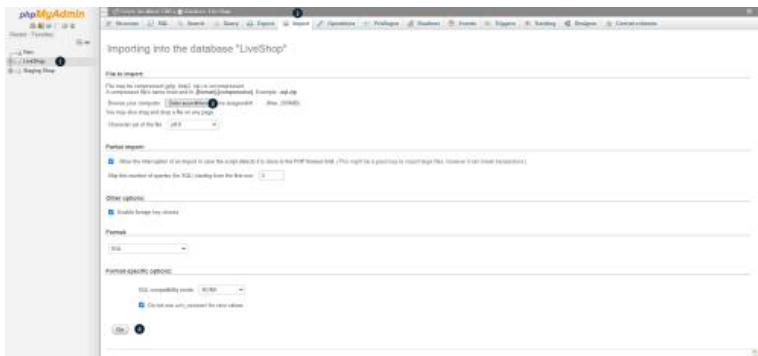
Copy the files

Open your preferred FTP client and copy the shopware root directory in the main routed directory of your live environment. Keep in mind, that you **exclude cache directories** from the upload, the cache folder can be found in `/var/cache`.

Copy the database



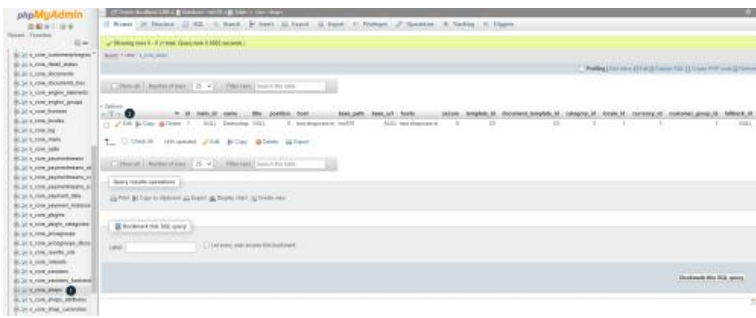
Open phpMyAdmin in your test system and log in. Choose your test database **(1)** and click export **(2)** to export **(3)** your test database.



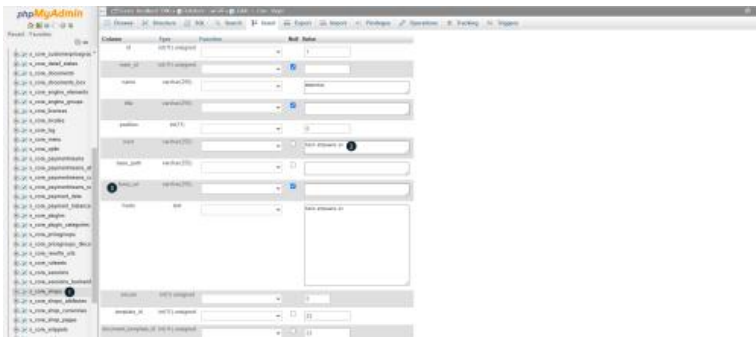
Now open the phpMyAdmin of your live system and choose the desired database **(1)**, click **import (2)** and choose your **exported sql file (3)** and start the import **(4)**.

It may happen, that your exported file is too big to get it imported at one, when this happens, you have to export your database dump using compression. If the file is still too big for the import, you have to use batch processing tools like BigDump for example to get your sql file imported.

Changing the shop URL



To make your shop reachable, you have to change the shop URL in the database. To do this, just open the table **s_core_shops (1)** and **edit (2)** the entry of your main shop.



In the column **host (2)** you have to change your URL from the test URL to the live URL. **Note** that this URLs must be entered **without http(s)://!** According whether you use your shop in a subfolder, enter this folder in the column **base-path (3)** or clear this value, if you don't use shopware in a subfolder. After doing this, save your changes.

Changing the config.php

Finally you have to connect your file system to your database, this can be done by editing the config.php in shopware's main directory. Just open the file and change the following data to those you got from your hoster (Normally they are the same like your phpMyAdmin credentials):

```
'username' => 'root',
'password' => 'root',
'host' => 'localhost',
'port' => '3306',
'dbname' => 'shopware',
```

If you copied the cache files, you now should delete your *production_REVISION* directories in */var/cache* or just execute the *clear_cache.sh* script in the cache folder. If you reopen your shop, the new cache files will be created automatically.

Permissions of the shopware directories

Now you should open the backend of your shop and open *Configuration > System info > Shopware directories*. If directories are marked with a red X, just reset the folder permissions to 0755. Except of cache directories, **NEVER** set permissions to 0777 because this gives everyone read and write access to your data, which should be avoid. After this step, clear all caches except SEO and search cache and reload the Shopware backend.

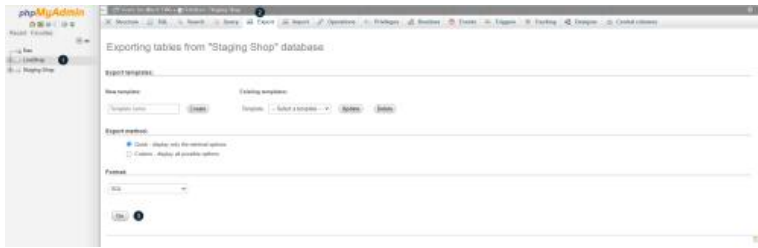
Copy a live system into a test system

If you setup yourself a testsystem for programming, we describe how this works.

Copy the files

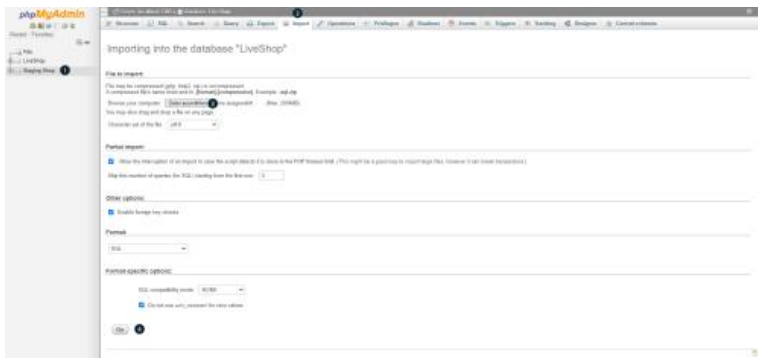
Open your preferred FTP client and transfer the files (excluded the cache folders /var/cache/production_REVISION) from your live system into your test system. Depending on which test system you use, the folder may be different, in Linux systems this is normally in /var/www. Copy all the files and overwrite all files.

Copy the database



Open phpMyAdmin in your live system and log in. Choose your database **(1)**, click on export **(2)** and export **(3)** your database.

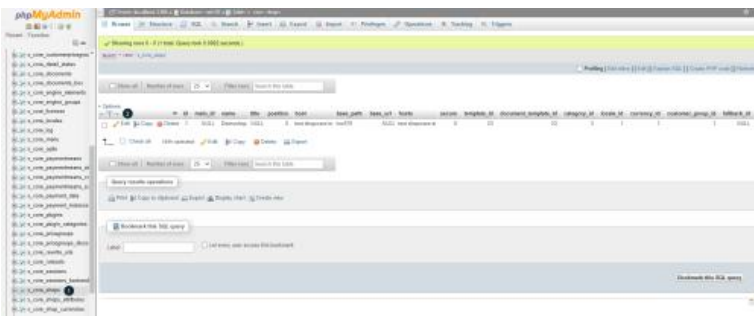
Now open your local phpMyAdmin and create a new database if needed. Otherwise clear your existing database you want to use.



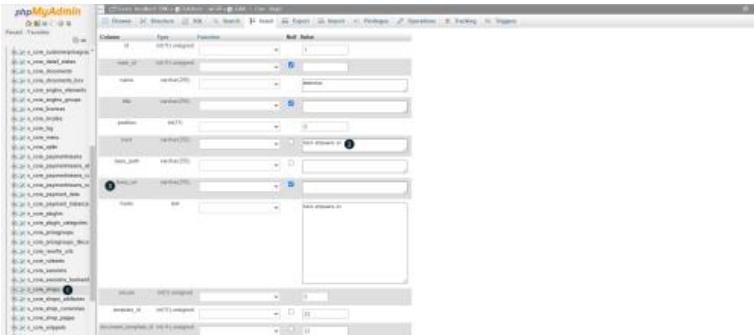
Now open your database **(1)** click import **(2)** and select the **file for the import (3)**. After that you can click on **Go (4)** to import the live database into the test system.

It may happen, that your exported file is too big to get it imported at one, when this happens, you have to export your database dump using compression. If the file is still too big for the import, you have to use batch processing tools like BigDump for example to get your sql file imported.

Changing the shop URL



To make your shop reachable, change the URL in your test database, just open phpMyAdmin and open the table `s_core_shops` **(1)** in your test database and **edit (2)** the host to (in the most cases) localhost.



Now change the entries **host (2)** and **base_path (3)**. In our example, the test environment should be directly accessible under "http://localhost". Accordingly, the entry host must be changed to localhost. Since we do not operate our shop in a subdirectory, the entry **base_path (3)** must be empty. Then save the changes you have made. If you are using your testshop in a subfolder, you also have to change the **base_path (3)**, otherwise, base_path has to be empty.

Changing the config.php

Finally you have to connect your file system to your database, this can be done by editing the config.php in shopware's main directory. Just open the file and change the following data to those you got from your hoster (Normally they are the same like your phpMyAdmin credentials):

```
'username' => 'root',
'password' => 'root',
'host' => 'localhost',
'port' => '3306',
'dbname' => 'shopware',
```

If you copied the cache files, you now should delete your `production_REVISION` directories in `/var/cache` or just execute the `clear_cache.sh` script in the cache folder. If you reopen your shop, the new cache files will be created automatically.

Tips & Tricks

Prevent indexing test systems in the www

If you have your test system on a public webserver, we recommend to avoid indexing by search engines.

Activating service mode and restricting by IP

Activate the Service Mode in the shop in Configuration > Basic settings > Settings > Service and restrict the access to your IP, so you can work in your test system. When you don't know your IP, you can check it on e.g. <https://www.whatismyip.com/>.

Prevent the shop of crawling by robots

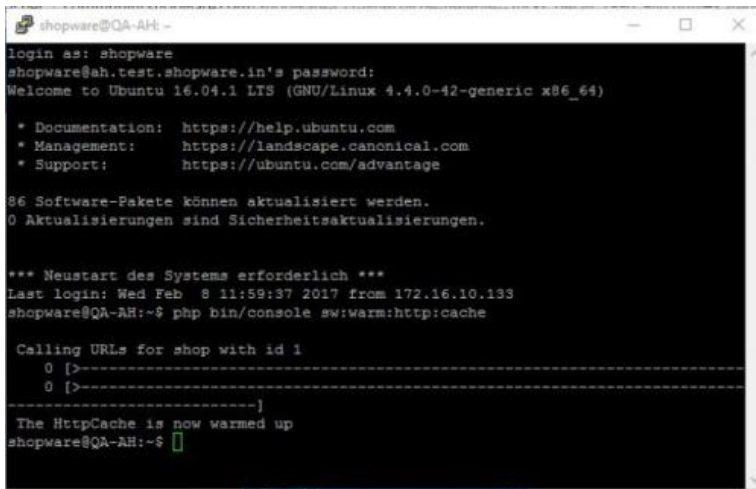
Create a file named robots.txt in the shopware root directory with the following content:

```
User-agent: *  
Disallow: /
```

This tells every robot to not crawl your shop. If you transfer your shop from Test to live, don't forget to delete this file!

Shopware CLI

Shopware offers some useful tools that can be executed via the "console". The so-called "cli" or "shell" commands can be executed without script limitations (for example, the script run time of the server like the php execution time) and are therefore recommended for processing large amounts of data. For the commands to be executed, a terminal software such as "PuTTY" is required, which conveniently is available for free on the internet. Utilizing a terminal program you can use an ssh-connection to access the web server, where commands can be executed.



```
shopware@QA-AH -  
login as: shopware  
shopware@ah.test.shopware.in's password:  
Welcome to Ubuntu 16.04.1 LTS (GNU/Linux 4.4.0-42-generic x86_64)  
  
 * Documentation:  https://help.ubuntu.com  
 * Management:   https://landscape.canonical.com  
 * Support:      https://ubuntu.com/advantage  
  
86 Software-Pakete können aktualisiert werden.  
0 Aktualisierungen sind Sicherheitsaktualisierungen.  
  
*** Neustart des Systems erforderlich ***  
Last login: Wed Feb 8 11:59:37 2017 from 172.16.10.133  
shopware@QA-AH:~$ php bin/console sw:warm:http:cache  
  
Calling URLs for shop with id 1  
0 [>-----]  
0 [>-----]  
]  
The HttpCache is now warmed up  
shopware@QA-AH:~$
```

The commands can be executed according to the following scheme:

bin/console <command> **OR** php bin/console <command>

Many commands also contain a help function, in which possible parameters are listed.

```
bin/console sw:plugin:list -h OR bin/console sw:plugin:list --help
```

Here are some important commands that make your everyday life with "shopware" much easier.

Generate thumbnails

If the standard image sizes are changed or new images are added to your database, it is necessary to regenerate the thumbnails. The problem here is that due to script engine limitations, using the provided backend functionalities is only possible if you are processing a very small stack size. Therefore it's recommended to generate the thumbnails using console commands, which still may use a bit of time, depending on the number of images to be processed and the resources available on the server. Generally thumbnails should not be regenerated periodically, because the process requires a lot of performance and is only necessary if new thumbnail sizes are generated or if any images were replaced.

```
php bin/console sw:thumbnail:generate
```

HTTP-Cache warmup

This command is particularly useful as a cronjob, as it considerably reduces the shop's loading times after emptying the cache. This process is not covered by the regular shopware cronjob, so it has to be stored as a separate cronjob in the server console.

```
php bin/console sw:warm:http:cache
```

With the addition "**--help**" you can display all available options, including the option to empty the existing cache beforehand or to divide the calls into several processes.

For more information and useful commands regarding cronjobs, please see the article [Shopware cronjobs](#). Also, please note that this process may take some time, depending on the number of processed articles or categories. The process is executed automatically for each subshop.

Rebuild category tree

In some cases, it might be necessary to rebuild the category tree. However, if you have large amounts of categories or articles hosted in your shop, you should avoid rebuilding the tree via the backend. We strongly recommend to use the console for this.

```
php bin/console sw:rebuild:category:tree
```

Please note that this process may take some time, depending on the number of articles or categories. This process is executed automatically for each subshop.

Overview of commands

A list of all possible commands is printed when entering the following command:

```
php bin/console
```

Core commands

sw:thumbnail:cleanup	Deletes thumbnails that lack their original (full size) image file.
sw:thumbnail:generate	Generates a new thumbnail.
sw:generate:attributes	Generates attribute models.
sw:warm:http:cache	Warms up http cache
sw:media:cleanup	Collects unused media and moves it to the trashbin.
sw:snippets:find:missing	Finds missing snippets in the database and dumps them into .ini files
sw:snippets:remove	Removes snippets from the database. Applicable for a specified folder
sw:snippets:to:db	Loads snippets from the "/snippets" folder in your installation's main directory to the database. To find the folder, click here
sw:snippets:to:ini	Dump snippets from the database into .ini files
sw:snippets:to:sql	Load snippets from .ini files into sql files

Plugin commands

sw:plugin:activate	Activates a plugin.
sw:plugin:config:list	Lists plugin configurations.
sw:plugin:config:set	Sets plugin configurations.
sw:plugin:deactivate	Deactivates a plugin.

sw:plugin:delete	Deletes a plugin.
sw:plugin:install	Installs a plugin.
sw:plugin:list	Lists plugins.
sw:plugin:refresh	Refreshes plugin list.
sw:plugin:uninstall	Uninstalls a plugin.
sw:plugin:update	Updates a plugin.

Create own commands with a plugin

If you want to build your own commands, take a look at this: [DevDocs](#).

Debugging Of Shopware Updates

Table s_schema_version

Because the deltas were imported by a php script, the execution and occurring errors are logged in the table **s_schema_version**. Errors are written into the column **error_msg**. You can find the deltas in /update-assets/migrations. If they were imported, you can see in the above mentioned database table. Check if every migration got a database entry and see that *start_date* and *end_date* are correct timestamps (YYYY-MM-DD HH:MM:SS) and the field *error_msg* has *Null* set.

If problems occur at importing the deltas, we recommend to recovery the whole database backup to correct the error. Alternatively you can import the delta manually (execute), in this case you need to set the *s_schema_version* entry manually too (see above). If the delta is already set in the database, you can set *Null* in "error_msg" and a correct timestamp at "end_date", if the updater runs another time, this delta is ignored.

FAQ

Update not possible: Software Subscription

The Auto Updater may display the following message:

"No active Shopware subscription could be found for your shop. The automatic update cannot be performed."

If you have an active subscription and this message still appears, a license is stored in the database that is not up-to-date. Then check the following steps. You may not be able to check all steps because the license manager is no longer available or the "Shopware License" area is loading:

- At **Configuration > Basic Settings > Additional Settings > License Manager** only currently valid license entries that have the domain as host that is currently registered in the Shopware account may be available.
- At **Configuration > Basic settings > Other settings > Shopware license** please enter the current license from your Shopware account.
- In the **s_core_licenses** table, the entry with the Shopware Core label must first be deleted. The Shopware license should then be entered via the backend at **Basic Settings > Additional Settings > Shopware License**. If this does not solve the behavior, the table must also be checked for expired plugin licenses. These entries also have to be deleted so that an update can be carried out.

Foreign Key Errors

Frontend:

```
No mapping found for field '***' on class 'Shopware\Models\Attribute\***'. in vendor/doctrine
```

or

```
Fatal error: Cannot redeclare Shopware\Models\Attribute\CustomerGroup::$customerGroup in /var/
```

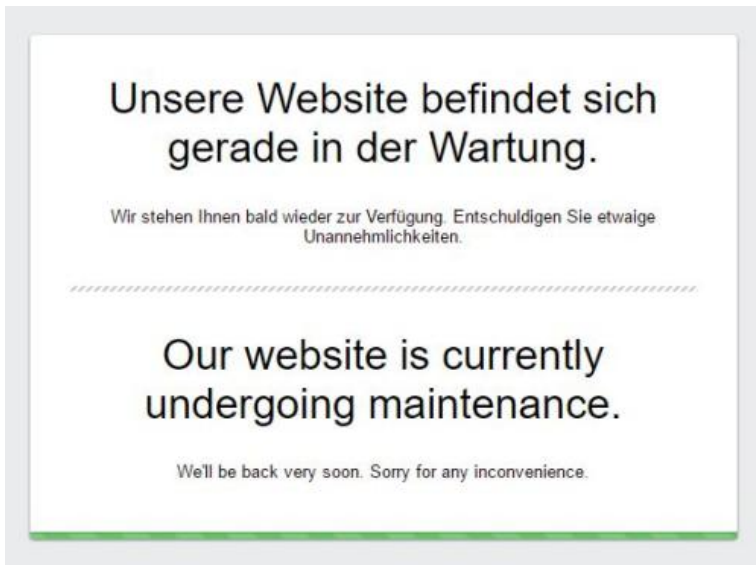
Backend:

```
Fatal error: Cannot redeclare Shopware\Models\Attribute\User::$user in /var/www/production/sho
```

A possible reason may be broken database table relations, to correct them you can [repair the foreign keys](#). After this, clear the cache manually by deleting the folders `/var/cache` and `web/cache`!

Maintenance mode won't disappear

If your frontend or backend still shows the maintenance mode after the update, you have to delete a folder in the Shopware directory. Below in the description we explain what these are.



The maintenance mode should automatically quit after updating Shopware. If this won't happen, it may be caused by insufficient permissions of the php process which executes the update script and tries to delete the update folder to quit the maintenance mode. Grant the user, which executes the update script, more permissions to delete this folder to avoid this problem in future updates. To manually solve this problem, delete the following folder:

At the auto update: /files/update

At the manual update: /update-assets

Last seen don't work correctly

If the "last seen" item history won't work (in such cases the footer of the page may be broken). The cause may be located in the deduction of your theme. That way, changes from the default theme can't be processed.

Please check the following file: /templates/emotion_meintemplate/frontend/index/index.tpl

Maybe it contains the following code:

```
{block name="frontend_index_footer"}
  {if $sLastArticlesShow}
    {action module=widgets controller=lastArticles}
  {/if}
{/block}
```

This has to be changed to:


```
{block name="frontend_index_footer"}
  {if $$LastArticlesShow && !$isEmotionLandingPage}
    {include file="frontend/plugins/index/viewlast.tpl"}
  {/if}
{/block}
```

We recommend to check additionally, if the file `/templates/_default/frontend/plugins/index/viewlast.tpl` is the default one. After that, clear the cache in the backend.

initTopSeller?offset=0&limit=15

When you get the error message *"Internal Server Error: initTopSeller?offset=0&limit=15"* the topseller box was deactivated (`chartInterval = 0`), this configuration may cause this problem. Just activate the topseller box.

The target-entity Shopware\Models\Attribute

The error message *"The target-entity Shopware\Models\Attribute\... cannot be found in ..."* appears, when the attribute models could not be generated after your update. If possible, clear the whole cache in the backend or use the console command `bin/console sw:generate:attributes`

importDatabase?XXX - Forbidden

When you get the message *"importDatabase?offset=XXX&totalCount=XXX - Forbidden"* check, if `mod_evasive` block the requests and deactivate this module if needed.

Choose a smaller batch size

If you get the message *"Choose a smaller batch size"* after an auto update check, if the directory `/files` contains defective update files e.g. 0 byte files and delete them.

Could not apply migration

Error message:

```
Error
Received the following error message:
Could not apply migration: SQLSTATE[HY000]: General error: 1005 Can't create table 'database.#

Please try to fix this error and restart the update.
Response
{"valid":false,"errorMsg":"Could not apply migration: SQLSTATE[HY000]: General error: 1005 Can
```

In this case one specific migration step can not be proceeded. You can find the migration steps in the directory `update-assets/migration` or on [GitHub](#) with the specific migration number. The number of the migration that is causing the error is named in the error message and you can with this you can look into the migration file to check the different queries that will be executed with this migration step.

Possible solutions:

- 1. Database strict mode still active.
- 2. Foreign keys defective ([to the wiki article](#)).
- 3. Check if you have enough free disk space.
- 4. System requirements: Database or MySQL version unfulfilled.
- 5. Error at execute an SQL delta (Checking the table s_schema_version - siehe above).
- 6. Table is supposed to be created that is already existing for example because the database has not been reset before restoring the backup when you encountered an update error once before.

SmartyCompilerException

The exception '*SmartyCompilerException*' with message '*Syntax Error in template "/x/x/x.tpl"...{s name='IndexXmlLang'}{/s} unknown tag "s"*' occurs when using incompatible plugins. The complete exception can be displayed while activating the debug-mode of shopware :

<https://developers.shopware.com/developers-guide/shopware-config/#exceptions>

Sitemap adjustments via config.php

You have the possibility to define exclusions for the sitemap via config.php or add your own URLs. You can find the config.php in the Shopware main directory on the web server.

Please note that due to incorrect entries in the config.php your shop may not be able to be accessed (correctly). Therefore we recommend that only experienced users carry out the adjustments described below.

Add an additional URL

To integrate an additional URL into the sitemap, an entry of the type custom_urls is added. The following information is required:

- url: The desired URL to be included in the sitemap
- priority: Priority of this URL compared to other URLs of your shop. This has no effect on the comparison to other websites/shops. Valid entries are values between 0 and 1; the default is 0.5
- changeFreq: How often is the content of the page expected to change. Possible are: always, hourly, daily, weekly, monthly, yearly, never
- lastMod: Date when the page was last modified. Format Y-m-d H:i:s
- shopId: For which Shop-ID should the URL be written in the sitemap? 0 = all shops. You can find the IDs of the shops in the database table "s_core_shops" in the column "id".

Example

The URL "myUrl.de" should be integrated for all shops. As priority 0.4 is deposited and the change is to take place weekly.

```
'sitemap' => [
  'custom_urls' => [
    [
      'url' => 'myUrl.com',
      'priority' => 0.4,
      'changeFreq' => 'weekly',
      'lastMod' => '2019-01-01 12:00:12',
      'shopId' => 0
    ]
  ]
],
```

Remove certain URLs from the sitemap

If certain pages should not be listed in the sitemap, this can be defined in more detail using `excluded_urls`. The following information is required for this:

- resource: Defines which page type should be excluded. Possible:
 - '\Shopware\Models\Article\Article::class': Exclusion of products.
 - '\Shopware\Models\Emotion\Emotion::class': exclusion of shopping worlds.
 - '\Shopware\Models\Article\Supplier::class': exclusion of manufacturer pages.
 - '\Shopware\Models\Blog\Blog::class': exclusion of blog articles.
 - '\Shopware\Models\Category\Category::class': exclusion of category pages.
 - '\Shopware\Models\Site\Site::class': exclusion of shop pages.
- identifier: The ID of the selected resource. 0 = completely excluded. The IDs of the entities from the resource can be found in the following places
 - product: Database table "s_articles" in the column "id"
 - campaign: database table "s_emotion" in the column "id".
 - manufacturer: database table "s_articles_supplier" in the column "id".
 - blog: database table "s_blog" in the column "id"
 - category: In the backend in the category administration in the heading "General Settings" in the entry System ID. Or in the database table "s_categories" in the column "id".
 - static: In the backend in the shop page administration (Contents > shop pages) in the menu on the left side in brackets behind the name of the shop page. Or in the database table "s_cms_static" in the column "id".
- shopId: The ID of the shop for which the exclusion should apply. 0 = exclusion for all shops. You can find the IDs of the shops in the database table "s_core_shops" in the column "id".

Example

Product

The product with ID 123 should not be included in the sitemap of the shop with ID 2.

```
'sitemap' => [
  'excluded_urls' => [
    [
      'resource' => \Shopware\Models\Article\Article::class,
      'identifier' => '123',
      'shopId' => 2
    ],
  ],
],
```

Shopping world

The shopping world with the ID 21 is to be removed from the sitemap for all shops.

```
'sitemap' => [
  'excluded_urls' => [
    [
      'resource' => \Shopware\Models\Emotion\Emotion::class,
      'identifier' => '21',
      'shopId' => 0
    ],
  ],
],
```

Category and shop pages combined

The category with the ID 246 and all shop pages should be removed from the sitemap for all shops.

```
'sitemap' => [
  'excluded_urls' => [
    [
      'resource' => \Shopware\Models\Category\Category::class,
      'identifier' => '246',
      'shopId' => 0
    ],
    [
      'resource' => \Shopware\Models\Site\Site::class,
      'identifier' => '0',
      'shopId' => 0
    ],
  ],
],
```

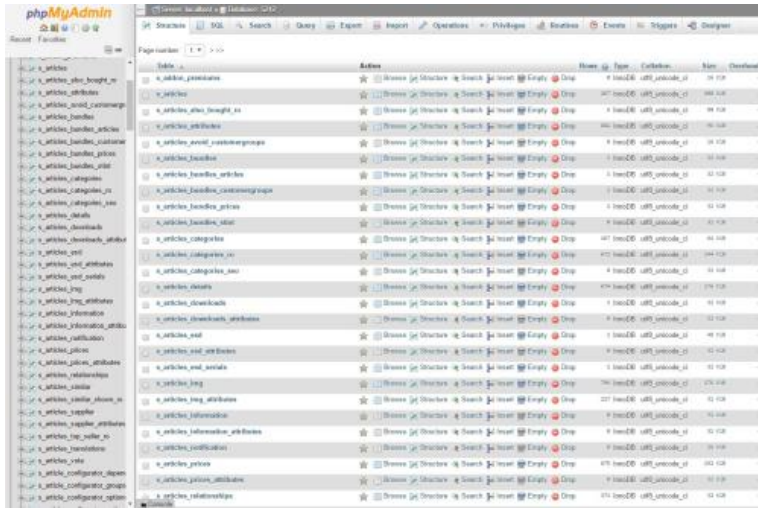
SQL Tips & Tricks

We recommend to only execute those statements with the necessary knowledge.

The provided information were tested at the release of this article, but can be change during the releases of Shopware. **The application of this tutorial appears on your own risk!**

Make a **Backup** of your shop before your start that you can use in case of failure. Keep in mind, that this is adapting Shopware an this is **not officially supported!**

In this article you find some useful SQL-statements that help you for some use cases. For the access to the database we would recommend the administration tool phpMyAdmin. This allows a graphical view of the database and makes it easier to execute the statements. Of course the given statements can be used with any other MySQL-client as well.



Customer data

Filter duplicate customer numbers

In case you have duplicate customer numbers in your system there can be errors and inconsistent datasets. With this statement you can display the duplicate customer numbers.

```
SELECT userID, COUNT( userID ) AS number
FROM `s_user_billingaddress`
GROUP BY `userID`
HAVING (COUNT( userID ) > 1
)
```

Item data

Filter duplicate item numbers

In the backend the creation of items with an already used item number is not possible. However interfaces or import-scripts can bypass this mechanism, because of that it might come to errors. This statement allows you to easily find double items.

```
SELECT ordernumber, COUNT( ordernumber ) AS number
FROM `s_articles_details`
GROUP BY `ordernumber`
HAVING (COUNT( ordernumber ) > 1
);
```

Filter duplicate SEO-URL's in s_core_rewrite_urls

If a specific SEO-link in your shop can not be generated, the id is often already in use. This statement sorts the entries and shows you possible duplicates.

```
SELECT *
FROM `s_core_rewrite_urls`
WHERE main =1
GROUP BY `org_path`, `main`, `subshopID`, `path`
HAVING COUNT(*) >1
```

Filter active products which are still not shown in the frontend

In some cases (often because of a wrong synchronizing of the ERP system) active items are not displayed in the frontend. The cause of this can be that the entry in the s_articles table has the value 1 for the active column, but the associated entry in the s_articles_details has the value 0 for this column. This statement shows you the affected items.

```
SELECT a.name, ad.ordernumber, a.active AS active1, ad.active AS active2
FROM `s_articles` AS a
JOIN s_articles_details AS ad
ON a.id = ad.articleID
WHERE a.active <> ad.active
```

Shop broken items without entry in the s_articles_details table

Sometimes items are created broken in the database by using plugins or interfaces. One clue for this could be a missing entry in the s_articles_details table. This statements shows such broken items, which should be recreated completely anew immediately.

```
SELECT *
FROM s_articles a
LEFT JOIN
s_articles_details ad
ON ad.articleID = a.id
WHERE ad.id IS NULL
```

Remove broken variants / variant-corpses

With this statement you can identify or remove variant options that are (no longer) connected. Those can cause inconsistent data and missing, wrong item displays in the frontend.

configurator_set_id IS NOT NULL

```
DELETE d.*, atr.*
-- SELECT d.*, atr.*
FROM s_articles_details d
JOIN s_articles a ON d.articleID = a.id AND a.configurator_set_id IS NOT NULL
JOIN s_articles_attributes atr ON d.id = atr.articleID
LEFT JOIN s_article_configurator_option_relations cr ON cr.article_id = d.id
WHERE cr.id IS NULL AND d.articleID = a.id;
```

```
DELETE r
-- SELECT r.*
FROM `s_article_configurator_option_relations` r LEFT JOIN s_articles_details d ON d.id = r.ar
```

configurator_set_id IS NULL and additional check for orphan variants

```
DELETE d.*, atr.*
-- SELECT d.*, atr.*
```

```

FROM s_articles_details d
JOIN s_articles a ON d.articleID = a.id AND a.configurator_set_id IS NULL
JOIN s_articles_attributes as atr ON d.id = atr.articleDetailsID
LEFT JOIN s_article_configurator_option_relations cr ON cr.article_id = d.id
WHERE cr.id IS NULL AND d.articleID = a.id and d.kind = 2;

```

Set variant preselection

With this statement variant items, that do not have a preselection, get the first variant as a preselection automatically. Without a preselection the variant item is not shown in the frontend properly.

```

UPDATE s_articles a
LEFT JOIN s_articles_details d
ON d.id = a.main_detail_id
SET a.main_detail_id = (
    SELECT id FROM s_articles_details WHERE articleID = a.id LIMIT 1
)
WHERE d.id IS NULL;
UPDATE s_articles a, s_articles_details d
SET d.kind = 1
WHERE d.id = a.main_detail_id;

```

Fix missing category assignment in the table s_article_categories_ro

If this assignment is missing items are shown as "not available" in the frontend - this statement fills the given table anew for all items. Especially for many items this has a higher performance than recreating the category-tree.

```

INSERT IGNORE INTO `s_articles_categories_ro` (`articleID`,
`categoryID`, `parentCategoryID`)
SELECT ac.articleID, c2.id AS categoryID, c.id AS parentCategoryID
FROM s_articles_categories ac
JOIN s_categories c
ON c.id = ac.categoryID
JOIN s_categories c2
ON c2.id = c.id
OR c.path LIKE CONCAT('%|', c2.id, '|%')
ORDER BY ac.articleID, c.id, c2.id

```

Fix preselected images for the suggestion search

This allows you to correct missing preselected images in different areas (item view in the backend, suggestion search, and more). This is sometimes necessary after an update from Shopware 4 to Shopware 5.

```

UPDATE s_articles_img i, s_media m
SET i.media_id = m.id
WHERE m.path = CONCAT('media/image/', i.img, '.', i.extension);

```

Deleting property options not used in any products

This query deletes all property options and the corresponding attributes that are currently not linked to a product / unused in the shop.

```

DELETE v, va

```

```

FROM `s_filter_values` v
LEFT JOIN
s_filter_articles a
ON a.valueID = v.id
JOIN s_filter_values_attributes va
ON v.id = va.valueID
WHERE a.valueID IS NULL

```

Orders

Display duplicate ordernumbers

If you have double ordernumbers in shopware you get an error in the backend and because of this you should not have those double ordernumbers. This might still occur because of plugins or interfaces. This statement displays you those double ordernumbers.

```

SELECT ordernumber, COUNT(ordernumber) AS amount
FROM s_order
GROUP BY ordernumber
HAVING ( COUNT(ordernumber) > 1 )

```

Alternatively you can also copy the orders in a temporary order table and mark the double ordernumbers in the original table.

```

CREATE TABLE `s_order_tmp` LIKE `s_order`;
INSERT INTO `s_order_tmp` SELECT * FROM `s_order`;
UPDATE s_order
SET ordernumber = CONCAT(ordernumber , '-doppelt')
WHERE id IN (SELECT max(id) FROM s_order_tmp GROUP BY ordernumber
HAVING ( COUNT(ordernumber) > 1 )) AND ordernumber <> 0;
DROP TABLE `s_order_tmp`;

```

In this case there is a "-double" added to the double ordernumbers.

Please mind, that the ordernumber "0" marks cancelled carts and this number may occur multiple times without causing any problem.

Migrate customers and orders of a database backup

It might occur that because of a problem or the usage of test-systems some orders are not in the live-database, but the ids of the orders are already in use by new orders. Currently it is not possible to migrate such orders and customers out of a backup database, but this query should help you with this.

Please mind that your system might contain further columns, tables or needs you might have to add. Also payments might have to be controlled manually. Afterwards you should also execute the above query for double ordernumbers so the ordernumbers can be checked after the migration in the new backend.

DBOLDNAME has to be replaced by the database name you want to migrate the data from and the IdOfFirstCustomerToMigrate and NumberOfFirstOrderToMigrate by the ordernumber/id of the first element you want to migrate.


```

ALTER TABLE s_user ADD idold INT(11) NULL AFTER customernumber;
INSERT INTO s_user (idold, password, encoder, email, active, accountmode, cONfirmationkey, pa
INSERT INTO s_user_attributes (userID) SELECT id FROM s_user WHERE id NOT IN (SELECT userid FR
INSERT INTO s_user_addresses (user_id, company, department, salutation, title, firstname, last
INSERT INTO s_user_addresses_attributes (address_id) SELECT id FROM s_user_addresses WHERE id
INSERT INTO `s_user_billingaddress` (`userID`, `company`, `department`, `salutation`, `firstna
INSERT INTO s_user_billingaddress_attributes (billingID) SELECT id FROM s_user_billingaddress
INSERT INTO `s_user_shippingaddress` (`userID`, `company`, `department`, `salutation`, `firstn
INSERT INTO s_user_shippingaddress_attributes (shippingID) SELECT id FROM s_user_shippingaddre
UPDATE s_user AS u JOIN s_user_addresses AS ua ON u.id = ua.user_ID SET u.default_billing_addr
UPDATE s_user AS u JOIN s_user_addresses AS ua ON u.id = ua.user_ID SET u.default_shipping_add
ALTER TABLE s_order ADD idold INT(11) NULL AFTER deviceType;
ALTER TABLE s_order_details ADD didold INT(11) NULL AFTER pack unit;
INSERT INTO s_order (idold, ordernumber, userID, invoice_amount, invoice_amount_net, invoice_s
INSERT INTO s_order_attributes (orderID, attribute1, attribute2, attribute3, attribute4, attri
SELECT o.id, oold.attribute1, oold.attribute2, oold.attribute3, oold.attribute4, oold.attribut
INSERT INTO s_order_details (didold, orderID, ordernumber, articleID, articleordernumber, pric
SELECT posold.id, o.id, posold.ordernumber, posold.articleID, posold.articleordernumber, posol
INSERT INTO s_order_details_attributes (detailID, attribute1, attribute2, attribute3, attribut
SELECT d.id, dold.attribute1, dold.attribute2, dold.attribute3, dold.attribute4, dold.attribut
INSERT INTO s_order_billingaddress (userID, orderID, company, department, salutation, customer
INSERT INTO s_order_billingaddress_attributes (billingID) SELECT id FROM s_order_billingaddress
INSERT INTO s_order_shippingaddress (userID, orderID, company, department, salutation, firstna
SELECT u.id, o.id, oldship.company, oldship.department, oldship.salutation, oldship.firstname,
INSERT INTO s_order_shippingaddress_attributes (shippingid) SELECT id FROM s_order_shippingadd
ALTER TABLE `s_user` DROP `idold`;
ALTER TABLE `s_order` DROP `idold`;
ALTER TABLE `s_order_details` DROP `didold`;

```

System / Configuration

Deactivate non default plugins

This statement deactivates all plugins that have been uploaded in the plugin manager or have been installed from the community store. This is very useful for debugging errors, because with this you can easily find or exclude the influence of plugins for your behaviour.

First you create a temporary table with the original values to restore them later.

```

CREATE TABLE `s_core_plugins_tmp` LIKE `s_core_plugins`;
INSERT INTO `s_core_plugins_tmp` SELECT * FROM `s_core_plugins`;

```

Now you disable the plugins with this statement.

```

UPDATE `s_core_plugins` SET `active` = 0 WHERE `source` <> 'Default' AND author <> 'shopware AG

```

For the restoring of the plugin values you put the data from the temporary table back in the original table and delete the no longer needed temporary table afterwards.

```

UPDATE s_core_plugins AS p JOIN s_core_plugins_tmp AS pt ON p.id = pt.id SET p.active = pt.act
DROP TABLE s_core_plugins_tmp;

```

Resetting the database - deleting test data

Please mind that the data you delete might be connected and used in further elements and tables. For example customers and products are used and referred to in the orders, so deleting them might lead to a no longer existing connection in those.

Notice to foreign keys

According to the configuration and version of your MySQL server it may happen, that the foreign key check doesn't appear as a checkbox before you submit the SQL query, but nevertheless need to be deactivated. In this case, you need to enter this in the query, so set SET foreign_key_checks = 0; as new first line to disable the foreign key check and use SET foreign_key_checks = 1; as last line to reactivate it.

```
SET foreign_key_checks = 0;
TRUNCATE `s_user`;
TRUNCATE `s_user_addresses`;
TRUNCATE `s_user_addresses_attributes`;
TRUNCATE `s_user_attributes`;
TRUNCATE `s_user_billingaddress`;
TRUNCATE `s_user_billingaddress_attributes`;
TRUNCATE `s_user_shippingaddress`;
TRUNCATE `s_user_shippingaddress_attributes`;
SET foreign_key_checks = 1;
```

Deleting customers

This query truncates all customer data from the user tables.

```
TRUNCATE `s_user`;
TRUNCATE `s_user_addresses`;
TRUNCATE `s_user_addresses_attributes`;
TRUNCATE `s_user_attributes`;
TRUNCATE `s_user_billingaddress`;
TRUNCATE `s_user_billingaddress_attributes`;
TRUNCATE `s_user_shippingaddress`;
TRUNCATE `s_user_shippingaddress_attributes`;
```

This query deletes all customers for a WHERE condition you have to add yourself.

```
DELETE aa.*, ad.*, ua.*, ba.*, ub.*, sa.*, us.*, u.*
FROM `s_user` AS u
LEFT JOIN s_user_addresses AS ad ON u.id = ad.user_id
LEFT JOIN s_user_addresses_attributes AS aa ON ad.id = aa.address_id
LEFT JOIN s_user_attributes AS ua ON u.id = ua.userID
LEFT JOIN s_user_billingaddress AS ub ON u.id = ub.userID
LEFT JOIN s_user_billingaddress_attributes AS ba ON ub.id = ba.billingID
LEFT JOIN s_user_shippingaddress AS us ON u.id = us.userID
LEFT JOIN s_user_shippingaddress_attributes AS sa ON us.id = sa.shippingID
WHERE XXXX;
```

Deleting orders

With this query you truncate all order data from the database.

```
TRUNCATE `s_order`;
TRUNCATE `s_order_attributes`;
TRUNCATE `s_order_basket`;
TRUNCATE `s_order_basket_attributes`;
TRUNCATE `s_order_billingaddress`;
TRUNCATE `s_order_billingaddress_attributes`;
TRUNCATE `s_order_comparisons`;
TRUNCATE `s_order_details`;
TRUNCATE `s_order_details_attributes`;
TRUNCATE `s_order_documents`;
TRUNCATE `s_order_documents_attributes`;
TRUNCATE `s_order_esd`;
TRUNCATE `s_order_history`;
TRUNCATE `s_order_notes`;
TRUNCATE `s_order_shippingaddress`;
```

```
TRUNCATE `s_order_shippingaddress_attributes`;
```

With this query you can delete the orders for a WHERE condition you have to add yourself.

```
DELETE osa.*, os.*, odda.*, odd.*, oda.*, od.*, oba.*, ob.*, oa.*, o.*
FROM s_order AS o LEFT JOIN s_order_attributes AS oa ON o.id = oa.orderID
LEFT JOIN s_order_billingaddress AS ob ON o.id = ob.orderID
LEFT JOIN s_order_billingaddress_attributes AS oba ON ob.id = oba.billingID
LEFT JOIN s_order_details AS od ON o.id = od.orderID
LEFT JOIN s_order_details_attributes AS oda ON od.id = oda.detailID
LEFT JOIN s_order_documents AS odd ON o.id = odd.orderID
LEFT JOIN s_order_documents_attributes AS odda ON odd.id = odda.documentID
LEFT JOIN s_order_shippingaddress AS os ON o.id = os.orderID
LEFT JOIN s_order_shippingaddress_attributes AS osa ON os.id = osa.shippingID
WHERE XXXX;
```

Deleting articles

By using this query you clear all product data from the database.

```
TRUNCATE `s_addon_premiums`;
TRUNCATE `s_articles`;
TRUNCATE `s_articles_also_bought_ro`;
TRUNCATE `s_articles_attributes`;
TRUNCATE `s_articles_avoid_customergroups`;
TRUNCATE `s_articles_categories`;
TRUNCATE `s_articles_categories_ro`;
TRUNCATE `s_articles_categories_seo`;
TRUNCATE `s_articles_details`;
TRUNCATE `s_articles_downloads`;
TRUNCATE `s_articles_downloads_attributes`;
TRUNCATE `s_articles_esd`;
TRUNCATE `s_articles_esd_attributes`;
TRUNCATE `s_articles_esd_serials`;
TRUNCATE `s_articles_img`;
TRUNCATE `s_articles_img_attributes`;
TRUNCATE `s_articles_information`;
TRUNCATE `s_articles_information_attributes`;
TRUNCATE `s_articles_notification`;
TRUNCATE `s_articles_prices`;
TRUNCATE `s_articles_prices_attributes`;
TRUNCATE `s_articles_relationships`;
TRUNCATE `s_articles_similar`;
TRUNCATE `s_articles_similar_shown_ro`;
TRUNCATE `s_articles_supplier`;
TRUNCATE `s_articles_supplier_attributes`;
TRUNCATE `s_articles_top_seller_ro`;
TRUNCATE `s_articles_translations`;
TRUNCATE `s_articles_vote`;
TRUNCATE `s_article_configurator_dependencies`;
TRUNCATE `s_article_configurator_groups`;
TRUNCATE `s_article_configurator_options`;
TRUNCATE `s_article_configurator_option_relations`;
TRUNCATE `s_article_configurator_price_variations`;
TRUNCATE `s_article_configurator_sets`;
TRUNCATE `s_article_configurator_set_group_relations`;
TRUNCATE `s_article_configurator_set_option_relations`;
TRUNCATE `s_article_configurator_templates`;
TRUNCATE `s_article_configurator_templates_attributes`;
TRUNCATE `s_article_configurator_template_prices`;
TRUNCATE `s_article_configurator_template_prices_attributes`;
TRUNCATE `s_article_img_mappings`;
TRUNCATE `s_article_img_mapping_rules`;
TRUNCATE `s_filter_articles`;
```

By using this query you can delete all products for a specific WHERE condition you have to add yourself.

```
DELETE aimr.*, aim.*, acta.*, actpa.*, actp.*, act.*, acor.*, av.*, atr.*, atop.*, ASiro.*, AS
FROM s_articles AS a
LEFT JOIN s_articles_also_bought_ro AS bro ON a.id = bro.article_id
```

```

LEFT JOIN s_articles_details AS ad ON a.id = ad.articleID
LEFT JOIN s_articles_attributes AS aa ON ad.id = aa.articleIDdetailsID
LEFT JOIN s_articles_avoid_customerGroups AS acu ON a.id = acu.articleID
LEFT JOIN s_articles_categories AS ac ON a.id = ac.articleID
LEFT JOIN s_articles_categories_ro AS acr ON a.id = acr.articleID
LEFT JOIN s_articles_categories_seo AS acs ON a.id = acs.article_id
LEFT JOIN s_articles_downloads AS ado ON a.id = ado.articleID
LEFT JOIN s_articles_downloads_attributes AS adoa ON ado.id = adoa.downloadID
LEFT JOIN s_articles_esd AS aesd ON a.id = aesd.articleID
LEFT JOIN s_articles_esd_attributes AS aesda ON aesd.id = aesda.esdID
LEFT JOIN s_articles_esd_serials AS aesds ON aesd.id = aesds.esdID
LEFT JOIN s_articles_img AS ai ON a.id = ai.articleID
LEFT JOIN s_articles_img_attributes AS aia ON ai.id = aia.imageID
LEFT JOIN s_articles_information AS ain ON a.id = ain.articleID
LEFT JOIN s_articles_information_attributes AS aina ON ain.id = aina.informationID
LEFT JOIN s_articles_notifications AS an ON ad.orderNumber = an.orderNumber
LEFT JOIN s_articles_prices AS ap ON a.id = ap.articleID
LEFT JOIN s_articles_prices_attributes AS apa ON ap.id = apa.priceID
LEFT JOIN s_articles_relationships AS ar ON a.id = ar.articleID
LEFT JOIN s_articles_similar AS ASi ON a.id = ASi.articleID
LEFT JOIN s_articles_similar_shown_ro AS ASiro ON a.id = ASiro.article_id
LEFT JOIN s_articles_top_seller_ro AS atop ON a.id = atop.article_id
LEFT JOIN s_articles_translations AS atr ON a.id = atr.articleID
LEFT JOIN s_articles_vote AS av ON a.id = av.articleID
LEFT JOIN s_article_configurator_option_relations AS acor ON ad.id = acor.article_id
LEFT JOIN s_article_configurator_templates AS act ON a.id = act.article_id
LEFT JOIN s_article_configurator_template_prices AS actp ON act.id = actp.template_id
LEFT JOIN s_article_configurator_template_prices_attributes AS actpa ON actp.id = actpa.templa
LEFT JOIN s_article_configurator_templates_attributes AS acta ON act.id = acta.template_id
LEFT JOIN s_article_img_mappings AS aim ON ai.id = aim.image_id
LEFT JOIN s_article_img_mapping_rules AS aimr ON aim.id = aimr.mapping_id
WHERE XXXX;

```

Resetting number ranges

In this version, the number ranges in Shopware are deleted and reapplied with default values.

```
DROP TABLE `s_order_number`;
```

```

CREATE TABLE IF NOT EXISTS `s_order_number` (
  `id` int(11) NOT NULL AUTO_INCREMENT,
  `number` int(20) NOT NULL,
  `name` varchar(255) COLLATE utf8_unicode_ci NOT NULL,
  `desc` varchar(255) COLLATE utf8_unicode_ci NOT NULL,
  PRIMARY KEY (`id`),
  UNIQUE KEY `name` (`name`)
) ENGINE=InnoDB DEFAULT CHARSET=utf8 COLLATE=utf8_unicode_ci AUTO_INCREMENT=10 ;

```

```

INSERT INTO `s_order_number` (`id`, `number`, `name`, `desc`) VALUES
(1, 20000, 'user', 'Kunden'),
(2, 20000, 'invoice', 'Bestellungen'),
(3, 20000, 'doc_1', 'Lieferscheine'),
(4, 20000, 'doc_2', 'Gutschriften'),
(5, 20000, 'doc_0', 'Rechnungen'),
(6, 10000, 'articleordernumber', 'Artikelbestellnummer '),
(7, 10000, 'sSERVICE1', 'Service - 1'),
(8, 10000, 'sSERVICE2', 'Service - 2'),
(9, 110, 'blogordernumber', 'Blog - ID');

```

Resetting statistics

Some statistics are obtained from the orders and customer data. The other values can be deleted with the following query:

```

TRUNCATE `s_statistics_currentusers`;
TRUNCATE `s_statistics_pool`;
TRUNCATE `s_statistics_referer`;
TRUNCATE `s_statistics_search`;
TRUNCATE `s_statistics_visitors`;

```

Resetting categories and shopping worlds

With this version, all categories as well as created shopping worlds will be removed. The main category will be automatically reset to "Deutsch" with the system ID 3.

```
TRUNCATE `s_categories`;
TRUNCATE `s_categories_attributes`;
TRUNCATE `s_categories_avoid_customergroups`;
TRUNCATE `s_emotion`;
TRUNCATE `s_emotion_attributes`;
TRUNCATE `s_emotion_categories`;
TRUNCATE `s_emotion_element`;
TRUNCATE `s_emotion_element_value`;

INSERT INTO `s_categories` (`id`, `parent`, `path`, `description`, `position`, `left`, `right`
(1, NULL, NULL, 'Root', 0, 0, 0, 0, '2012-07-30 15:24:59', '2012-07-30 15:24:59', NULL, NULL,
(3, 1, NULL, 'Deutsch', 0, 0, 0, 0, '2012-07-30 15:24:59', '2012-07-30 15:24:59', NULL, NULL,

INSERT INTO `s_categories_attributes` (`id`, `categoryID`, `attribute1`, `attribute2`, `attrib
(1, 3, NULL, NULL, NULL, NULL, NULL, NULL);
```

Set up SSL certificate in the shop

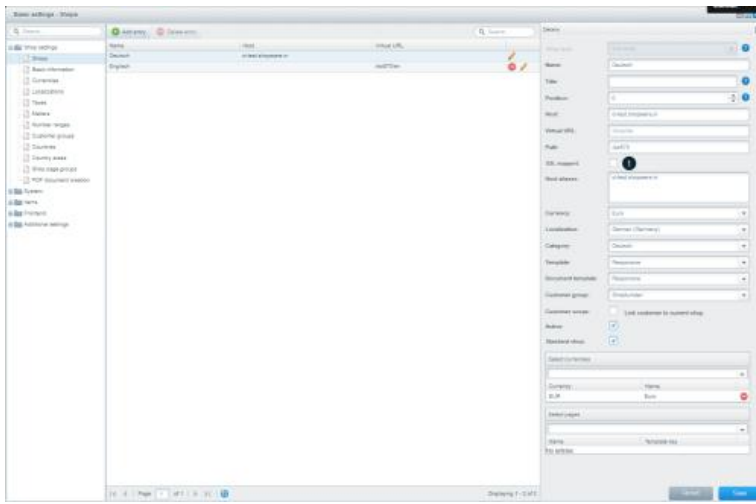
SSL / TLS

SSL / TLS are general terms for a network protocol for the secure transmission of data. TLS is the successor to SSL. If you secure the connections to the shop with SSL / TLS, the data between the shop and the connected user are transmitted in encrypted form and thus offer far less scope for attack than an unencrypted connection.

You can decide for yourself which technology you should use, but the current browsers already give various recommendations here, for example, SHA-1 algorithms should no longer be used from 2017, as they are no longer classified as secure. In principle, however, it is advisable to use current technologies such as: TLS 1.2 with SHA-2 algorithm. Which certificate you use is completely irrelevant for Shopware, the main thing is that it is technically correct and installed.

Activate SSL

Once the certificate was installed, you have to activate SSL in Shopware, how that works, you can see here.



Just open the shop settings in "*Configuration > Basic settings > Shop settings > Shops*" and choose the desired shop, which should get SSL activated. Activate the option **SSL support (1)** to secure all frontend connections.

Redirect all requests

On simple cases, you might just want to redirect all HTTP requests to the equivalent HTTPS route. In these cases, you can use the following generic rule in the htaccessfile which you can find in the mainfolder of your shop:

```
RewriteCond %{HTTPS} !=on  
RewriteRule (.*) https://%{HTTP_HOST}%{REQUEST_URI} [L,R=301]
```

Example:

- `http://foo.example.com -> https://foo.example.com`
- `http://bar.example.com -> https://bar.example.com`

Redirect all subdomains

```
RewriteCond %{HTTPS} !=on  
RewriteRule (.*) https://secure.example.com%{REQUEST_URI} [L,R=301]
```

Example:

- `http://foo.example.com -> https://secure.example.com`
- `http://bar.example.com -> https://secure.example.com`

Source domain specific

Should your shop require a more specific approach, you can also use a per domain approach. The following example show how you can redirect all unsecure requests to <http://unsecure-domain.com> to <https://secure-domain.com>

```
RewriteCond %{HTTPS} !=on
RewriteCond %{HTTP_HOST} (www\.)?unsecure-domain\.com [NC]
RewriteRule ^(.*)$ https://secure-domain\.com%{REQUEST_URI} [R=301,L]
```

Debugging the Search

Introduction

Your shop search does not provide you with the results you are looking for? Then you are exactly right here! We will check some points so you can quickly check if the shopware search does exactly what the search should do.

Please note that the search will not give you any results if more than $\approx 80\%$ of your article master would be returned as results.

Check the search index

First of all, it is important to determine whether the search index was built up completely. This can be found in your database. Go to your database and search for the table `s_search_keywords`. If there are entries for the column "Soundex" set to NULL for a keyword which have no number as an entry, this is an indication that the search index was not built up correctly.

+ Options

	id	keyword	soundex	cologne_phonetic	metaphone
<input type="checkbox"/> Edit <input type="checkbox"/> Copy <input type="checkbox"/> Delete	2604	calculation	C42435	454526	KLKLN
<input type="checkbox"/> Edit <input type="checkbox"/> Copy <input type="checkbox"/> Delete	2605	sold	S430	852	SLT
<input type="checkbox"/> Edit <input type="checkbox"/> Copy <input type="checkbox"/> Delete	2606	out	O300	2	OT
<input type="checkbox"/> Edit <input type="checkbox"/> Copy <input type="checkbox"/> Delete	2607	reviews	R120	7338	RFS
<input type="checkbox"/> Edit <input type="checkbox"/> Copy <input type="checkbox"/> Delete	2608	image	I520	64	IMJ
<input type="checkbox"/> Edit <input type="checkbox"/> Copy <input type="checkbox"/> Delete	2615	english	E5242	6458	ENKLN
<input type="checkbox"/> Edit <input type="checkbox"/> Copy <input type="checkbox"/> Delete	2616	time	T500	26	TM
<input type="checkbox"/> Edit <input type="checkbox"/> Copy <input type="checkbox"/> Delete	2617	electronic	E423652	582768	ELKTRNK
<input type="checkbox"/> Edit <input type="checkbox"/> Copy <input type="checkbox"/> Delete	2618	bakery	B260	147	BKR
<input type="checkbox"/> Edit <input type="checkbox"/> Copy <input type="checkbox"/> Delete	2619	products	P63232	17288	PRTKTS
<input type="checkbox"/> Edit <input type="checkbox"/> Copy <input type="checkbox"/> Delete	2620	fish	F200	38	FX
<input type="checkbox"/> Edit <input type="checkbox"/> Copy <input type="checkbox"/> Delete	2621	sweets	S320	838	SWTS
<input type="checkbox"/> Edit <input type="checkbox"/> Copy <input type="checkbox"/> Delete	2622	women	W500	366	WMN
<input type="checkbox"/> Edit <input type="checkbox"/> Copy <input type="checkbox"/> Delete	2623	men	M000	66	MN

Strategy for building the search index

To be able to analyze the described behavior more precisely you should first check how the search index is built up in your shop. To do this, go to Settings > Cache/Performance > Settings (tab) > search (left), which is selected here as a corresponding strategy for the structure for the search index.

If the "Live" option is used to build the search cache, this generates a certain number of entries for each page call in the frontend or checks for actuality. The search index is updated very frequently in your shop. Especially since the Fuzzy search is a "learning" search, some search queries would be necessary here in order to provide optimal search results. During this time suboptimal search results could occur. The update strategy "Live" also has the disadvantage that additional server load is generated with each call in the front end. In terms of performance, this setting is not recommended.

To limit this behavior, we recommend you change the search strategy to "Cronjob". With the use of cronjobs you can move the load also collected at night.

If this does not help again, we recommend you that you try to build the search index on the console: `php bin/console sw:refresh:search:index`

Did you get an error message similar like this:

```
PHP Fatal error: Allowed memory size of 536870912 bytes exhausted (tried to allocate 72 bytes)
```

So you should try to set the `memory_limit` higher. Depending on the system this already works with the following extension of the CLI call with: `-d memory_limit = 4096M` Thus: `php -d memory_limit = 4096M bin/console sw:refresh:search:index`

If the search index cannot be built up due to a standard batch size of 4000, you can get around this quickly and easily by adapting the `config.php` on your server.

To do this, put the following into your `config.php`:

```
'search' => [
  'indexer' => [
    'batchsize' => 3250,
  ],
],
```

If you have further questions about the `memory_limit`, you can contact your hosting partner directly.

Debug with the Fuzzy Search preview

If you do not find the desired results with your search, you can use the Fuzzy Search preview here. If the information of the preview with the data of the factor and the relevance does not provide the desired information, You can use Google Chrome's Developer Tools.

Here you can find a list-egg under the Network

DeinShop.de/backend/SwagFuzzyPreview/list?_dc=1486566464437&shopId=1&page=1&start=0&limit=20&filter=%5B%7B%22property%22%3A%22search%22


```

▼ {success: true, data: [{(name: "Main product with properties", stock: 50, shortDescription: "",,-),,-}],-}
▶ data: [{(name: "Main product with properties", stock: 50, shortDescription: "",,-),,-}]
▼ keywords: [{(id: "2589", term: "main", word: "main", relevance: "400"),,-}]
▼ 0: {id: "2589", term: "main", word: "main", relevance: "400"}
    id: "2589"
    relevance: "400"
    term: "main"
    word: "main"
▼ 1: {id: "3", term: "main", word: "mail", relevance: 75}
    id: "3"
    relevance: 75
    term: "main"
    word: "mail"
▼ 2: {id: "2623", term: "main", word: "men", relevance: 50}
    id: "2623"
    relevance: 50
    term: "main"
    word: "men"
success: true
total: 9

```

With this entry, you can quickly see what the search term is and what keywords are found and used here. In addition, you will get an overview of how many search terms were found. If you would now suspect that your search term is pretty close to the given search terms but nevertheless it is not listed as a term, then you should increase the value "Maximum number of displayed results and the used keywords" for the search. Here you can specify how much deviating keywords are used for the search.

Products with long product names are not found

If you have long product names that you can not find, you can create individual keywords for these long compiled product names. In addition you should increase the relevance for the keyword.

Example: Donaudampfschiffahrtsgesellschaftskapitänsmütze

Keywords: Donau dampf schiff fahrts gesellschaft kapitän mütze Donaudampf schiffahrt gesellschaft kapitänsmütze Donaudampfschiffahrt gesellschaft schiffahrtgesellschaft kapitänsmütze

Check the search logic

If you do not find products when you are using multiple keywords, you can check the search logic used.

You can find them under: **Settings> Preferences> Storefront> Search> "And" Use Search Logic: No**

Afterwards, you can rebuild the search index via: **Settings> Cache / Performance> Settings> Search> Rebuild search index.**

When this is done. Clear the cache (CTRL + ALT + X) and reload the backend (CTRL + F5).

Check product numbers

In general, it is important to make sure that you use the correct article numbers when searching. Information about the article numbers supported in Shopware can be found in the [Wiki article on article numbers](#).

If multiple searchterms switching directly to a product you may have a product with a defective ordernumber. The best way to check this is to look up the database in the table *s_products_details* to see if there is an product that has no or only 2 characters as ordernumber.

General questions about setting up the Smart Search

[Wiki article for the Smart Search](#)

System Requirements / Hosting

Server

The required server hardware is mostly dependent on the number of articles available in your shop and on the expected traffic (visitors per day). The requirements are covered by most hosting providers and certified hosters should be able to provide a suitable configuration.

Requirements (since 5.7)

- Linux-based operating system with Apache 2.2 or 2.4 web server with mod_rewrite module and the possibility to change web server settings in .htaccess.
- PHP 7.4 / 8.0 / 8.1 (since 5.7.8) / 8.2 (since 5.7.17)
7.4.14 and 8.0.1 are not compatible
- MySQL >=5.7 (except 8.0.20, 8.0.21 and 8.0.29)
- Possibility to set up cron jobs
- Minimum 4 GB available hard disk space

Support for PHP 7.4 was already discontinued by PHP. PHP 8.0 will be discontinued soon. You can find more information about this [here](#).

PHP Extensions / Web server

- Apache mod_rewrite
- GD Library version 2.0 or higher
- cURL Library installed
- Character type checking (ctype)
- Document Object Model (dom)
- HASH Message Digest Framework(hash)
- iconv
- JavaScript Object Notation (json)

- Multibyte String (mbstring)
- OpenSSL
- Session Handling (session)
- SimpleXML
- xml
- zip
- zlib
- PDO/MySQL
- INTL

Further compatibilities

- Varnish version 5.0 or higher
- Node-js version 12 or higher
- Redis 5 or higher
- Elasticsearch 7.0 or higher

PHP OPcache

For performance reasons it is strongly recommended to check if the PHP OPcache is enabled.

Minimum web server/PHP-configuration

- memory_limit > 256M
- upload_max_filesize >6M
- allow_url_fopen > activated

Recommended settings

- [APCu](#)
- If the Shopware ESD functions are to be used, the use of Apache **mod_xsendfile** is strongly recommended.

Optional MySQL settings

- MySQL variable **group_concat_max_len** should be set to 2048. (default: 1024)
- **innodb_buffer_pool_size** should be as large as the complete database.
- **query_cache_size** should normally be between 100 - 200MB.

Please note that the values here are to be seen as an absolute minimum. As soon as you have a high number of items in the store or a high traffic is expected and the store has to process many requests at the same time, it is strongly recommended to increase the server resources.

Alternative setup

The requirements listed above represent the recommended setup to install Shopware. Generally, the operation is also possible on similar setups (Mac OS, nginx, MariaDB etc.). However, please take into account that we cannot provide official support for setting up such server systems and setups. The setup of alternative server systems with all necessary requirements is the responsibility of the respective hosting partner.

Elasticsearch

Shopware supports [Elasticsearch](#) Server by default. You can find more information [here](#).

Client

The administration of your store can be done completely online through your web browser. For this it is necessary that you have Javascript and cookies enabled. Through the use of cookies, a unique identification of your person is possible, thus ensuring a high level of security. The use of Javascript increases the usability of the system enormously and allows in conjunction with Ajax as the core technology of Shopware the comfortable and fast administration of your store

Requirements

- In current version: Mozilla Firefox, Chrome, Safari or Internet Explorer version 11 or higher.
- JavaScript and Cookies enabled
- 4 GB RAM
- Dual-core CPU
- Minimum backend resolution: 1366 x 768 pixels

Tracking numbers and tracking

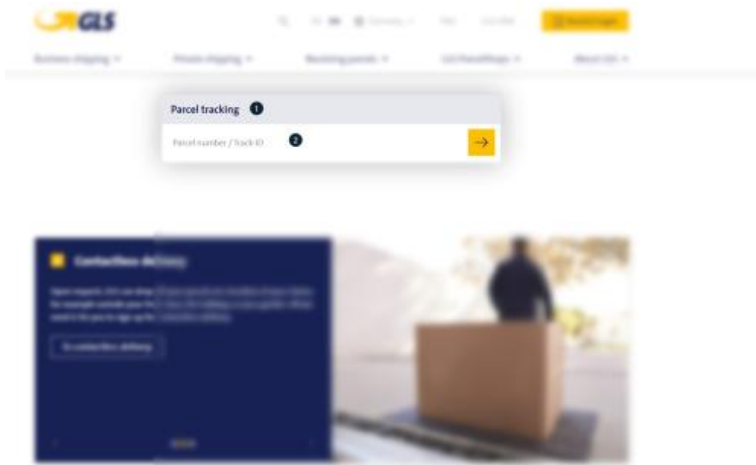
Introduction

This tutorial explains how to insert a tracking link into an email template for your customers. This allows keeping track of their order's delivery status at any time.

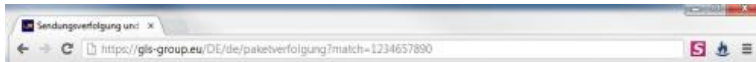
Get tracking links from logistics partners

Find out the required tracking URL of your logistics partner. In this example we will explain how to find a tracking link for GLS. This example is transferable to almost all other logistics companies.

On the GLS Group website you will find a **form (1)** that allows you to enter a tracking ID/code. Enter a **valid parcel number (2)** here:



After you have entered a valid parcel number, you should see the track and trace information. Note the composition of the URL:



In this case, the tracking URL reads as follows:

`https://gls-group.eu/DE/de/paketverfolgung?match=`

The package number `1234567890` will be replaced by a variable later.

Extend email template with tracking URL

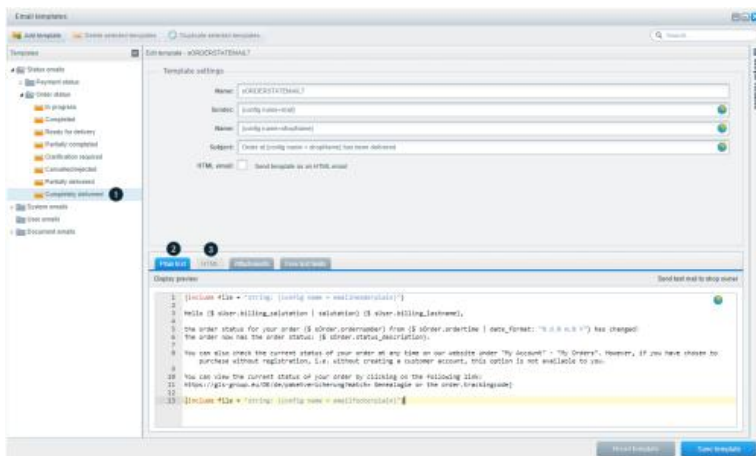
Now switch to the email templates and select the desired template. In our case we have chosen the email template **completely delivered (1)**.

Here we have stored with the tracking link and the required variable for the package number.

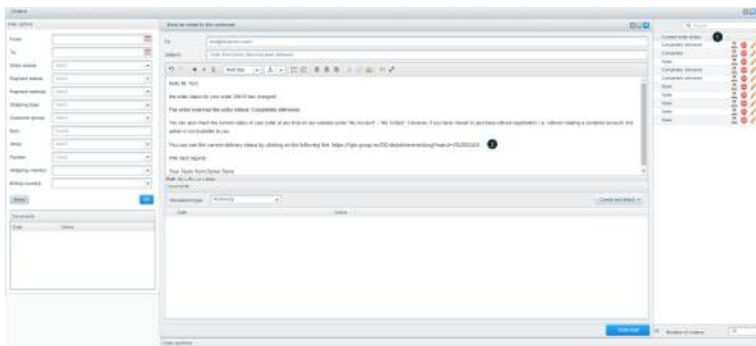
Link (example GLS) for **text emails (2)**

Track & Trace: `https://gls-group.eu/DE/de/paketverfolgung?match={$sOrder.trackingcode}`

As of Shopware 5.2 it is possible to also send the status emails in **HTML format (3)**.



If you now change the status of the order to **completely delivered** under **current order status (1)** in the orders, the email template customized in step 2 will automatically appear with the corresponding **tracking link (2)**.



Within the email templates, there are corresponding variables available for each individual template. You can find out which ones are available in detail for the e-mail template Completely delivered by clicking on the arrow of the variable information. Please note that the variables provided here cannot necessarily be used in other email templates.

Providing tracking URLs in the frontend

Now go to **Configuration/Shipping costs** and enter the tracking URL from step 1. Here you don't have to use HTML or JavaScript, because the link will be built from the template. Just enter the URL from the logistics partner with the variable for the tracking number, all else is accomplished by the template.

The screenshot shows a 'Shipping costs' configuration window. It includes a 'Configuration' section with the following fields:

- Name: Standard shipping
- Description: Description
- Tracking URL: [Track & Trace: https://glz-group.eu/DE/de/paketverfolgung?match={offerPosition.trackingcode}](https://glz-group.eu/DE/de/paketverfolgung?match={offerPosition.trackingcode}) (highlighted with a red circle and '1')
- Comments: (empty)
- Sorting: 0
- Active:
- Shop: All shops
- Customer groups: All groups
- Calculate dispatch costs based on: Weight
- Shipping free from: None
- Shipping type: Default shipping type
- Surcharge type: Display in shipping cart item
- Tax: auto-detection

Below the configuration is a table with columns: From, To, Shipping costs, and Factor(%). The table contains one row: From: 0, To: Arbitrary, Shipping costs: 4.9, Factor(%): (empty). At the bottom are 'Cancel' and 'Save' buttons.

The following is an example of the tracking URL:

```
https://glz-group.eu/DE/de/paketverfolgung?match={offerPosition.trackingcode}
```

As the tracking URL has already been provided in the shipping costs, it is therefore automatically available in your customer account, in the tab **orders (1)** and the **order details (2)**. This is only available for customers that have registered for a customer account.

The screenshot shows the 'DemoShop' customer account 'Orders' page. The page has a navigation bar with 'Home', 'Mountain air & adventure', 'Cooking pleasure & Provence', 'Craft & tradition', and 'Blog'. The 'Customer account' section shows 'Orders'.

The 'Orders' section has a sub-header 'Orders' and a note: 'Here you can find your orders sorted according to date.' Below this is a table with columns: Date, Order number, Shipping type, Order status, and Special offers. The table contains one row:

Date	Order number	Shipping type	Order status	Special offers
22 Jan 2018 09:30:00 AM	20002	GLS	The order has been sent	View

Below the table is a detailed view of the order with columns: Product, Quantity, Price per unit, and Sum. The items are:

- Rainbow trout: Quantity 1, Price per unit 14,90 €*, Sum 14,90 €*
Content: 0.6 Kilogram
Base price: 1 Kilogram = €24.83 *
Current unit price: €14.90 * price
- 3L (M) PANTS ALAGNA WOMAN M: Quantity 1, Price per unit 395,95 €*, Sum 395,95 €*
Current unit price: €395,95 *
- Gallant 250 g: Quantity 0, Price per unit 8,90 €*, Sum 53,40 €*
Content: 0.25 Kilogram
Base price: 1 Kilogram = €35.40 *
Current unit price: €8.90 *

At the bottom, there is a summary section:

From:	22 Jan 2018 09:30:00 AM	Shipping costs:	0 €
Order number:	20002	Total sum:	448,25 €
Shipping type:	GLS		
Tracking your package:	GLS12345, GLS23456, GLS34567		

Page 1 From 1

Customize the email template for multiple tracking numbers

For large orders it is common to split it up into more than one package. That way you will end up with a bunch of different tracking numbers for only one order. In our example, we assume that you include these, separated with a comma in the order details.

Order details: 20001

Overview Details Communication Packages Documents Status history

Order details

Shop:	Demoshop	Referer:
Language:	English (United Kingdom)	IP address:
Order date/time:	Fri Jan 19 2018 11:43:20 GMT+0100 (Mitteleuropäische Zeit)	Device type:
Order number:	20001	
Currency:	EUR	
Total amount:	1,668.15	
Selected shipping type:	Standard shipping	
Partner ID:		
E-Mail:	Demoshop@shopware.com	

Edit order

Paid on:

Tracking code:

Shipping costs (EUR):

Shipping costs net (EUR):

Order status:

Payment status:

Free text fields

Free text fields

No free text fields configured for this module.

[Configure now](#) [Refresh](#)

[Cancel](#) [Save](#)

In this case the example above wouldn't work because you can only transfer one tracking number per link. It is therefore necessary to generate several tracking links for this kind of dispatch. For example, you can extend the email template with the following content:

```
{assign var='Paket' value=1}
{assign var='Tracking' value=', '|explode: {$sOrder.trackingcode}}
{foreach $Tracking as $temp}
Link zu Paket {$Paket}: {$temp}
{assign var='Paket' value={$Paket+1}}
{/foreach}
```

The content of the email could look like this:

Send an email to the customer

To:

Subject:

Hello Mr customer,

The order status of your order 20002 has changed!
Your order now has the following status: Completely delivered.

You can check the current status of your order on our website under "My account" - "My orders" anytime. In case you have purchased without a registration or a customer account, you do not have this option.

Tracking code for package 1: <https://gls-group.eu/DE/de/paketverfolgung?match=GLS12345>
Tracking code for package 2: <https://gls-group.eu/DE/de/paketverfolgung?match=GLS23456>
Tracking code for package 3: <https://gls-group.eu/DE/de/paketverfolgung?match=GLS34567>

Best regards
Your team of Demoshop
Kind Regards,
Your Demoshop team

[Send mail](#)

You can also use the HTML template and add a list element for better representation.

Customize the email template for various logistic companies

If you want to dispatch the order via different logistics companies, but still include the appropriate tracking links, an adaptation of the email template is necessary. The following example shows how it is done using the example of two major logistics companies, DHL and GLS:

```
{if $$Dispatch.name == "DHL"}
Tracking Code: https://nolp.dhl.de/nextt-online-public/de/search?piececode={$sOrder.trackingcode}
{else if $$Dispatch.name == "GLS"}
Tracking Code: https://gls-group.eu/DE/de/paketverfolgung?match={$sOrder.trackingcode}
{else}
No tracking possible
{/if}
```

If the shipping type "DHL" is chosen, the corresponding DHL link will be used. The GLS link gets applied to "GLS", when using another shipping type the text "No tracking possible" is displayed (conceivable, for example, for standard postal mailing)

How to go live

In most cases you will not be able to go live with your Shopware shop under the desired target domain immediately. For example, if an old shop or website is still being operated under the target domain. In this section, we show you the most important points to bear in mind if you want your shop to go live at a later date.

Adjust manually maintained links

You may have manually maintained links in your test installation that lead to a resource in your shop. If you now change the domain of the shop, you should also adjust the links that lead to this shop.

Links are often used in the following places and should possibly be updated.

Categories:

Categories can be linked directly. Click on the respective category under Articles > Categories and adjust the link in the Link to **external page field**.

You may also have entered a link in the category text. You can find the category text in the **CMS functions** section.

Article:

You can also add your own links in the item description. To do this, click on Article > Overview in the backend, select your article and update any links in the description.

Shopping worlds:

You may also have created your own shopping worlds in which your own links have been stored. If so, you should also update these. You can find the shopping worlds in the backend under Marketing > Shopping

Worlds. It is best to check all the shopping worlds created there.

Remove demo data

If you have installed the demo data extension for test purposes, you should of course remove this demo data before the go-live. However, deactivating or uninstalling the demo data extension does not remove the demo data again. The demo data must therefore be removed manually.

Folgende Daten werden durch die Demodaten Erweiterung hinzugefügt und sollten wieder entfernt werden:

- **Categories:** Food, Clothing, Leisure & Electrical including the subcategories.
- **Manufacturer:** Shopware Fashion, Shopware Food, Shopware Freetime
- **Article:** All articles of the named manufacturers.
- **Shopping world:** Start page, if this is no longer required.
- **Properties:** When no longer needed, the Clothing, Food, Electronics and Fashion sets can be removed as well as the Properties groups item types, size, material, target group and ingredients.

Check shipping methods

Check whether the currently created shipping methods should also be available in the live system. It is possible that temporary shipping types have been created here for testing purposes, which should now be removed.

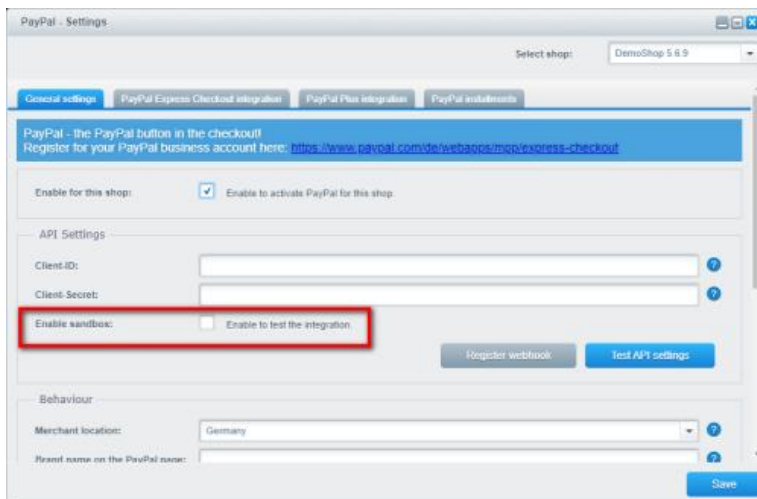
If desired, also check in the frontend whether the shipping types are only available under the conditions under which they should be available.

Information on setting up the shipping costs can be found [here](#).

Deactivate sandbox mode of the payment methods

If you have added more payment methods, you will probably have activated the sandbox mode of these payment methods in the test phase in order not to generate real payments for test orders.

In the case of the latest PayPal version, for example, you can find this option in the backend under **Customer > Payments > PayPal > Settings > Basic settings**. There you can deactivate the sandbox option in the **API settings section**.



Not every payment method uses the same method to deactivate the sandbox mode. You can find out how to deactivate the sandbox mode in the documentation of the respective provider..

Check the order process

Before you go live, you should carry out at least one complete test order. This is the only way to find out where things might not be running smoothly. It is better to carry out several different orders so that as many different cases as possible can be checked.

Are all the desired payment methods displayed? Are the desired shipping methods available? Does the checkout process work without problems? Is the order confirmation sent properly?

Check email templates and mailers

Before going live, it is important to check your email templates and mailer. To do this, place a test order and check whether you receive the order confirmation without any problems.

If no mail is sent, check your [mailer settings](#).

When your mail arrives, check that all the details are in the mail.

- Is the logo shown correctly?
- Are the articles displayed correctly?
- Are all variables such as name, salutation and address of the customer filled correctly?
- Did you give the bank details?

If something still doesn't fit, you'll find all the information you need to make adjustments to the email template [here](#).

Reset number ranges if necessary

During your tests, you will probably have placed orders, created customers and possibly even created invoices. If necessary, you can reset the number ranges to a starting value before the website goes live.

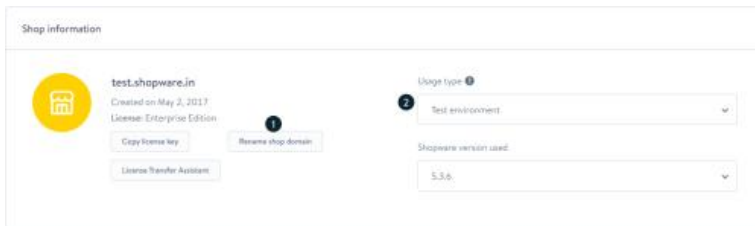
You can do this in the backend under Settings > Basic settings > Shop settings > Number ranges. The respective number range should not be changed after the shop has gone live. On the one hand, the current legal situation requires consecutive invoice numbers, and on the other hand, there may be coordination problems between the programs if an ERP or CRM system are connected.

If you use ERP or CRM systems, it is best to check with the manufacturer before if changing the number ranges is still unproblematic.

Shopware Account licence changes

When creating the Shopware account, it is necessary to enter a licence domain. However, if you are still running an old shop or website on your main domain, you may have temporarily entered a subdomain. For correct licensing, it is necessary to replace this temporary domain with the final licence domain under which the shop will be accessible in the future before the shop goes live.

In your **Shopware account** at account.shopware.com, click on Shops and then on your corresponding shop domain that you want to change..



In the Shop Information area you will now find the **Rename shop domain button (1)** to rename your domain.

Note that you can only rename the shop domain once. You need to be sure that the new domain is your final domain.

It is then necessary to change the **usage type (2)** of the domain to productive environment.

Adapt domain at the hoster

Once you have checked all points and your shop is functioning, the last step is to route the domain of the hosting service to the main directory of your Shopware shop, so that your shop can now be reached over the final domain.

You can find out exactly how to adjust the routing of your domain from your hosting service. As a rule, this can be configured yourself using a corresponding admin interface.