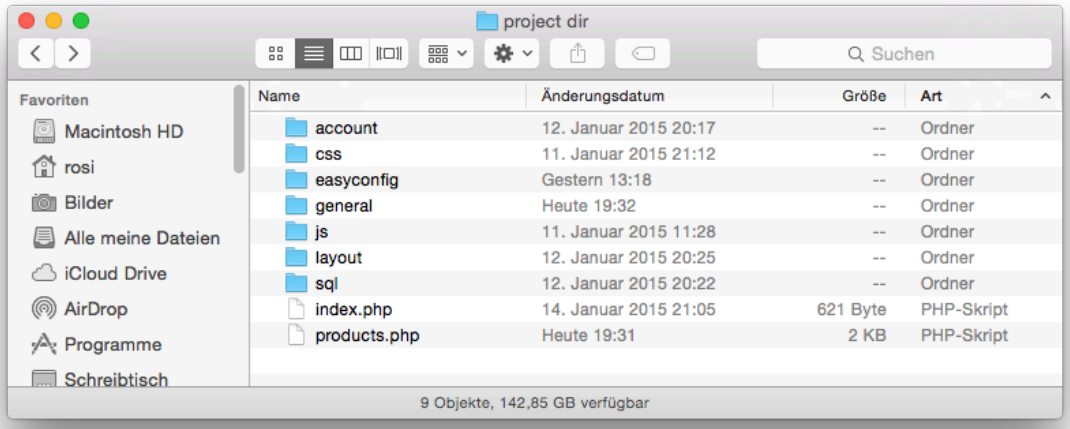
EasyConfig Documentation

Version: 1.1

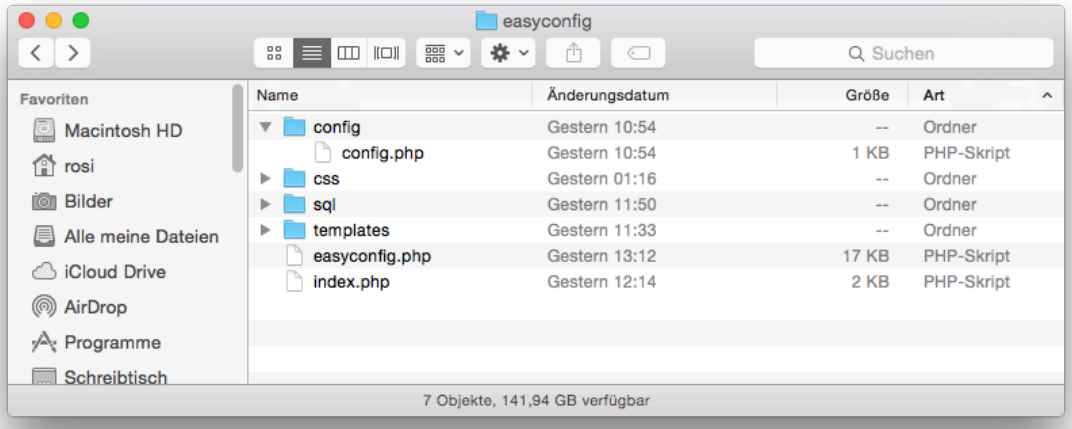
1 Table of contents

2 How to install EasyConfig

To install EasyConfig just extract the zip archive to your applications root directory. I personally would leave EasyConfig in its own directory and not drag the EasyConfig files in your project directory. But its your choice, in this documentation I'd like to think that all EasyConfig files are in the EasyConfig directory. So after deployment it possibly looks like this:

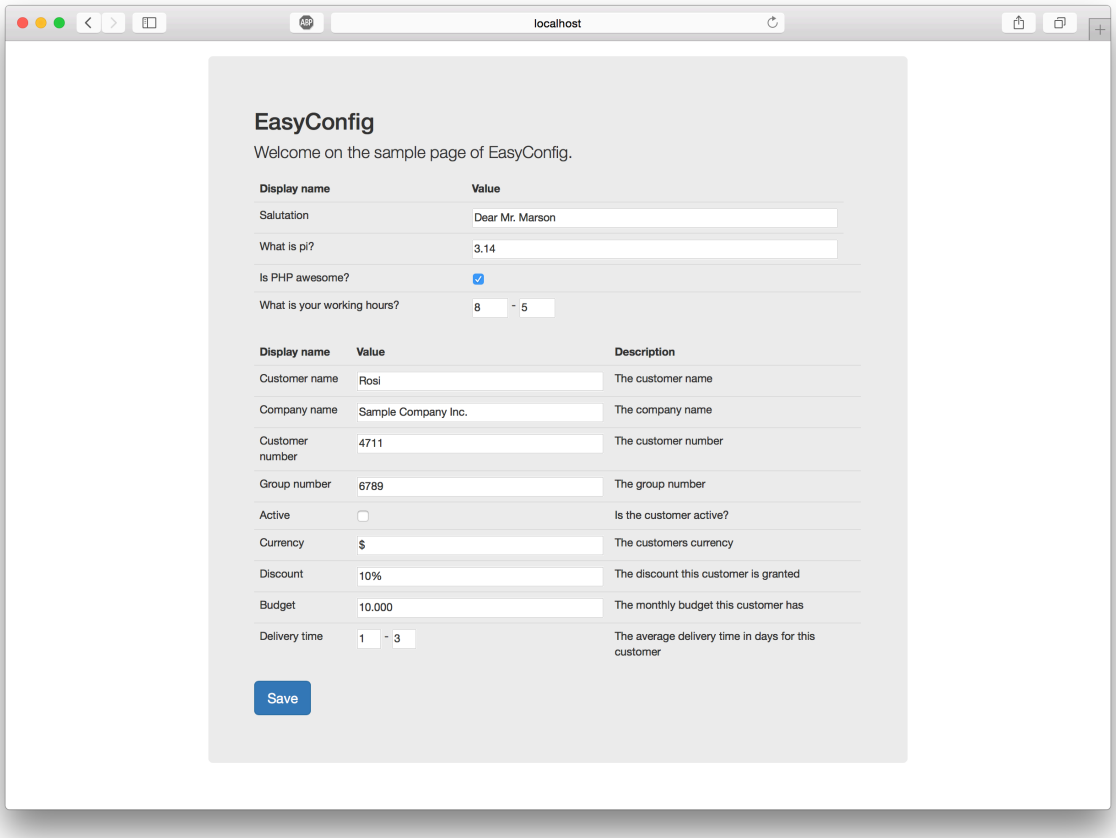


There is your project directory and EasyConfig is in it.
Now configure EasyConfig to use your MySQL Server for the configuration entries.
Open the config.php in the config directory in the EasyConfig root directory:



Now enter the appropriate values and create the configurations table (or let EasyConfig do it for you, see chapter **How to use EasyConfig** for more details) with the sql script in the sql

directory. This script also creates a few example entries, which you can directly see by opening the index.php in your browser, it should look like this:



EasyConfig is now installed.

3 How to use EasyConfig

(Information: All examples and code in this documentation is also included in the “index.php” in your EasyConfig root directory)

If you want to use EasyConfig in your projects, add the following to the top of your php script:

```
require_once('easyconfig/easyconfig.php');
$easyConfig = new EasyConfig();
```

3.1 Optional: Let EasyConfig create the table structure

If you are not very familiar with the database and sql, let EasyConfig create the configurations table. Just call the Install function with true or false as parameter (true will create the examples, false won't):

```
$easyConfig = new EasyConfig();
$easyConfig->Install();
```

Install will create the table "Configurations" if you have not changed it in config.php. After the table is successfully created you may remove this line, its not needed anymore.

3.2 Creating input-fields, labels and description manually

Now chose a place in your HTML where you want to add an input-field and add this code:

```
<?php $easyConfig->GetInputField('TEXT_TEST'); ?>
```

This will generate the HTML code for an input-field for the key "TEXT_TEST".
Make sure that the keys exist in your database before you add input-fields for them.
Now a label for that input-field would be nice and possibly a description, so that the user knows what he or she should insert.

Add this code to the place where you want the display name of the key to appear:

```
<?php $easyConfig->EchoDisplayName('TEXT_TEST'); ?>

<?php $easyConfig->EchoDescription('TEXT_TEST'); ?>
```

The first function echoes the display name of "TEXT_TEST" and the second his description string.

3.3 Create a whole table row…

…filled with the input-field, a label and the description. Use this code to automatically create the HTML code for a whole row:

```
<?php $easyConfig->GetRow('TEXT_TEST'); ?>
```

This function will take care of the label, input-field and description. It will produce HTML code similar to this:

```
<tr>
    <td>Salutation</td>
    <td>
        <input type="text" class="input-text"
               id="TTEXT_TEST" name="TTEXT_TEST"
               value="Dear Mr. Marson">
    </td>
    <td>Enter the salutation for this user (example: "Dear Mr. Marson")</td>
</tr>
```

You may also build the row yourself with the three functions mentioned above.

3.4 Create a whole table ~~row~~

If you have many keys in your configuration table and don't want to create a row for each one, there is also a pretty simpler method.
Use the *GetTableByStartingKey* function to create the full HTML code of a table with header and body:

```
<?php $easyConfig->GetTable("CUSTOMERS_NEW_"); ?>
```

This function uses all keys from the database which start with "CUSTOMERS_NEW_" and creates a table similar to this one:

Display name	Value	Description
Customer name	Rosi	The customer name
Company name	Sample Company Inc.	The company name
Customer number	4711	The customer number
Group number	6789	The group number
Active	☑	Is the customer active?
Currency	$	The customers currency
Discount	10%	The discount this customer is granted
Budget	10.000	The monthly budget this customer has
Delivery time	1 - 3	The average delivery time in days for this customer

There is also the *GetTableByKeys* function, this takes an array of keys that is displayed in the order the keys are supplied.

4 Form submission

EasyConfig also takes care of your form values. If you put the table or any of the created input-fields in a form, you are able to handle the form values, validate them and put the values back in the database if the validation was successful.
Just add this code to the top of your php script where the form is in:

```
if (isset($_POST['submitTable'])) {
    $easyConfig->HandleFormSubmit();
}
```

This code will check if the submit button “submitTable” was clicked and then automatically validate all fields and write its contents to the database. More about the validation and how it works can be seen in the chapter **Datatypes**

5 Datatypes

Currently there are four datatypes in EasyConfig available:

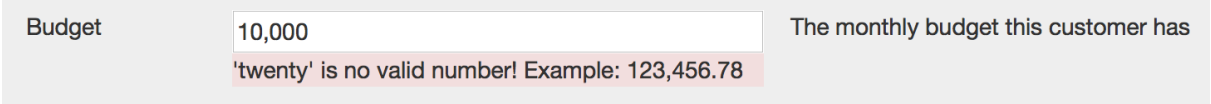
- text
- number (gets validated)
- logical
- interval (gets validated)

5.1 text

Text is the datatype for anything that shouldn’t get validated, its like a string and any data (letters, numbers, special characters) can be inserted here.

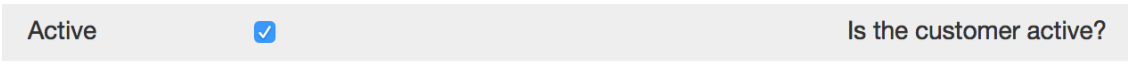
5.2 number

Number is the datatype for numbers only, after submission it gets validated and if the entered string is no number, an error will be shown right under the input-field and the old value is restored:



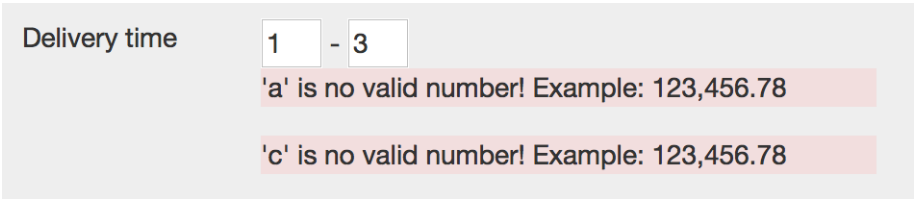
5.3 logical

Logical is really simple, this datatype understands only the values true or false (1 or 0). And therefore the input-field is displayed as checkbox. Needless to say that this field is not validated.



5.4 interval

Interval is the datatype for intervals. It only supports numbers too and it gets validated, like the number datatype.



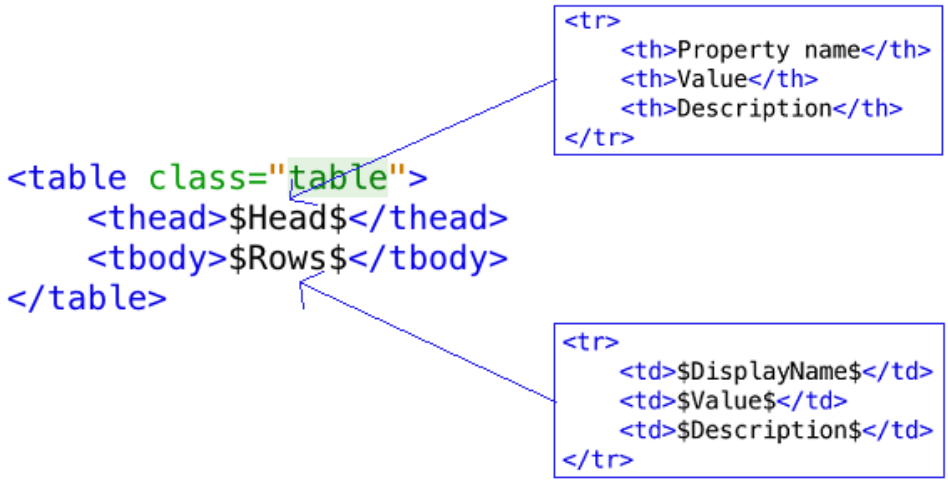
6 Manually get and set values

Of course you can get/set the configuration values manually to use them in your project. Just take a look at the two functions *GetValue* and *SetValue*.

GetValue will return you the value at a given key and *SetValue* will set a value to a key (with an optional parameter to turn on validation). Please be careful if you are entering values without validation!

7 Template system

To use the template system in tables you need to call the *GetTableByStartingKey* or *GetTableByKeys* function. These two functions have three optional parameters to define a template for the whole table, the table head and the table body (the rows):



In the table template you have the two placeholder *$Head$* and *$Rows$.* In the head placeholder the head template will be inserted and in the rows placeholder the table body. In the rows template you may then define how the display name, value and descriptions is presented.