

# **EWA Page Templates (EWA-Seitenklassen)**

Version 3.0  
12/16/2020 4:55:00 PM

# Table of Contents

Page .....	3
Class Documentation.....	4
Page .....	4
PageTemplate .....	6
Index .....	8

# Namespace Documentation

## Page Namespace Reference

---

### Detailed Description

Templates

#### Author:

Bernhard Kreling, `bernhard.kreling@h-da.de`

Ralf Hahn, `ralf.hahn@h-da.de`

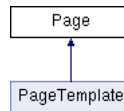
#### Version:

3.0

# Class Documentation

## Page

Inheritance diagram for Page:



### Public Member Functions

- `__destruct ()`

### Protected Member Functions

- `__construct ()`
- `generatePageHeader (string $title="")`
- `generatePageFooter ()`
- `processReceivedData ()`

### Protected Attributes

- `MySQLi $_database`

---

### Detailed Description

This abstract class is a common base class for all HTML-pages to be created. It manages access to the database and provides operations for outputting header and footer of a page. Specific pages have to inherit from that class. Each derived class can use these operations for accessing the database and for creating the generic parts of a HTML-page.

#### Author:

Bernhard Kreling, `bernhard.kreling@h-da.de`  
Ralf Hahn, `ralf.hahn@h-da.de`

---

### Constructor & Destructor Documentation

#### `__construct ()` [protected]

Connects to DB and stores the connection in member `$_database`. Needs name of DB, user, password.

#### `__destruct ()`

Closes the DB connection and cleans up

---

### Member Function Documentation

#### `generatePageFooter ()` [protected]

Outputs the end of the HTML-file i.e. `</body>` etc.

#### Returns:

void

**generatePageHeader (string \$title = "") [protected]**

Generates the header section of the page. i.e. starting from the content type up to the body-tag. Takes care that all strings passed from outside are converted to safe HTML by htmlspecialchars.

**Parameters:**

<i>\$title</i>	\$title is the text to be used as title of the page
----------------	---

**Returns:**

void

**processReceivedData () [protected]**

Processes the data that comes in via GET or POST. If every derived page is supposed to do something common with submitted data do it here. E.g. checking the settings of PHP that influence passing the parameters (e.g. magic\_quotes).

**Returns:**

void

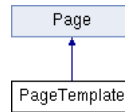
---

**Member Data Documentation****MySqlLi \$\_database [protected]**

Reference to the MySqlLi-Database that can be used by all operations of the class or inherited classes.

## PageTemplate

Inheritance diagram for PageTemplate:



### Public Member Functions

- `__destruct ()`

### Static Public Member Functions

- `static main ()`

### Protected Member Functions

- `__construct ()`
- `getViewData ()`
- `generateView ()`
- `processReceivedData ()`

### Additional Inherited Members

---

### Detailed Description

This is a template for top level classes, which represent a complete web page and which are called directly by the user. Usually there will only be a single instance of such a class. The name of the template is supposed to be replaced by the name of the specific HTML page e.g. baker. The order of methods might correspond to the order of thinking during implementation.

#### Author:

Bernhard Kreling, `bernhard.kreling@h-da.de`  
Ralf Hahn, `ralf.hahn@h-da.de`

---

### Constructor & Destructor Documentation

#### `__construct ()` [`protected`]

Instantiates members (to be defined above). Calls the constructor of the parent i.e. page class. So, the database connection is established.

#### Exceptions:

<i>Exception</i>	
------------------	--

#### `__destruct ()`

Cleans up whatever is needed. Calls the destructor of the parent i.e. page class. So, the database connection is closed.

---

## Member Function Documentation

### **generateView ()** [protected]

First the required data is fetched and then the HTML is assembled for output. i.e. the header is generated, the content of the page ("view") is inserted and -if available- the content of all views contained is generated. Finally, the footer is added.

**Returns:**

void

### **getViewData ()** [protected]

Fetch all data that is necessary for later output. Data is returned in an array e.g. as associative array.

**Returns:**

array An array containing the requested data. This may be a normal array, an empty array or an associative array.

### **static main ()** [static]

This main-function has the only purpose to create an instance of the class and to get all the things going. I.e. the operations of the class are called to produce the output of the HTML-file. The name "main" is no keyword for php. It is just used to indicate that function as the central starting point. To make it simpler this is a static function. That is you can simply call it without first creating an instance of the class.

**Returns:**

void

### **processReceivedData ()** [protected]

Processes the data that comes via GET or POST. If this page is supposed to do something with submitted data do it here.

**Returns:**

void

# Index

`$_database`  
    Page 5  
`__construct`  
    Page 4  
    PageTemplate 6  
`__destruct`  
    Page 4  
    PageTemplate 6  
`generatePageFooter`  
    Page 4  
`generatePageHeader`  
    Page 5  
`generateView`  
    PageTemplate 7  
`getViewData`  
    PageTemplate 7  
`main`  
    PageTemplate 7

Page 3, 4  
    `$_database` 5  
    `__construct` 4  
    `__destruct` 4  
    `generatePageFooter` 4  
    `generatePageHeader` 5  
    `processReceivedData` 5  
PageTemplate 6  
    `__construct` 6  
    `__destruct` 6  
    `generateView` 7  
    `getViewData` 7  
    `main` 7  
    `processReceivedData` 7  
`processReceivedData`  
    Page 5  
    PageTemplate 7