**Main structure**

*url:* [*http://localhost:8080/stoli/bottles*](http://localhost:8080/stoli/bottles)

**bottleController**

• view = model

• render(bottles/index)

*loadModel*

*\_construct :: parent*

**Main model**

• Database + PDO

**Main Controller**

• FB auth();

•new View();

**Bottles\_Model**

• mysql queries

**Bootstrap**

• explode url

• select controller

**Index.php**

• Require libs

**.htaccess**

*\_construct :: parent*

*render(bottles/index)*

*New View()*

**Templates**

• templates/nav.php

**Main View**

• run render()

• run renderTemplate

*Require header.php*

**footer.php**

• renderTemplate(modal)

**bottles/index.php**

• $this->bottles

**Header.php**

• renderTemplate(nav)

• get(‘users’);

**Controller BottleController**

$this -> view -> bottles = $this -> model -> getBottles();

**View Bottles/index.php**

$this->bottles

**$this->view->Render(bottles/index);**

*Wordt éénmalig aangeroepen vanuit bottle\_controller*

**Header.php**

• getTemplate(nav);

• get(‘users’);

**footer.php**

• getTemplate(modal);

**View->Render()**

• require header.php;

• require bottles/index

• require footer.php

**Bottles/index.php**

• $this -> bottles

• foreach()

**$this->renderTemplate(templates/nav);**

*kan voor meerdere keren worden aangeroepen voor statische html vanuit templates*

**Templates**

• templates/nav.php

**Main View**

• $this -> bottles

• foreach()

**Template**

• vb: header.php;

• renderTemplate()

**$this->get(‘users’);**

*kan voor meerdere keren worden aangeroepen voor dynamische content vanuit templates*

**UserController**

• view = model

• renderTemplate

**Template Header**

• $this->get(‘user’);

**Bottles\_Model**

• mysql queries

**Main View**

• new UserController

• New User\_Model

**Template User**

• $this -> userData

**Template**

$this -> get(‘user’,true); // gets replaced by content of /templates/user.php

**Echo $this->helperClass->action();**

*kan voor meerdere keren worden aangeroepen voor eenvoudige dynamische content vanuit templates*

**Main View**

• New Helper;

**Template**

• $this -> getHelper();

• echo HelperClass

**Helper Class**

• return $data;

**Template**

$this -> getHelper(‘menu’);Echo $this -> menuHelper -> activeClass(‘dashboard’);