

1. **Создание нового проекта**
2. **Анализ нового проекта**
3. **Создание макетов**

# **Создание проекта Zend**

Лекция №2

# 1. Создание нового проекта

- `cd C:\Webservers\home`
- `zf.bat create project newproject`

**В корне проекта находятся следующие файлы**

- *`/application/Bootstrap.php` – файл загрузки проекта*
- *`/application/configs/application.ini` – конфигурационный файл используется файлом загрузки и `Zend_Application`*
- *`/public/index.php` – создает экземпляр класса `Zend_Application` и запускает приложение*

## 2. Основные понятия

- Класс **Bootstraper** определяет, какие ресурсы и компоненты будут инициализированы в проекте. По умолчанию инициализирован **Front Controller**, который использует каталог */application/controllers/* как директорию по умолчанию для поиска контроллеров действий. Данный класс выглядит примерно следующим образом

# Макет класса Bootstrap

```
1. // application/Bootstrap.php
2. class Bootstrap extends Zend_Application_Bootstrap_Bootstrap
3. {
4. }
```

Ключевое  
слово class

Название  
класса

Ключевое слово,  
означающее что  
класс может  
расширяться

Базовый  
(наследуемый)  
класс

- **Configuration (Конфигурация).** Основной файл конфигурации по умолчанию хранится в файле: *application/configs/application.ini* и содержит основные директивы настройки вашей PHP среды (например включение и отключение отчета об ошибках). Помимо этого в данном файле указан путь к классу начальной загрузки (Bootstrap), и путь к контроллерам действий (Action Controllers). При использовании ini конфигурационного файла вы можете использовать константы.

# Конфигурационный файл

```
; application/configs/application.ini  
[production]  
phpSettings.display_startup_errors = 0  
phpSettings.display_errors = 0  
includePaths.library = APPLICATION_PATH "../library"  
bootstrap.path = APPLICATION_PATH "/Bootstrap.php"  
bootstrap.class = "Bootstrap"  
appnamespace = "Application"  
resources.frontController.controllerDirectory = APPLICATION_PATH "/controllers"  
resources.frontController.params.displayExceptions = 0  
[staging : production]  
[testing : production]  
phpSettings.display_startup_errors = 1  
phpSettings.display_errors = 1  
[development : production]  
phpSettings.display_startup_errors = 1  
phpSettings.display_errors = 1
```

Комментари  
и

Название  
секции

Переменные  
и их значения

Переменные  
PHP

Секции,  
наследующие  
секцию production

- **Контроллеры действий** являются основной частью приложения и отвечают за обработку запросов, выбор данных из моделей (models) и передачу их в представления (views).

Контроллер действия должен иметь один или несколько методов заканчивающихся словом «**Action**», эти методы будут запускаться при запросе какой либо страницы приложения.

- По умолчанию URL в Zend Framework следуют следующей схеме: `"/controller/action"` где **controller** — текущий контроллер, а **action** — текущее действие.
- Например: *mysite/simple/test*. Здесь Контроллер — **simple** а действие **test**.
- Обычно приложение Zend Framework должно иметь два контроллера действия это **IndexController** — контроллер который является контроллером по умолчанию или домашней страницей вашего сайта и **ErrorController** предназначенный для обработки ошибок таких как HTTP 404 (Страница не найдена) или HTTP 500 (Ошибка приложения).

# IndexController

```
1.  
2. // application/controllers/IndexController.php  
3. class IndexController extends Zend_Controller_Action  
4. {  
5.     public function init()  
6.     {  
7.         /* Initialize action controller here */  
8.     }  
9.     public function indexAction()  
10.    {  
11.        // action body  
12.    }  
13. }
```

# ErrorController

```
1. // application/controllers/ErrorController.php
2. class ErrorController extends Zend_Controller_Action
3. {
4.     public function errorAction()
5.     {
6.         $errors = $this->getParam('error_handler');
7.         switch ($errors->type) {
8.             case Zend_Controller_Plugin_ErrorHandler::EXCEPTION_NO_ROUTE:
9.             case Zend_Controller_Plugin_ErrorHandler::EXCEPTION_NO_CONTROLLER:
10.            case Zend_Controller_Plugin_ErrorHandler::EXCEPTION_NO_ACTION:
11.                // 404 error -- controller or action not found
12.                $this->getResponse()->setHttpResponseCode(404);
13.                $this->view->message = 'Page not found';
14.                break;
15.            default:
16.                // application error
17.                $this->getResponse()->setHttpResponseCode(500);
18.                $this->view->message = 'Application error';
19.                break;
20.        }
21.        $this->view->exception = $errors->exception;
22.        $this->view->request = $errors->request;
23.    }
24. }
```

- **Сценарии вида (views)** в Zend Framework по умолчанию используют простой php синтаксис. Сценарии вида находятся в каталоге *application/views/scripts/*, где они распределены по каталогам в зависимости от контроллера действия.
- В стандартном проекте существуют контроллеры **IndexController** и **ErrorController**, соответственно сценарии вида для них находятся в подкаталогах *index/* и *error/*. В данных подкаталогах содержатся файлы типа **\*.phtml**, которые являются сценариями вида для действий. Так по умолчанию мы имеем сценарии вида: *index/index.phtml* и *error/error.phtml*. Для **Index** контроллера и **Error** Контроллера соответственно.
- Сценарии вида могут содержать какой угодно текст. Кроме того вы можете использовать php теги **<?php ?>** для вставки в них PHP директив.

# Сценарий вида для действия index контроллера indexController

```
1. <!-- application/views/scripts/index/index.phtml -->
2. <style>
22.     div#more-information
23.     {
24.         background-image: url(http://framework.zend.com/images/bkg_body-bottom.gif);
25.         height: 100%;
26.     }
27. </style>
28. <div id="welcome">
29.     <h1>Welcome to the <span id="zf-name">Zend Framework!</span><h1 />
30.     <h3>This is your project's main page<h3 />
31.     <div id="more-information">
32.         <p>
33.             
34.         </p>
35.         <p>
36.             Helpful Links: <br />
37.             <a href="http://framework.zend.com/">Zend Framework Website</a> |
38.             <a href="http://framework.zend.com/manual/en/">Zend Framework
39.                 Manual</a>
40.         </p>
41.     </div>
```

# 3. Создание макета (Layout)

- *Zend\_Layout* использует идеи шаблонов проектирования и при помощи данного средства, вместо того чтобы включать в каждый сценарий вида весь HTML код сайта, мы можем разбить его на «глобальный шаблон» или же макет, и сценарии вида которые занимаются выводом лишь информации из Action контроллеров.
- Для начала работы с *Zend\_Layout*, сначала мы должны настроить наш загрузчик (bootsarтер) на работу с макетами (layout). Это может быть сделано с помощью CLI приложения *Zend\_Tool*, а именно командны **zf enable layout**. Прежде чем исполнять данную команду необходимо перейти в каталог где находится проект.

- Для использования в приложениие XHTML Doctype необходимо добавить ресурс в загрузчик (bootstrap).
- Простейший путь добавить пользовательский ресурс в загрузчик — просто создавать защищенный метод, начинающийся фразой `_init` внутри класса загрузчика. (Например, для инициализации *doctype*, можно создать `_initDoctype` метод.

```
1. // application/Bootstrap.php
2. class Bootstrap extends Zend_Application_Bootstrap_Bootstrap
3. {
4.     protected function _initDoctype()
5.     {
6.     }
7. }
```

- В конфигурационный файл необходимо добавить массив, в котором будут храниться ресурсы вида, загружаемые Bootstrap'ом

```
; application/configs/application.ini  
; Add to [production] section:  
resources.view[] =
```

- Изменить метод `_initDoctype()`. Загрузить ресурс `View`. После чего извлечь объект вида (`view`), и установить для него необходимый `DocType`.

```
protected function _initDoctype()  
{  
    $this->bootstrap('view');  
    $view = $this->getResource('view');  
    $view->doctype('XHTML1_STRICT');  
}
```

```
1. <!-- application/layouts/scripts/layout.phtml -->
2. <?php echo $this->doctype() ?>
3. <html xmlns="http://www.w3.org/1999/xhtml">
4. <head>
5. <meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
6. <title>Zend Framework Quickstart Application</title>
7. <?php echo $this->headLink()->appendStylesheet('/css/global.css') ?>
8. </head>
9. <body>
10. <div id="header" style="background-color: #EEEEEE; height: 30px;">
11.     <div id="header-logo" style="float: left">
12.         <b>ZF Quickstart Application</b>
13.     </div>
14.     <div id="header-navigation" style="float: right">
15.         <a href="&#60;?php echo $this->url(
16.             array('controller'=>'guestbook'),
17.             'default',
18.             true) ?>">Guestbook</a>
19.     </div>
20. </div>
21. <?php echo $this->layout()->content ?>
22. </body>
23. </html>
```

Изменить  
файл  
layout.phtml