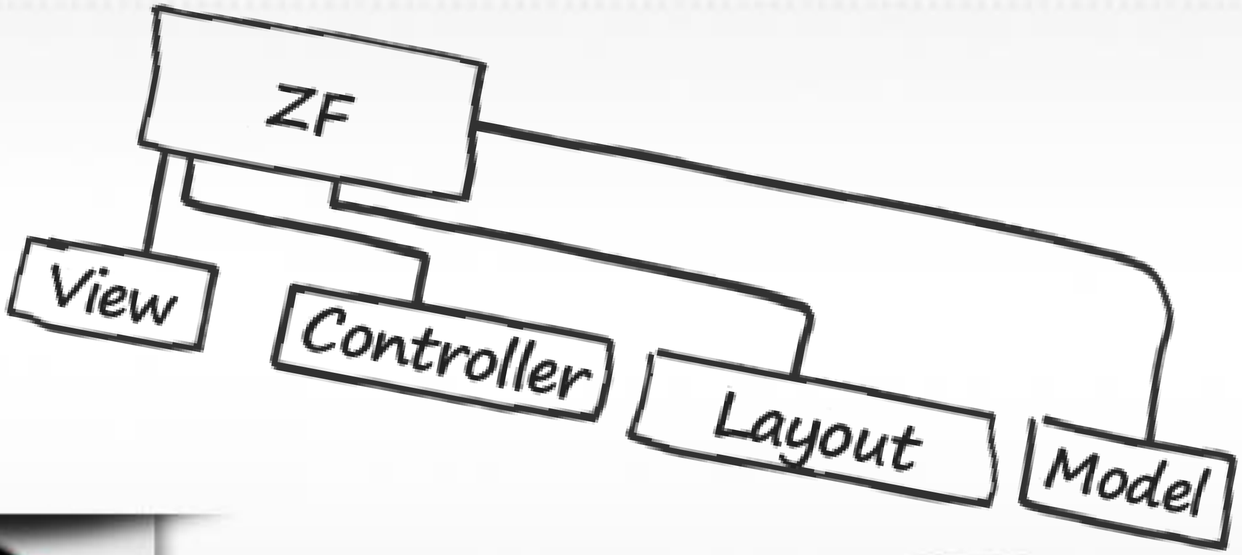
Zend Framework

NOW



OPEN SOURCE
DEPARTMENT

Information
Technology
Institute

Course Materials

You can access the course materials via this link

http://goo.gl/sef2ZN

Day 2 Contents

- Model
- CRUD Operations
- Layout

DB Connection

- Configure Zend framework (DbAdapter) to connect to MySQL:

```
zf configure dbadapter
"adapter=Pdo_Mysql&host=localhost
&username=test&password=test&dbna
me=test“ [-s] development
```

Model

- Models contain the business logic, to create a model simply use the Zend Tool as the following:

```
zf create model User
```

- Models can perform operations in more than one table but for simplicity we will link it to one table, edit your new model under application/models:

```
class Application_Model_User extends
Zend_Db_Table_Abstract
{
    /** Table name */
    protected $_name= 'user' ;
}
```

CRUD

- CRUD Operations are the four basic functions for data storage.

Operation	SQL
Create	INSERT
Read (Retrieve)	SELECT
Update (Modify)	UPDATE
Delete (Destroy)	DELETE

CRUD

- Firstly you need to create four actions for these operations, edit UserController.php add:

```
public function addAction()    {
// action body    }
public function editAction()    {
// action body    }
public function listAction()    {
$user_model = new Application_Model_User();
$this->view->users  = $user_model-
>listUsers();             }
public function deleteAction()    {
// action body    }
```

CRUD

- Add the corresponding function in User.php Model

```
function addUser($user_date){
$row = $this->createRow();
$row->full_name =
$user_date['full_name'];
$row->email = $user_date['email'];
$row->password =
md5($user_date['password']);
return $row->save();              }
function editUser($id, $user_data){
$this->update($user_data, "id=$id");
}
```

CRUD

```
function getUserById($id){
}
function listUsers(){
return $this->fetchAll()-
>toArray();      }
function deleteUser($id){
$this->delete("id=$id");
}
```

CRUD

```
// using function insert
$this->insert($data);
$data Asoc array (col=>value)
//select
$this->select()->from(“table”,$cols)->where->group()->having()->order();
$this->fetchAll($select);
```

Layout

- You may have noticed that the view scripts were HTML fragments- not complete pages. This is by design; we want our actions to return content only related to the action itself, not the application as a whole.
- To render the full design of the application we use layout. To create layout just run:

```
$ zf enable layout
```

Layout

- To initialize the view resource, add the following line to your application/configs/application.ini file, in the section marked production:

```
resources.view[] =
```

- The next step is to initialize the layout placeholders in bootstrap, add this to application/Bootstrap.php

Layout

```
protected function _initPlaceholders()
       {
           $this->bootstrap('View');
           $view = $this-
>getResource('View');
           $view->doctype('XHTML1_STRICT');
           //Meta
           $view->headMeta()-
>appendName('keywords', 'framework, PHP')
                    -
>appendHttpEquiv('Content-Type',
'text/html;charset=utf-8');
```

Layout

```
('Content-Type', 'text/html;charset=utf-8');

           // Set the initial title and separator:
           $view->headTitle('OS Site')
                ->setSeparator(' :: ');

           // Set the initial stylesheet:
           $view->headLink()-
>prependStylesheet('/css/site.css');

           // Set the initial JS to load:
           $view->headScript()-
>prependFile('/js/site.js');
       }
```

Layout

- The last step is to add this to your layout

```
<?php echo $this->doctype() ?>
<html>
    <head>
    <?php echo $this->headMeta(); ?>
    <?php echo $this->headTitle() ?>
    <?php echo $this->headLink() ?>
    <?php echo $this->headScript() ?>
</head>
<body>
```

Layout

- You can add whatever you want to your view script:

```
<?php $this->headTitle()->append('Some
Page'); // place after other segments
?>
<?php $this->headTitle()->prepend('Some
Page'); // place before ?>
<?php $this->headLink()-
>appendStylesheet('/css/user-list.css')
?>
<?php $this->headScript()-
>appendFile('/js/user-list.js') ?>
```

Layout

- You can divide your script into segments an load every segment as the following

```
<?php echo $this-
>partial('_header.phtml'); ?>
```