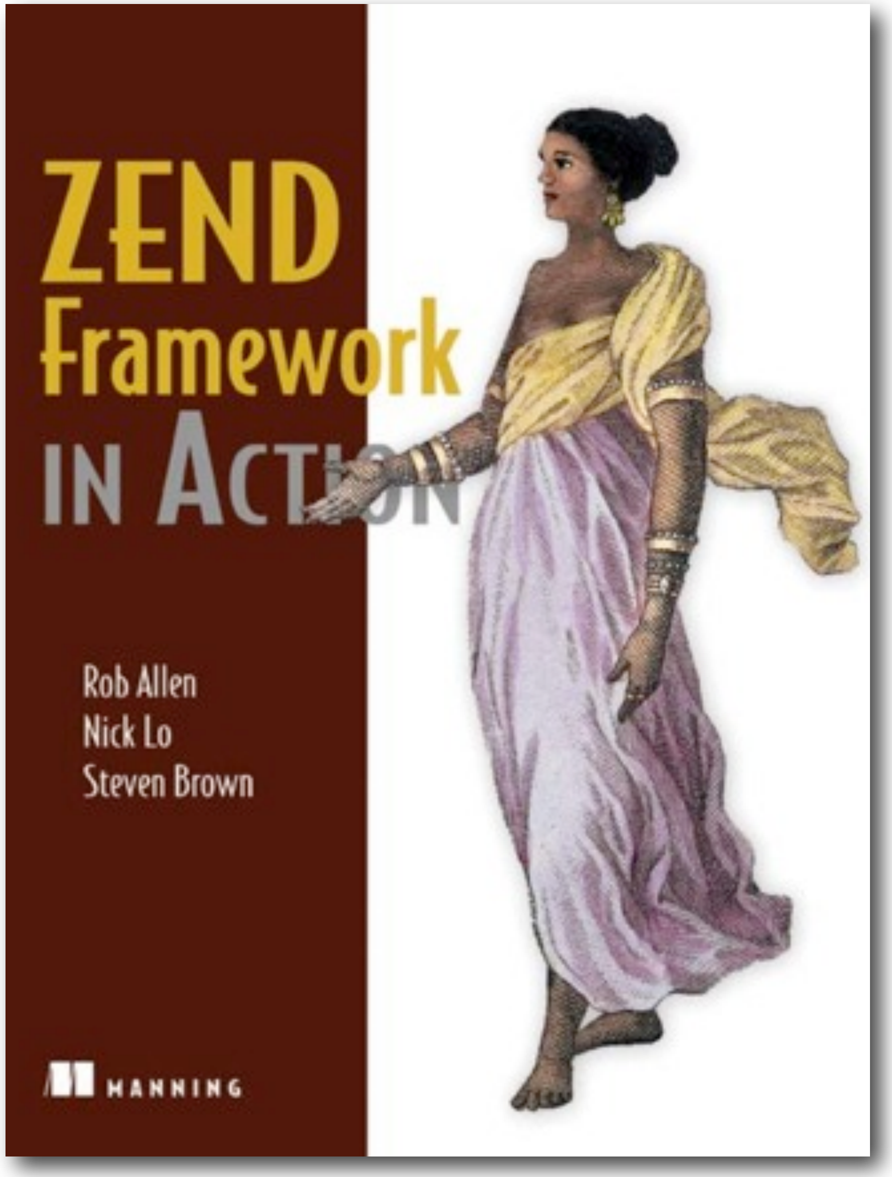
Building a Zend Framework application

Rob Allen

Rob Allen?

- PHP developer since 1999
- Wrote Zend_Config
- Tutorial at akrabat.com
- Book!



What we'll cover

- Overview of Zend Framework
- Getting and installing it
- The basics of a ZF MVC application
- An application: todolist
    - MVC application: DB access, layouts etc
    - Forms, caching, authentication

- Please ask questions as we go!

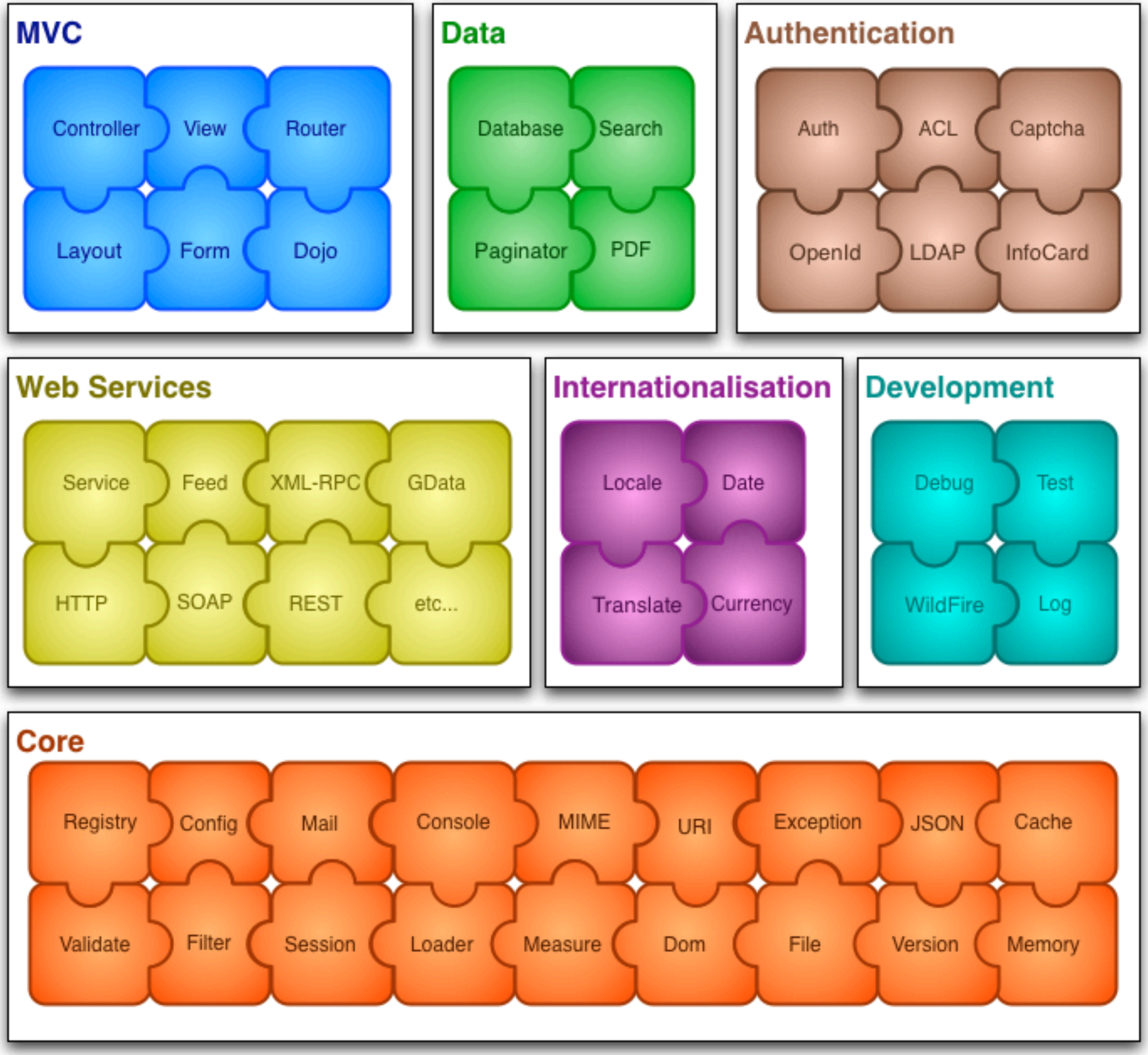
ZF overview

What is ZF?

- PHP5 component library
- MVC Components make a “framework”
- Open source - BSD license
- Documented
- Quality assured
- Certification
- Actively maintained by Zend

ZF Philosophy

- Use-at-will
- Simple usage 80% of the time
- Showcase current trends
    - OO best practice
    - web services
    - dev best practice: unit testing, documentation
    - for 2.0: PHP 5.3 namespaces & closures



MVC

Controller View Router

Layout Form Dojo

Data

Database Search

Paginator PDF

Authentication

Auth ACL Captcha

OpenId LDAP InfoCard

Web Services

Service Feed XML-RPC GData

HTTP SOAP REST etc...

Internationalisation

Locale Date

Translate Currency

Development

Debug Test

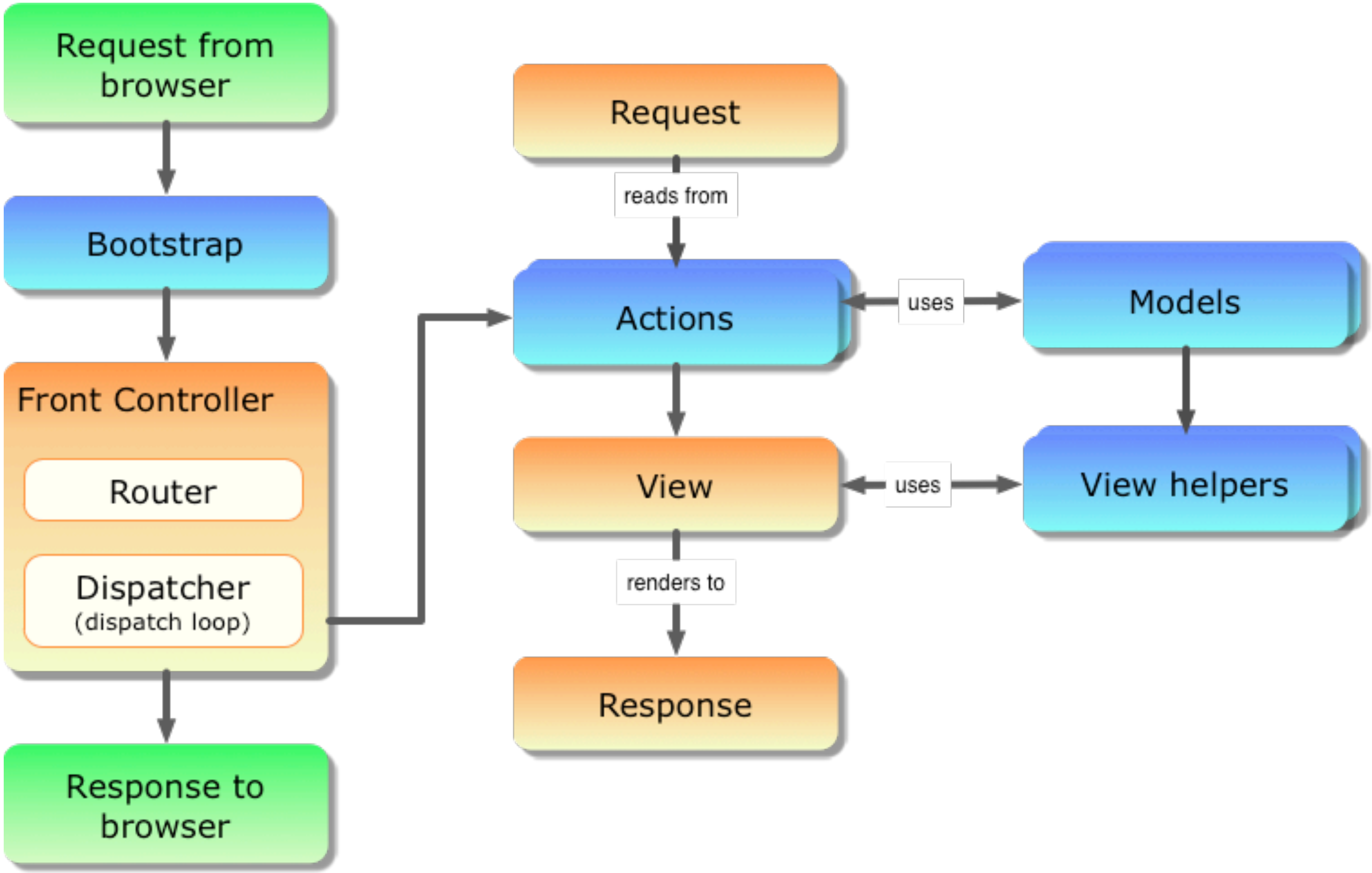
WildFire Log

Core

Registry Config Mail Console MIME URI Exception JSON Cache

Validate Filter Session Loader Measure Dom File Version Memory

The MVC system



Routing: URLs

default:

```
/controller/action/param1/value1/
```

custom:

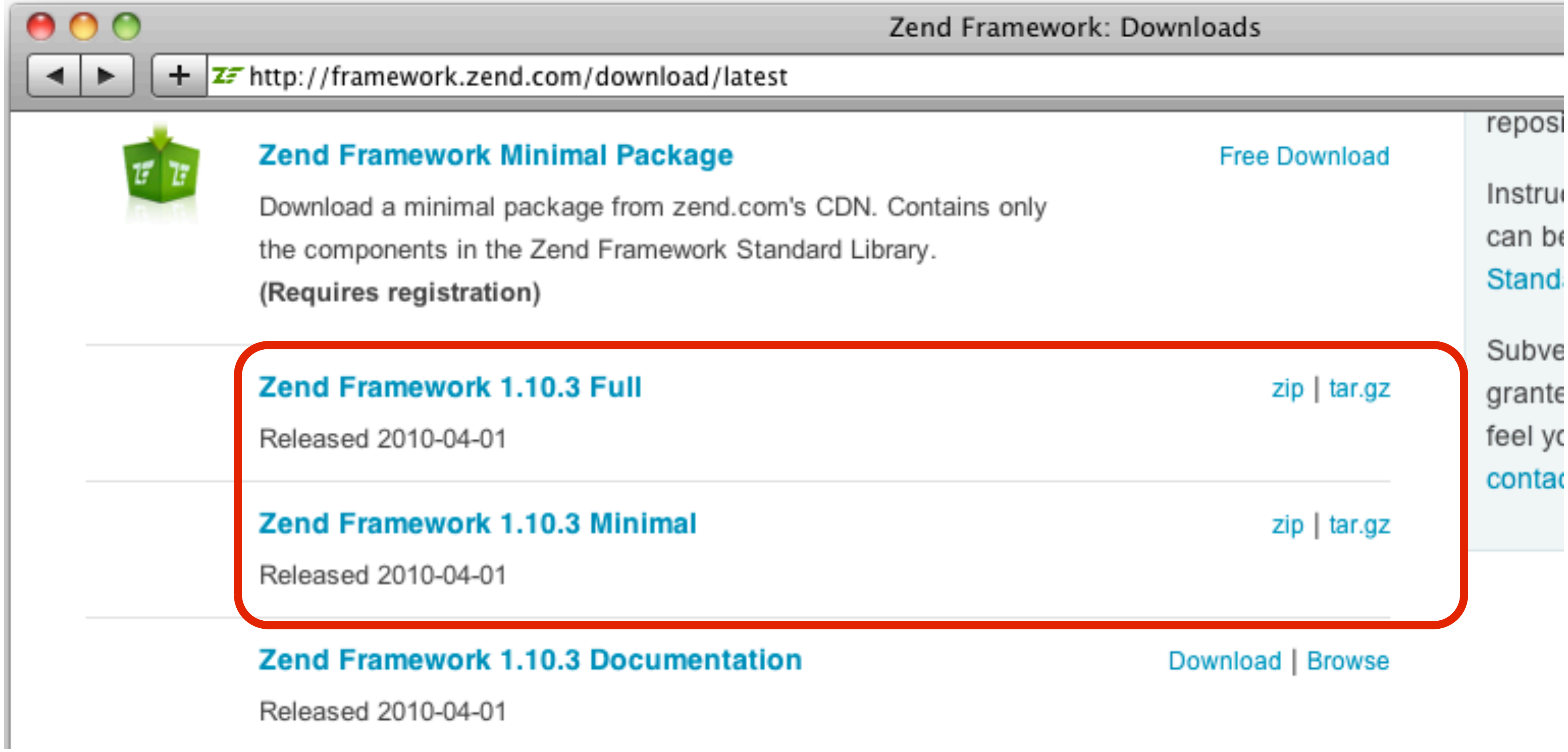
```
/news/2008/11
```

```
$route = new Zend_Controller_Router_Route(
    'news/:year/:month',
    array(
        'controller' => 'news',
        'action'     => 'archive'
    ),
);
```

Getting and installing ZF

Installation

- http://framework.zend.com/download/latest



Zend Framework: Downloads

http://framework.zend.com/download/latest

Zend Framework Minimal Package — Free Download

Download a minimal package from zend.com's CDN. Contains only the components in the Zend Framework Standard Library.

(Requires registration)

Zend Framework 1.10.3 Full — zip | tar.gz

Released 2010-04-01

Zend Framework 1.10.3 Minimal — zip | tar.gz

Released 2010-04-01

Zend Framework 1.10.3 Documentation — Download | Browse

Released 2010-04-01

reposi

Instru
can be
Stand

Subve
grante
feel yo
contac

Install

- Unzip or untar:
  - tar -zxf ZendFramework-1.10.3-minimal.tar.gz
  - unzip ZendFramework-1.10.3-minimal.zip
- Store in a common place
  - /usr/local/include/zendframework
  - c:\zendframework

Add to include_path

```
# Apache: <VirtualHost>, <Directory> or .htaccess:
php_value include_path "..:/usr/local/include/
zendframework/library"

# or PHP:
set_include_path(implode(PATH_SEPARATOR, array(
    '.',
    'c:/zendframework/library',
    get_include_path(),
)));
```

Install zf utility

- `bin/zf.sh` or `bin/zf.bat`
- Place on path (or create alias):
  - `alias zf=/usr/local/include/zendframework/bin/zf.sh`

or use PEAR

```
#pear channel-discover pear.zfcampus.org
Adding channel "pear.zfcampus.org" succeeded
Discovery of channel "pear.zfcampus.org" succeeded

#pear install zfcampus/
downloading ZF-1.10.2.tgz ...
Starting to download ZF-1.10.2.tgz (3,520,994 bytes)
......................................................
.... done : 3,520,994 bytes
install ok: channel://pear.zfcampus.org/ZF-1.10.2

#which zf
/usr/bin/zf
```

A Zend Framework MVC application

A to-do list application

- Create a list of todo items
  - with optional notes and due dates
- Single tasks database table
- Main list shows open tasks
- Logged in user sees only their own tasks

Specification

A todo item consists of:

task	string (required)
notes	string (optional)
due date	datetime (optional)
completed	boolean (required)
identifier	integer (required - primary key)
created_by	integer (for use later)

Create the project

```
zf create project todolist

# if ZF is not on your path:
cd todolist/library
ln -s /usr/local/include/zendframework/library/Zend .
# or
cp -r /usr/local/include/zendframework/library/Zend .
```

Apache vhost

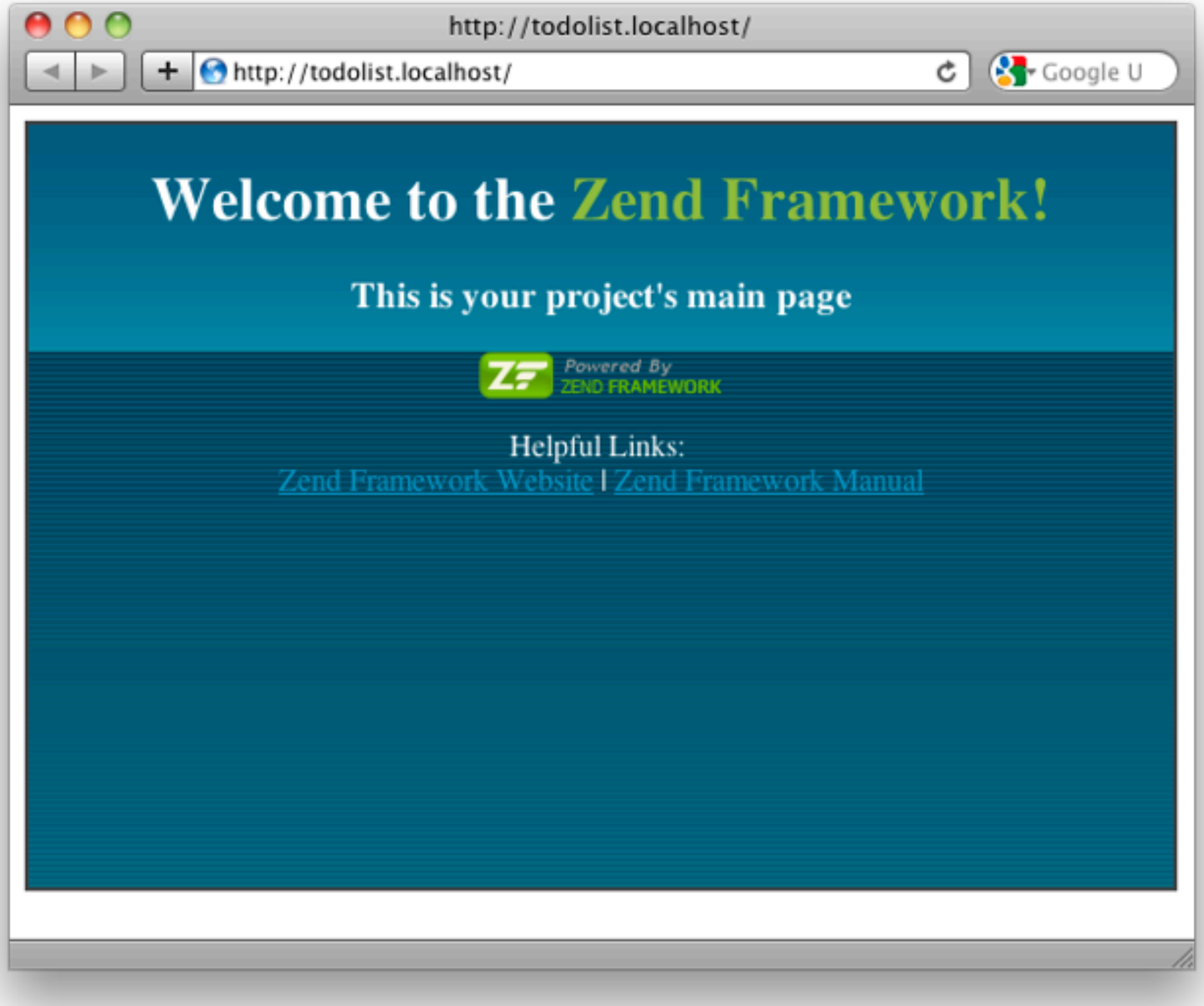
```
vim /etc/apache2/extra/httpd-vhosts.conf
```

```
# From todolist/docs/README.txt
<VirtualHost *:80>
   DocumentRoot "/www/todolist/public"
   ServerName todolist.localhost
   # This should be omitted in the production environment
   SetEnv APPLICATION_ENV development

   <Directory "/www/todolist/public">
       Options Indexes MultiViews FollowSymLinks
       AllowOverride All
       Order allow,deny
       Allow from all
   </Directory>
</VirtualHost>
```

hosts file

```
vim /etc/hosts
or notepad c:\windows\system32/drivers/etc/hosts

127.0.0.1 todolist.localhost
```

Run it!



http://todolist.localhost/
http://todolist.localhost/

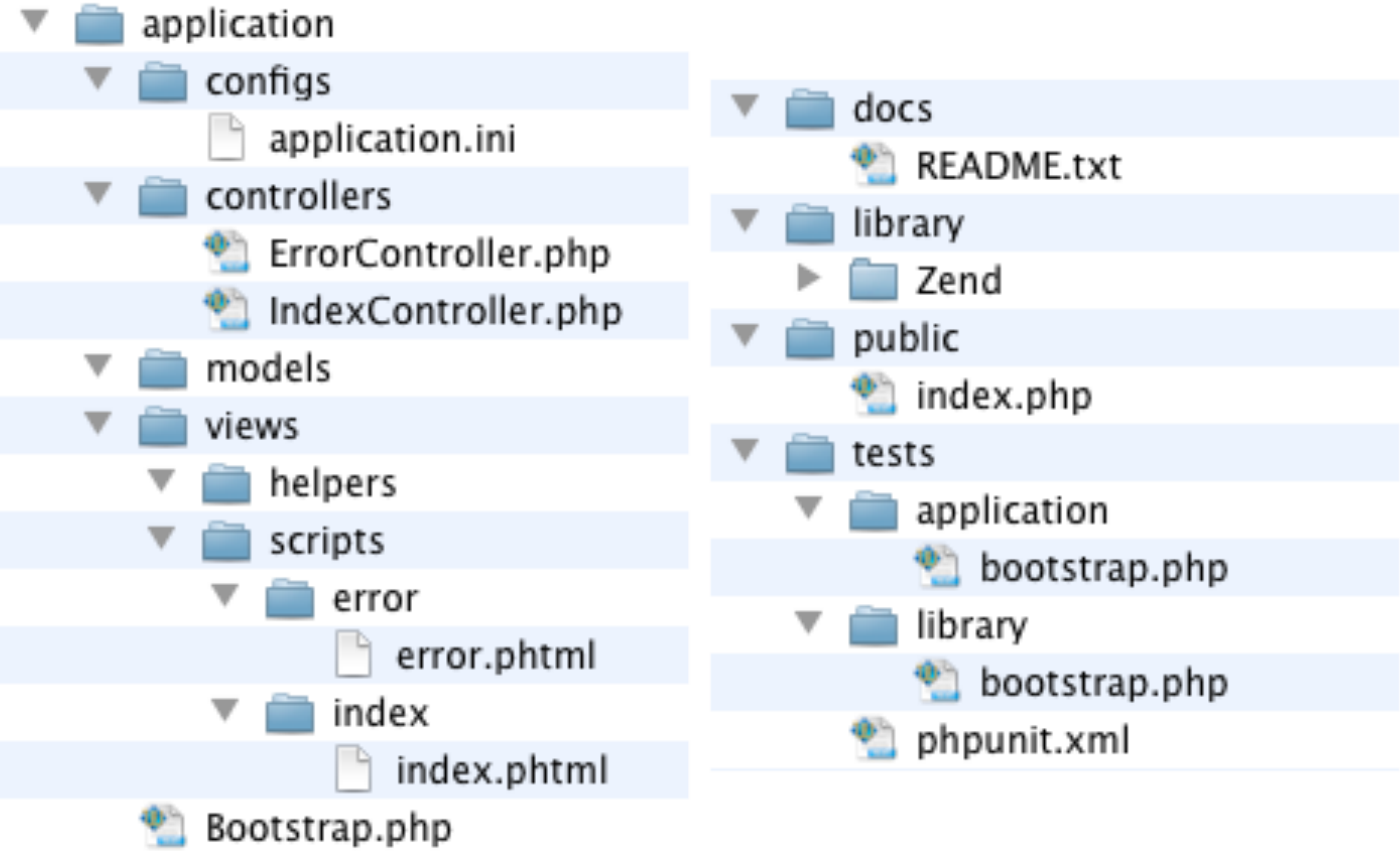
Welcome to the Zend Framework!

This is your project's main page

Powered By ZEND FRAMEWORK

Helpful Links:
Zend Framework Website | Zend Framework Manual

Directories

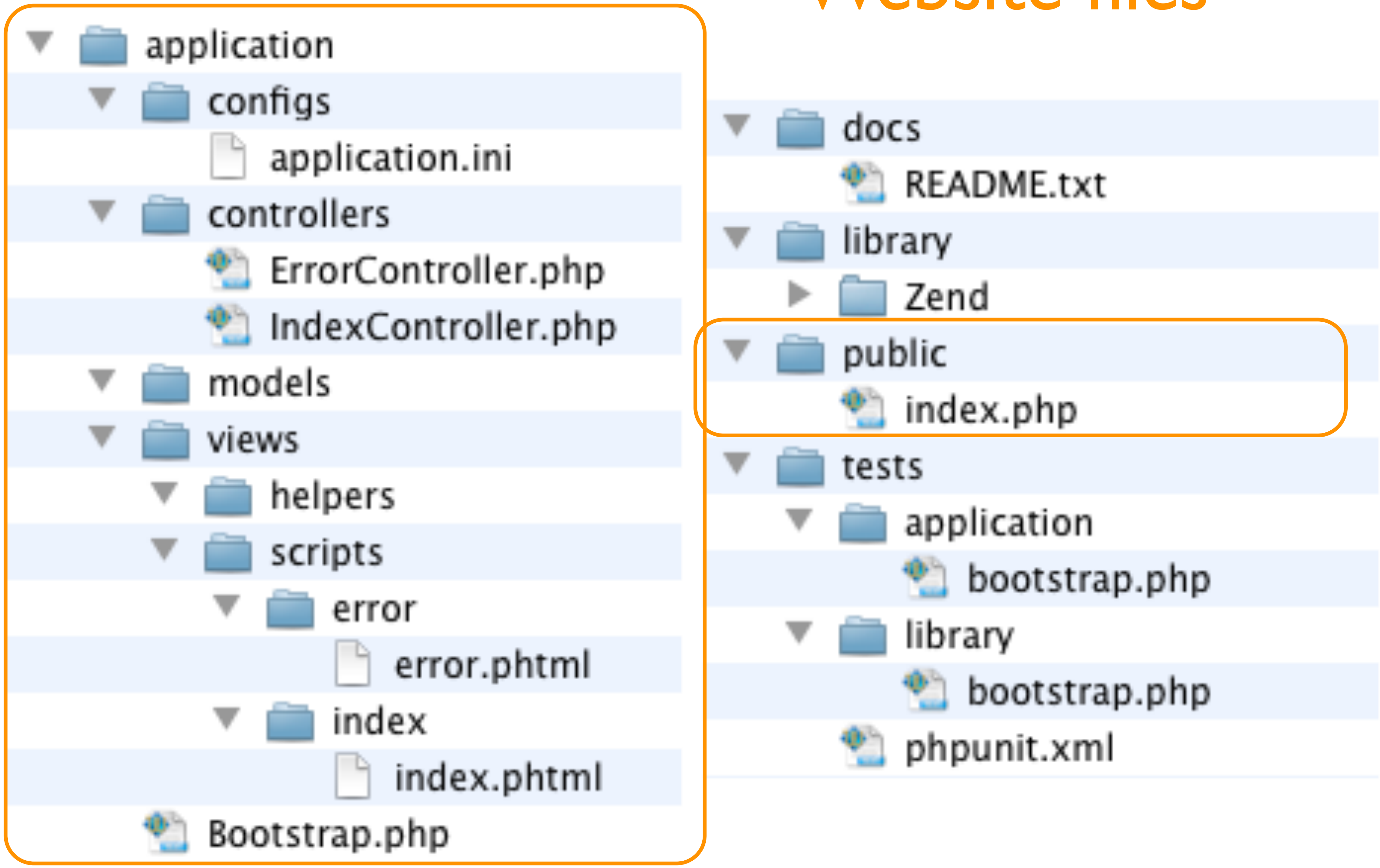


```
application
    configs
        application.ini
    controllers
        ErrorController.php
        IndexController.php
    models
    views
        helpers
        scripts
            error
                error.phtml
            index
                index.phtml
    Bootstrap.php
```

```
docs
    README.txt
library
    Zend
public
    index.php
tests
    application
        bootstrap.php
    library
        bootstrap.php
    phpunit.xml
```

Directories

Website files

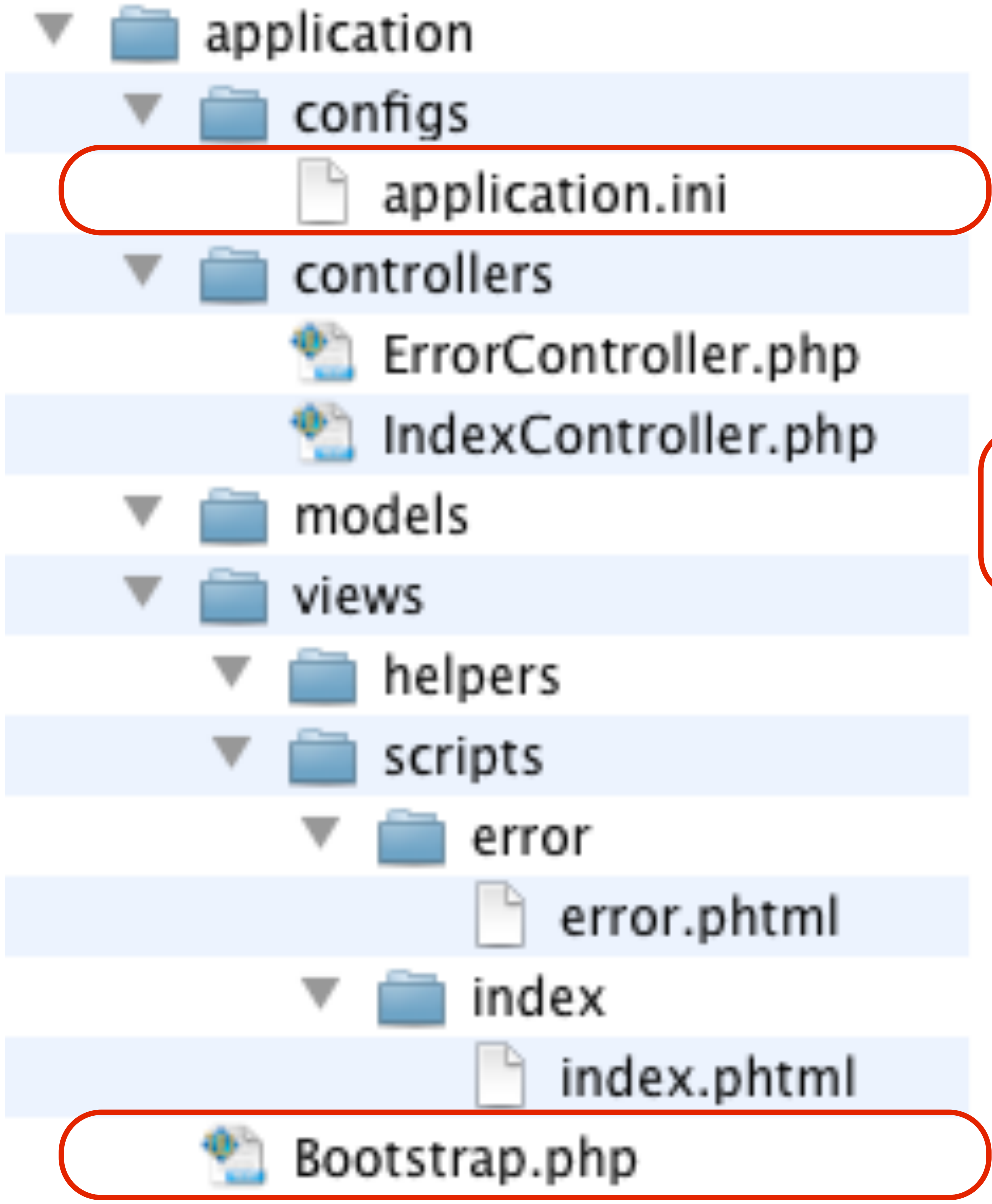


```
▼ application
    ▼ configs
        application.ini
    ▼ controllers
        ErrorController.php
        IndexController.php
    ▼ models
    ▼ views
        ▼ helpers
        ▼ scripts
            ▼ error
                error.phtml
            ▼ index
                index.phtml
    Bootstrap.php
```

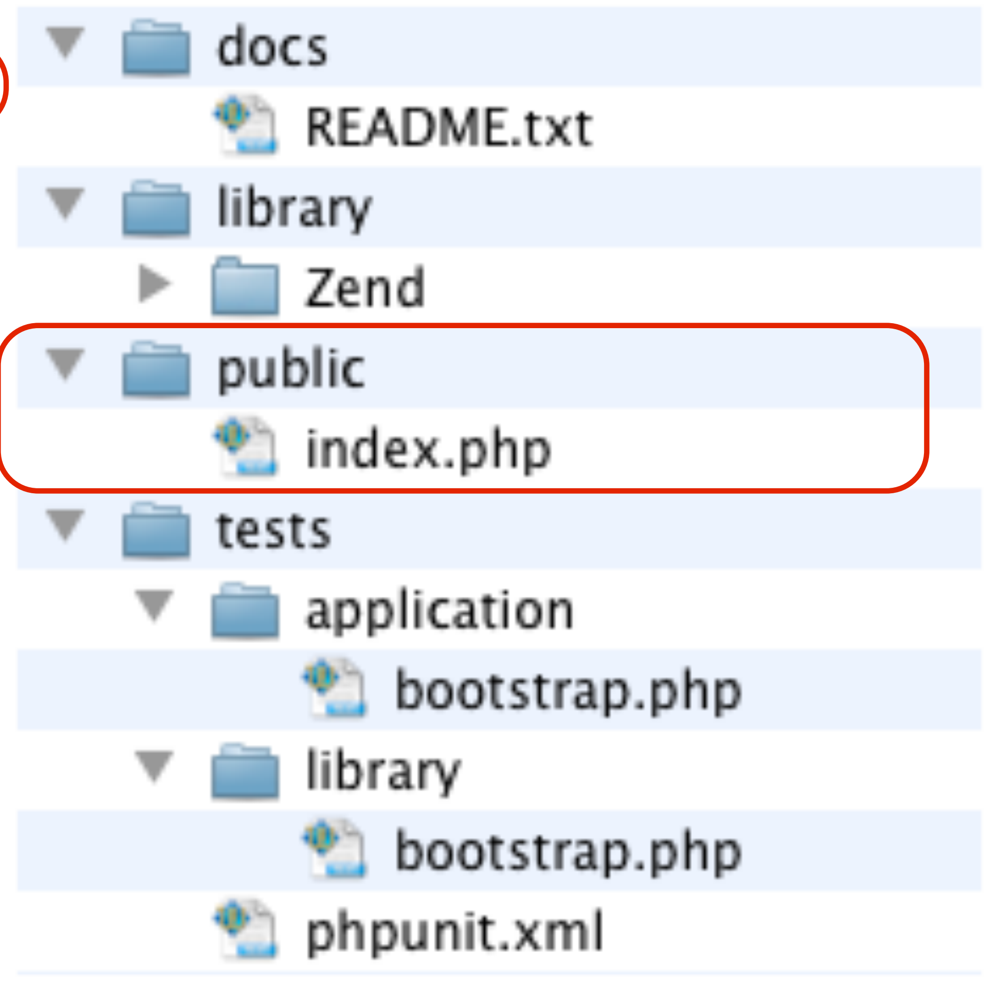
```
▼ docs
    README.txt
▼ library
    ▶ Zend
▼ public
    index.php
▼ tests
    ▼ application
        bootstrap.php
    ▼ library
        bootstrap.php
    phpunit.xml
```

Directories

Bootstrapping

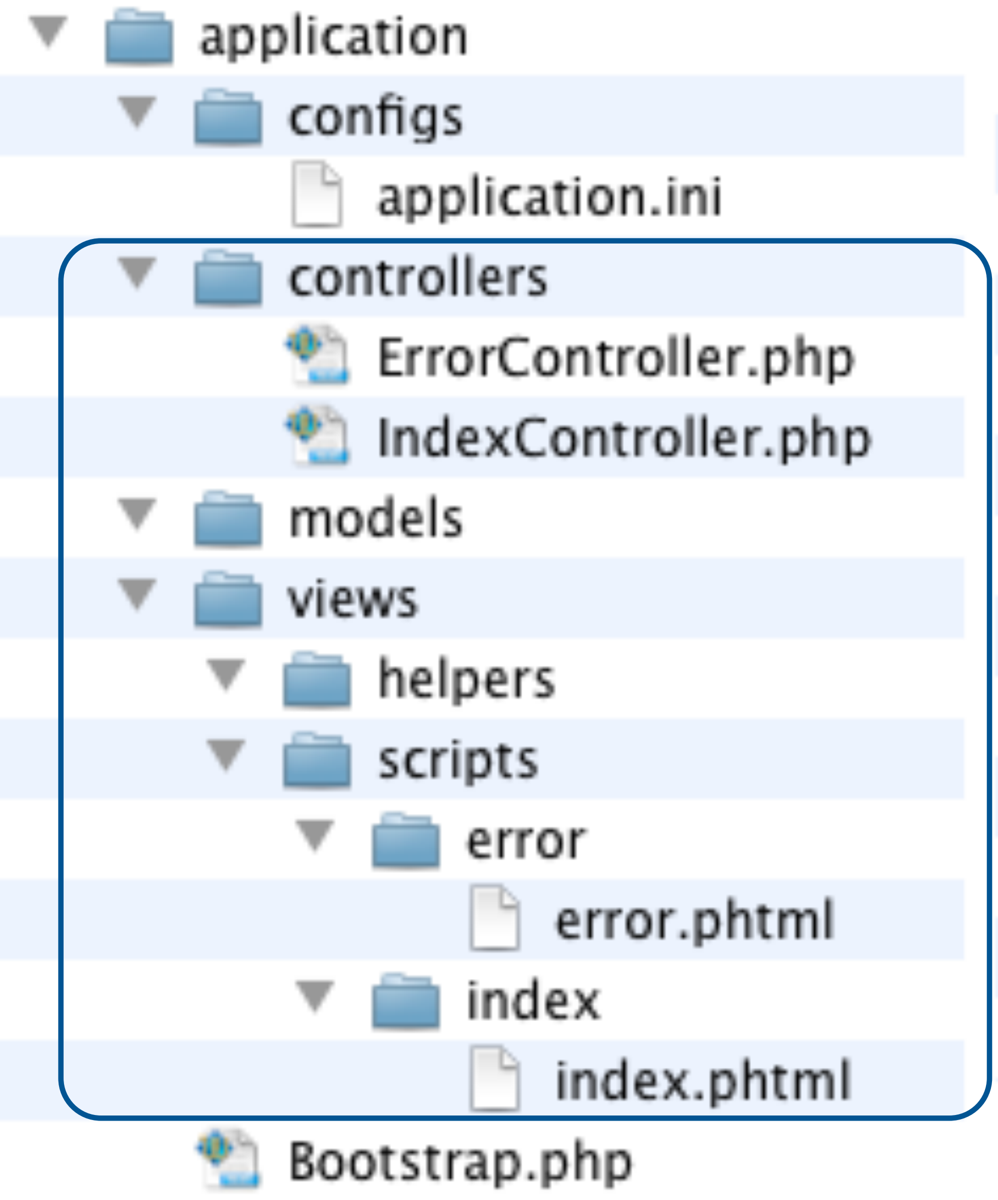


```
application
    configs
        application.ini
    controllers
        ErrorController.php
        IndexController.php
    models
    views
        helpers
        scripts
            error
                error.phtml
            index
                index.phtml
    Bootstrap.php
```



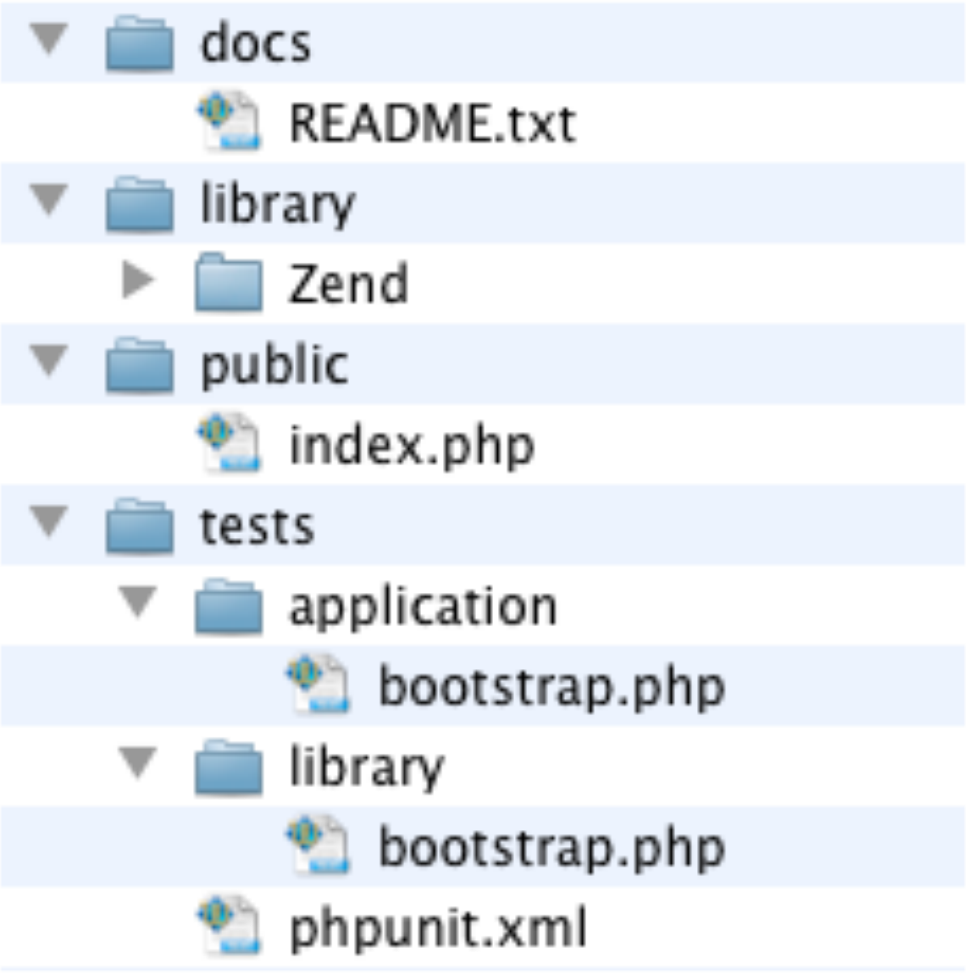
```
docs
    README.txt
library
    Zend
public
    index.php
tests
    application
        bootstrap.php
    library
        bootstrap.php
    phpunit.xml
```

Directories



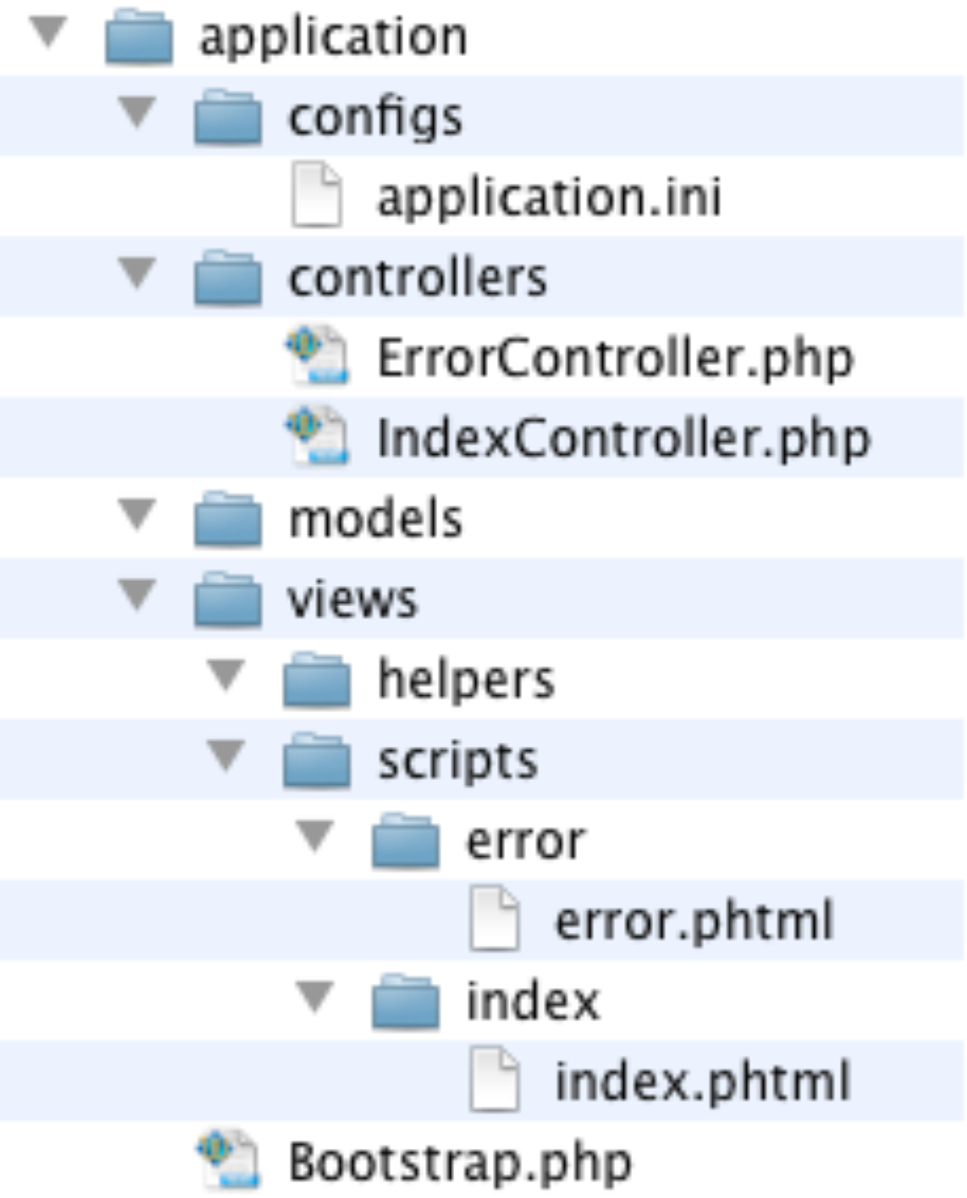
```
▼ application
    ▼ configs
          application.ini
    ▼ controllers
          ErrorController.php
          IndexController.php
    ▼ models
    ▼ views
        ▼ helpers
        ▼ scripts
            ▼ error
                  error.phtml
            ▼ index
                  index.phtml
      Bootstrap.php
```

MVC



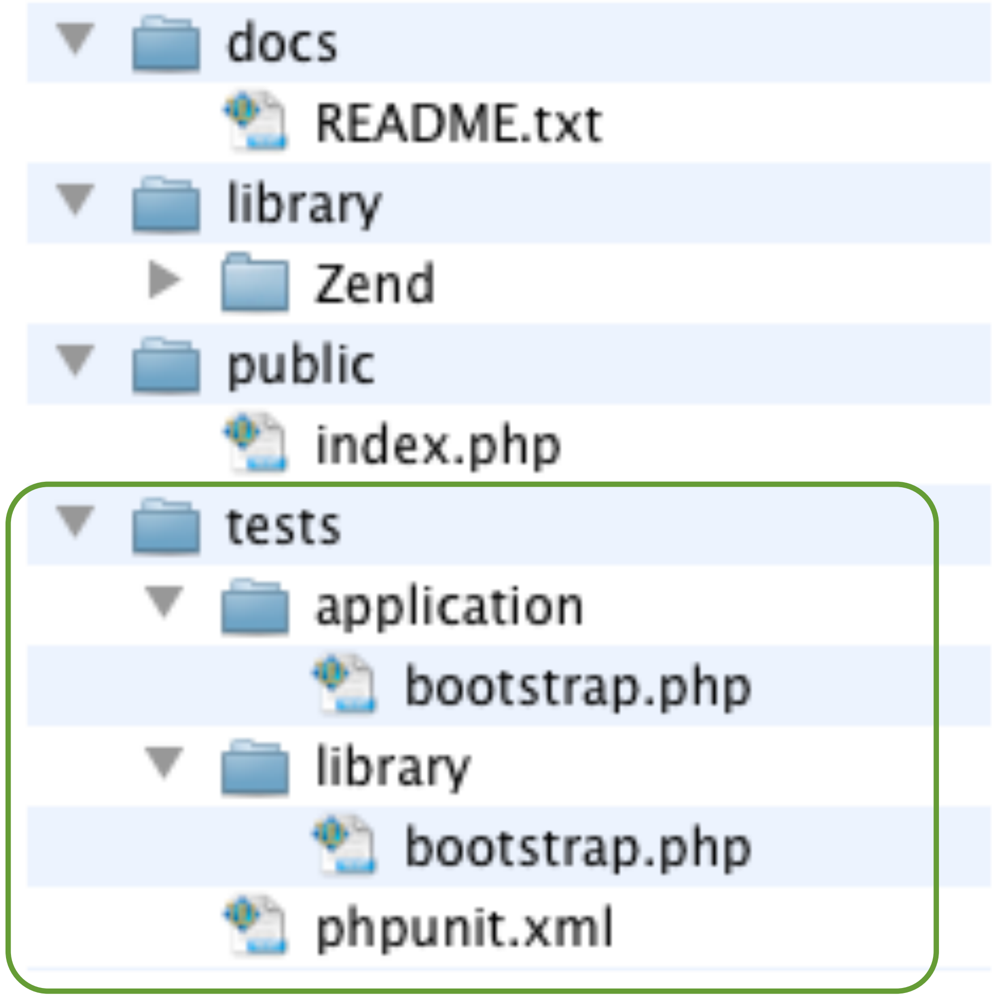
```
▼ docs
      README.txt
▼ library
    ▶ Zend
▼ public
      index.php
▼ tests
    ▼ application
          bootstrap.php
    ▼ library
          bootstrap.php
      phpunit.xml
```

Directories



```
▼ application
    ▼ configs
        application.ini
    ▼ controllers
        ErrorController.php
        IndexController.php
    ▼ models
    ▼ views
        ▼ helpers
        ▼ scripts
            ▼ error
                error.phtml
            ▼ index
                index.phtml
    Bootstrap.php
```

Unit testing



```
▼ docs
    README.txt
▼ library
    ▶ Zend
▼ public
    index.php
▼ tests
    ▼ application
        bootstrap.php
    ▼ library
        bootstrap.php
    phpunit.xml
```

Bootstrapping

public/.htaccess

public/index.php

application/Bootstrap.php

application/configs/application.ini

.htaccess

```
RewriteEngine On
RewriteCond %{REQUEST_FILENAME} -s [OR]
RewriteCond %{REQUEST_FILENAME} -l [OR]
RewriteCond %{REQUEST_FILENAME} -d
RewriteRule ^.*$ - [NC,L]
RewriteRule ^.*$ index.php [NC,L]
```

index.php (environment)

```
<?php
// Define path to application directory
defined('APPLICATION_PATH')
  || define('APPLICATION_PATH',
     realpath(dirname(__FILE__) . '/../application'));

// Define application environment
defined('APPLICATION_ENV')
  || define('APPLICATION_ENV',
    (getenv('APPLICATION_ENV') ? getenv('APPLICATION_ENV') :
'production'));

// Ensure library/ is on include_path
set_include_path(implode(PATH_SEPARATOR, array(
    realpath(APPLICATION_PATH . '/../library'),
    get_include_path())));
```

index.php (application)

```php
/** Zend_Application */
require_once 'Zend/Application.php';

// Create application, bootstrap, and run
$application = new Zend_Application(
    APPLICATION_ENV,
    APPLICATION_PATH . '/configs/application.ini'
);
$application->bootstrap()
            ->run();
```

Bootstrap.php

```
class Bootstrap extends Zend_Application_Bootstrap_Bootstrap
{
}
```

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]

[development : production]
phpSettings.display_startup_errors = 1
phpSettings.display_errors = 1
resources.frontController.params.displayExceptions = 1
```

Action controllers

- Contain your application workflow
- Map requests to models and views.
- Class name: `NewsController`
- Action methods: `listAction()`
- Home: `IndexController::indexAction()`
- `ErrorController::errorAction()` for 404 and 500s

Views

- Controller integration using ViewRenderer
- `$view` property in controller:
- Folder: application/views/scripts/
- Script naming:

```
NewsController::detailAction() => news/detail.phtml
```

- In view scripts:

```
<?= $this->escape($this->item->title); ?>
```

Model

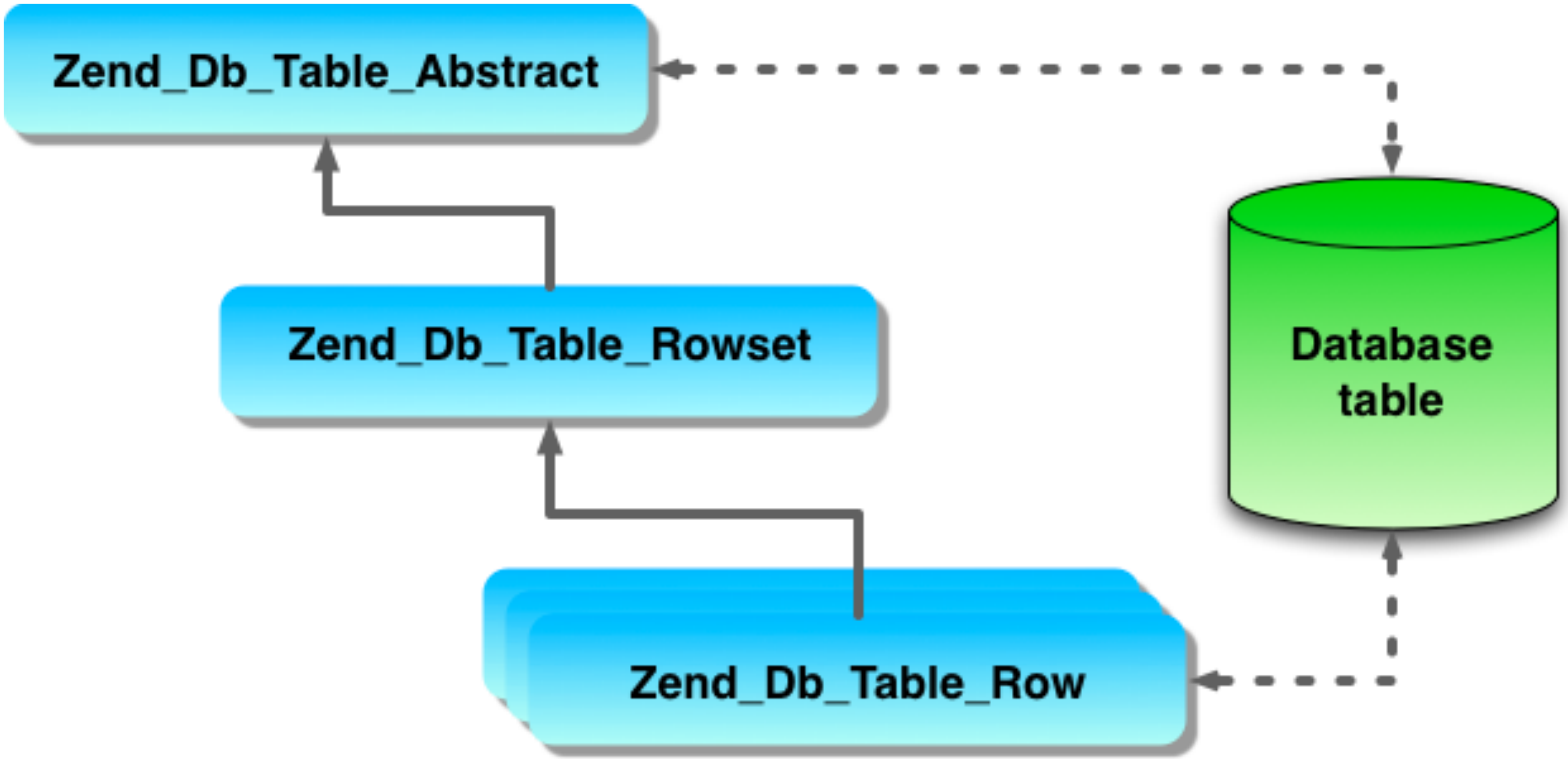
- There is no Zend_Model!
- Database adapters: Zend_Db
- Table row gateway: Zend_Db_Table
- Web services: Zend_Service_Xxx, Zend_Soap
- ORMs: Doctrine, Propel
- NoSQL: CouchDB, MongoDB, Cassandra

Let’s build an app

Database table

```
CREATE TABLE IF NOT EXISTS tasks (
  id int NOT NULL AUTO_INCREMENT,
  title varchar(200) NOT NULL,
  notes text,
  due_date datetime,
  completed tinyint(1) NOT NULL DEFAULT '0',
  date_created datetime NOT NULL,
  PRIMARY KEY (id)
) ENGINE=InnoDB DEFAULT CHARSET=utf8;

INSERT INTO tasks (title, due_date, date_created)
  VALUES ('Task 1', '2010-05-30', NOW());
INSERT INTO tasks (title, due_date, date_created) VALUES
  ('Task 2', '2010-05-010', NOW());
```

Zend_Db_Table model



Configure db access

```
[production]
...
resources.db.adapter = "Pdo_Mysql"
resources.db.params.charset = "utf8"
resources.db.params.host = "localhost"
resources.db.params.username = "todolist"
resources.db.params.password = "password"
resources.db.params.dbname = "todolist"
resources.db.params.profiler.enabled = false

[development : production]
...
```

Zend_Db_Table model

```
$zf create db-table Tasks tasks
Creating a DbTable at /www/todolist/application/models/DbTable/Tasks.php
Updating project profile '/www/todolist/.zfproject.xml'
```

```
class Application_Model_DbTable_Tasks
    extends Zend_Db_Table_Abstract
{
    protected $_name = 'tasks';
    protected $_rowClass = 'Application_Model_DbTable_Task';
}
```

Row object

```
class Application_Model_DbTable_Task
    extends Zend_Db_Table_Row_Abstract
{
    protected function _insert()
    {
        $this->date_created = date('Y-m-d H:i:s');
    }

}
```

Model access method

```php
// Application_Model_DbTable_Tasks

public function fetchOutstanding()
{
    $select = $this->select();
    $select->where('completed = 0');
    $select->order(array('date_created DESC', 'id ASC'));
    return $this->fetchAll($select);
}
```

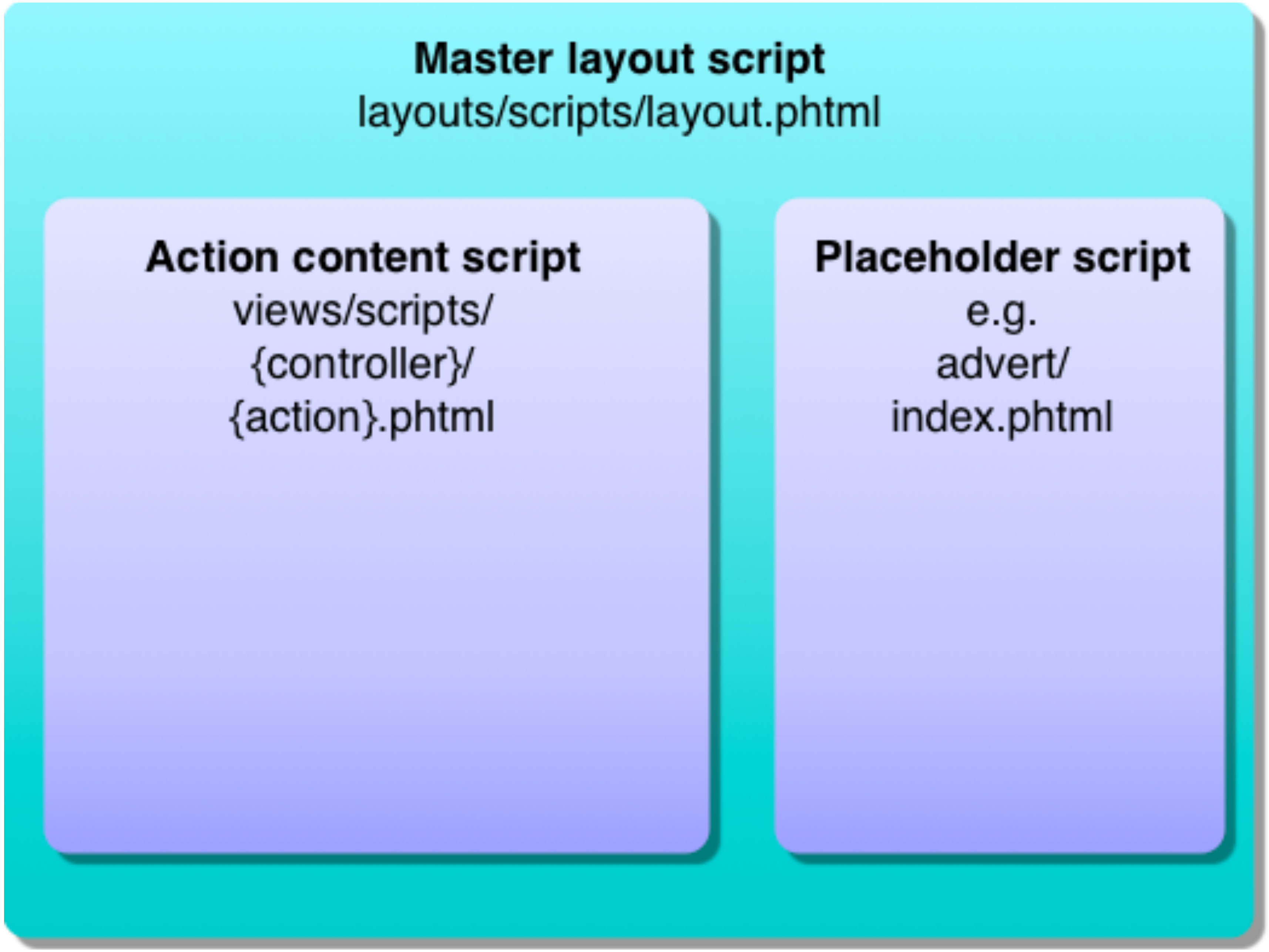
Index controller

```
class IndexController extends Zend_Controller_Action
{
  public function indexAction()
  {
    $this->view->title = 'Outstanding tasks';
    $tasksGateway = new Application_Model_DbTable_Tasks();
    $this->view->tasks = $tasksGateway->fetchOutstanding();
  }
}
```

indexAction() view script

```
<h1><?php echo $this->escape($this->title); ?></h1>
<table>
<tr>
    <th>Task</th>
    <th>Due</th>
</tr>
<?php foreach ($this->tasks as $task) : ?>
<tr>
    <td><?php echo $this->escape($task->title);?></td>
    <td><?php echo $task->due_date ?
        date('d m Y', strtotime($task->due_date)) : '-';?></td>
</tr>
<?php endforeach;?>
</table>
```

Layouts: Composite view



Master layout script
layouts/scripts/layout.phtml

Action content script
views/scripts/
{controller}/
{action}.phtml

Placeholder script
e.g.
advert/
index.phtml

Enable layout

```
$zf enable layout
Layouts have been enabled, and a default layout created at
/www/todolist/application/layouts/scripts/layout.phtml
A layout entry has been added to the application config file.
```

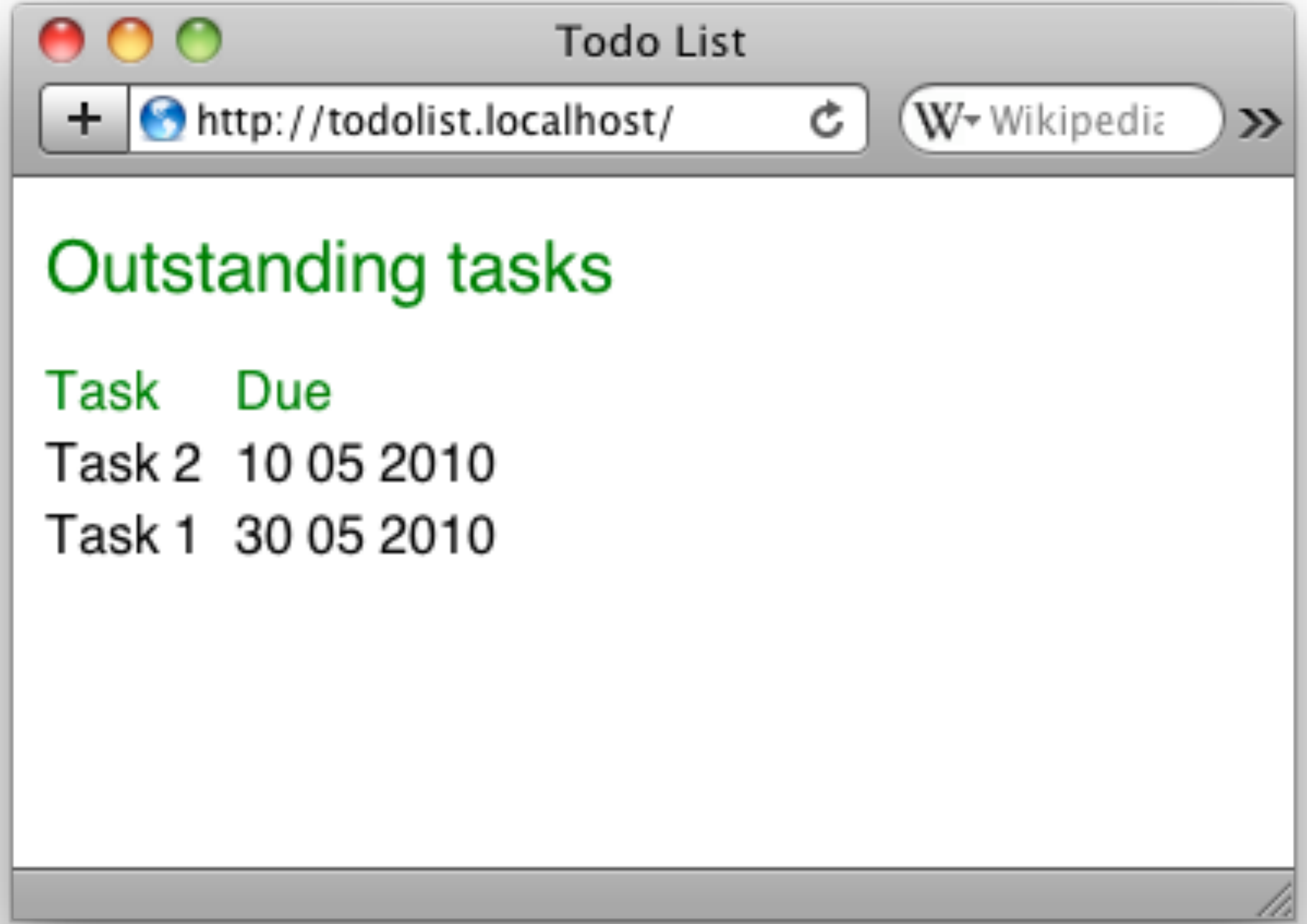
```php
<?php echo $this->layout()->content; ?>
```

Layout placeholders

```
// controller action method
$html = $view->render('sidebar.phtml');
$layout->sidebar = $html;


// view script
<?php echo $this->layout()->sidebar; ?>
```

Site-wide layout file

```
<?php
$this->headMeta()->appendHttpEquiv('Content-Type', 'text/html;charset=utf-8');
$this->headTitle('Todo List')->setSeparator(' - ');

echo $this->doctype(); ?>
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
    <?php echo $this->headMeta(); ?>
    <?php echo $this->headTitle(); ?>
    <?php echo $this->headLink()->appendStylesheet(
        $this->baseUrl().'/css/site.css'); ?>
</head>
<body>
<div id="content">
    <?php echo $this->layout()->content; ?>
</div>
</body>
</html>
```



Outstanding tasks

Task	Due
Task 2	10 05 2010
Task 1	30 05 2010

Forms

Zend_Form architecture

- Forms: the centre piece
- Forms are made up of:
  - Elements, Display groups, Sub-forms
- Each element is made up of:
  - Filters, Validators, Decorators

Create a form

```
$ zf create form task
Creating a form at /www/todolist/application/forms/Task.php
Updating project profile '/www/todolist/.zfproject.xml'
```

```
class Application_Form_Task extends Zend_Form
{
    public function init()
    {
      /* Form Elements & Other Definitions Here ... */
    }
}
```

Add an element

```
$title = new Zend_Form_Element_Text('title');
$title->setRequired(true);
$title->setLabel('Title');
$title->addFilters(array(
    'StringTrim', 'StripTags'
    ));
$title->addValidators(array(
    new Zend_Validate_StringLength(array('min'=>5))
    ));
$title->addElement($email);
```

Submit button

```php
$element = new Zend_Form_Element_Submit('submit-button');
$element->setIgnore(true);
$element->setLabel('Add task');
$this->addElement($element);
```

Action code

```php
public function addAction()
{
    $request = $this->getRequest();

    $form = new Application_Form_Task();
    if ($request->isPost()) {
        if ($form->isValid($request->getPost())) {
            // success
            $formData = $form->getValues();
        }
    }

    $this->view->form = $form;
}
```

View script

```
<h1>Add task</h1>
<?php
$this->form->setAction($this->url());
echo $this->form;
?>
```

Store to model

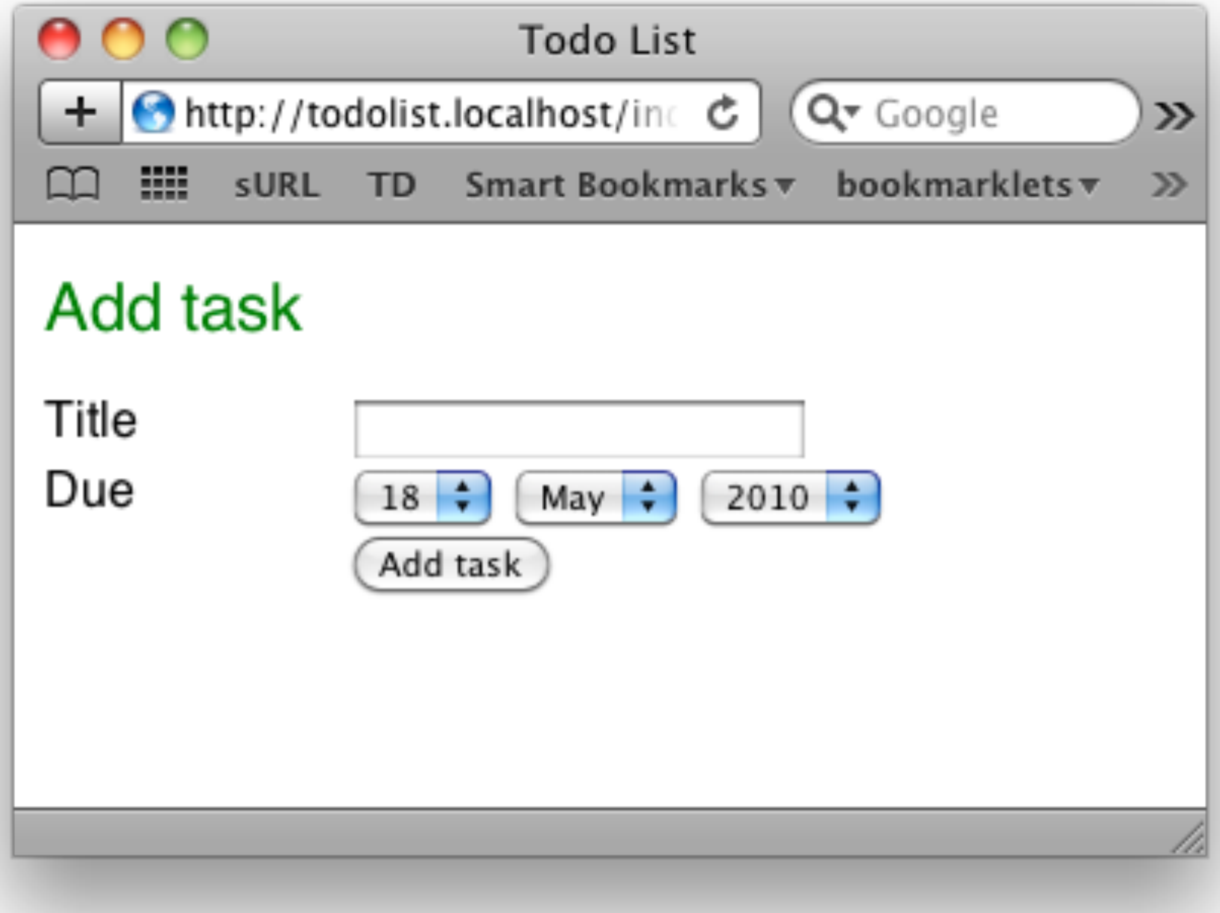
```
// IndexController::addAction()
if ($request->isPost()) {
    if ($form->isValid($request->getPost())) {
        // success
        $formData = $form->getValues();
        $tasks = new Application_Model_DbTable_Tasks();
        $tasks->addTask($formData);
        $this->_helper->redirector('index');
    }
}
```

Model methods

```
// Application_Model_DbTable_Tasks
public function addTask(array $taskData)
{
    $data = array('title' => $taskData['title'],
     'due_date' => $taskData['due_date']);
    $this->insert($data);
}

public function updateTask($id, $taskData)
{
    $data = array('title' => $taskData['title'],
     'due_date' => $taskData['due_date']);
    $this->update($data, 'id = ' . (int)$id);
}
```

Custom form element

```
// Application_Form_Task
$due = new Application_Form_Element_Date('due_date');
$due->setRequired(false);
$due->setLabel('Due');
$this->addElement($due);
```



Add task

Title

Due 18 May 2010

The date element (1)

```
// application/form/element/Date.php
class Application_Form_Element_Date
    extends Zend_Form_Element_Xhtml
{
    public $helper = 'formDate';

    public function init ()
    {
        $this->addValidator(new Zend_Validate_Date());
    }
```

The date element (2)

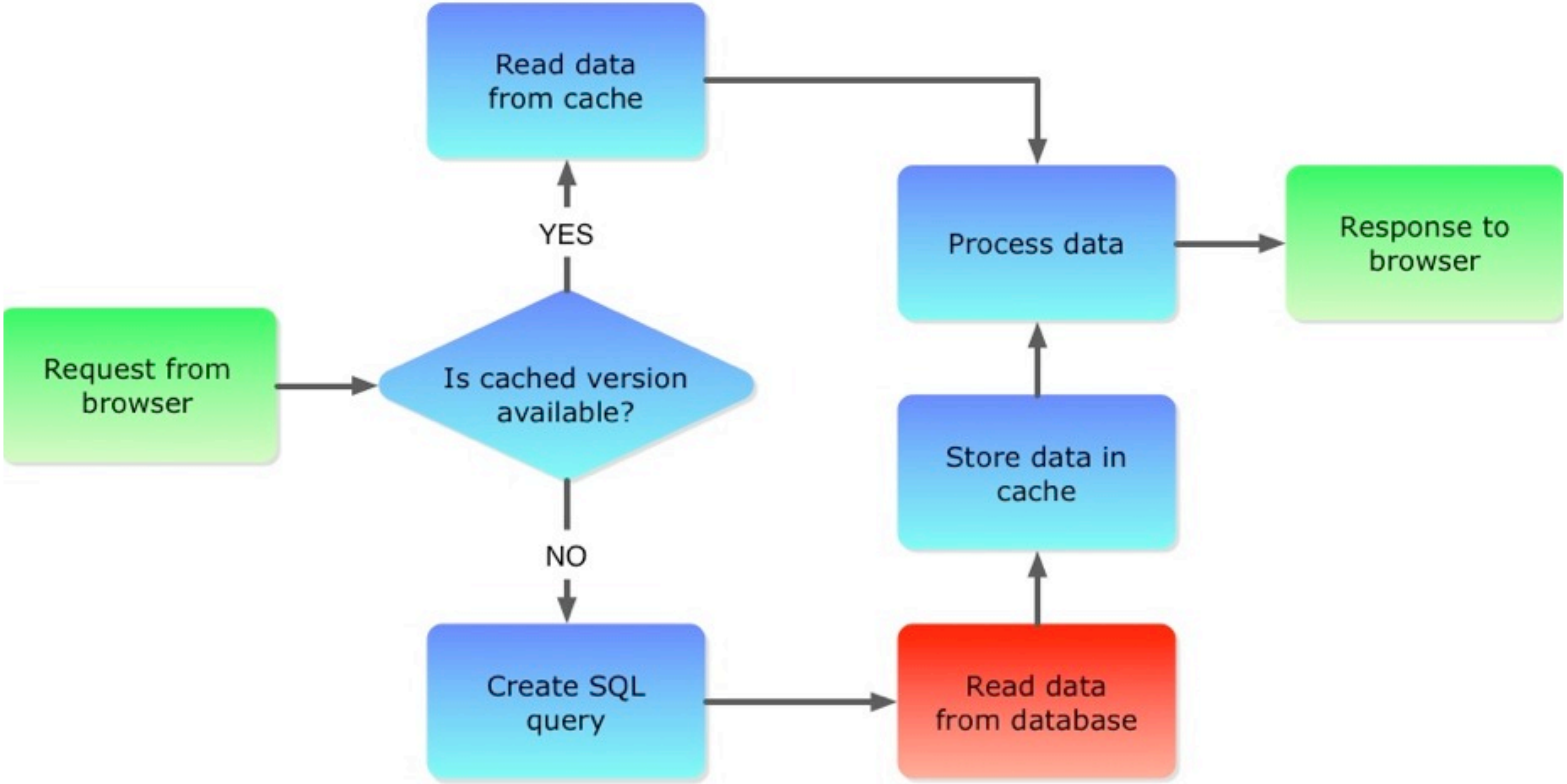
```
public function isValid ($value, $context = null)
{
    if (is_array($value)) {
        $value = $value['year'] . '-' .
                 $value['month'] . '-' .
                 $value['day'];
    }
    if($value == '--') {
        $value = null;
    }
    return parent::isValid($value, $context);
}
```

Date view helper

```
class Zend_View_Helper_FormDate extends Zend_View_Helper_FormElement
{
  public function formDate ($name, $value=null, $attribs=null)
  {
    $day = $value['day'];
    $dayMultiOptions = array(''=>'');
    for ($i = 1; $i <= 31; $i ++) {
      $key = str_pad($i, 2, '0', STR_PAD_LEFT);
      $dayMultiOptions[$key] = str_pad($i, 2, '0', STR_PAD_LEFT);
    }
    return $this->view->formSelect($name . '[day]',
        $day, $dayAttribs, $dayMultiOptions) . '&nbsp;' .
      $this->view->formSelect($name . '[month]',
        $month, $monthAttribs, $monthMultiOptions) . '&nbsp;' .
      $this->view->formSelect($name . '[year]',
```

Caching

Principle


Read data from cache
YES
Process data
Response to browser
Request from browser
Is cached version available?
Store data in cache
NO
Create SQL query
Read data from database

Bootstrap initialisation

```php
class Bootstrap extends Zend_Application_Bootstrap_Bootstrap
{
    function _initCache()
    {
      $frontendOptions = array('lifetime' => '7200',
        'automatic_serialization'=>true);
      $backendOptions = array(
        'cache_dir' => APPLICATION_PATH . '/../var/cache');
      $cache = Zend_Cache::factory('Core', 'File',
          $frontendOptions, $backendOptions);
      return $cache;
    }
}
```

Current code

```php
// Application_Model_DbTable_Tasks
public function fetchOutstanding()
{
    $select = $this->select();
    $select->where('completed = 0');
    $select->order(array('due_date ASC', 'id ASC'));
    return $this->fetchAll($select);
}
```

Zend_Cache in use

```
public function fetchOutstanding()
{
    $cacheId = 'outstandingTasks';
    $cache = $this->getCache();
    $rows = $cache->load($cacheId);
    if ($rows === false) {
        $select = $this->select();
        $select->where('completed = 0');
        $select->order(array('due_date ASC', 'id DESC'));
        $rows =  $this->fetchAll($select);
        $cache->save($rows, $cacheId, array('tasks'));
    }
    return $rows;
}
```

Unique id

Re-use existing code

Store to cache

Tag

Zend_Cache in use

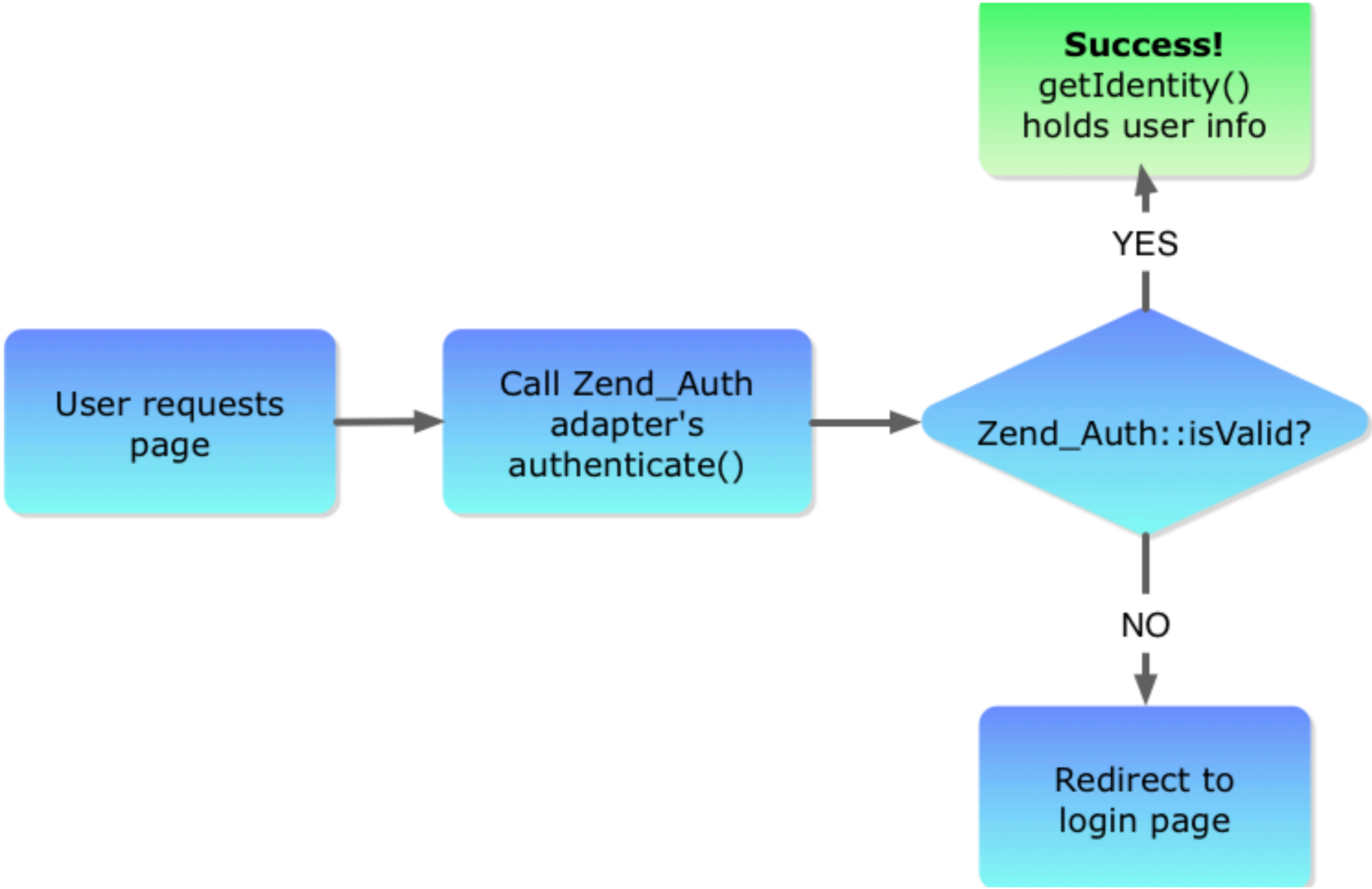
```php
// Application_Model_DbTable_Tasks
public function getCache()
{
    if (!$this->_cache) {
        $fc = Zend_Controller_Front::getInstance();
        $cache = $fc->getParam('bootstrap')
                    ->getResource('cache');
        $this->_cache = $cache;
    }
    return $this->_cache;
}
```

Emptying Zend_Cache

```
// Application_Model_DbTable_Tasks
protected function _cleanCache()
{
    $this->getCache()->clean(
        Zend_Cache::CLEANING_MODE_MATCHING_TAG,
        array('tasks'));
}
```

Authentication

Zend_Auth process



User requests page

Call Zend_Auth adapter's authenticate()

Zend_Auth::isValid?

YES

Success! getIdentity() holds user info

NO

Redirect to login page

AuthController

```
$ zf create controller Auth
Creating a controller at /www/todolist/application/controllers/
AuthController.php
Creating an index action method in controller Auth
Creating a view script for the index action method at /www/
todolist/application/views/scripts/auth/index.phtml
Creating a controller test file at /www/todolist/tests/
application/controllers/AuthControllerTest.php
Updating project profile '/www/todolist/.zfproject.xml'

$ zf create form Login
Creating a form at /www/todolist/application/forms/Login.php
Updating project profile '/www/todolist/.zfproject.xml'
```

Users table

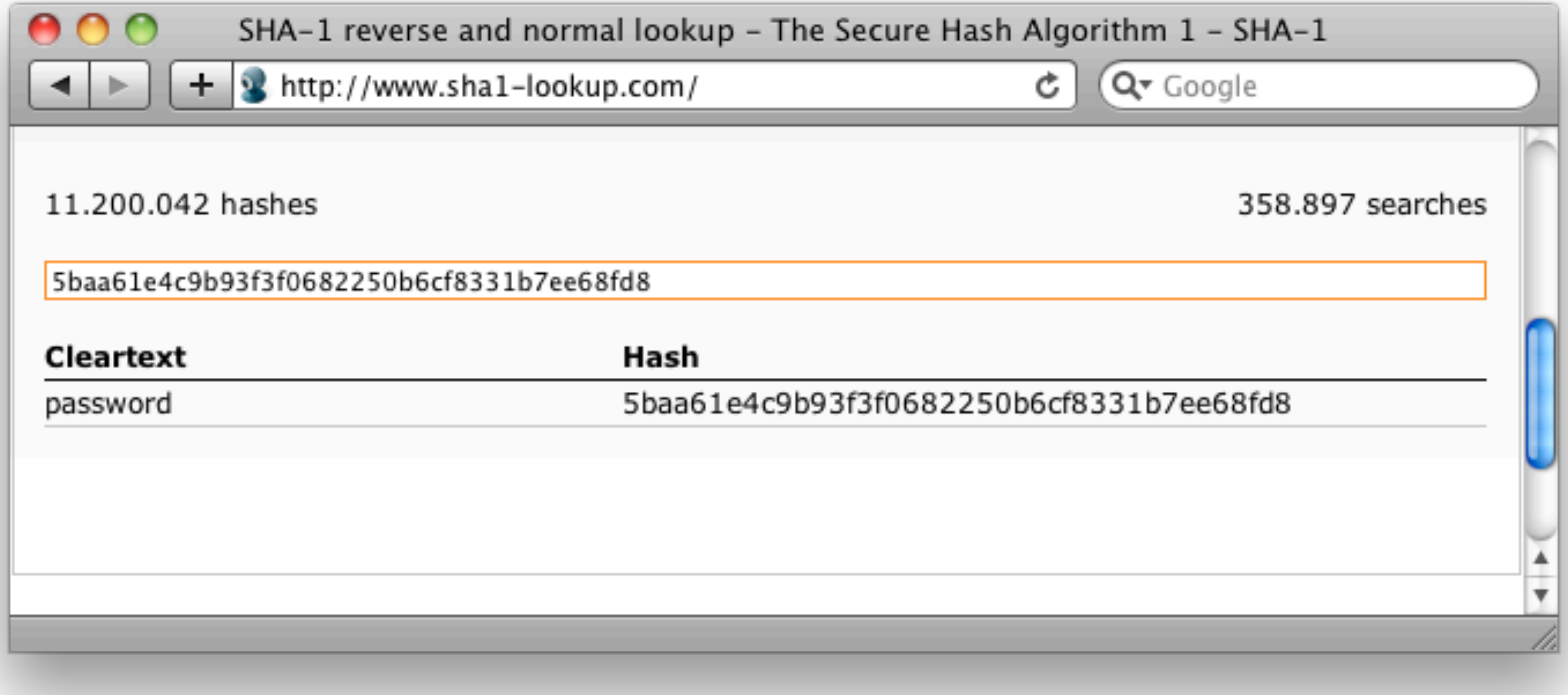
```
CREATE TABLE IF NOT EXISTS users (
  id int NOT NULL AUTO_INCREMENT,
  username varchar(50) NOT NULL,
  password varchar(50) NOT NULL,
  date_created datetime NOT NULL,
  PRIMARY KEY (id)
) ENGINE=InnoDB DEFAULT CHARSET=utf8;

-- insert first user
INSERT INTO users (username, password, date_created) VALUES
('admin', SHA1('vF%sdf&A!agDktpassword'), NOW());
```

application.ini

```
auth.salt = "vF%sdf&A!agDkt"
```

Reverse SHA1 lookups

```
SHA1('password') = 5baa61e4c9b93f3f0682250b6cf8331b7ee68fd8
SHA1('vF%sdf&A!agDktpassword') = e6ff3f23138c2975220f93b3ef7ffd159fb9171b
```


SHA-1 reverse and normal lookup - The Secure Hash Algorithm 1 - SHA-1
http://www.sha1-lookup.com/
Google

11.200.042 hashes

358.897 searches

5baa61e4c9b93f3f0682250b6cf8331b7ee68fd8

Cleartext	Hash
password	5baa61e4c9b93f3f0682250b6cf8331b7ee68fd8

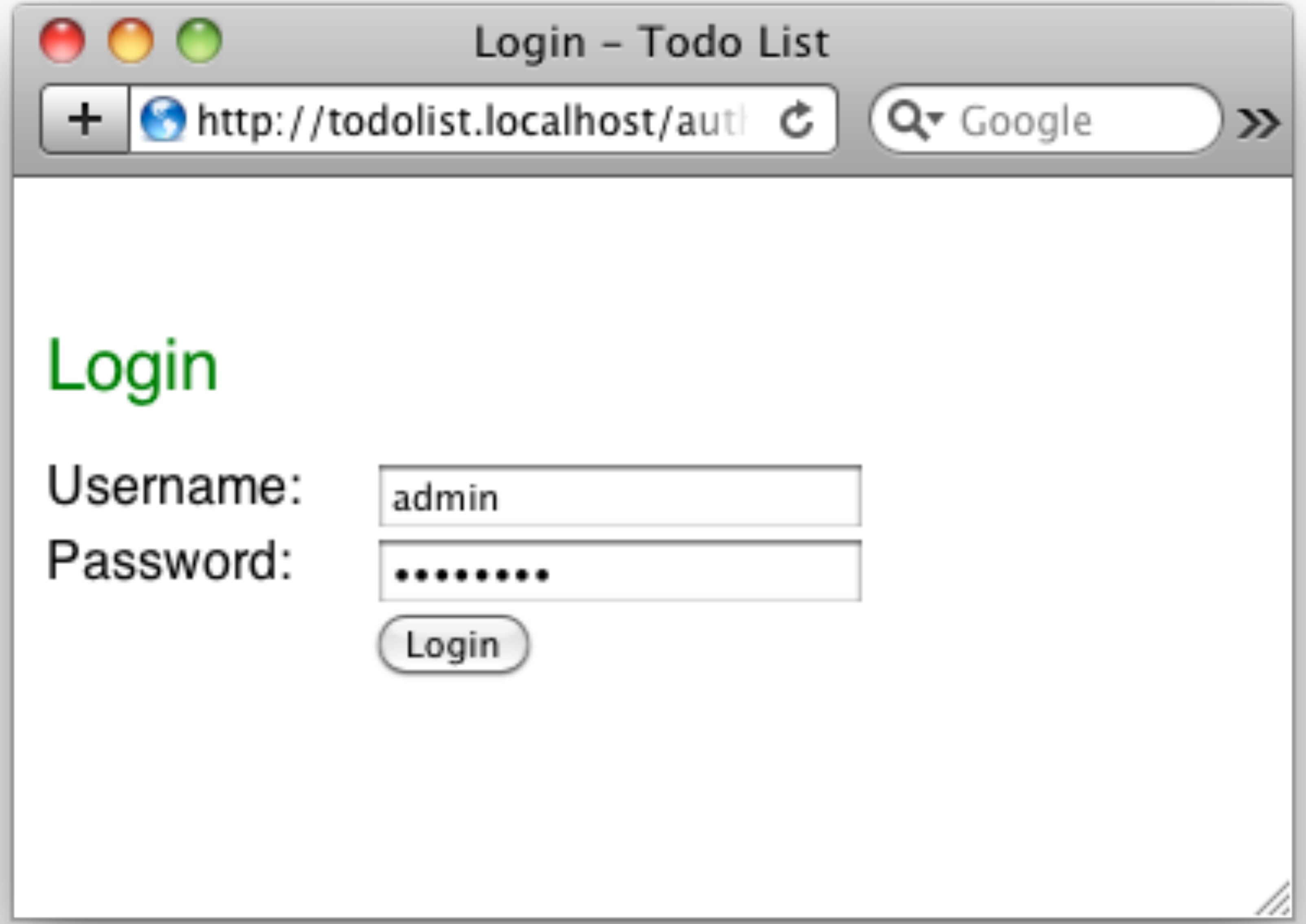
Login form

```
class Application_Form_Login extends Zend_Form
{
    public function init()
    {
        $this->addElement('text', 'username', array(
            'filters' => array('StringTrim', 'StringToLower'),
            'required'   => true,
            'label'      => 'Username:',
        ));
        $this->addElement('password', 'password', array(
            'filters'    => array('StringTrim'),
            'required'   => true,
            'label'      => 'Password:',
        ));
        // There's also a submit button
```

AuthController

```
class AuthController extends Zend_Controller_Action
{
    public function indexAction()
    {
        $form = new Application_Form_Login();
        $request = $this->getRequest();
        if ($request->isPost()) {
            if ($form->isValid($request->getPost())) {
                if ($this->_process($form->getValues())) {
                    // Success Redirect to the home page
                    $this->_helper->redirector('index', 'index');
                }
            }
        }
        $this->view->form = $form;
    }
```

Login form



Login

Username: admin

Password: •••••••••

AuthController

```php
protected function _process($values)
{
    // Get our authentication adapter and check credentials
    $adapter = $this->_getAuthAdapter($values);
    $auth = Zend_Auth::getInstance();
    $result = $auth->authenticate($adapter);
    if ($result->isValid()) {
        $data = $adapter->getResultRowObject();
        $auth->getStorage()->write($data);
        return true;
    }
    return false;
}
```

AuthController

```
protected function _getAuthAdapter($formData) {
    $dbAdapter = Zend_Db_Table::getDefaultAdapter();
    $authAdapter = new Zend_Auth_Adapter_DbTable($dbAdapter);
    $authAdapter->setTableName('users')
        ->setIdentityColumn('username')
        ->setCredentialColumn('password')
        ->setCredentialTreatment('SHA1(?)');

    $fc = Zend_Controller_Front::getInstance();
    $options = $fc->getParam('bootstrap')->getOptions();
    $password = $options['auth']['salt'].$formData['password'];

    $authAdapter->setIdentity($formData['username']);
    $authAdapter->setCredential($password);

    return $authAdapter;
}
```

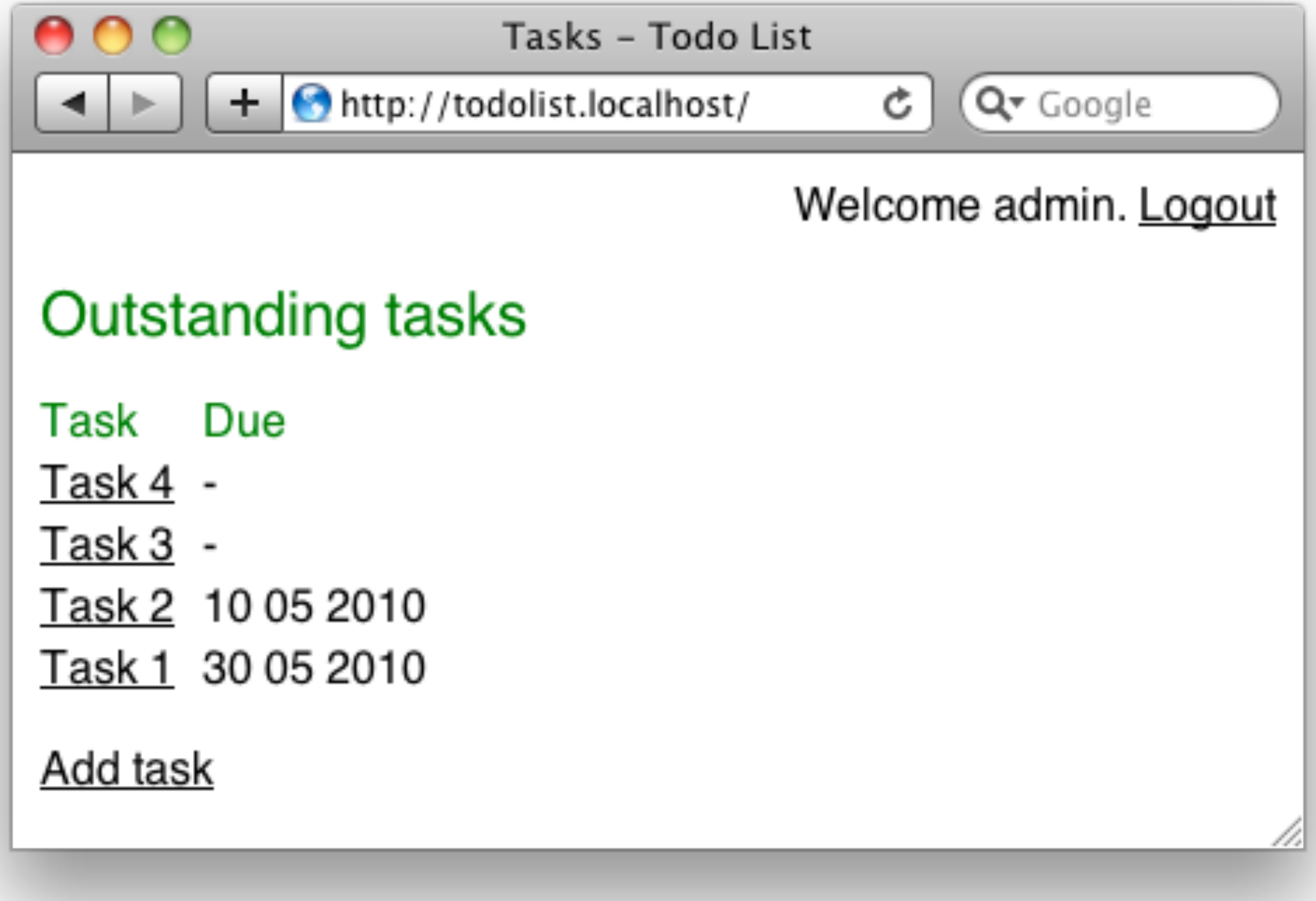
LoggedInAs view helper

```
class Zend_View_Helper_LoggedInAs extends Zend_View_Helper_Abstract
{
  public function loggedInAs ()
  {
    $auth = Zend_Auth::getInstance();
    if ($auth->hasIdentity()) {
      $username = $auth->getIdentity()->username;
      $url = $this->url(array('controller'=>'auth',
           'action'=>'logout'), null, true);
      return 'Welcome '.$username.'<a href="'.$url.'">Logout</a>';
    }

    $url = $this->url(array('controller'=>'auth', 'action'=>'index'));
    return '<a href="'.$url.'">Login</a>';
  }
}
```

Layout.phtml

```
<div id="header">
    <div id="logged-in-as"><?php echo $this->loggedInAs(); ?>
    </div>
</div>
```

LoggedInAs in action



Welcome admin. Logout

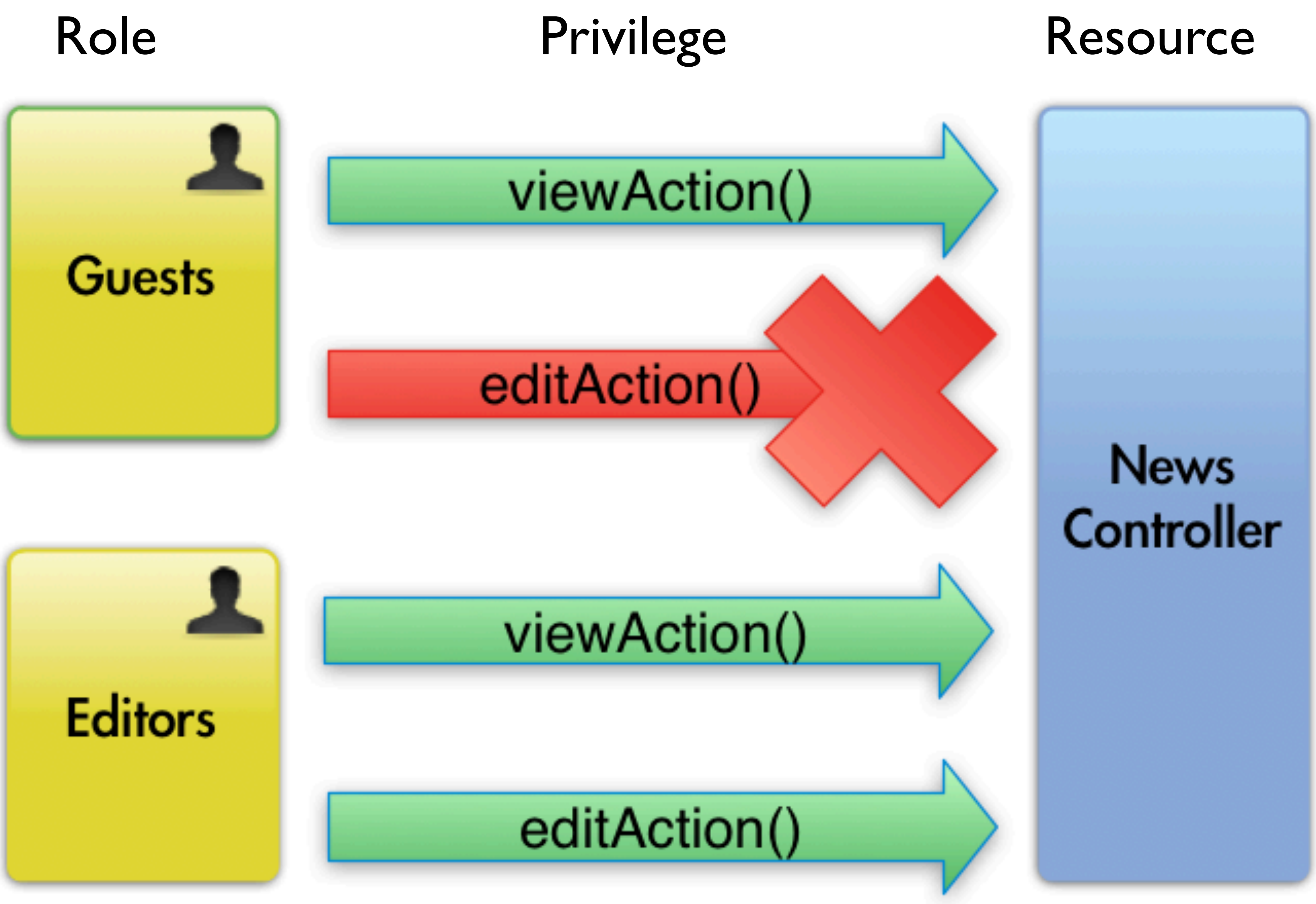
Outstanding tasks

Task	Due
Task 4	-
Task 3	-
Task 2	10 05 2010
Task 1	30 05 2010

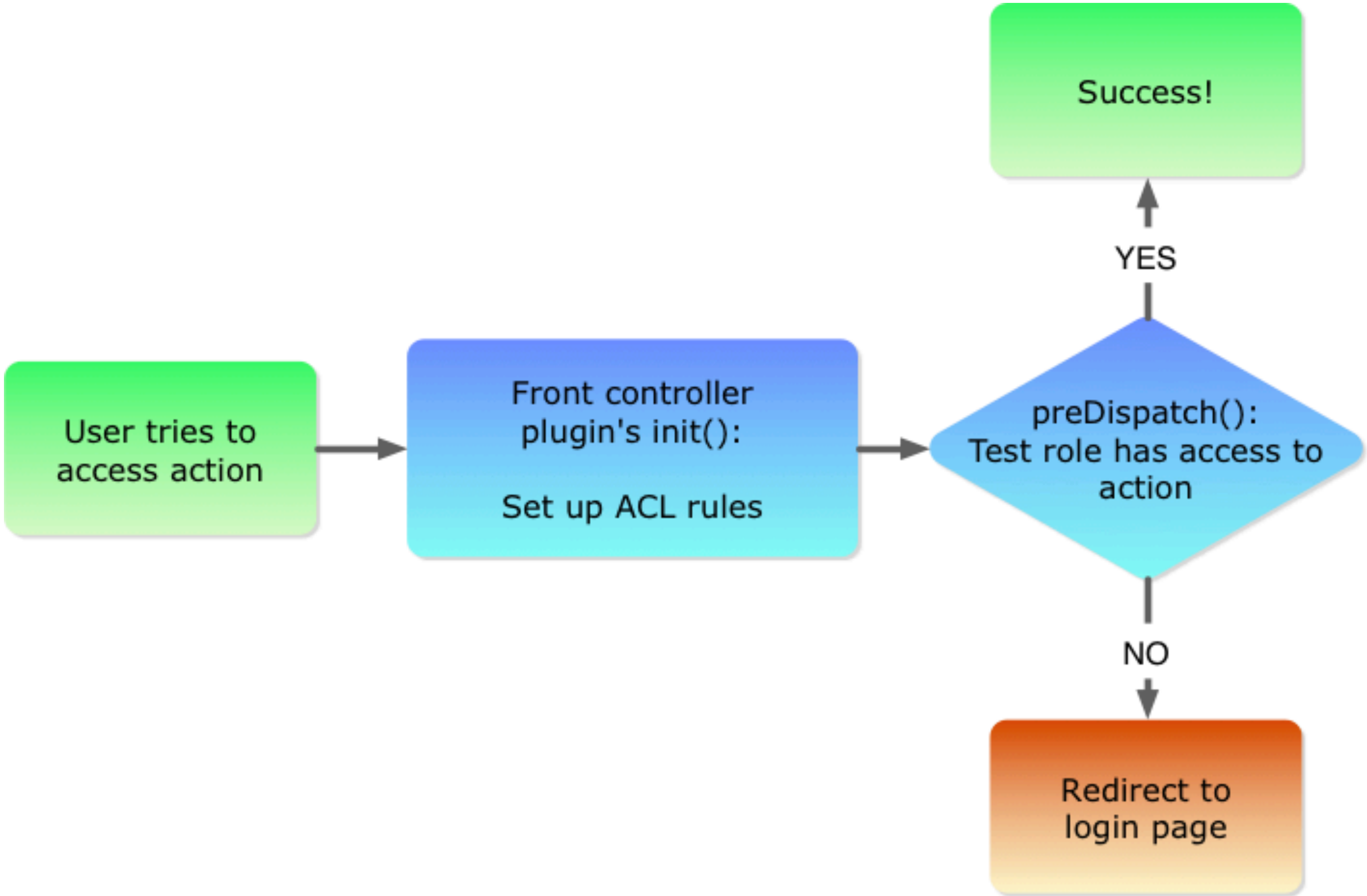
Add task

Access control

ACL Jargon


Role
Privilege
Resource
Guests
viewAction()
editAction()
Editors
viewAction()
editAction()
News Controller

ACL process



User tries to access action

Front controller plugin's init():

Set up ACL rules

preDispatch(): Test role has access to action

Success!

Redirect to login page

Front Controller plugin

```
// application/plugins/Acl.php
class Application_Plugin_Acl extends Zend_Controller_Plugin_Abstract
{
    public function dispatchLoopStartup(
        Zend_Controller_Request_Abstract $request)
    {
    }
}
```

```
// application/configs/application.ini
resources.frontController.plugins.acl = Application_Plugin_Acl
```

Acl plugin

```
public function dispatchLoopStartup(
    Zend_Controller_Request_Abstract $request)
{
  $acl = $this->_getAcl();
  $role = $this->_getRole();
  $resource = $request->getControllerName();
  $privilege = $request->getActionName();

  $allowed = $acl->isAllowed($role, $resource, $privilege);
  if (!$allowed) {
    $controller = 'auth';
    $action = 'index';
    $redirector = new Zend_Controller_Action_Helper_Redirector();
    $redirector->gotoSimpleAndExit($action, $controller);
  }
}
```

Setup Zend_Acl (1)

```
protected function _getAcl()
{
    if (null === $this->_acl) {
        $acl = new Zend_Acl();

        // Roles
        $acl->addRole('guest');
        $acl->addRole('user', 'guest');
        $acl->addRole('admin', 'user');

        // Resources
        $acl->add(new Zend_Acl_Resource('index'));
        $acl->add(new Zend_Acl_Resource('auth'));
        $acl->add(new Zend_Acl_Resource('error'));
```

Setup Zend_Acl (2)

Role Resource Privileges

```
        // Rules
        $acl->deny();
        $acl->allow('user', 'index',
            array('index', 'add', 'edit', 'view'));
        $acl->allow('admin', 'index', array('delete'));
        $acl->allow('guest', 'auth', null);
        $acl->allow('guest', 'error', null);
        $this->_acl = $acl;
    }

    return $this->_acl;
}
```

Get current role

```php
protected function _getRole()
{
  $auth = Zend_Auth::getInstance();
  if ($auth->hasIdentity()) {
    $identity = $auth->getIdentity();
    $role = empty($identity->role) ? 'user': $identity->role;
  } else {
    $role = 'guest';
  }

  return $role;
}
```

Zend Framework simplifies
development and maintenance

Devote your time to your
domain models as they are key.

Questions?

Thank you

feedback: http://joind.in/1562

email: rob@akrabat.com

twitter: @akrabat

QR Code