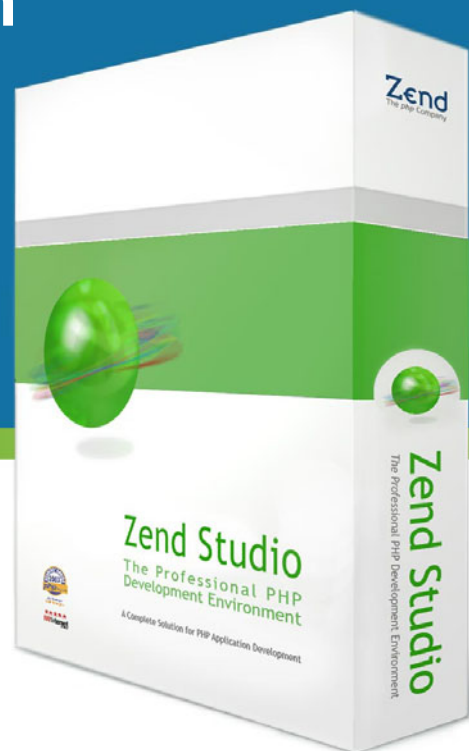




The PHP Company

Demo Video Instructions Rich Web Application Development

By Zend Technologies, Inc.



The following instructions explain how to recreate the demo application shown in the Zend Framework and Dojo demo video, using Zend Studio for Eclipse:

Prerequisites:

- Zend Studio for Eclipse 6.1 and above (www.zend.com/products/studio)
- A PHP Web Server configured to run Zend Framework 1.6-based applications.

Note:

The Zend PHP Web Server (www.zend.com/app-server/downloads) is automatically configured to run Zend Framework applications.

To create a new Zend Framework 1.6 project:

1. Launch Zend Studio for Eclipse.
2. Go to New | Project | Zend Framework Project.
The new Zend Framework Project wizard is displayed.
3. Ensure that Zend Framework 1.6 is selected in the Zend Framework version drop-down list and that the 'Enable Dojo support' checkbox is marked.
This will add Zend Framework 1.6 libraries and the Dojo toolkit to the project's include path and give you access to the relevant [code assist](#) options.
4. Click Finish.
5. Click Yes if prompted to open the Zend Framework perspective.
This displays the MVC Outline view and the Zend Framework search box.
A new PHP project will be created, arranged in a Model-View-Controller pattern.
6. You will now need to edit the auto-generated project with the code required for your application.

To edit the code in the project to create your application:

1. Right-click the project's "library" folder and select New | PHP Class.
The New PHP Class wizard is displayed.
2. Enter 'MeetingForm' in the Class Name field This will automatically create the name for both the class and the file.
3. Click Browse next to the Superclass field.
The Superclass selection dialog is displayed. This allows you to select any PHP superclass, including those provided in Zend Framework 1.6.
4. Enter "Zend_Dojo_Form" in the superclass selection field.

Note:

Typing the first few letters of the required superclass will filter the list. Wild cards are also enabled, so typing '*dojo_form' will display the required option.

5. Select Zend_Dojo_Form and click OK.

- Click Finish in the New PHP Class Wizard.

A template class file will be created, giving you access to the Zend Framework classes and Dojo widgets required for creating a form.

- In this class you want to create fields for entering a name and meeting date, and a submit button to post the information.

To create the fields in your application, enter the following code on line #6 (after `class MeetingForm extends Zend_Dojo_Form {}`):

```
public function init() {
    // Set up form
    $this->setAction($this->getView()->url()->setMethod('post');
    $this->setName("Name");
    $this->addAttribs(array('onSubmit' => 'return
validate(this)'));

    // Meeting recipient name field
    $name = $this->createElement('ValidationTextBox', 'name',
        array ('regExp' => '[a-zA-Z ]+',
            'require' => 'true',
            'invalidMessage' => 'Invalid Meeting Recipient
Name'
        )
    );
    $name->addValidator(new Zend_Validate_Alpha(true));
    $name->setLabel('Name')->setRequired(true);

    // Date field
    $date = $this->createElement('DateTextBox', 'date',
        array('require' => 'true',
            'invalidMessage' => 'Invalid date format'
        )
    );
    $date->setLabel('date')->setRequired(true);

    // Submit button
    $submit = $this->createElement('SubmitButton', 'submit',
        array('required' => false, 'ignore' => true )
    );

    // Add elements to form:
    $this->addElements(array($name, $date, $submit));
}
```

8. If you type the code out manually, you will get code assist options for the various elements.

Note:

The ValidationTextBox and DateTextBox fields are Dojo widgets which allow you to easily create fields with input validation. This means that if a user enters input which is invalid (e.g. entering letters in the date field), an error pop-up will be displayed. The DateTextBox comes with automatic date validation, as well as providing a pull-down calendar for automatic date selection.

9. You now need to define the actions to be executed after the form's Submit button has been clicked. This is handled by the project's Controller module.
10. Open the IndexController.php file, created in the project's application | default | controllers folder.

11. Enter the following code on line #11 (after `require_once`

```
'Zend/Controller/Action.php');
```

```
require_once 'library/MeetingForm.php';
```

12. Enter the following code on line #15 (after `class` `IndexController` `extends` `Zend_Controller_Action` {}):

```
public function getForm() {  
    return new MeetingForm();  
}
```

13. Replace line #24 (`// TODO` Auto-generated

`IndexController::indexAction()` action), with the following code:

```
// Create the form  
$form = $this->getForm();  
$this->view->form = $form;  
  
if (!$this->getRequest()->isPost()) {  
    return;  
}  
  
if (!$form->isValid($_POST)) {  
    return;  
}
```

14. This code posts the information submitted by the user in the input fields.

15. You should now need to write code which schedules a meeting in a calendar. However, for the purposes of this demonstration you will simply return a message incorporating the inputted information, by entering the following code:

```

        $values = $form->getValues();
// Output Message
    {
        $this->view->message = "<b> Meeting with " .
$values ['name']. " is scheduled for " . $values ['date'] . "<b>";
    }
// TODO: schedule a meeting in the database here...

```

16. You now need to make some small changes to your view files in order to define the 'look' of the application. This is easily done using Dojo's built in styles.
17. Open the main.phtml file, which was created in the project's application | default | layouts folder.

This file defines the layout of the application.

18. Set the style to Dojo's 'tundra' style by doing the following:

- i) On line #22, enter `echo $this->dojo();`
- ii) On line #26, replace `<body>` with `<body class="tundra">`.

19. Open the index.phtml file created in the project's application | default | views | scripts | index folder.

This is a view file, so code assist options for all [Zend View Helpers](#) will be available.

20. Replace the titles on lines #10 and #11 with 'Meeting Scheduler' . i.e:

```

$this->headTitle('Meeting Scheduler');
$this->placeholder('title')->set('Meeting Scheduler');

```

21. Add the required style sheets by entering the following code on line #12:

```

$this->dojo()-
>addStylesheet('scripts/dojo/dojo/resources/dojo.css');
$this->dojo()->addStylesheetModule('dijit.themes.tundra');

```

22. On line #17, replace `Hello, world!` with:

```

<?php echo $this->form ?>
<?php echo $this->message ?>

```

23. The application is now ready to be deployed and run.

To run your application:

1. Copy the application files to a PHP Web server (we recommend the [Zend PHP Web Server](#)).

Note:

Ensure your PHP Web server is configured to run Zend Framework 1.6 projects. (Go to http://files.zend.com/help/index.htm#Zend-Studio-Eclipse-Help/running_zend_framework_projects.htm for more information).

2. From within Zend Studio for Eclipse, run the application by right-clicking the index.php file (in the project's public folder) and selecting Run As | PHP Web Page -or- going to Run | Open Run Dialog and configuring the running options and server information. (See http://files.zend.com/help/index.htm#Zend-Studio-Eclipse-Help/running_php_web_pages.htm for information on running applications from Zend Studio for Eclipse).
3. If prompted to set [Path Mapping](#) for the Abstract.php file, select the Zend\Controller\Plugin\Abstract.php option from the list. This dialog will not appear the next time you run your application.
4. The application will be displayed in the browser.
5. To use the application, enter different values in the Name and Date fields. You should see that if you enter an invalid value (e.g. numbers in the 'Name' field or an invalid date in the 'Date' field), a popup will appear indicating that the input is invalid.
6. Click the Submit button for a message to be displayed using the input you entered.

What Next?

- To learn more about using Zend Framework in order to enhance your application, go to <http://framework.zend.com>.
- To learn more about using Zend Studio for Eclipse's Zend Framework functionality, go to http://files.zend.com/help/index.htm#Zend-Studio-Eclipse-Help/zend_framework_integration.htm.
- For more Zend Studio for Eclipse videos, go <http://www.zend.com/en/products/studio/videos>.