About the Tutorial

Phalcon is an open source framework quite popular among developers. It is a combination of PHP and C language. Phalcon is developed by Andres Gutierrez and his group of collaborators.

This tutorial provides an overall idea on Phalcon PHP framework and how you can use it.

Audience

This tutorial is basically developed for those who want to learn Phalcon from ground up. The target audience of learning this framework includes students, PHP developers, web designers and web developers.

Prerequisites

Before starting with this tutorial, the user should have knowledge of HTML, CSS and PHP along with an understanding of MVC framework. It would be an added advantage if you have prior exposure to other traditional frameworks like Laravel, Yii, or Codeigniter.

Copyright and Disclaimer

© Copyright 2016 by Tutorials Point (I) Pvt. Ltd.

All the content and graphics published in this e-book are the property of Tutorials Point (I) Pvt. Ltd. The user of this e-book is prohibited to reuse, retain, copy, distribute or republish any contents or a part of contents of this e-book in any manner without written consent of the publisher.

We strive to update the contents of our website and tutorials as timely and as precisely as possible, however, the contents may contain inaccuracies or errors. Tutorials Point (I) Pvt. Ltd. provides no guarantee regarding the accuracy, timeliness or completeness of our website or its contents including this tutorial. If you discover any errors on our website or in this tutorial, please notify us at contact@tutorialspoint.com
## Table of Contents

About the Tutorial ............................................................................................................. i  
Audience ........................................................................................................................... i  
Prerequisites ..................................................................................................................... i  
Copyright and Disclaimer ............................................................................................... i  
Table of Contents ............................................................................................................. ii  

1. PHALCON – OVERVIEW ............................................................................................... 1  
2. PHALCON – ENVIRONMENTAL SETUP ...................................................................... 3  
3. PHALCON – APPLICATION STRUCTURE ................................................................... 8  
4. PHALCON – FUNCTIONALITY ..................................................................................... 12  
5. PHALCON – CONFIGURATION ..................................................................................... 15  
6. PHALCON – CONTROLLERS ....................................................................................... 20  
7. PHALCON – MODELS ................................................................................................. 23  
8. PHALCON – VIEWS ..................................................................................................... 29  
9. PHALCON – ROUTING ................................................................................................. 33  
10. PHALCON – DATABASE CONNECTIVITY .................................................................... 35  
11. PHALCON – SWITCHING DATABASES ....................................................................... 37  
12. PHALCON – SCAFFOLDING APPLICATION ................................................................ 39  
    Designing the Login Page ............................................................................................ 46  
    Creating Views ............................................................................................................. 49  
    Category Management ............................................................................................... 51
13. PHALCON − QUERY LANGUAGE ................................................................. 64
   Creating a PHQL Query ............................................................................. 64
   PHQL Life Cycle ...................................................................................... 66

14. PHALCON − DATABASE MIGRATION .................................................... 67
   Understanding the Anatomy of Migrated Files ......................................... 69

15. PHALCON − COOKIE MANAGEMENT .................................................. 72

16. PHALCON − SESSION MANAGEMENT .................................................. 76

17. PHALCON − MULTI-LINGUAL SUPPORT ............................................. 79

18. PHALCON − ASSET MANAGEMENT ..................................................... 82

19. PHALCON − WORKING WITH FORMS ............................................... 86

20. PHALCON − OBJECT DOCUMENT MAPPER ........................................ 89

21. PHALCON − SECURITY FEATURES ..................................................... 94
   Hashing Password .................................................................................... 94
   Cross-Site Request Forgery (CSRF) ....................................................... 97
Phalcon is introduced as one of the recent frameworks of PHP, developed by a group of enthusiastic developers. Phalcon is a loosely coupled framework, which means it allows its objects to be treated like glue components, based on the needs of application.

Phalcon offers some unique features in comparison to other frameworks (traditional or existing) in PHP. Following are some of the most prominent features of Phalcon:

- It is a full stack open source framework.
- A user needs quite less amount of code to take advantage of several components.
- It can be used to create an independent framework as required. For example, if we just need Phalcon's Cache component, we can use it in any application written either in pure PHP or using a framework.
- For a developer having knowledge of Model-View-Controller (MVC) and Object-Relational Modeling (ORM), working with Phalcon is like a cakewalk.

Performance

The performance of Phalcon is a feature which distinguishes it from other traditional frameworks of PHP. Phalcon has a combination of both PHP and C; each of them can be used as a stand-alone module. The compilation of every request is considered on a higher speed in Phalcon which makes everything seem out-of-the-box.
**C Language**

Phalcon is compatible with C which increases the compilation rate. Also, C in combination with Phalcon provides Object Relational Mapping (ORM) which provides consistency with models created. Every model created in Phalcon is associated with the table of relational database. ORM in Phalcon is purely implemented in C.

**Developer Tools**

Developer tools are used for developing web application. These tools help in generating scaffold application with a combination of all features (C – Create, R – Read, U – Update, D – Delete). Developer tools also include extensible support for third party libraries to be implemented in Phalcon.

**Object Relational Mapping**

Phalcon supports a wide range of databases. It is not limited to access of relational databases. It supports both relational and non-relational databases which is like adding a feather to the cap of Phalcon framework.

**Phalcon Vs Other Frameworks**

The following table highlights how Phalcon differs from other popular frameworks such as Yii and Laravel.

<table>
<thead>
<tr>
<th></th>
<th>Yii</th>
<th>Laravel</th>
<th>Phalcon</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Type of Projects</strong></td>
<td>Yii helps in creating large scale projects like forums, portals, CMS, RESTful web services, etc.</td>
<td>Laravel is used for building web applications. It is known for exquisite and sophisticated syntax.</td>
<td>Phalcon is used to design variety of projects.</td>
</tr>
<tr>
<td><strong>Database Support</strong></td>
<td>Yii supports all relational and non-relational databases.</td>
<td>Laravel supports all relational databases.</td>
<td>Phalcon gives equal support to relational and non-relational databases.</td>
</tr>
<tr>
<td><strong>Language</strong></td>
<td>Yii framework is purely written in PHP.</td>
<td>Laravel is written in PHP and follows MVC pattern.</td>
<td>Phalcon includes both PHP and C.</td>
</tr>
<tr>
<td><strong>Scalability</strong></td>
<td>Yii is quite scalable for small and medium projects.</td>
<td>Scalability is high for Laravel with all kinds of projects.</td>
<td>Good for medium projects.</td>
</tr>
<tr>
<td><strong>Performance</strong></td>
<td>Comparatively low.</td>
<td>High but less in comparison with Phalcon.</td>
<td>High performance.</td>
</tr>
</tbody>
</table>
Prerequisites: We need WAMP/LAMP/MAMP or XAMPP stack for this framework.

Following are the steps for the installation process of Phalcon framework in Windows.

Step 1: Phalcon installation is completely dependent on dll file. DLL (Dynamic Link Library) creates the required package and plugins for Phalcon.

The following link is used for downloading dll file: https://phalconphp.com/en/download

Step 2: Download the required dll file. Check for the appropriate configuration of the system and download the required dll file. After downloading the file, extract phalcon-php.dll to /php/ext in the xampp folder.

Step 3: Edit the path in php.ini file to get it configured in a similar manner of other .dll files.
Step 4: Once the path is edited, restart the **xampp/wamp** stack. It will be clearly visible in the dashboard, once the **dll** file is properly set.

```php
extension=php_gd2.dll
extension=php_gettext.dll
;extension=php_gmp.dll
;extension=php_intl.dll
;extension=php_ldap.dll
;extension=php_interbase.dll
;extension=php_iconv.dll
extension=php_mbstring.dll
extension=php_pdo_mysql.dll
extension=php_cex.dll ; Must be after mbstring as it depends on it
extension=php_mysql.dll
extension=php_mysql.dll
extension=php_mysqli.dll ; Use with Oracle 10gR2 Instant Client
extension=php_mysqlndi.dll ; Use with Oracle 11gR2 Instant Client
extension=php_openssl.dll
extension=php_pdo_firebird.dll
extension=php_pdo_mysql.dll
extension=php_pdo_oci.dll
extension=php_pdo_odbc.dll
extension=php_pdo_pgisql.dll
extension=php_pdo_sqlite.dll
extension=php_pdo_sqlite.dll
extension=php_pdo_sqlite1.dll
extension=php_smf.dll

; The MIBS data available in the PHP distribution must be installed.
;extension=php_mysql.dll
```

### phalcon

Web framework delivered as a C-extension for PHP

<table>
<thead>
<tr>
<th>phalcon</th>
<th>enabled</th>
</tr>
</thead>
<tbody>
<tr>
<td>Author</td>
<td>Phalcon Team and contributors</td>
</tr>
<tr>
<td>Version</td>
<td>3.0.1</td>
</tr>
<tr>
<td>Build Date</td>
<td>Aug 24 2016 11:24:53</td>
</tr>
<tr>
<td>Powered by</td>
<td>Zephir</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Directive</th>
<th>Local Value</th>
<th>Master Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>phalcon.db.escape_identifiers</td>
<td>On</td>
<td>On</td>
</tr>
<tr>
<td>phalcon.db.force_casing</td>
<td>Off</td>
<td>Off</td>
</tr>
<tr>
<td>phalcon.orm.cast_onhydrate</td>
<td>Off</td>
<td>Off</td>
</tr>
<tr>
<td>phalcon.orm.column_renaming</td>
<td>On</td>
<td>On</td>
</tr>
<tr>
<td>phalcon.orm.enable_implicit_joins</td>
<td>On</td>
<td>On</td>
</tr>
<tr>
<td>phalcon.orm.enable_literals</td>
<td>On</td>
<td>On</td>
</tr>
<tr>
<td>phalcon.orm.events</td>
<td>On</td>
<td>On</td>
</tr>
<tr>
<td>phalcon.orm.exception_on_failed_save</td>
<td>Off</td>
<td>Off</td>
</tr>
<tr>
<td>phalcon.orm.ignore_unknown_columns</td>
<td>Off</td>
<td>Off</td>
</tr>
<tr>
<td>phalcon.orm.late_state_binding</td>
<td>Off</td>
<td>Off</td>
</tr>
<tr>
<td>phalcon.orm.not_null_validations</td>
<td>On</td>
<td>On</td>
</tr>
<tr>
<td>phalcon.orm.virtual_foreign_keys</td>
<td>On</td>
<td>On</td>
</tr>
</tbody>
</table>
**Step 5:** After downloading the package, set the path variable in the system properties.

![System Properties](image)

**Step 6:** The **dll** files and Phalcon tools together help in creating the project/web application. The user can verify through command prompt whether Phalcon framework has been successfully installed. The output will be displayed as shown in the following screenshot.
Step 7: Once this necessary output is received, create a project using the following command:

```
phalcon create-project <project-name>
```

The following output will be displayed.

Step 8: The web application is successfully created. Click the following URL:

```
http://localhost/demo1
```
The output will be displayed as shown in the following screenshot. It is the welcome page for Phalcon PHP.

Congratulations!

You're now flying with Phalcon. Great things are about to happen!
This page is located at views/index/index.volt
In this chapter, we will discuss the Application Structure of Phalcon. Following is the complete directory structure of a Phalcon project.

There is one root folder which is considered as the **code base** and is publicly available for the web server. It is also called as **web directory**. Other folders outside the web root directory are considered out of reach for the web server and for Phalcon project.

Once a project is created, the directory structure will be visible as follows in the **wamp/xampp** folder. Consider for the project which we created in the previous chapter.
Following are the folders and sub-folders of the project.

**App**

This folder consists of all vital script files and folders. The complete web application is designed on the basis of “app” folder. The configuration files help in assisting the necessary configuration for running the application smoothly.

Following is the detailed view of app folder for the given Phalcon web application.

It consists of config, controllers, library, migrations, models and views.

**Config**
All the configuration required for the web application in Phalcon is comprised in this folder. It includes information related to database connectivity, third-party libraries to be added if any, and the services to be included.

**Controllers**

All the controllers are included in this folder. They are used for processing requests and generating response.

**Library**

Third-party libraries for the web application (apart from the existing Phalcon framework).

**Migrations**

This sub-folder consists of all the files associated with data migration, which can also be used in any other framework.

**Models**

Models include all the logic required to interact with the database. It is actually used for data representation.

**Views**

It constitutes all the views related to the web application. These views are displayed to the end users with the help of controllers.

**Cache**

This directory includes data related to caching, which helps in improving the performance.

**Public**

It includes all the folders for asset management purpose which comprises of CSS, JavaScript, files to be uploaded, and some meta data.
.htaccess File

Web servers running on Apache Web Server software use .htaccess as a configuration file. When it is placed in a directory, all the necessary configuration is loaded as soon as the server is started.

For example, it is possible to configure a website so that it will available only to specific IP addresses with .htaccess file.
**Model View Controller (MVC)** is a software design and structural pattern for developing web-based applications. This software architectural separates the representation of information from the user's interaction with it.

The MVC model defines the web applications with three logic layers.

**Model**
Models are objects which represent knowledge. There should be a one-to-one relationship between the model and its parts. It includes all the logic to be used for database connectivity and performing CRUD operations.

**View**
A view is a visual representation of its model. View interacts with the model or its parts and gets the data necessary for the presentation from the model. This is achieved by sending requests and receiving appropriate responses. View includes all the data that end user sees.

**Controller**
A controller acts as the intermediary between the user and the system (model and view). It accepts the request from the user, through the view sends it to the model. The model manipulates it and sends the response to the controller, which is displayed as the output to the end user through view.
The controller receives such user output and translates it into the appropriate messages. These messages are used by view to display as appropriate responses.

**Workflow in Phalcon**

The workflow in Phalcon is as follows:

- The user interacts with the user interface (view) and the interaction is maintained with the help of some method/event.
- These methods and events are handled by the controller.
- The controller accesses the model by updating the user’s action.
- View uses the model to generate an appropriate output.
- View fetches data from its model. The model has no direct interaction with view.
- The user interface waits for further user interactions, which starts with a new cycle of request and response.
Phalcon includes directories for Model, View, and Controller. The following screenshot gives a better scenario.
All business logic is described in the controller, and the model interacts with the database which includes all files with respect to each and every table.

**Note:**
- All the controllers created in Phalcon web application extends **Phalcon\Mvc\Controller**.
- All the models associated with the database tables extends **\Phalcon\Mvc\Model**.
End of ebook preview
If you liked what you saw...
Buy it from our store @ https://store.tutorialspoint.com