

# [By OnlineInterviewQuestions.com](http://OnlineInterviewQuestions.com)

## Phalcon interview questions

### Phalcon interview questions and answers

#### Q1. What is Phalcon?

**Phalcon** is a free open source PHP framework based on MVC design pattern that is delivered to end users as a C -extension for the PHP language providing high performance and lower resource consumption. Phalcon is developed by Andres Gutierrez and written in Zephir/C.

#### Q2. Explain Dependency Injection in Phalcon?

**Phalcon** is built upon a powerful yet easy to understand and use a pattern called Dependency Injection.

Phalcon's Dependency injection initializes or define services once – and use them virtually anywhere throughout the application.

#### Q3. List basic features provided by Phalcon?

Below is the list of basic features provided by Phalcon PHP framework.

- Low overhead
- MVC & HMVC Pattern
- Dependency Injection
- Support for Rest
- Autoloader
- Router

#### Q4. Explain directory structure of Phalcon Framework?

The root folder of Phalcon Php contains files and directories

- App
  - Config
  - Controllers
  - Library
  - Migrations
  - Models
  - Views

- Cache
- Public
  - Css
  - files
  - img
  - js
  - temp
- .htaccess
- .htaccess
- .htrouter.php
- index.html

Read 40+ [Core PHP Interview Questions](#)

## Q5. [How to register namespaces/prefixes, directories or classes in Phalcon?](#)

In Phalcon you can register namespaces, prefixes, directories or classes by loader class. Below are sample usage for same.

### Registering namespaces/prefixes in Phalcon

```
use Phalcon\Loader;
$loader = new Loader();

// Register some namespaces
$loader->registerNamespaces(
    [
        "Example\Base"    => "vendor/example/base/",
        "Example\Adapter" => "vendor/example/adapter/",
        "Example"         => "vendor/example/",
    ]
);

// Register autoloader
$loader->register();
```

### Registering directories in Phalcon

```

use Phalcon\Loader;

// Creates the autoloader
$loader = new Loader();

// Register some directories
$loader->registerDirs(
    [
        "library/MyComponent/",
        "library/OtherComponent/Other/",
        "vendor/example/adapters/",
        "vendor/example/",
    ]
);

// Register autoloader
$loader->register();

```

### Registering classes in Phalcon

```

use Phalcon\Loader;

// Creates the autoloader
$loader = new Loader();

// Register some classes
$loader->registerClasses(
    [
        "Some"          => "library/OtherComponent/Other/Some.php",
        "Example\Base" => "vendor/example/adapters/Example/BaseClass.php",
    ]
);

// Register autoloader
$loader->register();

```

## Q6. [Explain Routing in Phalcon?](#)

Routing in Phalcon Framework is managed by its Router component. Router components allow you to define routes that are mapped to controllers or handlers that should receive the request from your application. A router simply parses a URI to determine this information. The router has two modes: MVC mode and match-only mode. The first mode is ideal for working with MVC applications.

## Q7. [How can you add validations in Phalcon?](#)

Phalcon uses an independent module “Phalcon\Validation” for validating users requests. This component can be used to implement validation rules on data objects that do not belong to a model or collection. The loosely-coupled design of this component allows you to create your own validators along with the ones provided by the framework.

The following example shows basic usage of validations in Phalcon :

```

<?php

use

```

```

Phalcon\Forms\Element\Text;
use Phalcon\Validation\Validator\PresenceOf;
use Phalcon\Validation\Validator\StringLength;

$name = new Text(
    "name"
);

$name->addValidator(
    new PresenceOf(
        [
            "message" => "The name is required",
        ]
    )
);

$name->addValidator(
    new StringLength(
        [
            "min"           => 10,
            "messageMinimum" => "The name is too short",
        ]
    )
);

$form->add($name);

```

## **Q8. Explain PHQL In Phalcon?**

PHQL stands for **Phalcon Query Language**, PHQL is a high-level, object-oriented SQL dialect that allows writing queries using a standardized SQL-like language. PHQL is implemented as a parser (written in C) that translates syntax in that of the target RDBMS.

## **Q9. List some database related functions in Phalcon?**

Few database related functions in Phalcon are find(), findFirst(), query(), findFirstBy(), count(), save(), delete()

## **Q10. How to read, write and delete sessions in Phalcon?**

Phalcon session component provides object-oriented wrappers to access session data. Below is usage guide to Read, Write and Delete sessions in Phalcon

### **Writing or Creating session in Phalcon**

```
$this->session->set("user-name", "Michael");
```

### **Reading or Retrieving session in Phalcon**

```
$this->session->get("user-name");
```

## Deleting or Removing session in Phalcon

```
$this->session->remove( "user-name" );
```

## Q11. List Database engines supported by Phalcon?

Phalcon uses PDO\_ to connect to databases. The following database engines are supported:

- Phalcon\Db\Adapter\Pdo\Mysql : Mysql
- Phalcon\Db\Adapter\Pdo\Postgresql : Postgresql
- Phalcon\Db\Adapter\Pdo\Sqlite : SQLite

## Q12. How to increase CSRF timeout in Phalcon?

Phalcon CSRF tokens use sessions, so if you increase your session timeout it will automatically increase the token time as well.

## Q13. Explain MVC in Phalcon?

**Phalcon** offers the object-oriented classes, necessary to implement the Model, View, Controller architecture (often referred to as MVC) in your application. This design pattern is widely used by other web frameworks and desktop applications.

MVC benefits include:

- Isolation of business logic from the user interface and the database layer.
- Making it clear where different types of code belong for easier maintenance.

If you decide to use MVC, every request to your application resources will be managed by the MVC architecture. Phalcon classes are written in C language, offering a high-performance approach to this pattern in a PHP based application.

## Q14. What is Zephir in Phalcon?

**Zephir** – Ze(nd Engine) Ph(p) I(nt)r(mediate) – is a high-level language that eases the creation and maintainability of extensions for PHP. Zephir extensions are exported to C code that can be compiled and optimized by major C compilers such as gcc/clang/vc++. Functionality is exposed to the PHP language

source (<https://github.com/phalcon/zephir>)

## Q15. How to pass data from the controller to view in Phalcon?

The setVar() method is used to pass data from the controller to view template in Phalcon.

**Usage:-** `$this->view->setVar("username", $user->username);`

## Q16. [Which template engine Phalcon use?](#)

Phalcon uses a powerful and fast templating engine called Volt.

**Volt** is an ultra-fast and designer-friendly templating language written in C for PHP. It provides you a set of helpers to write views in an easy way. Volt is highly integrated with other components of Phalcon, just as you can use it as a stand-alone component in your applications.

Also, Read Best [Laravel 5 interview questions](#)

## Q17. [How can you inject Services into a Volt Template?](#)

If a service container (DI) is available for Volt, you can use the services by only accessing the name of the service in the template:

```
{# Inject the 'flash' service #}
<div id="messages">{{ flash.output() }}</div>

{# Inject the 'security' service #}
```

## Q18. [Explain ODM in Phalcon?](#)

**ODM (Object-Document Mapper)** offers a CRUD functionality, events, validations among other services in Phalcon.

## Q19. [Explain Single or Multi-Module Applications in Phalcon?](#)

**Single Module Application:** Single MVC applications consist of one module only. Namespaces can be used but are not necessary.

**Multi-Module Application:** A multi-module application uses the same document root for more than one module.

## Q20. [List various type of Application Events in Phalcon?](#)

Below is the list of Application Events in Phalcon.

<b>Event Name</b>	<b>Triggered</b>
boot	Executed when the application handles its first request
beforeStartModule	Before initializing a module, only when modules are registered
afterStartModule	After initializing a module, only when modules are registered
beforeHandleRequest	Before executing the dispatch loop
afterHandleRequest	After executing the dispatch loop

Please Visit [OnlineInterviewquestions.com](https://www.onlineinterviewquestions.com) to download more pdfs