**Manzar Ali**

**BSCS 2**

**00765**

**Lab 6**

Instructions:

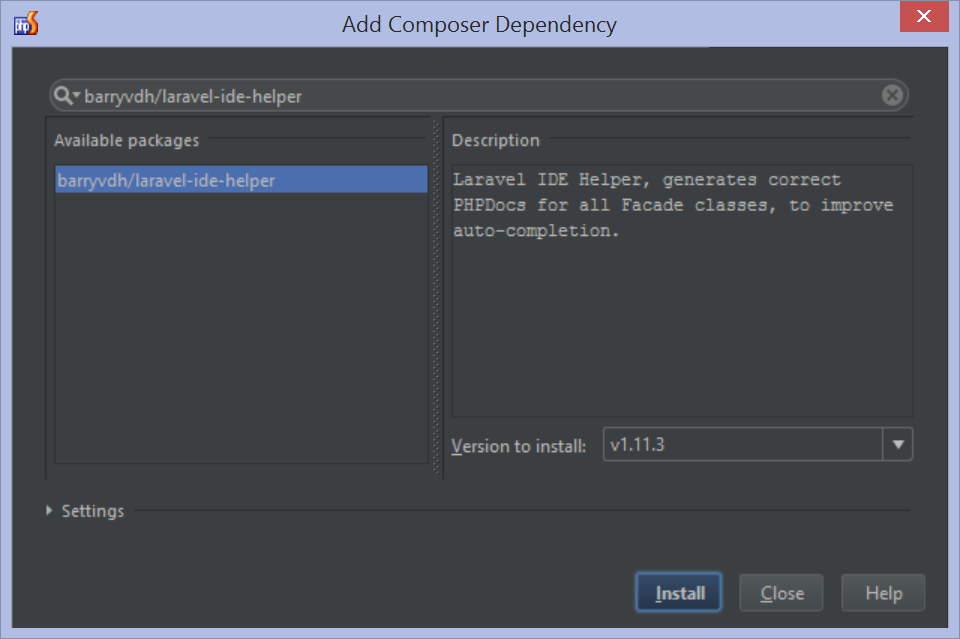
I followed and test the application through following instructions using online tutorials. I made the configuration in phpstorm IDE.

**Ensuring Composer is initialized**

One thing we want to make sure beforehand is that composer is initialized and configured in phpstorm .

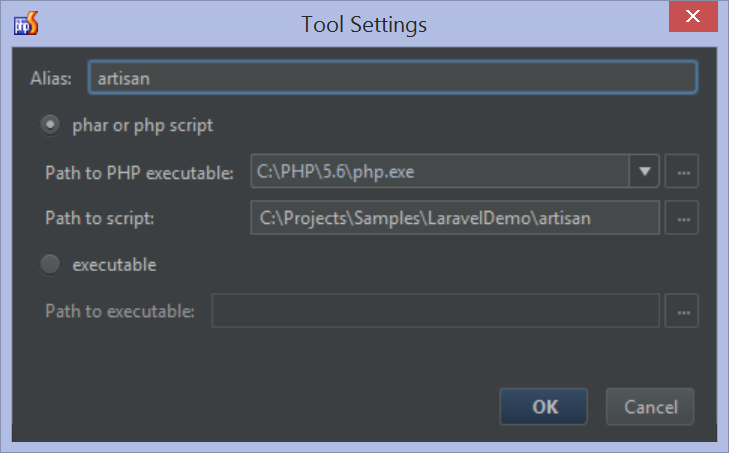
### Installing the Laravel IDE Helper

Once Composer is available for use in our project, we can install the laravel ide helper into our project.

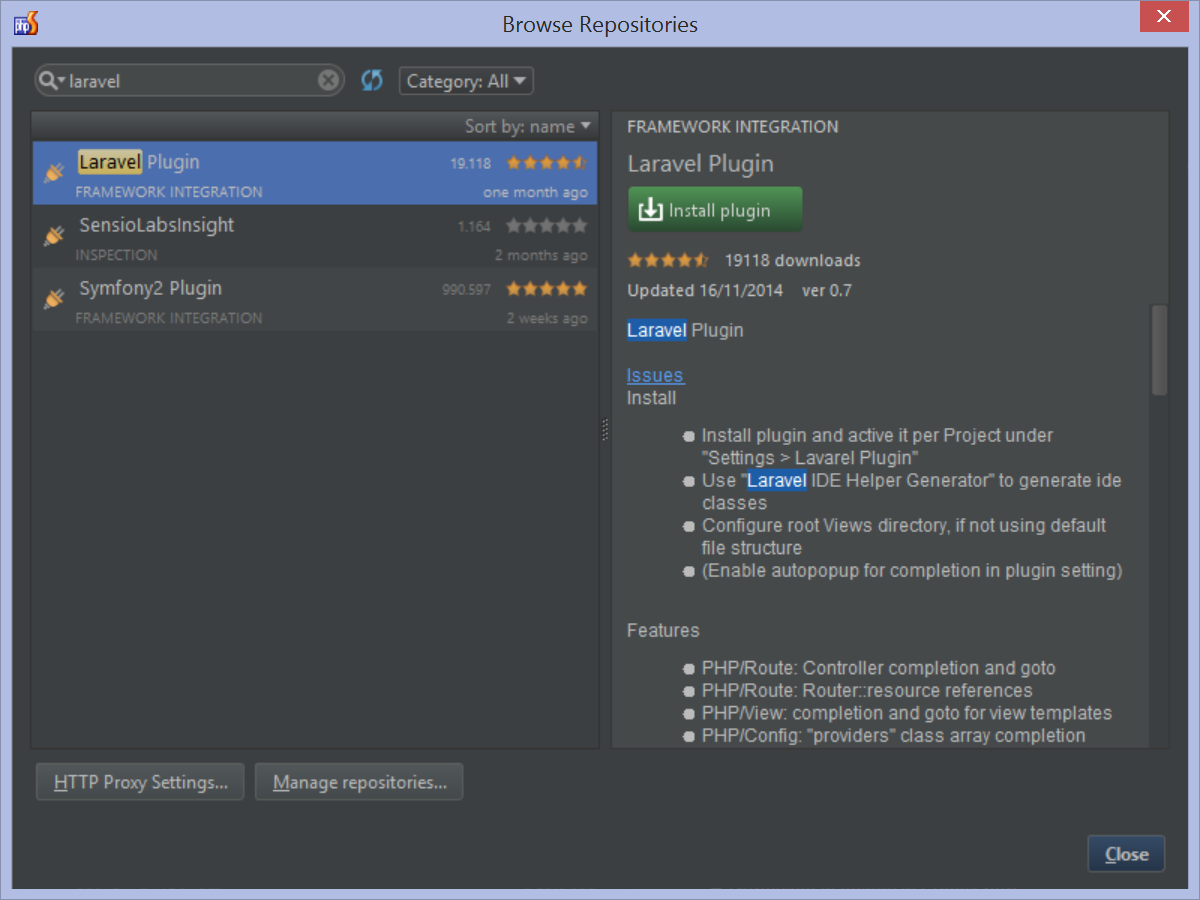


### Generating the PHPDoc Helper File using Artisan

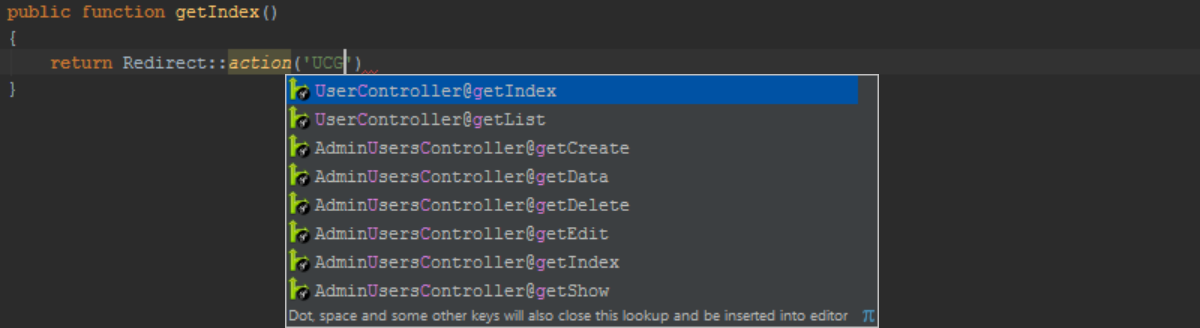
After installing the Laravel IDE Helper, we can use artisan to generate a PHPDoc helper file that PhpStorm and the Laravel plugin will use to provide code completion and navigation.

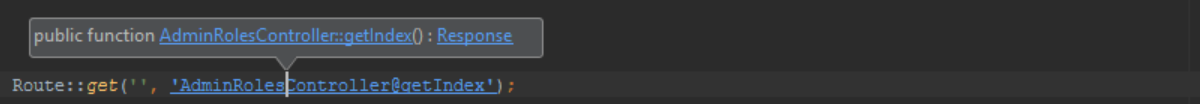


### Installing and enable the Laravel Plugin



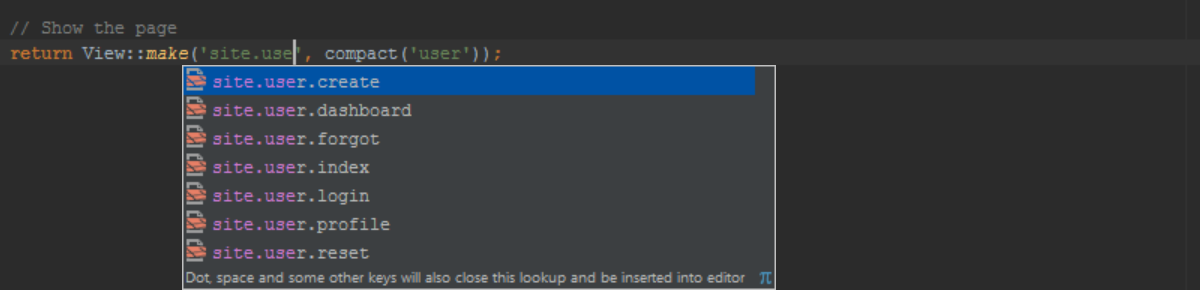
### Code Completion and Navigation for Controllers and Routes





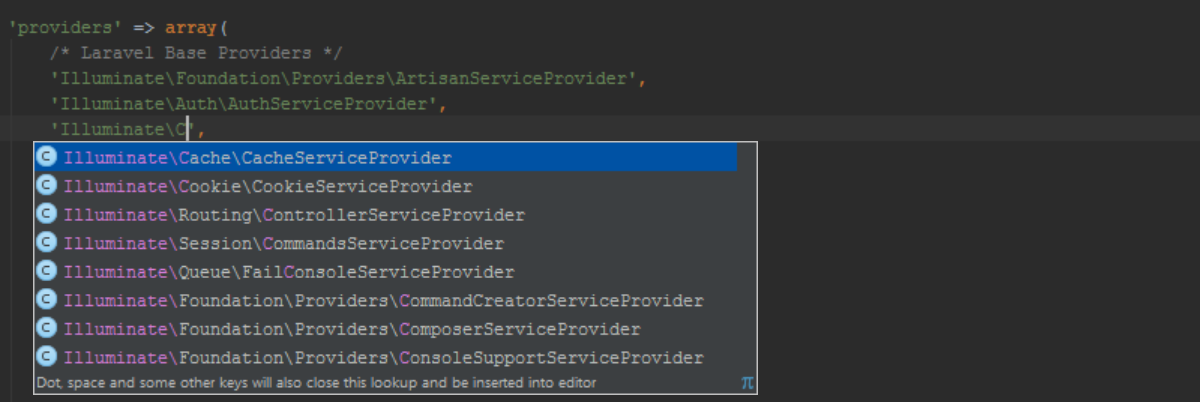
### Code Completion and Navigation for Views

Using the View facade, we can reference a Blade template (or view). The Laravel plugin provides completion for view names when using this facade:



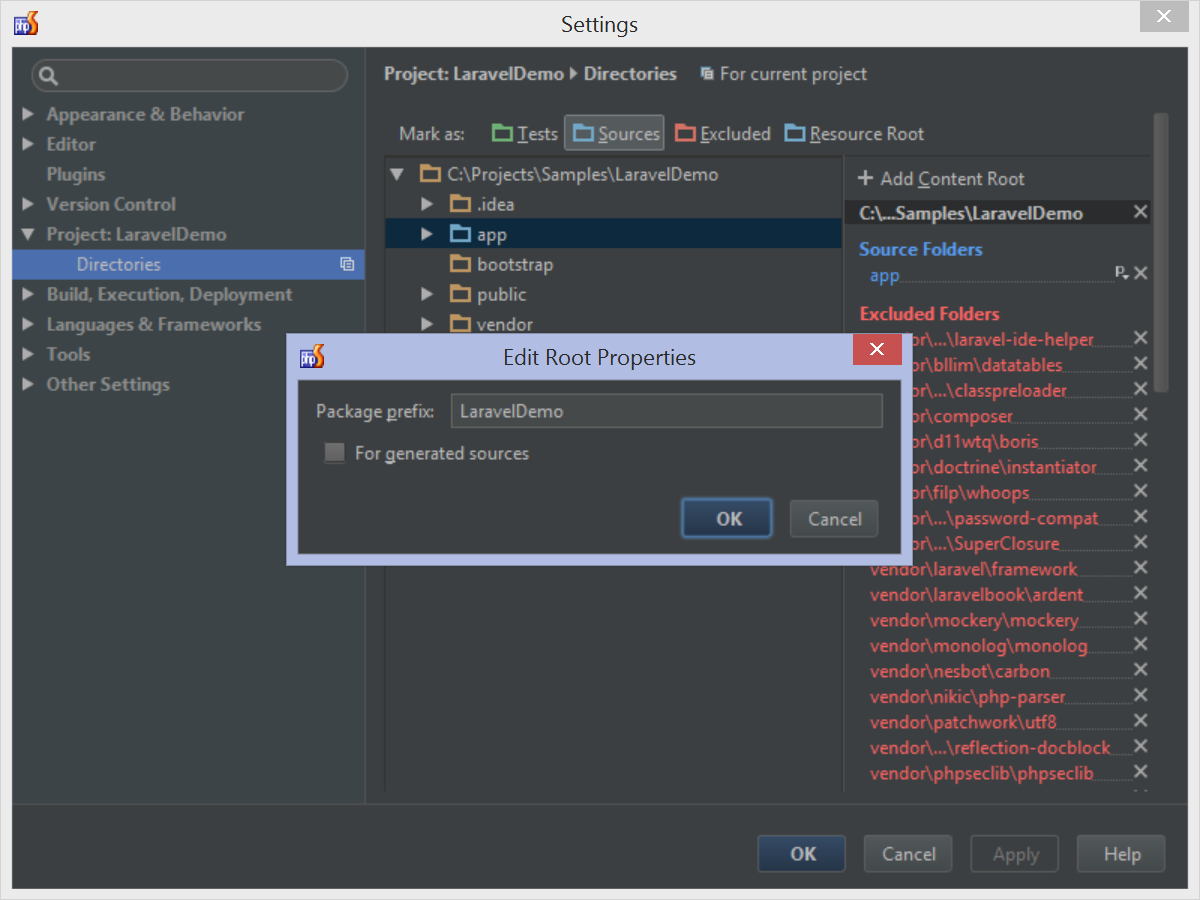
### Code Completion and Navigation for Configuration and Services

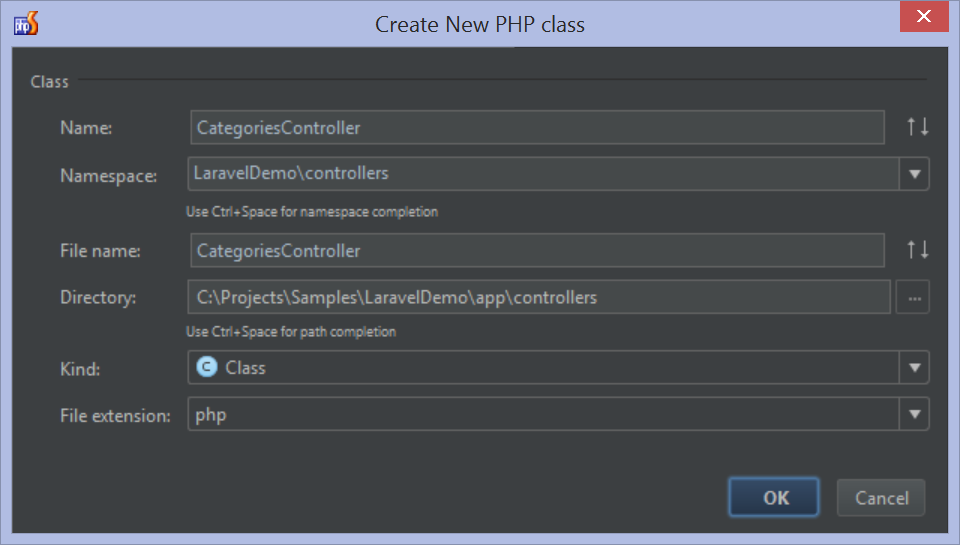
When working with Laravel configuration using the Configuration facade, we get code completion for the various keys that are defined in our application's settings.



### Automatic PSR-4 Name spacing

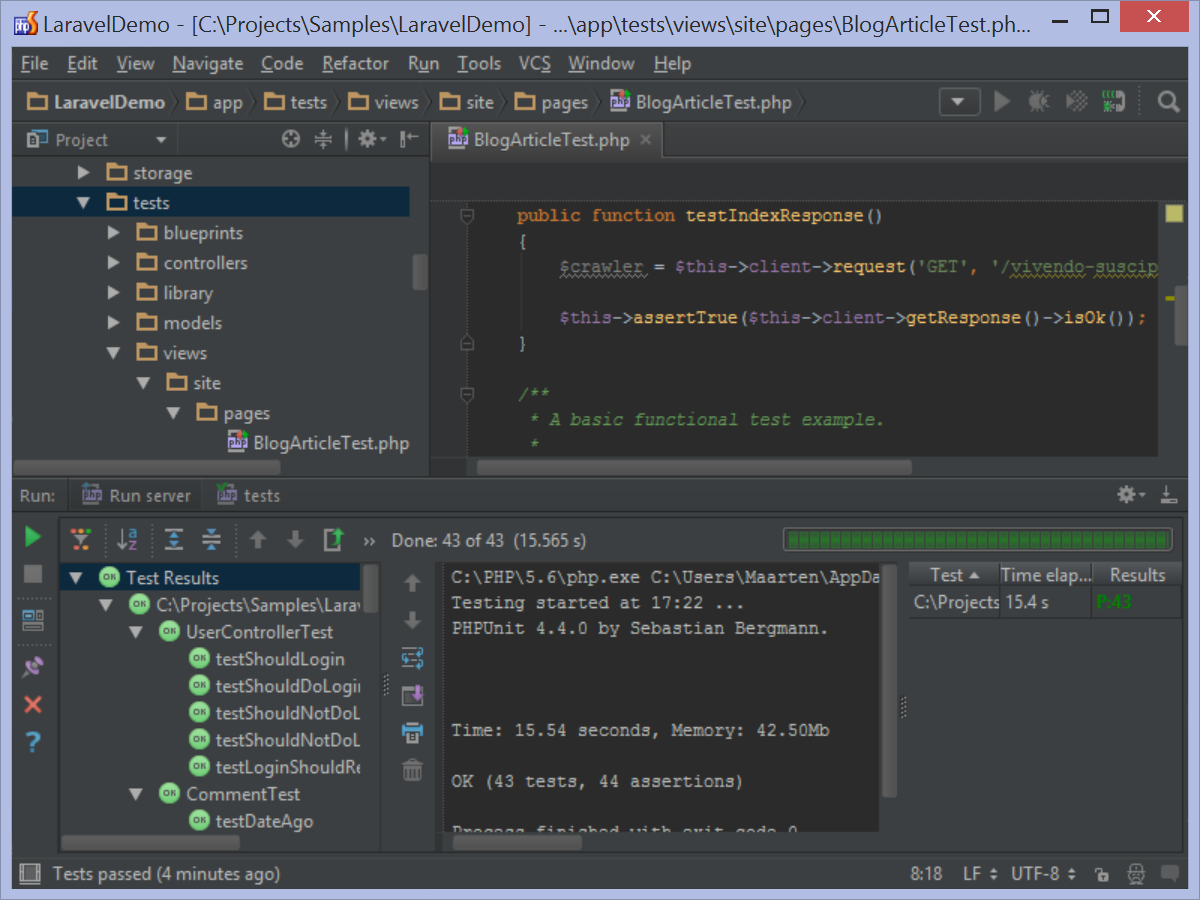
Through the project settings, we can configure the default namespace for various directories in our application, for example for the app directory. Once configured, PhpStorm will follow the PSR-4 standard to automatically namespace new classes created in our application.





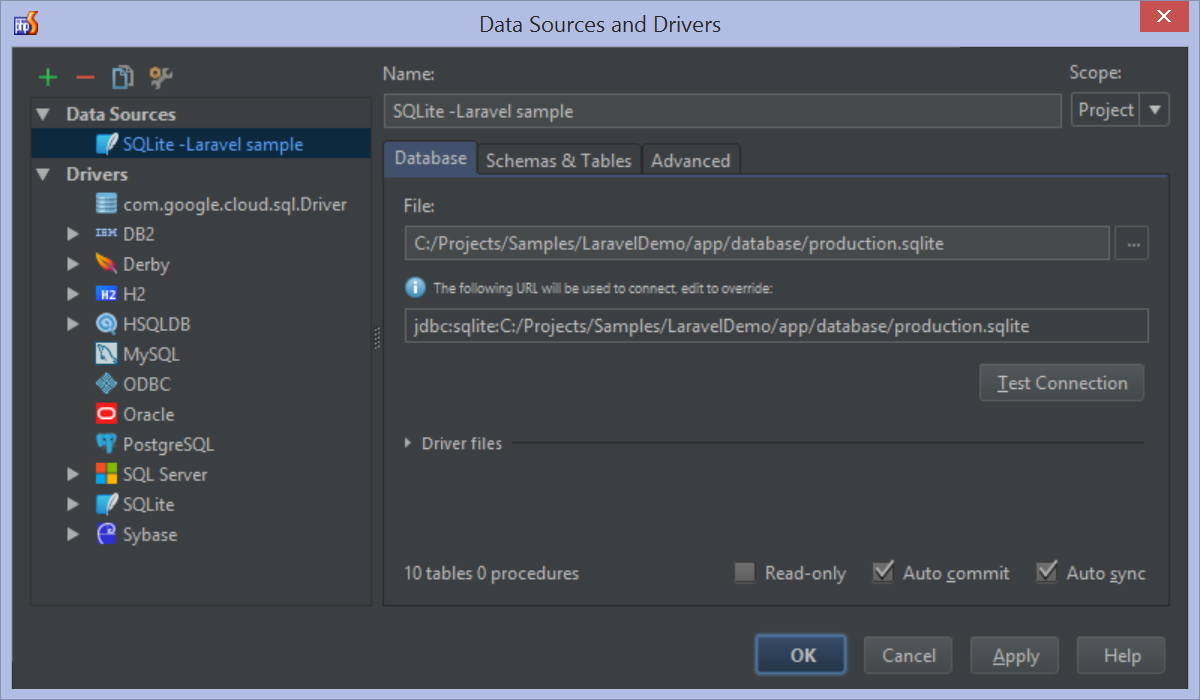
## Laravel Unit Testing with PhpStorm

With unit testing, we can verify parts of our source code are working as expected. After we've changed our code or performed a refactoring, unit tests can tell us if the changes we did break existing functionality or not. Only when all the tests are "green" (all tests pass) can we be sure that we're not breaking the functionality of our code.



## Database Support in PhpStorm

Laravel projects are typically backed by a database, which we can manage from within PhpStorm.



**Testing Final sample Application**:

On browser:



END…….