

LARAVEL - REQUEST

Retrieving the Request URI

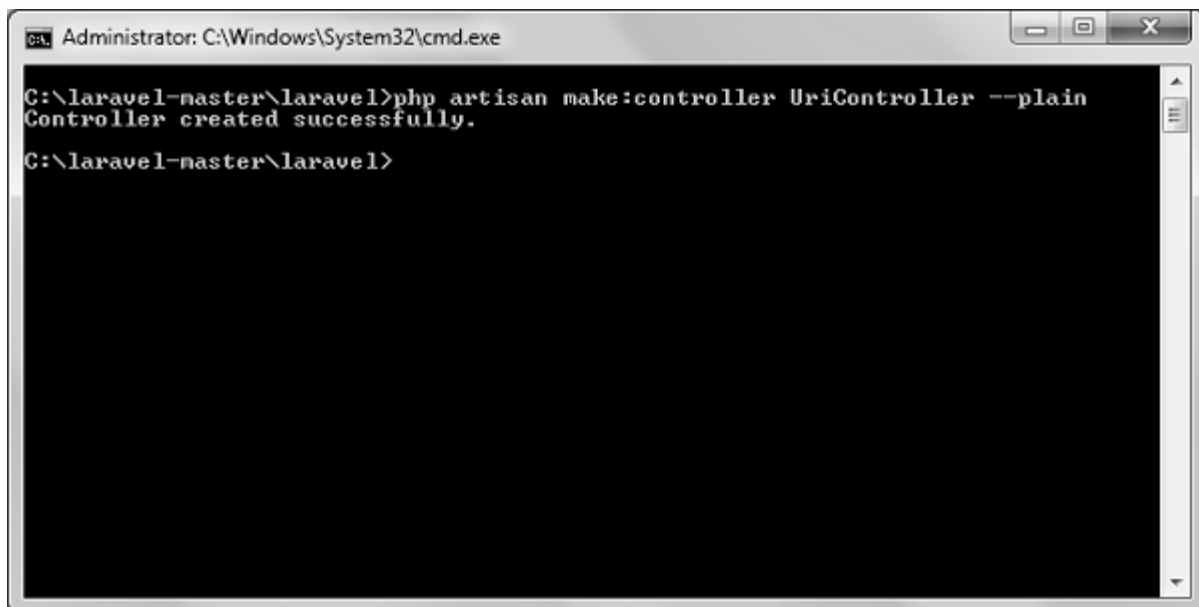
The “**path**” method is used to retrieve the requested URI. The “**is**” method is used to retrieve the requested URI which matches the particular pattern specified in the argument of the method. To get the full URL, we can use the “**url**” method.

Example

Step 1 – Execute the below command to create a new controller called **UriController**.

```
php artisan make:controller UriController --plain
```

Step 2 – After successful execution of the URL, you will receive the following output –



```
Administrator: C:\Windows\System32\cmd.exe

C:\laravel-master\laravel>php artisan make:controller UriController --plain
Controller created successfully.

C:\laravel-master\laravel>
```

Step 3 – After creating a controller, add the following code in that file.

app/Http/Controllers/UriController.php

```
<?php
namespace App\Http\Controllers;
use Illuminate\Http\Request;
use App\Http\Requests;
use App\Http\Controllers\Controller;

class UriController extends Controller {

    public function index(Request $request){
        // Usage of path method
        $path = $request->path();
        echo 'Path Method: '.$path;
        echo '<br>';

        // Usage of is method
        $pattern = $request->is('foo/*');
        echo 'is Method: '.$pattern;
    }
}
```

```
echo '<br>';  
  
// Usage of url method  
$url = $request->url();  
echo 'URL method: '.$url;  
}  
}
```

Step 4 – Add the following line in the **app/Http/route.php** file.


app/Http/route.php

```
Route::get('/foo/bar', 'UriController@index');
```

Step 5 – Visit the following URL.

http://localhost:8000/foo/bar

Step 6 – The output will appear as shown in the following image.



```
Path Method: foo/bar  
is Method: 1  
URL method: http://localhost:8000/foo/bar
```

Retrieving Input

The input values can be easily retrieved in Laravel. No matter what method was used “**get**” or “**post**”, the Laravel method will retrieve input values for both the methods the same way. There are two ways we can retrieve the input values.

- Using the input() method
-
- Using the properties of Request instance

Using the input() method

The input() method takes one argument, the name of the field in form. For example, if the form contains username field then we can access it by the following way.

```
$name = $request->input('username');
```

Using the properties of Request instance

Like the input() method, we can get the username property directly from the request instance.

```
$request->username
```

Example

Step 1 – Create a Registration form, where user can register himself and store the form at

```
<html>

<head>
  <title>Form Example</title>
</head>

<body>
  <form action = "/user/register" method = "post">
    <input type = "hidden" name = "_token" value = "<?php echo csrf_token()
?>">

    <table>
      <tr>
        <td>Name</td>
        <td><input type = "text" name = "name" /></td>
      </tr>

      <tr>
        <td>Username</td>
        <td><input type = "text" name = "username" /></td>
      </tr>

      <tr>
        <td>Password</td>
        <td><input type = "text" name = "password" /></td>
      </tr>

      <tr>
        <td colspan = "2" align = "center">
          <input type = "submit" value = "Register" />
        </td>
      </tr>
    </table>

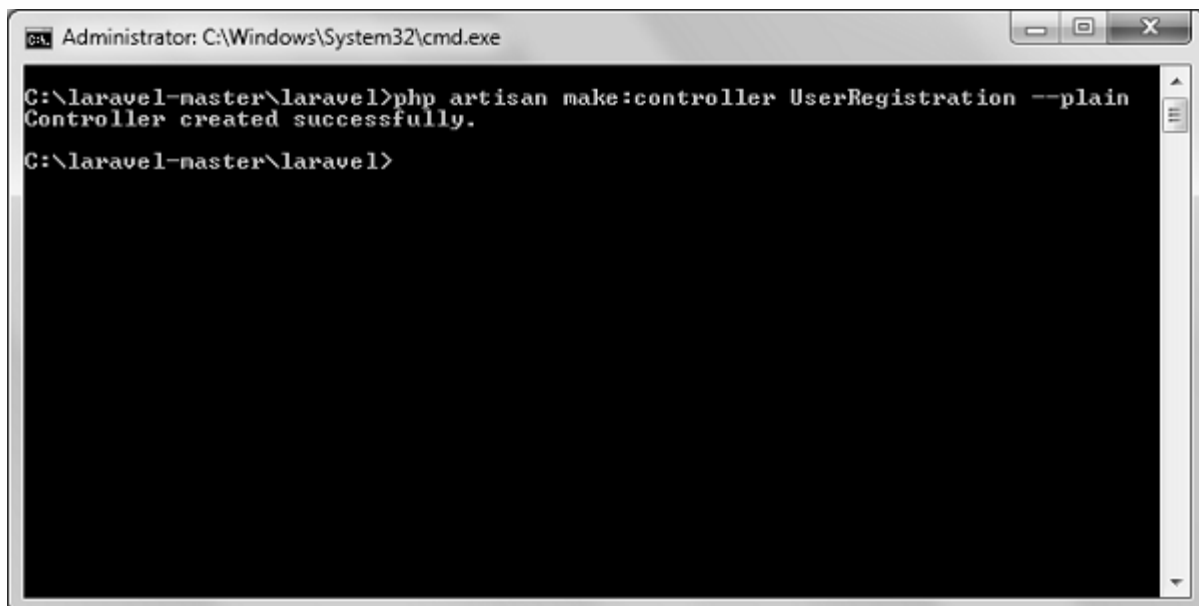
  </form>

</body>
</html>
```

Step 2 – Execute the below command to create a **UserRegistration** controller.

```
php artisan make:controller UserRegistration --plain
```

Step 3 – After successful execution, you will receive the following output –



The screenshot shows a Windows command prompt window titled "Administrator: C:\Windows\System32\cmd.exe". The prompt is at "C:\laravel-master\laravel>". The user has entered the command "php artisan make:controller UserRegistration --plain". The output of the command is "Controller created successfully." followed by a new prompt "C:\laravel-master\laravel>".

Step 4 – Copy the following code in **app/Http/Controllers/UserRegistration.php** controller.

app/Http/Controllers/UserRegistration.php

```
<?php
namespace App\Http\Controllers;
use Illuminate\Http\Request;
use App\Http\Requests;
use App\Http\Controllers\Controller;

class UserRegistration extends Controller {
    public function postRegister(Request $request){
        //Retrieve the name input field
        $name = $request->input('name');
        echo 'Name: '.$name;
        echo '<br>';

        //Retrieve the username input field
        $username = $request->username;
        echo 'Username: '.$username;
        echo '<br>';

        //Retrieve the password input field
        $password = $request->password;
        echo 'Password: '.$password;
    }
}
```

Step 5 – Add the following line in **app/Http/routes.php** file.

app/Http/routes.php

```
Route::get('/register',function(){
    return view('register');
});
Route::post('/user/register',array('uses'=>'UserRegistration@postRegister'));
```

Step 6 – Visit the following URL and you will see the registration form as shown in the below figure. Type the registration details and click Register and you will see on the second page that we have retrieved and displayed the user registration details.

http://localhost:8000/register

Step 7 – The output will look something like as shown in below the following images.

The diagram illustrates the registration process. On the left, a registration form is shown with three input fields: 'Name' containing 'Virat', 'Username' containing 'virat.gandhi', and 'Password' containing 'asdadf'. Below these fields is a 'Register' button. A large blue arrow points from the form to the right, where the output is displayed. The output shows the retrieved details: 'Name: Virat', 'Username: virat.gandhi', and 'Password: asdadf'.