

Laravel Routing

All Laravel routes are defined in the app/Http/routes.php file, which is automatically loaded by the framework. The most basic Laravel routes simply accept a URI and a Closure, providing a very simple and expressive method of defining routes:

```
Route::get('foo', function () {      return 'Hello World'; });
```

Required Parameters

```
Route::get('user/{id}', function ($id) {      return 'User '.$id; });
```

```
Route::get(
    'posts/{post}/comments/{comment}', function ($postId,
    $commentId) {      // });
```

Optional Parameters

Occasionally you may need to specify a route parameter, but make the presence of that route parameter optional. You may do so by placing a ? mark after the parameter name. Make sure to give the route's corresponding variable a default value:

```
Route::get('user/{name?}', function ($name = null) {      return
$name; });      Route::get('user/{name?}', function ($name = 'John') {
    return $name; });
```

Regular Expression Constraints

You may constrain the format of your route parameters using the where method on a route instance. The where method accepts the name of the parameter and a regular expression defining how the parameter should be constrained:

```
Route::get('user/{name}', function ($name) {      // }) ->where(
'name', '[A-Za-z]+');      Route::get('user/{id}', function ($id) {
```

```
        // }) ->where('id', '[0-9]+');    Route::get(
'user/{id}/{name}', function ($id, $name) {        // }) ->where([
'id' => '[0-9]+', 'name' => '[a-z]+']);
```

Named Routes

Named routes allow the convenient generation of URLs or redirects for specific routes. You may specify a name for a route using the `as` array key when defining the route:

```
Route::get('user/profile', ['as' => 'profile', function () {
    // }]);
```

You may also specify route names for controller actions:

```
Route::get('user/profile', [        'as' => 'profile', 'uses' =>
'UserController@showProfile'    ]);
```

Route Groups

Route groups allow you to share route attributes, such as middleware or namespaces, across a large number of routes without needing to define those attributes on each individual route. Shared attributes are specified in an array format as the first parameter to the `Route::group` method.

To learn more about route groups, we'll walk through several common use-cases for the feature.

Middleware

To assign middleware to all routes within a group, you may use the `middleware` key in the group attribute array. Middleware will be executed in the order you define this array:

```
Route::group(['middleware' => 'auth'], function () {        Route::get(
'/', function () {
    // Uses Auth Middleware    });        Route::get(
'user/profile', function () {
    // Uses Auth Middleware    });    });
```