

## 3. Маршруты с параметрами

### Обязательные параметры

---

Иногда разработчику может понадобиться **захватить сегменты URI** определенного маршрута. Например, если необходимо захватить **id** пользователя **из URL**. Это можно сделать, определив следующий **параметр маршрута**:

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

**Параметры маршрута всегда заключаются в фигурные скобки** и должны состоять из буквенных символов. Параметры маршрута не могут содержать символ `-`.

Пример определения маршрута с параметром:

Файл **ex\_1/web.php**. Пример маршрута с параметром

```
// регистрируем маршрут http://host/music/team/{параметр}  
// пример: http://host/music/team/3  
  
Route::get('/music/team/{idTeam}', function ($idTeam){  
  
    echo "<h1>Страница: Вывод информации о группе</h1>";  
  
    echo "<h3>Идентификатор группы: $idTeam</h3>";  
  
});
```

### Маршруты с несколькими параметрами

---

Разрабатывая приложение, вы можете создавать маршруты, определяя при этом **сколько угодно параметров**:

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

    // ...

});
```

Файл **ex\_2/web.php**. Маршрут с двумя параметрами

```
// регистрируем маршрут:
// http://host/music/team/{параметр1}/album/{параметр2}
// пример: http://host/music/team/1/album/10
Route::get ('/music/team/{idTeam}/album/{idAlbum}', function
($idTeam, $idAlbum){

    echo "<h1>Страница: Вывод информации об альбоме</h1>";

    echo "<h3>Идентификатор группы: $idTeam</h3>";

    echo "<h3>Идентификатор альбома: $idAlbum</h3>";

});
```

**Важно.** Никакой связи между именем параметра в **маршруте** и именем параметра в **функции** нет. Важен порядок следования.

```
Route::get('posts/{post}/comments/{comment}', function ($var1,
$var2) {

    echo "Пост: " . $var1 . "<br />";

    echo "Содержит комментарий: " . $var2;

});
```

И хоть пример ниже уже на практике демонстрирует отсутствие связи между параметрами маршрута и функции, делать так по понятным причинам не рекомендуется. Главный принцип написания кода - *упрощение восприятия программного кода человеком, минимизация нагрузки на память и зрение при чтении программ* (<https://ru.wikipedia.org/>).

Файл **ex\_3/web.php**. Порядок следования параметров маршрута

```
//регистрируем маршрут:

http://host/music/team/{параметр1}/album/{параметр2}/track/{параметр3}

//пример: http://host/music/team/3/album/7/track/15

Route::get
('/music/team/{idTeam}/album/{idAlbum}/track/{idTrack}',
function ($par3, $par1, $par2){

    echo "<h1>Страница: Вывод информации о треке</h1>";

    echo "<h3>Идентификатор группы: $par3</h3>";

    echo "<h3>Идентификатор альбома: $par1</h3>";

    echo "<h3>Идентификатор трека: $par2</h3>";

});
```

## Необязательные параметры

---

Иногда необходимо указать параметр маршрута, но при этом сделать его наличие **необязательным**. Это можно сделать, **поместив знак вопроса ? после названия параметра**.

Для параметра маршрута, отмеченного как **необязательный**, необходимо задать **значение по умолчанию**

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

    return $name;

});

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

    return $name;

});
```

## Методы маршрутизатора

---

Маршрутизатор позволяет регистрировать маршруты для любого **HTTP-запроса**:

```
Route::get($uri, callback);  
Route::post($uri, callback);  
Route::put($uri, callback);  
Route::patch($uri, callback);  
Route::delete($uri, callback);
```

Иногда необходимо зарегистрировать маршрут, который отвечает на HTTP-запросы нескольких типов. Это можно сделать методом **match()**.

```
Route::match(['get', 'post'], '/', function () {  
    // ...  
});
```

Или вы можете зарегистрировать маршрут, отвечающий на HTTP-запросы всех типов, с помощью метода **any()**:

```
Route::any('foo', function () {  
    // ...  
});
```

## Файлы раздаточного материала

---

**ex\_1/** - примеры маршрутов с одним параметром.

**ex\_2/** - примеры маршрутов с двумя параметрами.

**ex\_3/** - примеры маршрутов с тремя параметрами.

**ex\_4/** - примеры маршрутов с разными типами параметров.