

Роутер, контроллеры, ParamConverter, views

№ урока: 2 **Курс:** Веб разработка на PHP Symfony

Средства обучения: Командная строка, любой современный веб-браузер, Eclipse PHP Development Tools

Обзор, цель и назначение урока

После прохождения данного урока вы усвоите то, как именно можно использовать маршрутизацию (роутинг), контроллеры, шаблоны и ParamConverter в Symfony приложении. Кроме этого, будет немало практики по маршрутизации и созданию контроллеров, а также конфигурации самого приложения.

Изучив материал данного занятия, учащийся сможет:

- Понять, как создавать и использовать в общих чертах собственные контроллеры в symfony приложении.
- Изучить и понять, как работает маршрутизация в Symfony приложение, кроме этого, как её настраивать и создавать вообще собственные url адреса для собственных контроллеров.
- Пользоваться маршрутизацией в symfony framework, также к своим маршрутам параметры - т.е. делать их более динамичными как у всех современных приложениях.
- Понимать, как можно генерировать url адреса в контроллерах без создания отдельной маршрутизации.
- Понимать, как создавать Bundle как отдельные плагины/мини-приложения в самом приложении, которые можно использовать для разных сервисных целей.
- Использовать ParamConverter для облегчения разработки функций для контроллеров и получения данных в них благодаря конвертации запросов в объекты.

Содержание урока

1. Про контроллеры и роутинг
2. Annotations vs Yaml
3. Параметры url и их валидация
4. Генерация Url
5. Установка ParamConverter, понятие Bundle
6. Использование ParamConverter

Резюме

- Контроллеры - представляют из себя отдельные классы, в которых можно создавать отдельные функции для прописывания логики целого приложения, а также маршрутов (url адресов для приложения). Для того чтобы создать новый контроллер можно воспользоваться командной строкой приложения, но перед этим нужно установить пакет для генерации нужных объектов в приложении, сделать это можно с помощью команды пакетного менеджера Composer:
`composer require symfony/maker-bundle`
- Установив данный пакет, можно приступить к генерации нового контроллера с помощью консоли приложения. Для этого нужно воспользоваться следующей командой:
`php bin/console generate:controller NewController`

Где NewController — это название контроллера, который нужно создать. Только что сгенерированный контроллер будет создан в папке src/Controller. В контроллере есть возможность создавать функции и сразу же к ним прописывать url маршруты. Маршруты могут принимать любые типы HTTP запросов (POST, GET, PUT, DELETE). Роутинг, или же маршрутизация – это то, на чем строится вся логика url адресов в приложении, то есть создание собственных url адресов для функций контроллеров. Прописывать маршруты можно с помощью YAML в файле config/routes.yaml - при этом нужно будет обязательно указать какой контроллер и название маршрута.

- Также можно создавать маршруты в самих функциях контроллеров с помощью комментариев от annotations. Пример такого комментария, который прописывается прямо перед созданием новой функции в коде:

```
/**
 * @Route("/home")
 */
```

- Важно также заметить, что для того, чтобы использовать их в приложении, нужно установить сам пакет annotations - сделать это можно с помощью Composer:

```
composer require doctrine/annotations
```

- После установки этого пакета можно будет полноценно использовать Annotations в своем приложении. Настроить откуда Annotations должно брать url маршруты можно в файле: config/routes/annotations.yaml. Можно сразу же заметить, что благодаря использованию Annotations намного проще создавать url адреса в своем приложении. Если же смотреть на маршрутизацию с помощью YAML - здесь уже нужно по отдельности прописывать маршруты в файле конфигурации config/routes.yaml.
- Если есть потребность в том, чтобы передать какие-то данные через url адрес, то здесь можно воспользоваться созданием url параметров к маршрутам при прописывании маршрутов к функциям контроллеров. Пример создания маршрута, который принимает в своем адресе параметр name:

```
/**
 * @Route("/posts/{id}")
 */
```

- После этого в данном url можно будет указывать какие-то данные для передачи, например номер id: /posts/34. Если нужно сделать то же самое, только уже для YAML маршрутизации - то это можно сделать следующим путем:

```
posts_show:
    path: /posts/{id}
    controller: App\Controller\PostsController:show
```

- Валидация самих же url адресов представляет из себя использование регулярных выражений - для того, чтобы можно было задать правила того, как должен выглядеть сам url адрес, указанный пользователем в адресной строке. Пример:

```
/**
 * @Route("/posts/{id}", requirements={"id"="\d+"})
 */
```

- Если уже нужно сгенерировать url адрес в функции контроллера без создания дополнительной какой-то маршрутизации, то здесь можно воспользоваться функциями генерации url адресов в Symfony:

```
$lists_show_page = $this->generateUrl('show-lists');
```

- Важно заметить, что данный код по генерации нужно прописывать внутри функций контроллера. По примеру будет сгенерирован url с адресом show-lists.
- Bundle - представляет из себя плагин/мини-приложения которые можно подключать в самом приложении. К примеру, в новом созданном приложении уже есть подключённые Bundl'ы от разработчиков Symfony (FrameworkBundle, SecurityBundle, TwigBundle и т.д.). Посмотреть конфигурацию Bundl'ов можно в файле config/bundles.php. Для создания нового Bundle в первую очередь нужно создать папку с названием по адресу

src/Acme/MyBundle. Acme – это может быть название разработчика либо компании. Следующим шагом уже будет создать файл с названием Bundle'a.

- После создания самого Bundle'a нужно будет его подключить в файле конфигурации config/bundles.php. Кроме того, чтобы Bundle'ы представляли собой полноценные плагины, которые можно подключать к приложению, они также являются небольшими мини-приложениями, которые также могут задавать в приложении свои маршруты и контроллеры. Вот структура директорий в Bundle'e:
 - Controller/ - контроллеры в Bundle'e
 - DependencyInjection/ - DependencyInjection классы хранятся здесь
 - Resources/config/ - файлы конфигурации (к примеру, для маршрутизации)
 - Resources/views/ - шаблоны и внешний вид в общем
 - Resources/public/ - публичные файлы (CSS, JS, изображения и т.д.)
 - Tests/ - модульные тесты
- ParamConverter – инструмент, который можно использовать в аннотациях для того, чтобы конвертировать параметры запроса в нужные объекты. В общих чертах уже понятно, что благодаря ParamConverter мы конвертируем аргументы запроса в объект нужного нам класса. Конвертацию можно сделать автоматической, для этого нужно сначала её включить в файле config/packages/sensio_framework_extra.yaml.

Закрепление материала

- Описать в нескольких словах, что из себя вообще представляют контроллеры в приложении и для чего их нужно использовать.
- Что нужно использовать для того, чтобы создать новый контроллер в приложении? Какая команда и какой пакет нужно перед этим установить, чтобы в дальнейшем использовать команду для создания контроллеров?
- Какие есть типы маршрутизации в Symfony Framework?
- Как можно использовать маршрутизацию с помощью файлов конфигурации YAML?
- Что такое Annotations в Symfony? Как их можно использовать?
- Приведите пример созданного маршрута, который в своем url будет принимать один аргумент и выводить его в шаблон в контроллере.
- Что из себя представляет генерация url? С помощью чего можно сгенерировать новый url внутри функции?
- Приведите пример валидации url в функции контроллера с помощью регулярных выражений.
- Как можно описать понятие Bundle? Что нужно, чтобы создать свой Bundle и какая структура внутри директории собственного Bundle?
- Для чего можно использовать ParamConverter? Как можно сделать автоматическое конвертирование запросов в функциях контроллеров приложения с помощью данного инструмента?

Дополнительное задание

Задание

Создать свой первый контроллер BlogController и добавить в него функцию blogPage в которой должен будет задан маршрут /blog и выводить надпись "Hello world!" на страницу. После этого добавить вывод ссылки, которая была сгенерирована на любой другой маршрут.

Самостоятельная деятельность учащегося

Задание 1

Создать контроллер NewController, который будет иметь три функции с тремя разными маршрутами. Первая будет иметь маршрут - /foo, вторая будет иметь маршрут - /bar/test, и третья будет иметь маршрут /post/{id} - и здесь понадобится выводить в шаблон id поста, который достаётся через URL параметр.

Задание 2

Создать маршрут /blog через YAML и дать название эту маршруту main_blog. После этого в любом доступном контроллере сгенерировать ссылку на данный маршрут и вывести его в шаблон.

Задание 3

В маршруте /blog добавить URL параметр {id}, после чего, с помощью трёх функций для URL параметров достать такие данные как: название маршрута, его параметр id, и получить в конце все параметры маршрута.

Рекомендуемые ресурсы

Официальная документация Symfony Framework

<https://symfony.com/doc/current/index.html>

Документация symfony на русском языке

<https://symfony.com.ua/doc/current/index.html>