

Symfony Security

№ урока: 6 **Курс:** Веб разработка на PHP Symfony

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

Обзор, цель и назначение урока

В данном уроке рассматривается использование компонента Symfony Security. Этот компонент является важным как для реализации безопасности в приложении, так и для реализации авторизации/регистрации для пользователей. В уроке рассматривается основной функционал компонента и реализовывается регистрация, и авторизация для пользователей.

Изучив материал данного занятия, учащийся сможет:

- Научиться использовать User Provider для того, чтобы реализовать систему пользователей и авторизации в своем приложении.
- Благодаря Twig registration form пользователь сможет быстро реализовать систему регистрации для пользователей, используя уже готовые инструменты фреймворка.
- Использовать firewalls для того, чтобы задавать правила авторизации пользователей в приложении.
- Использовать систему ролей для пользователей, т.е. задавать то, какие есть роли в приложении, задавать с помощью них доступы для пользователей и также реализовывать иерархию ролей.

Содержание урока

1. Установка Security Bundle
2. User Provider
3. Авторизация
4. Firewalls
5. Security Voters
6. Роли
7. Иерархия ролей

Резюме

- Перед тем как начать использовать функционал компонента Security Bundle, нужно убедиться, что он установлен в приложении. Поэтому нужно воспользоваться командой в консоли:
`composer require symfony/security-bundle.`
- User Provider позволяет реализовать в приложении систему авторизации для пользователей. Для того чтобы начать его использовать, нужно с помощью консольной команды создать класс, который и будет отвечать за пользователей в систему. Для этого нужно ввести команду:
`php bin/console make:user`
- После того как класс будет создан, нужно также создать новую миграцию для базы данных и также запустить саму созданную миграцию.
- Приложение автоматически уже сконфигурируется под User Provider и система пользователей будет уже готова в БД.

- Для того чтобы реализовать авторизацию в приложении, нужно воспользоваться в консоли командой от maker-bundle. Это будет команда `php bin/console make:auth`
После запуска, генератор новой формы для авторизации спросит, что именно мы хотим создать и как назвать класс для авторизации. Поскольку нужно создать новую форму для авторизации, то в первом вопросе нужно выбрать цифру 1, а уже после этого указать имя класса для авторизатора — это может быть `LoginFormAuthenticator`.
- Чтобы создать уже форму регистрации, которая будет работать с User Provider, то нужно будет ввести команду `php bin/console make:registration-form`. После этого сразу же будет автоматически создан контроллер и шаблон для формы регистрации.
- `Firewalls` представляет из себя параметр в `config/packages/security.yaml` - который позволяет настраивать правила авторизации в приложении для пользователей. Также настраивает какие классы отвечают за безопасность при авторизации.
- В `Symfony` уже по умолчанию реализован функционал для системы ролей для пользователей. Название каждой роли в приложении имеет префикс `ROLE_`. Для того чтобы добавить новую роль в приложение - нужно перейти в файл `config/packages/security.yaml`, и в параметре `access_control` добавить новый объект с данными доступа для определенной роли. Например: `{ path: ^/admin, roles: ROLE_ADMIN }`. В данном примере доступ к маршрутам `/admin` будет доступен только тем пользователям, которые имеют роль `ROLE_ADMIN`.
- Для реализации иерархии ролей в приложении нужно перейти в файл `config/packages/security.yaml` к параметру `role_hierarchy`. Дальше нужно выбирать роль, которой можно будет давать доступы других ролей. Например, если прописать `ROLE_ADMIN: ROLE_MODERATOR`, то `ROLE_ADMIN` получит те же доступы, что и пользователи, которые имеют роль `ROLE_MODERATOR`. Можно также и массивом задать доступы ролей, которых получит пользователь, например: `ROLE_ADMIN: [ROLE_MODERATOR, ROLE_USER]`.

Закрепление материала

- Для чего нужен User Provider и с помощью какой команды можно создать класс, который будет отвечать за пользователей в приложении?
- Как можно реализовать авторизацию с помощью генерации с maker-bundle?
- Как можно реализовать регистрацию в приложении с помощью уже готового инструмента, который также генерирует контроллер и шаблон через maker-bundle?
- Для чего нужны `firewalls` в приложении?
- Как работают роли в приложении и как их можно использовать в пользователях?
- Как можно задавать иерархию ролей в приложении?

Дополнительное задание

Задание

Реализовать в приложении авторизацию и регистрацию с помощью уже готовых инструментов `maker-bundle`. После этого протестировать авторизацию и регистрацию. Дальше потребуется создать 2 пользователя, которые будут иметь роли `admin` и `guest`. Пользователь, который имеет роль `admin`, будет иметь доступы к маршрутам, которые имеют в себе `/admin`. А пользователь, который имеет в себе роль `guest`, будут иметь доступ только к тем маршрутам, которые имеют в себе такой маршрут как `/guest`. Для самих маршрутов можно создать отдельные функции в одном контроллере, либо же можно отдельно создать два контроллера которые будут иметь в себе функции, которые отвечают за данные маршруты. Данные функции могут выводить любой шаблон, главное реализовать логику доступов.

Самостоятельная деятельность учащегося

Задание 1

С помощью maker-bundle реализовать регистрацию и авторизацию в приложении. После этого добавить отдельную страницу с маршрутом /profile, которая будет иметь в себе вывод данных о пользователях. Все данные для поиска можно брать с помощью данных, которые находятся в сессии, либо делать поиск их через БД.

Задание 2

Создать фикстуру, которая будет хранить в себе демо таблицы для 4 пользователей приложения. После этого создать маршрут /users, который с помощью функции dump будет выводить массив данных всех пользователей из БД. Кроме этого, нужно также настроить чтобы доступ к данной странице имели только авторизированные пользователи и те, которые имеют роль ROLE_MODERATOR.

Задание 3

Сделать небольшую иерархию ролей с ролями ROLE_ADMIN, ROLE_MODERATOR, ROLE_AUTHOR, ROLE_USER. Настроить так, чтобы ROLE_ADMIN имел одновременно доступы всех других ролей, а роль ROLE_MODERATOR имел доступы только ROLE_AUTHOR и ROLE_USER.

Рекомендуемые ресурсы

Официальная документация Symfony

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

Документация Symfony на рус. языке (имеет в себе некоторые устаревшие материалы)

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