



Веб разработка на PHP Symfony

Symfony Security



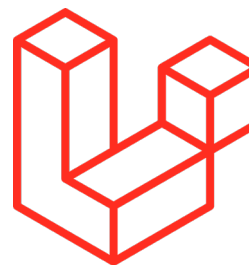
Веб разработка на PHP Symfony

Introduction



Демьян Костельный
Middle PHP Developer

 demian-kostelny-613b90151



Веб разработка на PHP Symfony

Тема урока

Symfony Security

Веб разработка на PHP Symfony

План урока

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

Установка Security Bundle

Веб разработка на PHP Symfony

Установка Security Bundle

Security Bundle представляет из себя полноценный bundle для Symfony, который предоставляет весь нужный функционал для реализации авторизации и регистрации в приложении, но и кроме этого множество других особенностей, которые могут сделать приложение безопасным. Чтобы пользоваться Security Bundle в своем приложении, в первую очередь нужно его установить через Composer:

```
composer require symfony/security-bundle
```

Веб разработка на PHP Symfony

Установка Security Bundle

После установки у нас появится возможность полностью использовать Security Bundle в коде своего приложения.

Веб разработка на PHP Symfony

User Provider

Веб разработка на PHP Symfony

User Provider

Как было сказано ранее, Security Bundle предоставляет возможности для создания полноценной системы пользователей в приложении. Перед тем как мы будем делать систему регистрации и авторизации, нам нужно создать класс, который будет отвечать за пользователей через командную строку. Для этого нужно ввести следующую команду:

```
php bin/console make:user
```

После этого генератор класса задаст несколько вопросов о том, как именно мы хотим сгенерировать класс для пользователей.

Веб разработка на PHP Symfony

User Provider

После того как в нашем приложении будет сгенерировано класс для пользователей - нужно будет запустить миграции Doctrine.

```
php bin/console make:migration
```

```
php bin/console doctrine:migrations:migrate
```

Веб разработка на PHP Symfony

User Provider

Как ни странно, но после генерации класса для пользователей файл конфигурации безопасности config/packages/security.yaml - автоматически сконфигурировался под него:

```
security:
    encoders:
        App\Entity\User:
            algorithm: auto

    # https://symfony.com/doc/current/security.html#where-do-users-come-from-user-providers
    providers:
        # used to reload user from session & other features (e.g. switch_user)
        app_user_provider:
            entity:
                class: App\Entity\User
                property: email
```

Веб разработка на PHP Symfony

User Provider

Кроме этого, уже сразу же настроено шифрование паролей, в строке `algorithm` - указывается алгоритм для шифрования паролей. В нашем случае он выбирается автоматически.

```
security:
  encoders:
    App\Entity\User:
      algorithm: auto
```

Веб разработка на PHP Symfony

User Provider

Теперь, когда у нас установлена и настроена система пользователей, мы можем создать фикстуру для таблицы users в нашей БД:

```
C:\Dev\symfonyApp> php bin/console make:fixtures  
  
The class name of the fixtures to create (e.g. AppFixtures):  
> UserFixtures
```

И чтобы зашифровать пароли в фикстурах достаточно воспользоваться следующей командой:

```
php bin/console security:encode-password
```

Авторизация

Веб разработка на PHP Symfony

Авторизация

Настало время реализовать в нашем приложении систему авторизации и регистрации. В Symfony уже есть готовое решение касательно этого. Сгенерировать логин форму можно прямо через консоль с помощью `make bundle`:

```
C:\Dev\symfonyApp> php bin/console make:auth

What style of authentication do you want? [Empty authenticator]:
  [0] Empty authenticator
  [1] Login form authenticator
> 1

The class name of the authenticator to create (e.g. AppCustomAuthenticator):
> LoginFormAuthenticator
```

Веб разработка на PHP Symfony

Авторизация

Теперь в приложении есть полностью реализованная система авторизации + ещё и с полной безопасностью через проверку CSRF токена. Реализация регистрации не представляет из себя что-то трудное. По факту создаётся форма, которая будет отправлять данные в контроллер и делать новую запись в БД об новом пользователе. Форму можно создать сразу же через `maker bundle` в консоли:

```
php bin/console make:registration-form
```


Веб разработка на PHP Symfony

Авторизация

Создав новый контроллер для регистрации и привязав форму к нашему контроллеру, можно будет полноценно регистрироваться в приложении.

Firewalls

Веб разработка на PHP Symfony

Firewalls

Firewalls - является компонентом в Security Bundle, который задаёт правила как пользователи смогут авторизироваться в приложении. Кроме этого firewalls могут создавать ограничения для того чтобы задать безопасность в нужных частях приложения. В файле `config/packages/security.yaml` - можно рассмотреть то как у нас настроены firewalls в приложении:

```
firewalls:
  dev:
    pattern: ^/(_(profiler|wdt)|css|images|js)/
    security: false
  main:
    anonymous: true
    lazy: true
    provider: app_user_provider
    guard:
      authenticators:
        - App\Security\LoginFormAuthenticator
    logout:
      path: app_logout
```

Веб разработка на PHP Symfony

Роли

Веб разработка на PHP Symfony

Роли

В Symfony уже сразу есть встроенная система ролей для пользователей, и в классе `src/Entity/User.php` - который мы сгенерировали, уже есть методы и функции, которые направлены на работу с ролями пользователя:

```
/**
 * @ORM\Column(type="json")
 */
private $roles = [];
```

```
/**
 * @see UserInterface
 */
public function getRoles(): array
{
    $roles = $this->roles;
    // guarantee every user at least has ROLE_USER
    $roles[] = 'ROLE_USER';

    return array_unique($roles);
}

public function setRoles(array $roles): self
{
    $this->roles = $roles;

    return $this;
}
```

Веб разработка на PHP Symfony

Роли

Название каждой роли в системе всегда начинается с префикса `ROLE_`. Давайте рассмотрим небольшой пример того как даётся блокировка доступа для простых пользователей в приложении в файле конфигурации Security Bundle в параметре `access_control`:

```
access_control:
- { path: ^/admin, roles: ROLE_ADMIN }
- { path: ^/profile, roles: ROLE_USER }
```

Выше приводится пример раскомментированных частей кода, в которых доступ для url с префиксов `/admin` предоставляется только пользователям, у которых есть роль админа. В нижней же строчке кода доступ к страницам с url префиксом `/profile` получают те пользователи, у которых есть роль `ROLE_USER`. Можно прописывать не только одиночные.

```
access_control:
- { path: ^/admin, roles: [ROLE_ADMIN, ROLE_MODERATOR] }
```

Веб разработка на PHP Symfony

Роли

Кроме этого можно ограничивать доступы по ролям к чётко заданным путям, например:

```
access_control:
    - { path: ^/admin/users, roles: ROLE_SUPER_ADMIN }
```

Если понадобится ограничить доступ через PHP код контроллера, это можно сделать с помощью следующей функции:

```
public function adminDashboard()
{
    // Ниже код не будет давать допуск к функции если
    // у пользователя нету роли ROLE_ADMIN
    $this->denyAccessUnlessGranted('ROLE_ADMIN');

    // + можно вывести сообщение для пользователей
    $this->denyAccessUnlessGranted('ROLE_ADMIN', null, 'Чтобы иметь доступ к странице - нужно
        иметь роль ROLE_ADMIN');
}
```

Веб разработка на PHP Symfony

Роли

В шаблонах проверка ролей делается с помощью такого способа:

```
{% if is_granted('ROLE_ADMIN') %}  
    <a href="...">Delete</a>  
{% endif %}
```


Иерархия ролей

Веб разработка на PHP Symfony

Иерархия Ролей

Чтобы определить кто какие имеет в системе преимущества - нужно воспользоваться иерархией ролей, которая задаётся в файле конфигурации `config/packages/security.yaml`:

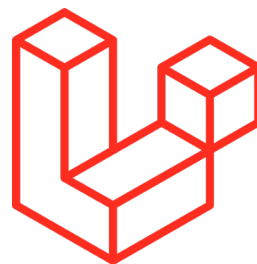
```
role_hierarchy:  
  ROLE_ADMIN: ROLE_USER  
  ROLE_SUPER_ADMIN: [ROLE_ADMIN, ROLE_MODERATOR]  
  
# Здесь роль ROLE_ADMIN получает такие же доступы как и ROLE_USER  
# А роль ROLE_SUPER_ADMIN получает доступы ролей ROLE_ADMIN и ROLE_MODERATOR
```

Веб разработка на PHP Symfony

Спасибо за внимание! До новых встреч!



Демьян Костельный
Middle PHP Developer



Проверка знаний

TestProvider.com



Проверьте как Вы усвоили данный материал на TestProvider.com

TestProvider – это online сервис проверки знаний по информационным технологиям. С его помощью Вы можете оценить Ваш уровень и выявить слабые места. Он будет полезен как в процессе изучения технологии, так и для общей оценки знаний IT специалиста.

Успешное прохождение финального тестирования позволит Вам получить соответствующий Сертификат.

Информационный видеосервис для разработчиков программного обеспечения

