Министерство Образования, Культуры и Исследований

Молдавский Государственный Университет

Факультет Математики и Информатики

Департамент Информатики

**Лабораторная работа №2**

По курсу: Frameworks

На тему: Основы Symfony.

Выполнил:Voronin Alexandr,

группа IA2102

Проверил: **Nartea Nichita**,

lector universitar

Кишинев, 2023

Оглавление

[Введение 2](#_Toc149223782)

[Цель работы 2](#_Toc149223783)

[Выполнение 2](#_Toc149223784)

[Ответы на вопросы: 3](#_Toc149223785)

[Ссылки 4](#_Toc149223786)

# Введение

Данная лабораторная работа призвана ввести студентов в захватывающий мир продвинутой функциональности Фреймворка Symfony, предоставляя им возможность глубже погрузиться в его возможности и освоить ключевые аспекты разработки веб-приложений.

# Цель работы

Данная лабораторная работа показывает более продвинутый процесс разработки приложения который включает в себя:

**Сервисы**

1. Реализовать пользовательский сервис `TaskService`, который обеспечивает создание, редактирование, удаление и получение списка задач в приложении.

**Безопасность**

Настроить аутентификацию и авторизацию.

1. Реализовать механизм аутентификации, который позволит пользователям входить в систему с использованием своего логина и пароля.

2. Внедрить меры авторизации, которые ограничат возможность создания новых задач только для авторизованных пользователей.

**Тестирование**

1. Используя Фреймворк PHPUnit, провести тестирование выбранной функциональности приложения, включая интеграционные тесты, для обеспечения надежности и корректной работы данной части приложения.

# Выполнение

**Сервисы:**

В созданном файле TaskService в папке Services, я создам функции для каждой возможности моего приложения, которые я буду вызывать их в контроллере. Для основы взаимодействия с нашей бд а именно поиск записей по id, добавление записей, показ всех записей, и удаление записи по id, я создам метод:

\_\_construct(TaskRepository $taskRepository, EntityManagerInterface $entityManager)

где будет доступ доступ к TaskRepository , EntityManagerInterface для взаимодействия с нашей бд.

Для получения нашего метода в контроллере я путем типизирования аргумента с классом сервиса получаю доступ к нашему сервису, далее на примере получение всех задач в контроллере я вызываю наш метод:

$task = $taskService->getTaskList();

**Безопасность:**

Для аутентификации в нашем приложении мы должны создать регистрацию и вход в систему.

Для этого в cmd мы вводим:

**make:registration-form**

После этого мы проходим опрос и у нас создается Контроллер, форма, и twig шаблон, Далее мы вводим в cmd: **make:auth**, также нужно создать пользователя в cmd: это ***make:user.***

После этого мы можем с помощью маршрутов, пользователя при запуске сервера направлять пользователя сразу на страницу аутентификации где пользователь может войти на сайт, а если у него нет акк, он может зарегистрировать акк.

Чтобы огранить доступ нашего приложения для не аутентифицированных пользователей, я добавил перед нашем маршрутом:

#[IsGranted('IS\_AUTHENTICATED\_FULLY')]

Что закрывает доступ к странице без аутентификации.

Для удобства и комфорта мы можем использовать в шаблоне переменную

{{ app.user.userIdentifier }}

Которая показывает ник пользователя, также если мы хотим сделать какой-то элемент недоступный для пользователя, который не вошел, мы можем в шаблоне добавить:

{%if app.user%}

[элемент страницы]

{%endif%}

**Тестирование**

Для тестирования приложения, надо скачать если не установлен phpunit в cmd:

**composer require --dev symfony/test-pack**

Чтобы запустить наш тест мы пишем в cmd

php bin/phpunit

PS: Если не уточнять папку то он запустит все тесты которые имеет

Я проведу тест приложения, а именно проверка view нашего task в нашем приложении

Для создания нашего теста мы используем команду в cmd: make:test

После этого проходим опрос и создается наш тест.

В тесте мы должны прописать маршрут к нашей странице

$client->request('GET', '/task/view/1');

Также в конце мы проверяем наличие класса container в нашем приложении

$this->assertCount(1, $crawler->filter('.container'));

## Ответы на вопросы:

**Что представляет собой `Service Layer` (Слой сервисов) в архитектуре Model-View-Controller (MVC) веб-приложений, и какую роль он играет в разделении бизнес-логики?**

Service Layer представляет собой паттерн проектирования, который заключает в себе бизнес-логику приложения и помогает разделить функции приложения на множество не больших сервисов. Для разделения бизнес-логики он играет значимую роль, благодаря ему мы можем для каждой конкретной задачи использовать свой сервис, а в контроллерах обращаться к сервису

**Объясните понятия аутентификации и авторизации в веб-разработке. Как они связаны и почему они важны для безопасности приложений?**

Аутентификация-это процесс идентификации в системе при помощи логина и пароля,

Авторизация-это процесс определения ролей для пользователя, а именно доступ пользователю на переход на другие страницы или совершение каких-либо действий на сайте

**Как, в выбранном Вами фреймворке, можно реализовать авторизацию пользователей и ограничение доступа к определенным маршрутам или действиям на основе их ролей?**

В symfony можно реализовать авторизацию при помощи Ролей пользователя. Чтобы ограничить доступ к конкретной странице можно в файле security.yaml прописав куда пользователь может попасть, или можно прописать атрибут #[IsGranted(роль)] непосредственно в контроллере для маршрута.

**В чем заключается разница между юнит-тестированием, интеграционным тестированием?**

*Юнит-тестирование* проверяет отдельный модуль приложения на его функциональное и графическое соответствие.

*Интеграционное тестирование* проверяет большую часть приложения, а именно взаимодействие компонентов между собой

# Ссылки

[Curs: Frameworks for Web Application Development (usm.md)](https://moodle.usm.md/course/view.php?id=6440)

[Symfony Documentation](https://symfony.com/doc/current/index.html)

[Form controls · Bootstrap v5.3 (getbootstrap.com)](https://getbootstrap.com/docs/5.3/forms/form-control/)

[AlexVor550/ToDo\_2.0 (github.com)](https://github.com/AlexVor550/ToDo_2.0)=ссылка на мой github