HwProj

Веб-приложение для сдачи и проверки заданий по программированию

Мендалиев Р.Р., 243 Гирин А.Р., 244 Афанасов А.К., 241

Научный руководитель: к.т.н. Литвинов Ю.В.

Проблематика

- Необходимость в дистанционной проверке заданий по программированию
- Прекращение поддержки проекта HwProj
- Наиболее приближенная альтернатива Stepik
 - Неудобно создавать курсы
 - Закрытый доступ платный
 - Не наглядно

Цель

Реализовать веб-приложение для проверки заданий по программированию

Задачи

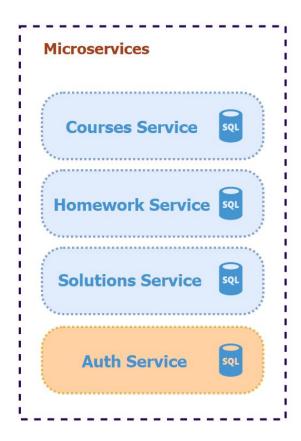
- Backend
 - Система ролей
 - Управление аккаунтами
 - Управление курсами
 - Взаимодействие студентов с курсами
 - Способы проверки заданий и отслеживание прогресса
- Frontend
 - Пользовательский интерфейс

Задачи

- Настроить непрерывную интеграцию, а также автоматизировать развёртывание в облачную платформу
 - Выбрать и научиться использовать:
 - облачную платформу
 - сервис для СІ и автоматизации развёртывания
 - Развернуть сервер баз данных в облачной платформе и ограничить доступ к базам данных
 - Настроить непрерывную интеграцию микросервисов
 - Автоматизировать развёртывание микросервисов на виртуальной машине

Архитектура

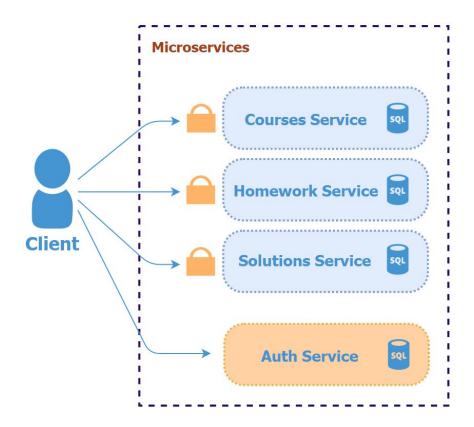
- Микросервисная архитектура
- Каждый микросервис представляет ASP.NET Core Web API приложение



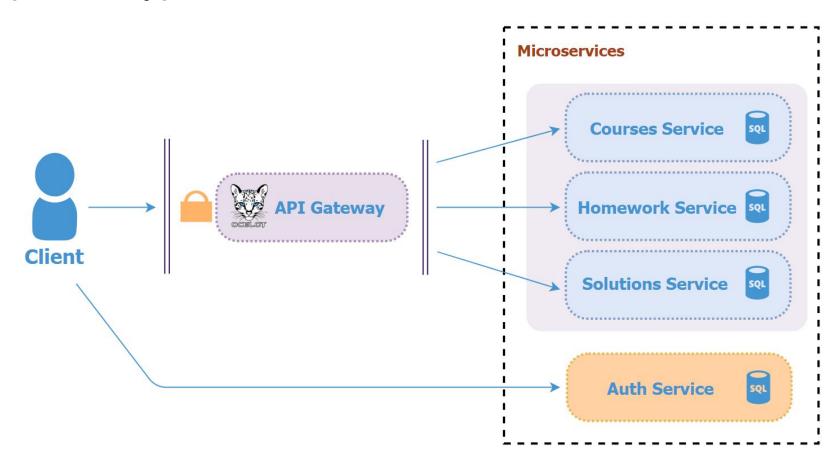
Архитектура. Изначальный план

Прямое взаимодействие клиента с микросервисами

- Проблема безопасности
- Проблемы сквозной функциональности



Архитектура. Итог

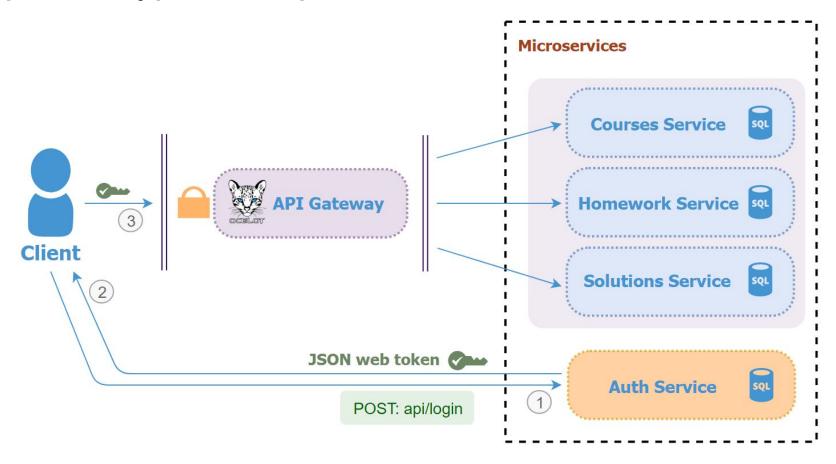


Шлюз API Gateway

- ASP.NET Core Web API приложение
- Реализует паттерн API Gateway / Backends for Frontends
 - о Отделяет клиента от микросервисов
 - о Предоставляет единую авторизацию для всех микросервисов
 - Преобразует запросы, добавляет в них необходимую для сервисов информацию о пользователе
- Использует Ocelot
 - Open Source проект для работы с ASP.NET Core,
 предоставляющий инструментарий для реализации шлюза Gateway
 - Не требует много ресурсов и работает быстро
 - Аналоги: Apigee, Kong, MuleSoft



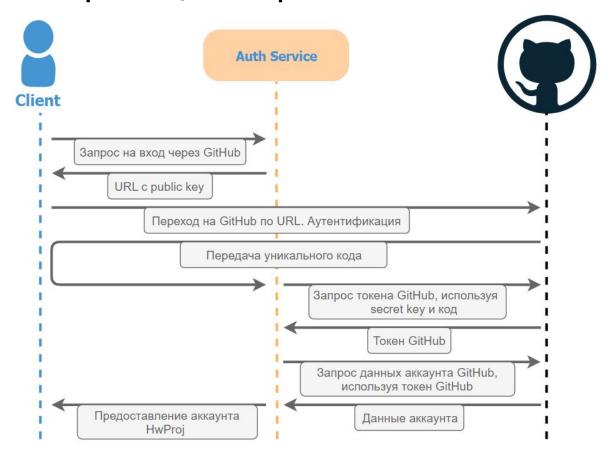
Архитектура. Авторизация

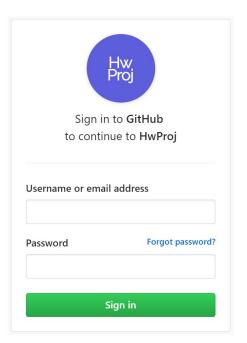


Авторизация и аутентификация

- Использует ASP.NET Core Identity для управления учетными записями пользователей
 - Регистрация пользователей
 - Вход и выход из системы
 - Редактирование аккаунта
- Служба для работы с токенами
 - Создание токенов доступа для пользователей
 - Используется HMAC-SHA256 с секретным ключом
- Служба оповещений
 - Подтверждение действий над аккаунтом через Email
- Возможность авторизации через GitHub

Авторизация через GitHub

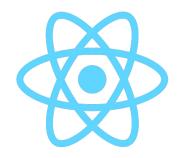


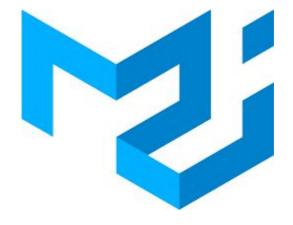


Front-end

- TypeScript
- React
- Material-Ul
- Swagger

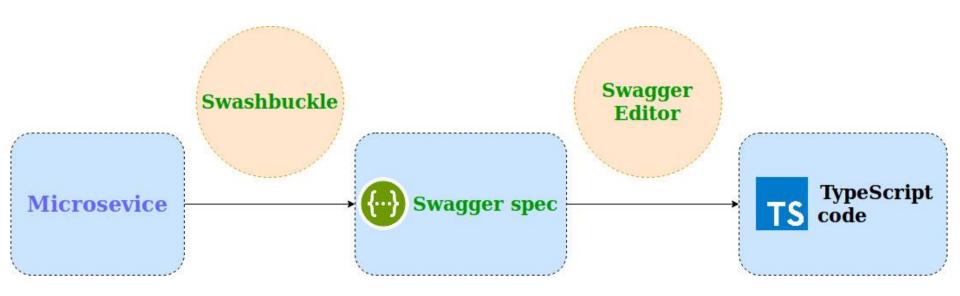








Генерация клиентов



ТЕКУЩИЕ КУРСЫ

ЗАВЕРШЕННЫЕ КУРСЫ

- ООП на Java 244
- Программирование 243

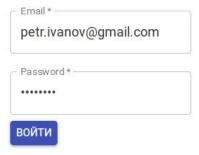
HWPROJ ВХОД РЕГИСТРАЦИЯ

Регистрация

| Петр | |
|-----------|----------------|
| ПСТР | |
| Фамилия | A * |
| Ивано | В |
| Email * – | |
| petr.iva | anov@gmail.com |
| Пароль ' | * |
| ••••• | |
| Поддвер | одите пароль * |
| | |

HWPROJ ВХОД РЕГИСТРАЦИЯ

Войти



Программирование 🖍

Рома Мендалиев

243

admin@gmail.com

| Студент | | 1 | |
|-------------|-----------|-----------|------------|
| | Факториал | Фибоначчи | Сортировка |
| Петров Ваня | | | |
| Петров Иван | | | |

Новые заявки на вступление в курс:

1. Иванов Петр





добавить домашку

Задачи

Введение, С# 24.05.2019

Презентация

1. Факториал 📋 🧪

Посчитать факториал

2. Фибоначчи 📋 🧪

Посчитать числа Фибоначчи

3. Сортировка 📋 🧪

Отсортировать массив какой-либо из сортировок

ДОБАВИТЬ ЗАДАЧУ

Добавить домашку

Название домашки * ООП, ООП в С#

Описание домашки

[ООП, ООП в C# (презентация)](https://github.com/yurii-litvinov/courses/blob/master/programming-2nd-semester/02-oop/02-oop-slides.pdf)

[ООП, ООП в C# (конспект)](https://github.com/yurii-litvinov/courses/blob/master/programming-2nd-semester/02-oop/02-ooptext.pdf)

1. Задача

УБРАТЬ ЗАДАЧУ

Название задачи *

Хэш-таблиц

Условие задачи

Написать хеш-таблицу в виде класса с использованием класса-списка. Должно быть можно добавлять значение в хеш-таблицу, удалять и проверять на принадлежность

ЕЩЁ ЗАДАЧУ

НШРРОЈ РЕДАКТИРОВАТЬ ДАННЫЕ ВЫЙТИ

Назад к курсу

Хэш-таблиц

Написать хеш-таблицу в виде класса с использованием класса-списка. Должно быть можно добавлять значение в хеш-таблицу, удалять и проверять на принадлежность

| Ссылка на решение * https://github.com/reacheight | | |
|---|--|--|
| Комментарий | | |
| ДОБАВИТЬ РЕШЕНИЕ ОТМЕНИТЬ | | |

Программирование

Рома Мендалиев

243

admin@gmail.com

| Студент | 1 | | | 2 |
|-------------|-----------|-----------|------------|------------|
| | Факториал | Фибоначчи | Сортировка | Хэш-таблиц |
| Петров Ваня | | | | |
| Петров Иван | | | | |
| Иванов Петр | | | | |

Задачи

ООП, ООП в С# 24.05.2019

ООП, ООП в С# (презентация) ООП, ООП в С# (конспект)

1. Хэш-таблиц

Написать хеш-таблицу в виде класса с использованием класса-списка. Должно быть можно добавлять значение в хеш-таблицу, удалять и проверять на принадлежность

Введение, С# 24.05.2019

Презентация

1. Факториал

Посчитать факториал

2. Фибоначчи

Посчитать числа Фибоначчи

3. Сортировка

Введение. СІ и развёртывание

- Облачные платформы
- Виртуальные машины
- Развертывание
- Непрерывная интеграция

Похожие работы

- НwProj прошлого года
- Активный HwProj

Используемые технологии и средства

- Docker
- Docker Hub
- Amazon Web Services
 - Ho не Microsoft Azure
 - База данных RDS
 - Виртуальная машина ЕС2
 - Но не кластеры (например, Amazon Kubernetes, Azure kubernetes)
- TeamCity
 - Но не Jenkins







Docker

• Образ -> Контейнеры

- Микросервисы => использование контейнеров
 - Изолированность
 - Самодостаточность
 - Лёгкая доставка
 - Лёгкая переносимость
 - Лёгкое обновление

Автоматизация с помощью TeamCity

- Для каждого микросервиса:
 - Коммит
 - |>
 - Тестирование
 - |>
 - Docker
 - Собрать образ
 - 0 |>
 - Отправить образ в Docker Hub
 - |>
 - Запустить контейнер на виртуальной машине

```
[05:00:40] ► Step 1/5: Tests (.NET CLI (dotnet)) (7s)
[05:00:47] ► Step 2/5: Docker login (Command Line) (2s)
[05:00:50] ► Step 3/5: build (Command Line) (53s)
[05:01:43] ► Step 4/5: Docker push (Command Line) (5s)
[05:01:49] ► Step 5/5: connect ssh and exit (SSH Exec) (2s)
```

Проблема

Docker отказывался работать через TeamCity

Решение

• Переменные окружения системного и пользовательского аккаунтов

Результаты

- Афанасов А.К.
 - Настроил непрерывную интеграцию микросервисов, а также автоматизировал их развёртывание в облачную платформу Amazon Web Services
 - Выбрал и научился использовать:
 - о облачную платформу AWS
 - TeamCity для CI и автоматизации развёртывания
 - Развернул сервер баз данных в облачной платформе и ограничил доступ к базам данных
 - Настроил непрерывную интеграцию микросервисов
 - Автоматизировал развёртывание микросервисов на виртуальной машине, доставляя их Docker контейнерами

Результаты

- Мендалиев Р.Р.
 - Управление курсами
 - Взаимодействие студентов с курсами
 - Способы проверки заданий и отслеживание прогресса
 - Пользовательский интерфейс
- Гирин А.Р.
 - Авторизация и аутентификация
 - Управление аккаунтами
 - Шлюз Арі Gateway