

HwProj 2.0.1

ASP.NET Core сервис для проверки заданий по
программированию

Мендалиев Р. Р.
Гирин А. Р.

Проблематика

- Необходимость в дистанционной проверке заданий по программированию
- Прекращение поддержки проекта HwProj (последние обновления – в декабре 2016)
- Наиболее приближенная альтернатива - Stepik
 - Неудобно создавать курсы
 - Закрытый доступ - платный
 - Не наглядно

HwProj 2.0

- Монолитная архитектура
- Не современные технологии
 - .NET Framework
 - ASP.NET MVC 5
 - JQuery

<https://github.com/IntelligeNET/HwProj-2.0>

Задачи

Реализовать часть функциональности (back end) согласно уже разработанной архитектуре

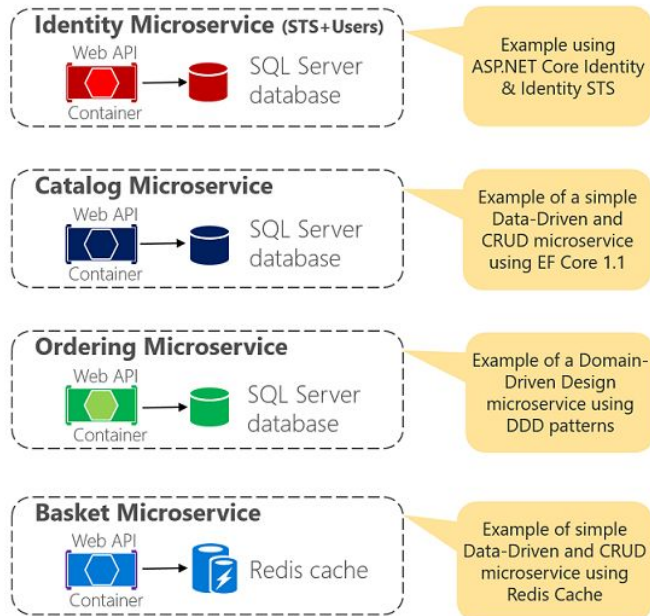
- Авторизация и аутентификация
- Управление курсами
- Управление задачами курса

Используемые технологии

- .NET Core
- ASP.NET Core
- Entity Framework Core, MS SQL

Микросервисная архитектура

Different types of microservices

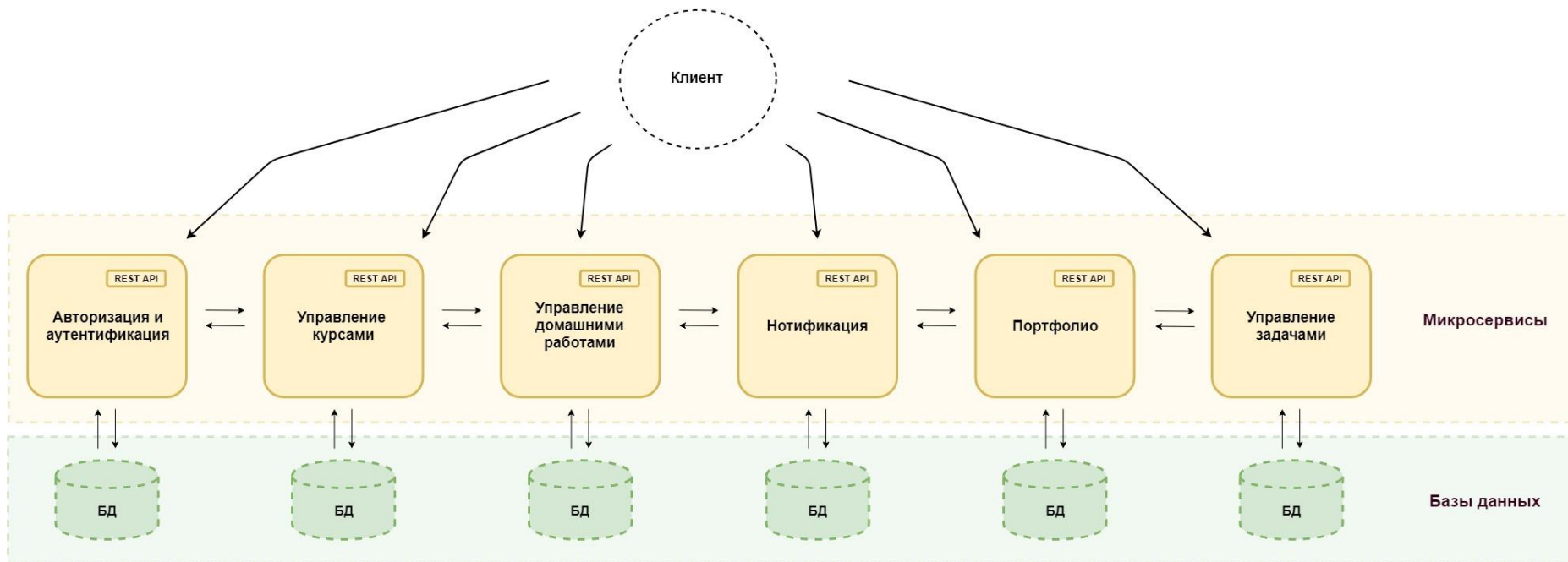


<https://github.com/dotnet-architecture/eShopOnContainers>

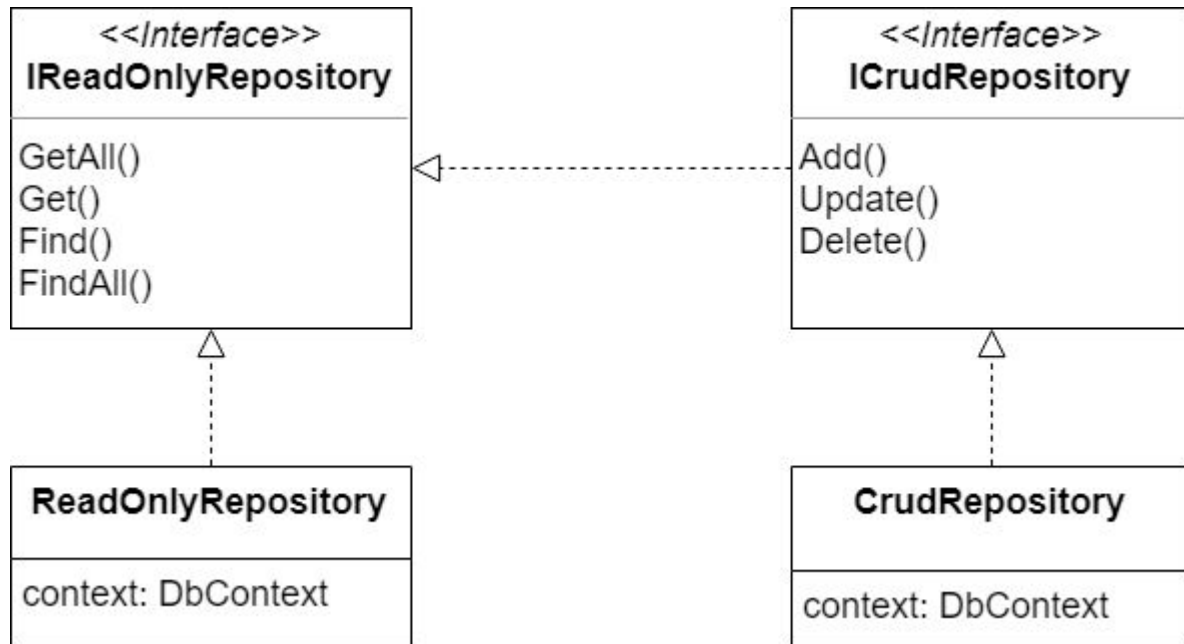
Плюсы микросервисной архитектуры

- Декомпозиция большого приложения
- Независимая разработка микросервисов
- Возможность использовать любые технологии
- Независимое развёртывание
- Отдельные микросервисы легче тестировать

Микросервисная архитектура в HwProj 2.0.1



Репозитории



Микросервис управления курсами

GET /api/Courses

POST /api/Courses

GET /api/Courses/{id}

DELETE /api/Courses/{id}

POST /api/Courses/update/{courseId}

POST /api/Courses/sign_in_course/{courseId}

POST /api/Courses/accept_student/{courseId}

POST /api/Courses/reject_student/{courseId}

GET /api/Courses/student_courses/{userId}

GET /api/Courses/mentor_courses/{userId}

```
public class Course
{
    public long Id { get; set; }

    public string Name { get; set; }

    public string GroupName { get; set; }

    public bool IsOpen { get; set; }

    public bool IsComplete { get; set; }

    public string MentorId { get; set; }

    public List<CourseStudent> CourseStudents { get; set; }
}
```

GET

/api/Courses/{id}

Response body

```
{
  "id": 60003,
  "name": "Java programming",
  "groupName": "244",
  "isOpen": false,
  "isComplete": false,
  "mentorId": "Mentor",
  "students": [
    {
      "id": "user183",
      "isAccepted": true
    },
    {
      "id": "user56",
      "isAccepted": true
    },
    {
      "id": "user7",
      "isAccepted": true
    }
  ]
}
```

Download

Пример: редактирование курса

GET

/api/Courses/{id}

Response body

```
{
  "id": 60003,
  "name": "Java programming",
  "groupName": "244",
  "isOpen": false,
  "isComplete": false,
  "mentorId": "Mentor",
  "students": [
    {
      "id": "user183",
      "isAccepted": true
    },
    {
      "id": "user56",
      "isAccepted": true
    },
    {
      "id": "user7",
      "isAccepted": true
    }
  ]
}
```

Download

POST

/api/Courses/update/{courseId}

```
{
  "name": "C# programming",
  "groupName": "243",
  "isOpen": false,
  "isComplete": false
}
```

GET

/api/Courses/{id}

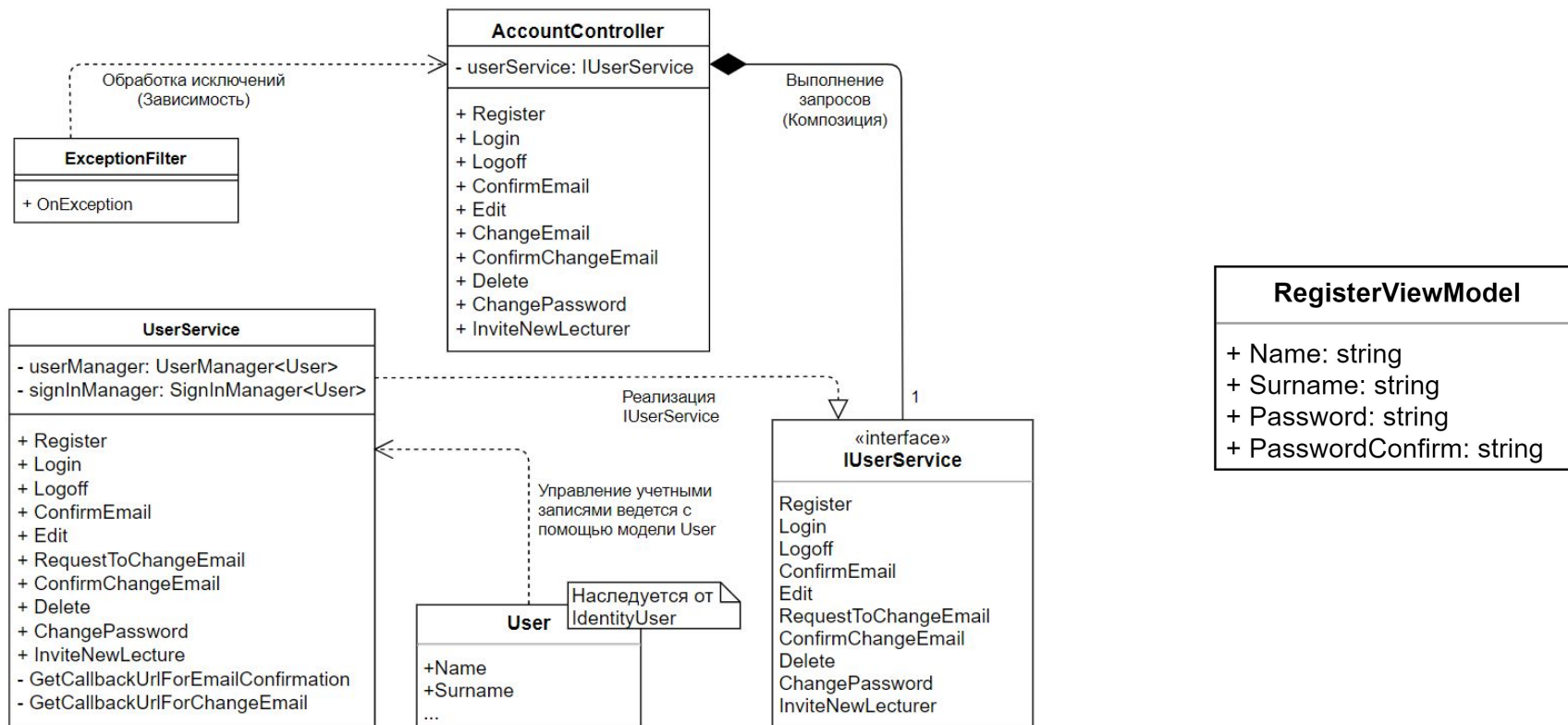
```
{
  "id": 60003,
  "name": "C# programming",
  "groupName": "243",
  "isOpen": false,
  "isComplete": false,
  "mentorId": "Mentor",
  "students": [
    {
      "id": "user183",
      "isAccepted": true
    },
    {
      "id": "user56",
      "isAccepted": true
    },
    {
      "id": "user7",
      "isAccepted": true
    }
  ]
}
```

Download

Авторизация и аутентификация

- ASP.NET Core Web API приложение
- Управление учетными записями, аутентификация пользователей
- Используется система ASP.NET Core Identity

Авторизация и аутентификация



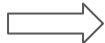
Авторизация и аутентификация

Пример - регистрация нового пользователя

RegisterViewModel

```
{
  "name": "TestName",
  "surname": "TestSurname",
  "email": "Test@gmail.com",
  "password": "Test@1234",
  "passwordConfirm": "Test@1234"
}
```

POST: Register



Response body

```
"https://localhost:44338/api/account/confirmemail?userId=4a405723-afa5-47ac-922f-314f9fc9912e&code=CfDJ8JlnHot%2Bw%2FNbvglX1LvV1a2SoAXv1Bnt%2FH%2FkY1CAmoM2vrAKR4Nc6kQeyYALH%2F4%2FHsgfteWwWkzPnVp01Nd16qJaFDuyIwF6KEtm%2Bt3Xj7%2FtZU8yYC1KfE8fobTvb:
```

Попытка входа без подтверждения регистрации

Code

400

Response body

"Email не был подтвержден"

Response body

```
{
  "name": "TestName",
  "surname": "TestSurname",
  "id": "4a405723-afa5-47ac-922f-314f9fc9912e",
  "userName": "Test@gmail.com",
  "normalizedUserName": "TEST@GMAIL.COM",
  "email": "Test@gmail.com",
  "normalizedEmail": "TEST@GMAIL.COM",
  "emailConfirmed": false,
}
```

Вход после перехода по ссылке

Code

200

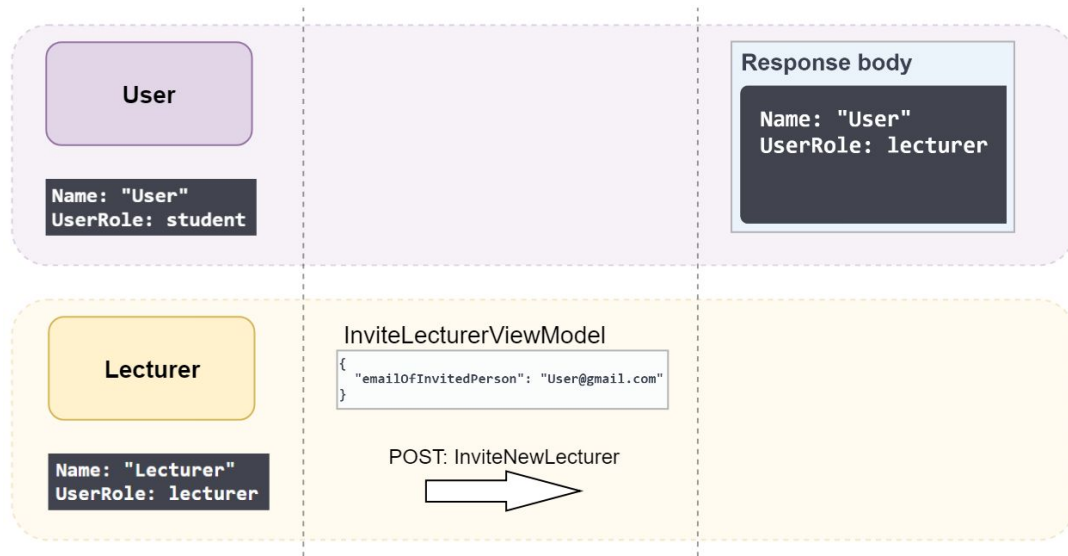
Response body

```
{
  "name": "TestName",
  "surname": "TestSurname",
  "id": "4a405723-afa5-47ac-922f-314f9fc9912e",
  "userName": "Test@gmail.com",
  "normalizedUserName": "TEST@GMAIL.COM",
  "email": "Test@gmail.com",
  "normalizedEmail": "TEST@GMAIL.COM",
  "emailConfirmed": true,
}
```

Авторизация и аутентификация

Система ролей представлена студентами и преподавателями.

Преподавателем можно стать после получения приглашения от другого преподавателя



Response body

Name: "User"
UserRole: lecturer

```
public class InviteLecturerViewModel
{
    [Required, EmailAddress]
    public string EmailOfInvitedPerson { get; set; }
}
```

AccountController:

```
[HttpPost, Route("InviteNewLecturer"), ExceptionFilter]
[Authorize(Roles = "lecturer")]
public async Task<IActionResult> InviteNewLecturer(
    InviteLecturerViewModel model)
{
```


Микросервис управления заданиями

GET /api/Homework/{homeworkId}

GET /api/Homework/course_homework/{courseId}

POST /api/Homework/add_homework/{courseId}

POST /api/Homework/update_homework/{homeworkId}

DELETE /api/Homework/delete_homework/{homeworkId}

POST /api/Homework/add_application/{homeworkId}

POST /api/Homework/update_application/{applicationId}

DELETE /api/Homework/delete_application/{applicationId}

GET /api/Homework/get_task/{taskId}

POST /api/Homework/add_task/{homeworkId}

POST /api/Homework/update_task/{taskId}

DELETE /api/Homework/delete_task/{taskId}

Модель домашней работы

```
public class Homework
{
    public long Id { get; set; }

    public string Title { get; set; }

    public long CourseId { get; set; }

    public List<HomeworkApplication> Applications { get; set; }

    public List<HomeworkTask> Tasks { get; set; }
}
```

Пример: добавление задачи в домашнюю работу

GET

/api/Homework/{homeworkId}

```
{
  "id": 2,
  "title": "first homework",
  "applications": [
    {
      "id": 1,
      "title": "presentation",
      "link": "link.to/presentation"
    }
  ],
  "tasks": []
}
```

Download

POST

/api/Homework/add_task/{homeworkId}

```
{
  "title": "Lazy",
  "description": "Implement lazy evaluation interface"
}
```

Пример: добавление задачи

GET

/api/Homework/{homeworkId}

```
{
  "id": 2,
  "title": "first homework",
  "applications": [
    {
      "id": 1,
      "title": "presentation",
      "link": "link.to/presentation"
    }
  ],
  "tasks": [
    {
      "id": 2,
      "title": "Lazy",
      "description": "Implement lazy evaluation interface"
    }
  ]
}
```

Download

Результаты

- Мендалиев Р. Р.
 - Сервис управления курсами
 - Создание, редактирование, удаление курсов
 - Вступление в курсы
 - Сервис управления задачами курса
 - Создание, редактирование, удаление
- Гирин А. Р.
 - Сервис авторизации и аутентификации
 - Реализованы все необходимые методы для управления аккаунтами пользователей

<https://github.com/IntelligeNET/HwProj-2.0.1>