

Санкт-Петербургский государственный университет

Кафедра системного программирования

Группа 22.Б10-мм

Улучшение инструментов создания и редактирования домашних заданий в сервисе HwProj 2.0.1

Позиев Алексей Владимирович

Отчёт по учебной практике
в форме «Производственное задание»

Научный руководитель:
старший преподаватель кафедры СП, к. т. н., Ю. В. Литвинов

Консультант:
инженер-программист «JetBrains», А. В. Бережных

Санкт-Петербург
2024

Оглавление

Введение	3
1. Постановка задачи	5
2. Обзор	6
2.1. Обзор используемых технологий	6
2.2. Существующие решения	6
2.3. Обзор архитектуры HwProj 2.0.1	7
3. Реализация	9
3.1. Серверная часть задания общих свойств дат	9
3.2. Валидация для принимаемых значений	10
3.3. Поддержка задания общих свойств на фронтенде	11
3.4. Исправление ошибки с возможностью изменения курса всеми лекторами	13
3.5. Реализация интеграционных тестов и поддержание исправ- ности существующих	14
3.6. Локализация времени	15
3.7. Обобщение компонентов	15
4. Апробация	17
Заключение	18
Список литературы	19

Введение

В большинстве курсов по программированию или информатике, одними из ключевых аспектов качественного обучения являются домашние задания и доступность получения материалов курса. Эти аспекты порождают организационные проблемы, к примеру, связанные с их публикацией, в случае с домашними заданиями: сдачей, выставлением оценок и контроля дедлайнов. В целях решения подобных проблем был создан HwProj 2.0.1 — веб-сервис с микросервисной архитектурой, предназначенный для сдачи и проверки домашних работ, поддерживающий особенности курсов кафедры системного программирования СПбГУ.

Таким образом, важной задачей сервиса является предоставление удобных и функциональных инструментов для создания и редактирования домашних работ. Но, к сожалению, несмотря на важность этих инструментов у них есть несколько недостатков.

Первым существенным недостатком является то, что на данный момент у лекторов нет возможности задать общие даты для всех задач в домашнем задании. Поэтому при создании новой задачи или редактирования старой требуется каждый раз выставлять, вероятно, одни и те же даты, что чрезвычайно неудобно.

Следующей долгосуществующей проблемой является работа со временем и датами в HwProj. Сервис привязан к московскому времени, это приводит к неудобствам для людей которые находятся в другом часовом поясе. Также путаница в коде при работе с датами приводит к различным ошибкам, к примеру: при создании задач все даты выставляются на три часа позже.

Третьим недостатком стала возможность использования лекторами инструментов создания и редактирования домашних работ для изменения курсов, в которых они не являются лекторами. Это является большой проблемой безопасности сервиса, потому что иметь возможность влиять на содержание курса должны только менторы самого курса. Это стало особенно важно, так как ассистентами на курсах с недавних пор могут быть студенты старших курсов, которые сами учатся через

HwProj.

И наконец, важной проблемой является запутанность и многочисленное копирование кода в компонентах, связанных с созданием и редактированием домашних заданий в кодовой базе клиентской части, что сильно усложняет исправление ошибок в данных компонентах.

Данная работа посвящена улучшению инструментов для создания и редактирования домашних работ посредством исправления вышеперечисленных недостатков.

1. Постановка задачи

Целью работы является улучшение инструментов создания и редактирования домашних заданий в сервисе HwProj 2.0.1, посредством устранения их недостатков. Для её выполнения были поставлены следующие задачи.

1. Спроектировать функциональность для задания общих свойств для задач в домашнем задании и реализовать ее на стороне бэкенда HwProj 2.0.1.
2. Реализовать валидацию для сущностей домашних заданий и задач, получаемых с клиента.
3. Обобщить компоненты, связанные с публикацией и редактированием домашних работ, в кодовой базе клиентской части.
4. Поддержать корректную работу существующих и реализовать новые интеграционные тесты для методов создания и редактирования домашних заданий.
5. Поддержать задание общих свойств для задач в домашнем задании на стороне фронтенда HwProj 2.0.1.
6. Исправить проблему с возможностью изменения курса всеми лекторами.
7. Локализовать время для пользователей сервиса.

2. Обзор

2.1. Обзор используемых технологий

Большая часть технологий уже до этого применялись в сервисе HwProj 2.0.1 и не изменялись.

- C# [3] — основной язык серверной части HwProj 2.0.1.
- ASP.NET Core [1] — кроссплатформенный фреймворк для разработки пользовательских веб-сервисов.
- AutoFixture [2] — библиотека для автогенерации данных для тестов.
- Fluent Validation [4] — библиотека .NET для создания строго типизированных правил валидации.
- NUnit [5] — популярный фреймворк для написания юнит-тестов.
- TypeScript [8] — сильнотипизированный язык на основе JavaScript. Основной язык клиентской части HwProj 2.0.1.
- React [7] — библиотека для создания пользовательского интерфейса с помощью компонентов.

2.2. Существующие решения

Обзор существующих решений проводился в целях определения преимуществ и недостатков в установке дат в инструментах создания и редактирования домашних заданий для формирования наиболее удобной реализации.

2.2.1. Blackboard Learn

Blackboard Learn [10] — платформа предоставляет единую интерактивную среду для обучения и взаимодействия между обучаемыми или студентом и преподавателем. Имеет гибкий набор инструментов для управления курсами и домашними заданиями.

На Blackboard Learn нельзя задавать общие даты для всех задач в домашнем задании, это необходимо делать вручную для каждой задачи, что крайне долго, если задач много. Редактирование для каждой задачи также отдельно.

2.2.2. Stepik

Stepik [11] — образовательная платформа и конструктор онлайн-курсов.

Stepik позволяет устанавливать даты для всего модуля, что, безусловно, удобно и ускоряет создание необходимых модулей. Однако отсутствует возможность установить индивидуальные даты для отдельных уроков внутри модуля. Данное решение позволяет избавиться от многих проблем с валидацией, но также отнимает у создателей курсов возможность, к примеру, поставить дедлайн для одного урока позже чем у другого, что выглядит вполне реальной ситуацией. Редактирование изменяет даты сразу для всех уроков.

2.2.3. MS Teams

MS Teams [9] — это платформа для командной работы и обмена информацией. Предоставляет возможность создавать Assignments, но только в единственном экземпляре за раз, без возможности задать какие-либо значения по-умолчанию.

2.3. Обзор архитектуры HwProj 2.0.1

Серверная часть HwProj 2.0.1 разделена на четыре микросервиса: AuthService, NotificationsService, SolutionsService, CoursesService. Для связи между ними используется брокер сообщений RabbitMQ [6]. Общение между бэкендом и фронтендом происходит через API Gateway — REST API сервиса (Рис. 1).

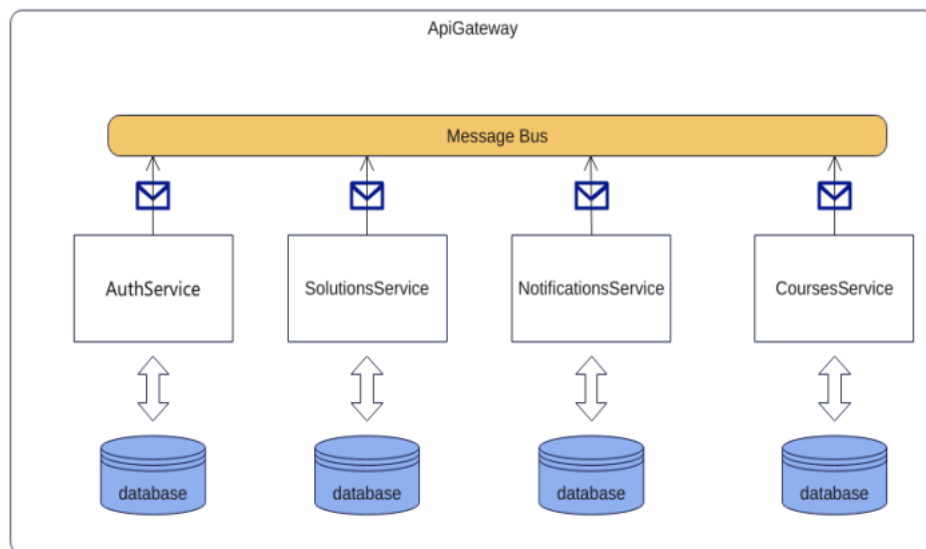


Рис. 1: Архитектура серверной части

3. Реализация

3.1. Серверная часть задания общих свойств дат

По мере реализации функциональности для задания общих свойств дат первоначальной идеей решения было добавление свойств дат в сущность домашних заданий и их удаление из задач, но в таком случае пропадала возможность смены дат для некоторых задач, что, к примеру, лишало возможности поставить дедлайн более легкой задачи раньше, чем более сложной. Поэтому от данного решения пришлось отказаться.

В связи с решением дать возможность переопределять даты в некоторых задачах было решено не только добавить свойства дат в сущность домашних заданий, но и оставить их в задачах. Для корректного определения итоговых значений свойств дат задач были сформулированы следующие правила:

- если для задачи заданы свойства, они имеют приоритет и не меняются при редактировании домашнего задания, даты домашнего задания для задачи никак не учитываются;
- если для задачи не заданы свойства, используются свойства домашнего задания.

Таким образом появилась необходимость уметь отличать в задачах заданные свойства от незаданных. Решением стало изменить типы свойств дат на *nullable-типы* (Рис. 2), в таком случае, значение *null* говорит о том, что свойство не задано и его следует брать из домашнего задания. Однако, в случае дедлайнов, которых может просто не быть в задаче, определиться с тем, есть ли у нее дедлайн, позволяет свойство *HasDeadline*.

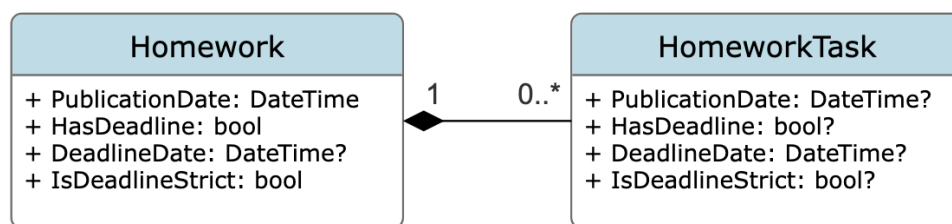


Рис. 2: Классы домашнего задания и задачи

Сервис курсов возвращает задачи с уже заполненными, то есть подставленными по вышеперечисленным правилам, свойствами. Таким образом остальные сервисы освобождены от данной логики. Но инструментам редактирования для корректного отображения необходима информация о том, какие свойства задачи изменены, для этого реализованы методы *GetForEditingHomework* и *GetForEditingTask*, которые возвращают сущности без заполненных свойств дат в задачах.

3.2. Валидация для принимаемых значений

В связи с появлением дат в сущности домашних заданий появились новые инварианты, к примеру, дата публикации задачи не может быть раньше даты публикации содержащей ее домашнего задания. Кроме того были и давно существующие старые инварианты, например, что дедлайн не может быть раньше даты публикации. Ранее сервис пропускал сущности с некоторыми нарушенными инвариантами, а некоторые просто менял без оповещения пользователя. Поэтому было решено реализовать валидатор для проверки сущностей домашних заданий и задач для поддержания инвариантов и, в случае их нарушения, недопущению их попадания в базу данных.

Первоначальное решение предполагало использование библиотеки *Fluent Validation*. Даже был написан код, но возникли трудности: к примеру, задача может поступать на бекэнд в виде двух сценариев: как часть нового домашнего задания или как новая задача в существующее домашнее задание. В зависимости от сценария валидатор должен вызывать различные асинхронные методы для получения информации из базы данных. Но, во-первых, так как правила для валидаторов в данной

библиотеке задаются в конструкторе — мы не можем вызывать асинхронные методы внутри него. Во-вторых, после создания объекта валидатора ему нельзя передавать параметры для валидации в зависимости от сценария. Поддержание этих особенностей библиотеки порождало тяжело читаемый и сложно поддерживаемый код, поэтому от ее использования были вынуждены отказаться.

Вместо этого было написано два статических метода: *ValidateHomework* и *ValidateTask*; благодаря тому что это методы, мы можем свободно передавать им необходимую информацию о, допустим, существующем домашнем задании. В случае нарушения инварианта, методы возвращают список строк, описывающих ошибки. Если данный список не пуст, контроллер возвращает *BadRequest* вместе со строками ошибок.

3.3. Поддержка задания общих свойств на фронтенде

Благодаря тому, что бэкенд возвращает уже заполненные свойства дат у задач, удалось обеспечить корректную работу без многочисленных исправлений. Поэтому главной задачей стало предоставить пользователю удобный интерфейс для выставления дат домашнего задания и их возможного переопределения в задачах.

Так как изменение дат в задачах не очень частое явление, компоненты, позволяющие это делать, не должны сильно бросаться в глаза, поэтому было решено показывать их только после нажатия на ссылку (Рис. 3).

1. Задача

УБРАТЬ ЗАДАЧУ

Название задачи *

TestTask

Баллы *

10

РЕДАКТИРОВАТЬ

ПРЕВЬЮ

Условие задачи

Даты задачи

Нужны особые даты?

Рис. 3: Создание задачи с неоткрытыми датами

Компонент выставления дат работает следующим образом:

- Если значение в input какого-то свойства задачи совпадает со значением аналогичного свойства в input домашнего задания, то свойство принимает значение *undefined*.
- Если значение в input изменилось относительно аналогичного свойства в домашнем задании, то оно принимает это значение и с помощью текста под input уведомляет, что значение изменено.

К сожалению, данное решение уменьшает возможности серверной части, не позволяя определить для задач даты такие же как у домашнего задания, которые бы при изменении дат домашнего задания не изменились, но в то же время делает компонент менее загроможденным.

Также требовалось поддерживать инварианты. Если они нарушаются, то приложение оповещает об этом пользователя и забирает возможность отослать такое задание (Рис. 4).

Название *

TestHomework

РЕДАКТИРОВАТЬ

ПРЕВЬЮ

Описание

Дата публикации

12/20/2023 12:00 AM

☒

Дедлайн

Дедлайн задания *

01/25/2024 11:59 PM

☐

Строгий

1. Задача

УБРАТЬ ЗАДАЧУ

Название задачи *

TestTask

Баллы *

10

РЕДАКТИРОВАТЬ

ПРЕВЬЮ

Условие задачи

Оставить обычные даты

Дата публикации

12/21/2023 12:00 AM

☒

Дедлайн

Было 12/20/2023, 12:00 AM

Дедлайн задания

12/15/2023 11:59 PM

☒

Строгий

Дедлайн задачи раньше ее публикации: 12/21/2023, 12:00 AM

Был нестрогий

ЕЩЁ ЗАДАЧУ

ДОБАВИТЬ ДОМАШНЕЕ ЗАДАНИЕ

ОТМЕНИТЬ

Рис. 4: Создание домашнего задания и задачи

3.4. Исправление ошибки с возможностью изменения курса всеми лекторами

В *CoursesService* для проверки того, является ли пользователь, вызвавший метод, лектором данного курса, используется атрибут *CourseMentorOnlyAttribute*, но несмотря на то, что он был применен ко всем методам, предназначенным для редактирования и добавления домашних заданий, в них он не работал. Это происходило по причине того, что для его работы требовался *CourseId*, который он брал из пути запроса, но

некоторым методам не нужен данный параметр, поэтому не находя его, он просто пропускал запрос дальше.

Для решения данной проблемы было добавлено еще два случая обработки: с *Id* задачи и домашнего задания, которые аналогично с логикой для *CourseId* запрашивают массив лекторов своего курса и проверяют на то, что вызвавший пользователь есть в этом списке.

3.5. Реализация интеграционных тестов и поддержание исправности существующих

3.5.1. Поддержание существующих тестов

AutoFixture, в силу случайной генерации данных, порождал сущности, нарушающие новые инварианты, из-за чего не проходили тесты в *SolutionsService*. Решением стало отключение генерации для свойств *IsDeadlineStrict* и *HasDeadline* и задание нового правила генерации дат, по которому дата публикации не может быть раньше даты дедлайна

3.5.2. Реализация новых интеграционных тестов

Для тестирования инструментов создания и редактирования домашних заданий в *CoursesService* были реализованы следующие тестовые сценарии.

- Получение домашнего задания методами, заполняющими свойства дат в задачах;
- Получение домашнего задания методами, не заполняющими свойства дат в задачах;
- Создание и редактирование домашних заданий и задач с выполненными инвариантами;
- Создание и редактирование домашних заданий и задач нарушающих инварианты;

- Использование методов влияющих на материал курса лектором другого курса;

3.6. Локализация времени

В целях избавления от зависимости времени сервиса от конкретного часового пояса, в нашем случае московского, были приняты следующие решения.

- В серверной части даты хранятся и используются в формате UTC, что означает отказ от свойства *DateTime.MoscowNow*, повсеместно использовавшегося в сервисе. Это также исправило ошибку, из-за которой студенты могли получить уведомление о появлении нового домашнего задания на три часа раньше его публикации;
- Клиентская часть теперь работает с локальным временем, для этого, аналогично с бэкендом, отказались от методов, предназначенных для работы только с московским временем, к примеру метод *GetMoscowNow*;

Перевод дат в локальное время берет на себя конструктор класса *Date*. До этого, конструктор данного класса некорректно переводил время, так как дата, приходящая с серверной части в виде строки, не содержала в конце «Z», из-за чего конструктор полагал, что полученная дата уже в локальном времени. Проблема была решена выставлением значения *DateTimeZoneHandling.Utc* при создании сериализатора на бэкенде. При отправке даты на серверную часть автоматически вызывается метод *ToISOString*, который переводит ее в UTC;

3.7. Обобщение компонентов

Первой очевидной проблемой реализации компонентов было то, что код для создания задачи повторялся в компонентах *EditTask* и *AddHomework*, разница была лишь в том, что в компоненте *AddHomework*

может создаваться несколько задач сразу. Для решения проблемы реализован компонент *CreateTask*, благодаря которому можно задать все необходимые значения в задаче.

Второй проблемой стало многократное повторение реализации выбора дат во всех компонентах. Решением данной проблемы стала реализация двух компонентов: *PublicationAndDeadlineDates* и *TaskPublicationAndDeadlineDates*, для задач и домашних заданий соответственно (Рис. 5).

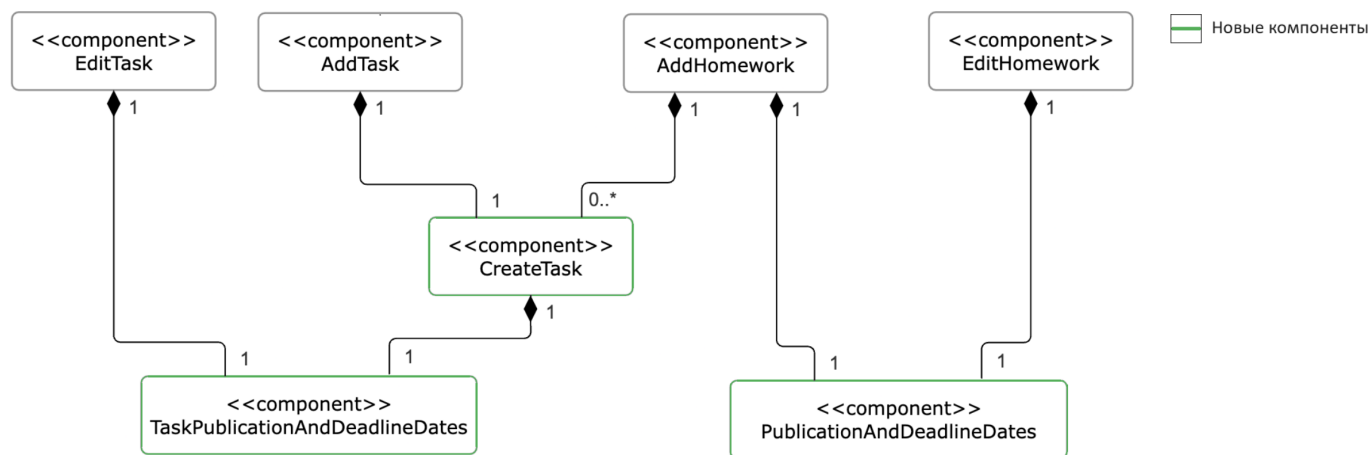


Рис. 5: Отношение компонентов на клиентской части

4. Аprobация

Пулл-реквест с изменениями одобрен для слияния с основной веткой, однако было решено отложить интеграцию до окончания зачетов. Аprobация была проведена на консультанте с целью демонстрации выставления общих дат для задач в домашнем задании в сервисе HwProj 2.0.1.

Заключение

В рамках данной работы были выполнены следующие задачи:

- Спроектирована и реализована функциональность для задания общих свойств задач в домашнем задании на стороне бэкенда HwProj 2.0.1.
- Реализована валидация сущностей домашних заданий и задач, получаемых с клиента.
- Обобщены компоненты, связанные с публикацией и редактированием домашних работ на стороне фронтенда.
- Добавлены новые интеграционные тесты и обновлены существующие.
- Поддержано задание общих свойств для задач в домашнем задании на стороне фронтенда HwProj 2.0.1.
- Исправлена проблема с возможностью изменения курса всеми лекторами.
- Локализовано время для пользователей сервиса.

С кодом реализации можно ознакомиться в пулл-реквесте: <https://github.com/IntelligenceNET/HwProj-2.0.1/pull/320>

Список литературы

- [1] ASP.NET Core. — URL: <https://dotnet.microsoft.com/en-us/learn/aspnet> (дата обращения: 11 декабря 2023 г.).
- [2] AutoFixture. — URL: <https://autofixture.github.io> (дата обращения: 11 декабря 2023 г.).
- [3] C#. — URL: <https://dotnet.microsoft.com/en-us/languages/csharp> (дата обращения: 11 декабря 2023 г.).
- [4] Fluent Validation. — URL: <https://docs.fluentvalidation.net/en/latest> (дата обращения: 11 декабря 2023 г.).
- [5] NUnit. — URL: <https://nunit.org> (дата обращения: 11 декабря 2023 г.).
- [6] RabbitMQ. — URL: <https://www.rabbitmq.com> (дата обращения: 11 2023 г.).
- [7] React. — URL: <https://react.dev> (дата обращения: 11 декабря 2023 г.).
- [8] Typescript. — URL: <https://www.typescriptlang.org> (дата обращения: 11 декабря 2023 г.).
- [9] Официальный сайт MS Teams. — URL: <https://teams.microsoft.com/> (дата обращения: 11 декабря 2023 г.).
- [10] Сайт Blackboard Learn. — URL: <https://bb.spbu.ru/> (дата обращения: 11 декабря 2023 г.).
- [11] Сайт Stepik. — URL: <https://stepik.org/> (дата обращения: 11 декабря 2023 г.).