

# utf-documentation

Пьянков Даниил, 5130904/10101

22 сентября 2024 г.

## Содержание

<b>1</b>	<b>Описание и основной функционал</b>	<b>2</b>
<b>2</b>	<b>High-level архитектура UTF</b>	<b>3</b>
2.1	Описание основных микросервисов . . . . .	3
2.1.1	user-service . . . . .	3
2.1.2	form-service . . . . .	3
2.1.3	assign-service . . . . .	3
2.1.4	answer-service . . . . .	3
2.1.5	report-service . . . . .	3
2.1.6	statistic-service . . . . .	3
2.2	Взаимодействие микросервисов между собой . . . . .	3
<b>3</b>	<b>Архитектура баз данных для каждого микросервиса. Описание таблиц</b>	<b>3</b>
3.1	user-service . . . . .	3
3.1.1	Таблица <i>student</i> . . . . .	3
3.1.2	Таблица <i>group</i> . . . . .	4
3.1.3	Таблица <i>teacher</i> . . . . .	4
3.2	form-service . . . . .	4
3.2.1	Таблица <i>form</i> . . . . .	4
3.2.2	Таблица <i>question</i> . . . . .	5
3.2.3	Таблица <i>answer</i> . . . . .	5
3.3	assign-service . . . . .	5
3.3.1	Таблица <i>assignment</i> . . . . .	5
3.4	answer-service . . . . .	5
3.4.1	Таблица <i>student_answer</i> . . . . .	5
3.5	report-service . . . . .	5

3.6	statistic-service . . . . .	5
4	<b>API микросервисов. Их HTTP-эндпоинты с примерами запросов и ответов</b>	<b>6</b>
4.1	user-service . . . . .	6
4.2	form-service . . . . .	6
4.3	assign-service . . . . .	6
4.4	answer-service . . . . .	6
4.5	report-service . . . . .	6
4.6	statistic-service . . . . .	6
5	<b>Стек технологий</b>	<b>6</b>

# 1 Описание и основной функционал

**UTF** (*University Test Forms*) - это веб-приложение, предоставляющее **преподавателям** следующий функционал:

1. Создавать формы тестирования студентов.
2. Настраивать формы, добавляя в них разделы и вопросы с одним или несколькими вариантами ответов.
3. Указывать временной промежуток, в который студенты могут выполнять тест. Также можно для каждого задания теста отдельно задавать максимальное время на выполнение.
4. Назначать формы тестирования всем студентам из определенных групп.
5. По истечении времени выполнения теста автоматически получать на указанную почту статистику по каждому студенту в виде Excel-отчетов.

Также UTF позволяет **студентам** выполнять следующие действия:

1. Просматривать назначенные им формы тестирования, с возможностью отображения даты дедлайна.
2. Выполнять задания тестов, а также в завершении тестирования видеть свой результат.

Тесты будут открываться у всех студентов одновременно, причем сами задания будут у всех студентов отображаться в случайном порядке. Каждое задание теста будет находиться на отдельной странице, чтобы перейти к следующему заданию, нужно ответить на текущее задание.

Такая логика работы тестирующей системы позволит избежать ситуаций, когда первый студент заходит на тестирование, сохраняет все задания, после чего оставшиеся студенты уже знают задания теста.

Кроме того, зайти на тест с "левого" аккаунта не получится. Все аккаунты, как преподавателей, так и студентов, должны быть верифицированы администраторами, для того, чтобы они могли создавать или выполнять тесты. То есть зайти на тестирование и сохранить задания просто так не получится, нужно будет входить в свой аккаунт, и по честному выполнять тест.

## 2 High-level архитектура UTF

### 2.1 Описание основных микросервисов

**2.1.1 user-service** user-service отвечает за хранение пользовательской информации преподавателей и студентов. Именно в этом микросервисе будет находиться логика регистрации, верификации аккаунта, аутентификации для получения токенов. Также этот сервис будет способен отдавать пользовательскую информацию по идентификатору пользователя.

**2.1.2 form-service** form-service отвечает за создание форм тестирований. Он содержит в себе эндпоинт, принимающий данные об уже сконфигурированной форме тестирования, и сохраняет эти данные в базу, чтобы потом студенты могли получить данные этой формы.

**2.1.3 assign-service** assign-service отвечает за то, чтобы назначать формы тестирования студентам определенных групп. После того, как преподаватель назначает тестирование определенной группе, у всех студентов этой группы в личном кабинете появляется назначенное тестирование, которое необходимо будет выполнить.

**2.1.4 answer-service** answer-service содержит логику сохранения пользовательских ответов. Он принимает в себя идентификатор студента, вопроса, а также ответа студента на вопрос. Далее при завершении тестирования он может определить на сколько вопросов студент ответил правильно.

**2.1.5 report-service** report-service отвечает за формирование Excel-отчетов для преподавателей по результатам прошедшего тестирования. В отчете можно увидеть данные о том, сколько баллов набрал каждый студент (данные сохраняются по группам, один лист - одна группа), а также посмотреть статистику по каждому вопросу, то есть сколько процентов ответили на него правильно / не правильно.

**2.1.6 statistic-service** statistic-service выступает в роли агрегатора данных. По запросу report-service на предоставление ему статистики для формирования отчета statistic-service ходит в assign, answer и form-сервисы для сбора данных о результатах студентов, которые прошли определенный тест. Далее эти данные обрабатываются и рассчитываются необходимые показатели, которые возвращаются обратно в report-service.

### 2.2 Взаимодействие микросервисов между собой

## 3 Архитектура баз данных для каждого микросервиса. Описание таблиц

### 3.1 user-service

**3.1.1 Таблица *student*** В таблице *student* хранятся данные о студентах. Эта таблица имеет следующие колонки:

- id (UUID)
- firstName (varchar)
- lastName (varchar)
- middleName (varchar)

- username (varchar)
- hashed\_password\_bytes (bytea)
- group\_id (FK to group table) (UUID)
- verified (bool)

**3.1.2 Таблица *group*** В таблице *group* хранятся данные о группах студентов. Эта таблица имеет следующие колонки:

- id (UUID)
- institute\_number (varchar)
- group\_number (varchar)

**3.1.3 Таблица *teacher*** В таблице *teacher* хранятся данные о преподавателях. Эта таблица имеет следующие колонки:

- id (UUID)
- firstName (varchar)
- lastName (varchar)
- middleName (varchar)
- username (varchar)
- hashed\_password\_bytes (bytea)
- verified (bool)

## 3.2 form-service

**3.2.1 Таблица *form*** В таблице *form* хранятся данные о созданной преподавателем форме тестирования. Эта таблица имеет следующие колонки:

- id (UUID)
- name (varchar)
- description (varchar)
- teacher\_id (UUID)
- create\_date (timestamp)
- start\_date (timestamp)
- end\_date (timestamp)

**3.2.2 Таблица *question*** В таблице *question* хранятся данные о вопросах на форме тестирования. Эта таблица имеет следующие колонки:

- id (UUID)
- text (varchar)
- description (varchar)
- form\_id (UUID) (FK на form (id))
- time\_amount\_ms (bigint)
- order\_number (integer)

**3.2.3 Таблица *answer*** В таблице *answer* хранятся данные о вариантах ответов на вопросы на форме тестирования. Эта таблица имеет следующие колонки:

- id (UUID)
- text (varchar)
- question\_id (UUID) (FK на question(id))
- correct (bool)

### 3.3 assign-service

**3.3.1 Таблица *assignment*** В таблице *assignment* хранятся данные о том, каким группам назначена определенная форма тестирования. Эта таблица имеет следующие колонки:

- id (UUID)
- form\_id (UUID)
- group\_id (UUID)

### 3.4 answer-service

**3.4.1 Таблица *student\_answer*** В таблице *student\_answer* хранятся данные о том, какие варианты ответа выбрал пользователь на форме тестирования. Эта таблица имеет следующие колонки:

- id (UUID)
- student\_id (UUID)
- question\_id (UUID)
- chosen\_answer\_id (UUID)

**3.5 report-service** Поскольку report-service лишь дергает statistic-service и по полученной статистике формирует Excel-отчет, ему не нужно сохранять никакие данные в БД.

**3.6 statistic-service** Поскольку statistic-service лишь собирает данные с других сервисов, и рассчитывает необходимые показатели, у него нет необходимости сохранять какие-либо данные в БД.

## 4 АРІ микросервисов. Их HTTP-эндпоинты с примерами запросов и ответов

4.1 user-service

4.2 form-service

4.3 assign-service

4.4 answer-service

4.5 report-service

4.6 statistic-service

## 5 Стек технологий