Министерство науки и высшего образования Российской Федерации

**федеральное государственное автономное образовательное учреждение высшего образования**

**«Национальный исследовательский университет ИТМО»**

**(Университет ИТМО)**

**Факультет Программной инженерии и компьютерной техники**

# Общие сведения

Обучающийся: Ореховский Антон P4119

Тема проекта: Реализация контрактного тестирования микросервисов

Тема задания:

Место выполнения проекта: GS Labs

Ментор от организации: Громов П., Пешков А., Шахматов А.

Куратор от университета: Маркина Татьяна Анатольевна

Цель проекта: обеспечить возможность проведения контрактного тестирования микросервисов в предоставленной инфраструктуре.

# Пожелания на проект/Ожидаемые результаты

Здесь и далее:

* Регистратор контрактов (РК) – разрабатываемый сервис, который предоставит возможность осуществления контрактного тестирования
* Поставщик – сервис, который предоставляет данные или функциональность другим сервисам (вне контекста контрактного тестирования)
* Потребитель – сервис, который использует данные или функциональность другого сервиса
* В рамках HTTP: потребитель генерирует HTTP запрос, поставщик генерирует HTTP ответ

Разрабатываемый РК должен:

1. регистрировать контракты поставщиков,
2. регистрировать контракты потребителей,
3. поддерживать совместимость между контрактами потребителей и поставщиков,
4. следить за эволюцией схемы контрактов,
5. оповещать клиентов об изменениях схемы контракта поставщика,
6. иметь возможности для мониторинга и настройки.

# Задачи на проект

## 1 семестр (сентябрь 2021 - январь 2022)

### 1 эпоха

1. Составление технического задания на проект.
2. Согласование технического задания на проект.

### 2 эпоха

1. Исследование аналогов в области контрактного тестирования с целью выбора наиболее удачных практик применимых к разрабатываемому РК.
2. Исследование возможностей составления контрактов и документирование применимых к разрабатываемому РК возможностей.
3. Исследование возможностей публикации контрактов потребителями и поставщиками с целью их дальнейшего сравнения и выбора наиболее применимой к разрабатываемому РК.
4. Получение знаний о возможностях развертывания контейнеров/кластеров.

### 3 эпоха

1. Исследование возможных видов запросов от потребителя к поставщику и их документация.
2. Исследование способов сопоставления контрактов поставщика и потребителя и разработка алгоритма, который будет применен в разрабатываемом РК для сопоставления контрактов.
3. Проектирование концепта DSL для описания контрактов.
4. Исследование и документирование возможностей интеграции РК с элементами CI/CD на примере GitLab Pipeline.
5. Исследование возможностей генераций OpenAPI схем.

### 4 эпоха

1. Разработка архитектуры сервиса в виде UML диаграмм.
2. Документирование предполагаемых конечных точек РК.
3. Составление структуры БД и ее представления в виде ER-модели.
4. Согласование набора технологий.
5. Документирование вариантов использования разрабатываемого сервиса.

## 2 семестр (февраль 2022 - июнь 2022)

Общая цель на семестр – прототип сервиса в программном окружении учащегося с возможностью публиковать схемы поставщика, контракты потребителя, следить за изменением API, оповещать об устаревшем API.

### 1 эпоха

1. Составление технического задания на проект.
2. Согласование технического задания на проект.
3. Документирование сценариев добавления, обновления, удаления и сопоставления контрактов и схем в виде диаграммы последовательностей
4. Изучение форматеров и линтеров:
   1. Форматеры – black, isort
   2. Линтеры – flake8, flake8-bugbear, flake8-pytest-style, flake8-quotes
5. Внедрение форматеров и линтеров в разрабатываемый сервис
6. Изучение библиотек тестирования кода – pytest, pytest-cov
7. Внедрение библиотек тестирования кода в разрабатываемый сервис
8. Изучение видов типизаций в python, внедрение типизации mypy в разрабатываемый сервис
9. Изучение подхода “conventional commits” и его внедрение в разрабатываемый сервис

### 2 эпоха

1. Создание алгоритма разграничения доступа на основе встроенных примитивов языка python (например, словарь)
2. Создание конечной точки сервиса для публикации контрактов
3. Создание конечной точки сервиса для обновления контрактов
4. Создание конечной точки сервиса для удаления контрактов
5. Создание конечной точки сервиса для публикации схем
6. Создание конечной точки сервиса для обновления схем
7. Создание конечной точки сервиса для удаления схем
8. Создание конечной точки сервиса для пометки схемы устаревшей
9. Создание конечной точки сервиса для пометки поля схему устаревшей
10. Тестирование работоспособности сервиса на предмет операций CRUD со схемами и контрактами (покрытие тестами не менее 90% кода)

### 3 эпоха

1. Тестирование алгоритма сопоставления схем и контрактов (покрытие тестами не менее 90% кода)
2. Модернизация алгоритма сопоставления схем и контрактов с учетом ссылочных значений
3. Модернизация алгоритма сопоставления схем и контрактов с учетом различных видов описания файлов (yaml, json)
4. Повторное тестирование алгоритма сопоставления схем и контрактов (покрытие тестами не менее 90% кода)
5. Подключение базы данных для хранения контрактов и схем
6. Модернизация алгоритма разграничения доступа с использованием специально создаваемых токенов

### 4 эпоха

1. Создание алгоритма отслеживающего версионирование схем и контрактов
2. Тестирование алгоритма отслеживающего версионирование схем и контрактов (покрытие тестами не менее 90% кода)
3. Модернизация алгоритма операций CRUD для схем и контрактов с учетом алгоритма версионирования и подключенной базы данных
4. Тестирование алгоритма операций CRUD для схем и контрактов (покрытие тестами не менее 90% кода)
5. Исследование способов оповещения потребителей об используемой ими устаревшей версии схемы
6. Создание алгоритма оповещения об устаревшей схеме
7. Тестирование алгоритма оповещения об устаревшей схеме (покрытие тестами не менее 90% кода)

## 3 семестр (сентябрь 2022 - январь 2023)

Общая цель на семестр – рабочая альфа-версия решения, полная интеграция с элементами GitLab Pipeline.

### 1 эпоха

*Планирование задач на семестр*

### 2 эпоха

### 3 эпоха

### 4 эпоха

## 4 семестр (февраль 2023 - июнь 2023)

Общая цель на семестр – исправление недочетов, документирование, приемочное тестирование, интеграция разработанного решения в программном окружении «заказчика» (в инфраструктуре компании, за которой я закреплен).

### 1 эпоха

*Планирование задач на семестр*

### 2 эпоха

### 3 эпоха

### 4 эпоха

# Требования к проекту

## 1. Требования к внешним интерфейсам

### 1. Интерфейсы пользователя

Доступ к РК осуществляется посредством веб-приложения, которым могут пользоваться две группы:

* Администраторы. Интерфейс администратора должен иметь раздел мониторинга и конфигурации сервиса, раздел просмотра опубликованных API поставщиков и их версий, а также раздел опубликованных контрактов потребителей.
* Клиенты сервиса, то есть поставщики/потребители. Интерфейс должен иметь раздел просмотра опубликованных API поставщиков и их версий, а также раздел опубликованных контрактов потребителей.

### 2. Интерфейсы аппаратного обеспечения

### 3. Интерфейсы программного обеспечения

Доступ к РК сервисов потребителей/поставщиков осуществляется посредством REST API с использованием HTTP+JSON.

### 4. Интерфейсы взаимодействия

## 2. Функциональные требования

1. РК должен регистрировать в нем поставщиков с использованием токенов (например, GitLab tokens)
2. РК должен регистрировать в нем потребителей с использованием токенов (например, GitLab tokens)
3. РК должен предоставлять возможность снимать с регистрации потребителей
4. РК должен предоставлять возможность снимать с регистрации поставщиков
5. РК должен предоставлять возможность поставщикам публиковать свой API в унифицированном виде
6. РК должен предоставлять возможность публиковать контракты, описывающие планируемое использование API поставщика, в унифицированном виде
7. При появлении новой версии API поставщика необходимо оповестить всех потребителей, которые опубликовали контракты с использованием данного API, о данном событии
8. В случае, если опубликованный контракт использует устаревшую версию API, необходимо оповестить об этом потребителя, который опубликовал данных контракт (должно отобразиться в CI/CD Pipeline на стороне потребителей)
9. В случае, если при тестировании на стороне поставщика будет выявлено, что текущая версия API не может удовлетворить запросы потребителей, которые описаны в опубликованных контрактах, необходимо оповестить об этом поставщика (должно отобразиться в CI/CD Pipeline на стороне поставщика)
10. В случае, если API изменился либо расширился, но сохранил ту же версию, необходимо оповестить об этом все контракты, которые используют этот API

(должно отобразиться в CI/CD Pipeline на стороне потребителей)

1. РК должен предоставлять возможность поставщикам помечать опубликованные версии схем устаревшими.

## 3. Требования к производительности

Сервис не является высоконагруженным, но с высокой доступностью.

На конфигурации Intel Core i7 2.5 GHz + 12 Гб ОЗУ необходимо обеспечить 50 одновременных соединений с конвейерами CI/CD.

## 4. Требования к юзабилити

Отсутствуют

## 5. Требования к безопасности

Доступ к веб-приложению необходимо разграничить путем использования токенов для аутентификации (GitLab tokens)

## 6. Требования к данным и их источникам

Основными данными являются контракты, которые необходимо описывать в унифицированном стиле, который совместим с OpenApi.

Источники – поставщики и потребители.

## 7. Требования к логической структуре БД

БД должна являться реляционной. Хранить в ней необходимо OpenAPI схемы поставщиков и их изменения, контракты потребителей, а также токены для аутентификации пользователей веб-приложения

## 8. Проектные ограничения (и ссылки на стандарты)

Спецификация OpenAPI: <https://swagger.io/specification/>

## 9. Нефункциональные требования (надежность, доступность, безопасность и пр.)

1. РК должен хранить состояния всех API
2. Контракты необходимо описывать при помощи DSL (совместимым с OpenAPI)
3. Доступ 24/7
4. Интеграция с GitLab Pipeline

## 10. Требования к обслуживанию системы на протяжении ее жизненного цикла/Требования к эксплуатации

Отсутствуют

## 11. Другие требования

Отсутствуют

# Пользовательские сценарии

Отсутствуют

# Тестирование и проверка

Сквозное (e2e) тестирование с использованием и поставщиков, и потребителей сервиса.

Тестирование интерфейса приложения

* Тестирование возможности администрирования РК
* Тестирование возможности просмотра OpenAPI схем/контрактов поставщиками/потребителями
* UX тестирование

## Планируемые курсы к изучению

* Программирование и разработка веб-приложений (<https://openedu.ru/course/ITMOUniversity/PWADEV/>)
* Программирование и разработка веб-приложений. Часть 2 (<https://openedu.ru/course/ITMOUniversity/PWADEV2/>)
* Python FTW: Under the Hood <https://www.educative.io/courses/python-ftw-under-the-hood>
* Lean Product Management <https://www.educative.io/courses/lean-product-management>
* Devops fro developers <https://www.educative.io/path/devops-for-developers>
* Web Application Security for the Everyday Software Engineer <https://www.educative.io/courses/web-application-security-everyday-software-engineer>
* Web Security and Access Management: JWT, OAuth2 & OpenId Connect <https://www.educative.io/courses/web-security-access-management-jwt-oauth2-openid-connect>
* Python Concurrency for Senior Engineering Interviews <https://www.educative.io/courses/python-concurrency-for-senior-engineering-interviews>
* Learn Advanced Level Programming in Python

<https://www.udemy.com/course/learn-advanced-level-programming-in-python/>

## Планируемые конференции к выступлению

Отсутствуют