



**Курсовой проект  
SMarket**

# Описание проекта

- Проект SMarket представляет собой интернет-магазин товаров повседневного спроса.
- Платформа имеет удобный интерфейс, где все товары разделены по категориям.
- Благодаря простой навигации, пользователю не составит труда определиться с выбором товара.



•Перейдя по интересующей категории, пользователь может ознакомиться со списком товаров, а если товар его заинтересовал он может ознакомиться с подробным описанием.

•Администратор магазина со своей стороны может добавлять товары, изменить цену на товар, назначить или удалить цену, сняв товар с продажи.

•В дальнейшем планируется добавить функционал покупки и резервирования товара.



# Проект SMarket

Welcome to SMarket

Консервы



Конфеты



Алкоголь



Колбасы



Овощи



Хлебобулочные изделия



Фрукты



Сыры



# Панель администратора

Please sign in

Sign in

# Техническая часть

Интернет-магазин SMarket - проект, основанный на микросервисной архитектуре. Каждый микросервис выполняет определённый функционал. Проект включает в себя пять сервисов взаимосвязанных между собой через REST, а также через брокер сообщений Apache Kafka.

- Сервис обеспечивающий работу с товарами (**Stock-Service**)
- Сервис отвечающий за ценообразование товаров (**Price-Service**)

- Сервис обеспечивающий работу с авторизацией пользователей (**User-Service**)
- Фронтное приложение для вывода товаров (**Market-Front-App**)
- Фронтное приложения для администрирования цен, добавления/удаления товаров (**Admin-Front-App**)

# Stock-service

- Spring Boot приложение с подключением к базе данных PostgreSQL .
- Сервис хранит описание товаров: цена, название, категория, количество, производитель и др. Позволяет создавать, редактировать, резервировать, удалять товары, получать информацию о товарах.
- Подключение к базе данных происходит через стандарт JDBC. Структура базы данных создается при помощи Liquibase.
- Данный сервис использует программного брокера сообщений Apache Kafka для оперативного уведомления об удалении, добавлении, изменении состояния товара.

# API сервиса

- GET /stock/item?itemId={itemId}
- GET /stock/item/all/byCategory?categoryId={categoryId}
- GET /stock/Category/all
- GET /stock/item/all
- POST /stock/item
- POST /stock/reserve?itemId={itemId}&count={count}
- POST /stock/unreserve?itemId={itemId}&count={count}
- PUT /stock/item
- DELETE /stock/item?itemId={itemId}

Ссылка на репозиторий:

- //TODO добавить ссылку на репозиторий

# Price-service

- Сервис отвечающий за ценообразование товаров.
- Spring Boot приложение с подключением к базе данных PostgreSQL с использованием библиотеки Hibernate.
- Структура базы данных создается при помощи Liquibase.
- Данный сервис использует программного брокера сообщений Apache Kafka для оперативного уведомления об удалении, добавлении, изменении состояния цены на товар.

# API сервиса

- GET /price?itemId={itemId}
- DELETE /price?itemId={itemId}
- PUT /price/all
- GET /price/all
- PUT /price

Ссылка на репозиторий:

- //TODO добавить ссылку на репозиторий

# User-service

- Сервис представляет собой Spring Boot приложение, которое обеспечивает работу с учётными записями пользователей.
- Информация о пользователях хранится в базе данных PostgreSQL. Взаимодействие с базой данных осуществляется посредством Spring Data.

# API сервиса

- GET /auth/user/byUserId?userId={userId}
- GET /auth/user/byLogin?login={login}
- DELETE /auth/user/user?userId={userId}
- POST /auth/user
- PUT /auth/user
- GET /auth/user/all
- GET /auth/user/all/byRoles?roles={roles}

Ссылка на репозиторий:

- //TODO добавить ссылку на репозиторий

# Market-front-app

- Фронтное приложение реализует пользовательский интерфейс и позволяет клиенту ознакомиться со списком товаров. Источником данных является stock-service и price-service, взаимодействие осуществляется через REST.
- Данный сервис оперативно реагирует на изменение списка товаров и цен на них, вычитывая сообщения из Apache Kafka;

# API сервиса

- GET market/category/all
- GET market/item/all/byCategory?categoryId={categoryId}

Ссылка на репозиторий:

- //TODO добавить ссылку на репозиторий

# Admin-front-app

- Фронтовое приложения для администрирования цен, добавления/удаления товаров. Источником данных является stock-service и price-service, взаимодействие осуществляется через REST.
- Данный сервис оперативно реагирует на изменение списка товаров и цен на них, вычитывая сообщения из Apache Kafka;
- Вход в административную панель осуществляется с использованием Spring Security.

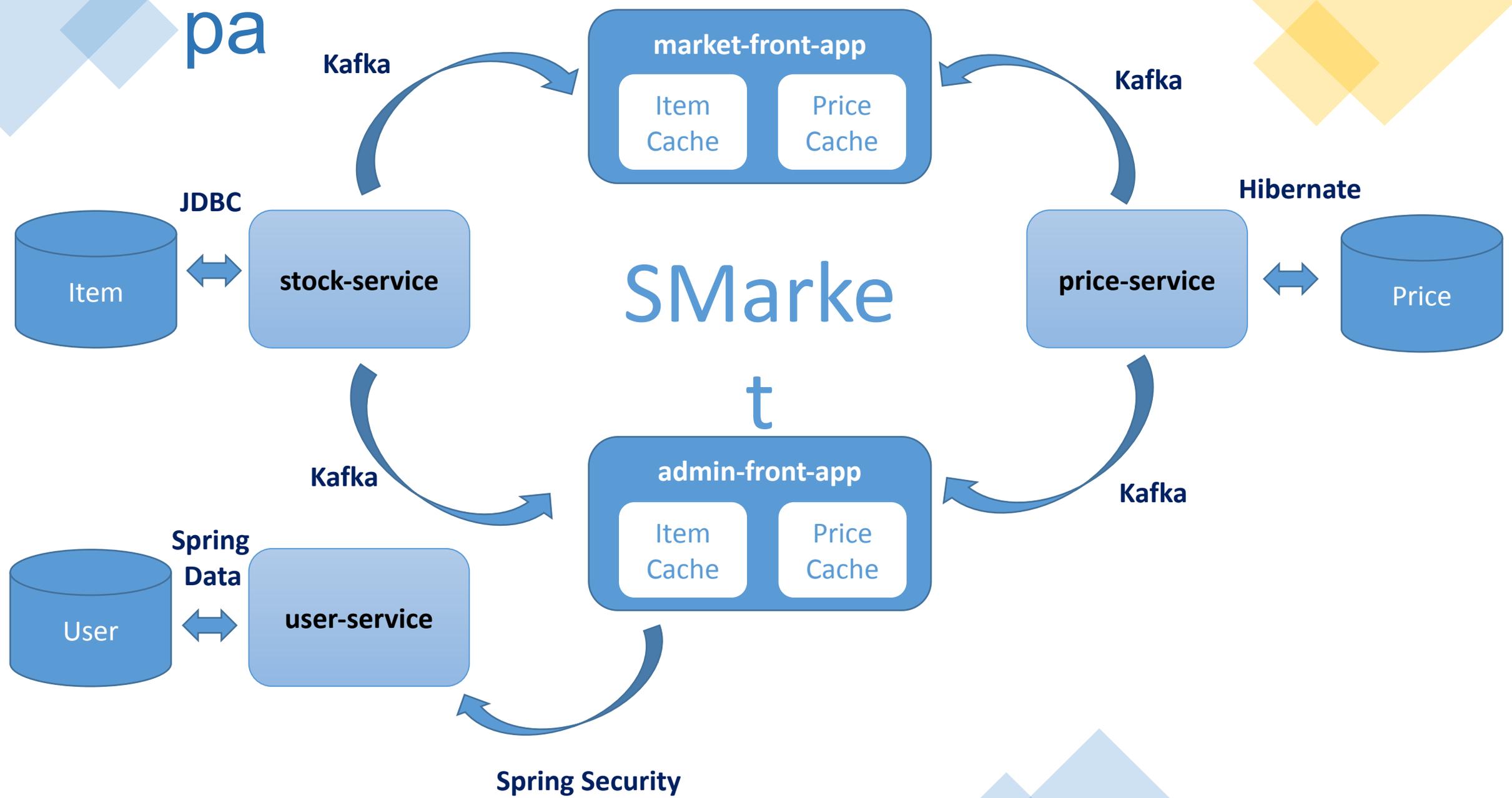
# API сервиса

- GET admin/item/all
- PUT admin/item/price?itemId={itemId}&price={price}
- DELETE admin/item/price?itemId={itemId}
- POST admin/item

Ссылка на репозиторий:

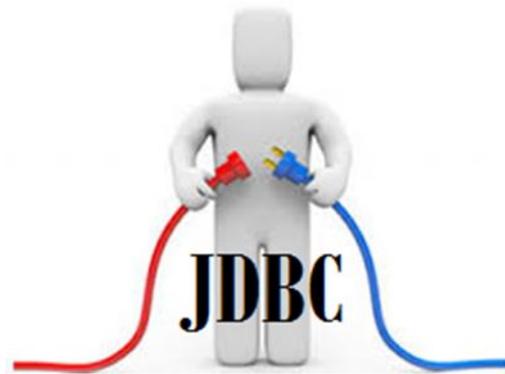
- //TODO добавить ссылку на репозиторий

# Архитектура



# Используемые технологии

- JDBC
- Liquibase
- Kafka
- JUnit
- Mockito
- Git



- Spring Boot
- Spring MVC
- Spring Rest Template
- Spring Data
- Spring Security
- Hibernate



**SPRING MVC**

Create  
Read  
Update  
Delete



**Spring Rest Template**



**Spring Data**

# ДЕМО

//TODO адрес сервера - Admin-front-app

//TODO адрес сервера - Market-front-app

# Команда проекта

//TODO перечислить участников проекта  
Куратор проекта Мартынец Александр



# ФИО участника

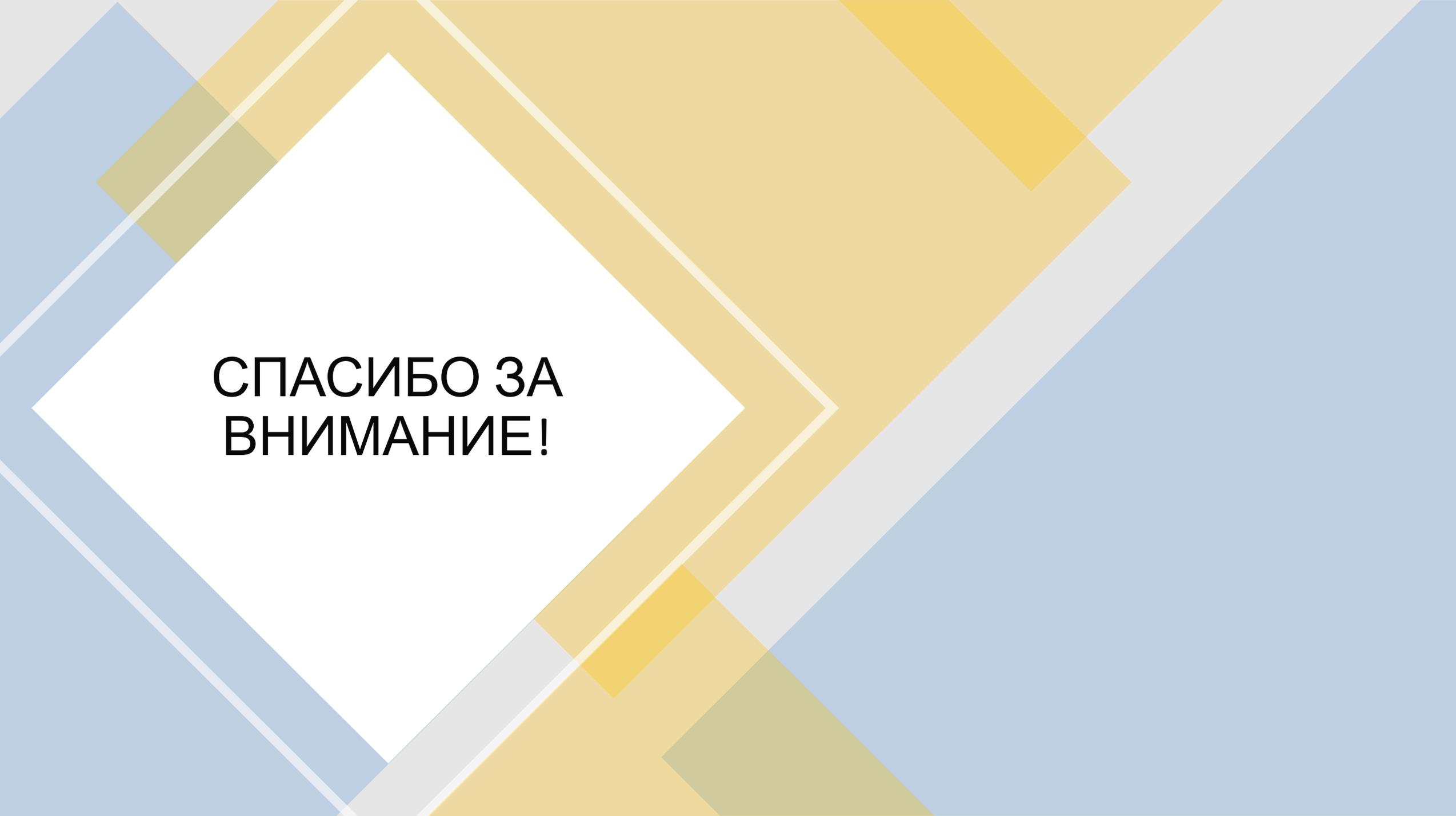
//TODO коротко о себе (образование, опыт, увлечение и пр., опыте работы в Сбере и контакты (слева место для фото))

# Участие в проекте «SMarket»

- //TODO что сделал для проекта
- //Example User-service. Назначен ответственным за этот сервис. Мной реализован API для получения списка всех пользователей, соответствующий ему слой бизнес логики и слой доступа к данным.

# О проекте ПЕРЕЗАГРУЗКА

- //TODO Отзыв участника о проекте



**СПАСИБО ЗА  
ВНИМАНИЕ!**