**Министерство образования и науки Российской Федерации**

федеральное государственное автономное образовательное учреждение высшего образования

**“НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ**

**УНИВЕРСИТЕТ ИТМО**”

**Факультет программной инженерии и компьютерной техники**

**ЛАБОРАТОРНАЯ РАБОТА № 3**

**ПО ДИСЦИПЛИНЕ «ТЕХНОЛОГИИ ВЕБ-СЕРВИСОВ»**

Студент: Носков Федор Андреевич

Группа: P41142

Преподаватель: Дергачев Андрей Михайлович

\

Санкт-Петербург

2020

1.Техническое задание

Основываясь на информации из раздела 2.8, добавить поддержку обработки ошибок в сервис реализованный во второй лабораторной работе.

В соответствии с изменениями сервиса необходимо обновить и клиентское приложение.

2.Описание работы

Сервер написан на ejb, с использованием jdbc для подключения к базе данных. В качестве базы данных используется MySQL. Таблица использованная в лабораторной работе называется Lab1 и состоит из 5 полей: Id: int, Fild1: varchar(20), Fild2: varchar(20), Fild3: int, Fild4: int.

Было добавлено два класса ошибок: BadFormatException – ошибка, возникающая при несовпадении форматов полей и IdNotFoundException – ошибка, которая возникает, когда клиент пытается удалить или обновить запись с id которого не существует.

Лабораторная работа и консольный клиент расположены на GitHub по адресу: https://github.com/xim123456/WebLabs

3.Вопросы на защиту

1. Предположим, при реализации SOAP веб-сервиса Вам необходимо передавать двоичные данные (аватары пользователей, архивы...). Какие стратегии Вы можете предложить для работы с binary attachments? Реализуйте одну из них.

Существует две стратегии передачи binary attachments в SOAP. Первая это создать метод с аргументом массив байт и передавать данные массивом. Но этот способ не очень удобный из-за того, что так можно передавать файлы только небольшого размера.

Второй способ — это воспользоваться механизмом MTOM. Можно в место файла в аргументах отправить специальный идентификатор, а сами данные отправить в конце SOAP сообщения.

Так как второй способ гораздо практичнее им я и воспользовался.

1. Для чего нужны аннотации @Produces и @Consumes при реализации REST-сервиса?

Аннотации описываю с помощью каких MIME-типов будут общаться сервер и клиент.

Аннотация @Produces описывает какой MIME-тип будет получен клиентом от сервера при обращении к методу.

Аннотация @Consumes описывает какой MIME-тип будет получен сервером от клиента при обращении к методу.

1. Модифицируйте REST и SOAP сервисы так, чтобы для CREATE, DELETE, UPDATE операций требовалась Basic аутентификация. Соответственно внесите такие же изменения в клиентские приложения. Login / Password для простоты можно захардкодить.

Для того чтобы реализовать Basic аутентификацию в SOAP сервисе нужно чтобы клиент в каждом своём сообщении отправлял в заголовке сообщения пароль и логин, а сервер проверял их, получая из контекста пришедшего сообщения. Если логин или пароль не совпадают, то сервер выбрасывает исключение.

Для того чтобы реализовать Basic аутентификацию в REST сервисе нужно чтобы клиент в каждом своём сообщении отправлял в заголовке сообщения заголовок с именем authorization, который бы представлял из себя закодированное сообщения вида “логин:пароль”. Сервер в свою очередь должен в при каждом обращении доставать этот заголовок из контекста сообщения, расшифровывать и проверять. Если логин или пароль не совпадают, то сервер выбрасывает исключение.

1. Предположим, что одним из нефункциональных требований к вашему сервису является поддержка throttling (умышленное ограничение количества одновременно выполняемых запросов; если поступает новый запрос, а в это время уже выполняется максимально разрешенное количество, то необходимо прервать выполнение запроса, например, путем выброса ThrottlingException (класс исключения нужно создать самостоятельно)). Как можно это реализовать?

Ответ тот же что и в вопросе 4.

1. Реализуйте endpoint, позволяющий юзеру загружать файлы (для простоты не нужно делать никаких проверок корректности имени файлов, дубликатов файлов, итп). Также реализуйте клиентскую часть, взаимодействующую с сервисом.

Так как файлы это binary attachments можно воспользоваться решением предложенным в вопросе 5 и использовать MTOM.

1. В чем принципиальное различие при реализации обработки ошибок в REST и SOAP-сервисах с точки зрения разработки сервиса, а также с точки зрения реализации клиента?

В SOAP в отличие от REST есть стандартизация исключений. Исключение в SAOP должны быть помечены специальной аннотацией, иметь специальные конструкторы и реализовывать два метода getFaultInfo() и getMessage(). Отправлять исключения нужно через специальный bean.

В REST ничего этого не нужно. Можно воспользоваться стандартными исключениями или написать своё унаследовав от Exception и написать mapper для того чтобы отправлять их.

Для клиента SOAP отличается от REST тем что исключения в SOAP выбрасывает метод, который принимает сообщения от сервера, определив его по ответу сервера. В REST о том, что случилось можно судить по коду ответа от сервера и тексту сообщения.