**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 № 6**

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

Студент: Калинин Даниил Леонидович

Группа: P41141

Преподаватель: Дергачев Андрей Михайлович

\

Санкт-Петербург

2021

Задание:

Необходимо выполнить задание из третьей работы, но с использованием REST-сервиса. Таблицу базы данных, а также код для работы с ней можно оставить без изменений.

Этапы выполнения:

Для выполнения задания было решено изменить реализацию исключений, созданных в третьей лабораторной работе.

1. Исключение для пустого запроса

public class EmptyRequestException extends PersonCRUDException {  
  
 public EmptyRequestException(String message) {  
 super(message);  
 }  
}

@Provider  
public class EmptyRequestExceptionMapper implements ExceptionMapper<EmptyRequestException> {  
  
 @Override  
 public Response toResponse(EmptyRequestException e) {  
 return Response.*status*(400).entity(e.getMessage()).build();  
 }  
}

1. Исключение для несуществующего человека

public class PersonDoesNotExistException extends PersonCRUDException {  
  
 public static PersonDoesNotExistException *DEFAULT\_INSTANCE* = new  
 PersonDoesNotExistException("Person with given parameters does not exist");  
  
 public PersonDoesNotExistException(String message) {  
 super(message);  
 }  
}

@Provider  
public class PersonDoesNotExistExceptionMapper implements ExceptionMapper<PersonCRUDException> {  
  
 @Override  
 public Response toResponse(PersonCRUDException e) {  
 return Response.*status*(404).entity(e.getMessage()).build();  
 }  
}

1. Внутренние исключения, такие как ошибка конвертации объекта в SQL-запрос, наследуются от объединены единым маппером и наследуются от ServerException. Такие исключения возвращают ответ с кодом ошибки 500

public class ServerException extends Exception {  
  
 public static ServerException *DEFAULT\_INSTANCE* = new  
 ServerException("Internal server error");  
  
 public ServerException(String message) {  
 super(message);  
 }  
}

public class SQLConvertException extends ServerException {  
  
 public static SQLConvertException *DEFAULT\_INSTANCE* = new  
 SQLConvertException("Internal server error");  
  
 public SQLConvertException(String message) {  
 super(message);  
 }  
}

@Provider  
public class ServerExceptionMapper implements ExceptionMapper<ServerException> {  
  
 @Override  
 public Response toResponse(ServerException e) {  
 return Response.*status*(500).entity(e.getMessage()).build();  
 }  
}

На стороне клиента в случае, если код ответа отличается от 200 – OK, вызывается исключение с сообщением из тела ответа

private void processError(ClientResponse response){  
 String x = response.getEntity(String.class);  
 throw new IllegalStateException(x);  
}

Вывод

В данной лабораторной работе была организована обработка исключений на стороне сервера и отправка клиенту соответствующих им ответов с кодами ошибок и сообщениями о проблеме.

Вопросы:

Ссылка на GitHub:

https://github.com/KalininDL/web\_services\_spring