Федеральное государственное автономное образовательное учреждение высшего образования

«**Университет ИТМО»**

**Факультет ПИиКТ**

**Дисциплина: Технологии веб-сервисов**

**Лабораторная работа 6**

Выполнили: Гурин Евгений Иванович, Камышанская Ксения Васильевна

Преподаватель: Дергачев Андрей Михайлович

Группа: Р4216

Санкт-Петербург 2024г.

# Задача

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.

# Выполнение

В данной лабораторной работе были добавлены классы Exception, аналогичные ЛР 3, но с другой структурой. Кроме того были добавлены Mapper классы для преобразования ошибок к JSON формату

|  |
| --- |
| /\*\*  \* Exception for validation.  \*/  @Getter  public class ValidationFailed extends Exception {  /\*\*  \* Info about fields errors.  \*/  private final FieldValidationFault faultInfo;  /\*\*  \* Constructor.  \*  \* @param message The error message  \* @param faultInfo additional info  \*/  public ValidationFailed(final String message, final FieldValidationFault faultInfo) {  super(message);  this.faultInfo = faultInfo;  }  /\*\*  \* Constructor.  \*  \* @param message The error message  \* @param faultInfo additional info  \* @param cause exception cause  \*/  public ValidationFailed(final String message, final FieldValidationFault faultInfo, final Throwable cause) {  super(message, cause);  this.faultInfo = faultInfo;  }  /\*\*  \* Default constructor.  \*/  public ValidationFailed() {  super(ErrorMessages.VALIDATION\_FAILED);  this.faultInfo = FieldValidationFault.builder().build();  }  /\*\*  \* Constructor from ConstraintViolationException.  \*  \* @param exception original validation exception  \*/  public ValidationFailed(final ConstraintViolationException exception) {  super(ErrorMessages.VALIDATION\_FAILED);  this.faultInfo = new FieldValidationFault(exception);  }  /\*\*  \* Constructor.  \*  \* @param faultInfo additional info  \*/  public ValidationFailed(final FieldValidationFault faultInfo) {  super(ErrorMessages.VALIDATION\_FAILED);  this.faultInfo = faultInfo;  }  } |

|  |
| --- |
| /\*\*  \* ExceptionMapper for ValidationFailed.  \* Maps an ValidationFailed exception to json response.  \*/  @Provider  public class ValidationMapper implements ExceptionMapper<ValidationFailed> {  /\*\*  \* {@inheritDoc}  \*/  @Override  public Response toResponse(final ValidationFailed e) {  return Response.status(Response.Status.BAD\_REQUEST)  .entity(ErrorDTO.builder()  .code(ErrorCode.VALIDATION\_FAILED)  .message(e.getMessage())  .detail(e.getFaultInfo())  .build())  .type(MediaType.APPLICATION\_JSON)  .build();  }  } |

Mapper классы необходимо было зарегистрировать в RestApplication классе

|  |
| --- |
| /\*\*  \* Rest application configuration.  \*/  @ApplicationPath("/api")  @OpenAPIDefinition(servers = @Server(url = "/lab6-server"))  public class RestApplication extends Application {  /\*\*  \* The logger.  \*/  private static final Logger LOGGER = LogManager.getLogger(RestApplication.class);  /\*\*  \* Provide services and mappers classes for app.  \* @return classes set.  \*/  @Override  public Set<Class<?>> getClasses() {  return Stream.of(  CityResource.class,  OpenApiResource.class,  ThrowableMapper.class,  NotAllowedMapper.class,  EntryNotFoundMapper.class,  FieldIsNotFilterableMapper.class,  ValidationMapper.class)  .collect(Collectors.toSet());  }  /\*\*  \* Post init.  \*/  @PostConstruct  public void init() {  LOGGER.info("init RestApplication");  }  } |

Кроме того был обновлен класс преобразования ошибок в консольном клиенте

# Вывод

В данной лабораторной работе были добавлены классы Exception и Mapper для преобразования в JSON в рамках REST сервиса