

ьм в ъ мм ъ м м ж ъ ьм мв ъ ч в
ж ъ мв

Университет ИТМО»

Факультет ПИИКТ

Дисциплина: Технологии веб-сервисов

Лабораторная работа 6

мв з зм дѣ мѣи ъ м ъз ч м мв з ы ъм
ъ ы в ъ м и зф и ъз
п

м в вѣ

Задача

ф з б мв ыж мв зж ы и м
з ьж б мв бз : з ж мм ф
м и м бз ы жзж м мви:

Выполнение

мм и м и з б м
м вз м п м в и и: в з б м
ж б мв ч з м

```
/**
 * 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();
    }
}

```

М ϕ з ж вэ з δ ыδ

```
/**
 * 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");
    }
}
```

в м δ м ж δ мв ч з δ м ьм
з м

Вывод

мм и м и з δ м з
ж δ мв δ Мδ ϕ δз