

Федеральное государственное автономное образовательное  
учреждение высшего образования

Университет ИТМО

Дисциплина: Технологии веб-сервисов

## **Лабораторная работа 6**

**Выполнили:**

Кривоносов Егор Дмитриевич

**Группа:** Р4214

**Преподаватель:**

Сафронов Андрей Геннадьевич

2025 г.

Санкт-Петербург

# Оглавление

Техническое задание	3
Постановка задачи	3
Этапы выполнения	3
Комментарии к архитектурным и функциональным аспектам реализации	3
Ссылка на код	4
Вывод	5

# Техническое 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.

## Постановка задачи

Разработать систему обработки исключений для REST-сервиса, работающего с сущностью `Person`. Система должна:

- Перехватывать исключения бизнес-логики (`PersonServiceException`) и формировать читаемые ответы в формате JSON.
- Возвращать статус HTTP 400 (Bad Request) при ошибках, вызванных некорректными запросами.
- Использовать объект `FaultBean` для передачи детализированной информации об ошибке.

## Этапы выполнения

1. Создание класса исключения
2. Создание объекта для информации об ошибке
3. Реализация маппера исключений
4. Регистрация маппера в конфигурации сервера
5. Доработка клиента

## Комментарии к архитектурным и функциональным аспектам реализации

Реализован класс `PersonServiceException`, который наследуется от `RuntimeException` и содержит поле `FaultBean` для хранения информации об ошибке.

```
public class PersonServiceException extends RuntimeException {
    private final FaultBean faultInfo;

    public PersonServiceException(String message, FaultBean faultInfo) {
        super(message);
        this.faultInfo = faultInfo;
    }
}
```

```

public PersonServiceException(String message, FaultBean faultInfo, Throwable cause) {
    super(message, cause);
    this.faultInfo = faultInfo;
}

public FaultBean getFaultInfo() {
    return faultInfo;
}
}

```

Класс FaultBean содержит сообщение об ошибке.

```

@Data
@NoArgsConstructor
@AllArgsConstructor
public class FaultBean {
    private String message;
}

```

PersonServiceExceptionHandler перехватывает исключения PersonServiceException и формирует HTTP-ответ с кодом 400.

```

@Provider
public class PersonServiceExceptionHandler implements ExceptionMapper<PersonServiceException> {

    @Override
    public Response toResponse(PersonServiceException exception) {
        return Response.status(Response.Status.BAD_REQUEST)
            .entity(exception.getFaultInfo())
            .build();
    }
}

```

В конфигурацию REST-приложения добавлен вызов регистрации PersonServiceExceptionHandler.

```

config.register(new PersonServiceExceptionHandler());

```

## Ссылка на код

<https://github.com/RedGry/TVS-LABS/tree/lab456>

## Вывод

В рамках работы была реализована система обработки исключений для REST-сервиса. Использование маппера исключений и структуры FaultBean позволило централизовать обработку ошибок и предоставить клиенту понятные ответы. Система соответствует требованиям технического задания и REST-принципам.