# Android

## 8 Lab

1. В чем состоит основная концепция Material Design?

В том, что-бы человек без знаний теории дизайна смог склепать неплохо выглядещее приложение

2. Для чего используется класс Palette?

Инструмент, который помогает создают цветовую палитру сочетаемых друг с другом цветов

3. Как добавить RecyclerView ? Какие есть способы размещение представлений?

RecyclerView можно найти в layouts

4. Для чего и как испольуется ViewHolder?

ViewHolder - отвечает за обработку любых событий, которые происходят на определенном элементе, который отображает RecyclerView

5. Как добавить CardView?

CardView — это виджет, имплементирующий такой элемент дизайна Material, как карточка. По сути это контейнер, у которого можно задавать радиус скругленности углов, цвет карточки и высоту по оси z. Вот так можно её создать: <Android.support.v7.widget.CardView.>

6. Как обеспечить обработку щелчков в RecyclerView?

Через ViewHolder, не помню реализовывал ли я это

## 9 Lab

Суть лабы в привязках UI элементов к данным, то есть.

Вопросы при защите:

1) Поясните что такое

a LiveData - хранилище данных, работающее по принципу паттерна Observer (наблюдатель). Это хранилище умеет делать две вещи:

1) В него можно поместить какой-либо объект

2) На него можно подписаться и получать объекты, которые в него помещают.

Если быть кратким - из-за него работает привязка данных и оно уведомляет все, что привязано к ней, о изменении этих данных

b ViewModel - ViewModel - класс, позволяющий Activity и фрагментам сохранять необходимые им объекты живыми при повороте экрана.

c Dao - Data Access Object. Обьект через который, можно получить доступ к данным (репозиторий old school)

d Repository - (DAO new school)

e Room - (DB framework, что то вроде Entity Framework)

2) Что такое Lifecycle и как он используется? - (Create, Start, Resume, Pause, Stop, Destroy) при создании новой Activity она проходит через все стадии и вызываются соответсвующие методы on + StateName

3) Какие основные свойства LiveData? Как добавить и получить

данные?

LiveData.setValue() и LiveData.getValue(). Про основные свойства вероятно имела ввиду их же, другие мы не использовали

4) Как создать ViewModel? Как связать ее с Activity или Fragment?

создать просто через конструктор класса, который расширяет ViewModel

new MyViewModel(someParams...)

Ниже написано как связать

ViewModelProviders.of(\_\_Activity\_\_).get(\_\_ViewModelClass\_\_);

5) Как ViewModel связана с LiveData? Где их хранить?

Смотри класс CardViewModel

6) Что такое Room, Dao, Entity? - Смотри вопрос 1. Entity - сущность, которая храниться в бд и по которой будет создана таблица

7) Что такое Data Binding? Расскажите как организовать привязку. - Вспоминай WPF и её биндинги, где изменения значения поля обьекта влечет

за собой изменения на UI

8) Как организовать механизмы observable objects, observable fields,observable collection? Для чего они используются?

- Организовать механизмы через кастомные Listener’ы и EventRisers’ы, Если что эти названия и придумал только-что сам. EventRisers хранят в себе ссылку на Listeners и когда у EventRisers вызывается какой-то метод они вызывают Listeners, звучит странно, но поверьте так оно и есть, подойдите и спросите меня об этом если не поняли

9) Как организуется архитектура клиент-серверных приложений Android? Расскажите какие могут быть варианты.

Андроид - клиент, сервер - обычный сервер,

Network операции не могут выполняться в main потоке

## 10 Lab

1. Назвоните основные преимущества языка Kotlin.

Много синтаксического сахара, а по сути никаких плюсов нет

2. Куда может компилироваться Kotlin?

Byte-code

3. Что такое функциональное программирование?

Подход к программированию, где все основано на функциях, а не на других сущностях, к Ивану он знает

4. Расскажите как определить функцию. Должна ли она принадлежать классу?

fun, не обязательно

5. Приведите пример определения функции путем присвоения выражения (тело-выражение), например if, try …

чекайте код

6. Как объявить переменную? Нужен ли тип? В чем отличие val и var?

чекайте код. нет не нужен. var - изменяемое. val - final (const)

7. Можно ли поменять объект, на который указывает val, может быть изменяемым?

нет. да

8. Приведите пример использования конструкции when.

тот же switch. чекайте код

9. Как используются is и as?

как в шарпах. is - instanceof, as - обычный каст одного типа в другой

10. Какие циклы можно использовать в Kotlin?

do while, for, foreach, while

11. Приведите пример задания диапазона. Какие функции можно использовать в интервалах?

1..6, функции можно использовать те же, что и в коллекциях

12. Как происходит обработка исключений?

try catch finally

13. Как определить тип способный хранить null?

int? - вопросительный знак в конце

14. Как и в каких случаях можно использовать операторы ?: !! ?.

!! выполнить в любом случае даже если null.

?. выполнить если не null

15. Различает ли Kotlin примитивные типы и типы-оберток? Как происходит компиляция типов с поддержкой null?

Не различает. В обьекты типа Optional

16. Выполняет ли Kotlin автоматическое преобразования чисел из одного типа в другой?

Да

17. Расскажите про типы Any и Any?

Object, тот же что и в джаве

18. Расскажите про тип Unit и Nothing.

не знаю

19. Как компилируются функции верхнего уровня? Создается ли для них класс?

Создается и укладываются в статический класс.

20. Какова роль функций расширений в Kotlin? Приведите пример такой функции.

Расширить функционал другого класса новыми методами

21. Как определяется свойство-расширения?

fun ClassName.NewMethodName в любом классе

22. Как задать функцию с переменным числом параметров?

fun printStrings(vararg strings: String) - ключевое слово vararg

23. Для чего используются локальный (вложенные) функции? Как их вызвать?

Локальные функции определяются внутри других методов и используются там же. Вызываются как обычные функции, например fun(). Объявляются в методах поскольку вне они не нужны

24. Задайте функциональный тип?

Интерфейс с одним методом. Функциональный интерфейс это так называется в Java.

25. Как используется ключевое слово it в анонимной функции?

fun() { it.SomeFun() }

26. Как задать и использовать ссылку на функцию?

someObject::someMethod

27. Приведите пример использования функций apply, let, run, with, also, takeIf.

Здесь описание - <https://stackoverflow.com/questions/45582732/difference-between-kotlin-also-apply-let-use-takeif-and-takeunless-in-kotlin>

функции называются *функциями области видимости*.

28. Как в Kotlin записываются лямбда – выражения?

it -> { doSmething…}

29. Как создать изменяемую и неизменяемую коллекцию List, Set, Map?

List = [1,2,3,4], Set = {1,2,3,4}, Map = { 1:2, 3:4, 5:6}

30. Как добавить, удалить элементы из коллекций, преобразовать к другому типу?

collection.add() / remove(), map()

31. Приведите пример деструктуризации списка.

Деструктуризация списка - присвоение каждому элементу списка его собственной ссылке

32. Как создать массив значений примитивного типа без оберток

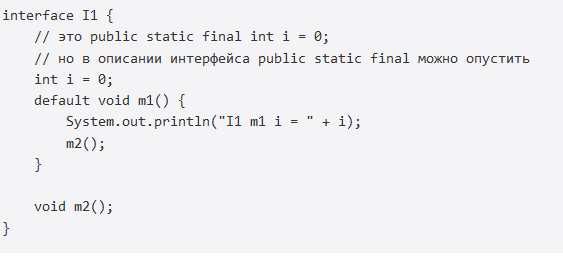
Никак, в котлине нет примитивных типов

## 11 Lab

1. Как определить интерфейс и что он может содержать?

ключевое слово interface, может содержать все то же, что и Java - сигнатуры методов и константы

2. Приведите пример реализации метода интерфейса по умолчанию.



3. Что такое функциональное программирование?

Подход к программированию, где все основано на функциях, а не на других сущностях, к Ивану он знает

4. Когда используются модификаторы open, abstract и final?

Kotlin - Java

open - public static

abstract - abstract

final - final

5. Расскажите о модификаторах видимости.

private, default, protected, public. Как везде, только по умолчанию в котлин public

6. В чем отличие внутреннего и вложенного класса?

внутрений это public class ClassName{}

вложенный это public static class ClassName{}

7. Какие бывают конструкторы?

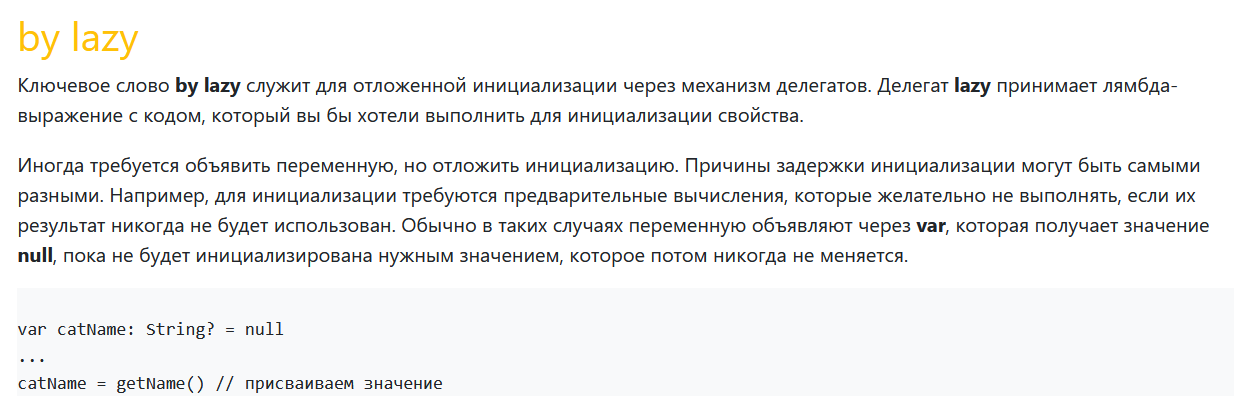
По умолчанию

С параметрами и без параметров

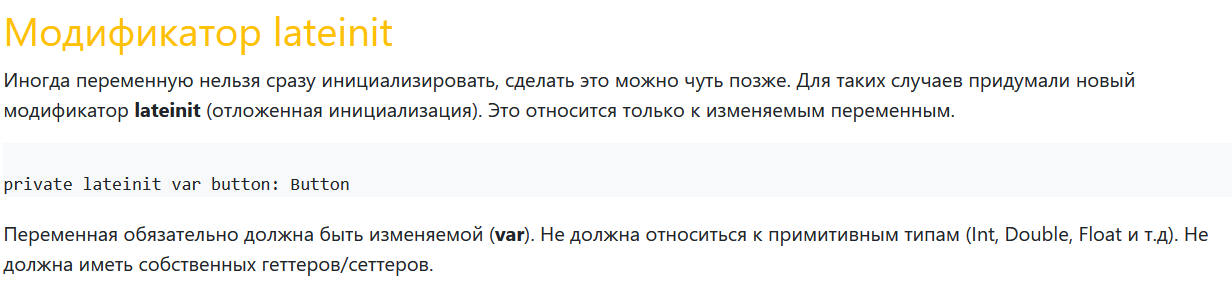
8. Что такое init? В какой последовательности происходит инициализация свойств в классе.



9. Что такое поздняя инициализация?



10. Что такое отложенная инициализация?



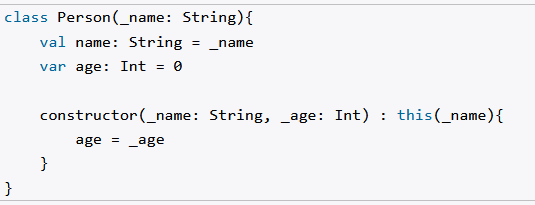
11. Как объявляются свойства класса? (val и var ).

public val или public var

12. Если в классе не определять конструктор, то его не будет вообще?

Будет, по умолчанию

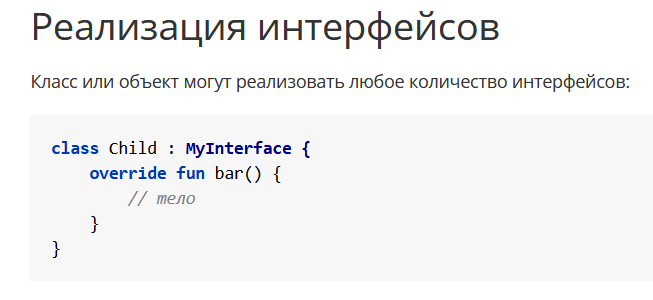
13. Приведите пример вторичного конструктора. Можно и в нем объявить свойства?



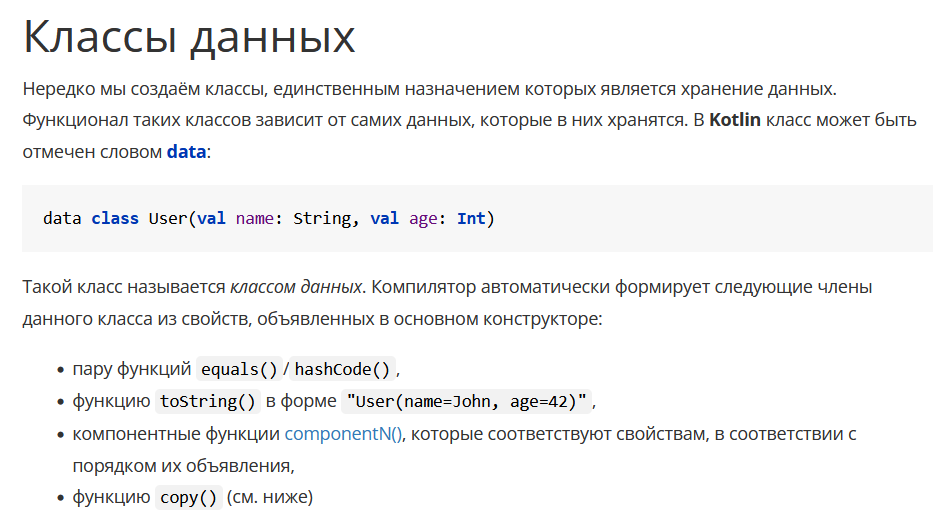
14. Как сделать наследование классов?

Apprentice Class : Ancestor Class

15. Как реализуются свойства интерфейсов?



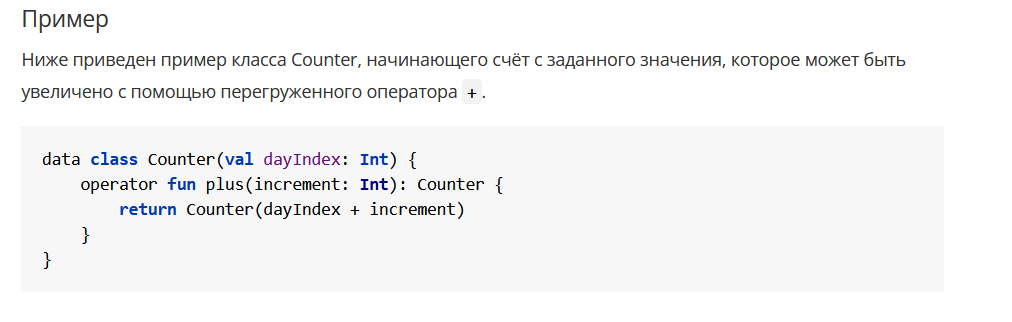
16. Что такое классы данных? Расскажите про автоматическую деструктуризацию. Какие есть ограничения для классов данных.



17. Как используется object в Kotlin? Поясните в каких ситуациях.

Встроенная реализация Singleton

18. Как выполняется перегрузка операций для класса?

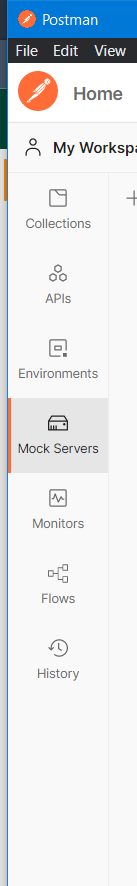
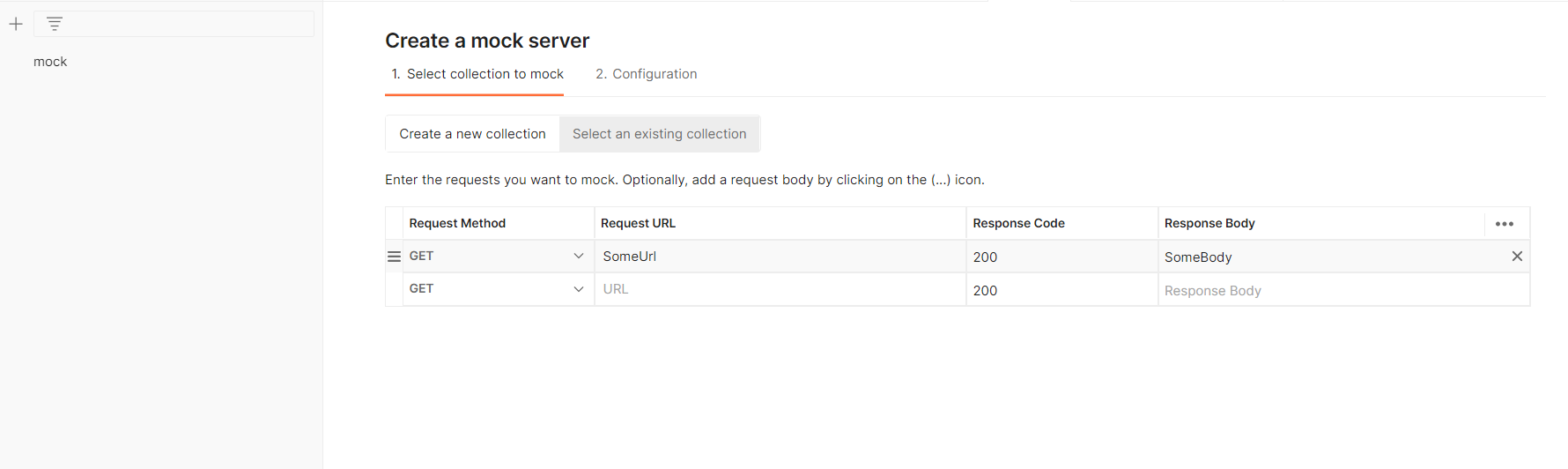
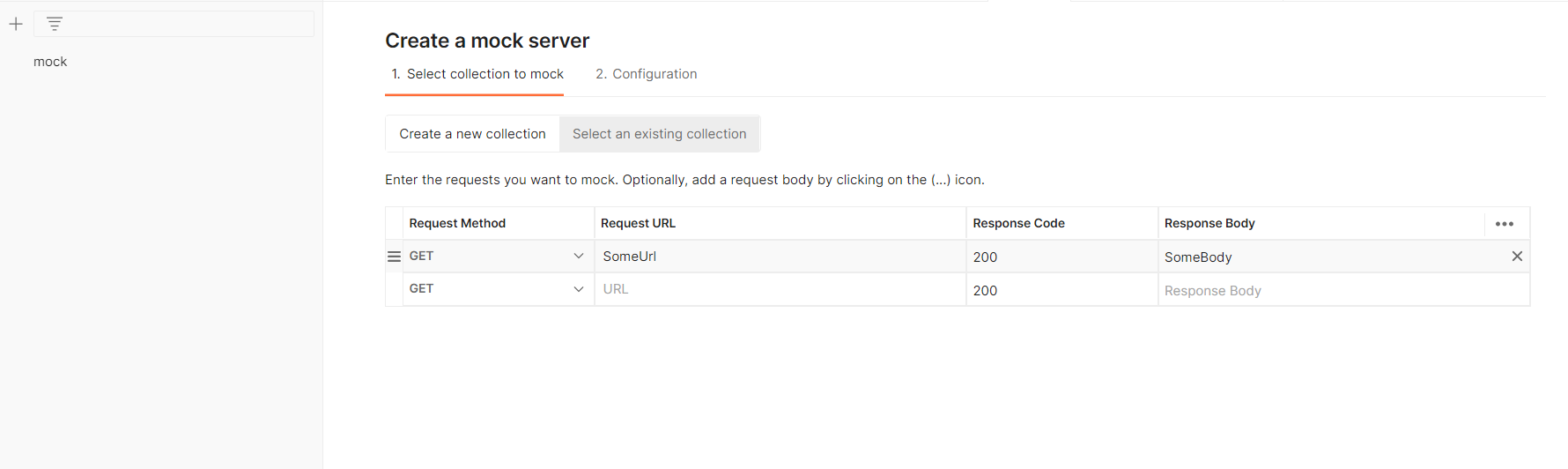
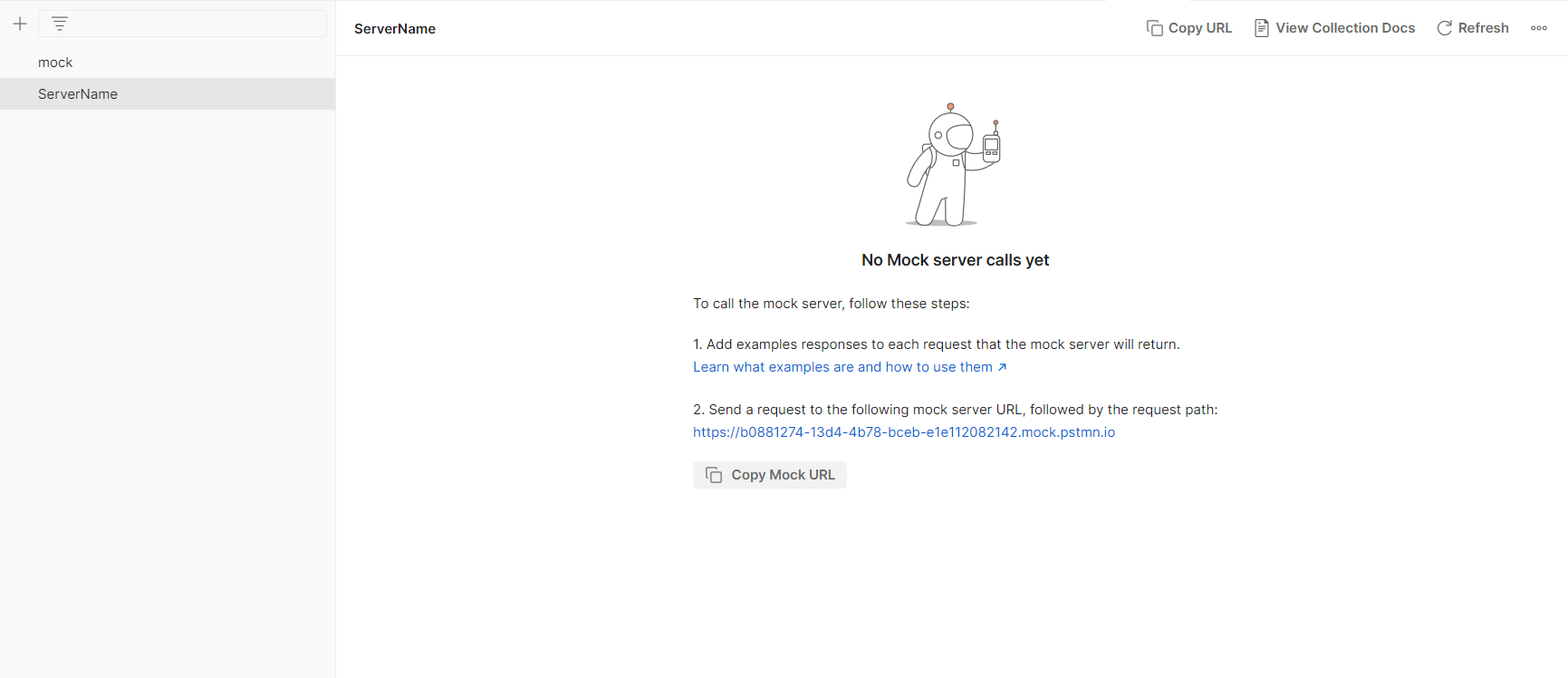
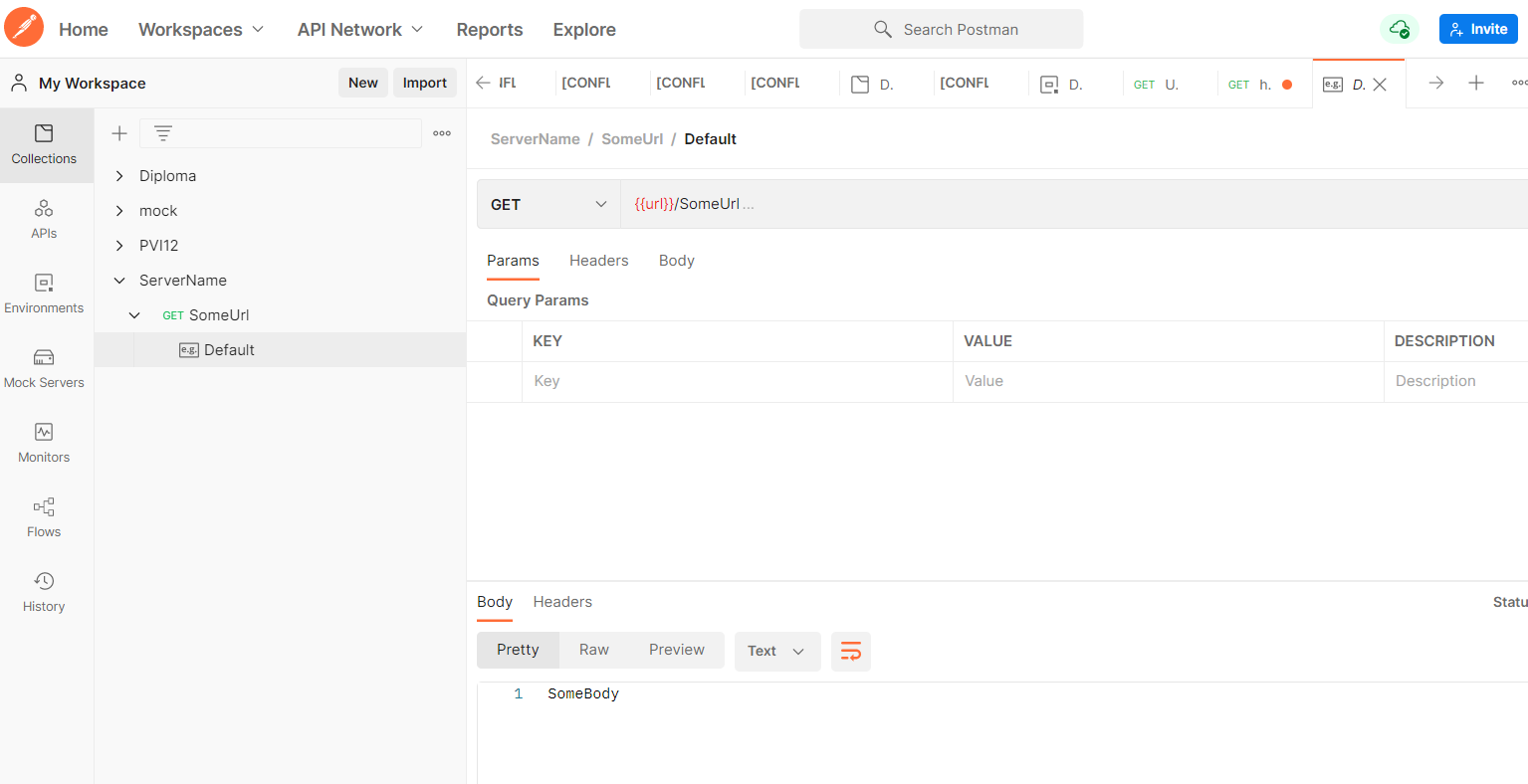
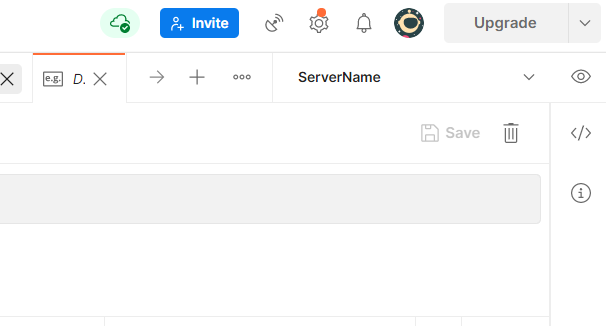
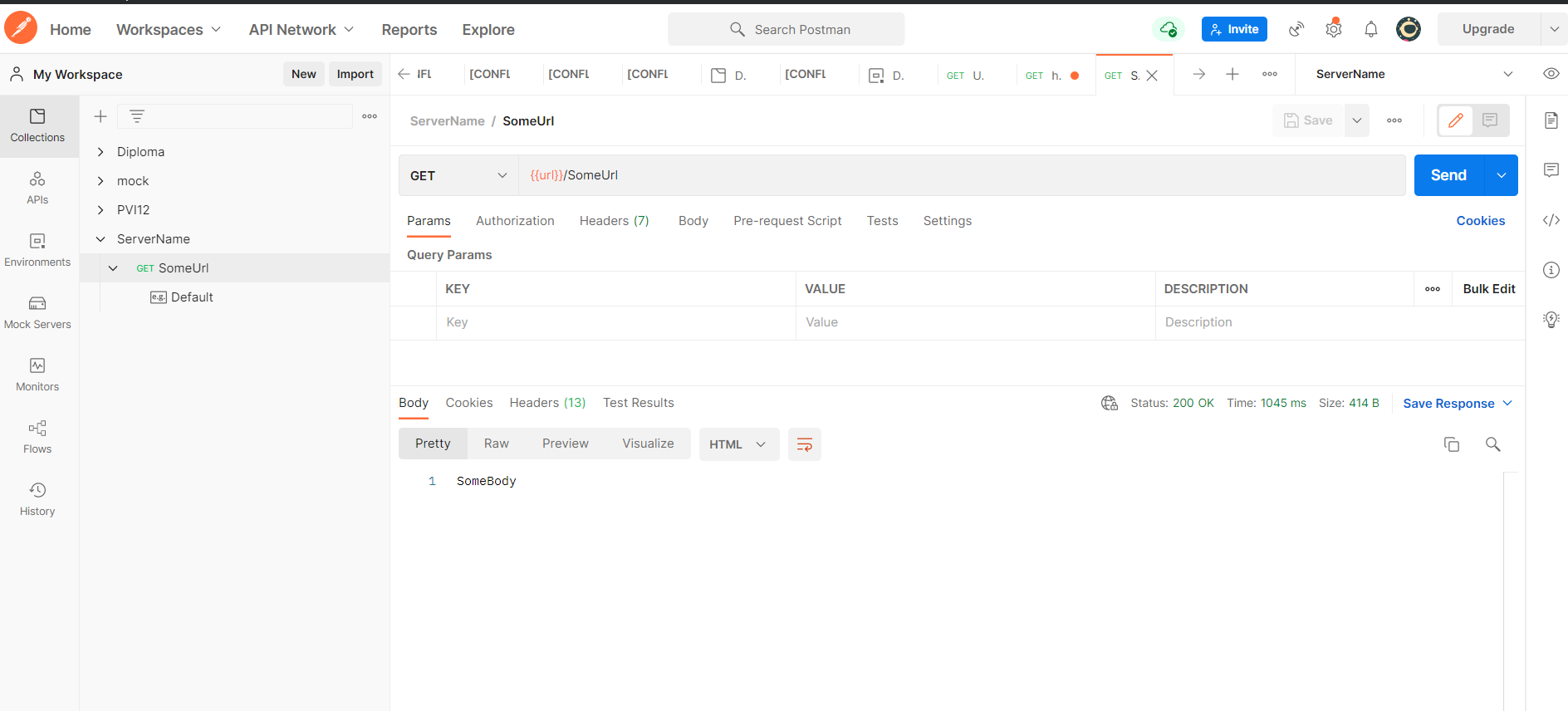


19. Как выполнить делегирование свойств



## 12 Lab

В 12 лабороторной вопросов нет, но я объясню how to mock запросы через postman, чтобы не использовать всякие левые ресурсы

1. Нажимаем на Mock Servers и на плюсик
2. Заполняем как на картинке ниже
3. Создаем (Везде вместо /SomeUrl пишем /things)
4. Здесь история 
5. Переходим обратно в Collctions и находим конфигурацию ответов  
   
6. Выбираем сверху справа нужный Environment  
   
7. Выполняем  
     
   8. Url настроить помогу вручную, Mock должен возвращать вот это:

[

{

"name": "theFirstName",

"description": "someDescription"

},

{

"name": "theSecondName",

"description": "someDescription"

},

{

"name": "theThirdName123123",

"description": "someDescription"

},

{

"name": "theFourthName12312312",

"description": "someDescription"

},

{

"name": "theFifthName22222",

"description": "someDescription"

}

]

## 13 Lab

<https://github.com/SmartToolFactory/Animation-Tutorials>

# CW

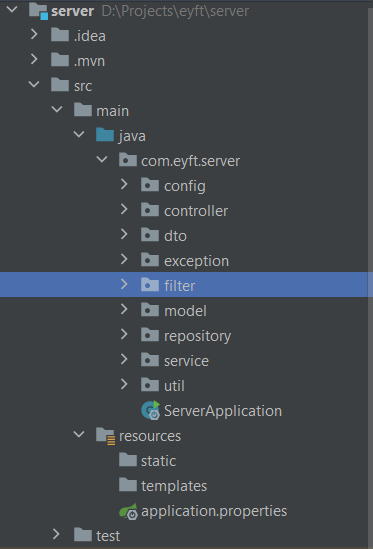
Раздел для тех кто будет делать кп на Java и Spring

Проект состоит из двух модулей

Backend - Rest Application. [Resource](https://spring.io/projects/spring-boot)

Frontend - React application

## Структура Backend



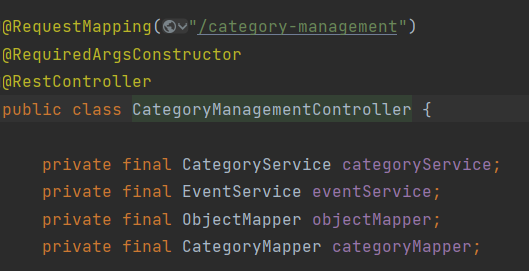
* Config - Здесь хранятся классы конфигурирующие приложение и хранящие в себе [бины](#_c42aqu98ws4w)
  + [CoreConfig](#_zyqk1gsfjlv)
  + [WebConfig](#_2g6pnnacqpr3)
* Controller - Тут контроллеры, они обрабатывает Http запросы. Основные аннотации
  + @GetMapping, @PostMapping (Pattern : [MethodName] + Mapping ) so on and so on - Обрабатывают определенный метод по URL, что указан в аннотации. E.g. @GetMapping(“/users/”). Указывается над методами
  + @RequestMapping(“/someUrl”) - это аннотацию стоит указывать на контроллером, если все методы это контроллера имеют общую приставку. В комбинации с предыдущей аннотацией @GetMapping(“/users/”), получим связку /someUrl/users
* DTO - data transfer object - объекты, служащие только для обмена данными, вы принимаем json от фронта и Spring маппит их в эти DTO.
* Exception - все кастомные excpetions
* Filter - содержит в себе JWT Filter для авторизации. Как работает:
  + на /auth отправляются credentials, если они верны получаем токен.
  + большинство URL защищены этим фильтром (Защиту можно посмотреть в controller.WebSecurityConfig.configure(HttpSecurity http)), этот фильтр проверяет на валидность и права владельца этого токена, если все отлично, то дальше обработка запроса переходит на контроллер  
    ([REST API с использованием Spring Security и JWT](https://habr.com/ru/post/545610/) - Здесь можно подробнее ознакомится с этим, но особо досконального понимания не требует)
* Model - классы-модели, то же самое, что и domain-классы.
  + Mapper - пакет с мапперами model->DTO и обратно
* Repository - репозитории, интерфейсы расширяющие JpaRepository, на основе которых создаются таблицы в БД.
* Service - сервисы, ни слова более (Ivanshka: бизнес-логика)
* util - утилиты, которые вынесены в отдельные классы

Используем Spring Boot в нем есть основные сущности, которые помечаются соответствующими аннотациями:

* Контроллер - @Controller (Если Rest то @RestController | Отвечает за обработку запросов [ Может содержать, а может и нет бизнес логику, но лучше нет ] )
* Конфиг класс - @Configuration (Класс, который определяет в себе бины @Bean)
* Сервис - @Service (То, что содержит в себе бизнес логику)
* Репозиторий - @Repository (То, что дает нам доступ как базам данным. Мы используем JPA, т.е. просто объявляем нужные интерфейсы и классы, которые описывают данные, и как результат, у будут созданы таблицы)
* Компонент - Все что перечислено выше является компонентами, но по-другому называются, для повышения читаемости. Допустим util-класс нельзя поместить в выше упомянутые категории поэтому это просто @Component. Любой класс помеченный этой аннотацией или упомянутыми выше становится [бином](#_c42aqu98ws4w).

##### @Bean

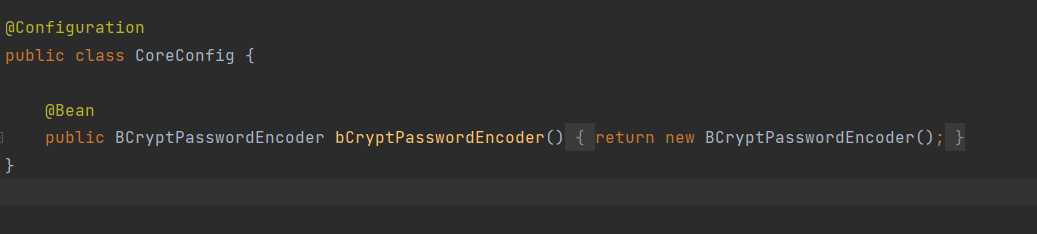
Метод помеченный этой аннотацией, возвращает объект, который будет инжектирован везде, где он требуется. Например тут требуется четыре объекта и здесь есть конструктор определенный lombok @RequiredArgsConstructor, который создаст конструктор для всех полей, имеющих final модификатор



## Классы

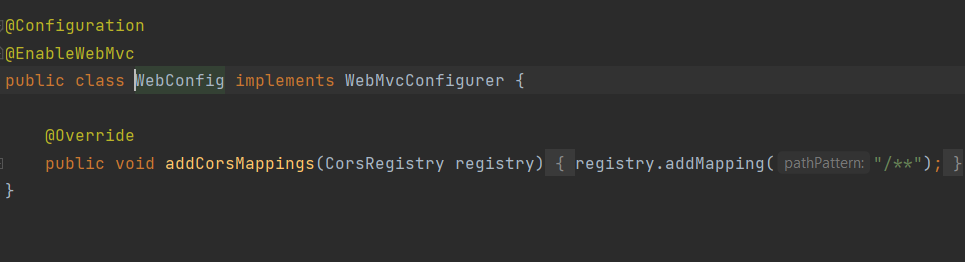
### Конфиг-классы

#### CoreConfig



BCryptPasswordEncoder - нужен для шифрования паролей

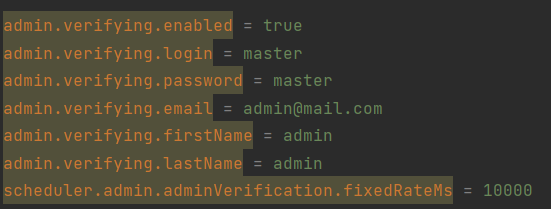
#### WebConfig



addCorsMappings - в этом методе, мы разрешаем [CORS](https://ru.wikipedia.org/wiki/Cross-origin_resource_sharing)

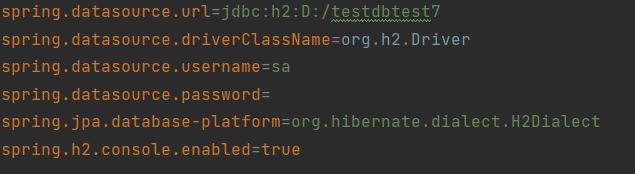
## Administration

В application.properties можно найти

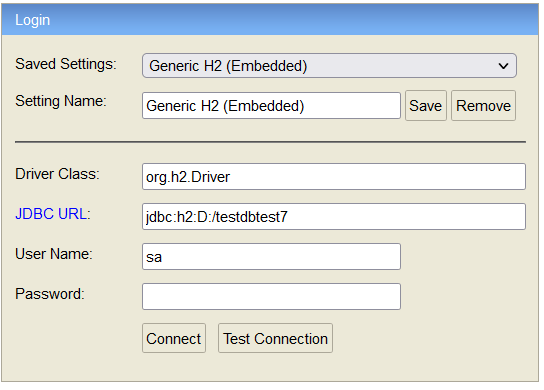


Это конфиги scheduler и он работает если enabled равен true. Он каждые fixedRateMs проверяет если ли пользователь с логином master и есть ли у него права admin. Если такого не находит создает его с указанными данными как password, email so on and so on.

## H2-Console



В браузере по url <http://localhost:8080/h2-console> можно получить доступ к бд.



В JDBC URL - вставляем то что в spring.datasource.url

## Запуск

1. Запускаем back (Idea)
2. Запускаем front (VS code -> terminal -> npm start)
3. Переходим на <http://localhost:3000/auth> и вводим login( master ) password( master ). Эта учетная запись создается автоматически, если вы не меняли то, что написано в [Administration](#_f1yxfolq32da)

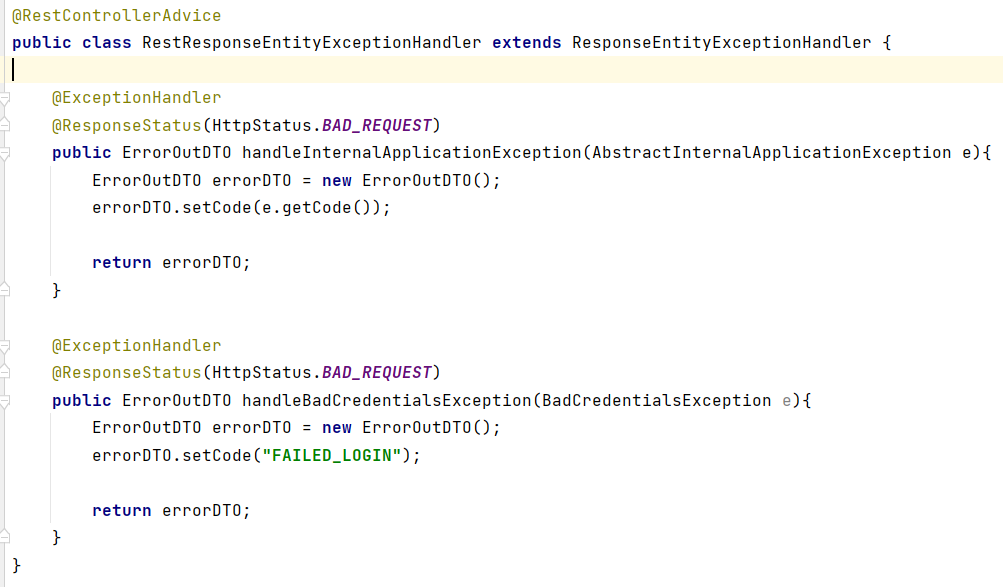
## Backend

(Ivanshka)

Помогло:

1. [Spring framework ПОЛНЫЙ КУРС для начинающих 2021 REST API](https://www.youtube.com/watch?v=q87Xxu4NPIc&ab_channel=alishev)
2. [Курс Java Collections - Лекция: Java Beans](https://javarush.ru/quests/lectures/questcollections.level06.lecture02)
3. Про обработку исключений в контроллерах  
   [Обработка исключений в контроллерах Spring](https://habr.com/ru/post/528116/)
4. Про классы WebConfig & WebSecurityConfig  
   [Настройка CORS в Spring Security](https://sysout.ru/nastrojka-cors-v-spring-security/)
5. Про то, как переделать модели в проекте:
   1. Главное: <https://javarush.ru/groups/posts/hibernate-java>
   2. Доп:  
      <https://www.baeldung.com/hibernate-lazy-eager-loading>  
      <http://java-online.ru/hibernate-entities.xhtml>
6. Про аннотацию @Transactional в контроллерах:  
   [Как Spring @Transactional действительно работает?](https://coderlessons.com/articles/java/kak-spring-transactional-deistvitelno-rabotaet)

## Backend FAQ

1. (Ivanshka)  
   Никита, забрось эту ссылку куда-нибудь здесь в нужное место, лично мне она помогла шо-та понять)  
   [REST API с использованием Spring Security и JWT](https://habr.com/ru/post/545610/)  
   (Nikita)  
   Добавил
2. (Ivanshka)  
   Кто и как обрабатывает исключение, брошенные в контроллерах? Например, здесь:  
     
   Подозреваю, что это делает сам Spring, однако что в этом случае он возвращает на front?  
   Подозреваю, что Spring, используя аннотации, использует вот эти обработчики для исключений, используя тот, аргумент которого соответствует типу исключения.  
     
   Все верно?   
   Ответ тут:  
   [Обработка исключений в контроллерах Spring](https://habr.com/ru/post/528116/)

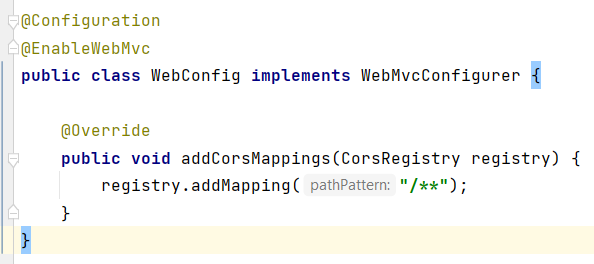
(nikita) Да, @RestControllerAdvice обрабатывает их

1. (Ivanshka)  
   Можно подробнее про @Bean? Ты написал выше, что метод возвращает объект, который инжектируется везде, где он требуется? Но я не понял, как помечается место, где он требуется.

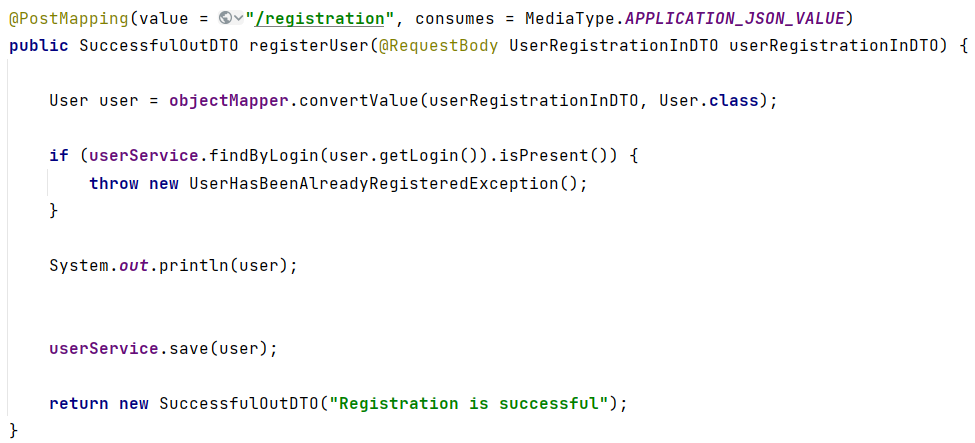
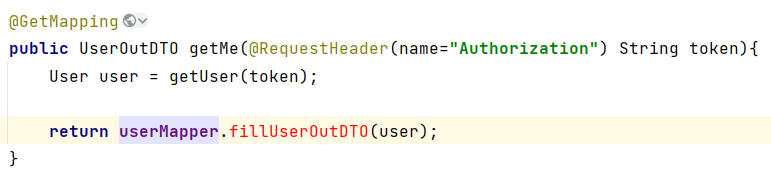
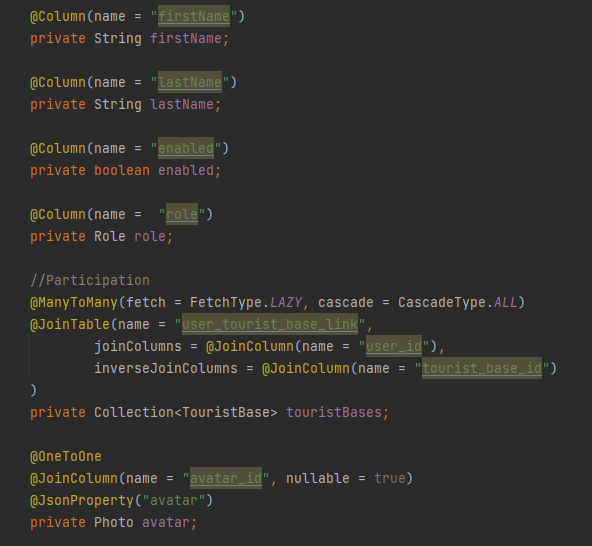
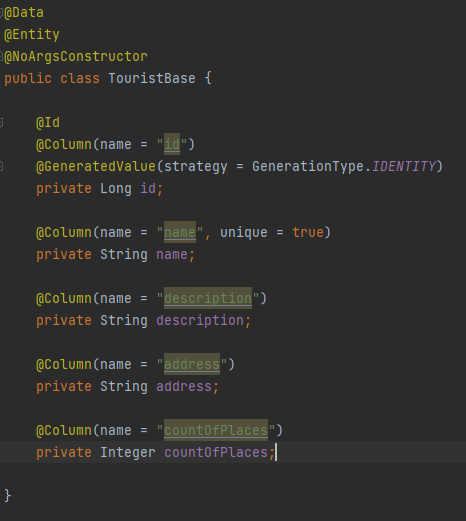
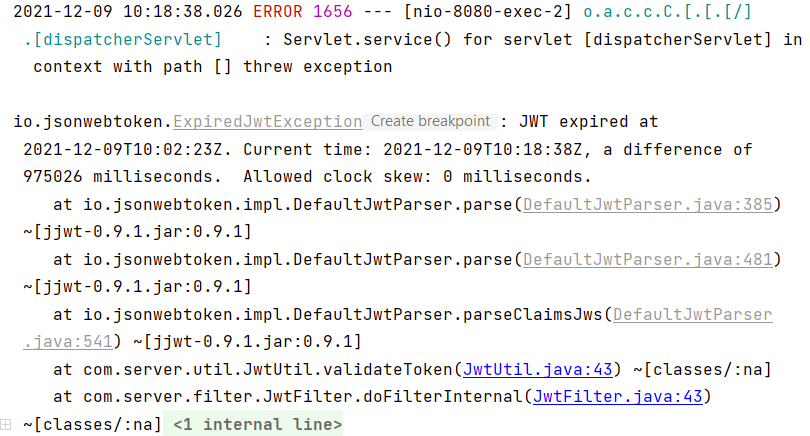
(nikita)

Lombok может работать со спрингом. private final - значит что поле обязательно

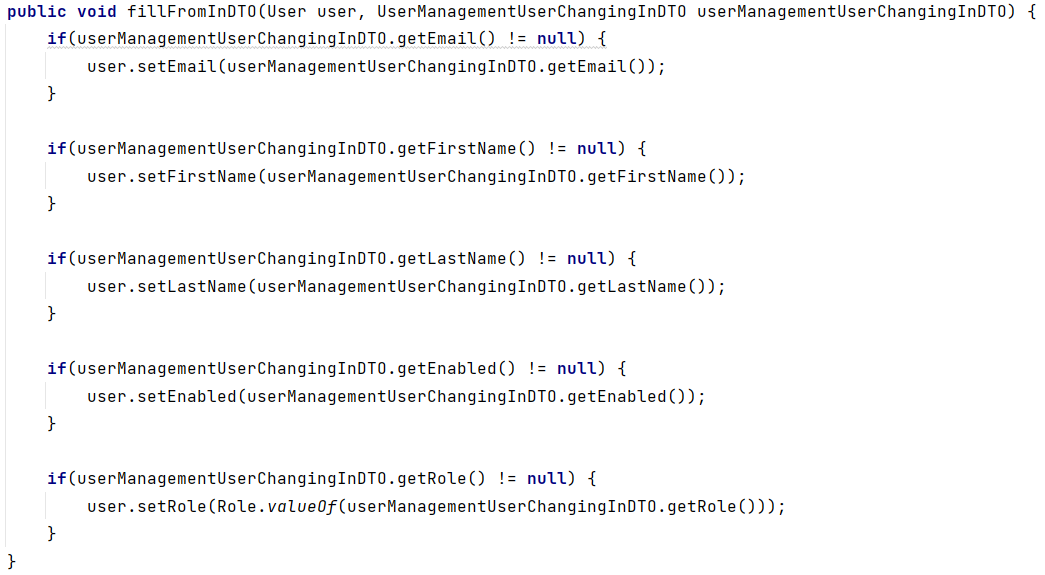
@RequiredArgsConstructor создает конструктор, в который все будет инжектировано.

1. (Ivanshka)  
   Можешь объяснить, что делает этот класс? Мы с Well’ом гуглили, но шо-то не особо разобрались(  
     
   Ответ найден самостоятельно здесь:  
   [Настройка CORS в Spring Security](https://sysout.ru/nastrojka-cors-v-spring-security/)

(nikita) Я отвечал на этот вопрос. Смотри раздел WebConfig

1. Почему в методе для регистрации мы используем просто convertValue:  
     
   А в методе для получения инфы о юзере - fillUserOutDTO?  
     
   (nikita)   
   convertValue - это был мой первый подход, я его изменил
2. Не работает связь многие ко многим.  
   Связь выполняется между ToursitBase и User. Код User  
     
   Код TouristBase  
     
   Но при добавлении новой туристической базы, запись не добавляется в промежуточную таблицу user\_tourist\_base\_link.  
   (Ivanshka)  
   Well шарит за эту хрень достаточно Well, так что можешь его дергать по этому поводу, когда Никита уже будет спать)
3. (Ivanshka)  
   По истечению токена (пока спал, ноут был в гибернации) происходит исключение на сервере:  
     
   Ожидаемо, на front возвращается 500-я ошибка. Как это можно грамотно обработать? Например, чтобы клиента перебрасывало на страницу логина.

(nikita) Как бы это странно не казалось, но это правильно. Я сам раньше обращал на это внимание, но потом вычитал, что эта логика верна, что неверный токен должен обрабатываться как грубейшая ошибка системы

1. (Ivanshka)  
   Я так понимаю, этот код писался до того, как ты начал использовать String.isNullOrEmpty() и все проверки можно заменить на него, верно?  
   

(nikita) Да иван, ты прав. Это легаси

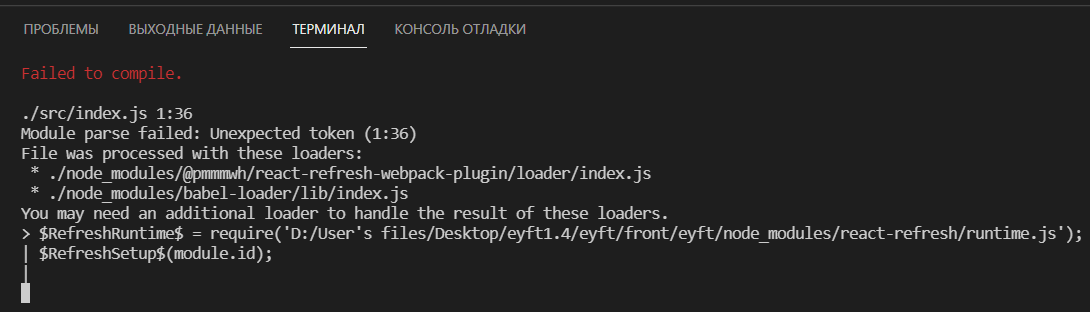
## Frontend

(Ivanshka)

Для тех, кто будет делать front на React.js. Это то, что читал я и мне помогло что-то понять, разобраться в чем-то.

1. В комментах говорят, что статья слабовата и даже вредна для новичков, но я хотя бы понял, с чем имею дело и как это +/- писать.  
   [React.js: понятное руководство для начинающих / Хабр](https://habr.com/ru/company/ruvds/blog/428077/)  
   (Nikita)  
   Я ориентировался на это  
   [Use React and Spring Boot to Build a Simple CRUD App](https://developer.okta.com/blog/2018/07/19/simple-crud-react-and-spring-boot)
2. Цикл статей “Погружение в React”, ⅔ статей. Ссылки на статьи в конце страницы.  
   [Погружение в React: роутер](http://prgssr.ru/development/pogruzhenie-v-react-router.html#heading-section)
3. Официальная документация React  
   [Привет, мир – React](https://ru.reactjs.org/docs/hello-world.html)

## Frontend FAQ

1. Как запустить frontend? Нужно ли для этого ставить node.js?  
   (Nikita)  
   Да нужно, скачайте и установите LTS [node.js](https://nodejs.org/en/), затем через VS Code откройте проект и “Терминал” -> “Создать терминал” -> npm start
2. Как лечить?  
   

(Nikita)   
Сперва попробуй npm install, затем npm start. Там много зависимостей, которые вероятно у тебя еще не стоят.

(Ivanshka)

Для Linux помогло, для Windows - нет. (-\_-)

1. как выключить фронт?) start - запустить, а выключить?)  
   (Ivanshka)  
   По ходу только с помощью твоего лучшего друга, “Ctrl + C”

(nikita) Ctrl + C

1. (Ivanshka)  
   Я так понимаю, у нас такой проблемы нет благодаря тому, что мы юзаем Spring, у которого есть свой собственный роутер. Все верно?  
   