https://github.com/TimurDushanov/SpringBoot\_HomeWork1\_CondithionalApp

**Задача Conditional приложение**

**Описание**

Как часто бывает в промышленной разработка, мы хотим в локальном или dev окружении использовать несколько упрощенные варианты наших сервисов. Поэтому сегодня мы напишем приложение на Spring boot, в котором в зависимости от параметров, будут вызываться разные сервисы.

Для работы необходимо подготовить несколько классов:

1. Создайте spring boot приложение, как было сделано на лекции
2. Нам нужен интерфейс, реализации которого мы будем вызывать в зависимости от параметров нашего приложения:

public interface SystemProfile {

String getProfile();

}

И две реализации:

public class DevProfile implements SystemProfile {

@Override

public String getProfile() {

return "Current profile is dev";

}

}

и

public class ProductionProfile implements SystemProfile {

@Override

public String getProfile() {

return "Current profile is production";

}

}

1. Вам необходимо написать JavaConfig, в котором вы объявите бины классов DevProfile и ProductionProfile примерно так:

@Bean

public SystemProfile devProfile() {

return new DevProfile();

}

@Bean

public SystemProfile prodProfile() {

return new ProductionProfile();

}

1. Дальше вам нужно создать application.properties в папке resources, если его нет, и добавить туда свой пользовательский параметр netology.profile.dev = true. Обратите внимание, что сейчас наш параметр имеет значение true.

Как раз таки в зависимости от значения параметра, мы и будем использовать ту или иную реализацию интерфейса SystemProfile.

1. Теперь, вам нужно использовать одну из аннотаций @Conditional и поместить ее на бины в вашем JavaConfig. Советую использовать @ConditionalOnProperty, и в зависимости от значения netology.profile.dev будет создаваться бин интерфейса SystemProfile той или иной реализации.
2. Чтобы проверить работоспособность нашей логики, создадим контроллер, который будет нам возвращать значения из нашего сервиса SystemProfile:

@RestController

@RequestMapping("/")

public class ProfileController {

private SystemProfile profile;

public ProfileController(SystemProfile profile) {

this.profile = profile;

}

@GetMapping("profile")

public String getProfile() {

return profile.getProfile();

}

}