

Resolver 搭建方式

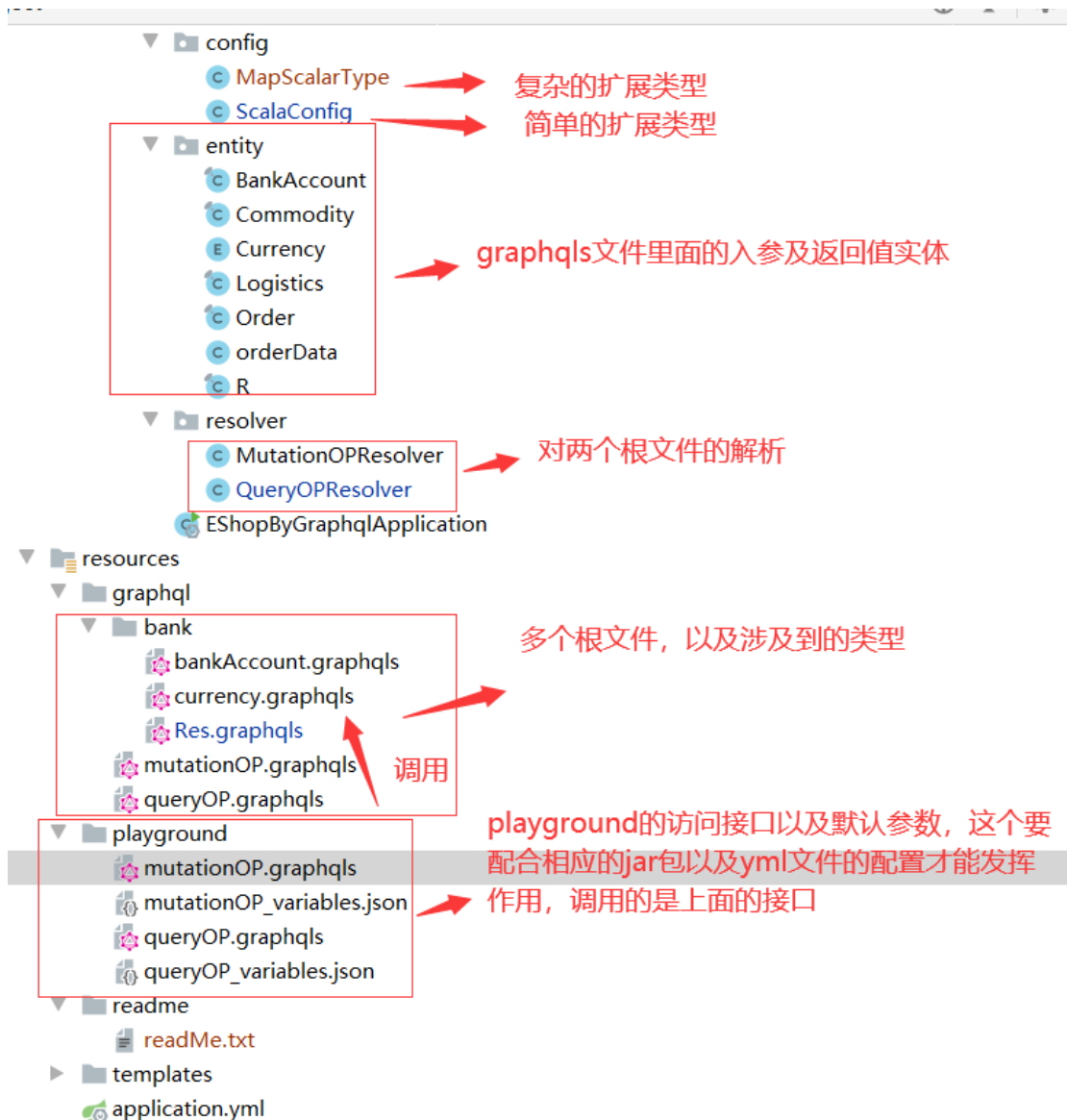
一、前言

1. 这种方式的特点如下:

- 1) 可以使用 `graphql.scalars.ExtendedScalars` 来进行类型的扩展, 类型有限, 但是方便快捷, 也可以定义复杂的类型扩展
- 2) 可以使用 `playground` 来进行访问, 也就是类似 `swagger` 的网页访问接口
- 3) 可以使用多个根 `graphqls`, 且不具有继承关系, 而不是只能有一个根 `graphqls` 文件, 其他的必须继承该根文件才行;
- 4) 不能使用 `provider` 进行整合, 不能使用 `controller` 进行访问
- 5) 总体来说**实用性很差**, 但是是视频里讲解的, 拓展性很强, 可以为其他包含 `provider` 的框架提供技术支持, 比如异常的处理, 批量处理等

二、搭建方式

1. 这是一个很基础的基础框架搭建教程, 所以步骤就不写了, 具体参照 `demo` 来进行开发, 下面对基础框架的结构进行介绍



2. 视频的完整 demo 见文件夹📁视频完整例子，里面有 resolver 方式的各种扩展，具体包括下面的几种;除了 ExtendedScalars，其他的可以扩展到含有 provider 方式。


视频选集 (1/17) 		自动连播 
 P1	01-课程说明和环境需求	16:49
P2	02-创建Schema和Query	22:56
P3	03-maxQueryDepth配置演示	14:03
P4	04-Playground配置演示	12:07
P5	05-Schema可视化Voyager演示	06:08
P6	06-数据解析器Resolvers	13:43
P7	07-定制异常处理Spring ExceptionHandler	11:00
P8	08-定制异常处理GraphQLExceptionHandler	08:08
P9	09-通过DataFetcherResult返回部分数据	05:01
P10	10-异步数据解析(Async Resolvers)	09:52
P11	11-改变数据(Mutations)	11:46
P12	12-文件上传(File Upload)	12:28
P13	13-DataFetchingEnvironment_Selection...	09:48
P14	14-定制标量(Custom Scalars)	13:03
P15	15-定制日期标量(Date Type Scalars)	09:00
P16	16-输入校验(Input Validation)	05:29
P17	17-请求监听器(Request Listener)	10:16

3. 对于该结构的访问，见视频中的例子，对于基础框架，一般启动后会有这样一个界面，然后进行访问

mutationOP × queryOP +

PRETTIFY HISTORY http://localhost:8099/graphql

```
1 # 查询样例，playground前端的默认显示
2 # 下面的ID位于json中
3 • mutation MUTATION_OP($num: NonNegativeInt) {
4     #调用query.graphqls里面的方法进行
5     #如果要调用其他的 需要手写在play ground里面
6     # bankAccount(id: $num) {
7     #     name
8     # }
9
10 #然后着手加 增加操作
11
12 createOrder(num: $num) {
13     msg
14 }
15 }
```



```
{
  "data": {
    "createOrder": {
      "msg": "创建成功，订单的数量为：15"
    }
  }
}
```