Provider结合resolver搭建方式

1. 前言
2. 这种方式的特点如下：
3. 既可以使用graphql.scalars.ExtendedScalars来进行类的扩展，也可以使用自定义的类扩展；
4. 不能使用playground插件（至少我没这么使用，而且项目中实用性比较低，如果愿意研究的话可以研究下，或许可以使用）
5. 可以利用provider等结合controller等来进行前后端交互
6. Graphqls必须只有一个根文件，其他的query或者mutation必须继承根文件的类型才可以正常使用
7. 总体来说，**实用性很强，**而且resolver方式相对于fetcher更加简单实用，所以推荐这种方式；
8. 框架搭建方式及技术升级
9. 框架搭建方式见demo，是个很简单的provider\_resolver框架，下面将通过图例的方式来介绍各部分的功能



1. 技术升级：技术升级参照官网及视频教程，官网的话比较全面，但是是全英文文档，视频的话讲解的比较详细，大概包括以下内容：



1. 然后就是被教程给出的例子的访问方式，下面给出了以postman的访问方式
2. 接口一：updateName

{

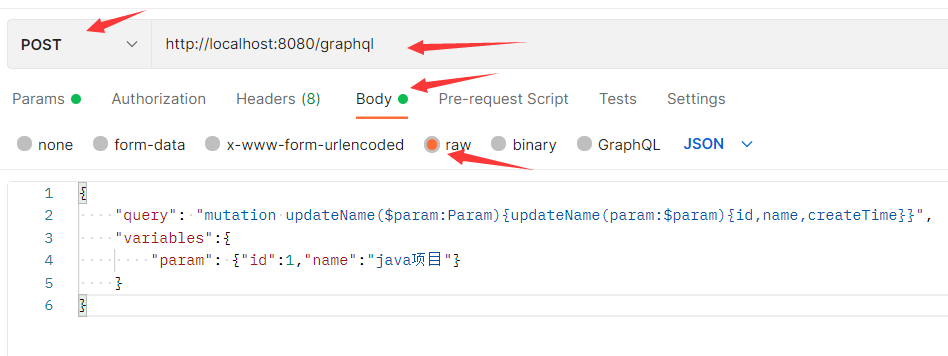
    "query": "mutation updateName($param:Param){updateName(param:$param){id,name,createTime}}",

    "variables":{

        "param": {"id":1,"name":"java项目"}

    }

}



1. 接口二：queryById

{

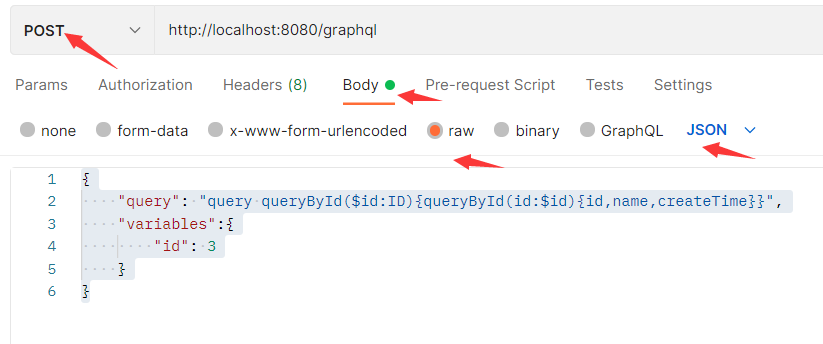
    "query": "query queryById($id:ID){queryById(id:$id){id,name,createTime}}",

    "variables":{

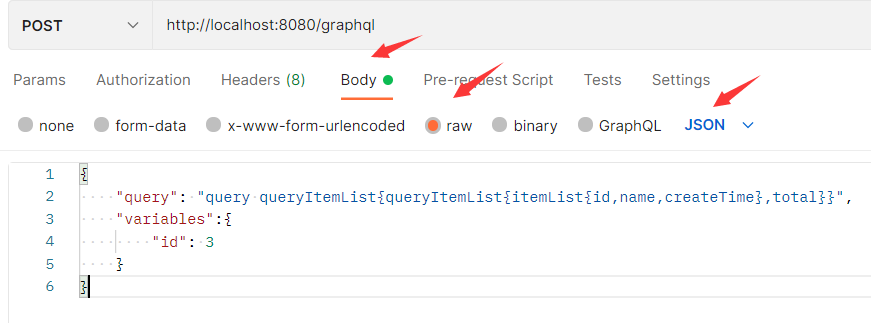
        "id": 3

    }

}



1. 接口三：queryItemList



{

    "query": "query queryItemList{queryItemList{itemList{id,name,createTime},total}}",

    "variables":{

        "id": 3

    }

}