



黑客派

GraphQL 从入门到放弃

作者: [anlingyi](#)

原文链接: <https://hacpai.com/article/1551930670842>

来源网站: 黑客派

许可协议: [署名-相同方式共享 4.0 国际 \(CC BY-SA 4.0\)](#)

<p>原文地址: https://xeblog.cn/articles/6 </p>

<h2 id="简介">简介</h2>

<blockquote>

<p>千言万语汇成一句话: 我的未来式由我做主。</p>

</blockquote>

<p>GraphQL 是一个用于 API 的查询语言, 是一个使用基于类型系统来执行查询的服务端运行时 (类型系统由你的数据定义)。GraphQL 并没有和任何特定数据库或者存储引擎绑定, 而是依靠你现有代码和数据支撑。(完美的一段 Copy) </p>

<h2 id="为什么要用它-">为什么要用它? </h2>

前端自己定义返回的数据及结构, 降低前后端沟通成本

无需编写接口文档 (GraphQL 会根据 schema 自动生成 API 文档)

schema 拼接, 可以组合和连接多个 GraphQL API, 合并为一个, 减少请求次数

<h2 id="运用">运用</h2>

<blockquote>

<p>基于 SpringBoot 的运用</p>

</blockquote>

<h2 id="依赖">依赖</h2>

```
<pre> <code class="highlight-chroma">&lt;dependency&gt;
    &lt;groupId&gt;com.graphql-java&lt;/groupId&gt;
    &lt;artifactId&gt;graphql-spring-boot-starter&lt;/artifactId&gt;
    &lt;version&gt;4.0.0&lt;/version&gt;
&lt;/dependency&gt;
&lt;dependency&gt;
    &lt;groupId&gt;com.graphql-java&lt;/groupId&gt;
    &lt;artifactId&gt;graphql-java-tools&lt;/artifactId&gt;
    &lt;version&gt;4.3.0&lt;/version&gt;
&lt;/dependency&gt;
&lt;!-- 可视化调试工具 --&gt;
&lt;dependency&gt;
    &lt;groupId&gt;com.graphql-java&lt;/groupId&gt;
    &lt;artifactId&gt;graphiql-spring-boot-starter&lt;/artifactId&gt;
    &lt;version&gt;4.0.0&lt;/version&gt;
&lt;/dependency&gt;
</code> </pre>
```

<h2 id="定义">定义</h2>

<h3 id="服务入口定义">服务入口定义</h3>

<blockquote>

<p>resources 目录下创建一个名为 root.graphqls 的文件</p>

</blockquote>

```
<pre> <code class="highlight-chroma"># 查询相关接口
type Query {
  # 通过id获取用户信息
  getUserById(id: Int!): User
  # 用户信息列表
  listUser: [User]
}
```

变更相关接口

```
type Mutation {
```

```

# 添加用户信息
saveUser(user: addUserInput!): Boolean
# 删除用户信息
deleteUser(id: Int!): Boolean
# 更新用户信息
updateUser(user: updateUserInput!): Boolean
}

```

</code> </pre>

<p>Query 定义的是查询相关的入口</p>

<p>Mutation 定义的是增删改相关的入口</p>

<h3 id="剖析">剖析</h3>

<pre> <code class="highlight-chroma">geUserById: 与java类中的方法名一致

id: 方法参数, Int(I大写) 表示是一个int类型, "!"表示非空

User: 返回值类型, 这里返回的是一个java对象

</code> </pre>

<h3 id="scheme定义">scheme 定义</h3>

<blockquote>

<p>resources 目录下创建一个{beanName}.graphqls 的文件
 这个文件是 query/mutation 体的 scheme 定义字段、类型</p>

</blockquote>

<p>示例: user.graphqls</p>

<pre> <code class="highlight-chroma"># 用户信息对象
type User {

编号

id: Int!

用户名

username: String!

密码

password: String!

年龄

age: Int

}

添加用户信息输入参数

```

input addUserInput {
  id: Int!
  username: String!
  password: String!
  age: Int
}

```

更新用户信息输入参数

```
input updateUserInput {  
  id: Int!  
  username: String!  
  password: String!  
  age: Int  
}
```

</code> </pre>

<h2 id="实现入口">实现入口</h2>

<h3 id="实体类">实体类</h3>

<p> </p>

<h3 id="Query相关Java类实现">Query 相关 Java 类实现</h3>

实现 GraphQLQueryResolver 接口

添加 @Component 注解

方法名、参数类型、返回值与 root.graphqls 内的定义一致

<p> </p>

<h3 id="Mutation相关Java类实现">Mutation 相关 Java 类实现</h3>

实现 GraphQLMutationResolver 接口

添加 @Component 注解

方法名、参数类型、返回值与 root.graphqls 内的定义一致

<p> </p>

<h2 id="接口调试">接口调试</h2>

<p>GraphQL 默认请求路径: /graphql</p>

<p>可视化调试界面: /graphiql</p>

<h3 id="使用GraphiQL调试">使用 GraphiQL 调试</h3>

<p>访问地址: http://localhost:8080/graphiql <p>

<p>界面是长这个样子的
 </p>

<h4 id="Query">Query</h4>

<h5 id="查询单条数据">查询单条数据</h5>

<p> </p>

<h5 id="查询所有数据">查询所有数据</h5>

<p> </p>
<h5 id="组合查询">组合查询</h5>
<p>相同名称的接口组合为一个接口调用，需要取一个别名用于区分它们</p>
<p>定义别名 类似这样的格式</p>
<pre><code class="highlight-chroma">别名: 接口
</code></pre>
<p></p>
<h4 id="Mutation">Mutation</h4>
<h5 id="添加">添加</h5>
<p></p>
<h5 id="修改">修改</h5>
<p></p>
<h5 id="删除">删除</h5>
<p></p>
<h5 id="Console">Console</h5>
<p></p>
<h3 id="使用Postman调试">使用 Postman 调试</h3>
<h4 id="Query-">Query</h4>
<h5 id="查询单条数据-">查询单条数据</h5>
<p></p>
<h5 id="查询所有数据-">查询所有数据</h5>
<p></p>
<h5 id="组合查询-">组合查询</h5>
<p></p>
<h4 id="Mutation-">Mutation</h4>
<h5 id="添加-">添加</h5>
<p></p>
<h5 id="修改-">修改</h5>
<p></p>
<h5 id="删除-">删除</h5>
<p></p>
<h2 id="参考">参考</h2>

<p>http://graphql.cn/learn/</p>