API接口

# 一、简介

## 1.1什么是接口

|  |
| --- |
|  |

## 1.2接口的作用

|  |
| --- |
|  |

## 1.3接口的开发与调用

|  |
| --- |
|  |

## 1.4接口的组成

|  |
| --- |
|  |

### 基本说明

|  |
| --- |
|  |

### 返回JSON实例

|  |
| --- |
|  |

# 二、RESTful API

|  |
| --- |
|  |

# 三、json-server工具，他是restful接口工具

|  |
| --- |
|  |

## 小案例：安装json-server后，在我们的代码根目录下面新建一个restful-api文件夹，在里面新建一个01\_json-server文件夹，然后在里面新建一个db.json,内容就是上面的歌曲

|  |
| --- |
|  |

## db.json

|  |
| --- |
| {      "song":[          {"id":1,"name":"干杯","singer":"五月天"},          {"id":2,"name":"当","singer":"动力火车"},          {"id":3,"name":"不能说的密码","singer":"周杰伦"}      ]  } |

## 打开终端并且定位到db.json所在的路径，然后运行：json-server --watch db.json,出现下面的界面说明启动成功

|  |
| --- |
|  |

## 然后我们可以打开浏览器，在地址栏里面输入：<http://localhost:3000/song> ,就可以看到所有的歌曲

|  |
| --- |
|  |

## 如果需要获取某一条歌曲，只需要在song后面拼接/id号就可以了如，我们请求最后一首：<http://localhost:3000/song/3> ，效果如下

|  |
| --- |
|  |

# 四、api测试接口：apipost

|  |
| --- |
|  |

## 1.下载apipost：[Apipost-API 文档、设计、调试、自动化测试一体化协作平台](https://www.apipost.cn/)

## 2.安装，安装提示一步步安装即可

## 3.使用

### a.安装完成后启动apipost，初始化界面如下

|  |
| --- |
|  |

### b.点击右边的“新建http接口”按钮，出现下面的界面

|  |
| --- |
|  |

### c.我们用json-server来模拟后端，所以需要把它启动起来

|  |
| --- |
|  |

### d、我们在新建接口一栏填写说明：批量获取歌曲，然后在地址栏里面填写json-server的接口地址，请求方法选择GET，然后点击发送，踩坑了，apipost说明接口不能访问

### e.我们需要用这个命令启动json-server: json-server --watch --host 127.0.0.1 db.json

### f.还有一个坑，新版本的apipost会自动在接口前面+http://所以我们的接口不能以http://开头，否则报错，书写正确后再来发送请求，效果如下

|  |
| --- |
|  |

### g.我们可以点击右边的保存按钮把接口保存起来，然后在左边的侧边栏会有记录

|  |
| --- |
|  |

#### 以后你想测试这个接口，只需要点击这个记录就可以打开测试界面

### 可以发送post请求，注意需要选择urlencoded格式

|  |
| --- |
|  |

#### 不过这里有一个小问题，就是id我们输入的是number可是保存后是字符串，因为json-server其实不需要设置id，会自动处理id自增，所以我们并不需要输入id

## 4.高级配置

### 我们每一次测试接口都需要输入一串很长的内容，不太方便，我们其实可以配置默认前缀的

#### a.点击右上角默认环境前面的小眼睛，会弹出一个界面

|  |
| --- |
|  |

#### b.在“前置url一栏输入：<http://127.0.0.1:3000>”,然后点击保存

|  |
| --- |
|  |

#### 然后地址栏会自动有上面的地址，我们只需要输入路径如/song,选择请求方式点击发送就可以了

|  |
| --- |
|  |

#### 注意 ：如果你想复杂这个链接，需要把空格删除

### 可以设置查询字符串

|  |
| --- |
|  |

### 也可以发送json格式的请求

|  |
| --- |
|  |

## 注意：json-server会自动处理id，我们不需要输入id

## 5.发送delete请求

|  |
| --- |
|  |

## 6.发送局部更新PATCH请求

|  |
| --- |
|  |

# 五、apipost的公共参数

## 我们上面已经有5个接口。进入我们规定他们每一个接口在发送请求的时候必须携带一个auth参数，那么我们改如何来设置这个auth公共参数？

## 1.点击左边侧边栏的新建按钮，选择新建目录，会弹出如下界面：

|  |
| --- |
|  |

## 2.在目录一栏填写：歌曲，然后点击保存目录

|  |
| --- |
|  |

### 可以看到，左边栏生成了一个歌曲目录

## 3.把我们前面新建的接口全部放进这个目录

|  |
| --- |
|  |

## 4.点击歌曲文件夹，出现如下界面

|  |
| --- |
|  |

## 5点击目录公用header，在参数名一栏填写auth，参数值一栏填写信息如supervisor

|  |
| --- |
|  |

## 6.点击保存目录，把刚刚设置的参数保存起来

## 7.然后我们点击任意一个接口发送请求，发现请求头里面多了一个auth键值就是supervisor

|  |
| --- |
|  |
|  |
|  |

# 六、apipost的文档功能

## 以前我们设计一个接口后，接口文档必须自己一点一点的写，现在不需要，

## 如我们给查询一首歌的接口添加一个search参数，值是hot，还有page参数值是1.我们点击query选项卡，填写好相关内容

|  |
| --- |
|  |

## 点击保存，然后点击分享按钮，会弹出如下界面

|  |
| --- |
|  |

## 点击复制内网链接按钮，此时就把这个链接复制了

## 在浏览器中打开一个新选项卡，在地址栏中粘贴链接，就会打开一个文档

|  |
| --- |
|  |

## 点击歌曲文件夹，可以看到我们设置的公共参数

|  |
| --- |
|  |

## 点击GET获取一首歌接口，会有下面的内容

|  |
| --- |
|  |

## 不过这里有一个小问题，没有响应实例，我们再做一个文档，这一次我们用所有歌曲接口

|  |
| --- |
|  |

## 填写完参数后，我们先发送请求，请求成功后我们把响应数据复制一份，然后点击响应实例选项卡，点击+新建一个响应实例，把响应数据粘贴到最大那个文本框，状态码填写200，名称填写：歌曲列表，点击确定

|  |
| --- |
|  |

## 点击确定后点击保存

|  |
| --- |
|  |

## 然后我们点击分享，弹出下面的窗口，点击复制内外链接

|  |
| --- |
|  |

## 然后打开浏览器窗口，在地址栏中粘贴链接回车，发现此时的文档就有响应实例了

|  |
| --- |
|  |

## 如果我们所有的接口都做完了，接口文档也做了，就可以点击菜单栏上面的分享项目按钮，出现下面的界面

|  |
| --- |
|  |

## 点击复制内网链接按钮，此时就把这个链接复制了

## 在浏览器中打开一个新选项卡，在地址栏中粘贴链接，就会打开一个项目文档

|  |
| --- |
|  |
|  |
|  |
|  |
|  |

## 还可以根据需要把这个文档导出，只需要点击导出文档按钮就会下载到本地

|  |
| --- |
|  |

# 七、接口测试工具postman

## 下载：[Download Postman | Get Started for Free](https://www.postman.com/downloads/)

## 安装，一路next安装完毕后提示你注册账号可以跳过不注册，安装完成后点击快捷方式，界面如下

|  |
| --- |
|  |

## 使用：直接在地址栏里面输入接口链接，选择请求发送，然后点击send就可以了，如我们访问上面的接口

|  |
| --- |
|  |

### postman需要你创建一个账号才给你保存，我们来注册一个账号，然后用这个账号登录

|  |
| --- |
|  |

## 登录后点击save按钮，会弹出保存界面，此时可以新建一个collection，然后保存到这里，保存后的界面如下

|  |
| --- |
|  |

## 可以给collection改名，如我们改为歌曲

|  |
| --- |
|  |

## 然后我们新建一个post请求来添加歌曲

|  |
| --- |
|  |

## 点击save，输入名称点击save

|  |
| --- |
|  |

## postman的很多功能和apipost是类似的。可以对照学习