

axios 异步请求文档

axios 是什么？

基于 promise¹用户浏览器和 node.js 的 Http 客户端。

特点

- 支持 promise
- 能拦截请求和响应
- 能转换请求数据和响应数据
- 能取消请求
- 自动转换 JSON 数据
- 浏览器端支持防止 CSRF（跨站请求伪造）

安装

npm 安装

```
$ npm install axios
```

bower 安装

```
$ bower install axios
```

引入 CDN 的方式

```
<script src="https://unpkg.com/axios/dist/axios.min.js"></script>
```

一个简单的例子

服务器端：

简单的 springboot 框架整合了 mybatis。

数据库：

¹ promise 主要用于异步计算，可以将异步操作队列化，按照期望的顺序执行，返回符合预期的结果。

栏位	索引	外键	触发器	选项	注释	SQL 预览
名	类型	长度	小数点	不是 null		
▶ id	bigint	20	0	<input checked="" type="checkbox"/>	1	
username	varchar	20	0	<input type="checkbox"/>		
age	int	10	0	<input type="checkbox"/>		
sex	varchar	10	0	<input type="checkbox"/>		
password	varchar	20	0	<input type="checkbox"/>		

id	username	age	sex	password
▶ 1	zhanghaokun	22	男	zhkun1999
2	zhangxiaotian	22	男	zxt666

实体类:

```
public class User implements Serializable {
    private static final long serialVersionUID = -94041081303457379L;

    private Integer id;

    private String username;

    private Integer age;

    private String sex;

    private String password;

    public Integer getId() {
        return id;
    }

    public void setId(Integer id) {
        this.id = id;
    }

    public String getUsername() {
        return username;
    }

    public void setUsername(String username) {
        this.username = username;
    }
}
```

```

public Integer getAge() {
    return age;
}

public void setAge(Integer age) {
    this.age = age;
}

public String getSex() {
    return sex;
}

public void setSex(String sex) {
    this.sex = sex;
}

public String getPassword() {
    return password;
}

public void setPassword(String password) {
    this.password = password;
}
}

```

Controller 层接口

```

@RequestMapping("/queryAll")
public JSONObject<User> queryAll(@RequestParam(defaultValue = "1") Integer page,
                                @RequestParam(defaultValue = "10") Integer pageSize,
                                User user){

    JSONObject<User> userJsonObject = new JSONObject<>();
    PageInfo<User> pageInfo = userService.queryAll(page,pageSize,user);

    userJsonObject.setCode(0);
    userJsonObject.setMsg("ok");
    userJsonObject.setData(pageInfo.getList());
    userJsonObject.setCount(pageInfo.getTotal());
    return userJsonObject;
}

```

前端:

通过前端向后台服务器发送 axios 请求

```
<script src="https://unpkg.com/axios/dist/axios.min.js"></script>
<script>
    axios.post("http://127.0.0.1:8080/user/queryAll").then(function (res) {
        console.log(res.data);
    }).catch(function (error) {
    })
}</script>
```

刷新页面，右键控制台，发现在控制台，打印出从后台返回的数据。



传参数案例

传递一个参数（用户的 id）

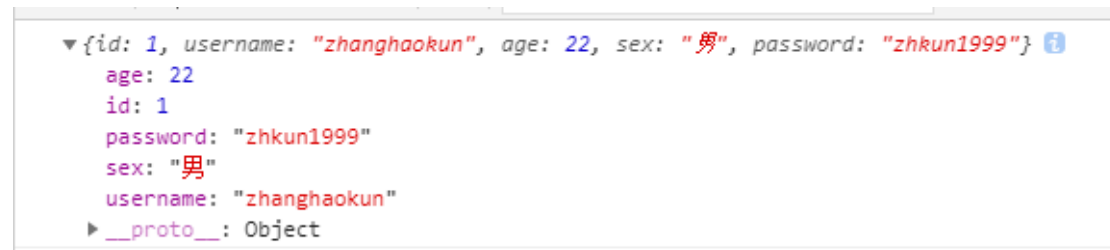
服务器端 Controller 方法接口:

```
@RequestMapping("/getById")
public User getById(Integer id){
    System.out.println("1111");
    User user = userService.queryById(id);
    return user;
}
```

前端 js 代码:

```
<script src="https://unpkg.com/axios/dist/axios.min.js"></script>
<script>
    axios.post("http://127.0.0.1:8080/user/getById",
        {
            params: {
                id: 1
            }
        })
    .then(function (res) {
        console.log(res.data);
    })
    .catch(function (error) {
```

```
    })
  }
</script>
最后请求结果
```



请注意:

当传递一个参数时，axios 的请求应该写成下面两种形式。

```
axios.post("http://127.0.0.1:8080/user/getById",
  {
    params: {
      id: 1
    }
  })
  .then(function (res) {
    console.log(res.data);
  })
  .catch(function (error) {
  })

<script src="https://unpkg.com/axios/dist/axios.min.js"></script>
<script>
  axios.post("http://127.0.0.1:8080/user/getById", {id: 1}).then(function (res) {
    console.log(res.data);
  }).catch(function (error) {
  })
</script>
</body>
</html>
```

传递多个参数（多条件查询）

后端接口与 queryAll 基本一致

@RequestMapping("/queryAll")

```
public JSONObject<User> queryAll(@RequestParam(defaultValue = "1") Integer page,
                                @RequestParam(defaultValue = "10") Integer pageSize,
                                @RequestBody User user){
```

```
    JSONObject<User> userJsonObject = new JSONObject<>();
```

```
    PageInfo<User> pageInfo = userService.queryAll(page, pageSize, user);
```

```
    userJsonObject.setCode(0);
```

```
    userJsonObject.setMsg("ok");
```

```

        userJsonObject.setData(pageInfo.getList());
        userJsonObject.setCount(pageInfo.getTotal());
        return userJsonObject;
    }

```

```

@RequestMapping("/queryAll")
public JsonObject<User> queryAll(@RequestParam(defaultValue = "1") Integer page,
                                @RequestParam(defaultValue = "10") Integer pageSize,
                                @RequestBody User user){

    JsonObject<User> userJsonObject = new JsonObject<>();
    PageInfo<User> pageInfo = userService.queryAll(page,pageSize,user);

    userJsonObject.setCode(0);
    userJsonObject.setMsg("ok");
    userJsonObject.setData(pageInfo.getList());
    userJsonObject.setCount(pageInfo.getTotal());
    return userJsonObject;
}

```

关键看前端传参部分

```

<script src="https://unpkg.com/axios/dist/axios.min.js"></script>
<script>
    var param = {}
    param.username = "zhanghaokun";
    param.sex = "男";
    axios.post("http://127.0.0.1:8080/user/queryAll",param).then(function (res)
    {
        console.log(res.data);
    }).catch(function (error) {
    })
</script>

```

返回结果

```

▼ {code: 0, msg: "ok", count: 1, data: Array(1)} ⓘ
  code: 0
  count: 1
  data: Array(1)
    ▼ 0:
      age: 22
      id: 1
      password: "zhkun1999"
      sex: "男"
      username: "zhanghaokun"
      __proto__: Object
    length: 1
    __proto__: Array(0)
  msg: "ok"
  __proto__: Object

```