[一.ajax 1](#_Toc11310786)

[1.get带参数发送请求（http协议的get方法） 1](#_Toc11310787)

[2.按照restful Api风格，get带参数发送请求 1](#_Toc11310788)

[3.nodejs接收请求数据的方式 2](#_Toc11310789)

[4.查询用户列表，把数据放在div中 2](#_Toc11310790)

[二.JSON数据格式 2](#_Toc11310791)

[1.js对象的格式 3](#_Toc11310792)

[2.JSON字符串的数据格式 3](#_Toc11310793)

[3.json解析 3](#_Toc11310794)

[4.restful风格，是针对json的 3](#_Toc11310795)

[三.XML数据格式 4](#_Toc11310796)

[项目 5](#_Toc11310797)

[一.restful API风格 5](#_Toc11310798)

day06

# 一.ajax

## 1.get带参数发送请求（http协议的get方法）

|  |
| --- |
| http://127...:8080/login?uname=123&upwd=123 |

## 2.按照restful Api风格，get带参数发送请求

|  |
| --- |
| <http://127...:8080/api/>版本号(v1)/login/:uname&:upwd  两个:参数中间的（&）符号可以替换成（-） |

## 3.nodejs接收请求数据的方式

①req.query.uname(get)

|  |
| --- |
| 只能接收 ?uname=dangdang |

②req.body.uname(post)

|  |
| --- |
| 接收请求主体的参数 |

③req.params.uname

|  |
| --- |
| 接收 :参数 /v3/login/:uname-:upwd |

④restful Api 接口风格

|  |
| --- |
| <http://1271:8080/api/v1/>业务名称/dangdang-123456 |

2.post带参数（使用post写登录，restful post用在注册）

|  |
| --- |
| 1.post/put的请求数据要放在请求主体中  var formdata="uname="+uname.value+"&upwd="+upwd.value;  xhr.send(formdata);  2.请求主体的数据携带特殊符号  需要设置请求头信息，设置的位置xhr.open~~~xhr.send之间  xhr.setRequestHeader(“Content-Type”,”application/x-www-form-urlencoded”); |

## 4.查询用户列表，把数据放在div中

# 二.JSON数据格式

|  |
| --- |
| 使用ajax访问服务器，服务器操作数据库，把结果通过响应传送给浏览器  在前台ajax代码中，xhr.responseText得到这个数据  这个数据的数据类型是string  这个数据叫做json数据 |

## 1.js对象的格式

|  |
| --- |
| var obj={uid:1,uname:”dingding”,upwd:”123456”}  var 对象名称= {属性1：值1，属性2：值2……}  js对象，属性名称不加双引号 |

## 2.JSON字符串的数据格式

|  |
| --- |
| JavaScript Object Notation  js 对象 表现方式  js对象表示法，以js对象的数据格式表现出来的字符串  js字符串的格式  1.json中一对{}表示一个对象  2.json中属性名，必须使用””包裹起来(单引号不报错，但是不推荐)  3.属性值，如果是字符串，也要带双引号  4.json本事是一个字符串，所以在最外层加单引号 |

## 3.json解析

|  |
| --- |
| var arr=JSON.parse(result); |

## 4.restful风格，是针对json的

# 三.XML数据格式

|  |
| --- |
| eXtensible Markup Language  可拓展 标记 语言  xml是html的一个变种，专门负责承载数据  所以操作xml也需要dom  xml的标签，都是没有预定义，需要自定义  xml只用于做数据的传递，而不做数据展示 |

1.xml语法

①xml文件顶端做声明

|  |
| --- |
| <?xml version="1.0" encoding="utf-8"?> |

②xml标签只有双标记

|  |
| --- |
| <name></name> |

③一个xml文件，有且只有一个根元素

④标记严格区分大小写

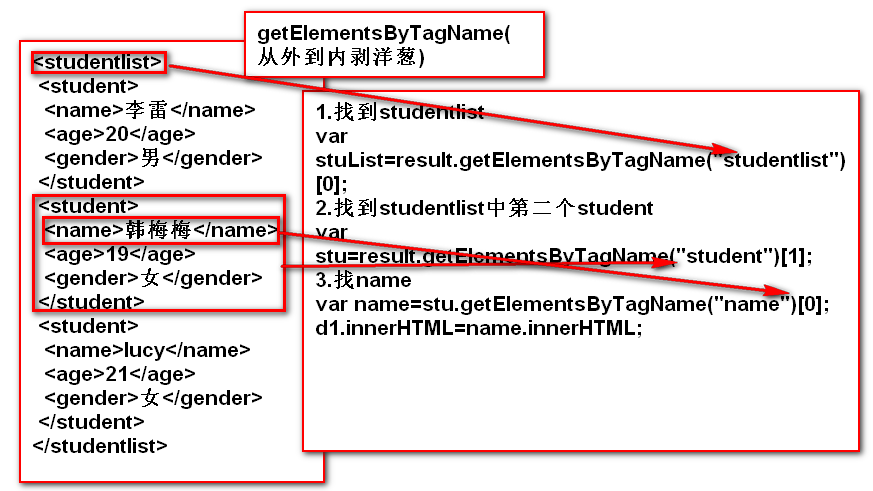
|  |
| --- |
| <name></name> <Name></Name> |

⑤可以嵌套，但是注意嵌套顺序

|  |
| --- |
| <student>  <name></name>  <age></age>  </student> |

⑥每个表及允许定义属性，但是属性的值，必须使用双引号

|  |
| --- |
| <student num=”10”></student> |



# 项目

# 一.restful API风格

|  |
| --- |
| 风格：不是强制性的，是前后端对接口格式一个统一协商  前后端开发的分离 |
| restful API接口文档 |

1.mypro.js

2.静态文件夹mypro

3.任务

①登录

|  |
| --- |
| 在mypro.js中，定义一个接口  关键代码，后台  router.grt(“/v1/login/:uname-:upwd”,(req,res)=>{  var $uname=req.params.uname;  var $upwd=req.params.upwd;  });  前台  xhr.open(“get”,”/mypro/v1/login”+uname.value+”-”+upwd.value,true); |

作业：

1.获取userlist，在table中显示

2.完善接口文档

3.登录，前台做非空验证

4.post传参