

## 文件下载

```
package com.southwind.servlet;

import javax.servlet.ServletException;
import javax.servlet.annotation.WebServlet;
import javax.servlet.http.HttpServlet;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;
import java.io.FileInputStream;
import java.io.IOException;
import java.io.InputStream;
import java.io.OutputStream;

@WebServlet("/download")
public class DownloadServlet extends HttpServlet {

    @Override
    protected void doGet(HttpServletRequest req, HttpServletResponse resp)
        throws ServletException, IOException {
        String type = req.getParameter("type");
        String fileName = "";
        switch (type){
            case "png":
                fileName = "1.png";
                break;
            case "txt":
                fileName = "test.txt";
                break;
        }
        //设置响应方式
        resp.setContentType("application/x-msdownload");
        //设置下载之后的文件名
        resp.setHeader("Content-Disposition", "attachment;filename="+fileName);
        //获取输出流
        OutputStream outputStream = resp.getOutputStream();
        String path = req.getServletContext().getRealPath("file/"+fileName);
        InputStream inputStream = new FileInputStream(path);
        int temp = 0;
        while((temp=inputStream.read())!=-1){
            outputStream.write(temp);
        }
        inputStream.close();
        outputStream.close();
    }
}
```

# Ajax

Asynchronous JavaScript And XML: 异步的 JavaScript 和 XML

AJAX 不是新的编程，指的是一种交互方式，异步加载，客户端和服务器的数据交互更新在局部页面的技术，不需要刷新整个页面（局部刷新）

优点：

- 1、局部刷新，效率更高
- 2、用户体验更好

基于 jQuery 的 AJAX

```
<!--
  Created by IntelliJ IDEA.
  User: southwind
  Date: 2019-12-10
  Time: 10:30
  To change this template use File | Settings | File Templates.
-->
<%@ page contentType="text/html; charset=UTF-8" language="java" %>
<html>
<head>
  <title>Title</title>
  <script type="text/javascript" src="js/jquery-3.3.1.min.js"></script>
  <script type="text/javascript">
    $(function(){
      var btn = $("#btn");
      btn.click(function(){
        $.ajax({
          url: '/test',
          type: 'post',
          data: 'id=1',
          dataType: 'text',
          success: function(data){
            var text = $("#text");
            text.before("<span>"+data+"</span><br/>");
          }
        });
      });
    })
  </script>
</head>
<body>
  <input id="text" type="text"/><br/>
  <input id="btn" type="button" value="提交"/>
</body>
```

```
</html>
```

不能用表单提交请求，改用 jQuery 方式动态绑定事件来提交。

Servlet 不能跳转到 JSP，只能将数据返回。

```
package com.southwind.servlet;

import javax.servlet.ServletException;
import javax.servlet.annotation.WebServlet;
import javax.servlet.http.HttpServlet;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;
import java.io.IOException;

@WebServlet("/test")
public class TestServlet extends HttpServlet {

    @Override
    protected void doPost(HttpServletRequest req, HttpServletResponse resp)
        throws ServletException, IOException {
        String id = req.getParameter("id");
        try {
            Thread.sleep(3000);
        } catch (InterruptedException e) {
            e.printStackTrace();
        }
        String str = "Hello World";
        resp.getWriter().write(str);
    }
}
```

## 传统的 WEB 数据交互 VS AJAX 数据交互

- 客户端请求的方式不同：

传统，浏览器发送同步请求（form、a）

AJAX，异步引擎对象发送异步请求

- 服务器响应的方式不同：

传统，响应一个完整 JSP 页面（视图）

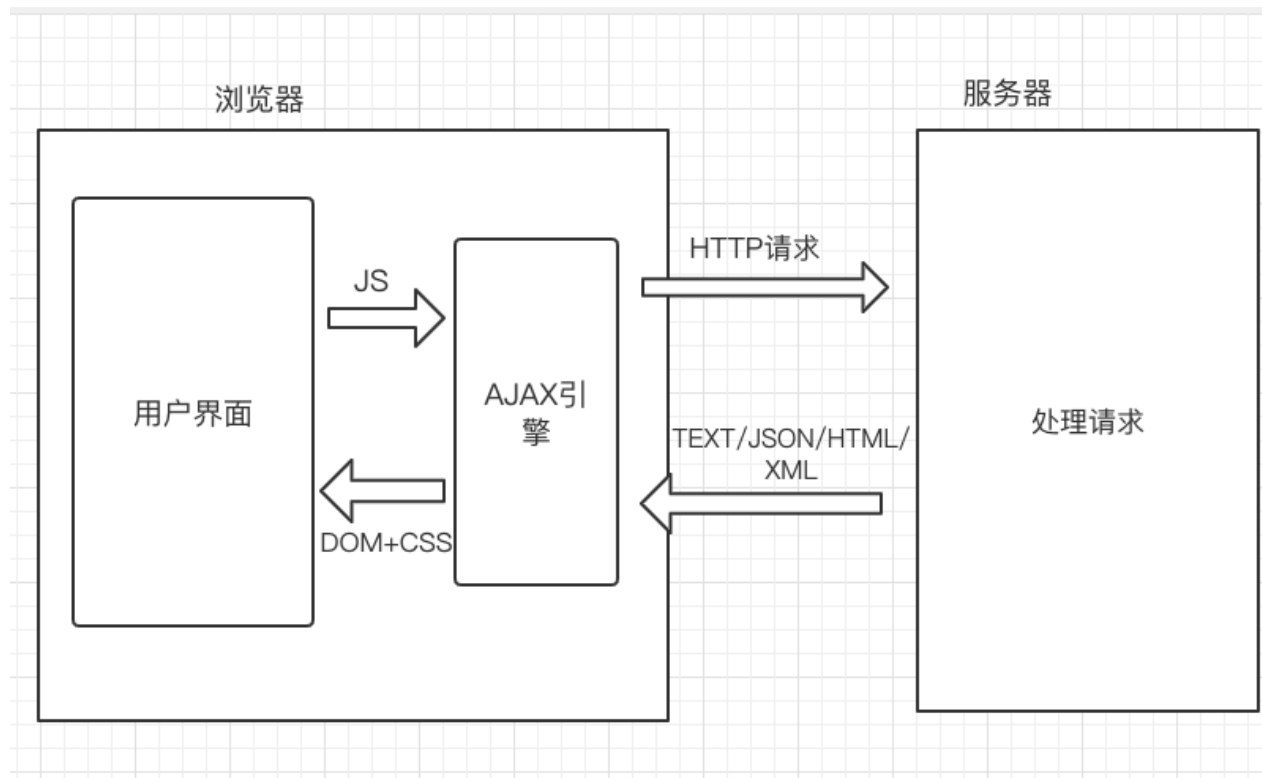
AJAX，响应需要的数据

- 客户端处理方式不同：

传统：需要等待服务器完成响应并且重新加载整个页面之后，用户才能进行后续的操作

AJAX：动态更新页面中的局部内容，不影响用户的其他操作

## AJAX 原理



## 基于 jQuery 的 AJAX 语法

`$.ajax({属性})`

常用的属性参数：

url：请求的后端服务地址

type：请求方式，默认 get

data：请求参数

dataType：服务器返回的数据类型，text/json

success：请求成功的回调函数

error：请求失败的回调函数

complete：请求完成的回调函数（无论成功或者失败，都会调用）

## JSON

JavaScript Object Notation，一种轻量级数据交互格式，完成 js 与 Java 等后端开发语言对象数据之间的转换

客户端和服务端之间传递对象数据，需要用JSON 格式。

```
package com.southwind.entity;

public class User {
    private Integer id;
    private String name;
    private Double score;

    public Integer getId() {
        return id;
    }

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

    public String getName() {
        return name;
    }

    public void setName(String name) {
        this.name = name;
    }

    public Double getScore() {
        return score;
    }

    public void setScore(Double score) {
        this.score = score;
    }

    public User(Integer id, String name, Double score) {
        this.id = id;
        this.name = name;
        this.score = score;
    }
}

User user = new User(1, "张三", 96.5);
```

```
var user = {
    id:1,
    name:"张三",
    score:96.5
}
```

```

package com.southwind.servlet;

import com.southwind.entity.User;
import net.sf.json.JSONObject;

import javax.servlet.ServletException;
import javax.servlet.annotation.WebServlet;
import javax.servlet.http.HttpServlet;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;
import java.io.IOException;

@WebServlet("/test")
public class TestServlet extends HttpServlet {

    @Override
    protected void doPost(HttpServletRequest req, HttpServletResponse resp)
throws ServletException, IOException {
        User user = new User(1, "张三", 96.5);
        //将 Java 对象转为 JSON 格式
        resp.setCharacterEncoding("UTF-8");
        JSONObject jsonObject = JSONObject.fromObject(user);
        resp.getWriter().write(jsonObject.toString());
    }
}

```

```

<%@ page contentType="text/html; charset=UTF-8" language="java" %>
<html>
<head>
    <title>Title</title>
    <script type="text/javascript" src="js/jquery-3.3.1.min.js"></script>
    <script type="text/javascript">
        $(function(){
            var btn = $("#btn");
            btn.click(function(){
                $.ajax({
                    url: '/test',
                    type: 'post',
                    dataType: 'json',
                    success: function(data){
                        $("#id").val(data.id);
                        $("#name").val(data.name);
                        $("#score").val(data.score);
                    }
                });
            });
        });
    </script>

```



```

        }
        $("#area").html(content);
    }
    });
});
//修改城市
$("#city").change(function(){
    var id = $(this).val();
    $.ajax({
        url:"/location",
        type:"POST",
        data:"id="+id+"&type=city",
        dataType:"JSON",
        success:function(data){
            var content = "";
            for(var i=0;i<data.length;i++){
                content += "<option>"+data[i]+"</option>";
            }
            $("#area").html(content);
        }
    });
});
});
</script>
</head>
<body>
    省: <select id="province">
        <option value="陕西省">陕西省</option>
        <option value="河南省">河南省</option>
        <option value="江苏省">江苏省</option>
    </select>
    市: <select id="city">
        <option value="西安市">西安市</option>
        <option value="宝鸡市">宝鸡市</option>
        <option value="渭南市">渭南市</option>
    </select>
    区: <select id="area">
        <option>雁塔区</option>
        <option>莲湖区</option>
        <option>新城区</option>
    </select>
</body>
</html>

```

```

package com.southwind.servlet;

import com.southwind.entity.Location;
import net.sf.json.JSONArray;
import net.sf.json.JSONObject;

```



```
import javax.servlet.ServletException;
import javax.servlet.annotation.WebServlet;
import javax.servlet.http.HttpServlet;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;
import java.io.IOException;
import java.util.ArrayList;
import java.util.HashMap;
import java.util.List;
import java.util.Map;

@WebServlet("/location")
public class LocationServlet extends HttpServlet {

    private static Map<String, List<String>> cityMap;
    private static Map<String, List<String>> provinceMap;

    static {
        cityMap = new HashMap<>();
        List<String> areas = new ArrayList<>();
        //西安
        areas.add("雁塔区");
        areas.add("莲湖区");
        areas.add("新城区");
        cityMap.put("西安市", areas);
        //宝鸡
        areas = new ArrayList<>();
        areas.add("陈仓区");
        areas.add("渭滨区");
        areas.add("新城区");
        cityMap.put("宝鸡市", areas);
        //渭南
        areas = new ArrayList<>();
        areas.add("临渭区");
        areas.add("高新区");
        cityMap.put("渭南市", areas);
        //郑州
        areas = new ArrayList<>();
        areas.add("郑州A区");
        areas.add("郑州B区");
        cityMap.put("郑州市", areas);
        //洛阳
        areas = new ArrayList<>();
        areas.add("洛阳A区");
        areas.add("洛阳B区");
        cityMap.put("洛阳市", areas);

        provinceMap = new HashMap<>();
    }
}
```

```

        List<String> cities = new ArrayList<>();
        cities.add("西安市");
        cities.add("宝鸡市");
        cities.add("渭南市");
        provinceMap.put("陕西省", cities);
        cities = new ArrayList<>();
        cities.add("郑州市");
        cities.add("洛阳市");
        cities.add("开封市");
        provinceMap.put("河南省", cities);
        cities = new ArrayList<>();
        cities.add("南京市");
        cities.add("苏州市");
        cities.add("南通市");
        provinceMap.put("江苏省", cities);
    }

    @Override
    protected void doGet(HttpServletRequest req, HttpServletResponse resp)
    throws ServletException, IOException {

    }

    @Override
    protected void doPost(HttpServletRequest req, HttpServletResponse resp)
    throws ServletException, IOException {
        String type = req.getParameter("type");
        resp.setCharacterEncoding("UTF-8");
        String id = req.getParameter("id");
        switch (type){
            case "city":
                List<String> areas = cityMap.get(id);
                JSONArray jsonArray = JSONArray.fromObject(areas);
                resp.getWriter().write(jsonArray.toString());
                break;
            case "province":
                List<String> cities = provinceMap.get(id);
                String city = cities.get(0);
                List<String> cityAreas = cityMap.get(city);
                Location location = new Location();
                location.setCities(cities);
                location.setAreas(cityAreas);
                JSONObject jsonObject = JSONObject.fromObject(location);
                resp.getWriter().write(jsonObject.toString());
                break;
        }
    }
}

```

```
package com.southwind.entity;

import java.util.List;

public class Location {
    private List<String> cities;
    private List<String> areas;

    public List<String> getCities() {
        return cities;
    }

    public void setCities(List<String> cities) {
        this.cities = cities;
    }

    public List<String> getAreas() {
        return areas;
    }

    public void setAreas(List<String> areas) {
        this.areas = areas;
    }
}
```