1. JSON概念 – 对象标记语言 – **使用字符串**

JSON(JavaScript Object Notation, JS 对象标记) 是**一种轻量级的数据交换格式**。它基于 ECMAScript (w3c制定的js规范)的一个子集，**采用完全独立于编程语言的文本格式来存储和表示数据。**

1. JSON和JS对象的关系

JSON 是 JS 对象的字符串表示法，它使用文本表示一个 JS 对象的信息，本质是一个字符串。

区别：

var obj = {a: 'Hello', b: 'World'}; //这是一个对象，注意键名也是可以使用引号包裹的

var json = '{"a": "Hello", "b": "World"}'; //这是一个 JSON 字符串，本质是一个字符串

1. JSON和JS对象的互相转换
2. 要实现从JS对象转换为 JSON 字符串，使用 JSON.stringify() 方法：

var json = JSON.stringify({a: 'Hello', b: 'World'}); //结果是 '{"a": "Hello", "b": "World"}'

1. 要实现从 JSON 转换为JS对象，使用 JSON.parse() 方法：

var obj = JSON.parse('{"a": "Hello", "b": "World"}'); //结果是 {a: 'Hello', b: 'World'}

1. JSON工作原理

JSON 可以将 JavaScript 对象中表示的一组数据转换为字符串，然后就可以在网络或者程序之间轻松地传递这个字符串，并在需要的时候将它还原为各编程语言所支持的数据格式，例如在 PHP 中，可以将 JSON 还原为数组或者一个基本对象。在用到AJAX时，如果需要用到数组传值，这时就需要用JSON将数组转化为字符串。

1. XML概念 – 主要特征是**使用标签**

可扩展标记语言，标准通用标记语言的子集，是一种用于标记电子文件使其具有结构性的标记语言。主要用来传输数据。标签没有被预定义，需要自行定义标签。

1. JSON和XML相同之处
2. JSON 是纯文本
3. JSON 具有"自我描述性"（人类可读）
4. JSON 具有层级结构（值中存在值）
5. JSON 可通过 JavaScript 进行解析
6. JSON 数据可使用 AJAX 进行传输
7. JSON和XML不同之处
8. 没有结束标签
9. 更短
10. 读写的速度更快
11. 能够使用内建的 JavaScript eval() 方法进行解析
12. 使用数组
13. 不使用保留字
14. 对于 AJAX 应用程序来说，JSON 比 XML 更快更易使用：

**使用 XML**

1. 读取 XML 文档
2. 使用 XML DOM 来循环遍历文档
3. 读取值并存储在变量中

**使用 JSON**

1. 读取 JSON 字符串
2. 用 eval() 处理 JSON 字符串
3. 相关方法
4. 将json字符串转换成js对象
5. eval方法、$.parseJSON方法、(new Function(“return”+json字符串))();

var result2 = eval("("+resp+")");

var result2 = $.parseJSON(resp);

var result2 = (new Function("return " + resp))();

alert(result2.model.pwdTime);

alert(resp);

1. jquery中调用$.ajax()方法

$.ajax({

url:'<%=request.getContextPath()%>/los/fzuche-intf-login.synServerConfig',

type:'POST',

datatype:'json',

success:function(resp){ //success中就是一个回调函数，调用成功后都在这里执行

//var result2 = eval("("+resp+")");

//var result2 = $.parseJSON(resp);

var result2 = (new Function("return " + resp))();

alert(result2.model.pwdTime);

alert(resp);

},

error:function(json){

alert("获取服务器时间报错");

}

})