**json和jsonp的使用区别**

**一．    跨域请求的概念**

　　JavaScript出于安全方面的考虑，不允许跨域调用其他页面的对象。



**二．    json和jsonp**

　　JSON是一种基于文本的数据交换方式（不支持跨域），而JSONP是一种非官方跨域数据交互协议。

　　使用json格式传递数据的客户端代码如下：

[复制代码](javascript:void(0);)

1 $(function () {

2 var user = {

3 "username": "HelloWorld"

4 };

5

6 $.ajax({

7 url: "http://localhost:8080/Changyou/UserInfo",

8 type: "POST",

9 contentType: "application/json; charset=utf-8",

10 dataType: "json", //json不支持跨域请求,只能使用jsonp

11 data: {

12 user: JSON.stringify(user)

13 },

14 success: function (data) {

15 $("#user\_name")[0].innerHTML = data.user\_name;

16 $("#user\_teleNum")[0].innerHTML = data.user\_teleNum;

17 $("#user\_ID")[0].innerHTML = data.user\_ID;

18 },

19 error: function () {

20 alert("请求超时错误!");

21 }

22 })

23 });

[复制代码](javascript:void(0);)

　　然而，简单地使用json并不能支持跨域资源请求，为了解决这个问题，需要采用jsonp数据交互协议。众所周知，js文件的调用不受跨域与否的限制，因此如果想通过纯web端跨域访问数据，只能在远程服务器上设法将json数据封装进js格式的文件中，供客户端调用和进一步处理，这就是jsonp协议的原理。该协议的一个要点就是允许用户传递一个callback参数给服务端，然后服务端返回数据时会将这个callback参数作为函数名来包裹住JSON数据，这样客户端就可以随意定制自己的函数来自动处理返回数据了。

　　简单的说，就是json不支持跨域，而js可以跨域，因此在服务器端用客户端提供的js函数名将json数据封装起来，再将函数提供给客户端调用，从而获得json数据。

　　开发过程中，如果出现类似 “Origin \*\*\*\*\*\* is not allowed by Access-Control-Allow-Origin.” 的错误，则可能是由于json数据不支持跨域导致的，应考虑使用jsonp协议。

　　如果出现类似 ”SyntaxError: Unexpected token ':'. Parse error.“ 的错误，则可能是由于返回的json数据没有用”callback“传递的函数名封装导致的。

　　客户端代码如下：

[复制代码](javascript:void(0);)

1 $(function () {

2

3 var user = {

4 "username": "HelloWorld"

5 };

6

7 $.ajax({

8 url: "http://localhost:8080/Changyou/UserInfo",

9 type: "POST",

10 contentType: "application/json; charset=utf-8",

11 dataType: "jsonp", //json不支持跨域请求,只能使用jsonp

12 data: {

13 user: JSON.stringify(user)

14 },

15 jsonp: "callback", //传递给请求处理程序或页面的，用以获得jsonp回调函数名的参数名,默认为callback

16 jsonpCallback: "userHandler", //自定义的jsonp回调函数名称，默认为jQuery自动生成的随机函数名，也可以写"?"，jQuery会自动为你处理数据

17 success: function (data) {

18 $("#user\_name")[0].innerHTML = data.user\_name;

19 $("#user\_teleNum")[0].innerHTML = data.user\_teleNum;

20 $("#user\_ID")[0].innerHTML = data.user\_ID;

21 },

22 error: function () {

23 alert("请求超时错误!");

24 }

25 })

26 });

[复制代码](javascript:void(0);)

 　　服务器端代码如下：

[复制代码](javascript:void(0);)

1 protected void doPost(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException {

2 response.setContentType("application/json; charset=utf-8");

3 String username = new String(request.getParameter("user").getBytes("ISO-8859-1"),"utf-8");

4 String callback = new String(request.getParameter("callback").getBytes("ISO-8859-1"),"utf-8");

5 System.out.println("接收到的数据：" + username);

6 System.out.println("callback的值：" + callback);

7 JSONObject user = JSONObject.fromObject(username);

8 System.out.println("接收到的用户名：" + user.get("username"));

9 JSONObject userinfo = new JSONObject();

10 userinfo.put("user\_name", "张鸣晓");

11 userinfo.put("user\_teleNum", "18810011111");

12 userinfo.put("user\_ID", "123456789098765432");

13 PrintWriter out = response.getWriter();

14 String backInfo = callback + "(" + userinfo.toString() + ")"; //将json数据封装在callback函数中提供给客户端

15 out.print(backInfo);

16 out.close();

17 }

[复制代码](javascript:void(0);)

尽管客户端没有实现userHandler函数，但也能成功运行，原因就是jquery在处理jsonp类型的ajax时，自动帮你生成回调函数并把数据取出来供success属性方法来调用。

更详细解释，参见<http://kb.cnblogs.com/page/139725/>