





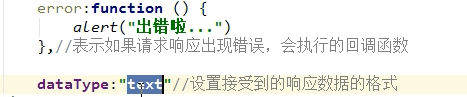


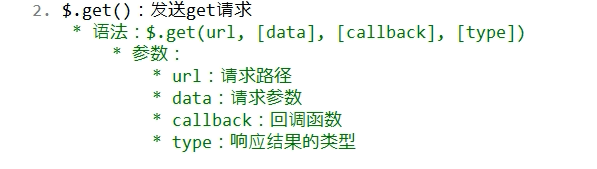


ajax的jquery实现方式：$.ajax( {键值对} )：通用的实现方式，



其它键值对：





中括号了的表示可选参数

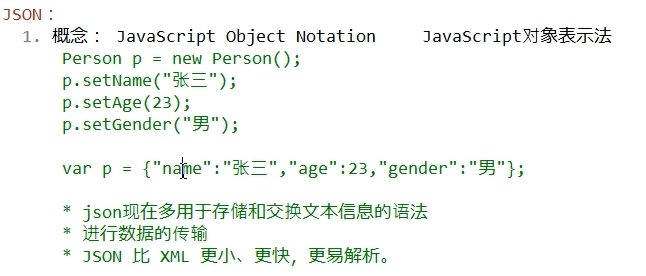


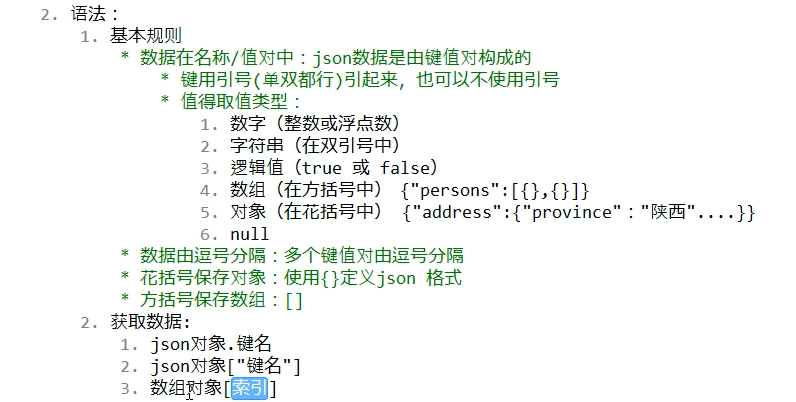


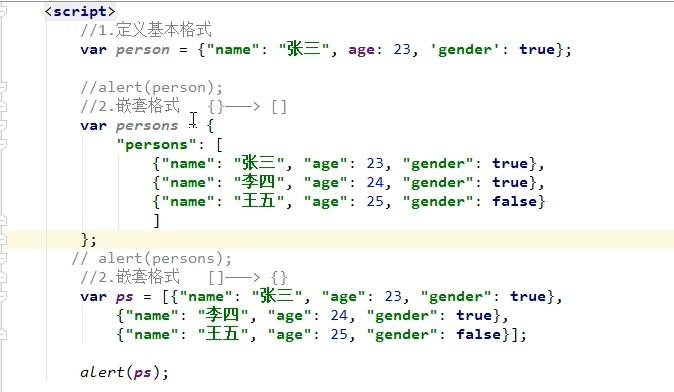
格式与get方式相同

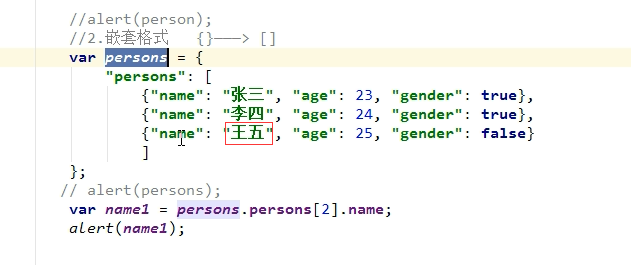
|  |
| --- |
| <!DOCTYPE html> <html lang="en"> <head>  <meta charset="UTF-8">  <title>Title</title>  <script src="/static/js/jquery-3.6.0.js"></script>  <script>  function *fun*(){  $.post("../ajaxServlet",{username:"Jerry"},function (data){  *alert*(data);  },"text")  }  </script> </head> <body> <input type="button" value="发送异步请求" onclick="*fun*();"> <input> </body> </html> |

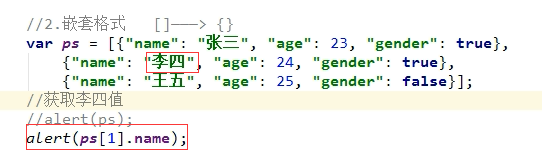
JSON





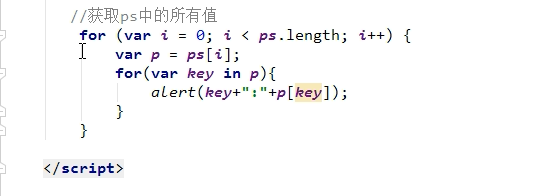


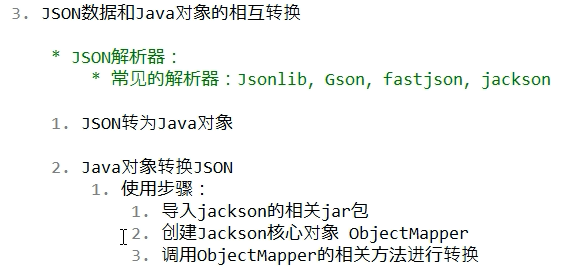




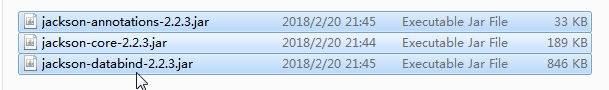
遍历JSON对象：







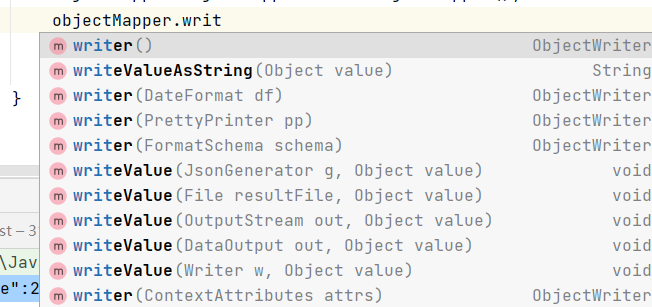
Jackson是SpringMVC内置的JSON解析器



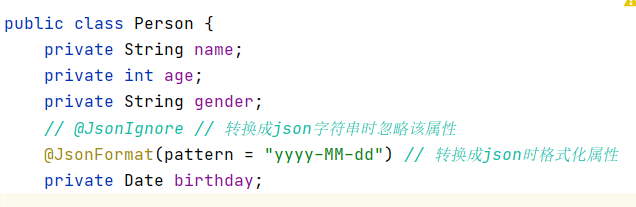
Java对象转JSON字符串

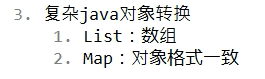


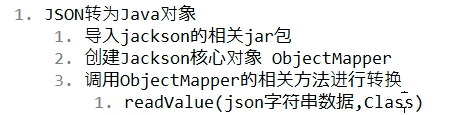
|  |
| --- |
| *// Java对象转为JSON字符串* @Test  public void test1() throws IOException {  Person person = new Person();  person.setName("张三");  person.setAge(23);  person.setGender("男");  *// 创建Jackson的核心对象，ObjectMapper* ObjectMapper objectMapper = new ObjectMapper();  String json = objectMapper.writeValueAsString(person);  System.*out*.println(json); *// {"name":"张三","age":23,"gender":"男"}* objectMapper.writeValue(new File("d://a.txt"),person);  objectMapper.writeValue(new FileWriter("d://b.txt"),person);  } } |

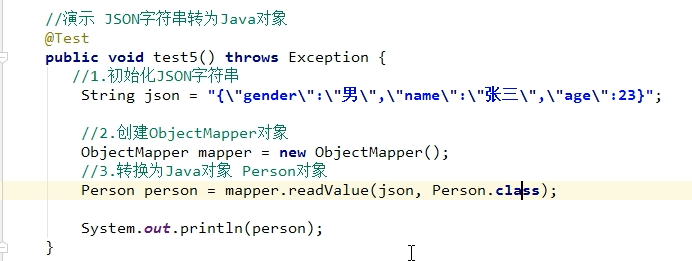


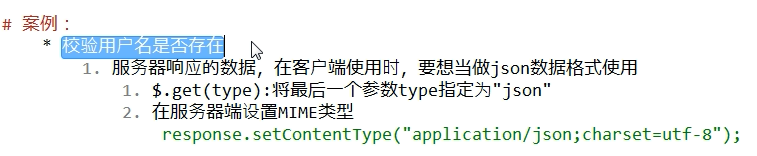












对于$.ajax的datatype参数：



|  |
| --- |
| <!DOCTYPE html> <html lang="en"> <head>  <meta charset="UTF-8">  <title>注册页面</title>  <script src="static/js/jquery-3.6.0.js"></script>  <script>  *// 在页面加载完成后执行* $(function (){  *// 给username绑定blur事件* $("#username").blur(function (){  *// 获取username文本输入框的值* var username = $(this).val();  *// 发送ajax请求  // 期望服务器响应的数据：{"userExist":true,"msg":"用户名已存在“}  // {"userExist":false,"msg":"用户名可用”}* $.get("findUserServlet",{username:username},function (data){  *// 判断userExist键的值是否为true* var span = $("#s\_username");  if (data.userExist){  *// 用户名已存在* span.css("color","red");  span.html(data.msg);  }else {  *// 用户名不存在* span.css("color","green");  span.html(data.msg);  }  }); *// ,"json"指定接收的服务器返回的数据格式为json，如果不指定，jquery将自动根据Http包MIME信息智能判断* });  });  </script> </head> <body>  <form>  <input type="text" id="username" name="username" placeholder="请输入用户名">  <span id="s\_username"></span>  <br>  <input type="password" name="password" placeholder="请输入密码"><br>  <input type="submit" value="注册">  </form>  </body> </html> |

|  |
| --- |
| @WebServlet("/findUserServlet") public class FindUserServlet extends HttpServlet {  @Override  protected void doGet(HttpServletRequest req, HttpServletResponse resp) throws ServletException, IOException {  this.doPost(req,resp);  }   @Override  protected void doPost(HttpServletRequest req, HttpServletResponse resp) throws ServletException, IOException {  *// 获取用户名* String username = req.getParameter("username");  *// 解决乱码问题  // resp.setContentType("text/html;charset=UTF-8");  // 设置响应的数据格式为json，application/json是json对应的mime-type* resp.setContentType("application/json;charset=UTF-8");  *// 调用service层判断用户名是否存在* Map<String,Object> map = new HashMap<String,Object>();   if ("Jerry".equals(username)){  *// 用户名存在* map.put("userExist",true);  map.put("msg","用户名已存在");  } else {  *// 用户名不存在* map.put("userExist",false);  map.put("msg","用户名可用");  }  *// 将map转为json，并传递给客户端浏览器* ObjectMapper mapper = new ObjectMapper();  mapper.writeValue(resp.getWriter(),map);  } } |