

使用 ActionContext 类获取

1、获取 ActionContext 类对象

Map<String, Object> getParameters()	返回一个包含所有 HttpServletRequest 参数信
-------------------------------------	---------------------------------

- (1)、因为这个不是静态方法，需要创建 ActionContext 类对象
- (2)、这个 ActionContext 类对象不是 new 出来的

static ActionContext getContext()	获取当前线程的 ActionContext 对象。
-----------------------------------	---------------------------

2、具体演示

- (1)、创建表单，提交表单到action 里面
- (2)、在 action 使用 ActionContext 获取数据

```
1      @Override
2      public String execute() throws Exception {
3          System.out.println("execute method in FormAction");
4          ActionContext context = ActionContext.getContext();
5          Map<String, Object> parameters = context.getParameters();
6          for (String key : parameters.keySet()) {
7              System.out.print(key);
8              /*
9               * map 里面的value的值是一个 object 的数组
10              * 因为因为里面是可能存在多选的情况所以这里是一个数组来存的
11              */
12              Object[] object = (Object[])parameters.get(key);
13
14              System.out.println(Arrays.toString(object));
15          }
16          return Action.SUCCESS;
17      }
18  }
```

注意：map 中存的是 <String, object> 这个 Object 是一个 Object 的数组，因为这个值可能多选的情况有多个。

使用 ServletContext 类来获取

1、获取方法

- static HttpServletRequest getRequest(): 获取 Web 应用的 HttpServletRequest 对象。
- static HttpServletResponse getResponse(): 获取 Web 应用的 HttpServletResponse 对象。
- static ServletContext getServletContext(): 获取 Web 应用的 ServletContext 对象。
- static PageContext getPageContext(): 获取 Web 应用的 PageContext 对象。

2、使用方法

(1)、通过调用类里面的静态方法，得到 request 对象

```
1      @Override
2      public String execute() throws Exception {
3          HttpServletRequest request = ServletActionContext.getRequest();
4          String name = request.getParameter("username");
5          String pswd = request.getParameter("password");
6          String age = request.getParameter("age");
7          System.out.println(name+" "+pswd+" "+age);
8          return Action.NONE;
9      }
```

使用接口注入（了解）

1、让 action 实现接口 ServletRequestAware，为了得到 request

```
1      private HttpServletRequest req;
2      @Override
3      public void setServletRequest(HttpServletRequest request) {
4          this.req = request;
5      }
6
7      @Override
8      public String execute() throws Exception {
9          String name = req.getParameter("username");
10         System.out.println(name);
11         return Action.NONE;
12     }
```

注：港曾，感觉很鸡肋。。。。

在 action 操作域对象

1、request、session、servletContext 域对象

2、使用 ServletActionContext 类操作

```
1      request.setAttribute("request", "reqValue");
2
3      HttpSession session = request.getSession();
4      session.setAttribute("session", "sessioinValue");
5
6      ServletContext servletContext = request.getServletContext();
7      servletContext.setAttribute("context", "ctxtValue");
```