

Struct-like structure

Controller extends HttpServlet

//Process the HTTP Get request

```
public void doGet(HttpServletRequest request,  
    HttpServletResponse response) throws  
    ServletException, IOException {  
    processRequest(request,response);  
}
```

//Process the HTTP Post request

```
public void doPost(HttpServletRequest request,  
    HttpServletResponse response) throws  
    ServletException, IOException {  
    processRequest(request,response);  
}
```

Controller extends HttpServlet

```
public void processRequest(HttpServletRequest request,
    HttpServletResponse response) throws ServletException, IOException {
    RequestDispatcher dispatcher = null;
    String pathInfoWithSlash = request.getPathInfo();
    if (pathInfoWithSlash == null || pathInfoWithSlash.substring(1) == null) {
        request.setAttribute("NULL_ERROR", ControllerErrors.PAGE_NOT_FOUND);
        dispatcher = this.getServletContext().getRequestDispatcher("/errorPage.jsp");
    } else
        String actionName = pathInfoWithSlash.substring(1);
        if (actionName == null) {
            request.setAttribute("NULL_ERROR", ControllerErrors.PAGE_NOT_FOUND);
            dispatcher = this.getServletContext().getRequestDispatcher("/errorPage.jsp");
        }
        else {
            Action action = ActionHelper.getAction(actionName);
```

Controller extends HttpServlet

```
if (action == null) {
    request.setAttribute("NULL_ERROR", ControllerErrors.PAGE_NOT_FOUND);
    dispatcher = this.getServletContext().getRequestDispatcher("/errorPage.jsp");
}
else {
    String nextPage = action.process(request, response);
    if (nextPage == null) {
        System.err.println("ERROR: actionName = " + actionName + " --> " +
            request.getAttribute("NULL_ERROR"));
        dispatcher = this.getServletContext().getRequestDispatcher("/errorPage.jsp");
    } else {
        dispatcher = this.getServletContext().getRequestDispatcher(nextPage);
    }
}
}
}
}
}
}
dispatcher.forward(request,response);
}
```

public class ActionHelper

```
public static HashMap actions = new HashMap();
static {
    actions.put("ViewHome", "com.travelagency.web.action.ViewHomeAction");
    actions.put("Login", "com.travelagency.web.action.LoginAction");
}
public static Action getAction(String actionName) {
    Action action = null;
    try {
        Class actionClass = Class.forName((String) actions.get(actionName));
        action = (Action) actionClass.newInstance();
    }
    catch (Exception e) {
        System.err.println("Exception in ActionHelper");
        e.printStackTrace();
    }
    return action;
}
```

Class Action

```
/*  
 * Abstract Action class for the Command and Controller  
 * pattern  
 */  
  
public abstract class Action {  
  
    public abstract String process  
        (HttpServletRequest request, HttpServletResponse response);  
}
```

Xaction extends Action

```
public class AddUserAction extends Action {  
    public String process(  
        HttpServletRequest request, HttpServletResponse response) {  
        String nextPage = null;  
        // do your business logic  
        if (everythingIsFine)  
            nextPage =  
                ActionPageMapping.getPage(this.getClass().getName() + ":ok");  
        else  
            nextPage =  
                ActionPageMapping.getPage(this.getClass().getName() + ":error");  
        return nextPage;  
    }  
}
```

ActionPageMapping

```
public class ActionPageMapping {  
    public static HashMap mappings = new HashMap();  
  
    static {  
        mappings.put("com.travelagency.web.action.LoginAction:ok",  
            "/home.jsp");  
        mappings.put("com.travelagency.web.action.LoginAction:error",  
            "/login.jsp");  
        ...  
    }  
    public static String getPage(String className) {  
        return (String) mappings.get(className);  
    }  
}
```