

## STIFWeb programming convention

All actions will be done using contexts instead of parameters. For example:

`/addtarget <-----` will be associated with servlet that will add new target

`/addclient <-----` will be associated with servlet that will add new client

...

etc

That association will be done in *web.xml* file.

The way we are going to implement MVC (Model Vision Controller) design pattern is as follows.

1. Servlet processes the request (i.e. Adds new client, retrieves list of hosts for client, etc) and put the data into either one:

- Request variable (it will only be visible in the scope of this request):

```
SomeClass value = new SomeClass(...);
request.setAttribute("key", value);
```

- Session variable (it will only be visible in the scope of this session):

```
SomeClass value = new SomeClass(...);
HttpSession session = request.getSession(true);
session.setAttribute("key", value);
```

- Servlet context variable (it will be visible by all requests from all sessions, i.e. global):

```
SomeClass value = new SomeClass(...);
getServletContext().setAttribute("key", value);
```

SomeClass must be a bean, that is it must provide `setXxxx/getXxxx` methods.

2. Servlet redirects request to corresponding JSP:

```
RequestDispatcher dispatcher =
    getServletContext().getRequestDispatcher("/stifweb/hosts.jsp");
dispatcher.forward(request, response);
```

3. JSP retrieves SomeClass from request/session/servlet context scope and displays it:

```
<jsp:useBean id="customer" class="moreservlets.TravelCustomer"
scope="session" />
```

and retrieves data using get methods:

```
<jsp:getProperty name="customer" property="fullName" />
```

## NOTES:

- to avoid complex JSP stuff i suggest that we do STIFWeb in similar way as shown here:  
<http://archive.moreservlets.com/JSP-Review-Code/moreservlets/TravelCustomer.java>  
That is there are getXxxx methods that return HTML representation of the data the bean represents. This way JSP would only call getHostTable method and HostList class would return HTML representation (i.e. HTML table) and look and feel would be controlled using CSS using div/span