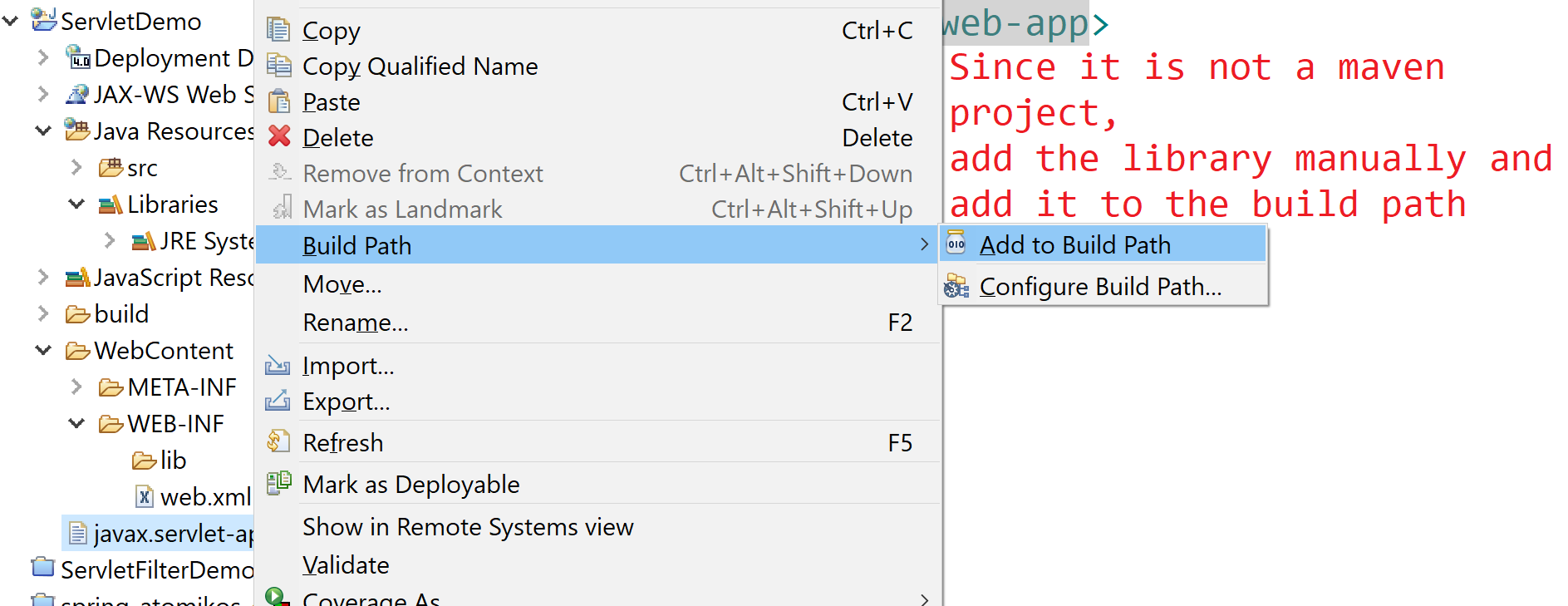


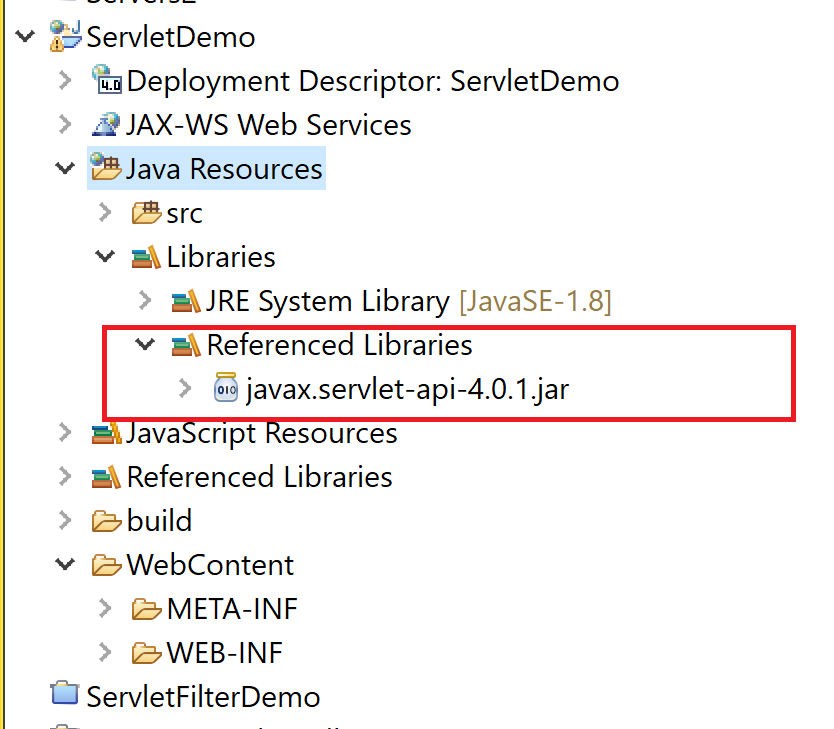
If I add the server to my project, I don’t need to add the server dependencies like servlet-api , jsp library dependencies. These are already present on the server.

Instead, add the libraries to the project and keep the scope to be **provided**.

1. Provided: Library shud only be there at compile time and not at runtime.
2. Compile: Library shud be there at compile time, runtime, testing
3. Runtime: Library is needed only at runtime, not during ur compile time
4. Test: Library is only needed for testing.

These scopes are valid only for a maven project

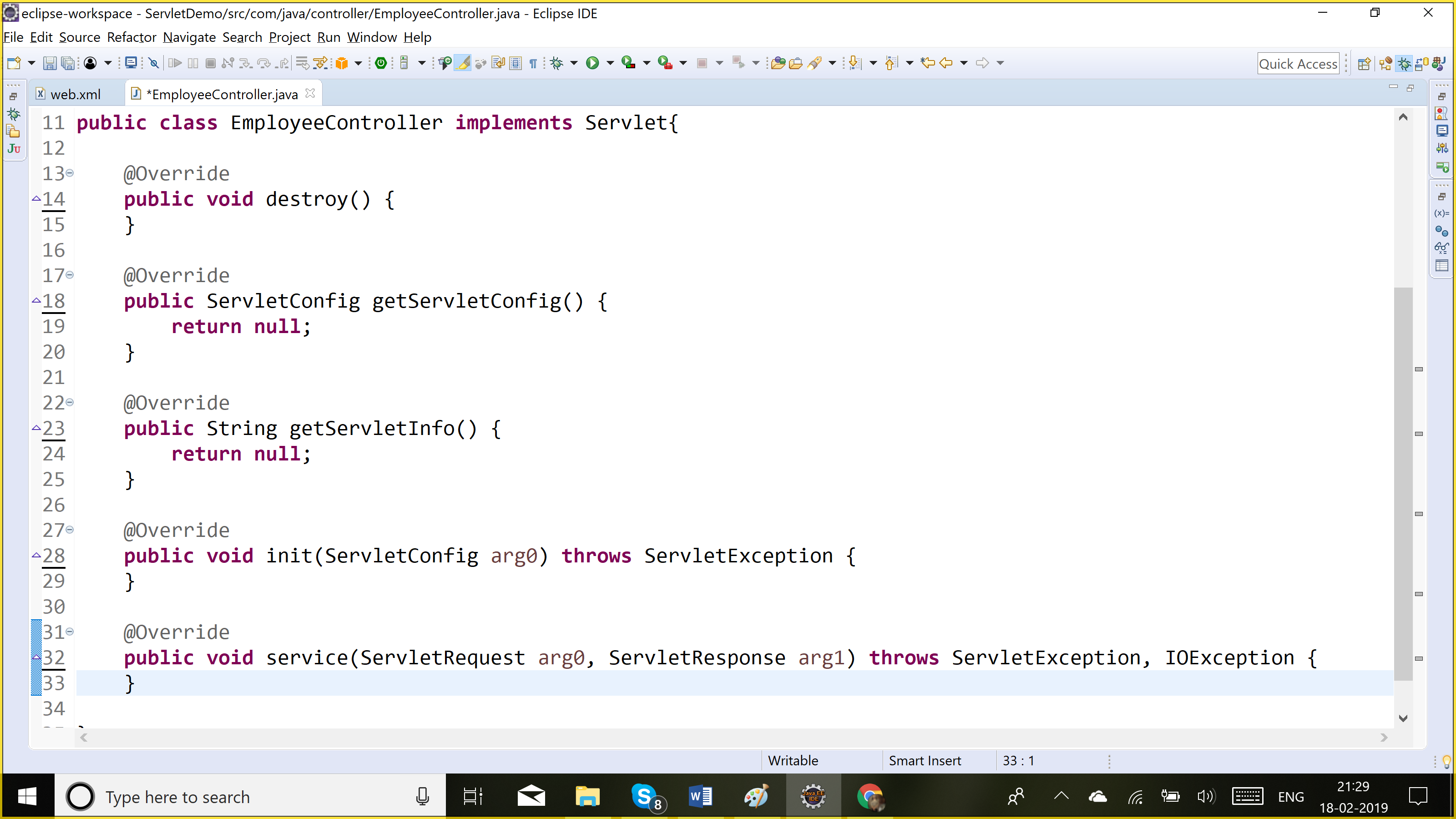




So now, u can see the library in ur library folder.

Three ways of creating a Servlet:

1. Implements Servlet: Override ->



**Init method:** It is called only once when the servlet object is created. To initialize the servlet

**Service:** Called for each and every request

**Destroy:** This is called only once before the object is removed from the container

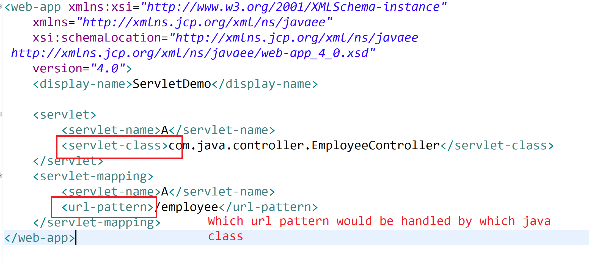
**getServletInfo:** U can override this method to give any information about ur servlet which is a string

**getServletConfig:** Here, u return the config object

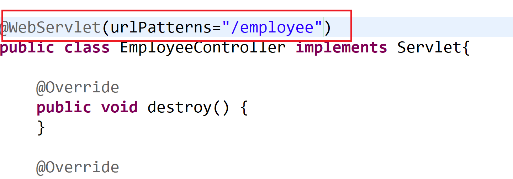
1. Extends GenericServlet: We need to override the service method. It was abstract
2. Extends HttpServlet: It is an abstract class with no abstract methods. So we should not override service method as service method internally makes a call to doGet, doPost, doPut, doDelete methods based on the request. So u should override particular method doGet/ dPost based on type of request.

Once u create the servlet, u need to do the url mapping for it that means which requests will be handled by this servlet

1. Xml config: In web.xml



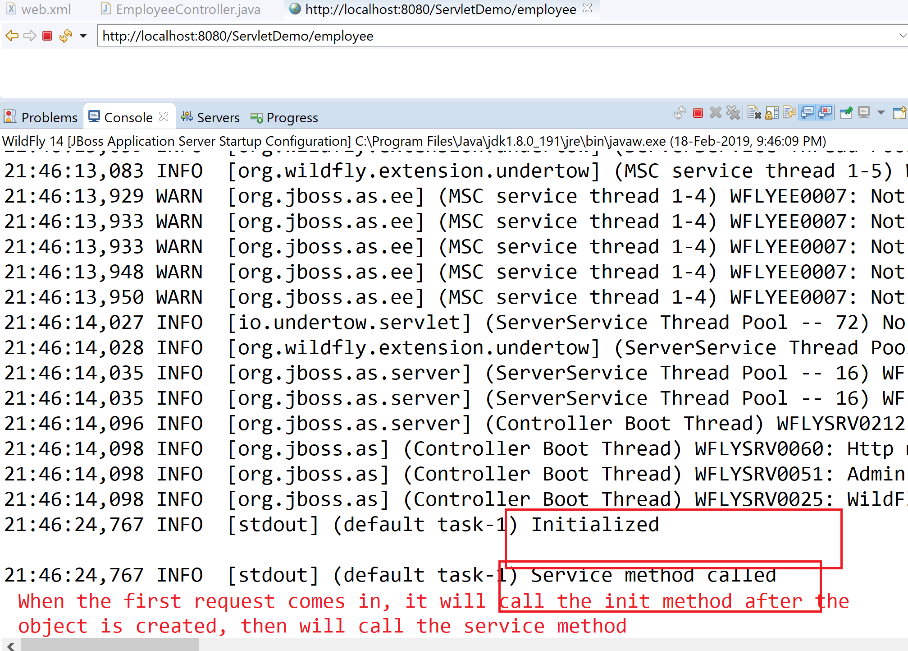
1. Annotations: @WebServlet(“/url”): This is a recommended approach



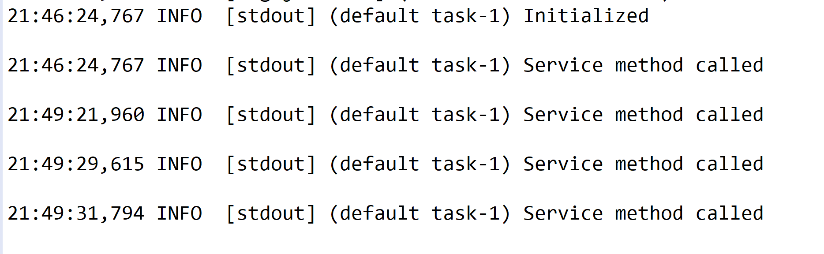
**Lifecycle of servlet:**

When the first request comes in, the container creates the object of the servlet, calls the init method to do the initialization, then calls the service method. Even for 100 requests for a given servlet, only 1 object would be created which is done at the beginning only. So, servlets are said to be multi-threaded. Each thread would handle 1 request. For other requests, it will just call the service method which will handled by each thread. And when the application is undeployed/ container is stopped, then the servlet object is destroyed and the destroy method is called.

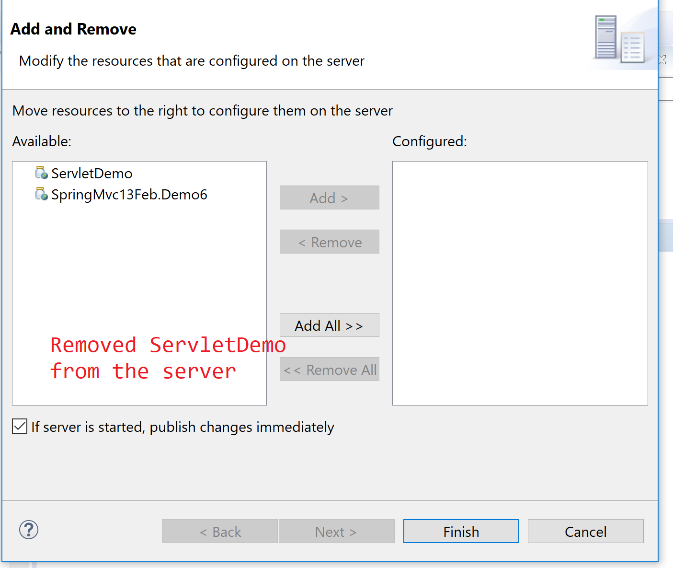
1. Init()
2. Service()
3. Destroy()



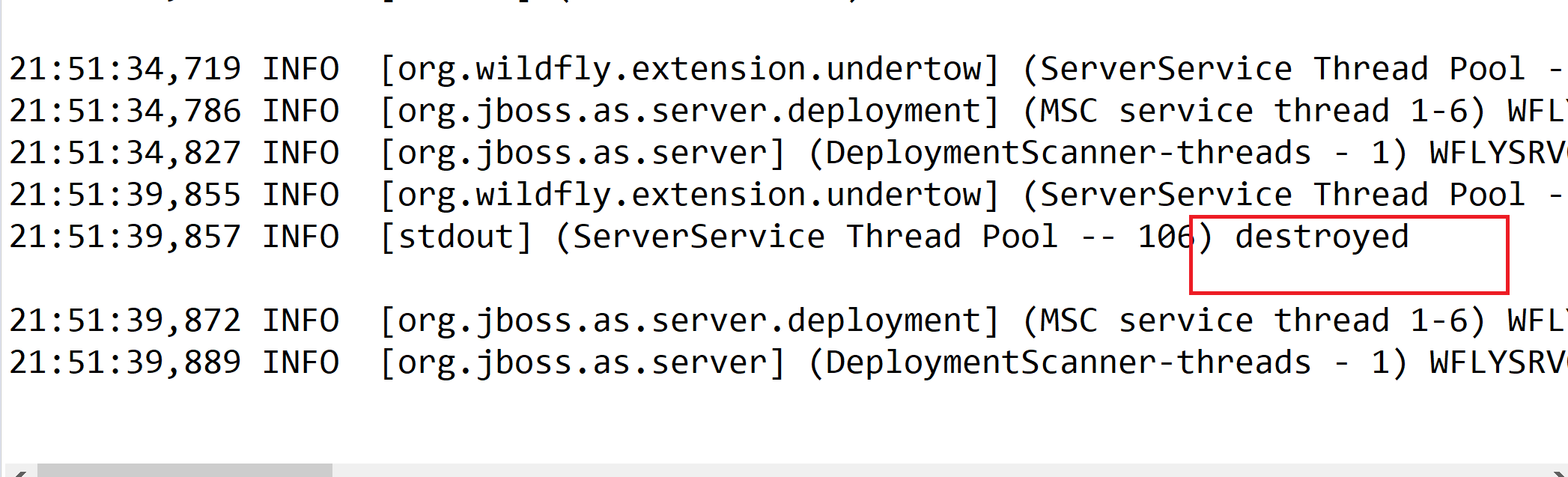
For other requests,it will just call the service method



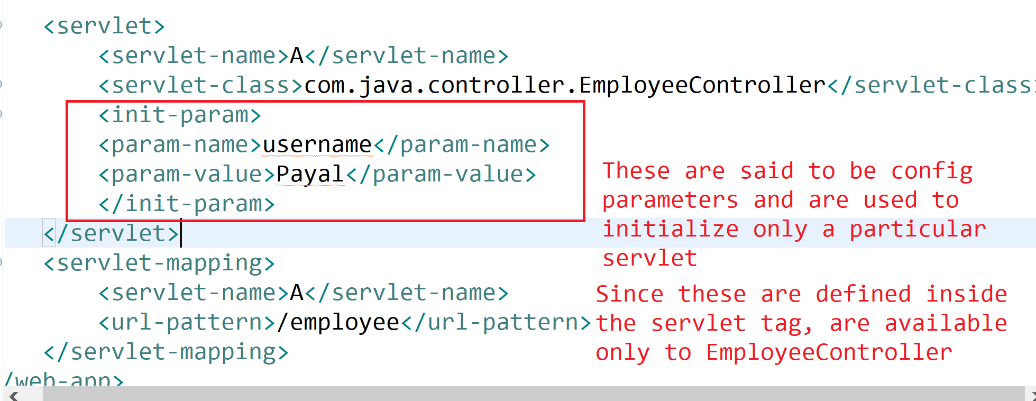
If undeploy my application,

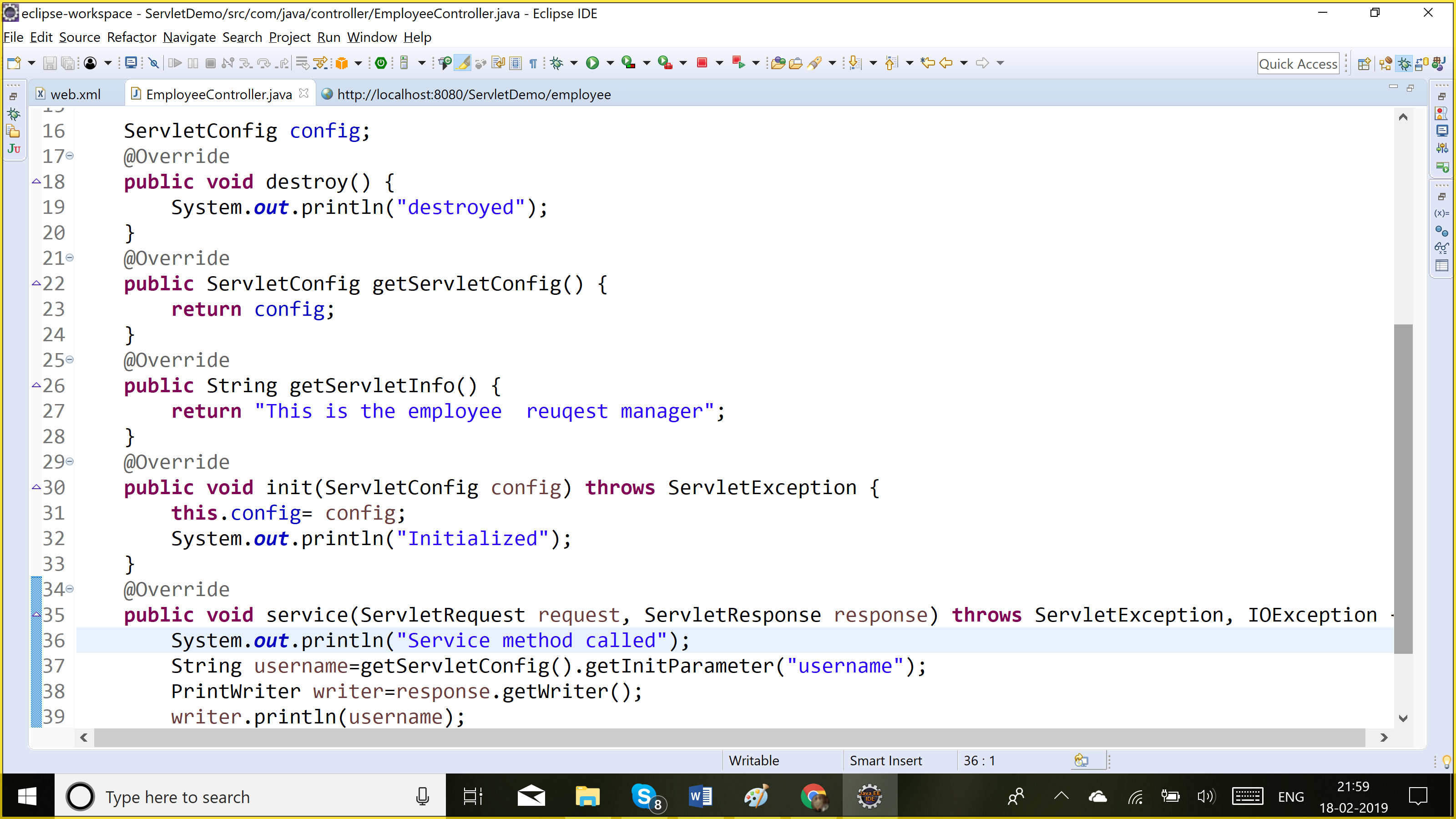


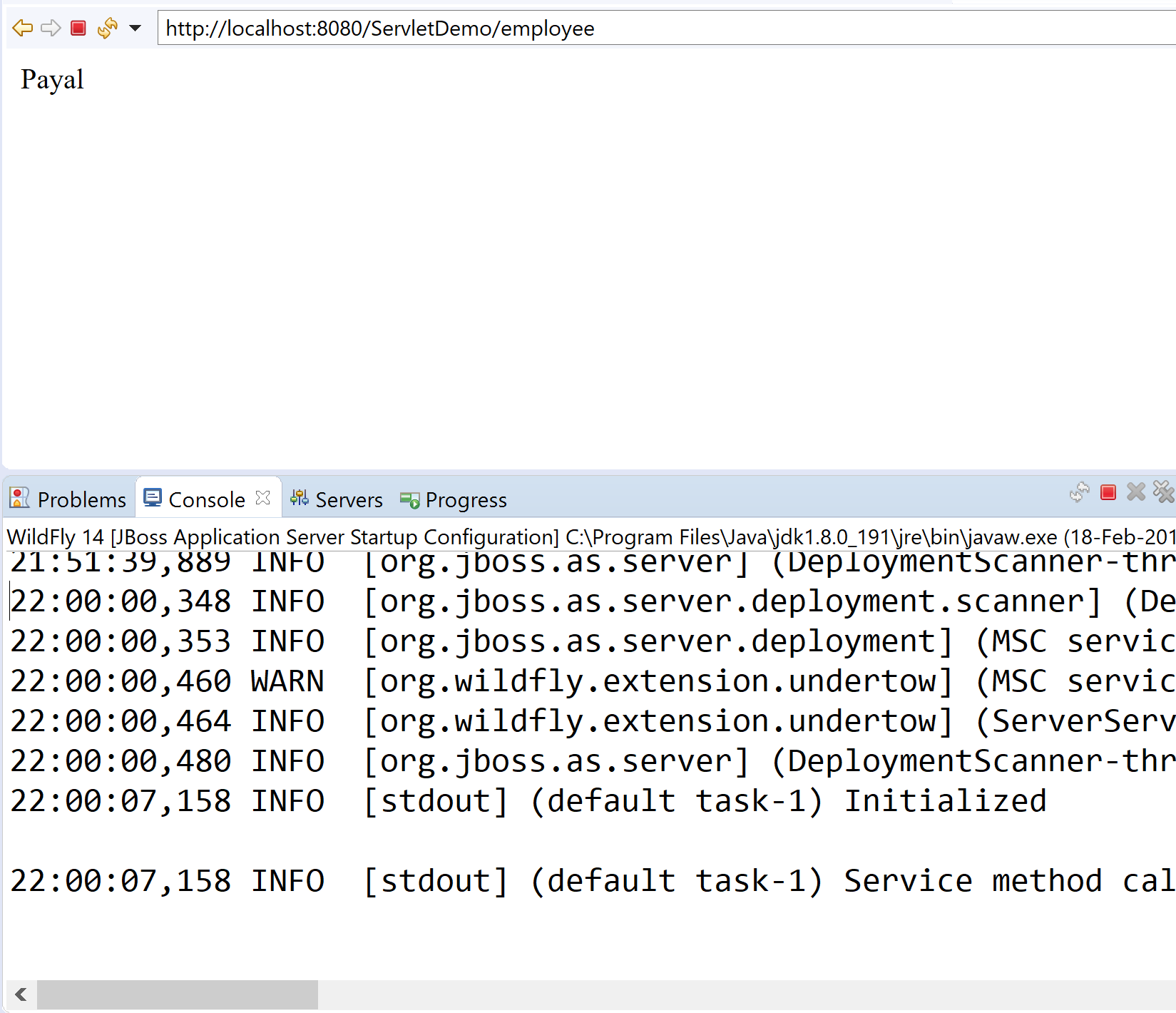
Servlet object would be removed from the container and destroy method would be called



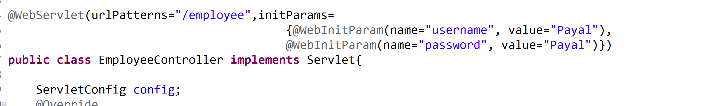
If I want to pass some parameters to my servlet during initialization:



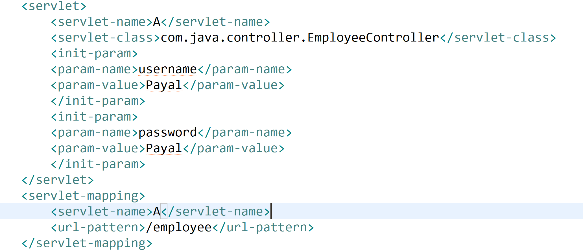




If using annotations, instead of web.xml to define config parameters:

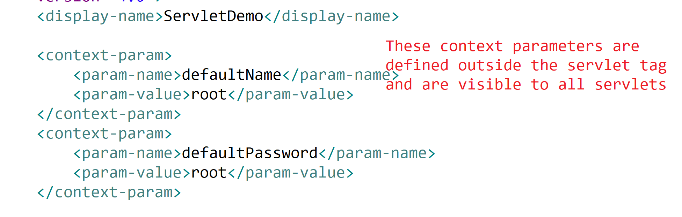


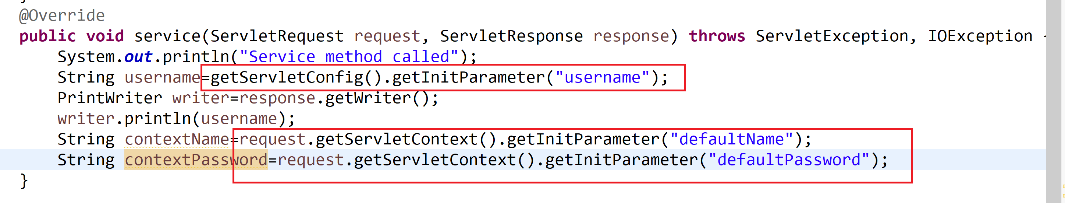
This is equivalent to:



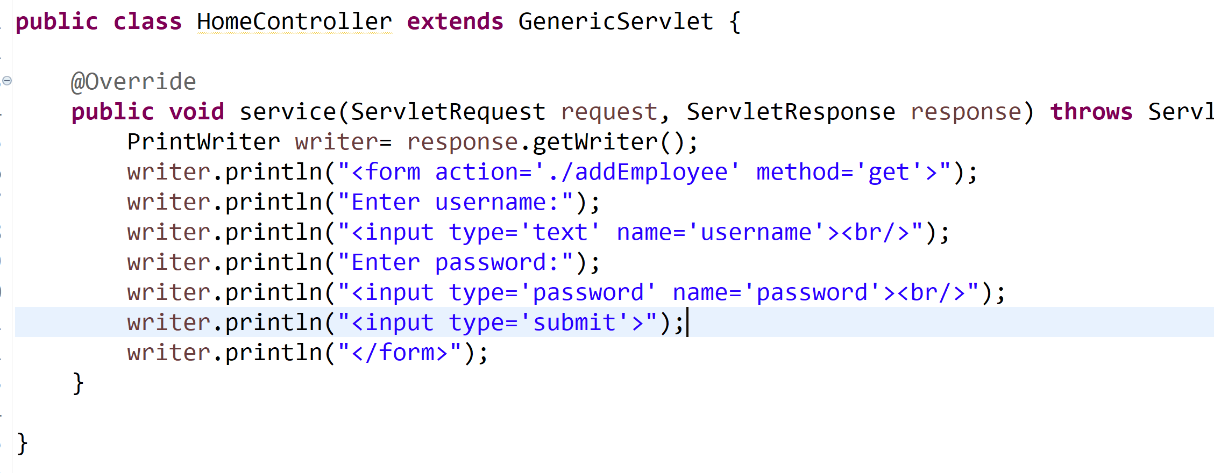
These config parameters are only available in the servlet in which they are defined.

If we want to define initial parameters which should be available to all servlets, we should define them as context parameters.





Root context: all ur servlets are available. So visible to all.

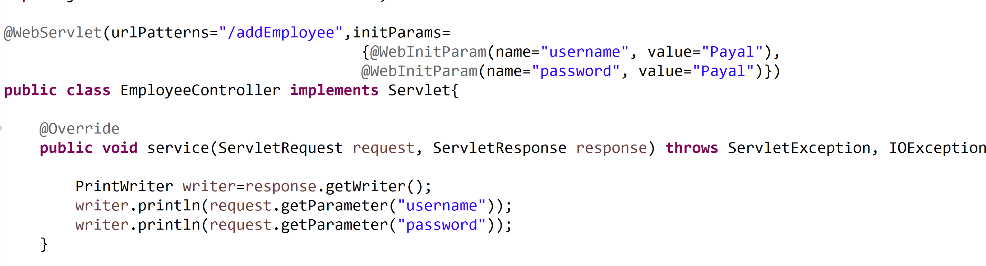


By-default the form method is get. So it appends every user input to the url.

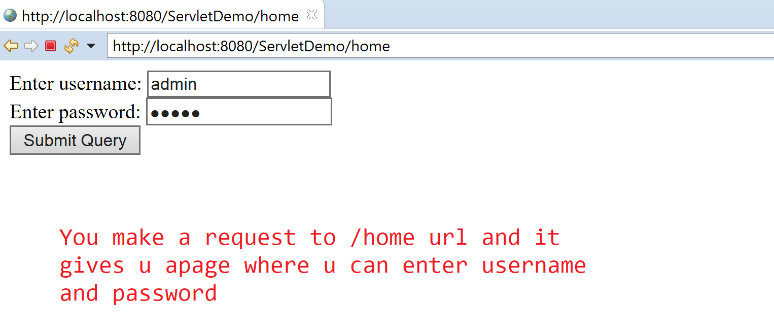
So when user makes a request to the /home url, it displays the the form

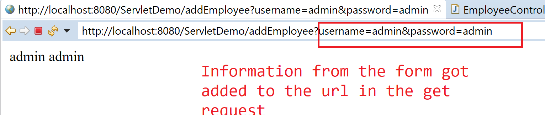


Then when he clicks on submit button, a new request would ba made to /addEmployee url and the username and password values would be passed as request parameters.

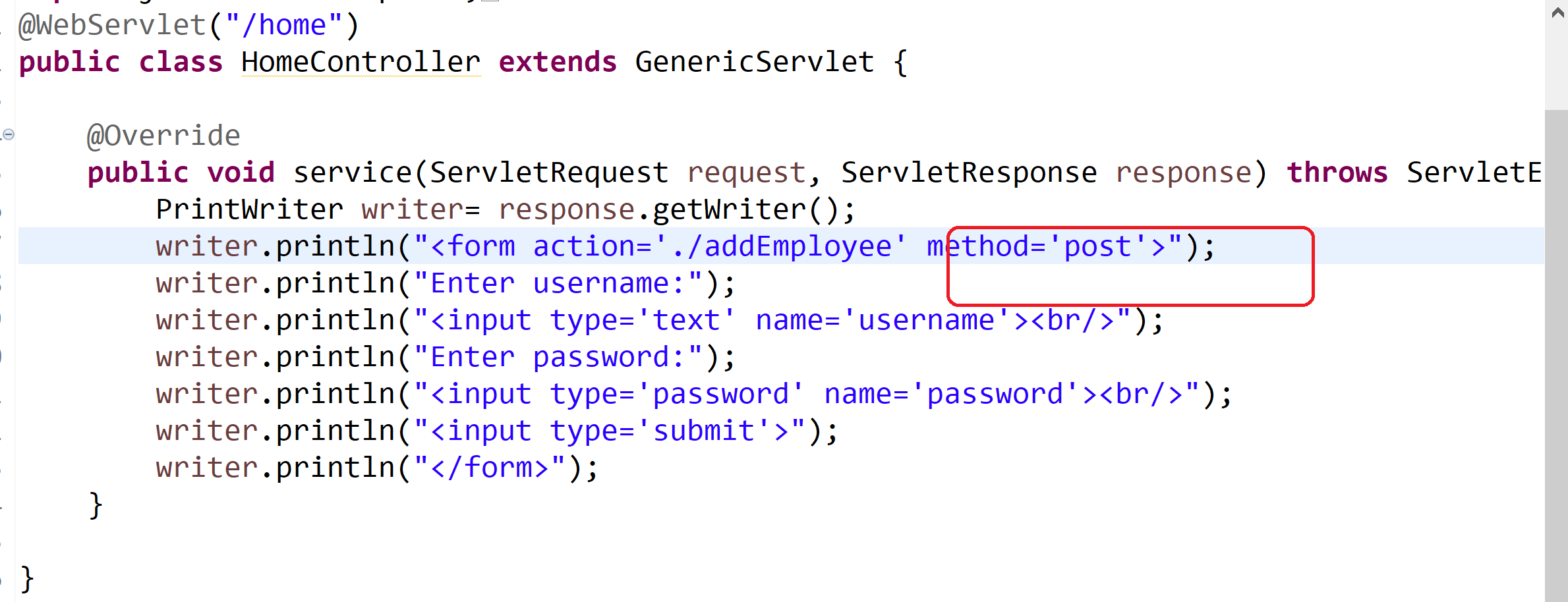


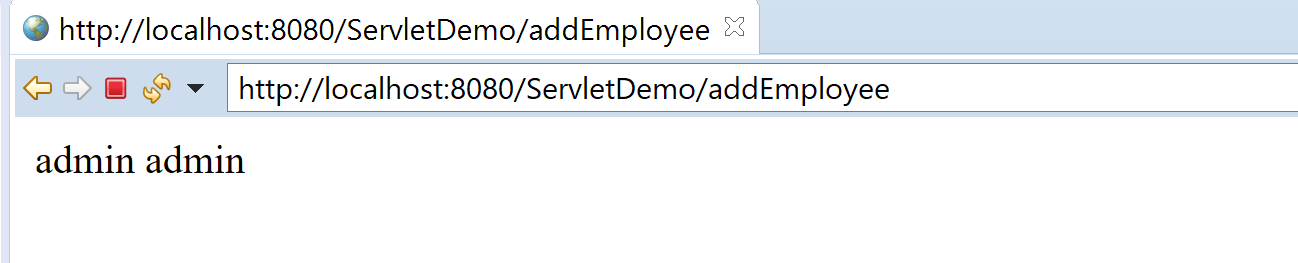
Then the request is handled by EmployeeController which returns a response displaying username and password.





But when we submit a form, we don’t want to see the filled details in the url (coz of security) so we change the method to POST.





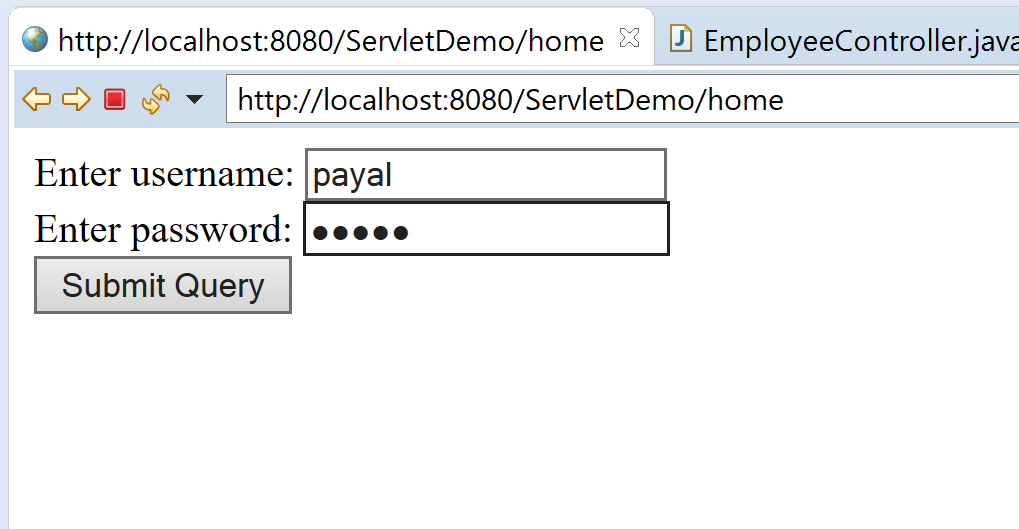
In the post request the data is transferred in the body whereas in the get, the data is transferred in the url.

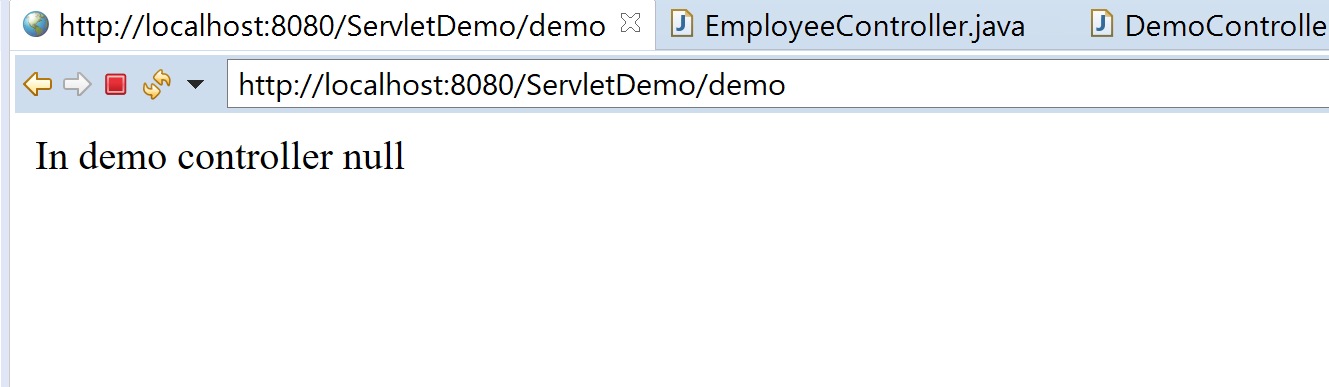
Generally we don’t want to see the filled in details as part of the url, so we use post for the forms.

**How do we send request from 1 page to other**

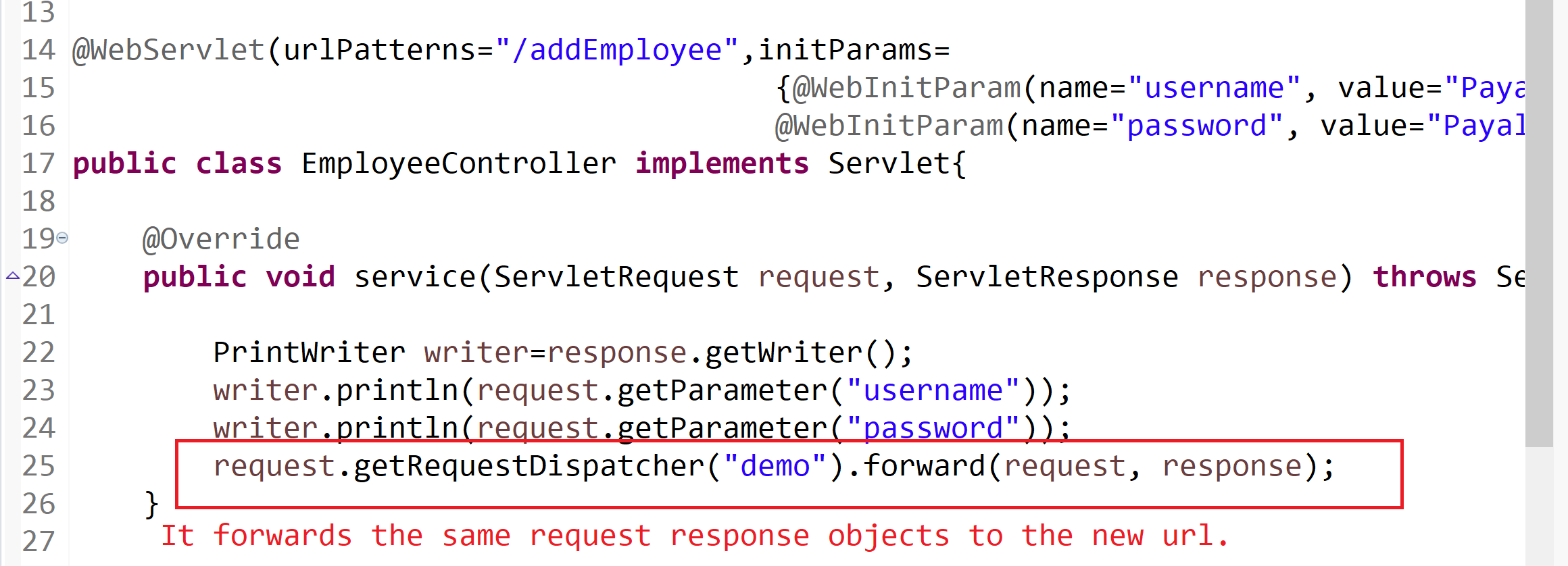
1. Using form and a submit button. In the action u define, where to take the request to
2. Response.sendRedirect(url);: It will send the response to the browser to make a new request to this url.



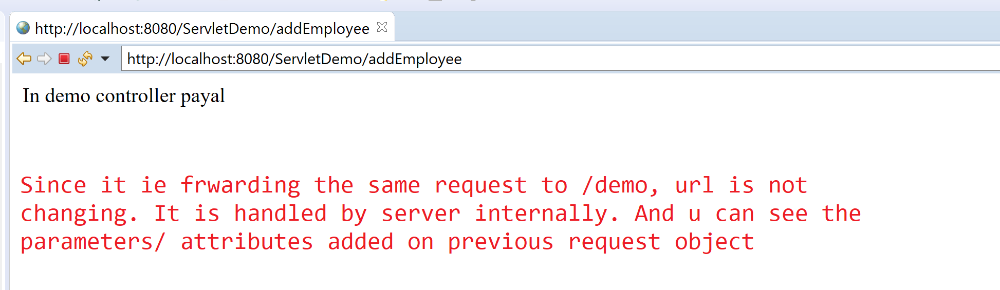




Url changed to demo. Coz it was a new request to /demo url.



Since it is forwarding the same request, url will not change



And here u should override doPost since the post request was made to /addEmployee url and same request is forwarded to /demo url.

***Quick Review***

1. Various ways of creating servlet
2. 5 servlet methods
3. Lifecycle of a servlet
4. Request Parameters:
5. Servlet config, context object / parameters
6. Inter-Servlet communication: response.sendRedirect(), request.getRequestDispatcher.forward(request, response)
7. Diff between Get & post

**Bydefault, http protocol is a stateless protocol. U make a request to a url, it gives u a response and keeps no information about ur previous requests. Every request is treated as a new request.**

So you have to mantain ur own session. Currently we can only transfer data in each request.

**Shopping cart application**: So what u save on the first page, should be available on the last page. And unitl u logout, it should remember u.

How to maintain a session in servlets/jsp

1. Session object
2. Cookies
3. Hidden form fields
4. url rewriting