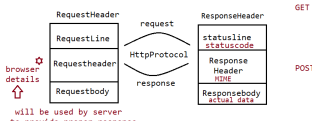
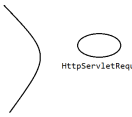


GET ====> 1. type url and hit enterkey
2. clicking on hyperlink
3. submiting form with method attribute as "get".
Note: data would be sent in RequestLine

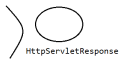
POST ====> 1.submitting form with method attribute as "post".
Note: data would be sent in RequestBody

Servlet
=======
1. public String getParameter()
2. public String[] getParameterValues()
3. public Enumeration<String> getHeaderNames()
4. public Object getHeader(String headerName)
5. public String getRemoteHost()
6. public String getRemoteAddr()
7. public String getRemotePort()
8. public String getServerName()
9. public String getServerPort()



Note: load-on-startup
a. value should be 0 and any positive number
b. if 2 servlets has same value then we can't predict the order
3. less the load-on-startup for a value,that particular servlet loading,instantiation,initializiation will happen.

1. public void setContentType(String name)
2. public PrintWriter getWriter()
3. public ServletOutputStream getOutputStream()



Note: Writing images,pdf's,..... as a response

```
response.setContentType("image/jpg");
   ;;;;
   ;;;;
ServletOutputStream os = response.getOutputStream();
```

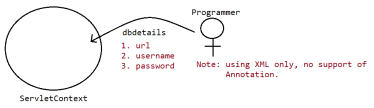
location of image file
==========
D:\servletpgms\.metadata\.plugins\org.eclipse.wst.server.core\tmp0\wtpwebapps\ImageResponseApp\nitin.JPG
smooth deployment
ProjectName
Image

Note:

```
PrintWriter out = response.getWriter();
ServletOutputStream os = response.getOutputStream();
```

In Servlet Coding we can't open 2 streams for response, if we try to do it would result in
java.lang.IllegalStateException

ServletContext(I)
==========
=> When we do deployment(manual), server will scan "webapps" folder and identifies the projects which is deployed.
=> All the identified Projects will be kept in Meta-space of server.
=> For Every Project which is deployed in Meta-Space automatically "ServletContext" object will be created.



web.xml

```
<web-app>
   <context-param>
        <param-name>jdbcURL</param-name>
        <param-value>
              jdbc:mysql://localhost:3306/enterprisejavabatch
        </param-value>
   </context-param>

   <context-param>
        <param-name>user</param-name>
        <param-value>root</param-value>
   </context-param>

   <context-param>
        <param-name>password</param-name>
        <param-value>root123</param-value>
   </context-param>

</web-app>
```

context object can be accessed in the entire project, meaning in all the servlets of the project

1. public Enumeration getInitParameterNames()
2. public String getInitParameter(String name)
3. public Enumeration getAttributeNames()
4. public String getAttribute(String name)
5. public void setAttribute (String k,Object v)

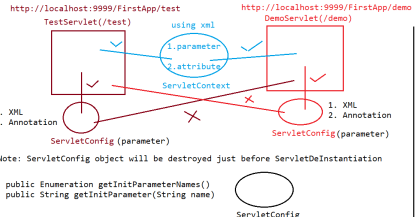
Note: In ServletContext Object, we can add parameter data as well as attribute data
from xml
during execution

Note: Context Object will be destroyed only when we undeploy the project.

ServletConfig
==========
=> This object is used to store the configuration details of a particular servlet like logical name of the servlet,initialization parameters, and so on...
=> Using ServletConfig we will get to know the complete view of a particular servlet.
=> Loading ===> static block
instantiation ===> public Zero argument constructor
ServletConfig
initialization ====> public void init(ServletConfig config) throws SE

RequestProcessing==> public void doXXXX(HSR request, HSR response) throws SE,IOE
DeInstantiation ==> public void destroy()

=> ServletConfig object is specific to a particular Servlet.



Note: ServletConfig object will be destroyed just before ServletDeInstantiation

1. public Enumeration getInitParameterNames()
2. public String getInitParameter(String name)

ServletConfig

XML
===

```
<web-app>
   <context-param>
        <param-name></param-name>
        <param-value></param-value>
   </context-param>
        ;;;
        ;;;
   <servlet>
        <servlet-name></servlet-name>
        <servlet-class></serlvet-class>
        <init-param>
             <param-name></param-name>
             <parma-value></param-value>
        </init-param>
        ;;;
        ;;;
   </servlet>

</web-app>
```

ServletContext
ServletConfig

```
<servlet>
        <servlet-name>TestServlet</servlet-name>
        <servlet-class>in. n.controller.TestServlet</servlet-class>
        <init-param>
            <param-name>jdbcurl</param-name>
            <param-value>jdbc:mysql:///enterprisejavabatch</param-value>
        </init-param>
        <init-param>
            <param-name>user</param-name>
            <param-value>root</param-value>
        </init-param>
        <init-param>
            <param-name>password</param-name>
            <param-value>root123</param-value>
        </init-param>
</servlet>
```

ServletConfig
jdbcurl
user
password

```
<servlet>
        <servlet-name>DemoServlet</servlet-name>
        <servlet-class>in.       .controller.DemoServlet</servlet-class>
        <init-param>
             <param-name>Company</param-name>
             <param-value>;       </param-value>
        </init-param>
        <init-param>
             <param-name>CourseName</param-name>
             <param-value>EnterpriseJava</param-value>
        </init-param>
        <init-param>
             <param-name>Mentor</param-name>
             <param-value:`  `  `  `y</param-value>
        </init-param>
</servlet>
```

ServletConfig
Company
CourseName
Mentor

```
@WebServlet
(
    urlPatterns = { "/test" },
    initParams = {
       @WebInitParam(name = "jdbcUrl", value = "jdbc:mysql:///enterprisejavabatch"),
       @WebInitParam(name = "user", value = "root"),
       @WebInitParam(name = "password", value = "root123")
    }

)
```

```
@WebServlet
(
    urlPatterns = { "/demo" },
    initParams = {
            @WebInitParam(name = "Company", value = "       "),
            @WebInitParam(name = "course", value = "Enterprisejava"),
            @WebInitParam(name = "mentor", value = "N+vinReddy")
    }

)
```