Introduction to ServletConfig interface

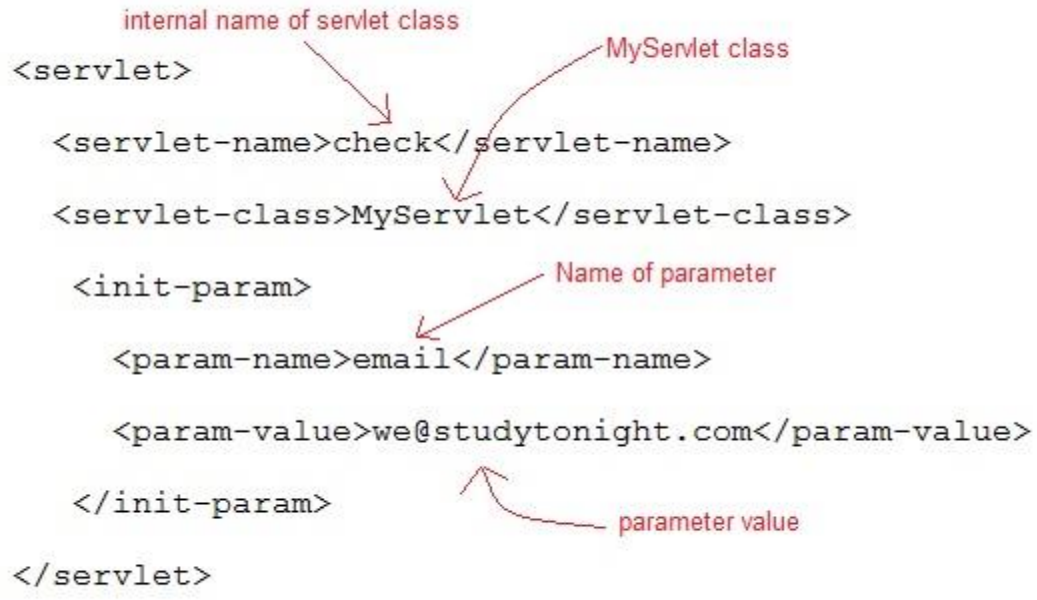
When the **Web Container** initializes a servlet, it creates a **ServletConfig** object for the servlet. ServletConfig object is used to pass information to a servlet during initialization by getting configuration information from**web.xml**(Deployment Descriptor).

Methods of ServletConfig

- String `getInitParameter(String name):` returns a String value initialized parameter, or NULL if the parameter does not exist.
- Enumeration `getInitParameterNames():` returns the names of the servlet's initialization parameters as an Enumeration of String objects, or an empty Enumeration if the servlet has no initialization parameters.
- ServletContext `getServletContext():` returns a reference to the ServletContext
- String `getServletName():` returns the name of the servlet instance

How to Initialize a Servlet inside web.xml

In the Deployment Descriptor(web.xml) file,




```
<servlet>
  <servlet-name>check</servlet-name>
  <servlet-class>MyServlet</servlet-class>
   <init-param>
     <param-name>email</param-name>
     <param-value>we@studytonight.com</param-value>
   </init-param>
</servlet>
```

Or, Inside the Servlet class, using following code,

```
ServletConfig sc = getServletConfig();
out.println(sc.getInitParameter("email"));
```

Example demonstrating usage of ServletConfig

web.xml

```
<web-app...>
   <servlet>
      <servlet-name>check</servlet-name>
      <servlet-class>MyServlet</servlet-class>
      <init-param>
            <param-name>email</param-name>
            <param-value>we@studytonight.com</param-value>
      </init-param>
    </servlet>
```

```
    <servlet-mapping>
        <servlet-name>check</servlet-name>
        <url-pattern>/check</url-pattern>
    </servlet-mapping>
    <welcome-file-list>
        <welcome-file>index.html</welcome-file>
     </welcome-file-list>
</web-app>
```

MyServlet class :

```
import java.io.*;
import javax.servlet.*;
import javax.servlet.http.*;

public class MyServlet extends HttpServlet {

  protected void doGet(HttpServletRequest request, HttpServletResponse response)
         throws ServletException, IOException {

        response.setContentType("text/html;charset=UTF-8");
        PrintWriter out = response.getWriter();
        ServletConfig sc=getServletConfig();
        out.println(sc.getInitParameter("email"));
    }
}
```

The `@WebServlet` annotation is used to declare a servlet.

Name	Type	Required	Description
value or **urlPatterns**	*String[]*	Required	Specify one or more URL patterns of the servlet. Either of attribute can be used, but not both.
name	*String*	Optional	Name of the servlet
displayName	*String*	Optional	Display name of the servlet
description	*String*	Optional	Description of the servlet
asyncSupported	*boolean*	Optional	Specify whether the servlet supports asynchronous operation mode. Default is false.
initParams	*WebInitParam[]*	Optional	Specify one or more initialization parameters of the servlet. Each parameter is specified by `@WebInitParam` annotation type.
loadOnStartup	*int*	Optional	Specify load-on-startup order of the servlet.
smallIcon	*String*	Optional	Specify name of the small icon of the servlet.
largeIcon	*String*	Optional	Specify name of the large icon of the servlet.