

# Üst Düzey Programlama

## Servlet

# Servlet Nedir?

Web sayfaları ilk başlarda durağan bir yapıya sahipti ve kullanıcıdan bilgi alarak işlemler yapmıyordu. Zamanın geçmesiyle kullanıcıya göre değişen ya da işlemler yapabilen web sayfalarının geliştirilmesi gerekli olmuştur.

Web sayfalarında kullanıcılardan veri alarak bu verilerin işlenmesi ve kullanıcıya işlemlerin sonuçlarının gösterilmesi önemli olmuştur.

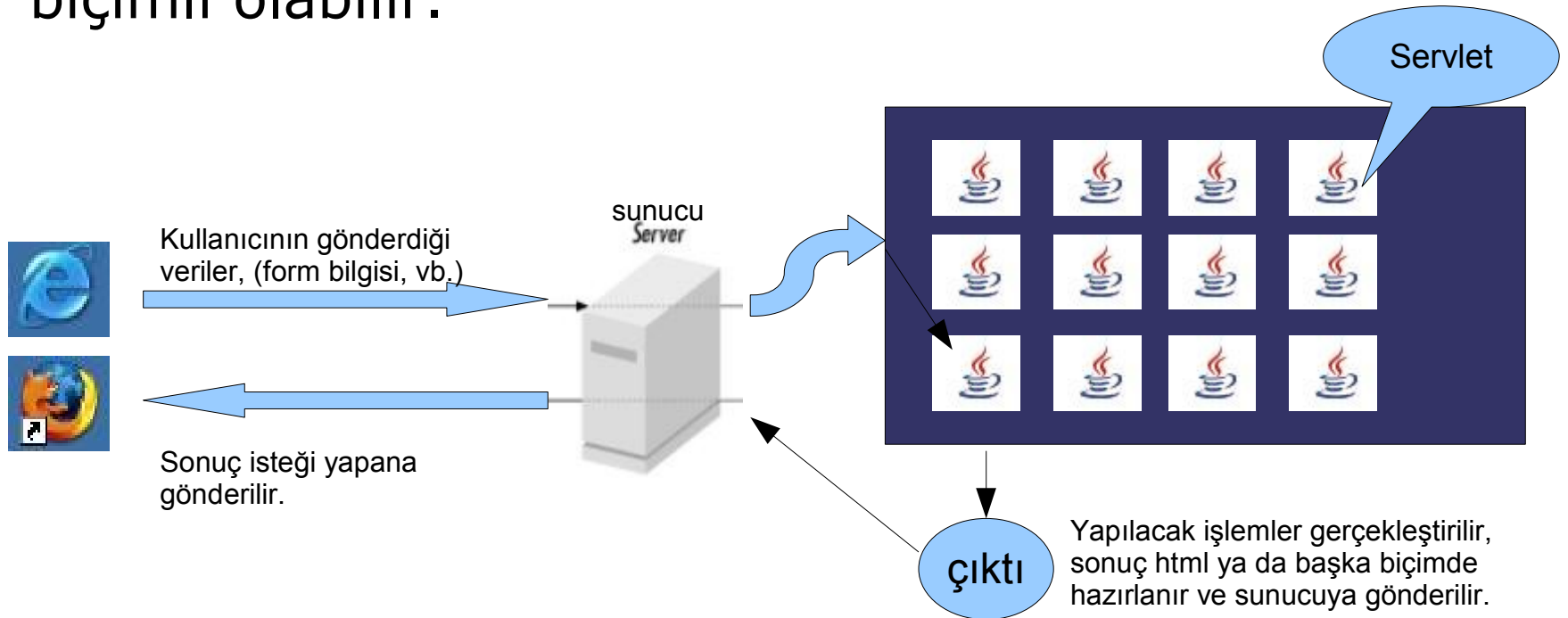
Java ile dinamik web uygulamaları geliştirmek için farklı teknolojiler bulunmaktadır. **Servlet** bu teknolojilerden biridir.

# Dinamik sayfa ?

1. Web sayfası kullanıcının gönderdiği veriler temelinde sonuç üretir. (Örn: Google gibi arama sayfaları)
2. Web sayfasında görüntülenen veriler sürekli güncellenmektedir. (Örn: haber siteleri, hava durumu gibi)
3. Web sayfası veri tabanı bilgisi gösteriyorsa, veri tabanı değiştiğinde web sayfası da güncellenmelidir.

# Servlet Nasıl Çalışır?

Servlet web sunucuları üzerinde çalışan, kullanıcıdan ya da farklı yerlerden aldığı verilere göre sonuç üreten Java sınıflarıdır. Oluşan sonuç HTML ya da başka biçimli olabilir.



# Servlet Nasıl Çalışır ?

1. Kullanıcı web tarayıcısından ulaşmak istediği sayfayı belirtir ve bu bilgiyi sunucuya gönderir ya da bir sayfada bulunan web form bilgilerini doldurur ve gönderme tuşuna tıklar.
2. Web sunucusu kullanıcı tarafından gelen bu http ya da başka biçimdeki isteği(request) alır. Bu isteğe uygun olan servlet'i belirler. Bu servlet'e ait bir nesne bellekte var mı diye kontrol eder. Bu kontrolde var ise istek ve tüm bilgi bu servlete gönderilir. Eğer istenilen servlet nesnesi bellekte yok ise yeni bir tane oluşturulur, istek ve bilgi daha sonra bu servlet'e gönderilir.

# Servlet Nasıl Çalışır ?

3. Servlet kendisine gelen istek ve verileri alır ve bunları kullanarak oluşturması gerekli olan sonucu oluşturur. Servlet'in oluşturacağı sonuç bilgisi farklı biçimlerde olabilir. Genel olarak oluşturulan sonuç HTML sayfası şeklindedir.
4. Servlet oluşturduğu sonucu web sunucusuna gönderir.
5. Web sunucusu servlet'ten gelen sonucu isteği yapan kullanıcıya gönderir.

# Servlet Nasıl Çalışır ?

Farklı sonuçlar üretmek için farklı sınıflar temelinde Servlet sınıfları üretmemiz gereklidir.

HTML isteklerini anlayarak HTML sonucu üreten ve yaygın olarak kullanılan temel Servlet sınıfı ***HttpServlet*** dir.

# Servlet Nasıl Çalışır ?

Bir web sunucusunun Java'nın ve servletlerin çalışmasını ve nasıl kullanacağını bilmesi için, web sunucusunda ***Servlet Makinesi (Servlet Engine)*** bulunmalıdır.

Bu makine servletleri JVM ile çalıştırabilir, oluşan sonuçları anlayarak kullanıcıya geri döndürebilir.

Web sunucusu ve servlet makinesi olarak kullanılan farklı firmalara ait sunucular bulunmaktadır.

- Apache Tomcat*
- Macromedia Jrun*
- OC4J : Oracle for Java*
- Caucho Resin*
- Jetty*



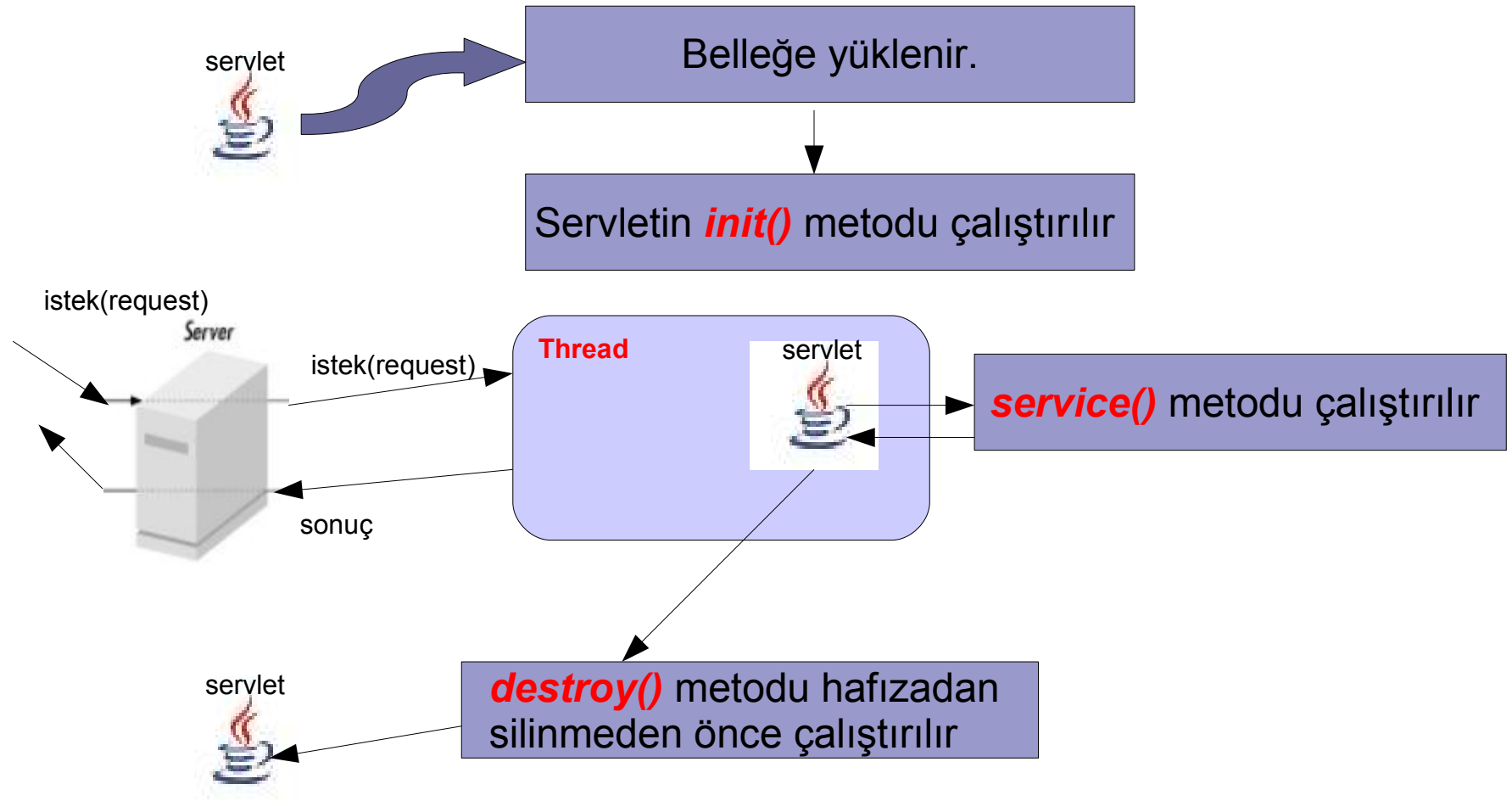
# Servlet Nasıl Çalışır ?

Web sunucusuna belirli bir servlet için istek geldiğinde web sunucusu bu servletin bellekte olup olmamasını kontrol eder. Eğer yok ise oluşturur.

Aynı servlet'e birden fazla istek aynı anda gelebilir. Sunucu bu servlet sınıfından tek bir nesne oluşturur. Bu nesneyi çok iş parçacıklı (multi threaded) şekilde tüm istekler için ortak kullanır.

Yazılan servlet kodu çoklu işparçacıklı(multi threaded) yapısına uygun olarak güvenli bir şekilde yazılmalıdır.

# Servlet in hayat döngüsü



# HttpServlet

HTTP iletişim protokolünde birden fazla istek bulunmaktadır. Bu istekler şunlar olabilir:

*GET, POST, HEAD, PUT, OPTIONS, DELETE, TRACE ...*

Servlet'in service() metodu gelen isteğin tipine bakar ve bu isteğe uygun gelen kendisindeki belirli metotları çalıştırır.

HttpServlet içerisinde yukarıda belirtilen isteklere karşılık gelen ve bu istek geldiğinde çalıştırılan metot isimleri **doXXX()** şeklindedirler.

# HttpServlet

***-doGet() : gelen istek GET isteğidir***

***-doPost() : gelen istek POST isteğidir.***

***-doDelete()***

***...***

# HttpServlet

HttpServlet'in `init()` metodu servlet ilk çalıştığında yapılacak genel işlemler için kullanılır. Örneğin, uygulamanızda veri tabanı işlemleri kullanıyorsanız. Veri tabanı bağlantısını bir kez burada oluşturabilir ve gelen istekler için aynı bağlantıyı kullanabilirsiniz.

# HttpServlet

## SingleThreadModel

Normalde sunucu servletten bir örnek oluşturur ve gelen her istek için bir thread oluşturur. Eğer önceki istek bitmemiş ise oluşan threadler eş zamanlı olarak (concurrent) aynı servlet'i kullanırlar. doGet ve doPost metotlarında servlet'te tanımlanan genel değişkenlere eş zamanlı erişim olacağı için bu değişkenler dikkatli kullanılmalıdır. Genel değişkenlere erişimi güvenli yapabilmek için her istek için tek bir thread kullanılmasını sağlayabiliriz.

# HttpServlet

## SingleThreadModel

Bunu sağlayabilmek için geliştirmiş olduğumuz servletlerin **SingleThreadModel** isimli *interface* i gerçekleştirmesi gereklidir.

```
public class Servletim extends HttpServlet  
    implements SingleThreadModel{  
  
}
```

Bu yöntemin dezavantajı, isteklere yanıt gecikeceği için performans düşer. Kullanılması önerilmez.(Servlet 2.4 ve JSP 2.0 da *deprecated* yapılmıştır.)

# HttpServlet

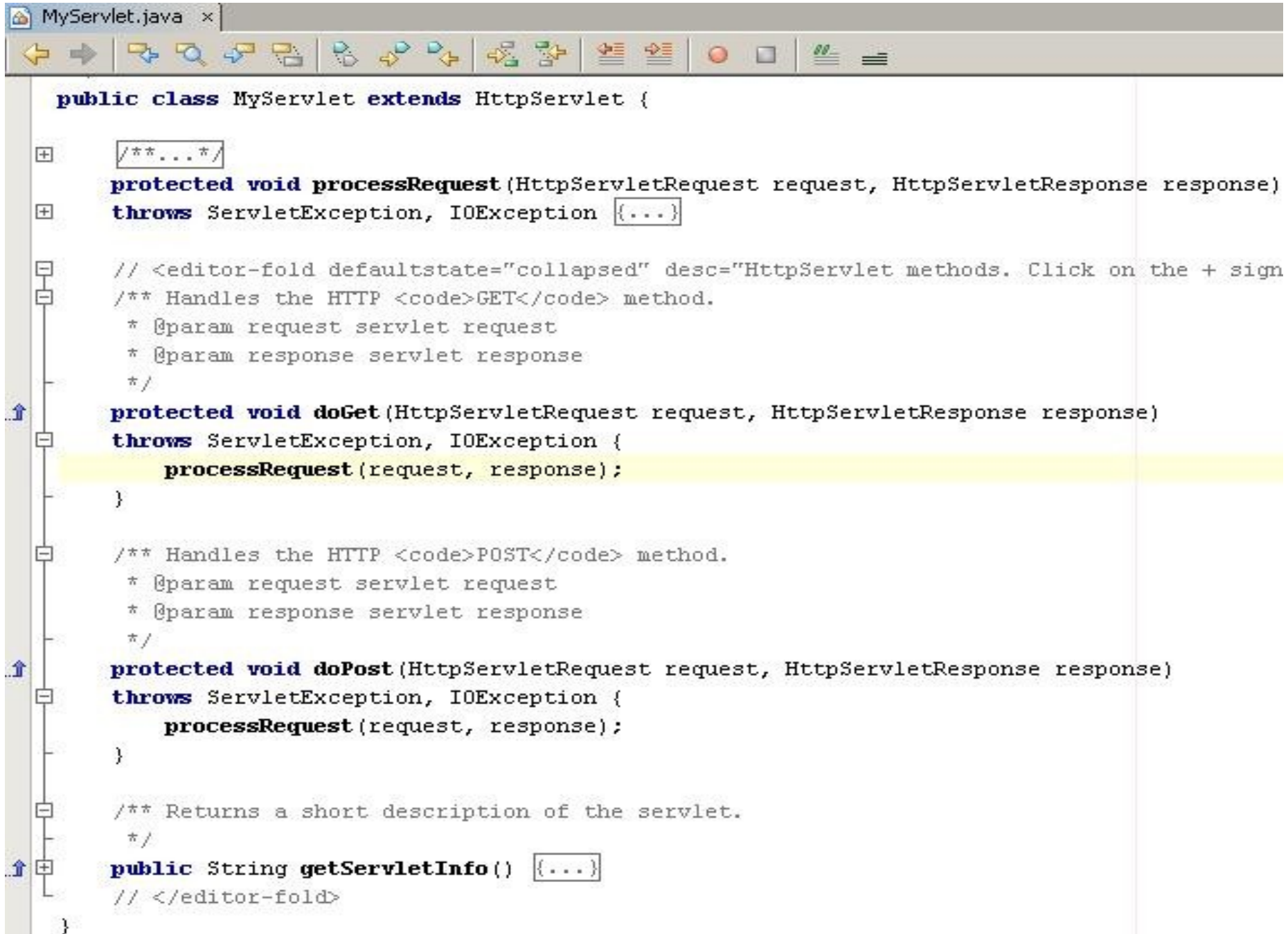
**HttpServlet** sınıfının **service()** metoduna ait iki adet parametre bulunmaktadır. Bu parametreler son derece önemlidir.

**HttpServletRequest:** Bu parametre istek ile ilgili olan tüm bilgileri tutar. Kullanıcıdan gelen tüm değişkenler veriler, form bilgileri bu değişkendedir.

**HttpServletResponse:** Servlet'in işini gerçekleştirdikten sonra oluşturacağı sonuç için bu parametre kullanılır. Bu parametre kullanıcıya isteğinin sonucu olarak gönderilir.



# Servlet'in oluşturulması



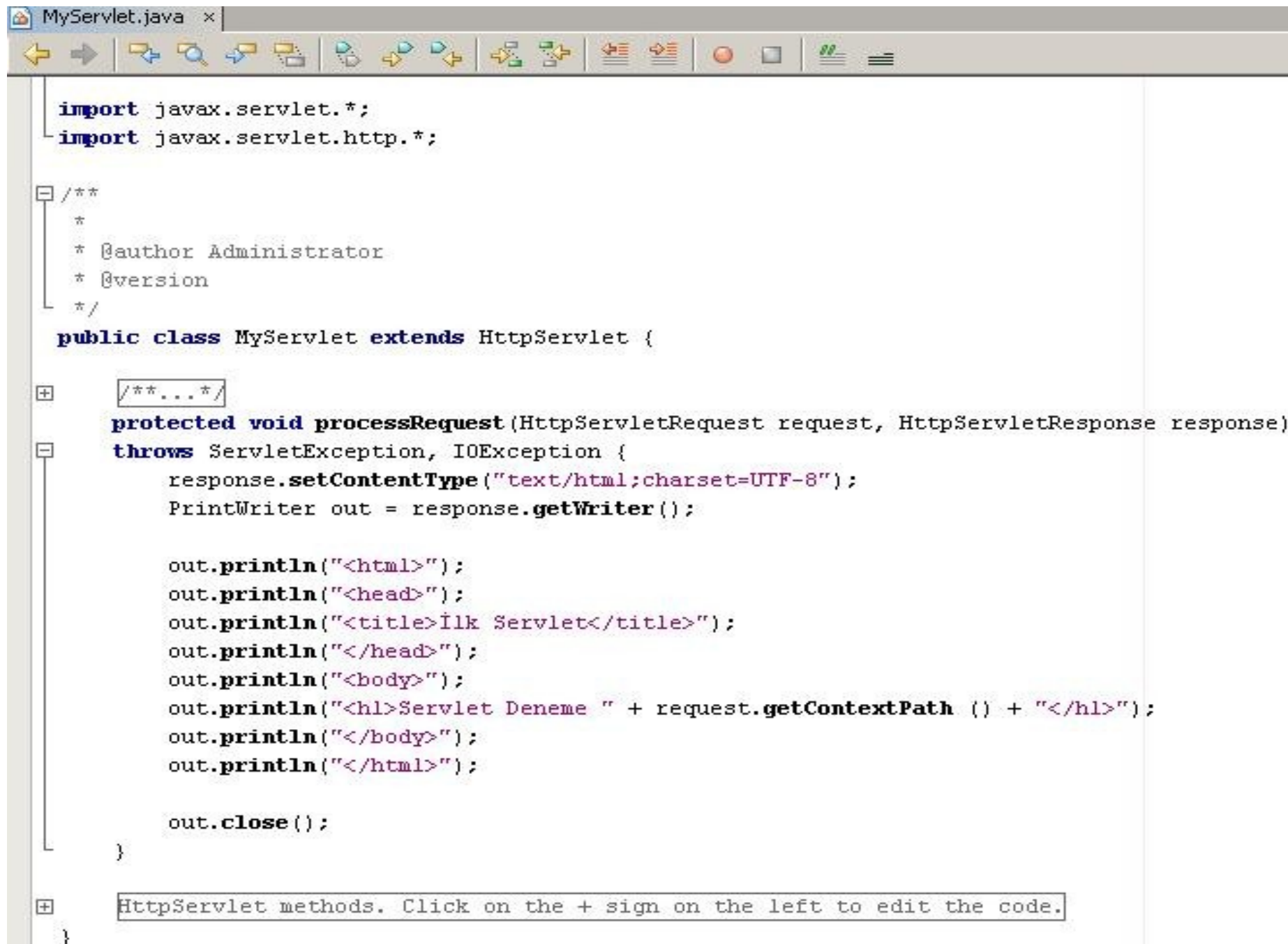
```
MyServlet.java x
public class MyServlet extends HttpServlet {

    /**...*/
    protected void processRequest(HttpServletRequest request, HttpServletResponse response)
        throws ServletException, IOException {...}

    // <editor-fold defaultstate="collapsed" desc="HttpServlet methods. Click on the + sign
    /** Handles the HTTP <code>GET</code> method.
     * @param request servlet request
     * @param response servlet response
     */
    protected void doGet(HttpServletRequest request, HttpServletResponse response)
        throws ServletException, IOException {
        processRequest(request, response);
    }

    /** Handles the HTTP <code>POST</code> method.
     * @param request servlet request
     * @param response servlet response
     */
    protected void doPost(HttpServletRequest request, HttpServletResponse response)
        throws ServletException, IOException {
        processRequest(request, response);
    }

    /** Returns a short description of the servlet.
     */
    public String getServletInfo() {...}
    // </editor-fold>
}
```



```
MyServlet.java x
import javax.servlet.*;
import javax.servlet.http.*;

/**
 *
 * @author Administrator
 * @version
 */
public class MyServlet extends HttpServlet {

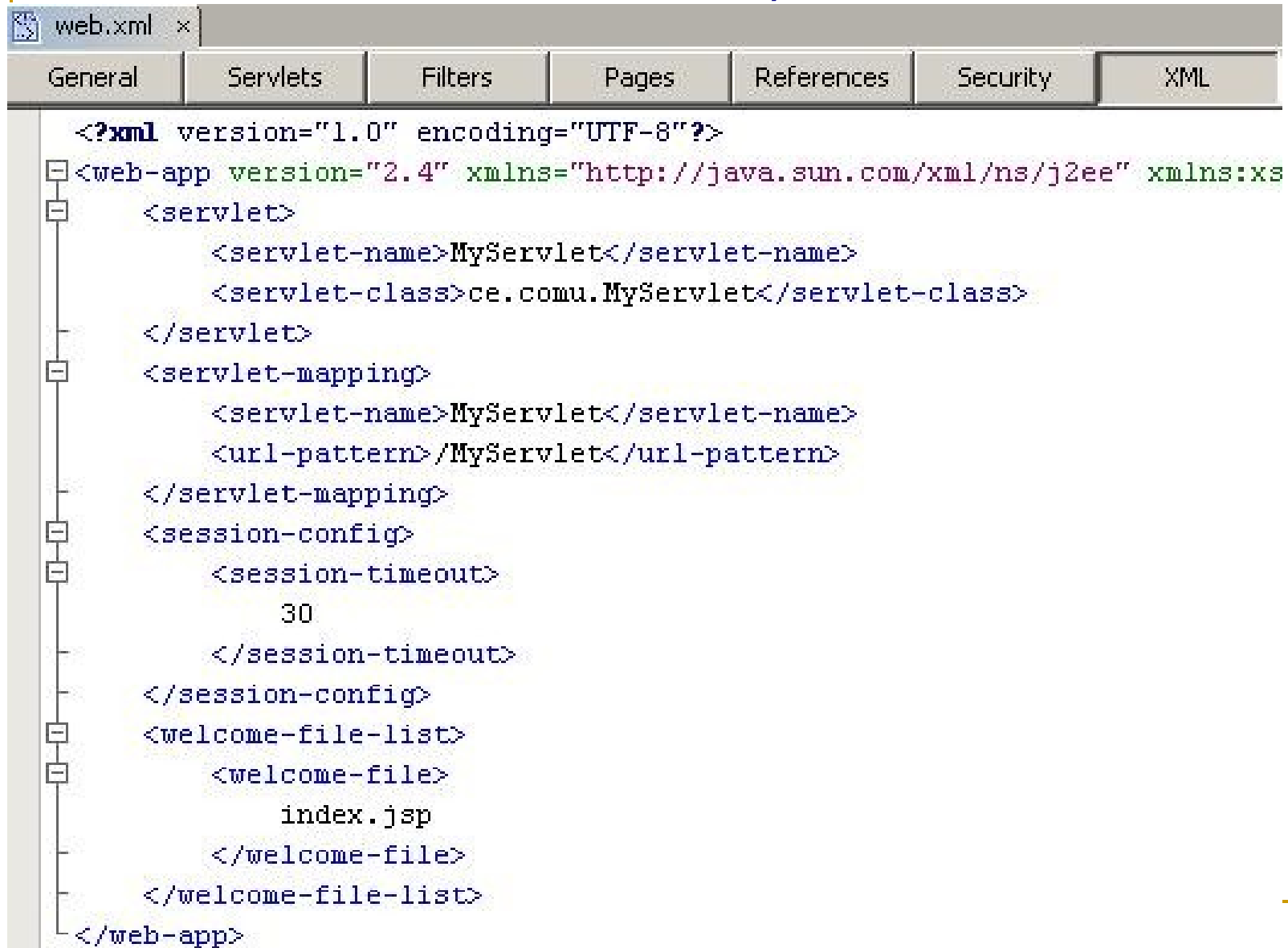
    /**...*/
    protected void processRequest(HttpServletRequest request, HttpServletResponse response)
        throws ServletException, IOException {
        response.setContentType("text/html;charset=UTF-8");
        PrintWriter out = response.getWriter();

        out.println("<html>");
        out.println("<head>");
        out.println("<title>İlk Servlet</title>");
        out.println("</head>");
        out.println("<body>");
        out.println("<h1>Servlet Deneme " + request.getContextPath () + "</h1>");
        out.println("</body>");
        out.println("</html>");

        out.close();
    }

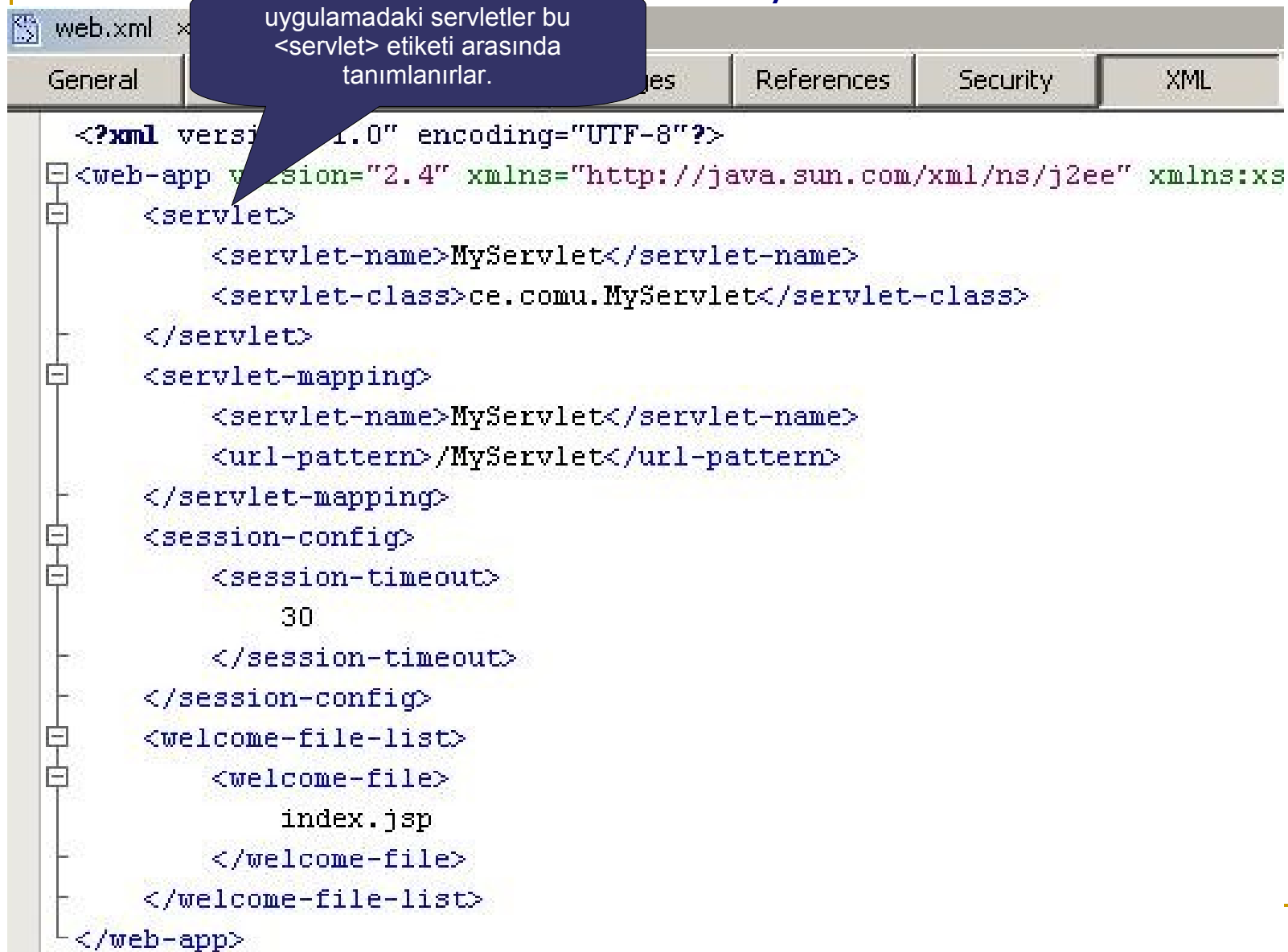
    HttpServlet methods. Click on the + sign on the left to edit the code.
}
```

# **web.xml** Sunucunun uygulama hakkında tüm bilgiyi aldığı XML dosyasıdır.

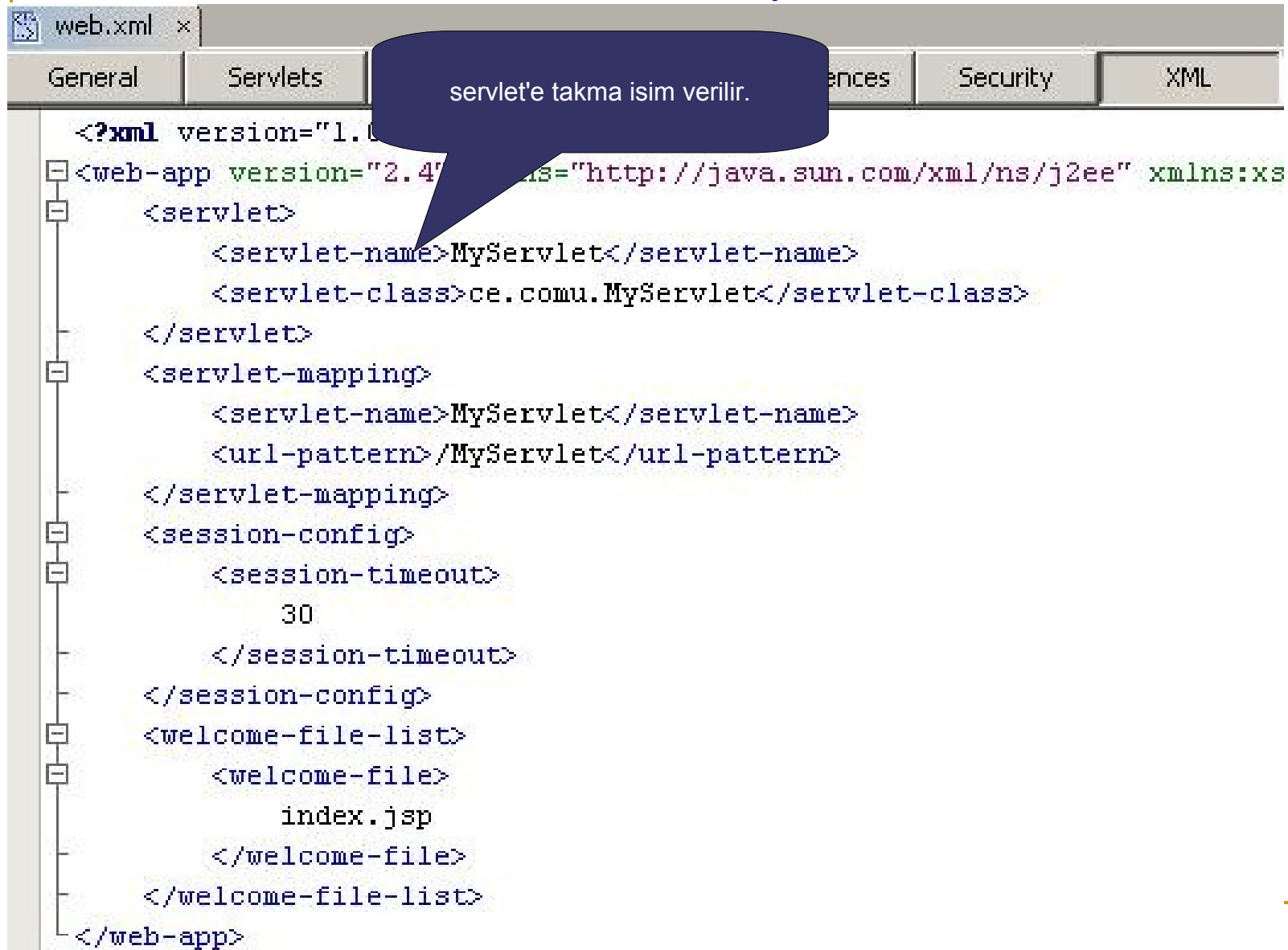


```
<?xml version="1.0" encoding="UTF-8"?>
<web-app version="2.4" xmlns="http://java.sun.com/xml/ns/j2ee" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <servlet>
    <servlet-name>MyServlet</servlet-name>
    <servlet-class>ce.comu.MyServlet</servlet-class>
  </servlet>
  <servlet-mapping>
    <servlet-name>MyServlet</servlet-name>
    <url-pattern>/MyServlet</url-pattern>
  </servlet-mapping>
  <session-config>
    <session-timeout>
      30
    </session-timeout>
  </session-config>
  <welcome-file-list>
    <welcome-file>
      index.jsp
    </welcome-file>
  </welcome-file-list>
</web-app>
```

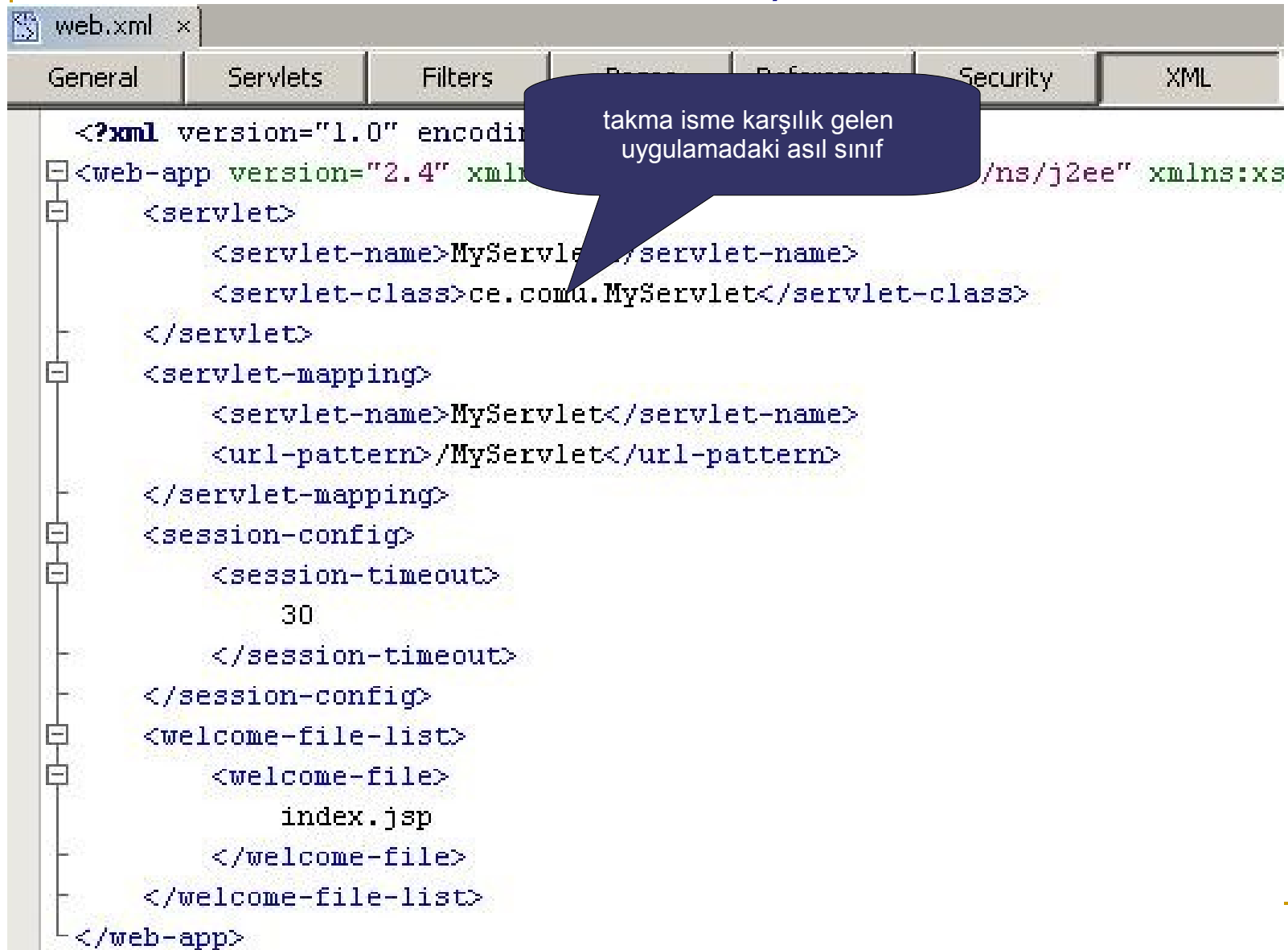
# **web.xml** Sunucunun uygulama hakkında tüm bilgiyi aldığı XML dosyasıdır.



# **web.xml** Sunucunun uygulama hakkında tüm bilgiyi aldığı XML dosyasıdır.



# **web.xml** Sunucunun uygulama hakkında tüm bilgiyi aldığı XML dosyasıdır.



```
<?xml version="1.0" encoding="UTF-8" standalone="no" ?>
<web-app version="2.4" xmlns="http://xmlns.j2ee.org/xmlns" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
    xsi:schemaLocation="http://xmlns.j2ee.org/xmlns http://java.sun.com/xml/ns/j2ee/web-app_2_4.xsd">
    <servlet>
        <servlet-name>MyServlet</servlet-name>
        <servlet-class>ce.comu.MyServlet</servlet-class>
    </servlet>
    <servlet-mapping>
        <servlet-name>MyServlet</servlet-name>
        <url-pattern>/MyServlet</url-pattern>
    </servlet-mapping>
    <session-config>
        <session-timeout>
            30
        </session-timeout>
    </session-config>
    <welcome-file-list>
        <welcome-file>
            index.jsp
        </welcome-file>
    </welcome-file-list>
</web-app>
```

# **web.xml** Sunucunun uygulama hakkında tüm bilgiyi aldığı XML dosyasıdır.





# **web.xml** Sunucunun uygulama hakkında tüm bilgiyi aldığı XML dosyasıdır.



The screenshot shows an IDE window titled 'web.xml' with tabs for General, Servlets, Filters, Pages, References, Security, and XML. The XML content is as follows:

```
<?xml version="1.0" encoding="UTF-8"?>
<web-app version="2.4" xmlns="http://xmlns.j2ee.org/xmlns" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <servlet>
    <servlet-name>MyServlet</servlet-name>
    <servlet-class>ce.c</servlet-class>
  </servlet>
  <servlet-mapping>
    <servlet-name>MyServlet</servlet-name>
    <url-pattern>/MyServlet</url-pattern>
  </servlet-mapping>
  <session-config>
    <session-timeout>
      30
    </session-timeout>
  </session-config>
  <welcome-file-list>
    <welcome-file>
      index.jsp
    </welcome-file>
  </welcome-file-list>
</web-app>
```

A callout box points to the `<url-pattern>` element in the `<servlet-mapping>` section, containing the text: "girilen <url-pattern> adresine karşılık verilen takma isme sahip servlet çalıştırılır."

# İki Servlet Geliştirelim

GetServlet : GET İsteklerine cevap versin

PostServlet: POST isteklerine cevap versin.

```
package ce.ustduzey.ders01;
```

```
+ import ...
```

```
public class GetServlet extends HttpServlet {
```

```
    protected void processRequest(HttpServletRequest request, HttpServletResponse response)
```

```
+    throws ServletException, IOException {...}
```

```
    protected void doGet(HttpServletRequest request, HttpServletResponse response)
```

```
-    throws ServletException, IOException {
```

```
        response.setContentType("text/html;charset=UTF-8");
```

```
        PrintWriter out = response.getWriter();
```

```
        out.println("<html>");
```

```
        out.println("<head>");
```

```
        out.println("<title>GetServlet</title>");
```

```
        out.println("</head>");
```

```
        out.println("<body>");
```

```
        out.println("<h1>Get Servlet Örneği</h1>");
```

```
        out.println("</body>");
```

```
        out.println("</html>");
```

```
        out.close();
```

```
    }
```

```
package ce.ustduzey.ders01;
```

```
+ import ...
```

```
public class PostServlet extends HttpServlet {
```

```
    protected void processRequest(HttpServletRequest request, HttpServletResponse response)
```

```
+    throws ServletException, IOException {...}
```

```
↑    protected void doGet(HttpServletRequest request, HttpServletResponse response)
```

```
+    throws ServletException, IOException {...}
```

```
↑    protected void doPost(HttpServletRequest request, HttpServletResponse response)
```

```
-    throws ServletException, IOException {
```

```
        response.setContentType("text/html;charset=UTF-8");
```

```
        PrintWriter out = response.getWriter();
```

```
        out.println("<html>");
```

```
        out.println("<head>");
```

```
        out.println("<title>PostServlet</title>");
```

```
        out.println("</head>");
```

```
        out.println("<body>");
```

```
        out.println("<h1>Post Servlet Örneği</h1>");
```

```
        out.println("</body>");
```

```
        out.println("</html>");
```

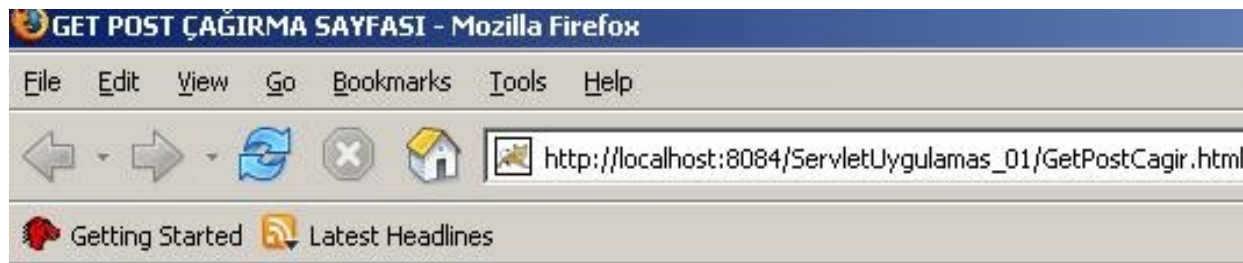
```
        out.close();
```

```
    }
```

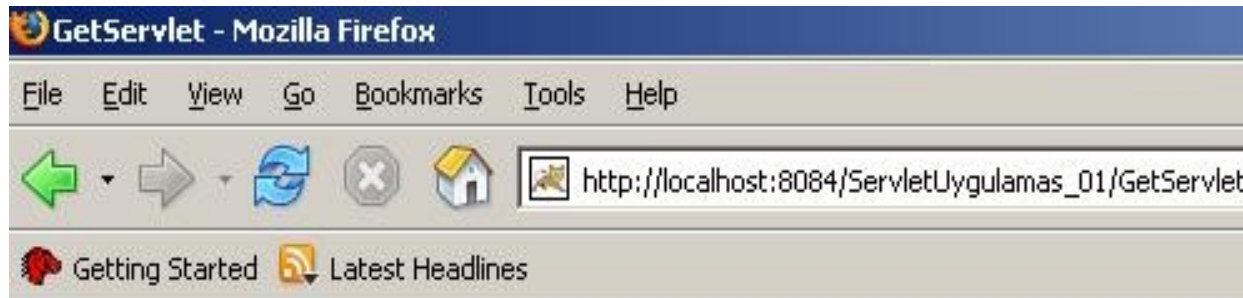


```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN">
```

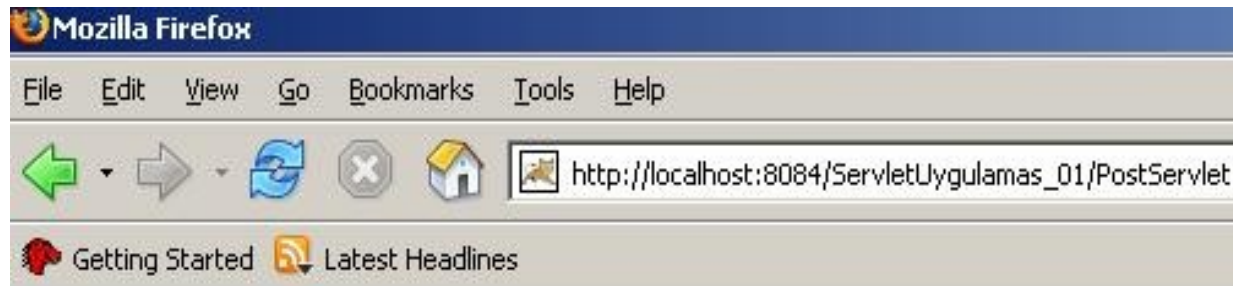
```
<html>
<head>
  <meta http-equiv="Content-Type" content="text/html; charset=iso-8859-9">
  <title>GET POST ÇAĞIRMA SAYFASI</title>
</head>
<body>
  <ul>
    <li><a href="GetServlet">GET SERVLET</a></li>
    <li><a href="PostServlet">POST SERVLET</a></li>
  </ul>
</body>
</html>
```



- [GET SERVLET](#)
- [POST SERVLET](#)



## Get Servlet Örneği



```
<?xml version="1.0" encoding="UTF-8"?>
<web-app version="2.4" xmlns="http://java.sun.com/xml/ns/j2ee" xmlns:
  <servlet>
    <servlet-name>Servlet01</servlet-name>
    <servlet-class>ce.ustduzey.ders01.MyServlet</servlet-class>
  </servlet>
  <servlet>
    <servlet-name>GetServlet</servlet-name>
    <servlet-class>ce.ustduzey.ders01.GetServlet</servlet-class>
  </servlet>
  <servlet>
    <servlet-name>PostServlet</servlet-name>
    <servlet-class>ce.ustduzey.ders01.PostServlet</servlet-class>
  </servlet>
  <servlet-mapping>
    <servlet-name>Servlet01</servlet-name>
    <url-pattern>/Servlet01</url-pattern>
  </servlet-mapping>
  <servlet-mapping>
    <servlet-name>GetServlet</servlet-name>
    <url-pattern>/GetServlet</url-pattern>
  </servlet-mapping>
  <servlet-mapping>
    <servlet-name>PostServlet</servlet-name>
    <url-pattern>/PostServlet</url-pattern>
  </servlet-mapping>
```

# Üst Düzey Programlama

## Servlet