

CGI

Common Gateway Interface

- **Prinzip:**

Ein Programm außerhalb des Webservers stellt einen Zugang (Gateway) zu geschützten, für den Webserver nicht erreichbaren Daten bereit.

[WT:IV-18 Server Technologies]

- "Standard für den Datenaustausch zwischen einem Webserver und dritter Software" [[Wikipedia](#)]
- Ermöglicht dynamische Generierung von Webseiten
- Direkt durch das Betriebssystem ausgeführt
 - Vorteil: Programm kann separat gespeichert werden → entlastet Server
 - Nachteil: zeitaufwändiger Start eines Prozesses

□ Prinzip:

Ein Programm außerhalb des Webservers stellt einen Zugang (Gateway) zu geschützten, für den Webserver nicht erreichbaren Daten bereit.

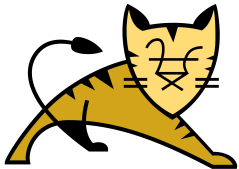
[WT:IV-18 Server Technologies]

- "Standard für den Datenaustausch zwischen einem **Webserver** und dritter Software" [Wikipedia]
- Ermöglicht dynamische Generierung von Webseiten
- Direkt durch das Betriebssystem ausgeführt
 - Vorteil: Programm kann separat gespeichert werden → entlastet Server
 - Nachteil: zeitaufwändiger Start eines Prozesses

Webserver

Allgemein

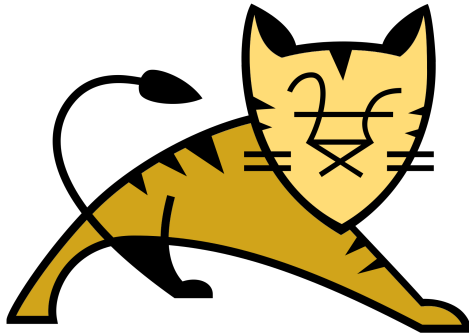
- ❑ "... computer system that processes requests via HTTP ..."
- ❑ Bezeichnung für Computer mit der entsprechenden Software oder die Webserver-Software selbst
- ❑ Beispiele für verbreitete Software:
 - Jetty
 - nginx
 - **Tomcat**



Webserver

Tomcat

- "... is an open source implementation of the **Java Servlet, JavaServer Pages (JSP), Java WebSocket** and Java Expression Language technologies."



- ... besteht aus:
 - *Catalina* → Servlet-Container
 - *Jasper* → JSP-Engine
 - *Coyote* → Connector-Framework

Webserver

Tomcat Installation

1. Download von offizieller Webseite → tomcat.apache.org
und Datei in gewünschtes Verzeichnis entpacken

Webserver

Tomcat Installation

1. Download von offizieller Webseite → tomcat.apache.org und Datei in gewünschtes Verzeichnis entpacken
2. **UNIX** (Terminal)
 - ❑ In Verzeichnis wechseln: `cd apache-tomcat-*\bin`
 - ❑ Falls nötig, fehlende Ausführungsrechte setzen:
`chmod +x startup.sh shutdown.sh catalina.sh`
 - ❑ Anwendung starten: `./startup.sh`

Windows (Eingabeaufforderung)

- ❑ In Verzeichnis wechseln: `cd apache-tomcat-*\bin`
- ❑ Anwendung starten: `startup.bat`

Webserver

Tomcat Installation

1. Download von offizieller Webseite → tomcat.apache.org und Datei in gewünschtes Verzeichnis entpacken
2. **UNIX** (Terminal)
 - ❑ In Verzeichnis wechseln: `cd apache-tomcat-*\bin`
 - ❑ Falls nötig, fehlende Ausführungsrechte setzen:
`chmod +x startup.sh shutdown.sh catalina.sh`
 - ❑ Anwendung starten: `./startup.sh`

Windows (Eingabeaufforderung)

- ❑ In Verzeichnis wechseln: `cd apache-tomcat-*\bin`
 - ❑ Anwendung starten: `startup.bat`
-
- ❑ Für andere Betriebssysteme siehe [Tomcat Wiki](#)
 - ❑ Wenn Installation und Start erfolgreich → <http://localhost:8080>

CGI

CGI mit Tomcat

- ❑ Tomcat 9.0.8 Dokumentation:

[...] CGI support is implemented using the servlet class

`org.apache.catalina.servlets.CGIServlet`.

Traditionally, this servlet is mapped to the URL pattern `"/cgi-bin/*"`.

By default CGI support is disabled in Tomcat. [...]

CGI

CGI mit Tomcat

❑ Tomcat 9.0.8 Dokumentation:

[...] CGI support is implemented using the servlet class

`org.apache.catalina.servlets.CGIServlet`.

Traditionally, this servlet is mapped to the URL pattern **`"/cgi-bin/*"`**.

By default CGI support is disabled in Tomcat. [...]

❑ CGI-Support aktivieren:

1. Beispiel-Eintrag für `servlet-` und `servlet-mapping`-Elemente von `/conf/web.xml` in `WEB-INF/web.xml` unserer Anwendung kopieren und entsprechend anpassen.

2. Datei `/conf/context.xml` in `META-INF/` Ordner unserer Anwendung kopieren und für `Context` `privileged="true"` setzen:

```
<Context privileged="true">...</Context>
```

CGI

CGI mit Tomcat

❑ Tomcat 9.0.8 Dokumentation:

[...] CGI support is implemented using the servlet class

`org.apache.catalina.servlets.CGIServlet`.

Traditionally, this servlet is mapped to the URL pattern `"/cgi-bin/*"`.

By default CGI support is disabled in Tomcat. [...]

❑ CGI-Support aktivieren:

1. Beispiel-Eintrag für `servlet-` und `servlet-mapping-`Elemente von `/conf/web.xml` in `WEB-INF/web.xml` unserer Anwendung kopieren und entsprechend anpassen.

2. Datei `/conf/context.xml` in `META-INF/` Ordner unserer Anwendung kopieren und für `Context privileged="true"` setzen:

```
<Context privileged="true">...</Context>
```

oder

In `/conf/context.xml` für `Context privileged="true"` setzen.

→ Aktiviert CGI-Support für **alle** Anwendungen im `webapps` Ordner!

CGI mit Tomcat

Beispiel (UNIX)

1. Archiv `cgi-example.war` von der Kursseite herunterladen
2. Dieses in `webapps`-Ordner von Tomcat ablegen
3. Tomcat starten (falls noch nicht getan)
→ Tomcat wird das Archiv automatisch entpacken.
4. Ausführungsrechte für das Shell-Script setzen
in: `cgi-example/WEB-INF/cgi`
mit: `chmod +x cgi-sample1.sh`
5. Im Browser öffnen: <http://localhost:8080/cgi-example/>
und auf den [Link](#) klicken

Werte einiger CGI-Variablen

Installierte Server-Software: TOMCAT

Aufrufender Web-Browser: Mozilla/5.0 (X11; Ubuntu; Linux x86_64; rv:61.0) Gecko/20100101 Firefox/61.0

Anfragemethode: GET

Query-String: test=23

CGI mit Tomcat

Beispiel (UNIX): web.xml

```
...  
  
<!-- Servlet definitions ->  
  
<servlet>  
  <servlet-name>cgi</servlet-name>  
  <servlet-class>org.apache.catalina.servlets.CGIServlet</servlet-class>  
  <init-param>  
    <param-name>cgiPathPrefix</param-name>  
    <!-- Name of the folder that holds all the CGI scripts ->  
    <param-value>WEB-INF/cgi</param-value>  
  </init-param>  
  <load-on-startup>5</load-on-startup>  
</servlet>  
  
...  
  
<!-- How to call the CGI by URL ->  
  
<servlet-mapping>  
  <servlet-name>cgi</servlet-name>  
  <url-pattern>/cgi-bin/*</url-pattern>  
</servlet-mapping>  
  
...
```

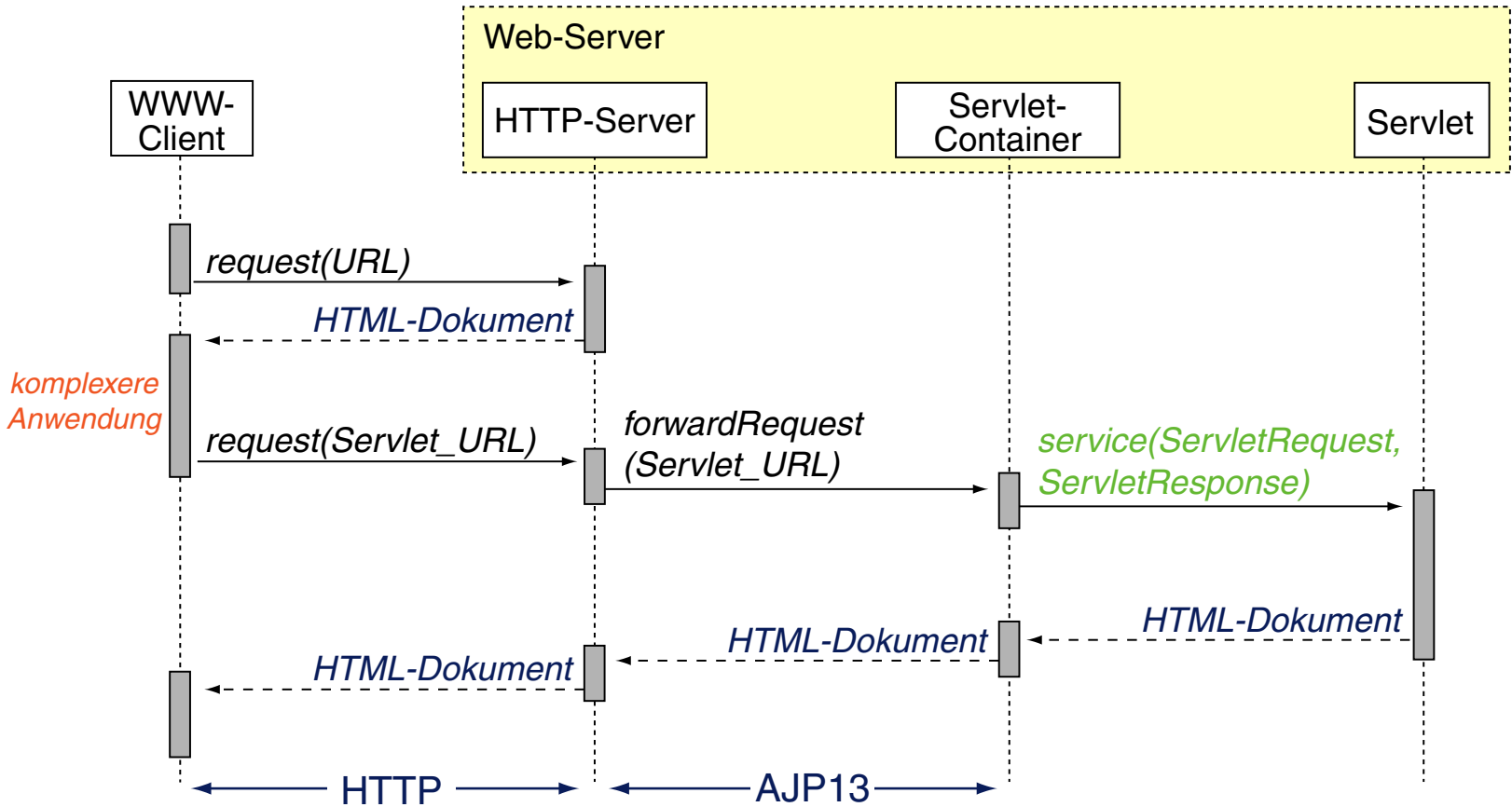
Servlets

Allgemein

- ❑ Weiterentwicklung von CGI
- ❑ Java-**Klassen**, deren Instanzen (innerhalb eines Webservers) Client-Anfragen entgegennehmen und beantworten
- ❑ Ermöglichen u.a. dynamische Generierung von Webseiten
- ❑ Werden in einem **Servlet-Container** verwaltet
(kontrolliert Lebenszyklus → instanziiert und löscht Servlets)
- ❑ Sind über Betriebssysteme und Webserver hinweg portabel
- ❑ Servlet-Instanz (Java-Objekt) verbleibt im Speicher des Webservers
→ nur einmal geladen, effizient

Servlets

Ablauf einer Servlet-Interaktion



Servlets mit Tomcat

Deployment Beispiel

Laden Sie die `servlet-example.war` von der Kursseite herunter und legen Sie dieses in den `webapps`-Ordner des entpackten Tomcat Archivs.

Stellen Sie sicher, dass Tomcat gestartet ist.

Tomcat entpackt das Archiv (`war = web application archive`) dann automatisch:

```
webapps/servlet-example/WEB-INF/web.xml
```

```
webapps/servlet-example/WEB-INF/classes/servlet/ServletHelloWorld.class
```

Öffnen Sie nun <http://localhost:8080/servlet-example/ServletHelloWorld> im Browser und Sie sollten folgendes sehen:

Hello World!

Servlets mit Tomcat

Deployment Beispiel

```
package servlet;

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

public class ServletHelloWorld extends HttpServlet {

    public void init() throws ServletException {
        // Nothing to do here.
    }

    public void doGet(HttpServletRequest request, HttpServletResponse response)
        throws ServletException, IOException {
        // Use "response" to specify the HTTP response line and headers
        // (e.g. specifying the content type, setting cookies).
        response.setContentType("text/html");
        PrintWriter out = response.getWriter();
        // Use "out" to send content to browser.
        out.print("<!DOCTYPE html>"
            + "<html><head><title>Hello World</title></head>"
            + "<body><h1>Hello World!</h1></body></html>");
    }
}
```


Servlets mit Tomcat

Deployment Beispiel

http	Anfrage per HTTP ...
localhost	... an diesen Rechner ...
:8080	... an Prozess unter Port 8080 (Tomcat).
<code>servlet-example</code>	Context-Path der anzufragenden Web-Anwendung
<code>/ServletHelloWorld</code>	Anfrage an die Web-Anwendung

→ Konfiguration Web-Anwendung in `webapps/servlet-example/WEB-INF/web.xml` :

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<web-app ...>
  <servlet>
    <servlet-name>HelloWorld</servlet-name>
    <servlet-class>servlet.ServletHelloWorld</servlet-class>
  </servlet>
  <servlet-mapping>
    <servlet-name>HelloWorld</servlet-name>
    <url-pattern>/ServletHelloWorld</url-pattern>
  </servlet-mapping>
</web-app>
```

→ Nutzung des Servlets

`webapps/servlet-example/WEB-INF/classes/servlet/ServletHelloWorld.class`

JSP

JavaServer Pages

- ❑ Einbettung von Java-Code in HTML → alles in einer Datei
- ❑ Java für dynamische, HTML für statische Dokumentbestandteile

```
...
<html>
  <body>
    <h1>Hello World</h1>
    <% String user=request.getParameter("user"); %>
    <%= clock.getDayOfMonth() %>
    <%! private int accessCount = 0; %>
    ...
  </body>
</html>
```

JSP-Dokument

- > JSP-Scriptlet (beliebiger Java-Code)
- > JSP-Expression (Ausgabe als String)
- > JSP-Declaration (Deklaration von Variablen und Methoden)

JSP-Prozessor

```
public class SampleServlet extends HttpServlet {
  ...
  private int accessCount = 0;
  public void _jspService(...) throws ServletException {
    out.println("<h1>Hello World</h1>");
    String user=request.getParameter("user");
    out.println(clock.getDayOfMonth().toString());
    ...
  }
}
```

JSP mit Tomcat

Beispiel

1. Archiv `jsp-example.war` von der Kursseite herunterladen
2. Dieses in `webapps`-Ordner von Tomcat ablegen
3. Tomcat starten (falls noch nicht getan)
→ Tomcat wird das Archiv automatisch entpacken.
4. Im Browser öffnen: <http://localhost:8080/jsp-example/>
Eingabe tätigen und auf `Anmelden` klicken.

Anmeldung

Benutzername:

Anmelden



Ihre Anmeldedaten:

Benutzername: Ken Thompson

JAXB Hinweis

XML-Verarbeitungsanweisungen nachträglich hinzufügen

- ❑ JAXB beachtet XML-Verarbeitungsanweisungen wie bspw.
`<?xml-stylesheet type="type" href="uri"?>`
beim Serialisieren eines Instanzdokuments nicht!
- ❑ Nachträgliches Hinzufügen mit `Marshaller.setProperty-Methode`
- ❑ Die Methode erwartet zwei Strings:
 - `name`: Bezeichner der zu setzende Eigenschaft:
`com.sun.xml.internal.bind.xmlHeaders`
 - `value`: Der zu setzende Wert. (Für uns eine vollständige Verarbeitungsanweisung)
- ❑ **Beispiel: Verweis auf XSL-Sheet**

```
String xslSheetPath = "xsl/atom-to-html.xsl";
```

```
String xslSheetPI = String.format(  
    "<?xml-stylesheet type='text/xsl' href='%s'?>", xslSheetPath  
);
```

```
marshaller.setProperty(  
    "com.sun.xml.internal.bind.xmlHeaders", xslSheetPI  
);
```