

JAVA goes Web!

Möglichkeiten und Beispiele des Einsatzes der Programmiersprache JAVA in Web-Anwendungen

Dr. Irene Weber

Ausgangspunkt und Ziel

- 2 Semester Java-Programmierung
- Im 3. Semester: Webanwendungen mit Java
- Ziel heute:
 - Komponenten einer Webanwendung
 - Zusammenspiel der Komponenten
 - JavaServerPages

Gliederung

- **Webanwendung**
- Der Webserver
- HTML - HTTP
- Das Servlet
- JavaServerPages

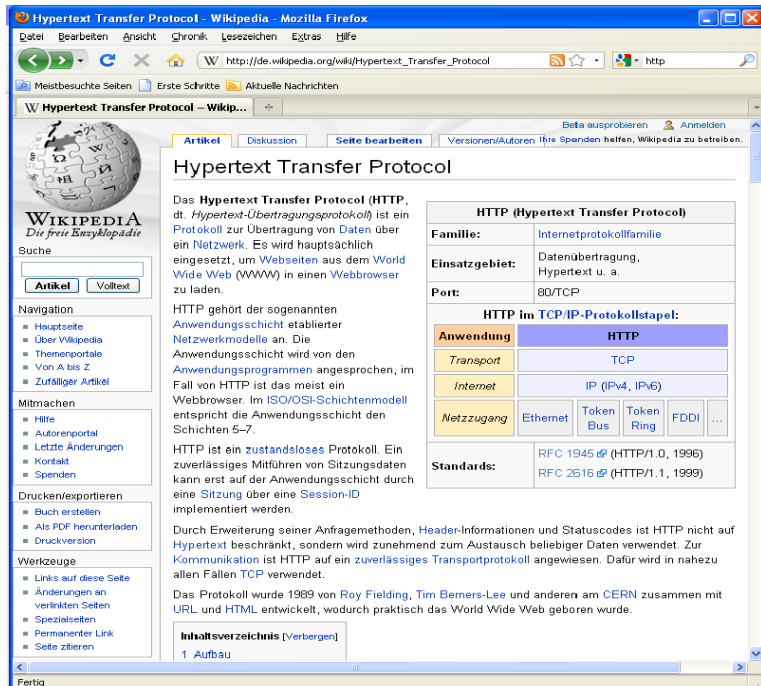
Web-Anwendung

Client-seitig

Internet
Intranet
Lokal

Server-seitig

Browser (Client)

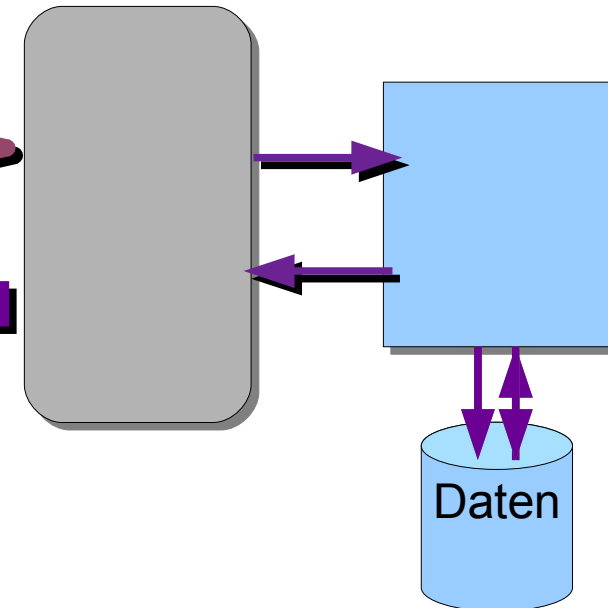


HTTP-Request

HTTP-Response

Webserver

Programm



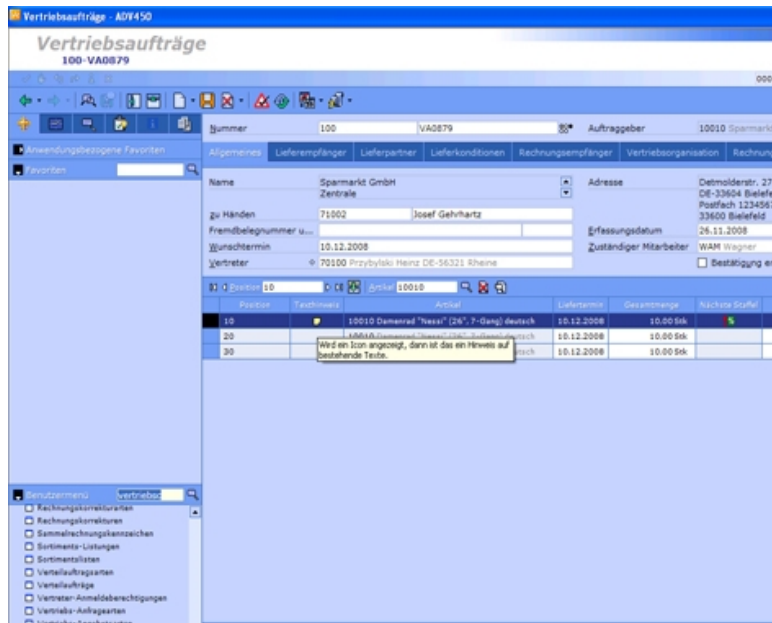
Web-Anwendung

Client-seitig

Internet
Intranet
Lokal

Server-seitig

Browser (Client)

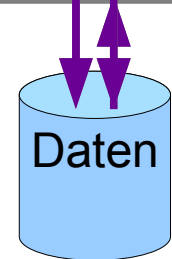
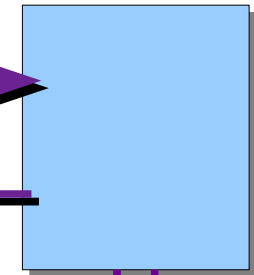
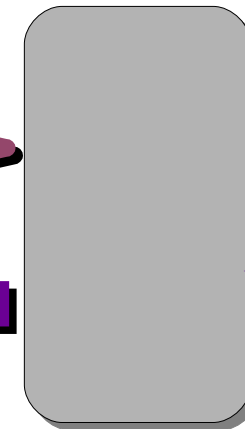


HTTP-Request

HTTP-Response

Webserver

Programm



Gliederung

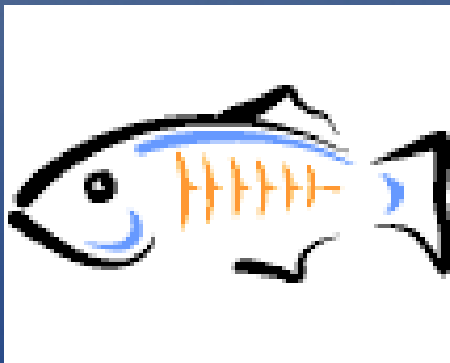
- Webanwendung
- **Der Webserver**
- HTML - HTTP
- Das Servlet
- JavaServerPages

Webserver

Tomcat Servlet Container



Sun Application Server
GlassFish



Web Server und Servlet Container



JBoss Application Server

Praxis



Webappjava vom USB_Stick direkt unter C:\ kopieren
ablauf.html öffnen

Gliederung

- Webanwendung
- Der Webserver
- **HTML - HTTP**
- Das Servlet
- JavaServerPages

HTML

The screenshot shows a Mozilla Firefox browser window with the title "oskar orakel - Mozilla Firefox". The address bar shows the file path "file:///C:/webappjava/HTML/intro-input.html". The browser displays a web page with the following content:

Ich bin Oskar, das moderne web-basierte Orakel für Hilfe in allen Lebenslagen

Hallo, Unbekannter!

Oskar sagt:

Wer kommt da und fragt?

Dein Name ist:

The source code window shows the following HTML code:

```
<!DOCTYPE HTML "PUBLIC -/W3C/DTD HTML 4.0 Transitional/EN">
<html>
<head><title>oskar orakel</title></head>
<style type="text/css">
body { background-color:#AAAAFF; }
</style>
</head>
<body>

<h2>Ich bin Oskar, das moderne web-basierte Orakel für Hilfe in allen Leb
<h3>Hallo, Unbekannter!</h3>

<h3>Oskar sagt:</h3> Wer kommt da und fragt?

<h3>Dein Name ist:</h3>
<form action = "intro-login.html" method = "get">
<input type="text" name="client">
</form>
</body></html>
```

Client an Server: HTTP Request

- Zwei wichtige Request-Methoden
 - GET
 - POST

```
GET /intro-input.html HTTP/1.0  
CONNECTION: Keep-alive  
...
```

Server an Client: HTTP Response

- Fehlercode
- Response-Header
- Inhalt

```
C:\> Eingabeaufforderung

HTTP/1.1 200 OK
Server: Apache-Coyote/1.1
Accept-Ranges: bytes
ETag: W/"509-1257694281175"
Last-Modified: Sun, 08 Nov 2009 15:31:21 GMT
Content-Type: text/html
Content-Length: 509
Date: Sun, 08 Nov 2009 19:51:18 GMT
Connection: keep-alive

<!DOCTYPE HTML "PUBLIC -/W3C/DTD HTML 4.0 Transitional/EN">
<html>
<head><title>oskar orakel</title></head>
<style type="text/css">
body { background-color:#AAAAFF; }
</style>
</head>
<body>

<h2>Ich bin Oskar, das moderne web-basierte Orakel f³r Hilfe in allen Lebenslagen</h2>
<h3>Hallo, Unbekannter!</h3>

<h3>Oskar sagt:</h3> Wer kommt da und fragt?

<h3>Dein Name ist:</h3>
<form action = "intro-login.html" method = "get">
<input type="text" name="client">
</form>
</body></html>

Verbindung zu Host verloren.
```

Praxis



<http.html> öffnen

Gliederung

- Webanwendung
- Der Webserver
- HTML - HTTP
- **Das Servlet**
- JavaServerPages

Servlet

- Eine Java-Klasse zur dynamischen Erzeugung von Webseiten
- Implementiert das Interface `javax.servlet.Servlet`
- Abgeleitet von `HttpServlet`

Servlet

```
import javax.servlet.*;

public class ControllerServlet extends HttpServlet
{
    @Override public void doGet(HttpServletRequest request,
        HttpServletResponse response) throws IOException,
        ServletException
    {
        PrintWriter out = response.getWriter();
        out.println("<html><body>Hallo, Welt</body></html>");
    }
}
```


Aufbau einer Webanwendung

Ordner	Inhalt
/	Html, jsp, statischer Dateien, Bilder
/WEB-INF	web.xml
/WEB-INF/classes	Klassendateien
/WEB-INF/lib	jar-Archive

Praxis



servlet.html öffnen

Gliederung

- Webanwendung
- Der Webserver
- HTML - HTTP
- Das Servlet
- **JavaServerPages**

Java Server Pages

- Java-Code eingebettet in Html.

```
<html>
<head><title>oskar orakel</title></head></head>
<body>
<% if (request.getParameter("client")== null) { %>
    <h3>Hallo, Unbekannter!</h3>
<% } else { %>
    <h3>Hallo, <%= request.getParameter("client") %>!</h3>
<% } %>
</body>
</html>
```

Praxis



jsp.html öffnen

Zusammenfassung

- Webanwendung
- Der Webserver
- HTML - HTTP
- Das Servlet
- JavaServerPages

Ausblick – wie geht es weiter?

- Eine Entwicklungsumgebung einrichten – Eclipse
- Cookies, Request Attribute, Session Attribute, ...
- JSP Sprachelemente: Tags, ...
- Systemaufbau: Model-Controller-View
- Java Server Faces
- ...