

# SERVLETS ÜBUNGEN

## In diesem Kapitel:

- Übungen
  - *Installation von Jakarta*
    - Hello World Servlet
  - *Installation des Java Web Servers*
    - Hello World Servlet
  - *Zugriff auf Servlet Parameter*
    - ServiceTime Servlet
  - *Generierung von Inline Content*
    - DateTime Servlet
  - *HTML Formulare*
    - Lesen und Anzeigen von DBMS Daten

## *Servlets* - *Praktische Beispiele*

### 1.1. **Einleitung**

#### 1.1.1. Generelle Bemerkungen zu den Übungen

Eine Übung besteht aus der Aufgabenstellung und Hilfestellungen. Einige der Übungen können Sie eventuell selbständig lösen; bei anderen werden Sie Hilfe oder Lösungshinweise benötigen, beispielsweise weil die Aufgabenstellung für Sie zu unverständlich ist! Sie können sich aber auch einfach die Musterlösung ansehen, sofern eine existiert (beispielsweise bei der Installation der Software kann man kaum eine Musterlösung erwarten: die Installation geschieht auf Ihrem Rechner). Die einzelnen Übungen bauen aufeinander auf. Aber da Sie in der Regel eine Musterlösung haben, sollte es Ihnen trotzdem möglich sein, den Übungen zu folgen.

#### 1.1.2. Die Anatomie einer Übung

Jede Übung enthält eine Liste jener Übungen, welche Sie bereits gelöst haben sollten, um die aktuelle Übung verstehen zu können. Zudem finden Sie Rahmenprogramme, welche Sie als Basis für die eigene Lösung verwenden können. Falls vorhanden, werden Sie auch noch Links zu den API Seiten finden, damit Sie eventuell benötigtes Hintergrundwissen ergänzen oder nachschlagen können. Jede Übung enthält eine Liste der Lernziele, also der Ziele, die Sie hoffentlich erreichen. Die Hilfestellungen finden Sie gleich anschliessend an die Fragestellungen, in der Regel Schritt für Schritt.

Sie finden also pro Übung folgende Informationen:

- **Hilfen**  
mit Hinweisen zur Lösung der gestellten Aufgabe;  
eine Lösungsskizze mit Kommentaren und allem drum und dran;
- **Musterlösung**  
Die Musterlösung enthält auch die Applet Tags und alle benötigten Dateien.
- **API Dokumentation**  
Links zu den Online Dokumentation bei Sun

# SERVLETS ÜBUNGEN

## 1.1.3. Design Ziele für die Übungen

Es gibt drei Arten von Übungen:

- **"Blank-Screen"** Übungen  
Sie finden einen leeren Bildschirm vor ("Blank Screen") und müssen die gesamte Lösung selber erarbeiten.
- **Erweiterungen**  
Sie erweitern eine bereits bestehende Lösung
- **Korrektur**  
Sie korrigieren das Verhalten eines bestehenden Programms.

Um Ihnen das Erlernen der Servlet Techniken zu erleichtern, werden in den Übungen in der Regel nur jene Teile zu erschaffen sein, welche spezifisch für Servlets sind. Alle irrelevanten und komplexen Bereiche, die Sie eventuell zur Lösung benötigen, werden Ihnen zur Verfügung gestellt.

In der Regel werden die Übungen mit Hilfe eines Web Servers / des Webs durchgeführt. In den Fällen, in denen Sicherheitsprobleme auftauchen könnten, werden die Beispiele lokal ausgeführt.

In dieser Kurseinheit lernen Sie praktisch, wie man Servlets erstellt und einsetzen kann. Nach dem Durcharbeiten dieser Kurseinheit sollten Sie wissen, wie man Servlets erstellt und in Web Servern einbaut. Sie werden auch lernen, wie Client Applets mit Servlets kommunizieren können, um Server oder Datenbank Informationen abzufragen. Darauf aufbauend bleibt es Ihnen überlassen, ausgefeilte Anwendungen zu entwickeln und die Tiefen der Technologie auszuloten....

# SERVLETS ÜBUNGEN

## 1.1.4. Übersicht über die Übungen

### 1. Installation des Servletrunners

In dieser Übung installieren Sie das JSDK um einfache Servlets zu entwickeln und mit dem Server zu testen..

Lernziele:

- übersetzen Sie ein einfaches "Hello World" Servlet.
- installieren Sie das Servlet im Server
- aktivieren Sie das Servlet und testen Sie es mit einem Browser.

### 2. Nutzen Sie ein Servlet in einem entsprechenden Server, (Java Web Server)

Lernziele:

- kreieren und übersetzen Sie ein einfaches Hello World Servlet.
- Installieren Sie das Servlet im Server
- Starten Sie das Servlet aus dem Browser

### 3. Zugriff auf die Service Parameter

Diese Übung zeigt, wie man auf Service Parameter zugreifen kann.

Lernziele:

- Sie lernen, wie man auf Service Parameter zugreift
- Sie formatieren die erhaltene Information und senden diese an den Client zurück

### 4. Generieren von Inline Code

Schreiben Sie ein Servlet, welches Inline Code generiert. Aktivieren Sie dieses Servlet mit einem Servlet Tag : <servlet>

Lernziele:

- Schreiben Sie ein Servlet, welches Inline Code generiert.
- Kreieren Sie eine "shtml" Datei, mit der Sie auf das neu kreierte Servlet zugreifen können.

### 5. Einsatz und Bearbeitung von HTML Formularen

Sie entwickeln zwei Servlets, FormDisplay.java und FormProcessing.java, mit denen Sie HTML Formulare analysieren und verarbeiten können.

Lernziele:

- Sie können HTML Formulare dynamisch mit Informationen auf der Harddisk kombinieren.
- Sie lernen, wie man HTML Formulare serverseitig analysiert und verarbeitet.

# SERVLETS ÜBUNGEN

## 1.2. Installation des Servletrunners

In dieser Übung installieren Sie das JSDK um einfache Servlets zu entwickeln und mit dem Server zu testen..

### 1.2.1. Voraussetzungen

Sie müssen die entsprechende Software herunterladen. Eine lokale Kopie finden Sie auf dem Server / der CD. Die aktuellen Versionen finden Sie unter <http://java.sun.com/products/servlet> speziell <http://java.sun.com/products/servlet/download.html>

### 1.2.2. Lernziele

Die speziellen Lernziele dieser Übung sind:

- übersetzen Sie ein einfaches "Hello World" Servlet.
- installieren Sie das Servlet im Server
- aktivieren Sie das Servlet und testen Sie es mit einem Browser.

### 1.2.3. Skeleton Code

```
import java.io.*;
import javax.servlet.*;
import javax.servlet.http.*;
/**
 * HelloWorldServlet sendet eine statische HTML Seite
 */
public class HelloWorldServlet extends HttpServlet
{
    public void doGet
    (
        HttpServletRequest req,    // Infos des Browsers
        HttpServletResponse res    // Infos an Browser
    ) throws
        ServletException,        // typische Servlet Exception
        IOException
    {
        // MIME Type setzen
        // hier Text/HTML
        res.setContentType( "text/html" );
        // Ausgabestrom
        ServletOutputStream out = res.getOutputStream();
        // generieren der HTML Seite
        out.println( "<html>" );
        out.println( "<head><title>Hello World</title></head>" );
        out.println( "<body>" );
        out.println( "<h1>Hello World</h1>" );
        out.println( "<p>Gratuliere, Ihr erstes Servlet funktioniert!</p>" );
        out.println( "</body>" );
        out.println( "</html>" );
        out.close();
    }
}
```

In dieser Übung lernen Sie, wie man ein denkbar einfaches Servlet schreibt. Fast wichtiger als das Servlet ist, dass Sie sich mit den Verzeichnissen der Referenzimplementation (Tomcat /

# SERVLETS ÜBUNGEN

Jakarta bzw. servletrunner) vertraut machen. Daher finden Sie die Lösung inklusive Batch Prozeduren für dieses Beispiel auf dem Server / der CD. Sie sollten einzip die Verzeichnisse überprüfen.

Das Servlet selber habe ich im JBuilder kreiert. Zuerst muss man die Bibliotheken für die Servlets hinzufügen. Ab JDK1.2 sind die Servlet Archive bereits Teil des JDK.

# SERVLETS ÜBUNGEN

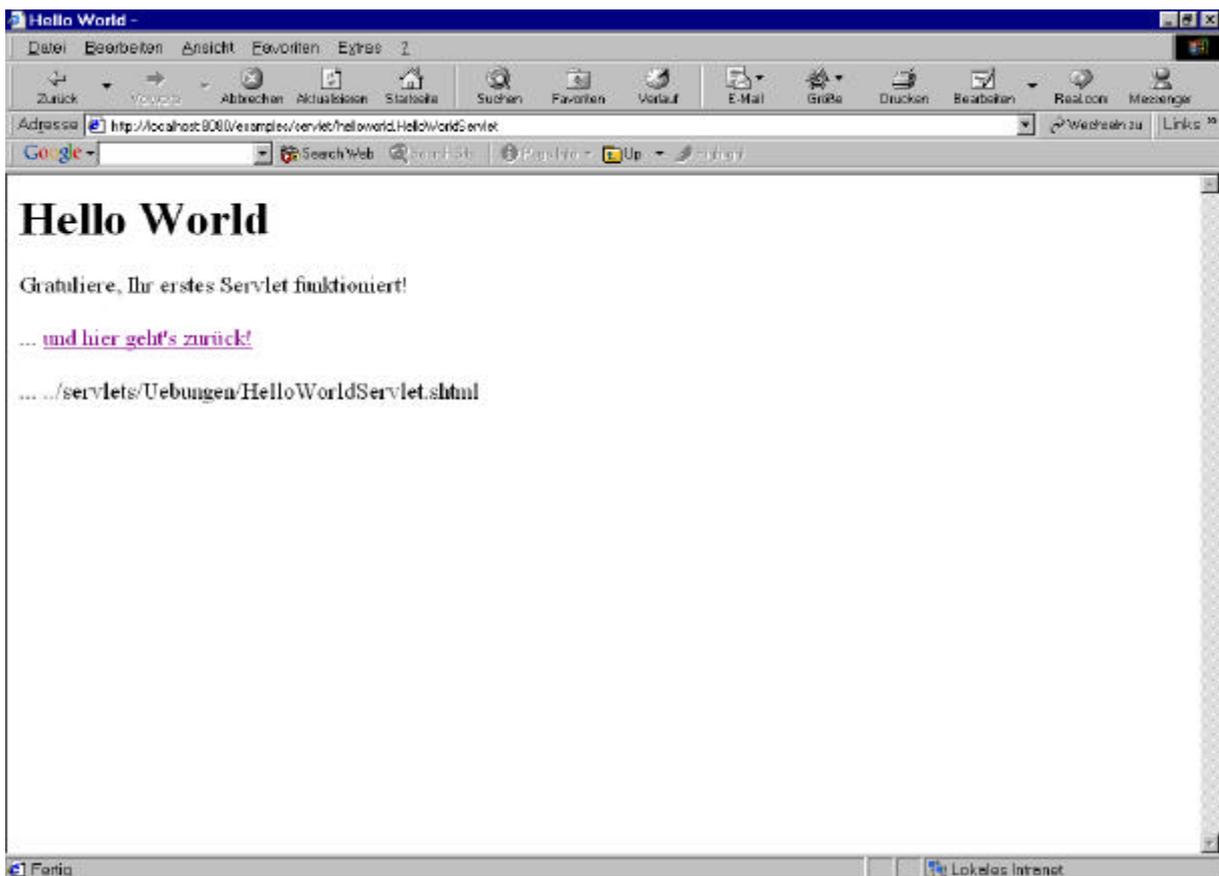
## 1.2.4. Ihre Aufgaben

1. übersetzen Sie das Hello World Servlet. Sie können es auch gleich modifizieren, falls Ihnen dies Spass macht. Aber besser wäre es, wenn Sie alles zuerst unverändert lassen würden.
2. starten Sie den Servletrunner (eigentlich Jakarta) mit dem die Servlets aktiviert werden sollen.
3. aktivieren Sie das Servlet im Browser. Geben Sie dazu folgende URL ein  
`http://localhost:8080/examples/servlet/helloworld.HelloWorldServlet`  
oder  
`http://localhost:8080/examples/servlets/Uebungen/HelloWorldServlet.shtml`

Beachten Sie die feinen Unterschiede:

- das Servlet selber wird mit ... examples/servlet ...
- die Web Seite mit ... examples/servlets... (mit s)  
gestartet.

## 1.2.5. Demo



# SERVLETS ÜBUNGEN

## 1.2.6. Lösungshinweise

1. *übersetzen Sie das Hello World Servlet. Sie können es auch gleich modifizieren, falls Ihnen dies Spass macht. Aber besser wäre es, wenn Sie alles zuerst unverändert lassen würden.*

```
javac -classpath .;%JAVASERVLET_HOME%\servlet.jar HelloWorldServlet.java
```

2. *starten Sie den Servletrunner (eigentlich Jakarta) mit dem die Servlets aktiviert werden sollen.*

In der Version 2.0 des JSDK existierte `servletrunner` als exe Datei im `bin` Verzeichnis. In der Version 2.1 und in der Jakarta (Tomcat) Version von Apache starten Sie einfach den Server und beachten die Verzeichnisse.

Falls Sie das JSDK 2.1 einfach ins Verzeichnis `c:\jsdk2.1` kopieren, dann finden Sie ein `Startup.bat` und ein `Shutdown.bat` Skript.

Starten Sie den Server und kontrollieren Sie Ihre Installation mit dem Browser:

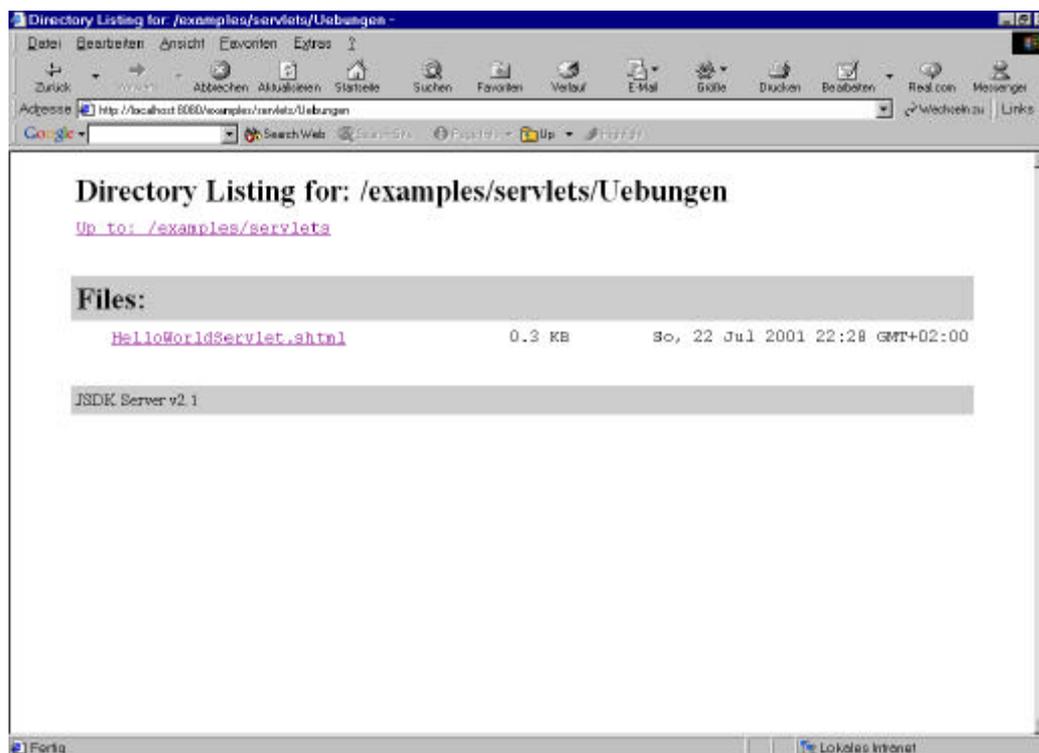
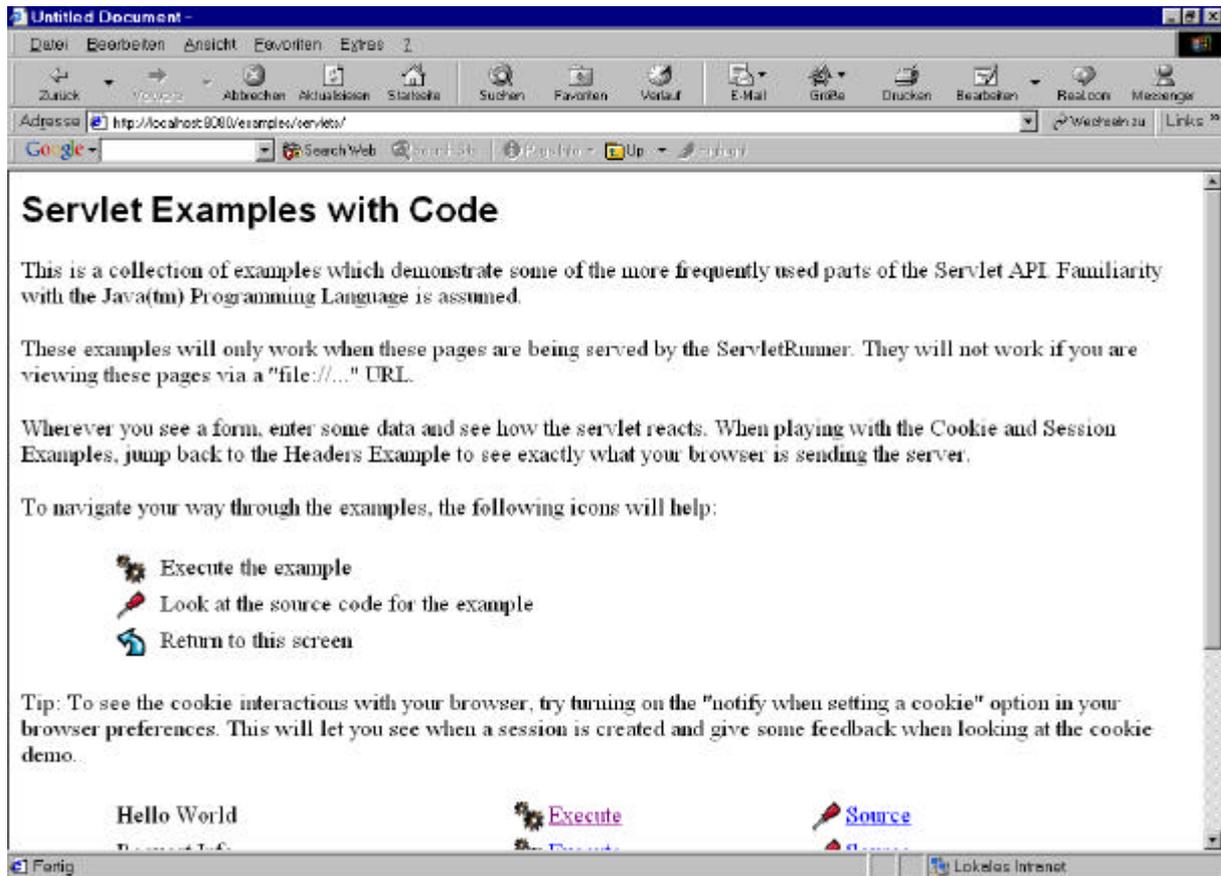
Sie sollten dann das Bild unten, mit Links in die Dokumentation und die Beispiele, sehen.

Testen Sie einige der Beispiele und schauen Sie vorallem auch nach, in welchen Verzeichnissen diese stehen.

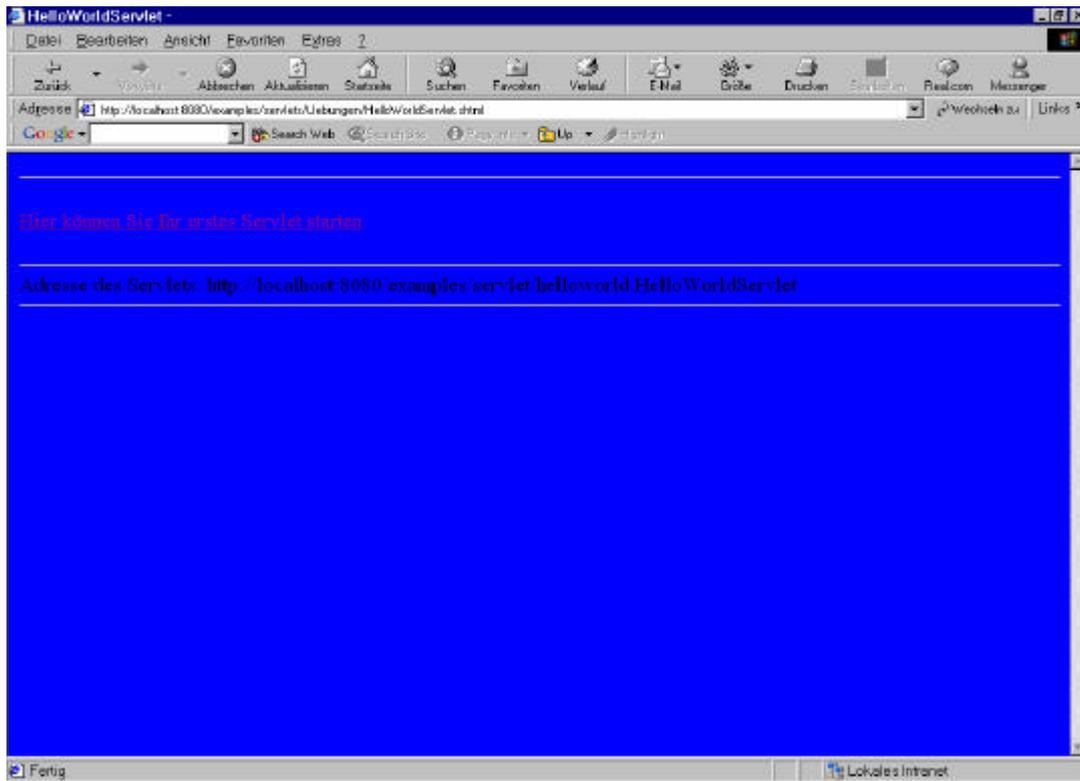
Ich habe für die Beispiele aus dieser Kurseinheit zusätzliche Verzeichnisse kreiert. Diese legt Ihnen auch das Skript `KreiereDirectories.bat` an.



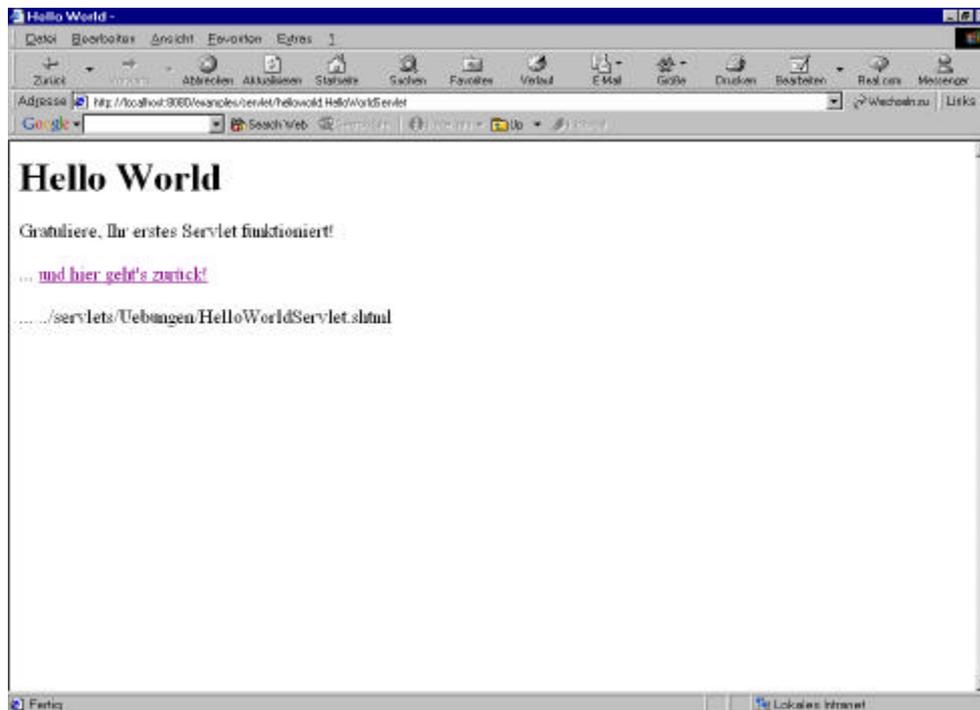
# SERVLETS ÜBUNGEN



# SERVLETS ÜBUNGEN



- aktivieren Sie das Servlet im Browser. Geben Sie dazu folgende URL ein  
`http://localhost:8080/examples/servlet/helloworld>HelloWorldServlet`  
oder



`http://localhost:8080/examples/servlets/Uebungen/HelloWorldServlet.shtml`

# SERVLETS ÜBUNGEN

## 1.2.7. Musterlösung

Sie finden die Musterlösung bereits oben und in leicht modifizierter Form auf dem Server, inklusive der Batch Dateien und der HTML Rahmen sowie des JBuilder Projekts.

# SERVLETS ÜBUNGEN

## 1.3. *Servlets auf dem Java Web Server*

In dieser Übung verwenden Sie im Wesentlichen das selbe Servlet wie in der vorigen Übung, aber ans Stelle von Jakarta verwenden Sie den JWS.

### 1.3.1. Voraussetzungen

Sie haben den Java Web Server zur Verfügung. Da er nicht mehr unterstützt wird, finden Sie eine Kopie auf dem Server / der CD.

### 1.3.2. Lernziele

Mit dieser Übung verfolgen wir folgende spezifischen Lernziele:

- kreieren und übersetzen Sie ein einfaches Hello World Servlet.
- Installieren Sie das Servlet im Server
- Starten Sie das Servlet aus dem Browser

### 1.3.3. Skeleton Code

```
import java.io.*;

import javax.servlet.*;
import javax.servlet.http.*;

/**
 * HelloWorldServlet : einfache statische HTML Seite
 */
public class HelloWorldServlet2 extends HttpServlet
{
    public void doGet
    (
        HttpServletRequest req,    // Infos vom Browser
        HttpServletResponse res    // Infos an den Browser
    ) throws
        ServletException,        // allgemeine Exception
        IOException
    {
        // MIME Type
        res.setContentType( "text/html" );

        // Referenz auf Output Stream
        // zum Browser (als dessen Input)
        ServletOutputStream out = res.getOutputStream();
        // Kreieren einer HTML Seite
        out.println( "<html>" );
        out.println( "<head><title>Hello World</title></head>" );
        out.println( "<body>" );
        out.println( "<h1>Hello World</h1>" );
        out.println( "<p>Gratuliere, auch das zweite Servlet funktioniert!</p>"
        );
        out.println( "</body>" );
        out.println( "</html>" );
    }
}
```

# SERVLETS ÜBUNGEN

## 1.3.4. Ihre Aufgaben

Sie sollten folgende Aufgaben lösen:

1. übersetzen Sie JWSHelloWorldServlet.java. Sie können dabei auf das JBuilder Projekt der Musterlösung zugreifen.
2. verschieben Sie das Servlet in das Standard Servlet Verzeichnis des JWS.  
`%JAVAWS_HOME%\servlets\jwshelloworld\`
3. Starten Sie den JWS.  
`%JAVAWS_HOME%\bin\jserv -javahome %JAVA_HOME%`
4. Installieren Sie das Servlet. Starten Sie dazu das Admin Servlet des JWS (<http://localhost:9090>).  
Installieren Sie das Servlet und kreieren Sie einen Alias 'HW2'.
5. Aktivieren Sie das Servlet aus dem Browser :  
<http://localhost:8080/servlet/jwshelloworld.JWSHelloWorldServlet>
6. Aktivieren Sie jetzt das selbe Servlet aber mit dem Alias:  
[http://localhost:8080/servlet/JWS\\_HW](http://localhost:8080/servlet/JWS_HW)  
Sie müssen dazu bei der Spezifikation des Servlets im Admin Tool in der ersten Maske den Alias als Namen angeben.

Nach dem Durcharbeiten dieser Übung sollten Sie ein Servlet erfolgreich im JWS installiert haben. Zudem haben Sie einen Alias Namen definiert, unter dem Sie das Servlet aufrufen können.

# SERVLETS ÜBUNGEN

## 1.3.5. Demo



mit dem Link:

<http://localhost:8080/servlet/jwshelloworld.JWSHelloWorldServlet>

oder als Alias:



mit dem Link

[http://localhost:8080/servlet/JWS\\_HW](http://localhost:8080/servlet/JWS_HW)

# SERVLETS ÜBUNGEN

## 1.3.6. Lösungshinweise

1. *übersetzen Sie JWSHelloWorldServlet.java. Sie können dabei auf das JBuilder Projekt der Musterlösung zugreifen.*

Sie können das Programm direkt mit

```
javac -classpath c:\JBuilder\lib\servlet.jar  
JWSHelloWorldServlet.java
```

übersetzen (alles auf einer Zeile). Alternativ dazu steht Ihnen das JBuilder Projekt zur Verfügung.

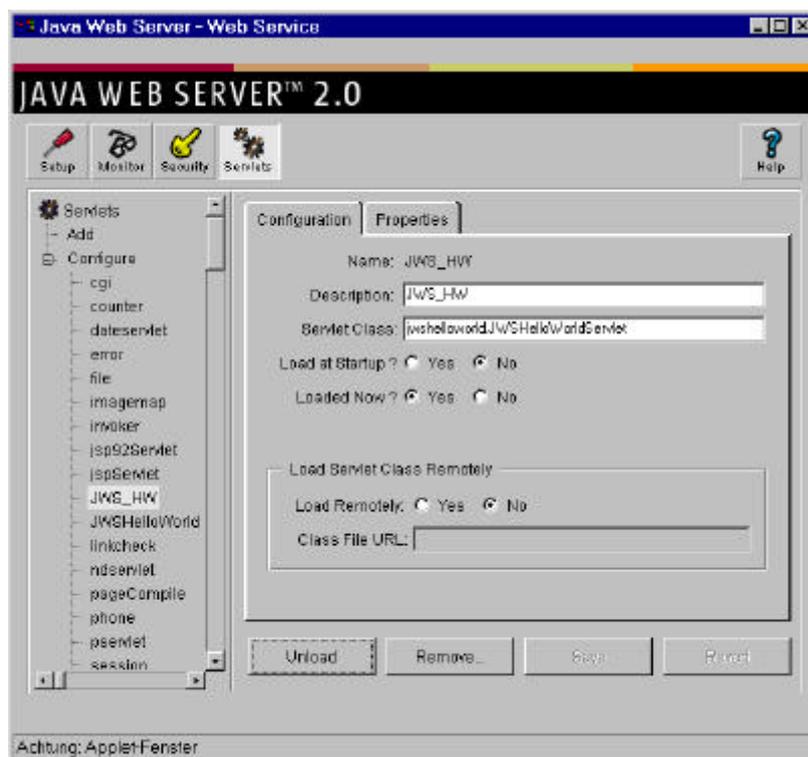
2. *verschieben Sie das Servlet in das Standard Servlet Verzeichnis des JWS.*  
`%JAWAWS_HOME%\servlets\jwshelloworld\`

Sie können dazu die Batch Datei `2KopierenDerDateienZumJavaWebServer.bat` verwenden

3. *Starten Sie den JWS.*  
`%JAWAWS_HOME%\bin\jserv -javahome %JAVA_HOME%`

Auch dafür steht Ihnen ein Skript zur Verfügung:  
`1StartenJavaWebServer.bat`

4. *Installieren Sie das Servlet. Starten Sie dazu das Admin Servlet des JWS*  
*(<http://localhost:9090>).*  
*Installieren Sie das Servlet und kreieren Sie einen Alias 'HW2'.*



# SERVLETS ÜBUNGEN

5. Aktivieren Sie das Servlet aus dem Browser :  
`http://localhost:8080/servlet/jwshelloworld.JWSHelloWorldServlet`
  
6. Aktivieren Sie jetzt das selbe Servlet aber mit dem Alias:  
`http://localhost:8080/servlet/JWS_HW`  
Sie müssen dazu bei der Spezifikation des Servlets im Admin Tool in der ersten Maske den Alias als Namen angeben.

Sie finden auch im JWS jede Menge Beispiele zu Java Server Pages (CheckBox, Counter und vieles mehr), und Servlets (Applet-Servlet Kommunikation, JDBC und RMI Anbindung und vieles mehr).

## 1.3.7. Musterlösung

Sie finden den Java Code auf dem Server / der CD. Zudem finden Sie dort mehrere Batch Skripte, mit denen Sie die Class Datei an den richtigen Ort verschieben können.

# SERVLETS ÜBUNGEN

## 1.4. Zugriff auf Servlet Service-Zeit Parameter

### 1.4.1. Voraussetzungen

Sie haben die zwei vorangehenden Übungen (Jakarta, Java Web Server Hosting) durchgearbeitet.

### 1.4.2. Lernziele

In dieser Übung greifen Sie auf zwei Parameter zu, die in der URL eingebettet sind.

Lernziele:

- Sie lernen, wie man auf Service Zeit Parameter des Servlets zugreifen kann.
- Sie formatieren die Information und geben diese an den Client zurück.

### 1.4.3. Skeleton Code

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

/**
 * ServletParameterTest servlet.
 * Das Servlet bestimmt die Parameter zur Laufzeit und
 * verwendet diese zur formatierten Ausgabe
 */
public
class ServletParameterTest extends HttpServlet {

    public void doGet
    (
        HttpServletRequest req,
        HttpServletResponse res
    )
    throws
        ServletException,
        IOException
    {

        // Variablen zur Aufnahme der Parameter
        String paramName;
        String paramPosition;

        // Ausgabestrom zum Browser
        ServletOutputStream out;

        // Ihre Eingabe (eine Zeile)
        // Output Stream ist HTML

        // Ihre Eingabe (eine Zeile)
        // Name und Positionsparameter

        // Ihre Eingabe (eine Zeile)
        // Ausgabestrom zum Client
    }
}
```

# SERVLETS ÜBUNGEN

```
out.println("<html>");
out.println("<head><title>Parameter Test Servlet</title></head>");
out.println("<body>");

out.println("<h1>Parameter Test Servlet</h1>");
out.println("<p>");
out.println( paramName + ", Gratuliere, Sie haben die Parameter korrekt
bestimmt " + paramPosition );
out.println("</p>");
out.println("</body>");
out.println("</html>");
out.close();
}

public String getServletInfo()
{
    return "Diese Servlet verwendet Parameter zur Formattierung der
Ausgabe";
}
}
```

## 1.4.4. Ihre Aufgaben

In dieser Übung lesen Sie Parameter, die zur Laufzeit an das Servlet übergeben werden und schickt diese an den Browser zurück. Diese Rückgabe erfolgt als HTML Seite.

Die zwei Parameter sind Name und Position und werden als Teil der URL an das Servlet übergeben. Das Servlet benutzt diese zwei Parameter (Name und Position) um die Ausgabe passend zu formattieren.

1. Ergänzen Sie das Rahmenprogramm, indem Sie an den vorgegebenen Stellen die benötigten Zeilen einfügen. Zudem sollte die Ausgabe besser formatiert werden.
2. Bestimmen Sie den Namen des Kunden und seine Position.
3. Bestimmen Sie eine Referenz auf den Ausgabe Strom zum Client, in den Sie die HTML Seite schreiben können.
4. Übersetzen Sie das Servlet
5. Installieren Sie das Servlet entweder im JWS oder im Jakarta Server
6. Starten Sie das Servlet aus dem Browser
7. Starten Sie nun das Servlet im Browser und übergeben Sie die zwei Parameter (name= und position=) mit irgendwelchen Werten.

# SERVLETS ÜBUNGEN

## 1.4.5. Demo

Schauen wir uns zuerst den Servlet Aufruf an, ohne irgend welche Parameter:



Als Antwort erhalten wir einfach null Werte bei der URL:

```
http://localhost:8080/servlet/servicetimeparameters.ServletParameterTest
```

Nun starten wir das selbe Servlet, aber mit Parametern.



mit den Parametern in der folgenden URL:

```
http://localhost:8080/servlet/servicetimeparameters.ServletParameterTest?name=Joller&position=Prof%20Dr
```

# SERVLETS ÜBUNGEN

## 1.4.6. Hilfestellung

1. Ergänzen Sie das Rahmenprogramm, indem Sie an den vorgegebenen Stellen die benötigten Zeilen einfügen. Zudem sollte die Ausgabe besser formatiert werden.

```
res.setContentType("text/html");
```

2. Bestimmen Sie den Namen des Kunden und seine Position.

```
paramName      = req.getParameter("name"      );  
paramPosition  = req.getParameter("position");
```

3. *Bestimmen Sie eine Referenz auf den Ausgabe Strom zum Client, in den Sie die HTML Seite schreiben können.*

```
out = res.getOutputStream();
```

4. *Übersetzen Sie das Servlet*

Dazu steht Ihnen das jBuilder projekt zur Verfügung. Vergessen Sie nicht die Servlet Klassen einzubinden.

5. *Installieren Sie das Servlet entweder im JWS oder im Jakarta Server*

Diesen Schritt haben wir in der vorigen Übung bereits beschrieben. Auch hier steht ein DOS Skript zur Verfügung. Dabei müssen Sie zuerst ein Unterverzeichnis im Servlet Verzeichnis anlegen. Dies geschieht auch im Skript. Das Installieren in den JWS geschieht wie in der vorigen Übung.

6. *Starten Sie das Servlet aus dem Browser*

Das Starten des Web Servers und des Browsers ist mit Hilfe der DOS Skripte möglich.

7. *Starten Sie nun das Servlet im Browser und übergeben Sie die zwei Parameter (name= und position=) mit irgendwelchen Werten.*

Sie finden auf dem Server / der CD ein DOS Skript, mit dem der Internet Explorer im localhost mit Parametern gestartet wird.

## 1.4.7. Musterlösung

Sie finden die Musterlösung auf dem Server / der CD. Die Beispielausgaben sehen Sie oben.

# SERVLETS ÜBUNGEN

## 1.5. Generierung von Inline Content

Bisher haben wir in der Regel die Servlets direkt gestartet. Nun wollen wir diese in eine HTML Seite eingetten. Dies kann auf unterschiedliche Art und Weise geschehen, beispielsweise mit dem Servlet Tag, oder aber in Formularen als Skript.

### 1.5.1. Voraussetzungen

Sie haben den Web Server mit der Servlet Engine installiert und die vorherigen Übungen durchgearbeitet.

### 1.5.2. Lernziele

In dieser Übung aktivieren Sie das Servlet mit dem `<ervlet>` Tag.

Lernziele:

- Sie bauen ein Servlet, welches Inline Content generiert
- Sie kreieren eine "shtml" Datei, welche von JWS gestartet werden kann, um ein Servlet aufzurufen.

### 1.5.3. Skeleton Code

```
package datumsausgabeservlet;

import java.io.*;
import java.util.*;
import java.text.*;

import javax.servlet.*;
import javax.servlet.http.*;

/**
 * Datumsausgabe Servlet
 *
 * Das Servlet schreibt das aktuelle Datum
 * in eine generierte HTML Seite
 * als Server Side Include
 */
public class DatumsAusgabeServlet extends HttpServlet
{
    public void service
    (
        HttpServletRequest req,
        HttpServletResponse res
    )
    throws
        ServletException,
        IOException
    {
        String          sToday;
        Date            today;
        ServletOutputStream out;

        // aktuelles Datum, formattiert
        today = new Date();
    }
}
```

# SERVLETS ÜBUNGEN

```
sToday = DateFormat.getDateInstance().format( today );

// Ihre Zeile
// Ausgabe als text/plain
res.setContentType("text/plain");

// Ihre Zeile
// Referenz auf Ausgabestrom
out = res.getOutputStream();

// Ihre Zeile
// Datum in den Ausgabestrom
// print() nicht println() !!

out.print(sToday);
}

public String getServletInfo()
{
    return "Liefert inline HTML Darstellung des aktuellen Datums";
}
}
```

## 1.5.4. Ihre Aufgaben

In dieser Übung bauen Sie ein Servlet, welches eine Ausgabe produziert, welche als Inline Content in einer Web Seite benutzt wird. Der wesentliche Unterschied zu den Servlets, die wir bisher benutzt und gebaut haben, besteht darin, dass wir nicht "text/html" sondern "text/plain" produzieren, da dieser Text in eine HTML Seite eingebettet wird.

Das Beispiel bedient sich des Servlet Tags, muss also mit dem JWS ausgeführt werden. Dieser Tag ist nicht auf allen Servern verfügbar. Die Form des Tags ist sehr einfach:

```
<ervlet code="ServletName">
  </ervlet>
```

Nun Ihre Aufgaben:

1. ergänzen Sie das Rahmenprogramm / Servlet : spezifizieren Sie den korrekten MIME Type. ("text/plain").
2. ergänzen Sie das Rahmenprogramm: bestimmen Sie eine Referenz auf den Ausgabestrom zum Client zurück.
3. übersetzen Sie das Servlet
4. kopieren Sie das Servlet und die shtml Seite (auf dem Server / der CD oder im JBuilder generiert) zum Web Server.
5. starten Sie den Web Server und greifen Sie auf diese shtml Seite zu.

# SERVLETS ÜBUNGEN

## 1.5.5. Demo

Hier einige Screen Shots. Sie müssen beachten, dass die Extension der Web Datei "shtml" sein muss, und dass der Code ohne ".class" angegeben wird.

Hier die Testseite, aus dem JBuilder Projekt

```
<HTML>

<HEAD>
  <TITLE> Datumsausgabe Servlet
</TITLE>
</HEAD>

<BODY>

  <h4>Server Side Include Test</h4>
  <BR>
  Und hier die Datumsausgabe:

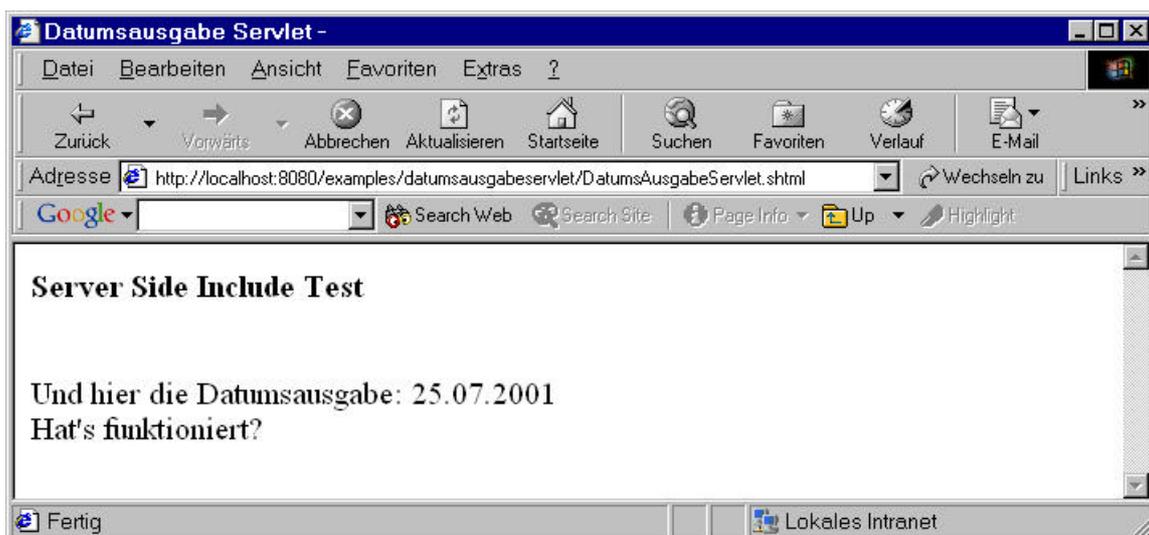
  <servlet code="datumsausgabeservlet.DatumsAusgabeServlet"></servlet>

  <BR>
  Hat's funktioniert?
  <BR>
</BODY>
</HTML>
```

Diese Datei wurde ins Verzeichnis

D:\JavaWebServer2.0\examples\datumsausgabeservlet>

verschoben und im Web Browser aufgerufen:



mit der URL

<http://localhost:8080/examples/datumsausgabeservlet/DatumsAusgabeServlet.shtml>

# SERVLETS ÜBUNGEN

## 1.5.6. Hilfestellung

Hilfestellungen zu den einzelnen Aufgaben:

1. *ergänzen Sie das Rahmenprogramm / Servlet : spezifizieren Sie den korrekten MIME Type. ("text/plain").*

```
res.setContentType("text/plain");
```

2. *ergänzen Sie das Rahmenprogramm: bestimmen Sie eine Referenz auf den Ausgabestrom zum Client zurück.*

```
out = res.getOutputStream();  
out.print(sToday);
```

3. *übersetzen Sie das Servlet*

wie gehabt: siehe letzte Übungen

4. *kopieren Sie das Servlet und die shtml Seite (auf dem Server / der CD oder im JBuilder generiert) zum Web Server.*

wie gehabt: siehe letzte Übungen

5. *starten Sie den Web Server und greifen Sie auf diese shtml Seite zu.*

Die URL und die Beispielausgabe sehen Sie oben unter "Demo".

## 1.5.7. Musterlösung

Diese sehen Sie bereits unter Demo. Nochmal: wichtig ist, dass die Datei die richtige Extension besitzt und das Servlet korrekt bezeichnet wird: ohne .class und ohne Verzeichnisangabe. In unserem Fall müssen wir das Package angeben, nicht ein Verzeichnis.

# SERVLETS ÜBUNGEN

## 1.6. Bearbeiten von HTML Formularen mit DBMS Zugriff

Wir wollen nicht nur einfache Formulare bearbeiten, sondern auch noch gleich den Zugriff auf eine Datenbank einbauen.

Sie entwickeln zwei Servlets, `FormDisplay.java` und `FormProcessing.java`, mit denen Sie HTML Formulare analysieren und verarbeiten können.

### 1.6.1. Lernziele

Ich gehe davon aus, dass Sie bereits die Kurseinheit zu JDBC durchgearbeitet haben!

Die spezifischen Lernziele dieses Beispiels sind:

- Sie können HTML Formulare dynamisch mit Informationen auf der Harddisk kombinieren.
- Sie lernen, wie man HTML Formulare serverseitig analysiert und verarbeitet.

### 1.6.2. Voraussetzungen

Sie haben die vorausgehenden Übungen durchgearbeitet, insbesondere haben Sie den Server aufgesetzt und die Übung zu den Service Zeit Parametern durchgearbeitet.

### 1.6.3. Skeleton Code

Da wir auch auf eine Datenbank zugreifen wollen, benötigen wir mehrere Programme:

- zwei Programme für den Datenbankzugriff  
`ColumnInfo.java`  
`CourseDB.java`
- zwei Programme für die Datenaufbereitung und Darstellung im Web.  
`FormDisplayServlet.java`  
`FormProcessingServlet.java`

### 1.6.4. Ihre Aufgaben

In dieser Übung lernen Sie, wie man HTML Formulare analysieren und in Servlets bearbeiten kann. Zusätzlich werden Sie lernen, wie Daten dynamisch dargestellt werden können, basierend auf Informationen aus dem Servlet zur Laufzeit (Service Time Parameter).

Das erste Servlet, `FormDisplayServlet`, ist bereits fertig: es akzeptiert zwei Parameter:

- *structurefile* und
- *datafile*

Der erste Parameter zeigt auf eine Datei, welche einen Data Dictionary enthält. Darin wird die Struktur der Daten beschrieben, welche dargestellt werden sollten.

Der zweite Parameter zeigt auf eine Datei, welche die aktuellen Daten enthält.

Das `FormDisplayServlet` bedient sich dieser zwei Parameter und generiert ein HTML Formular. Für den Benutzer sieht das Formular wie eine Tabelle voller Werte aus. Wenn man auf den Submit Knopf drückt, werden die Tabellen Daten an das Servlet gesandt und vom Servlet verarbeitet.

# SERVLETS ÜBUNGEN

Das Hauptziel der Übung ist, dass Sie das `FormProcessingServlet` vervollständigen.

Im speziellen sollten Sie folgende Aufgaben lösen:

- 1) Ergänzen Sie das `FormProceession Servlet` so, dass es die Daten, welche an den Server geschickt werden, analysieren kann.
- 2) Geben Sie ein Echo an den Browser: dieses enthält die eingegebenen Daten.
- 3) Schreiben Sie zusätzlichen Code, welcher die Daten in die Datei / die DBMS speichert.
- 4) Starten Sie das Servlet

```
http://localhost:8080/servlet/FormDisplayServlet?  
structurefile=struct.dat&datafile=data.dat
```

## 1.6.5. Demo

Wie Sie unten gleich lesen werden, ist diese Aufgabe nur unvollständig gelöst. Daher kann auch keine Demo angezeigt werden.

## 1.6.6. Hilfestellung

Die Hilfestellungen sind bewusst allgemein gehalten. Sie finden umfangreichere Servlets bei den Beispielen zum JWS.

- 1) *Ergänzen Sie das `FormProceession Servlet` so, dass es die Daten, welche an den Server geschickt werden, analysieren kann.*

Der Client sendet Informationen an den Server als HTTP POST Request. Um diese Informationen zu verarbeiten, muss die `doPost()` Methode überschrieben werden. Diese Methode muss einen `InputStream` öffnen. Alle Informationen des POST Requests werden über diesen Stream gelesen. Dieser Input Stream enthält die Parameter für die Strukturdatendatei und die Datendatei. Die Klasse `HttpUtil` enthält Methoden, mit deren Hilfe das Parsen der Informationen aus dem POST request durchgeführt werden kann.

- 2) *Geben Sie ein Echo an den Browser: dieses enthält die eingegebenen Daten.*

Dieser Teil der Übung wurde bereits fertig für Sie gelöst.

- 3) *Schreiben Sie zusätzlichen Code, welcher die Daten in die Datei / die DBMS speichert.*

Dieser Teil hängt von der DBMS ab und wurde weggelassen! Sie finden in der Lösung Hinweise und Skelette für eine Lösung. Eine vollständige Lösung liegt in diesem Sinne nicht vor!

# SERVLETS ÜBUNGEN

## 4) Starten Sie das Servlet

```
http://localhost:8080/servlet/FormDisplayServlet?  
structurefile=struct.dat&datafile=data.dat
```

### 1.6.7. Musterlösung

Eine eigentliche Musterlösung gibt es nicht, da die Lösung nicht vollständig ist: alle DBMS Teile fehlen!

# SERVLETS ÜBUNGEN

<b>SERVLETS - PRAKTISCHE BEISPIELE</b> .....	<b>1</b>
1.1. EINLEITUNG.....	1
1.1.1. <i>Generelle Bemerkungen zu den Übungen</i> .....	1
1.1.2. <i>Die Anatomie einer Übung</i> .....	1
1.1.3. <i>Design Ziele für die Übungen</i> .....	2
1.1.4. <i>Übersicht über die Übungen</i> .....	3
1.2. INSTALLATION DES SERVLETRUNNERS .....	4
1.2.1. <i>Voraussetzungen</i> .....	4
1.2.2. <i>Lernziele</i> .....	4
1.2.3. <i>Skeleton Code</i> .....	4
1.2.4. <i>Ihre Aufgaben</i> .....	6
1.2.5. <i>Demo</i> .....	6
1.2.6. <i>Lösungshinweise</i> .....	7
1.2.7. <i>Musterlösung</i> .....	10
1.3. SERVLETS AUF DEM JAVA WEB SERVER.....	11
1.3.1. <i>Voraussetzungen</i> .....	11
1.3.2. <i>Lernziele</i> .....	11
1.3.3. <i>Skeleton Code</i> .....	11
1.3.4. <i>Ihre Aufgaben</i> .....	12
1.3.5. <i>Demo</i> .....	13
1.3.6. <i>Lösungshinweise</i> .....	14
1.3.7. <i>Musterlösung</i> .....	15
1.4. ZUGRIFF AUF SERVLET SERVICE-ZEIT PARAMETER .....	16
1.4.1. <i>Voraussetzungen</i> .....	16
1.4.2. <i>Lernziele</i> .....	16
1.4.3. <i>Skeleton Code</i> .....	16
1.4.4. <i>Ihre Aufgaben</i> .....	17
1.4.5. <i>Demo</i> .....	18
1.4.6. <i>Hilfestellung</i> .....	19
1.4.7. <i>Musterlösung</i> .....	19
1.5. GENERIERUNG VON INLINE CONTENT .....	20
1.5.1. <i>Voraussetzungen</i> .....	20
1.5.2. <i>Lernziele</i> .....	20
1.5.3. <i>Skeleton Code</i> .....	20
1.5.4. <i>Ihre Aufgaben</i> .....	21
1.5.5. <i>Demo</i> .....	22
1.5.6. <i>Hilfestellung</i> .....	23
1.5.7. <i>Musterlösung</i> .....	23
1.6. BEARBEITEN VON HTML FORMULAREN MIT DBMS ZUGRIFF .....	24
1.6.1. <i>Lernziele</i> .....	24
1.6.2. <i>Voraussetzungen</i> .....	24
1.6.3. <i>Skeleton Code</i> .....	24
1.6.4. <i>Ihre Aufgaben</i> .....	24
1.6.5. <i>Demo</i> .....	25
1.6.6. <i>Hilfestellung</i> .....	25
1.6.7. <i>Musterlösung</i> .....	26