

Mini-Servlets

Abschlussprojekt Internetprogrammierung

Stefan Wehr

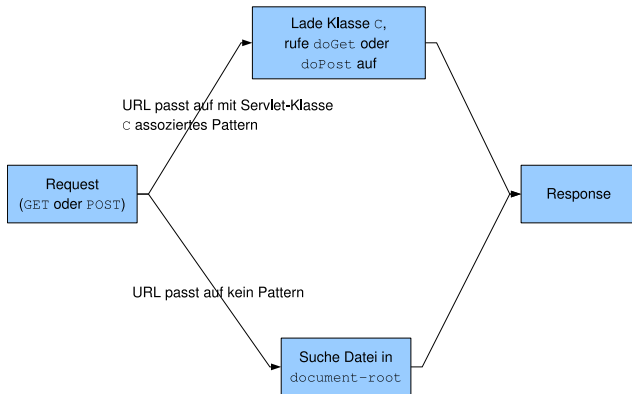
July 17, 2006

Implementierung eines einfachen Servletcontainers

- Webserver vorgegeben
- Parsen von Konfigurationsdateien
- Dynamisches Laden von Servletklassen
- Formularfelder
- Sessions

Requestbehandlung

Erweiterter Webserver



Session-ID in URL eingebettet (“URL Rewriting”)

```
http://localhost/HelloWorld?n=42
```

wird zu

```
http://localhost/HelloWorld;msessionId=74sf?n=42
```

Unterstützung des HTTP Protokolls

- GET: `application/x-www-form-urlencoded`
- POST:
 - `application/x-www-form-urlencoded`
 - `multipart/form-data`
- keine Unterstützung anderer Request-Arten
- Servlet Programmierer hat kein Zugriff auf HTTP Header
- keine Cookies

Servlet API

API angelehnt an originale API

```
public interface MServlet {  
    public void doGet(MServletRequest req, MServletResponse res);  
    public void doPost(MServletRequest req, MServletResponse res);  
}
```

Servlet API

```
public interface MServletRequest {  
    public String getServletPath();  
    public MSession getSession();  
    public RequestParameter getParameter(String name);  
    public RequestParameter[] getParameterValues(String name);  
    public Iterator<RequestParameter> getParameterNames();  
    public String encodeURL(String url);  
}
```

Servlet API

```
public interface RequestParameter {  
    public String getName();  
    public String getValue();  
}
```

```
public interface RequestFileParameter extends RequestParameter {  
    public String getContentType();  
    public InputStream getInputStream();  
}
```


Servlet API

```
public interface MServletResponse {  
    public void sendError(int status, String msg);  
    public void setContentType(String type);  
    public PrintWriter getWriter() throws IOException;  
    public PrintStream getOutputStream() throws IOException;  
}
```

Servlet API

```
public interface MSession {  
    public Object getAttribute(String name);  
    public void setAttribute(String name, Object value);  
    public Iterator<String> getAttributeNames();  
    public boolean isNew();  
}
```

Servlet API

```
public interface HttpConstants {  
    public static final int HTTP_OK = 200;  
    public static final int HTTP_BAD_REQUEST = 400;  
    public static final int HTTP_FORBIDDEN = 403;  
    public static final int HTTP_NOT_FOUND = 404;  
    /* ... */  
}
```

Konfiguration

- Variablen
 - `timeout` Der Timeout in Millisekunden (zwischen 1000 und 10000).
 - `workers` Die Anzahl der Worker-Threads (zwischen 5 und 25).
 - `logfile` Der Name der Logdatei.
 - `document-root` Das Verzeichnis unter dem statische Inhalte wie Bilder etc. abgelegt werden.
 - `class-root` Das Verzeichnis unter dem Servletklassen abgelegt werden.
- Servlet-Mapping: Abbildung Pattern → Klassenname

Konfiguration — Beispiel

```
<?xml version="1.0" encoding="iso-8859-1"?>
<mini-servlets-config>
  <timeout>2000</timeout>
  <workers>10</workers>
  <logfile>/home/wehr/mini-servlets/log</logfile>
  <document-root>/home/wehr/mini-servlets/docroot</document-root>
  <class-root>/home/wehr/mini-servlets/classroot</class-root>
  <servlet-mappings>
    <servlet-mapping>
      <url>*.do</url>
      <class>de.uni_freiburg.informatik.proglang.ControllerServlet</class>
    </servlet-mapping>
    <servlet-mapping>
      <url>/HelloWorld</url>
      <class>de.uni_freiburg.informatik.proglang.HelloWorldServlet</class>
    </servlet-mapping>
  </servlet-mappings>
</mini-servlets-config>
```