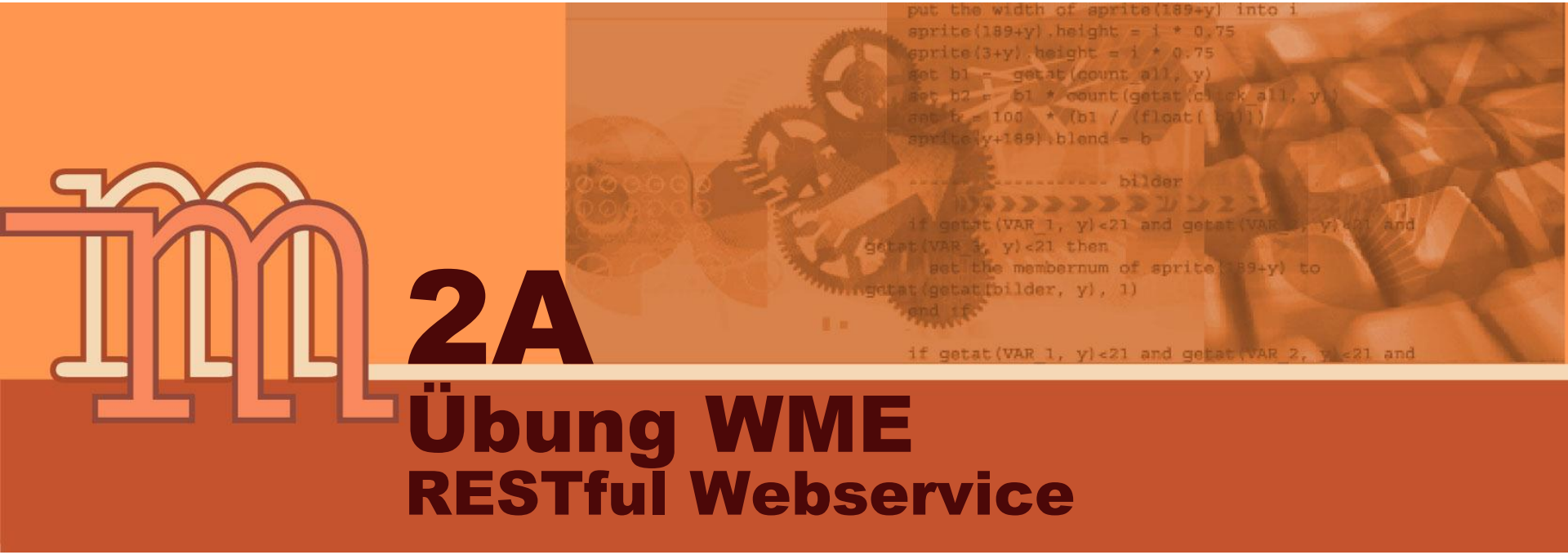


2A
Übung WME
RESTful Webservice

Dipl.-Medieninf. Matthias Niederhausen

Technische Universität Dresden
Fakultät Informatik
Institut für Software- und Multimediatechnik
Lehrstuhl für Multimediatechnik

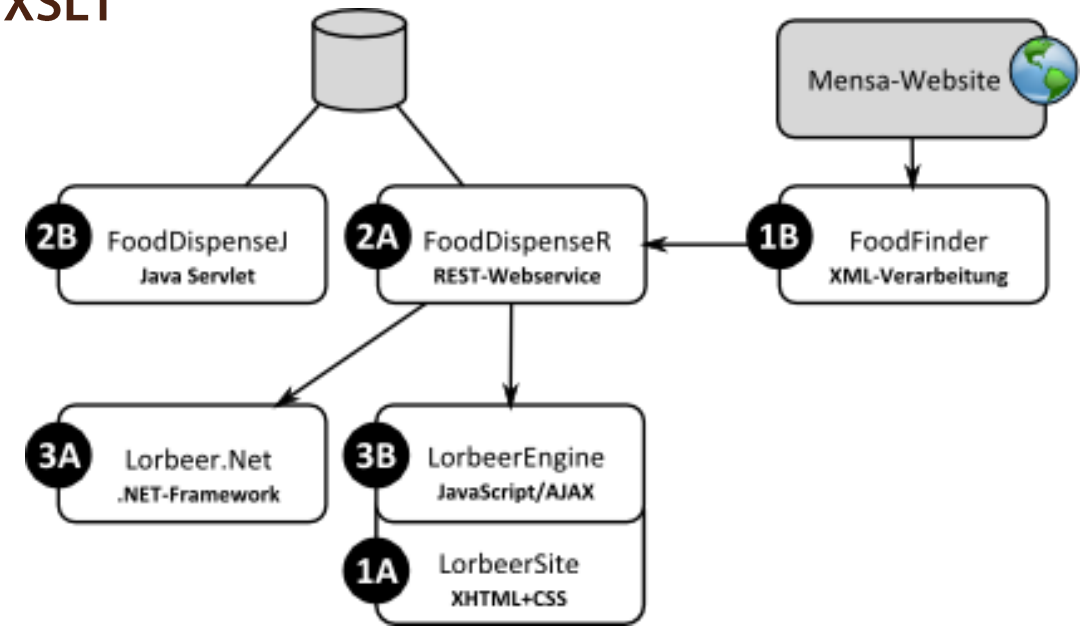
Gliederung

- Organisatorisches
  - Terminplan
- Aufgabenstellung
  - Hinweise zur Umsetzung
- Einstieg ins Thema
- Links

Organisatorisches: Ablauf der Übung

- Terminplan
  - 15.11.11 Übungseinführung 2A (Pflicht)
  - 22.10.11 Konsultation (Teilnahme freiwillig)
  - 29.11.11 Konsultation (Teilnahme freiwillig)
  - 06.12.11 Konsultation (Teilnahme freiwillig)
  - 13.12.11 Konsultation (Teilnahme freiwillig)
  - 19.12.11 Abgabe der Ergebnisse (bis spätestens 13:00 Uhr)
  - 20.12.11 Übungseinführung 3A (Pflicht)
  - …
  - 31.01.12 Abschlussveranstaltung und Präsentation (Pflicht)
- Details siehe Übungskalender

- Themenkomplex 1
  - 1A) XHTML und CSS
  - 1B) XML-Verarbeitung und XSLT
- Themenkomplex 2
  - 2A) RESTful Webservice
  - 2B) Java Servlets
- Themenkomplex 3
  - 3A) .NET/Silverlight
  - 3B) JavaScript und AJAX



- **Aufgabe 2A: Bereitstellen eines REST-Webservices für das Portal „Lorbeerblatt“**
  - Das Portal Lorbeerblatt soll Studenten eine Plattform bieten, auf der sie sich über das aktuelle Mensaessen austauschen können. Die Benutzer sollen Kommentare, Bewertungen und eigene Bilder der verschiedenen Speisen einstellen können. Um leicht Clientanwendungen für das Portal zu entwickeln, besteht Bedarf an einer öffentlichen REST-Schnittstelle.
  - Musterlösung: Live-Webservice
    - WADL-Beschreibung
    - Beispiel: Liste von Mensas


```
▼<dishes>
 ▼<dish>
    <id>1</id>
    <mensaId>1</mensaId>
    <date>2011-11-11T00:00:00+01:00</date>
    <name>Ungarischer Rindergulasch mit Makkaroni und
    <priceStudent>189</priceStudent>
    <priceEmployee>339</priceEmployee>
   ▼<sourceUrl>
      http://www.studentenwerk-dresden.de/mensen/speis
    </sourceUrl>
   ▼<photoUrl>
      http://bilderspeiseplan.studentenwerk-dresden.de
    </photoUrl>
  </dish>
 ▼<dish>
    <id>2</id>
    <mensaId>1</mensaId>
    <date>2011-11-11T00:00:00+01:00</date>
   ▼<name>
      Putensteak Hawaii mit Pommes frites, Pommes Twis
    </name>
    <priceStudent>247</priceStudent>
    <priceEmployee>397</priceEmployee>
   ▼<sourceUrl>
      http://www.studentenwerk-dresden.de/mensen/speis
    </sourceUrl>
   ▼<photoUrl>
      http://bilderspeiseplan.studentenwerk-dresden.de
    </photoUrl>
  </dish>
 ▼<dish>
    <id>3</id>
```

Aufgabenstellung im Detail

- Entwicklung eines Webservice für Lorbeerblatt
  - maschinenverarbeitbare Repräsentation von Informationen: standardisierte Struktur
  - Funktionen
    - Liste von Mensas (Ausgabe)
    - Gerichte einer Mensa (Ausgabe)
    - Benutzerfotos zu einem Gericht (Ausgabe, Erstellen, Löschen)
    - Kommentare zu einem Gericht (Ausgabe, Erstellen, Löschen)
    - Bewertungen zu einem Gericht (Ausgabe, Erstellen, Löschen)
  - Generierung sinnvoller Status- und Fehlercodes

Aufgabenstellung im Detail

- Randbedingungen
  - Fehlerbehandlung: ungültige Referenzen, fehlende und falsche Daten
  - ohne weitere Konfiguration unmittelbar lauffähig
  - Initialisierung mit Daten aus dem FoodFinder
  - Statuscodes, Fehlercodes und MIME-Types korrekt benutzen
  - Lizenzrechte bei verwendeten Fremdinhalten beachten!
- Abgabe
  - Quelldateien, evtl. als ZIP-Datei
  - lauffähige WAR-Anwendung

- REST-Paradigma
  - steht für Representional State Transfer
    - sämtliche Aktionen sind zustandslos, d.h. können immer einzeln durchgeführt werden -> Anfragen müssen sämtliche erforderlichen Informationen beinhalten
  - URLs repräsentieren Ressourcen
  - Ressourcen können verschieden repräsentiert werden
  - nutzt Möglichkeiten des HTTP-Standards aus
    - HTTP GET/POST/PUT/DELETE
    - Content-Types

- REST in Jersey
  - Basiskonfiguration in web.xml: Jersey-Servlet registrieren

```
<filter>
    <filter-name>Jersey REST Service</filter-name>
    <filter-class>com.sun.jersey.spi.container.servlet.ServletContainer</filter-class>
    <init-param>
        <param-name>com.sun.jersey.config.property.packages</param-name>
        <param-value>de.mmt.lorbeerblatt.webservice</param-value>
    </init-param>
</filter>
<filter-mapping>
    <filter-name>Jersey REST Service</filter-name>
    <url-pattern>/*</url-pattern>
</filter-mapping>
```

  - Annotationen an Klassen/Methoden markieren diese als Ressourcen/Aktionen:

```
@GET
@Path("count")
@Produces(MediaType.TEXT_PLAIN)
public String getCount()
```

  - Annotationen an Datenklassen erlauben automatische Serialisierung in XML und JSON (via JAXB-Technologie)

```
@XmlRootElement
public class Comment
```

  - automatische Bereitstellung einer WADL-Servicebeschreibung (Web Application Description Language) unter /application.wadl

Zum Weiterlesen…

- Jersey Libraries
  http://jersey.java.net/nonav/documentation/latest/user-guide.html#chapter_deps
- REST with Java using Jersey (Tutorial)
  http://www.vogella.de/articles/REST/article.html
- WADL-Generierung mit Jersey
  http://wikis.sun.com/display/Jersey/WADL
- JavaDoc-Tags für WADL-Generierung mit Jersey
  http://wikis.sun.com/display/Jersey/SupportedJavadocTagsForExtendedWADL