J2EE Courses

Eine Einführung in die Java 2 Enterprise Edition

Teil 1 - Überblick und Grundlagen

Inhalt

- Was ist die J2EE
- Probleme bei der Enterprise-Softwareentwicklung
- Ein Überblick über die J2EE-Komponenten
- Was ist ein Application Server
- Installation und Benutzung von JBoss
  - Installation
  - Bereitstellen der Anwendungsumgebung
  - Anwendungsdeployment
- JNDI als Rückgrat einer J2EE Anwendung
- Eine Beispiel-Webanwendung

Was ist die J2EE

- Ein Standard
- Fokus auf Geschäftsanwendungen
- Ein Komponentenmodell
- Lösung für immer wiederkehrende Probleme

Probleme bei der Enterprise-Softwareentwicklung

- Transaktionen
- Remoting
- Security
- Persistenz
- Pooling
- Skalierbarkeit
- Legacy-Integration

Probleme bei der Enterprise-Softwareentwicklung

- Transaktionen
  - Demarkierung von Transaktionen
  - Verteilte Transaktionen
- Remoting
  - Zugriff auf Server-Komponenten über Netzwerke
- Security
  - Absicherung von Server-Komponenten
  - Anmeldung am Client
  - Zugriff auf Anmeldedaten im Server-Code

Probleme bei der Enterprise-Softwareentwicklung

- Persistenz
  - Dauerhafte Speicherung von Objekten (Datenbank)
  - Unabhängigkeit von der Datenbank
  - Überbrücken der O/R Lücke
  - Caching
- Pooling
  - Verwalten von Pools (Datenbank-Verbindungen, Threads, Objekte)
- Skalierbarkeit
  - Anpassung an hohe Lasten
  - Clustering
- Legacy-Integration

Bestandteile der J2EE

- Servlets, JSP (Java Server Pages)
- JDBC
- JavaMail
- JNDI (Java Naming and Directory Interface)
- JMS (Java Message Service)
- EJB (Enterprise Java Beans)
- JAAS (Java Authentication and Authorization Service)
- JCA (J2EE Connector Architecture)
- JTA/JTS (Java Transaction API/Services)
- JAXP/JAXR/JAX-RPC/SAAJ (Webservices)

Bestandteile der J2EE

- Servlets, JSPs
  - Webanwendungen, Generierung von Websites
- JDBC
  - Datenbankzugriffe
- JavaMail
  - Versenden und Empfangen von Mails
- JNDI
  - Schnittstelle zu Namens- und Verzeichnissdiensten
  - z.B. LDAP, DNS, Dateisystem
  - Möglichkeit zum "Nachschlagen" der Serverdienste

Bestandteile der J2EE

- JMS
  - Schnittstelle zu Messaging-Systemen
  - Möglichkeit zum Versenden und Empfangen von Nachrichten
  - Asynchron
- EJB
  - Serverseitige Geschäftslogik-Komponenten
  - Definiertes Interface
  - Schnittstelle für die angebotenen Dienste:
    - Transaktionen
    - Security
    - Remoting
    - Persistenz

Bestandteile der J2EE

- JAAS
  - Authentifizierung
  - Zugriffskontrolle, Rollenverwaltung
- JCA
  - Definierte Schnittstelle zu anderen EIS
  - Integration des EIS in die Containerdienste
- JTA/JTS
  - Spezifikation für Transaktions-Manager
  - Schnittstelle für Transaktionen

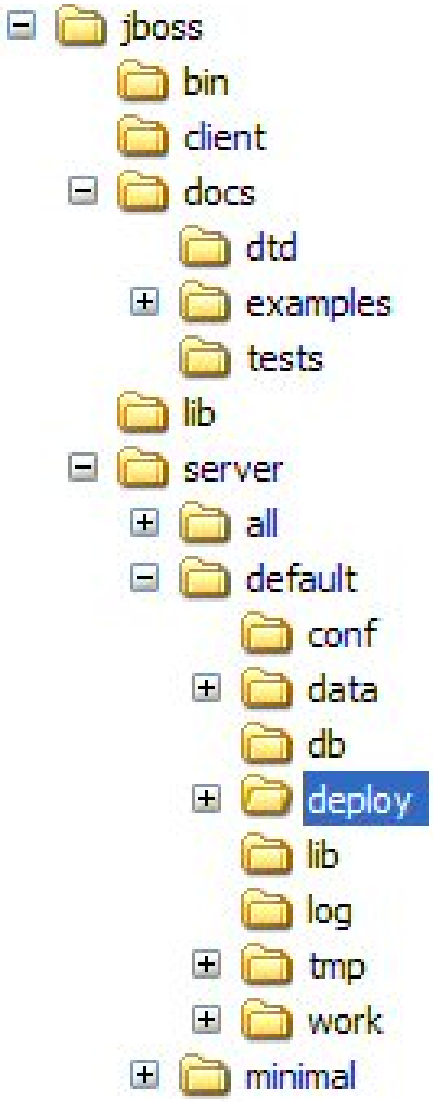
JNDI - Unser Backbone

```
Properties env = new Properties();
env.setProperty(
    "java.naming.factory.initial",
    "com.sun.jndi.dns.DnsContextFactory");

DirContext ictx = new InitialDirContext(env);
Attributes attrs1 = ictx.getAttributes(
    "www.heise.de",
    new String[] {"A"});

System.out.println("Heise IP: " + attrs1.get("a").get(0));
```

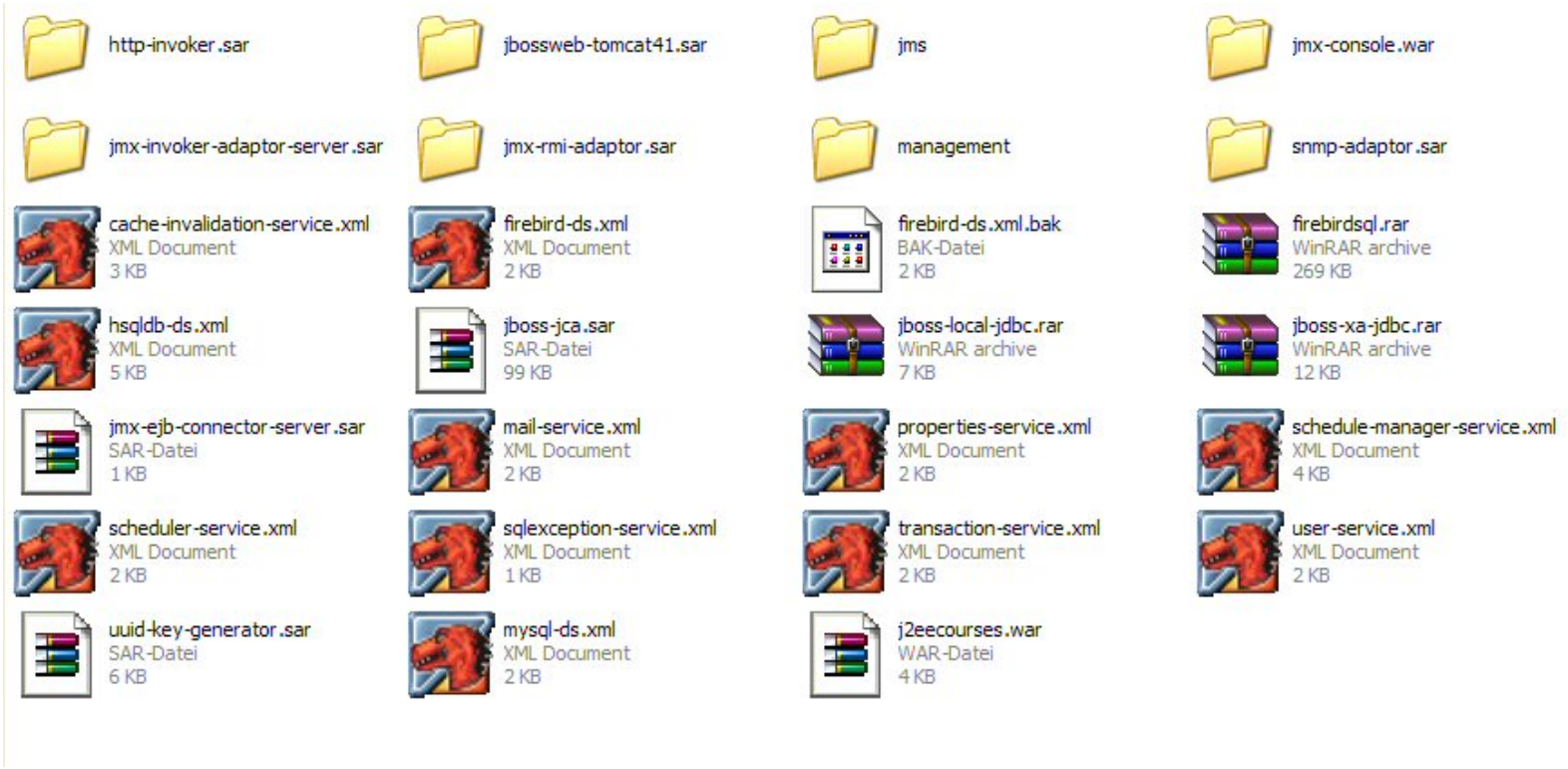
Ran an den Application Server - JBoss

- Open Source - LGPL
- http://www.jboss.org
- Sehr modular

Struktur von JBoss



Struktur von JBoss



Installation von neuen Services

```
<datasources>
  <local-tx-datasource>
    <jndi-name>NewDS</jndi-name>
    <connection-url>jdbc:mysql://localhost:3306/test</connection-u
    <driver-class>org.gjt.mm.mysql.Driver</driver-class>
    <user-name>root</user-name>
    <password></password>
  </local-tx-datasource>
</datasources>
```

Nutzung der Services - Eine "Webanwendung"

```
public class SimpleServlet extends HttpServlet {

    public void doGet(
        HttpServletRequest request,
        HttpServletResponse response) {

        response.setContentType("text/html");
        DataSource ds = lookupDS();
        List books = queryDB(ds);
        printBooks(books, response.getWriter());
    }
```

Nutzung der Services - JNDI-Lookup

```
private DataSource lookupDS() {
    try {
        InitialContext context = new InitialContext();
        return (DataSource) context.lookup("java:/NewDS");
    } catch (NamingException e) {
        throw new RuntimeException(e);
    }
}
```

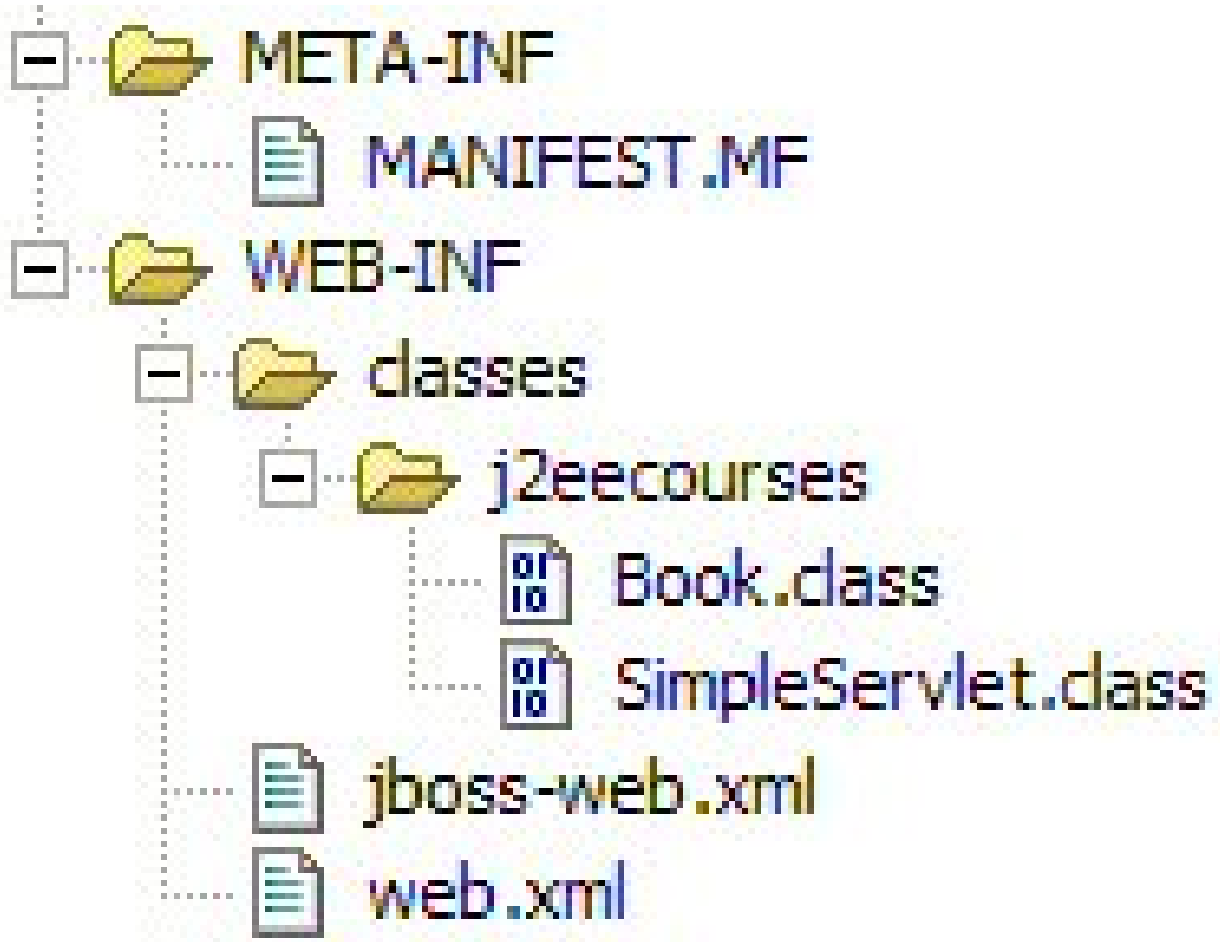
Nutzung der Services - DB-Zugriff

```
private List queryDB(DataSource ds) {
    try {
        Connection conn = ds.getConnection();
        Statement stmnt = conn.createStatement();
        ResultSet rs = stmnt.executeQuery("SELECT * FROM books");
        List result = new ArrayList();
        while (rs.next()) {
            result.add(new Book(rs.getString("title")));
        }
        stmnt.close(); conn.close();
        return result;
    } catch (SQLException e) {
        throw new RuntimeException(e);
    }
}
```

Nutzung der Services - Ausgabe

```
public static final String HTML_HEADER =
    "<html><head><title>Tolle Seite</title></head><body>";
public static final String HTML_BODY = "</body></html>";

private void printBooks(List books, PrintWriter writer) {
    writer.println(HTML_HEADER);
    for (Iterator iter = books.iterator(); iter.hasNext();) {
        Book book = (Book) iter.next();
        writer.println(book.getTitle() + "<br>");
    }
    writer.println(HTML_BODY);
}
```

Deployment der Webanwendung



Deployment der Webanwendung - web.xml

```
<web-app>
    <servlet>
        <servlet-name>servlet</servlet-name>
        <servlet-class>j2eecourses.SimpleServlet</servlet-class>
    </servlet>
    <servlet-mapping>
        <servlet-name>servlet</servlet-name>
        <url-pattern>/myServlet</url-pattern>
    </servlet-mapping>
</web-app>
```

Deployment der Webanwendung - jboss-web.xml

```
<jboss-web>
</jboss-web>
```

Installation des JavaMail-Services

```
<server>
  <mbean code="org.jboss.mail.MailService" name="jboss:service=Mail">
    <attribute name="JNDIName">java:/Mail</attribute>
    <attribute name="User">gloegl</attribute>
    <attribute name="Password">++++++</attribute>
    <attribute name="Configuration">
       <configuration>
          <property name="mail.store.protocol" value="pop3"/>
          <property name="mail.transport.protocol" value="smtp"/>
          <property name="mail.user" value="gloegl"/>
          <property name="mail.pop3.host" value="mail.fmi.uni-passau.de"/>
          <property name="mail.smtp.host" value="mail.fmi.uni-passau.de"/>
          <property name="mail.from" value="gloegl@oktiron.net"/>
          <property name="mail.debug" value="false"/>
       </configuration>
    </attribute>
  </mbean>
</server>
```

JavaMail - Ran an die Mails

```
public class MailServlet extends HttpServlet {

    public void doGet(
        HttpServletRequest request,
        HttpServletResponse response) {

        Session sess = lookupSession();
        sendMail(sess);
        response.setContentType("text/html");
        response.getWriter().println("Message sent");
    }
```

JavaMail - Ran an die Mails

```
private Session lookupSession() {
    try {
        InitialContext context = new InitialContext();
        Session sess = (Session) context.lookup("java:/Mail");
        return sess;
    } catch (NamingException e) {
        throw new RuntimeException(e);
    }
}
```

JavaMail - Ran an die Mails

```
private void sendMail(Session sess) {
    try {
        Message message = new MimeMessage(sess);
        message.setText("Hallo von J2EE");
        Address address =
            new InternetAddress("gloegl@fmi.uni-passau.de");
        message.setRecipient(Message.RecipientType.TO, address);
        message.setFrom(address);
        Transport.send(message);
    } catch (Exception e) {
        throw new RuntimeException(e);
    }
}
```

Abstraktion des JNDI - der ENC

```
private DataSource lookupDS() {
        InitialContext context = new InitialContext();
        return (DataSource) context.lookup("java:/NewDS");
}

private DataSource lookupDS() {
        InitialContext context = new InitialContext();
        return (DataSource)
            context.lookup("java:comp/env/jdbc/KrasseDS");
}
```

ENC - An der Schnittstelle Deployer/Developer

In der web.xml ...

```
... </servlet-mapping>
    <resource-ref>
        <description>Our new Datasource</description>
        <res-ref-name>jdbc/KrasseDS</res-ref-name>
        <res-type>javax.sql.DataSource</res-type>
        <res-auth>Container</res-auth>
    </resource-ref>
</web-app>
```

ENC - An der Schnittstelle Deployer/Developer

In der jboss-web.xml ...

```
<jboss-web>
    <resource-ref>
        <res-ref-name>jdbc/KrasseDS</res-ref-name>
        <res-type>javax.sql.DataSource</res-type>
        <jndi-name>java:/NewDS</jndi-name>
    </resource-ref>
</jboss-web>
```

Nochmal ENC - für Mail

```
private Session lookupSession() {
        InitialContext context = new InitialContext();
        Session sess = (Session) context.lookup("java:/Mail");
        return sess;
}

private Session lookupSession() {
        InitialContext context = new InitialContext();
        Session sess = (Session)
            context.lookup("java:comp/env/mail/Mailversender");
        return sess;
}
```

Nochmal ENC - für Mail

In der web.xml ...

```
... </resource-ref>
    <resource-ref>
        <description>Unser Mailsender</description>
        <res-ref-name>mail/Mailversender</res-ref-name>
        <res-type>javax.mail.Session</res-type>
        <res-auth>Container</res-auth>
    </resource-ref>
</web-app>
```

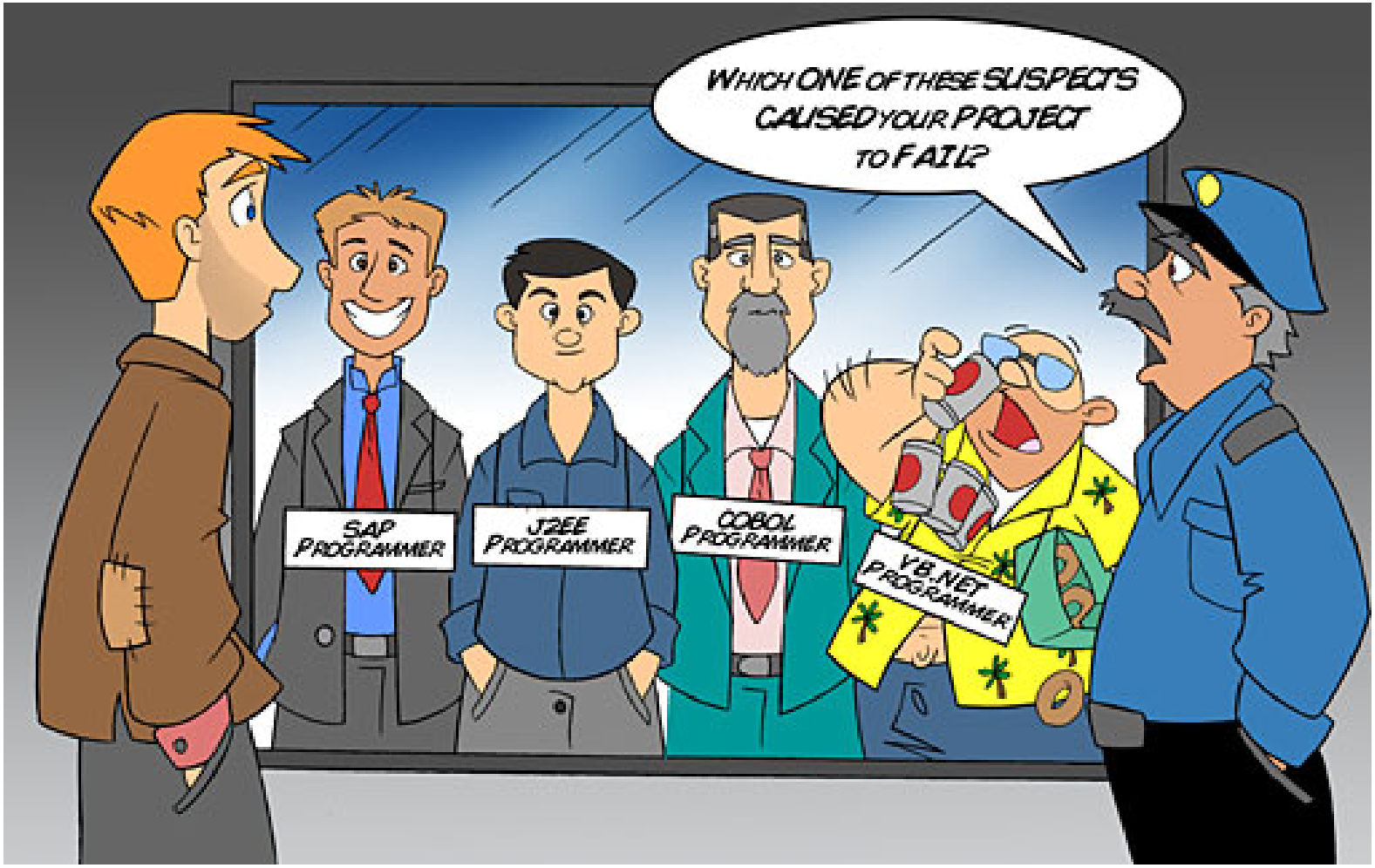
Nochmal ENC - für Mail

In der jboss-web.xml ...

```
<jboss-web>
    <resource-ref>
        ...
    </resource-ref>
    <resource-ref>
        <res-ref-name>mail/Mailversender</res-ref-name>
        <res-type>javax.mail.Session</res-type>
        <jndi-name>java:/Mail</jndi-name>
    </resource-ref>
</jboss-web>
```

Fragen?

Ende


WHICH ONE OF THESE SUSPECTS CAUSED YOUR PROJECT TO FAIL?
SAP PROGRAMMER
J2EE PROGRAMMER
COBOL PROGRAMMER
VB.NET PROGRAMMER