Web Applications
Lecture 6

Java Libraries for Programming Servlets

Bartosz Walter
<Bartek.Walter@man.poznan.pl>

Testing the Web

- Logging: log4j
- Multiparts: commons-fileupload
- Databases: JDBC
- Form validation: FormProc
- Authentication: Form-based

Log4j

- **Ligthweight**
  - quiet pass: 5-50 ns
  - logging pass: ca 1 ms
- **Easily configurable**
  - Loggers vs. Appenders
  - log4.properties

Loggers

- **A category in the logging space**
- **Named, case-sensitive**
- **Hierarchical**
  - pl.poznan.put is a parent of pl.poznan.put.student
  - pl.poznan.put is a child of **pl.poznan**
  - **unnamed root category**

Loggers - examples

- Logger root = Logger.getRootLogger();
- Logger aLog = Logger.getLogger("a.b.c");
- Logger bLog = Logger.getLogger("a.b.c");
- Logger cLog = Logger.getLogger (AClass.class);

Logging levels

- ordered levels of logging
  - debug, info, warn, error, fatal
  - logger.debug(msg);
- inherited from ancestor by default
- enabled request
  - its level is higher or equal of the logger's level

Appenders

- represent destinations for logging requests
- files, console, remote servers, NT Event Loggers, syslog
- one logger = many appenders
- each enabled request is forwarded to all appenders attached to a given logger
- additive inheritance of appenders

Layouts

- **PatternLayout**
  - %r [%t] %-5p %c - %m%n
  - 176 [main] INFO org.foo.Bar - Located nearest gas station.

Example

- **PropertyConfigurator**

  PropertyConfigurator.configure("log4j.properties");

```
# Set root logger level to DEBUG and its only appender to A1.
log4j.rootLogger=DEBUG, A1

# A1 is set to be a ConsoleAppender.
log4j.appender.A1=org.apache.log4j.ConsoleAppender

# A1 uses PatternLayout.
log4j.appender.A1.layout=org.apache.log4j.PatternLayout
log4j.appender.A1.layout.ConversionPattern=%-4r [%t] %-5p %c %x - %
m%n
```

Handling multipart requests

- RFC 1867 (multipart/form-data)
- http://jakarta.apache.org/commons/fileupload

```
boolean isMultipart = FileUpload.isMultipartContent(request);
// Create a new file upload handler
DiskFileUpload upload = new DiskFileUpload();
// Set upload parameters
upload.setSizeThreshold(yourMaxMemorySize);
upload.setSizeMax(yourMaxRequestSize);
upload.setRepositoryPath(yourTempDirectory);
// Parse the request
List /* FileItem */ items = upload.parseRequest(request);
```

Handling multipart requests (cont.)

```
Iterator iter = items.iterator();
while (iter.hasNext()) {
    FileItem item = (FileItem) iter.next();
    if (item.isFormField()) {
        String fieldName = item.getFieldName();
        String fileName = item.getName();
        String contentType = item.getContentType();
        boolean isInMemory = item.isInMemory();
        long sizeInBytes = item.getSize();     } else {

        item.write(new File("uploadedFile"));
        byte[] data = item.get();
    }
}
```

Basics of JDBC

- **Unique, generic protocol for accessing databases**
- **Basic objects:**
  - Connection
  - Statement, PreparedStatement
  - ResultSet

Connection

- represents the connection to database
- costly in creation
- created by
  - DriverManager – deprecated
  - DataSource – pooled, flexible, fast

Configuring DataSource in Tomcat

```
<Resource name="jdbc/TestDB" auth="Container"
    type="javax.sql.DataSource"/>
<ResourceParams name="jdbc/TestDB">
<parameter>
  <name>factory</name>
  <value>org.apache.commons.dbcp.BasicDataSourceFactory</value>
</parameter>
<parameter>
  <name>url</name>
  <value>jdbc:mysql://localhost:3306/javatest?autoReconnect=true</value>
</parameter>
</ResourceParams>
```

server.xml

Configuring DataSource in Tomcat (cont.)

```
<resource-ref>
  <description>DB Connection</description>
  <res-ref-name>jdbc/TestDB</res-ref-name>
  <res-type>javax.sql.DataSource</res-type>
  <res-auth>Container</res-auth>
</resource-ref>
```

web.xml

Configuring DataSource in Tomcat (cont.)

```
<resource-ref>
  <description>DB Connection</description>
  <res-ref-name>jdbc/TestDB</res-ref-name>
  <res-type>javax.sql.DataSource</res-type>
  <res-auth>Container</res-auth>
</resource-ref>
```

web.xml

Configuring DataSource in Tomcat (cont.)

```
public void init() {
  try {
    Context ctx = new InitialContext();
    if (ctx == null)
      throw new Exception("Boom - No Context");
    DataSource ds = (DataSource) ctx.lookup("java:comp/env/jdbc/TestDB");
    if (ds != null) {
      Connection conn = ds.getConnection();
      if (conn != null) {
        foo = "Got Connection " + conn.toString();
      }
      conn.close();
    }
  } catch (Exception e) { e.printStackTrace(); }
}
```

MyServlet.java

Statements

Statements represent commands to be executed by DB

```
sql = "select nazwisko from pracownicy"
   + " where imie = ? and rok_zatrudnienia = ?"

Statement stm = conn.createStatement();

PreparedStatement pstm = conn.prepareStatement(sql);
pstm.setString(1, "Bartosz");
pstm.setInt(2, 2000);
```

```
NAZWISKO
==========
WALTER
```

ResultSets

Represents cursors with data (forward, backward, random access)

```
ResultSet rs = stm.executeQuery(sql);
ResultSet rs = pstm.executeQuery();
while (rs.next()) {
      String nazwisko = rs.getString(1);
      System.out.println("Nazwisko=" + nazwisko);
}

stm.executeUpdate(sql);
pstm.executeUpdate()
```

Transactions

Transaction isolation:

```
conn.setTransactionIsolation(mode);
```

```
TRANSACTION_NONE, _READ_COMMITED, _READ_UNCOMMITED..
```

Auto commit:

```
conn.setAutoCommit(boolean);
```

Commit & rollback

```
conn.commit();
```

```
conn.rollback()
```

FormProc

- server-side form validation
- http://formproc.sf.net

formproc.xml

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<config>
<validator-map type="expression"
    classname="org.formproc.validation.REValidator"/>
<validator-map type="rule"
    classname="org.formproc.validation.RuleValidator"/>
<validator-map type="group"
    classname="org.formproc.validation.ValidatorGroup"/>
<shared-validator name="required">
   <validator type="expression">
      <pattern>.{1,}</pattern>
      <error lang="en">The field is required.</error>
      <error lang="fr">Exigée</error>
    </validator>
</shared-validator>
<form name="test" path="example-form.xml" monitor="true"
    loader="com.anthonyeden.lib.resource.ClassPathResourceLoader" />
</config>
```

formproc.xml - validators

```
<shared-validator name="age">

  <validator type="group">

    <validator type="rule">
      <rule>org.formproc.example.IsIntRule</rule>
      <error>Valid number required</error>
      <error lang="fr">L'âge valide a exigé.</error>
    </validator>

    <validator type="expression">
     <pattern>[1-9][0-9]*</pattern>
     <error>Age must be 1 or greater</error>
    </validator>

  </validator>

</shared-validator>
```

form.xml

```
<form>
<name>test</name>
<storer classname="org.formproc.store.MapStorer"/>
  <element name="name">
    <validator type="shared" name="required">
      <error>Name field required.</error>
    </validator>
    <message>Required</message>
  </element>
  <element name="birthdate">
     <validator type="expression">
      <pattern>\d\d/\d\d/\d\d\d\d</pattern>
      <error>Date must be in format MM/dd/yyyy</error>
    </validator>
    <message lang="en">Valid date (MM/dd/yyyy) required.</message>
     <converter classname="org.formproc.conversion.DateConverter">
       <parse type="custom" pattern="MM/dd/yyyy"/>
    </converter>
</element>
</form>
```

Form-based authentication

- j_username
- j_password
- j_security_check

```
<login-config>
    <auth-method>FORM</auth-method>
    <form-login-config>
       <form-login-page>/admin/main/login.html</form-login-page>
       <form-error-page>/admin/main/login_failed.html</form-error-page>
    </form-login-config>
 </login-config>
```

http://www.onjava.com/pub/a/onjava/2002/06/12/form.html

Realms

```
<Realm className="org.apache.catalina.realm.JDBCRealm"
    debug="99"
    driverName="oracle.jdbc.driver.OracleDriver"
    connectionURL="jdbc:oracle:thin:@{IPAddress}:{Port}:{Servicename}"
    connectionName="{DB Username}"
    connectionPassword="{Password}"
    userTable="users"
    userNameCol="username"
    userCredCol="password"
    userRoleTable="user_roles"
    roleNameCol="rolename" />
```

server.xml

Q & A