

Umstellung der WebApp auf Derby

Gründe für die Umstellung der Datenbank von Google AppEngine auf Apache Derby

- AppEngine läuft bei Google in der Cloud und hat daher Einschränkungen:
 - Max. 10 Anwendungen pro Nutzer
 - Max. Traffic 1 GB pro Tag, max. 56 MB pro Minute
 - Max. 6,5 CPU-Stunden pro Tag pro Anwendung
 - Abhängigkeit von anderem Anbieter, Server nicht an der HAW
 - Usw.
- Apache Derby ist ein leichtgewichtiges auf Java basierendes relationales Datenbank-Management-System und bietet uns Vorteile:
 - Sehr klein und leicht konfigurierbar
 - Keine Traffic- oder Zeit-Einschränkungen
 - Lässt sich im Hudson starten und macht die WebApp testbarer
 - Unabhängigkeit von Google, da der Server an der HAW laufen kann

Austausch der Pakete der WebApp

Die WebApp basierte bereits auf Google Web Toolkit und der DataNucleus Access Platform, die mit GWT mitgeliefert wird und eine Schnittstelle für die AppEngine bietet.

Da auch Apache Derby mit der DataNucleus Access Platform funktioniert, haben wir uns dazu entschlossen, die von Google mitgelieferte Version durch eine mit Apache Derby kompatible Version zu ersetzen. Durch die Beibehaltung der DataNucleus Access Platform muss nicht der komplette Code angepasst werden. Zu diesem Zweck haben wir alle alten JARs entfernt und die von uns benötigten neuen JARs hinzugefügt.

Pakete von Apache Derby

derby.jar: Ist die Engine von Derby. Ohne diese JAR ist keine Derby Datenbank möglich.

derbynet.jar: Ist der Derby Network Server und referenziert das Engine JAR.

derbyclient.jar: Ist der Derby Network Client und erlaubt es, Verbindungen mit einem Derby Network Server herzustellen.

derbytools.jar: Ist ein optionales Paket, das das „ij“ Tool enthält, mit dem man Zugriff auf die Datenbank aus der Konsole hat.

Pakete der DataNucleus Access Platform

DataNucleus ist ein Open-Source-Persistenz-Framework zur Speicherung von Java-Objekten in Datenbanken. Es unterstützt Java Data Objects (JDO) und die Java Persistence API (JPA). Die WebApp benutzt JDO.

datanucleus-core-2.1.3.jar: Ist der Kern der Plattform.

datanucleus-enhancer-2.1.2.jar: Damit Klassen mit DataNucleus persistiert werden können, müssen sie die Interfaces „Persistence Capable“ oder „Persistence Aware“ (JDO Interfaces) implementieren. Außerdem muss vor der Benutzung dieser Klassen ein Bytecode Enhancement stattfinden. Dieses

Enhancement fügt den Klassen die für diese Interfaces nötigen Methoden und Felder hinzu. Dadurch muss man selbst keine Änderungen an den eigenen Klassen vornehmen.

datanucleus-rdbms-2.1.2.jar: Dieses JAR stellt die nötige Funktionalität zur Verfügung, um relationale Datenbank-Management-Systeme (RDBMS) mit DataNucleus zu benutzen. Da Apache Derby zu den RDBMS gehört, wird es von uns benötigt.

asm-3.3.jar: Ist eine Abhängigkeit vom Enhancer. Dabei handelt es sich um ein Java Bytecode Manipulation Framework.

geronimo-jta_1.1_spec-1.1.jar: Ist eine Abhängigkeit des Cores. Es stellt die Java Transaction API (JTA) zur Verfügung.

jdo-api-3.0.jar: Ist eine Abhängigkeit des Cores. Es stellt Java Database Objects (JDO) zu Verfügung.

transaction-api-1.1.jar: Ist eine Abhängigkeit von der JDO API für die Java Transaction API (JTA).

Pakete des Google Web Toolkits

Das Google Web Toolkit (GWT) stellt bei uns die Oberfläche für den Benutzer dar.

gwt-user-2.0.4.jar: Dieses Paket enthält die Klassen für das Entwickeln der Anwendung auf Client-Seite. Dieses Paket wird nach dem Kompilieren nicht mehr benötigt.

gwt-servlet-2.0.4.jar: Dieses Paket enthält die Klassen für die Server-Seite für u.a. RPC und muss auf dem Server liegen.

gwt-dev-2.0.4.jar: Dieses Paket enthält unter anderem den GWT-Compiler.

Pakete für das Connection Pooling (nur benötigt bei Verbindung zu DB über das Netzwerk)

Wird keine lokale Derby Datenbank-Datei verwendet, sondern über das Netzwerk mit einer Datenbank verbunden (das gilt auch für localhost), müssen noch folgende Pakete vorhanden sein.

datanucleus-connectionpool-2.0.3.jar: Stellt Connection Pooling zu Verfügung.

commons-pool-1.2.jar, commons-dbcp-1.2.1.jar, commons-collections-3.1.jar: Diese Pakete sind Abhängigkeiten von datanucleus-connectionpool bzw. untereinander abhängig.

Anpassung der Datenbankanbindung von DataNucleus

Im Source-Verzeichnis der WebApp gibt es in dem Ordner META-INF eine Datei namens jdoconfig.xml. In dieser ist die Konfiguration für JDO gespeichert.

Diese Datei war durch die vorherige Verwendung von DataNucleus durch die AppEngine bereits vorhanden und musste von uns nur noch angepasst werden.

Alte Konfiguration:

```
<?xml version="1.0" encoding="utf-8"?>
<jdoconfig xmlns="http://java.sun.com/xml/ns/jdo/jdoconfig"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"

  xsi:noNamespaceSchemaLocation="http://java.sun.com/xml/ns/jdo/jdoconfig">
```

```

    <persistence-manager-factory name="transactions-optional">
        <property name="javax.jdo.PersistenceManagerFactoryClass"
value="org.datanucleus.store.appengine.jdo.DatastoreJDOPersistenceManagerFa
ctory"/>
        <property name="javax.jdo.option.ConnectionURL" value="appengine"/>
        <property name="javax.jdo.option.NontransactionalRead"
value="true"/>
        <property name="javax.jdo.option.NontransactionalWrite"
value="true"/>
        <property name="javax.jdo.option.RetainValues" value="true"/>
        <property name="datanucleus.appengine.autoCreateDatastoreTxns"
value="true"/>
    </persistence-manager-factory>
</jdoconfig>

```

Neue Konfiguration:

```

<?xml version="1.0" encoding="utf-8"?>
<jdoconfig xmlns="http://java.sun.com/xml/ns/jdo/jdoconfig"
    xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"

xsi:noNamespaceSchemaLocation="http://java.sun.com/xml/ns/jdo/jdoconfig">

    <persistence-manager-factory name="transactions-optional">
        <property name="javax.jdo.PersistenceManagerFactoryClass"
            value="org.datanucleus.jdo.JDOPersistenceManagerFactory"/>
        <property name="javax.jdo.option.NontransactionalRead"
value="true"/>
        <property name="javax.jdo.option.NontransactionalWrite"
value="true"/>
        <property name="javax.jdo.option.RetainValues" value="true"/>
        <property name="datanucleus.ConnectionDriverName"
value="org.apache.derby.jdbc.EmbeddedDriver"/>
        <property name="datanucleus.ConnectionURL"
value="jdbc:derby:TimadorusDB;create=true" />
        <property name="datanucleus.storeManagerType" value="rdbms" />
        <property name="datanucleus.ConnectionUserName" value="webapp" />
        <property name="datanucleus.ConnectionPassword" value="ty32re8pu" />
        <property name="datanucleus.autoCreateSchema" value="true"/>
    </persistence-manager-factory>
</jdoconfig>

```

Durchgeführte Änderungen

Wie man sehen kann mussten wir die „javax.jdo.PersistenceManagerFactoryClass“ von der speziellen AppEngine PM-Factory-Class

„org.datanucleus.store.appengine.jdo.DatastoreJDOPersistenceManagerFactory“ auf

„org.datanucleus.jdo.JDOPersistenceManagerFactory“ umstellen.

Außerdem wurde der „datanucleus.ConnectionDriverName“ mit dem Wert

„org.apache.derby.jdbc.EmbeddedDriver“ hinzugefügt, da wir den Embedded Apache Derby Treiber für die Datenbankverbindung benutzen.

Die „datanucleus.ConnectionURL“ wurde von „appengine“ auf

„jdbc:derby:TimadorusDB;create=true“ geändert. Dies bedeutet, dass wir mit dem JDBC eine Verbindung zu der Derby-Datenbank „TimadorusDB“ herstellen, diese wird falls noch nicht vorhanden erstellt. Außerdem wird sie auch gleich mit dem Programm mitgestartet.

Als „datanucleus.storeManagerType“ mussten wir „rdbms“ angeben, weil es sich bei Derby um ein relationales Datenbank-Management-System handelt.

„datanucleus.ConnectionUserName“ und „datanucleus.ConnectionPassword“ geben den Benutzernamen der Datenbank und deren Passwort an.

„datanucleus.autoCreateSchema“ mit dem Wert „true“ sorgt dafür, dass das Datenbankschema automatisch erstellt wird.

Quellen:

http://db.apache.org/derby/papers/DerbyTut/install_software.html

<http://db.apache.org/derby/papers/DerbyClientSpec.html>

http://de.wikipedia.org/wiki/Apache_Derby

<http://www.datanucleus.org/>

http://www.datanucleus.org/products/accessplatform/bytecode_enhancement.html

<http://de.wikipedia.org/wiki/DataNucleus>

<http://asm.ow2.org/>

http://www.jarvana.com/jarvana/view/org/apache/geronimo/specs/geronimo-jta_1.1_spec/1.1.1/geronimo-jta_1.1_spec-1.1.1-javadoc.jar!/index.html

<http://www.datanucleus.org/products/accessplatform/jdo/pmf.html>

<http://markmail.org/message/52zrbjjajelqdbdh>