HIBERNATE OGM
Hibernate OGM Architecture
Hibernate OGM Architecture

Data Persistence
How is Data Persisted

• Abstraction between the application object model and the persistent data model
• Persist data as basic types
• Keep the notion of primary key to address an entity
• Keep the notion of foreign key to link two entities
UserOne
- userid
- name
- address

Address
- street
- city
- pin

Tbl_userone
- userid_pk
- name
- street
- city
- pin
Key identifying an entity instance is composed of:

- the table name
- the primary key column name
- the primary key value

<table>
<thead>
<tr>
<th>Key</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>{userid=1,</td>
<td>Name=&quot;Gerad&quot;,</td>
</tr>
<tr>
<td>Tbl_userone,userid,1</td>
<td>address.street=&quot;18th main&quot;,</td>
</tr>
<tr>
<td></td>
<td>address.city=&quot;Bangalore&quot;,</td>
</tr>
<tr>
<td></td>
<td>Address.pin= 5760041,</td>
</tr>
<tr>
<td></td>
<td>}</td>
</tr>
</tbody>
</table>
Key for association data is composed of:

- the table name
- the column name representing the foreign key to the entity we come from
- the column value representing the foreign key to the entity we come from
<table>
<thead>
<tr>
<th>Key</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>tbl_userone,userid_pk,1</td>
<td>{userid_pk=1,name=“Gerad”}</td>
</tr>
<tr>
<td>tbl_userone,userid_pk,2</td>
<td>{userid_pk=2,name=“Samuel”}</td>
</tr>
<tr>
<td>tbl_address,addid_pk,11</td>
<td>{addid_pk=11,street=“18th main”}</td>
</tr>
<tr>
<td>tbl_address,addid_pk,12</td>
<td>{addid_pk=12,street=“20th main”}</td>
</tr>
<tr>
<td>tbl_userone_address,userid_fk,1</td>
<td>{{userid_fk=1,addid_fk=11},</td>
</tr>
<tr>
<td></td>
<td>{userid_fk=1,addid_fk=12}}</td>
</tr>
<tr>
<td>tbl_userone_address,userid_fk,2</td>
<td>{{userid_fk=2,addid_fk=13},</td>
</tr>
<tr>
<td></td>
<td>{userid_fk=3,addid_fk=14}}</td>
</tr>
<tr>
<td>tbl_userone_address,addid_fk,11</td>
<td>{{userid_fk=1,addid_fk=11},</td>
</tr>
<tr>
<td></td>
<td>{userid_fk=4,addid_fk=11}}</td>
</tr>
<tr>
<td>tbl_userone_address,addid_fk,12</td>
<td>{{userid_fk=1,addid_fk=12},</td>
</tr>
<tr>
<td></td>
<td>{userid_fk=5,addid_fk=12}}</td>
</tr>
</tbody>
</table>
Querying Data
Hibernate OGM Configuration
<?xml version="1.0"?>
<persistence
    xmlns="http://java.sun.com/xml/ns/persistence"
    xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
    xml/ns/persistence/persistence_2_0.xsd"
    version="2.0">
    <persistence-unit name="HibOGMPU" transaction-type="JTA">
        <provider>org.hibernate.ogm.jpa.HibernateOgmPersistence</provider>
        <properties>
            <property name="hibernate.transaction.jta.platform" value="org.hibernate.service.jta.platform.internal.jbossstandalone.jtaplatform" />
            <property name="hibernate.ogm.datastore.provider" value="mongo" />
        </properties>
    </persistence-unit>
</persistence>
MongoDB Installation and Configuration
MongoDB Installation

- Download the open source mongodb from
  http://www.mongodb.org/downloads (version 2.0.4) and unzip in a directory
- Create Directory c:\data\db for data files (or specify alternate path using
  - - dbpath argument)

- Starting MongoDB from the Command Prompt
  - From the unzipped directory – go to bin
  - Execute mongod.exe
    (or) mongod --dbpath D:\data\db
    (since the default C:\data\db is changed to D:\data\db – it has to be specified explicitly)
MongoDB Installation

a. http://localhost:28017/ - to check if it is working as in the next slide
b. Press <Ctrl –c> on the shell running the server, to stop the service
c. Open another command prompt → go to bin directory → execute mongo.exe (mongodb JavaScript shell)
Hibernate OGM Configuration
For MongoDB
## Hibernate OGM – MongoDB

<table>
<thead>
<tr>
<th>Configuration Key</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>hibernate.ogm.datastore.provider</td>
<td>mongodb</td>
</tr>
<tr>
<td>hibernate.ogm.mongodb.host</td>
<td>default value is 127.0.0.1</td>
</tr>
<tr>
<td>hibernate.ogm.mongodb.port</td>
<td>default value is 27017</td>
</tr>
<tr>
<td>hibernate.ogm.mongodb.database</td>
<td>no default value</td>
</tr>
<tr>
<td>hibernate.ogm.mongodb.username</td>
<td>Mongo username</td>
</tr>
<tr>
<td>hibernate.ogm.mongodb.password</td>
<td>Mongo password</td>
</tr>
<tr>
<td>hibernate.ogm.mongodb.safe</td>
<td>default value is true.</td>
</tr>
<tr>
<td>hibernate.ogm.mongodb.connection.connection_timeout</td>
<td>expressed in milliseconds. The default value is 5000.</td>
</tr>
<tr>
<td>hibernate.ogm.mongodb.association.s.store</td>
<td>GLOBAL_COLLECTION/COLLECTION/IN_ENTITY</td>
</tr>
</tbody>
</table>
Entity Classes
@Entity
public class UserOne {
  @Id
  private String userid;
  private String name;
  @Column(name="desc")
  private String description;

  //getters, setters ...
}
Mongo JSON Document

{

"_id" : "5678-9999-0123-8553",
"name" : "Gerad",
"desc" : "Male aged 32",
}


Primary Keys and Identifiers
Composite Primary Key

```java
@Entity
public class Employee {
  @EmbeddedId
  private IDClass id;
  //getters, setters ...
}

@Embeddable
public class IDClass implements Serializable {
  private String empid;
  private String deptid;
  //getters, setters ...
}
```
JSON Representation

{ 
   "_id": {
      "empid": "11234",
      "deptid": "777"
   }
}
Association Strategies

- IN_ENTITY (default)
- GLOBAL_COLLECTION
- COLLECTION
Association Relationship Between Entities
Generators and DataTypes

JPA id generators are supported:
• IDENTITY
• SEQUENCE
• TABLE
• AUTO

Java Types Supported:
• Boolean
• Byte
• Calendar (this may change)
• Class (this may change)
Generators and DataTypes

Java Types Supported:
- Date (this may change)
- Double
- Integer
- Long
- Byte Array
- String

Types Mapped as String:
- BigDecimal
- BigInteger
- Url
- UUID
Supported Associations

- @OneToOne
- @OneToMany
- @ManyToOne
- @ManyToMany
Querying MongoDB
Query Support

- Hibernate Search FullTextQueries
- Native query technology of MongoDB.
- While using JPQL:
  - Limitations are:
    - No join, aggregation, or other relational operations
    - Use Hibernate Session API
    - The entities and properties are indexed by Hibernate Search
- Criteria API not implemented yet
Q&A
Thank you