iBATIS, Hibernate, and JPA: Which is right for you?

Object-relational mapping solutions compared

By K. L. Nitin, Ananya S., Mahalakshmi K., and S. Sangeetha, JavaWorld.com, 07/15/08

SQLMap.xml

The other XML file is SQLMap.xml, which is in practice named after the table to which it relates. There can be any number of such files in a single application. This file is the place where domain objects are mapped to SQL statements. This descriptor uses parameter maps to map the inputs to the statements and the result maps for mapping SQL `ResultSet`s. This file also contains the queries. Therefore, to change the queries, you need to change the XML, not your application's Java code. The mapping is done by using the actual SQL statements that will interact with the database. Thus, using SQL provides greater flexibility to the developer and makes iBATIS easy to understand to anyone with SQL programming experience.

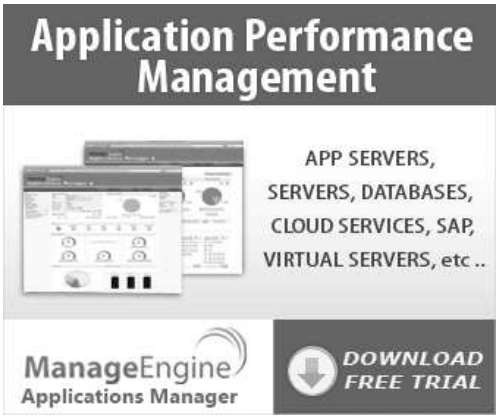
The SQLMap.xml file that defines the SQL statements to perform CRUD operations on the EMPLOYEE table is shown in Listing 2.

Listing 2. SQLMap.xml for operations on EMPLOYEE

```
<sqlMap namespace="Employee">
  <typeAlias alias="Employee" type="com.sample.Employee"/>
  <resultMap id="EmpResult" class="Employee">
    <result property="id" column="emp_id"/>
    <result property="firstName" column="emp_firstname"/>
    <result property="lastName" column="emp_lastname"/>
   </resultMap>
  <!-- Select all data from the table using the result map f
  <select id="selectAllEmps" resultMap="EmpResult">
    select * from EMPLOYEE
  </select>
<!-- Select the data from the table based on the id. -->
<select id="selectEmpById" parameterClass="int" resultClass=
 <select emp_id as id,emp_firstname as firstName,     emp_la
</select>
<!--  insert the data into the table -->
<insert id="insertEmp" parameterClass="Employee">
    insert into EMPLOYEE (
      emp_id,
       emp_firstname,
       emp_lastname)
    values (
      #id#, #firstName# , #lastName# )
  </insert>
<!-- update the Employee record based on the id -->
  <update id="updateEmp" parameterClass="Employee">
```

JW's Most Read

Scala on their minds -- bloggers debate Scala's complexity
Simplify the data access layer with Spring 3.1 and Java generics
Bitcoin for beginners, Part 2: Bitcoin as a technology and network
Why Yammer dropped Scala for Java
Engine Yard vs Heroku: Ruby clouds (for Java too)



Featured White Papers

Observation Driven Testing: What else is your code doing?
Software Agitation: Your Own Personal Code Reviewer
Early & Often: Avoiding Security Flaws with CI, High Coverage

Newsletter sign-up View all newsletters

Enterprise Java Newsletter
Stay up to date on the latest tutorials and Java community news posted on JavaWorld

Subscribe

Sponsored Links

Slow Apps? Overloaded DBMS? - FREE Download
FREE Download Feb 14 - March 7th Eliminate bottlenecks & maximize performance ScaleOut StateServer®

Sponsored Links

Optimize with a SATA RAID Storage Solution
Range of capacities as low as $1250 per TB. Ideal if you currently rely on servers/disks/JBODs

```
    where
      emp_id = #id#
  </update>
<!-- delete the Employee record based on the given id -->
    <delete id="deleteEmp" parameterClass="int">
    delete from EMPLOYEE where emp_id = #id#
  </delete>
</sqlMap>
```

In Listing 2, the `typeAlias` tag is used to represent type aliases, so you can avoid typing the full class name every time it would otherwise appear. It contains the `resultMap` tag, which describes the mapping between the columns returned from a query and the properties of the class represented by the `Employee` class. The `resultMap` is optional and it isn't required if the columns in the table (or aliases) match the properties of the class exactly. This `resultMap` tag is followed by a series of queries. SQLMap.xml can contain any number of queries. All the select, insert, update, and delete statements are written within their respective tags. Every statement is named using the `id` attribute.

The output from a select query can be mapped to a `resultMap` or to a result class that is a JavaBean. The aliases in the queries should match the properties of the target result class (that is, the JavaBean). The `parameterClass` attribute is used to specify the JavaBean whose properties are the inputs. Any parameters in the hash symbol are the properties of the JavaBean.

Loading the descriptor files to your Java application

After you have completed the entire configuration and mapped in both the XML files, SQLMapConfig.xml needs to be loaded by the Java application. The first step is to load the SQLMap.xml configuration file that was created earlier. To do this, you would use the `com.ibatis.common.resources.Resources` class, which is included with the iBATIS framework, as shown in Listing 3.

Listing 3. Loading SQLMap.xml

```
private static SqlMapClient sqlMapper;
...
 try {
      Reader reader = Resources.getResourceAsReader("com/mydomain/data/SqlMapConfig.xml");
      sqlMapper = SqlMapClientBuilder.buildSqlMapClient(reader);
      reader.close();
    } catch (IOException e) {
      // Fail fast.
      throw new RuntimeException("Something bad happened while building the SqlMapClient instance." + e, e);
    }
  }
```

The `SqlMapClient` class is used for working with `SQLMap`s. It allows you to run mapped statements like select, insert, update, and so on. The `SqlMapClient` object is thread safe; hence, one object is enough. This makes it a good candidate to be a static member. This object is created by reading a single SQLMapConfig.xml file. The iBATIS framework provides the `Resources.getResourceAsReader()` utility, with which you can read the SQLMapConfig.xml file. Thus, by using this instance of the `SQLMap`, you can access an object from the database -- in this case, an `Employee` object.

To invoke the operations on the EMPLOYEE table, different methods are provided on the `SQLMap`, such as `queryForList()`, `queryForObject()`, `insert()`, `update()`, and `queryForMap()`, among others. The `queryForList()` method, shown in Listing 4, returns a list of `Employee` objects.

Listing 4. queryForList()

```
sqlMapper.queryForList("selectAllEmps");
```

Similarly, you would use the `queryForObject()` method when only one row was returned as a result of the query. Both methods take the statement name as the parameter.

Corresponding methods are available for performing insert, update, and delete operations, as shown in Listing 5. These methods take both the statement name declared in the SQLMap.xml file and the `Employee` object as the input.

Listing 5. Insert, update, and delete operations

```
sqlMapper.insert("insertEmp", emp);
sqlMapper.update("updateEmp", emp);
```

When to use iBATIS

iBATIS is best used when you need complete control of the SQL. It is also useful when the SQL queries need to be fine-tuned. iBATIS should not be used when you have full control over both the application and the database design, because in such cases the application could be modified to suit the database, or vice versa. In such situations, you could build a fully object-relational application, and other ORM tools are preferable. As iBATIS is more SQL-centric, it is generally referred to as *inverted* -- fully ORM tools generate SQL, whereas iBATIS uses SQL directly. iBATIS is also inappropriate for non-relational databases, because such databases do not support transactions and other key features that iBATIS uses.

< Prev 1 2 3 4 5 6 7 Next >

Print Feedback Like

Like

Comments powered by Our Commenting Network | Policies | Privacy

Add New Comment Login

Showing 20 of 24 comments Sort by newest first

Ahamed Firzadh

Hats off gr8 article.... Thanks for the detailed explanation which is also easy on a newbie :)

11/21/2011 06:14 AM Like Reply

Anonymous

IBatis, Hibernate, JPA
Thank you for putting this together. It provides a very good introduction for me into ORM and persistence. Much appreciated.

06/26/2010 04:39 PM Like Reply

Anonymous

Thanks!!!
Thanks a lot. Very useful article.

05/30/2010 05:34 AM Like Reply

Anonymous

Buen inicio
Muchas gracias por el artículo, me dió lo necesario para empezar :)

05/22/2010 11:06 PM Like Reply

Anonymous

good article.
Thanks

05/19/2010 02:14 AM Like Reply

Anonymous

Thanks!
Great article!

Nice comparison
I was searching for a comparison between Hibernate and JPA. Thanks for the effort.

-Sunil.
http://techmindviews.blogspot....

03/26/2010 04:58 AM Like Reply

Anonymous

Hibernate Vs IBatis
The article is awesome for the beginners. Thanks for the valuable information that you have put on the Java World.

03/24/2010 02:21 AM Like Reply

Anonymous

great article and excellent explanation.

03/13/2010 09:21 AM Like Reply

Anonymous

Question
The article mentions that JPA is non-portable. But woudn't it be possible to use JPA in Rails using JRuby?

01/30/2010 08:51 AM Like Reply

Anonymous

Yes,It'a grate
Can i say that,If we want to separated Business layer and database layer that time,we can use this method....

01/22/2010 02:03 AM Like Reply

Anonymous

Hibernate Vs IBatis
IBatis is considered to be the way if you want to have full control over your queries + dont want to clutter your code with sqls.

I would contest that by saying hibernate provides the exact same feature with the concept of named queries if you ever needed that feature.

The only reason i think people tend to tilt towards IBatis is that it does not overwhelm you with tons of features which first time users find a bit daunting and get scared away.

01/21/2010 01:04 PM Like Reply

Anonymous

This is it.
I was searching for database layer solutions. This article helped me a lot. I am enlightened. Thank you so much.

12/06/2009 09:45 PM Like Reply

Anonymous

Control over queries...
"Hibernate provides a complete ORM solution, but offers you no control over the queries"

Dude, Session.createSQLQuery lets you use native sql, if that is not "full control over queries" then iBATIS does not have it either! PD: RTFM

10/08/2009 02:36 PM Like Reply

Anonymous

Hibernate also lets you custom code sql mappings for its ORM use just like ibatis does too. so if you want/need to code your own sql you can!
you can also configure it to use stored procs too.

Anonymous

Question for the authors
Excellent article, I can see a great research work behind.

I'm just nearly convinced to use iBATIS, but I'd like to double check with you my scenario. I'm using an existing database with loads of functions and stored procedures that I'll have to use in my application, that's why I choose iBATIS. On the other hand, I'll have control over both the Java and the DB; but no time to migrate the Stored Procedures to Java.

Performance will be a very important issue, as it's an application which will manage loads of users and heavy binary and xml stored data. I can tune the queries on iBATIS, but the cache mechanism provided by Hibernate may be a better option?

Thanks a lot!

10/06/2009 08:21 AM Like Reply

Anonymous

iBatis is my choice - from experience...
There's no silver bullet, so use the technology you are most familiar with or standard in the environment. BUT, having used iBatis and hibernate extensively, I always default to iBatis. I have spent too many hours trying to figure out what hibernate is doing - as performance can be an issue. I rarely work on a project that needs to access more than one database type (mysql, oracle, sybase, ms sql server, postgres etc).

I have just finished a .Net desktop project with iBatis as the db access layer. It works a treat!

A large financial organisation performed extensive research on java tools for their global architecture team ... and iBatis came out on top for the db access layer.

So if its a new project, give iBatis a try.

09/02/2009 12:26 AM Like Reply

Anonymous

Hibernate's Simplicity is "Good"? Bah...
How on Earth did you conclude that Hibernate's simplicity is 'good'? Let me tell you why it should be downgraded to 'average' if not 'poor'. The official Hibernate book is huge, and the number of times we've had to research how to solve a problem on our project is ridiculous. Ramp up time is long for new teammates. The reason why it seems community support is so good for Hibernate is because we see so many posts of people have problems with the technology.

08/03/2009 11:34 PM Like Reply

Anonymous

it is a very helpful article and also has a very simple manner of telling. I was not bored while reading.
Thank you so much..

07/17/2009 12:39 AM Like Reply

Anonymous

Hibarnate seems to be the best
Hibernate seems to be winner

07/09/2009 05:28 AM 1 Like Like Reply

Subscribe by email RSS

Load more comments

Archived Discussions (Read only)

Forum migration complete *By Athen*

Forum migration update *By Athen*

Learn more about the three technologies discussed in this article from the project homepages:

- Hibernate
- iBATIS
- The Java Persistence API

"Get started with Hibernate" (Christian Bauer and Gavin King, JavaWorld, October 2004) is a short introduction to Hibernate written by its creator, Gavin King. *Excerpted from Hibernate in Action; Manning 2004.*)

Java Persistence with Hibernate (Christian Bauer and Gavin King; Manning, November 2006) is the updated second edition of *Hibernate in Action*. Also see iBATIS in Action (Clinton Begin, Brandon Goodin, and Larry Meadors, 2007).

For broader coverage of Java persistence solutions including JDBC, OpenJPA, and pureQuery, see Persistence in the Enterprise: A Guide to Persistence Technologies (Roland Barcia, Geoffrey Hambrick, Kyle Brown, Robert Peterson, Kulvir Singh Bhogal; IBM Press, May 2008).

Ted Neward introduces the so-called object-relational impedance mismatch in his blog post "The Vietnam of computer science."

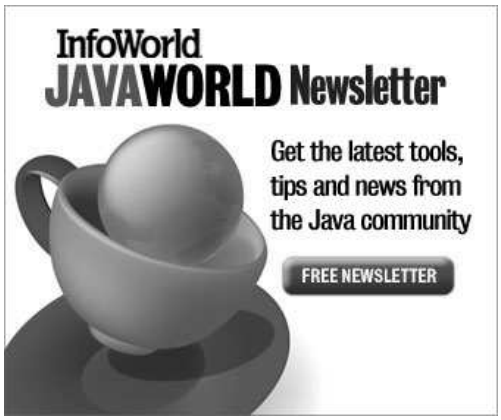
"Flexible reporting with JasperReports and iBatis" (Scott Monahan, JavaWorld, December 2007) is a hands-on introduction to the iBatis Data Mapper framework.

"Understanding the Java Persistence API" (Aditi Das, JavaWorld, January 2008) is a two-part introduction to Java-platform persistence with OpenJPA.

Java-source.net lists a roundup of open source persistence frameworks for Java.

Visit the JavaWorld Java Enterprise Edition research center for more articles about enterprise data management and Java persistence solutions.

Also see Network World's IT Buyer's Guides: Side-by-side comparison of hundreds of products in over 70 categories.



Sponsored Links

Web hosting reviews for you to choose a better host
ManageEngine: End-to-End Java Performance Management. Download Product Now!

Copyright, 2006-2011 Infoworld, Inc. All rights reserved.