

Objektově-relační mapování s nástrojem iBatis / MyBatis

Jaroslav Kačer, IBA CZ
jaroslav.kacer@ibacz.eu

Moderní trendy v softwarovém inženýrství
Plzeň, 8.12.2010



1. **Úvod do ORM**
2. Úvod do MyBatisu
3. Školní příklad
4. Konfigurace
5. Píšeme mappery
6. Co se nevešlo jinam
7. Příklad z praxe
8. Porovnání s JPA



- **V paměti** jsou data uložena jako **objekty** – instance tříd
 - 1 záznam = 1 objekt
 - 1 položka = 1 políčko objektu
 - Vztahy = reference mezi objekty
- **V relační databázi** jsou data uložena **v tabulkách**
 - 1 záznam = 1 řádek v tabulce
 - 1 položka = 1 sloupeček ve vybraném řádku
 - Vztahy = odkazy z tabulek do jiných tabulek pomocí cizích klíčů
- ORM:
 - Jak provést uložení objektů z paměti do databáze?
 - Jak načíst data z databáze do objektů?



1. JDBC


- Java Database Connectivity
- Nejstarší Java standard pro práci s DB
- Spousta nevýhod: vše jen v Java kódu, čistě orientované na SQL, nepřehledné zápisy SQL, snadno vede k chybám a nepřehlednému kódu

2. JPA


- Java Persistence API
- Poměrně nový standard, „ukradený“ z nástroje Hibernate
- Anotace objektů → entity, které lze jednoduše ukládat/načítat
- Existují různé implementace: Hibernate, TopLink, OpenJPA, ...

3. Vlastní řešení

- Např. iBatis/MyBatis

1. Úvod do ORM
-  2. **Úvod do MyBatisu**
3. Školní příklad
4. Konfigurace
5. Píšeme mappery
6. Co se nevešlo jinde
7. Příklad z praxe
8. Porovnání s JPA

- Existuje pro Javu a .NET
- Používá „čisté“ SQL napsané programátorem
- Základní stavební jednotkou je tzv. **mapper**
 - Kolekce SQL výrazů zapsaných ve XML
 - Java interface
 - Operace mapperů se volají jako Java metody nad instancemi mapperů, mají vstupy a výstupy
- Nepoužívá se návrhový vzor DAO
- Operace lze volat jen v rámci tzv. **session**
 - Otevřít
 - Provést operace
 - Zavřít



od Nathan Maves <nathan.maves@gmail.com> ☆
 předmět: **Re: will nested sessions work? I.e. will a child session commit be auto-ignored?**
 komu: mybatis-user@googlegroups.com

I am going to give you my personal/professional opinion so take it for what it is worth.

DAO's are dead. Part of the rewrite of MyBatis was to get rid of the idea of a DAO and our old DAO framework. A mapper is really your dao.

DAO's when used should be a single unit of work. If you find yourself calling one from the other you have done something wrong and rethink it. If it is hard to unit test rethink it 😊

Use a service layer class when you need to combine multiple units of work together. Given your example I would create a AccountService class that would use both the AccountMapper and the PlayerMapper. Hand the transaction there.

DI is your friend. USE IT

Nathan



- Hlavní autor Clinton Begin
- Počátek v roce 2002 jako součást kryptografického softwaru Secrets
 - iBatis od slova „abatis“
- Rok 2002:
 - Microsoft publikuje zprávu, že .NET je 10x rychlejší a 4x produktivnější než J2EE
 - Reakce: JPetStore aplikace, používající iBatis Database Layer (SQL Maps + DAO)
- Poté v roce 2004 darováno (včetně názvu) organizaci Apache Software Foundation → [Apache iBatis](#)
- V červnu 2010 migrace na Google Code
 - Nutnost změny jména → [MyBatis](#)
 - Stejná licence (Apache License 2.0)
 - Stejná jména, 100% kompatibilní
 - Java tým: 8 lidí, .NET tým: 3 lidi

1. Úvod do ORM
2. Úvod do MyBatisu
3. Školní příklad
4. Konfigurace
5. Píšeme mappery
6. Co se nevešlo jinam
7. Příklad z praxe
8. Porovnání s JPA



- Triviální konzolová aplikace
- Připojuje se k lokální DB Oracle
- 2 tabulky: knihy a jejich autoři
- Výpisy knih a autorů, přidávání

Connection Name	Local Oracle - TSI 2010		
Username	TSI2010		
Password	*****		
<input checked="" type="checkbox"/> Save Password			
Oracle		Access	
Role	default	<input type="checkbox"/> OS Authentication	
Connection Type	Basic	<input type="checkbox"/> Kerberos Authentication	
		<input type="checkbox"/> Proxy Connection	
Hostname	localhost		
Port	1521		
<input checked="" type="radio"/> SID	xe		
<input type="radio"/> Service name			

```

TSI 2010 MyBatis Demo
Q ... Quit
1 ... Print all books
2 ... Print all authors
3 ... Print an author's books
4 ... Add a new author
5 ... Add a new book
Your choice: 1
{ID: '1', Title: 'Učebnice jazyka C', ISBN: '1234567890', Author's ID: '1'}
{ID: '2', Title: 'Učebnice jazyka Java', ISBN: '0987654321', Author's ID: '1'}
{ID: '3', Title: 'Java - GUI a čeština', ISBN: '1212131313', Author's ID: '1'}
{ID: '4', Title: 'Java - Bohatství knihoven', ISBN: '3456543566', Author's ID: '1'}
{ID: '5', Title: 'Java a XML', ISBN: '8654578854', Author's ID: '1'}
{ID: '6', Title: 'XSLT 2.0 a SVG', ISBN: '3335457777', Author's ID: '1'}
{ID: '7', Title: 'LINUX - Internet server', ISBN: '8667786656', Author's ID: '2'}
{ID: '8', Title: 'IP v6', ISBN: '0988878777', Author's ID: '2'}
{ID: '9', Title: 'CSS kaskádové styly', ISBN: '5654555454', Author's ID: '3'}
{ID: '10', Title: 'CSS - Hotová řešení', ISBN: '7878778787', Author's ID: '3'}
{ID: '11', Title: 'OOP v C++', ISBN: '3493848484', Author's ID: '4'}
{ID: '12', Title: 'Třídy a objekty v C++', ISBN: '2332000922', Author's ID: '4'}
{ID: '13', Title: 'Umění programování 1', ISBN: '0939939393', Author's ID: '5'}
{ID: '14', Title: 'Umění programování 2', ISBN: '0990393382', Author's ID: '5'}
Your choice:
    
```

The Package Explorer shows the following structure:

- TsiMyBatisDemo
 - src/main/java
 - biz.kacer.vyuka.tsi.tsi2010.tsimybatisdemo
 - applogic
 - AppLogic.java
 - persistence
 - data
 - Author.java
 - Book.java
 - mappers
 - AuthorMapper.java
 - BookMapper.java
 - MyBatisSessionFactory.java
 - ui
 - CommandDispatcher.java
 - Main.java
 - src/test/java
 - src/main/resources
 - biz.kacer.vyuka.tsi.tsi2010.tsimybatisdemo.pe
 - mappers
 - AuthorMapper.xml
 - BookMapper.xml
 - SqlMapConfig.xml
 - commons-logging.properties
 - log4j.dtd
 - log4j.xml
 - JRE System Library [JavaSE-1.6]
 - Maven Dependencies
 - junit-4.8.1.jar - C:\Users\Duck\.m2\repository
 - log4j-1.2.16.jar - C:\Users\Duck\.m2\repository
 - commons-logging-1.1.1.jar - C:\Users\Duck\.m2\repository
 - ojdbc14-10.2.0.3.0.jar - C:\Users\Duck\.m2\repository
 - mybatis-3.0.3.jar - C:\Users\Duck\.m2\repository
 - src
 - target
 - pom.xml

Mappery

Konfigurace

Ovladač databáze Oracle


MyBatis

AUTHORS x						
Columns Data Constraints Grants Statistics Triggers Dependencies Details Partitions Indexes SQL						
Actions...						
COLUMN_NAME	DATA_TYPE	NULLABLE	DATA_DEFAULT	COLUMN_ID	COMMENTS	
ID	NUMBER	No	(null)	1	(null)	
NAME	VARCHAR2 (32 BYTE)	No				
SURNAME	VARCHAR2 (32 BYTE)	No				

BOOKS x						
Columns Data Constraints Grants Statistics Triggers Dependencies Details Partitions Indexes SQL						
Actions...						
COLUMN_NAME	DATA_TYPE	NULLABLE	DATA_DEFAULT	COLUMN_ID	COMMENTS	
ID	NUMBER	No	(null)	1	(null)	
TITLE	VARCHAR2 (64 BYTE)	No	(null)	2	(null)	
ISBN	VARCHAR2 (16 BYTE)	Yes	(null)	3	(null)	
AUTHOR	NUMBER	No	(null)	4	(null)	

BOOKS x				
Columns Data Constraints Grants Statistics Triggers Dependencies Details Partitions Indexes SQL				
Sort... Filter:				
ID	TITLE	ISBN	AUTHOR	
1	1 Učebnice jazyka C	1234567890	1	
2	2 Učebnice jazyka Java	0987654321	1	
3	3 Java - GUI a čeština	1212131313	1	
4	4 Java - Bohatství knihoven	3456543566	1	
5	5 Java a XML	8654578854	1	
6	6 XSLT 2.0 a SVG	3335457777	1	
7	7 LINUX - Internet server	8667786656	2	
8	8 IP v6	0988878777	2	
9	9 CSS kaskádové styly	5654555454	3	
10	10 CSS - Hotová řešení	7878778787	3	
11	11 OOP v C++	3493848484	4	
12	12 Třídy a objekty v C++	2332000922	4	
13	13 Umění programování 1	0939939393	5	
14	14 Umění programování 2	0990393382	5	

AUTHORS x			
Columns Data Constraints Grants Statistics			
Sort... Filter:			
ID	NAME	SURNAME	
1	1 Pavel	Herout	
2	2 Pavel	Satrapa	
3	3 Petr	Staniček	
4	4 Stanislav	Racek	
5	5 Donald	Knuth	

1. Úvod do ORM
2. Úvod do MyBatisu
3. Školní příklad
-  4. **Konfigurace**
5. Píšeme mappery
6. Co se nevešlo jina
7. Příklad z praxe
8. Porovnání s JPA



- Konfigurace je základem každé aplikace
- Statická, uložena ve XML souboru, nejlépe uvnitř JAR souboru
- Dle DTD
- Obsahuje (vše volitelně):
 - **Definici vlastností** – Lze použít dále v konfiguraci
 - **Globální nastavení** – Ovlivňují chování
 - **Typové aliasy** – Méně psaní
 - Definice **typových konverzí**
 - **Object factories** – Pro nestandardní vytváření objektů
 - **Pluginy** – Interceptory, vyvolají se v přesně daný čas
 - **Popis prostředí** – Kam se aplikace připojí
 - **Seznam mapperů** – Odkazy na extra XML soubory s definicí mapperů

```

1<?xml version="1.0" encoding="UTF-8"?>
2
3<!DOCTYPE configuration PUBLIC "-//ibatis.apache.org/DTD Config 3.0//EN" "http://ibatis.apache.org/dtd/ibatis-3-config.dtd">
4
5<configuration>
6

```

- Celkem 9 vlastností, platných skrz aplikaci
- Cache, lazy loading, vykonávání dotazů, timeout, ...

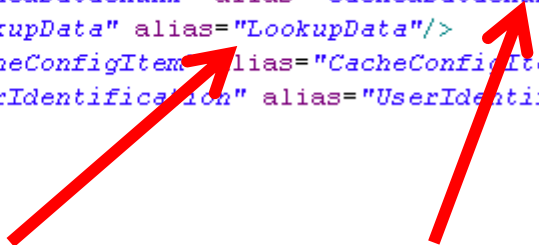
```
<!--
Global settings
-->
<settings>
  <!-- Reuse prepared statements -->
  <setting name="defaultExecutorType" value="SIMPLE"/>
  <!-- ATS jobs need a global timeout -->
  <setting name="defaultStatementTimeout" value="${Setting.DefaultStatementTimeout}"/>
</settings>
```

Vlastnost – Může být uvedena přímo v konfiguraci
anebo dodána dynamicky při inicializaci, např. načtena
z externího souboru.

- Pokud chceme k typu přistupovat pod jiným jménem
- Typicky ke zkrácení jména – vypuštění jména balíku

```
<typeAlias type="com.ibm.learning.careerdashboard.dto.cache.CachedDataChunk" alias="CachedDataChunk"/>
<typeAlias type="com.ibm.learning.careerdashboard.dto.cache.LookupData" alias="LookupData"/>
<typeAlias type="com.ibm.learning.careerdashboard.dto.cache.CacheConfigItem" alias="CacheConfigItem"/>
<typeAlias type="com.ibm.learning.careerdashboard.dto.cache.UserIdentification" alias="UserIdentification"/>
```

```
<select id="findDataChunkInsertedAfter" parameterType="LookupData" resultType="CachedDataChunk">
  select
    Chunk_Key    as key,
    Chunk_Data   as data,
    Valid_From   as validFrom,
    Valid_To     as validTo
  from
    CSD.Cache
  where
    (Chunk_Key = #{key})
    and
    (Valid_To > current timestamp)
    and
    (Valid_From > #{from})
  with ur
</select>
```



- iBatis má v sobě celkem 22 konvertorů mezi Java typy a DB typy
 - Pokrývají naprostou většinu běžných potřeb
- Pokud to nestačí, lze napsat další a zanést je do konfigurace
 - Globálně pro všechny výskyty: `<typeHandler/>`
 - Selektivně pro jednotlivou položku

```

/**
 * Custom iBatis type handler for boolean values represented in database as 'Y' or 'N'.
 *
 */
public class StringBooleanTypeHandler implements TypeHandler {

    @Override
    public Object getResult(ResultSet rs, String name) throws SQLException {
        return valueOf(rs.getString(name));
    }

    <typeAlias type="com.ibm.learning.careerdashboard.typehandler.StringBooleanTypeHandler" alias="StringBooleanTypeHand

    public Object getResult(CallableStatement cs, int position) throws SQLException {
        return valueOf(cs.getString(position));
    }

    @Override
    public void setParameter(PreparedStatement ps, int position, Object value) throws SQLException {
        Boolean value = (Boolean) value;
        ps.setString(position, valueOf(value));
    }

    public Object valueOf(String value) {
        if (value.equals("Y")) {
            return Boolean.TRUE;
        } else {
            return Boolean.FALSE;
        }
    }

    public boolean equals(Object value1, String value2) {
        return valueOf(value2).equals(value1);
    }
}

```

```

<resultMap id="readConfigForComponentMap" type="CacheConfigItem">
    <result column="component" property="component" />
    <result column="useDynaCache" property="useDynaCache" typeHandler="StringBooleanTypeHandler" />
    <result column="useDatabase" property="useDatabase" typeHandler="StringBooleanTypeHandler" />
    <result column="timeToLiveDynaCache" property="timeToLiveDynaCache" />
    <result column="timeToLiveDatabase" property="timeToLiveDatabase" />
    <result column="pastLimitDynaCache" property="pastLimitDynaCache" />
    <result column="pastLimitDatabase" property="pastLimitDatabase" />
    <result column="usePastLimit" property="usePastLimit" typeHandler="StringBooleanTypeHandler" />
</resultMap>

```



- Kam se bude aplikace připojovat?
 - Napřímo bez poolu konexí
 - Driver, URL, jméno, heslo
 - Napřímo s poolem konexí
 - Navíc nastavení poolu
 - Přes data source aplikačního serveru
 - JNDI jméno data sourcu na serveru
- Jak budou řízeny transakce?
 - JDBC
 - Volají se JDBC commits a rollbacks
 - Managed
 - Řídí kontejner, iBatis nedělá nic
 - Vlastní implementace

Jméno prostředí, bude použito v aplikaci

Transakce bude spravovat kontejner

```
<!-- Environment for the Career Dashboard database with a data source configured on WAS -->
<environment id="CSD_via_WAS">

    <!-- Transactions will be managed by the application server container -->
    <transactionManager type="MANAGED">
    </transactionManager>

    <!-- The data source is defined on the application server and accessed via JNDI -->
    <dataSource type="JNDI">
        <property name="data_source" value="jdbc/hr_careersmart_csddb"/>
    </dataSource>
</environment>
```

Budeme se připojovat přes JNDI jméno data sourcu na serveru

JNDI jméno data sourcu

Jméno prostředí, bude použito v aplikaci

Transakce bude spravovat iBatis přes JDBC volání.
Nutno v kódu explicitně volat commit() a rollback().

```
<environment id="CSD_Directly">

  <!-- Transactions will be managed by iBatis using JDBC -->
  <transactionManager type="JDBC">
  </transactionManager>

  <!-- The data source is defined explicitly here -->
  <dataSource type="UNPOOLED">
    <property name="driver" value="com.ibm.db2.jcc.DB2Driver" />
    <property name="url" value="${CSD.URL}" />
    <property name="username" value="${CSD.Username}" />
    <property name="password" value="${CSD.Password}" />
  </dataSource>
</environment>
```

Budeme se připojovat
napřímo bez poolu konexí

Nějaké vlastnosti se dodají
externě, například
z konfiguračního souboru

- Seznam XML souborů
- Každý má uvnitř své ID, podle něj se pak vyhledá

```
<mappers>
  <mapper resource="com/ibm/learning/careerdashboard/data/folder/FolderMapper.xml"/>
  <mapper resource="com/ibm/learning/careerdashboard/data/db2sys/Db2SysMapper.xml"/>
  <mapper resource="com/ibm/learning/careerdashboard/data/cache/CacheMapper.xml"/>
  <mapper resource="com/ibm/learning/careerdashboard/data/cacheconfig/CacheConfigMapper.xml"/>
  <mapper resource="com/ibm/learning/careerdashboard/data/taxonomy/TaxonomyMapper.xml"/>
  <mapper resource="com/ibm/learning/careerdashboard/data/skill/SkillMapper.xml"/>
  <mapper resource="com/ibm/learning/careerdashboard/data/xml2db/ExpertiseTaxonomyMapper.xml"/>
  <mapper resource="com/ibm/learning/careerdashboard/data/authorization/AuthorizationMapper.xml"/>
  <mapper resource="com/ibm/learning/careerdashboard/data/registrycontent/RegistryContentMapper.xml"/>
  <mapper resource="com/ibm/learning/careerdashboard/data/developmentandactivity/DevelopmentAndActivityMapper.xml"/>
  <mapper resource="com/ibm/learning/careerdashboard/data/searchcollection/SearchCollectionMapper.xml"/>
  <mapper resource="com/ibm/learning/careerdashboard/data/configuration/ConfigurationMapper.xml"/>
  <mapper resource="com/ibm/learning/careerdashboard/data/userdata/UserDataMapper.xml"/>
  <mapper resource="com/ibm/learning/careerdashboard/data/itfcpformance/InterfacePerformanceMapper.xml"/>
  <mapper resource="com/ibm/learning/careerdashboard/data/mentorship/MentorshipMapper.xml"/>
  <mapper resource="com/ibm/learning/careerdashboard/data/searchcollectioncopy/SearchCollectionCopyMapper.xml" />
  <mapper resource="com/ibm/learning/careerdashboard/data/timelogging/TimeLoggingMapper.xml"/>
  <mapper resource="com/ibm/learning/careerdashboard/data/cleanoffboardemployees/CleanOffboardEmployeesMapper.xml"/>
</mappers>
```

1. Otevři iBatis session
2. Najdi správný mapper (nebo více mapperů)
3. Proved' všechny operace
4. [JDBC] Proved' commit() / rollback()
5. Zavři session

try

```
// Get iBatis session
1. session = IBatisSessionFactory.getIBatisSqlSession(DatabaseEnvironment.CSD_VIA_WAS);

// Get the mapper for Search Collection Copy
2. SearchCollectionCopyMapper mapper = session.getMapper(SearchCollectionCopyMapper.class);

// Perform the SQL query
3. searchCollectionCopy = mapper.readSearchCollectionCopy(params);
```

catch

```
5. finally
{
    // Close the iBatis session
    if (session != null)
        session.close();
} // finally
```


1. Načíst XML konfiguraci
2. Vytvořit SqlSessionFactoryBuilder
 - Není třeba kešovat
3. Z něj vytáhnout SqlSessionFactory
 - Dobré kešovat pro celou aplikaci (singleton)
4. Z ní otevřít novou session
 - Použít a zahodit
 - Nekešovat, nesdílet mezi vlákny, nedávat do HTTP session apod.

```

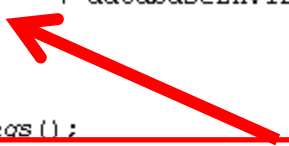
if (factory == null)
{
    log.info("iBatis SQL connection factory for database environment '" + databaseEnviron
1 Reader reader = Resources.getResourceAsReader(PATH_TO_IBATIS_CONFIG);
2 SqlSessionFactoryBuilder builder = new SqlSessionFactoryBuilder();
3 factory = builder.build(reader, databaseEnvironment.getId(), properties);
  sqlSessionFactories.put(databaseEnvironment, factory);
  log.info("iBatis SQL connection factory for database environment '" + databaseEnviron
} // if factory null


```

```

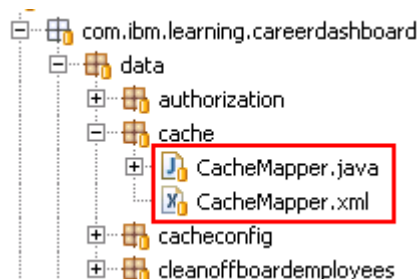
Properties settings = loadBuiltInSettings();
SqlSessionFactory factory = getIBatisSqlSessionFactory(databaseEnvironment, settings);
4 SqlSession session = factory.openSession();

```



1. Úvod do ORM
2. Úvod do MyBatisu
3. Školní příklad
4. Konfigurace
-  5. **Píšeme mappery**
6. Co se nevešlo jinde
7. Příklad z praxe
8. Porovnání s JPA

- **Java interface + XML soubor** nebo alespoň jedno z toho
- XML jde nahradit anotacemi
 - Ale čisté SQL je daleko přehlednější
- Interface jde vypustit
 - Ale je fajn mít ho kvůli přesnému typování



```

1 <?xml version="1.0" encoding="UTF-8"?>
2
3 <!DOCTYPE mapper PUBLIC "-//ibatis.apache.org//DTD Mapper 3.0//EN" "http://ibatis.apache.org/dtd/ibatis-mapper.dtd">
4
5 <mapper namespace="com.ibm.learning.careerdashboard.data.cache.CacheMapper">
6

```

```

2 package com.ibm.learning.careerdashboard.data.cache;
3
4 import java.sql.Timestamp;
5
6 /**
7  * iBatis mapper for the {@code CachedDataChunk} entity
8  *
9  * @author jkacer@at.ibm.com
10 */
11 public interface CacheMapper
12 {
13     /**

```

- V Java rozhraní metoda, ve XML element `<select/>`
- 1 vstup, 1 výstup
 - Primitivní typy, mapy, vlastní typy
 - Ve XML se nerozlišuje počet navrácených prvků, v Javě ano – `List<X>`

```
<select id="findDataChunkInsertedAfter" parameterType="LookupData" resultType="CachedDataChunk">
  select
    Chunk_Key    as key,
    Chunk_Data    as data,
    Valid_From    as validFrom,
    Valid_To      as validTo
  from
    CSD.Cache
  where
    (Chunk_Key = #{key})
    and
    (Valid_To >= current timestamp)
    and
    (Valid_From >= #{from})
  with ur
</select>
```

Položka
vstupu s
křížkem

```
/**
 * Reads a data chunk from the database, based on its key and expiration time.
 * If the chunk has expired, it will not be returned. It must be younger than the specified time.
 *
 * This select uses the "WITH UR" clause that causes DB2 not to acquire any locks.
 * It would hold locks for a long time and deadlocks would occur. On the other hand,
 * it allows for better concurrency.
 *
 * @param lookupData
 *      A DTO holding the key identifying the data chunk to be found and the
 *      minimum possible time.
 *
 * @return The chunk read or null if there is no such chunk.
 */
public CachedDataChunk findDataChunkInsertedAfter(LookupData lookupData);
```

```
public class LookupData {
    /**
     * Key of the chunk.
     */
    private String key;

    /**
     * The minimum possible time.
     */
    private Timestamp from;
```

```
public class CachedDataChunk implements Serializable {
    /**
     * Serial version UID.
     */
    private static final long serialVersionUID = 666444444444444444L;

    /**
     * Unique key of this data chunk.
     */
    private String key;

    /**
     * Cached data.
     */
    private byte[] data;
```

- Na výstupu lze použít i speciálně definovanou mapu
- Přínosy:
 - Lze nadefinovat mapování mezi DB a Javou
 - Lze použít typové konverze na určité sloupce
 - Lze použít další vychytávky, viz dále

Toto je definice
výstupní mapy

```
<resultMap id="publishRuleMap" type="PublishRule">
  <result property="laId" column="LA_ID"/>
  <result property="laRuleId" column="LA_RULE_ID"/>
  <result property="key" column="KEY"/>
  <collection property="values" javaType="ArrayList" column="VALUE" ofType="String"/>
</resultMap>
```

Vychytávka!

```
<select id="selectPublishRuleByAppIdAndLaIds" resultMap="publishRuleMap" parameterType="SelectPublishRuleParams">
  select
    prk.value,
    pr.la_id,
    pr.la_rule_id,
    prk.key
  from
    lnp.publish_rule as pr
  join
    lnp.publish_rule_key as prk
  on
    pr.la_rule_id = prk.la_rule_id
  where
    pr.app_id = #{appId} and
    pr.la_id in
    <foreach collection="listOfLaIds" item="item" index="index" open="(" close=")" separator=",">
      #{item}
    </foreach>
  with ur
</select>
```

```
public class PublishRule implements Serializable {
    /**
     * Serial version UID.
     */
    private static final long serialVersionUID = 1L;

    private int laId;
    private int laRuleId;
    private String key;
    //flag says if rule gone through evaluation
    private boolean passedThroughEvaluation;
    private List<String> values;
}
```

```
/**
 * Selects Rule Keys grouped by Rule Id from one App Id and and list of LA IDs
 * @param params - Parameters contain one app_id and list of la_id
 * @return - A list of PublishRule objects
 */
public List<PublishRule> selectPublishRuleByAppIdAndLaIds(SelectPublishRuleParams params)
```

- Elementy `<insert/>`, `<update/>`, `<delete/>`
- Pouze vstup, žádný výstup

```
<insert id="storeDataChunk" parameterType="CachedDataChunk">
    insert into CSD.Cache (Chunk_Key, Chunk_Data, Valid_From, Valid_To)
    values ({key}, #{data,jdbcType=BLOB}, #{validFrom}, #{validTo})
</insert>
```

```
<update id="overwriteDataChunk" parameterType="CachedDataChunk">
    update
        CSD.Cache
    set
        Chunk_Data = #{data,jdbcType=BLOB},
        Valid_From = #{validFrom},
        Valid_To = #{validTo}
    where
        Chunk_Key = #{key}
</update>
```


```
<delete id="removeDataChunk" parameterType="String">
    delete
    from
        CSD.Cache
    where
        Chunk_Key = #{key}
</delete>
```




- Uvnitř elementu `<resultMap/>` lze používat další elementy než `<result/>`:
 - `<id/>` - Unikátní položka, lepší výkon
 - `<constructor/>` - Nastavení položek výsledků přes konstruktor
 - `<association/>` - Asociace s vnořeným objektem
 - `<collection/>` - Asociace s vnořeným seznamem objektů
 - `<discriminator/>` - Podmíněné tvarování výstupní mapy na základě hodnoty sloupečku; ekvivalent výrazu switch v Javě
- Poměrně komplikované konstrukce u všech, nutno nastudovat z dokumentace

- SQL příkazy lze „ohýbat“ za běhu dle vstupních parametrů
- `<if test="...">`
 - Podmíněné vložení
- `<choose> <when test="..."> <when test="..."> ... <otherwise>`
 - Vícenásobné větvení (switch)
- `<foreach item index collection open separator close> ... </foreach>`
 - Iterace přes seznam hodnot
 - Vhodné např. pro IN (...)

```
<select id="selectBrokerLearningActivityKeysAndValuesByLaId" resultMap="keyValuesMap" parameterType="LaIdsAndColumns">
  select
    blak.VALUE,
    blak.LA_ID,
    blak.KEY
  from
    LNP.BROKER_LEARNING_ACTIVITY_KEY as blak
  where
    blaK.LA_ID in
    <foreach collection="laIds" item="item" index="index" open="(" close=")" separator=",">
      #{item}
    </foreach>
  with ur
</select>
```

1. Úvod do ORM
2. Úvod do MyBatisu
3. Školní příklad
4. Konfigurace
5. Píšeme mappery
-  6. **Co se nevešlo jinde**
7. Příklad z praxe
8. Porovnání s JPA



- iBatis má 2 úrovně vlastní keše
- Na úrovni mapperů
- Běžně je zapnuta jen první
 - Jen kešování uvnitř té samé session – lepší výkon, detekce cyklických odkazů
- Druhou je třeba zapnout elementem `<cache/>`
 - Výsledky SELECTů se budou kešovat
 - INSERT, UPDATE, DELETE nuluje keš
 - Vyhazování z paměti: LRU, dále lze FIFO, SOFT, WEAK
 - Maximálně 1024 výsledků, lze nastavit
 - Lze nastavit čas pro vyprazdňování
 - Read-only nebo read/write
 - RO rychlejší, ale dostaneme sdílený výsledek
 - RW pomalejší, ale bezpečný (kopie)

- SQL je možno generovat zcela dynamicky za běhu z Java kódu
- Select Builder poskytuje „makra“ odpovídající SQL klíčovým slovům
- Trošku lepší „čisté JDBC“
- Pokud nemusíte, nepoužívejte

```
private String selectPersonSql() {
    BEGIN(); // Clears ThreadLocal variable
    SELECT("P.ID, P.USERNAME, P.PASSWORD, P.FULL_NAME");
    SELECT("P.LAST_NAME, P.CREATED_ON, P.UPDATED_ON");
    FROM("PERSON P");
    FROM("ACCOUNT A");
    INNER_JOIN("DEPARTMENT D on D.ID = P.DEPAR");
    INNER_JOIN("COMPANY C on D.COMPANY_ID = C.");
    WHERE("P.ID = A.ID");
    WHERE("P.FIRST_NAME like ?");
    OR();
    WHERE("P.LAST_NAME like ?");
    GROUP_BY("P.ID");
    HAVING("P.LAST_NAME like ?");
    OR();
    HAVING("P.FIRST_NAME like ?");
    ORDER_BY("P.ID");
    ORDER_BY("P.FULL_NAME");
    return SQL();
}

private String selectPersonLike(Person p){
    BEGIN(); // Clears ThreadLocal variable
    SELECT("P.ID, P.USERNAME, P.PASSWORD, P.FIRST_NAME, P.LAST_NAME");
    FROM("PERSON P");
    if (p.id != null) {
        WHERE("P.ID like #{id}");
    }
    if (p.firstName != null) {
        WHERE("P.FIRST_NAME like #{firstName}");
    }
    if (p.lastName != null) {
        WHERE("P.LAST_NAME like #{lastName}");
    }
    ORDER_BY("P.LAST_NAME");
    return SQL();
}
```

- Práce s transakcemi vychází z nastavení prostředí:
 - JDBC
 - Aplikace volá metody iBatisu commit(), rollback()
 - iBatis volá JDBC metody ovladače
 - Managed
 - Aplikace si ošetří transakce po svém
 - EJB: CMT, BMT
 - Externí transakční manažer
 - iBatis nedělá nic

```
try
{
    start();
    UserDataMapper mapper = session.getMapper
    this.cnumsUserData = mapper.getAllCnums()
    log.debug("Read this count of cnums from
    session.commit();
} // try
catch (Exception e)
{
    log.error("An error occurred during datab
    session.rollback();
    throw e;
} // catch
```


1. Úvod do ORM
2. Úvod do MyBatisu
3. Školní příklad
4. Konfigurace
5. Píšeme mappery
6. Co se nevešlo jinde
- 7. Příklad z praxe**
8. Porovnání s JPA





- Webová aplikace, aplikační logika běží na IBM WebSphere AS 7.0, HTML a JavaScript na Apache HTTPD
- 3 různé databáze, všechny IBM DB2
 - Přístup k více DB v rámci jedné transakce → XA transakce, záležitost konfigurace zdrojů dat
 - Práce s DB výlučně přes iBatis 3
 - Navíc přístup do DB mimo WAS z periodicky spouštěných jobů (Java SE)
- Ajaxové načítání dat
 - Statické HTML stránky s JavaScriptem – [Dojo](#)
 - Pro načítání data [DWR](#) – klientská a serverová strana
 - Na titulní straně cca 20 paralelních požadavků na data → 20 paralelních XA transakcí, každá pracuje alespoň minimálně s DB

Hello Markus
Edit settings

Search w3
GO

w3 Home | BluePages | HelpNow | Feedback

IBM CareerSmart Home | PLAN your development | GAIN skills & experience | TRACK your progress | Search for learning | Help

IBM CareerSmart

IBM CareerSmart is the complete suite of career programs and resources available to you for growing your career at IBM. It is organized in an easy-to-navigate fashion:

[PLAN your development](#)
[GAIN skills and experience](#)
[TRACK your progress](#)

[Learn more](#) about how to grow your career at IBM.

Search for learning:
GO

Your career snapshot
About IBM CareerSmart

PLAN your development

Data	Details	Last updated	Actions
Career goals ⓘ	Your current job Your next job Your long-term goals	12 Nov 2010	Actions ▾
Business goals ⓘ	Review approved business goals View previous PBCs in the PBC tool View assessment history in About you	25 Mar 2010	Actions ▾
People management goals ⓘ	Review approved people management goals View previous PBCs in the PBC tool View assessment history in About you	25 Mar 2010	Actions ▾
Development goals and activities ⓘ	Review approved development goals and activities Create goals in the PBC tool Update results in the PBC Tool	25 Mar 2010	Actions ▾

GAIN skills and experience

Data	Details	Last updated	Actions
Required learning ⓘ	12 required learning activities	12 Nov 2010	Actions ▾

How to get started

- [PLAN your development](#)
- [GAIN skills and development](#)
- [TRACK your progress](#)

Important updates

This important update should be displayed for all IBMers [Profiled for IT Specialist, Professional Services, Business Subjects, Competitors, Solutions & Products, Technology, Global Business Services, IBM Employee, Regular, Northeast Europe IOT]

Important updates are finally coded so it's necessary to test if they work. [Profiled for Austria]

System alerts are coded also, but they will be tested later. [Profiled for All IBM]

Need Help?

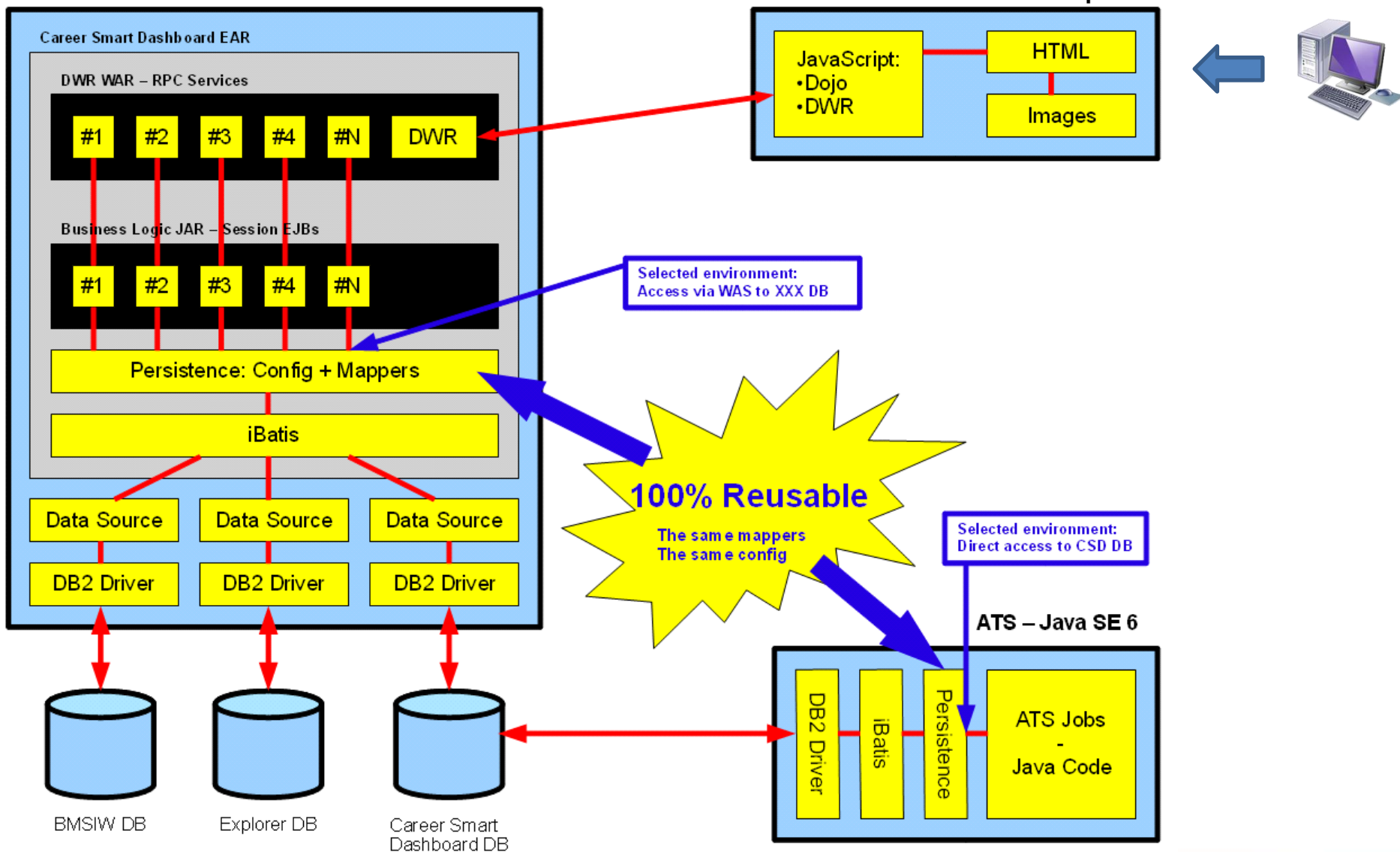
Review resources or consult a **Career Advisor** for questions or guidance.

[Resources and advisors](#)

For technical problems, visit [Learning Help](#).

Dynamicky načítaná data z WAS serveru

WAS – Java EE 5 over SE 6



1. Úvod do ORM
2. Úvod do MyBatisu
3. Školní příklad
4. Konfigurace
5. Píšeme mappery
6. Co se nevešlo jinam
7. Příklad z praxe



8. Porovnání s JPA

iBatis		JPA / Hibernate, TopLink, ...
Začínáte od databáze, důležitý je doménový model, ten „jen“ načítáte do Javy	Filozofický přístup	Začínáte od Javy, důležitý je objektový model, ten „jen“ ukládáte do databáze
Objekty dle výsledků dotazů	Mapování	Objekty dle tabulek (Nemusí být vždy pravda)
SQL, může být specifické pro DB	DB jazyk	Nic (nebo HQL)
Ne	Standard	Ano
Lze optimalizovat dotazy dle DB, indexů, využití, ...	Výkonnost	Závisí na implementaci
Záleží jen na napsaném SQL	Přenositelnost	Různé implementace, případně dialekty (Hibernate)
.NET	Jiné platformy	Ne (ano např. pro Hibernate)



iBatis je vhodný, když:

- Databáze není vaše, používá ji více aplikací
- Databáze již existuje
- Máte připravené SQL dotazy a umíte SQL
- Požadujete maximální průhlednost a „čistotu“ z pohledu databáze
- Potřebujete optimalizovat přímo pro danou DB
- Nechcete se zbytečně učit/nasazovat JPA
- Nepoužíváte kanón na vrabce



MyBatis:

- <http://www.mybatis.org/index.html>
- <http://code.google.com/p/mybatis/>
- <http://groups.google.com/group/mybatis-user>

Porovnání technologií:

- <http://www.javaworld.com/javaworld/jw-07-2008/jw-07-orm-comparison.html?page=1> (starší verze)
- <http://www.luckyspin.org/?p=68>