

Data Persistence on Teaching Material Management System based on iBatis

Zhe Li^{1,a}, Hui Ma^{2,b}

¹College of International Exchange, Bohai University, Jinzhou, 121013, China

²School of Mathematics and Physics, Bohai University, Jinzhou, 121013, China

^a503412508@qq.com, ^bhuima2016@163.com

Keywords: iBatis; teaching material management system; data persistence; data logical structure design

Abstract. Teaching materials are the basis for teachers to teach and do a good job teaching and educating people work, but also students access to system knowledge, the development of intellectual and ideological and moral awareness to improve the important tool. Teaching material management system is an important part of university educational administration system. Data persistence is an important work in the development of software system. Based on iBatis, this paper provides support for the development of teaching material management system and serves for improving the educational administration level. Realization of data persistence in iBatis frame structure and data logical structure design as the foundation, respectively, to achieve the overall configuration file, entity object class and entity mapping file, explains the implementation method and gives the specific code, developers can according to the implementation of the revised use.

Introduction

Teaching materials are the carrier of knowledge embodying teaching contents and teaching methods, which are the basic tools to carry out teaching work and the important guarantee to improve teaching quality. Teaching materials management is an important part of educational administration, teaching materials management is the basic guarantee for the smooth development of teaching work, but also to meet the teachers and students use the inherent requirements of teaching materials. At present, most colleges and universities based on the actual situation of the school, with different development methods to establish a textbook management information system, to a certain extent, to meet the current needs of teaching materials management to address the diversification of customers and personalized services and other issues. The construction of scientific and practical teaching materials management system, improve the educational management system in the teaching materials management deficiencies, so as to improve work efficiency and improve the educational administration level. Has the significance in the following three aspects: First, enhance the management level of teaching materials. The teaching materials management workflow, information and automation, not only improve work efficiency, and can handle the relevant work anytime, anywhere; The second is to reduce the burden on staff. The use of management systems, different positions of people in different places, different time to deal with day-to-day work, will reduce the work of academic staff pressure; Third, the materials management transparency. Management staff can use the network to deal with related work, students and teachers can view the work of the textbooks anytime, anywhere progress, view the cost of teaching materials, help to manage the transparency of the work.

Data persistence is the process of converting the in-memory data model into the storage model and the storage model into the data model in memory. It is an important content of software system development to realize the exchange of data in the memory and the database. Java development of the most commonly used in the field of persistent technology is Hibernate and iBatis. Hibernate is a "fully automatic" persistence technology, to achieve the JDBC lightweight object encapsulation, the program is more object-oriented, programmers do not need to master the complex syntax of the SQL statement structure. iBatis is an open source project for Apache. The term iBatis is derived

from the combination of "internet" and "abatis" and is an O / R Mapping solution. iBatis framework for the SQL statement input and output mapping, starting from the relational to the object-oriented ideas to solve the problem of database operation, according to the persistent object to get the results of the operation. Compared with Hibernate, iBatis is "semi-automatic", need to write part of the SQL statement, iBatis to achieve the Java objects and SQL statements match, both to play the programmers to write SQL statement highly optimized features, while solving the Java objects and input parameters And result set mapping, this paper based on iBatis implementation, for the development of teaching materials management system to provide support.

iBatis Framework Structure

iBatis is a semiautomatic data persistence framework. It consists of SQLMaps and DAO components. The SQL objects are used to complete the CRUD (Create, Delete, Update, Retrieve) operation of Java objects. The database can be accessed through DAO components. Compared with Hibernate, JPA, iBatis is characterized by semi-automatic. In terms of Hibernate, has been completely encapsulated the structure of the database, with a set from POJO to the database table mapping mechanism; In the development process only need to define its mapping, you can achieve database access, do not even need to master SQL syntax. This approach is fast, but in some specific cases, the need for some complex data manipulation. iBatis solves this problem by allowing the programmer to customize the value of the SQL configuration bean properties and also allows the use of the database extension standard SQL syntax. Can be configured by XML file name and ID, real-time interaction with the database, convenient and flexible for software developers to provide a larger free space. In the small application software and large and medium-sized enterprise-class applications in a wide range of applications, is currently convenient and flexible one of the persistence framework. iBatis framework shown in Fig. 1.

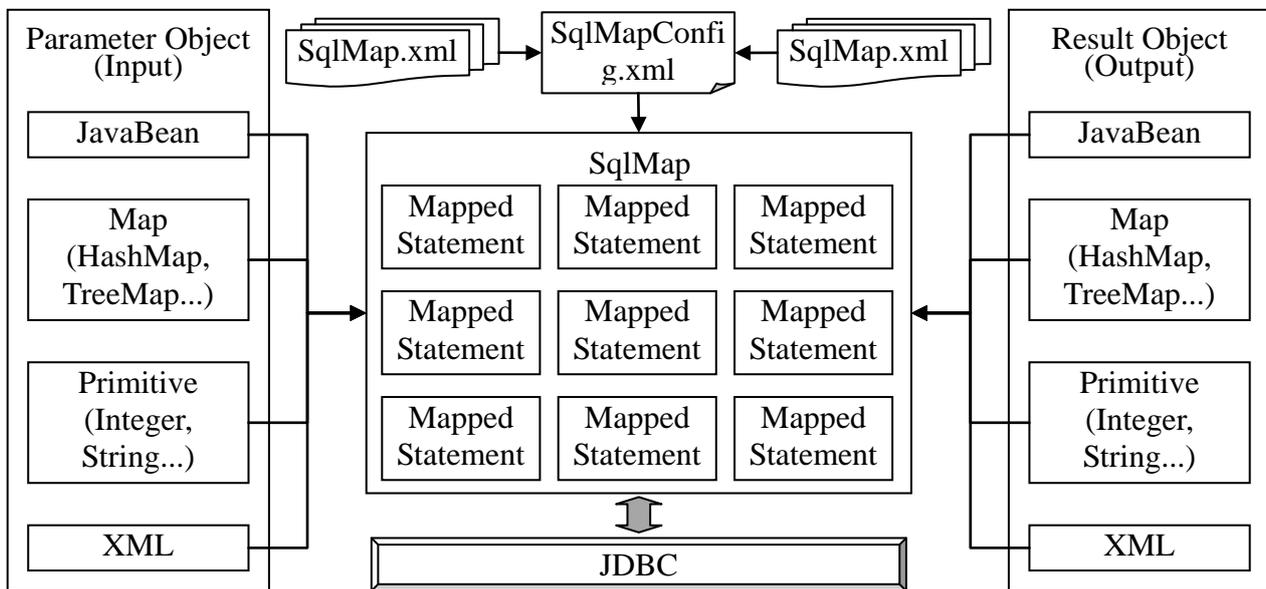


Fig. 1. Framework structure on iBatis

As can be seen from the figure, iBatis framework includes SqlMapConfig.xml, SqlMap.xml, ParameterObject, ParameterObject, ResultObject, SQLMap and JDBC and other six key structures. From left to right are the parameters of the input object (Parameter object), persistent operation (Persistent operation) and the outcome of the output object (Result object). Input and output objects can be JavaBean, Map, Primitive and XML and other types. iBatis can be on Oracle, SQLServer, DB2, MySql and PostgreSql other types of database for the persistence can also be one or a combination of several database systems. SqlMap Statement with the database through JDBC. JDBC is a Java API for executing SQL statements, consisting of a set of classes and interfaces written in the Java language that provide uniform access to multiple relational databases.

Data Logical Structure Design

Common data design includes conceptual structure design, logical structure design, physical structure design and security design. The core content of this paper is the persistence design, but the data structure is the basis of persistence, so only the logical structure design. The logical structure is related to a specific database management system. Considering the data standardization and integrity of the system, including five entities, which are "Teaching plan", "Teaching material plan", "Purchase order", "Invoice bills" and "Receive teaching material". There is a link between the various entities, with "Teaching plan" and "Teaching material plan" entity for example, between the two is "many to many", a "Teaching plan" Teaching material "corresponds to more than one plan", a "Teaching material plan" Teaching plan "corresponds to a number of".

Database logic structure based on Oracle database management system. Oracle is currently the most popular software product database system, is a kind of high efficiency and good reliability to meet the requirements of high throughput database solutions. Advantages embodied in five aspects: The advantages, introduced Shared SQL and multi-threading server architecture, reduce the resource usage, can take up the premise of lower resources to support more users; two advantages, provide security management role based on division of labor, has good performance in database management function, integrity checking, safety and consistency hand; The advantages of the three, to support a large number of multimedia data, such as binary graphics, sound, animation and multidimensional data structure etc.; Four advantages, provides the interface software PRO* series and the third generation of advanced language, can be in a high level language embedded SQL statements and process (PL/SQL) for database data manipulation; Four advantages, provides the interface software PRO* series and the third generation of advanced language, can be in a high level language embedded SQL statements and process (PL/SQL) for database data manipulation; Five advantages, provides a new distributed database, can easily read and write remote database data through the network, and symmetric replication technology. Taking "Teaching material plan" entity as an example, the results of the design of the logical structure of the database are shown in Table 1.

Table 1. Database logical structure on TeachingMaterialPlan

No	FieldName	DataType	Width
1	ID	NUMBER	10,0
2	CourseNumber	CHAR	10
3	UseTerm	CHAR	11
4	Class	CHAR	2
5	ResponsiblePerson	VARCHAR2	20
6	TeachingMaterialNumber	VARCHAR2	20
7	TeachingMaterialName	VARCHAR2	50
8	CIPnumber	VARCHAR2	20
9	AuthorName	VARCHAR2	50
10	Press	VARCHAR2	100
11	PublicationDate	DATE	7
12	UnitPrice	NUMBER	10,2
13	Quantity	NUMBER	10,0
14	Remarks	CLOB	

Data Persistence Implementation

iBatis persistence includes three major tasks: First, the overall configuration file sqlMapConfig.xml,

used to complete the connection with the database and related resource configuration; Second, the entity object class, that is, iBatis to operate the object is responsible for dealing with the database persistence layer object; Third, the entity mapping file, used to fill in the SQL statement, you need to sqlMapConfig.xml file to add the location of the mapping file.

(1) The overall configuration file. The SqlMapConfig.xml file contains three types of nodes: <settings> node, which defines information about the operation of the database; <TransactionManager> node, the node defines iBatis's transaction manager, currently includes JDBC, JTA and EXTERNAL, etc. or for selection; The <sqlMap> node specifies the location of the mapping file, and multiple sqlMap nodes can appear in the configuration to specify all of the mapping files contained within the project. As SqlMapConfig.xml configuration is large, and most of the optional configuration, the system configuration is as follows:

<pre> <!-- sqlMapConfig.xml Configuration file header --> <sqlMapConfig> <settings cacheModelsEnabled="true" enhancementEnabled="true" lazyLoadingEnabled="true" errorTracingEnabled="true" maxRequests="30" maxSessions="10" maxTransactions="5" <!--Use Statement Namespace--> useStatementNamespaces="false" /> <!-- Reference JDBC property profile --> <properties resource="oracle.properties" /> <transactionManager type="JDBC"> </pre>	<pre> <dataSource type="SIMPLE"> <!-- Database connection status detection --> <property name="Pool.PingQuery" value="select * from TeachingMaterialPlan "/> <!-- Allow connection state detection --> <property name="Pool.Enabled" value="true"/> <!-- Idle more than set value for detection --> <property name="Pool.onUsedFor" value="1"/> </dataSource> </transactionManager> <!-- Configure all entity class mapping file locations --> <sqlMap resource="com/sqlMap/TeachingMaterialPlan.xml"/> <sqlMap resource="com/sqlMap/TeachingPlan.xml" /> <sqlMap resource="com/sqlMap/TeachingMaterialPlan.xml"/> <sqlMap resource="com/sqlMap/PurchaseOrder.xml"/> <sqlMap resource="com/sqlMap/InvoiceBills.xml"/> <sqlMap resource="com/sqlMap/ReceiveTeachingMaterial.xml" /></sqlMapConfig> </pre>
---	--

(2) Entity object class. Entity object class is POJO (Plain Ordinary Java Object), can be understood as a simple entity class, without business logic java simple object, private definition of object properties for each attribute get and set methods as access interface, the entity object class does not Need to inherit the iBatis class or achieve iBatis interface, improved class independence. The entity object class for "TeachingMaterialPlan" is defined as follows:

<pre> package com.ibatis.model; public teachingMaterialPlan { private int id; private string courseNumber; private string useTerm; private string teacherID; private string class; private string responsiblePerson; private string teachingMaterialNumber; private string teachingMaterialName; private string cIPnumber; private string authorName; </pre>	<pre> private string press; private date publicationDate; private decimal unitPrice; private decimal unitPrice; private string remarks; //Attribute function public string iD { get {return iD;} set {iD=value} } //Omit other attribute functions } </pre>
--	---

(3) Entity mapping file. Used to complete the operation of the database object, including column mapping and dynamic SQL statements. Operation defined in the mapping file, to achieve the data storage logic and the upper logic code separation. For CRUD (Add Create, Read Retrieve, Update Update and Delete Delete) operations. The mapping file for the "TeachingMaterialPlan" entity is as follows:

<pre> <sqlMap namespace="TeachingMaterialPlan"> <!--Mapping attribute column name--> <resultMap id="result" class="MaterialPlanId"> <result property="iD" column="iD"></result> <!--Omit the other column name mapping--> </resultMap> <!--SQL statements for database operations, because many, CUID each give one--> <!--Insert 1 records--> <insert id="insertID" parameterClass="int"> insert into TeachingMaterialPlan VALUES (#iD#, #courseNumber#, #useTerm#, #teacherID#, #class#, #responsiblePerson#, #teachingMaterialNumber#, #teachingMaterialName#, #cIPnumber#, #authorName#, #press#, #publicationDate#, #unitPrice#, #unitPrice#, #remarks#) </insert> </pre>	<pre> <!--Update 1 records--> <update id="updateID" parameterClass="string"> update TeachingMaterialPlan set TeachingMaterialName=#teachingMaterialName# where TeachingMaterialName=#teachingMaterialName# </update> <!--Delete 1 records--> <delete id="deleteID" parameterClass="string"> delete from TeachingMaterialPlan where ID=#deleteID# </delete> <!--Query records--> <select id="findID" parameterClass="string" resultMap="result"> select * from TeachingMaterialPlan TeacherID uid=#teacherID# </select> </pre>
--	---

References

- [1] G. J. Liu, J. L. Xia, "Research and application of MVC and iBATIS framework based on Spring," *Computer Security*, vol. 12, no. 7, pp. 25-30, 2012.
- [2] T. T. Cui, "Research of Web-based Textbook Management System for Colleges and Universities," *Computer Knowledge and Technology*, vol. 12, no. 6, pp. 56-57, 2016.
- [3] 360 personal library, "A variety of system architecture diagram and its brief introduction," http://www.360doc.com/content/12/1029/12/168576_244433043.shtml, 2017-1-10.
- [4] Lanxue forest blog, "iBatis simple tutorial," <http://www.cnblogs.com/ycxyyzw/archive/2012/10/13/2722567.html>, 2017-1-10.
- [5] L. Gao, "iBatis Persistence technology and its application in network teaching evaluation," *Automation & Instrumentation*, vol. 36, no. 5, pp. 136-138, 2016.