

JAVA认证:JSP中表单数据存储的通用方法Java认证考试 PDF转换可能丢失图片或格式，建议阅读原文

https://www.100test.com/kao_ti2020/571/2021_2022_JAVA_E8_AE_A4_E8_AF_81_c104_571334.htm 本文以Oracle数据库为例，介绍了在采用JSP技术开发WEB应用时一种简便通用的表单数据存储处理方法，以减轻开发工作量，同时提供了主要的程序代码。引言 J2EE(Java 2 Enterprise Edition)技术已广泛应用在Web应用开发中，其中的JavaBean、Servlet技术为开发者提供了更为清晰的开发环境，使用JSP技术表现页面，使用Servlet技术完成大量的业务处理，使用Bean来存储数据及一些业务处理。在WEB应用中，业务数据存储到数据库中的处理工作经常很繁重，其中一种主要的形式就是表单数据存储到数据库，整个应用处理过程牵涉到大量的这种数据存储操作，对每个表单都要单独编写相应的数据存储程序，花费了开发人员大量的时间和精力。采用什么方法来减轻表单数据存储的开发工作量是值得研究的问题。两种常见的表单数据存储处理方法 1、对每一表单都编写相应的程序代码 在JSP页面或JavaBean或Servlet中，使用request.getParameter()函数逐一提取表单提交的数据，或编写相应的JavaBean，使用setProperty方法将数据自动取到JavaBean中，然后生成SQL语句（insert，update，delete），最后执行executeUpdate()函数完成数据表存储。 2、对每一数据表自动生成一个JavaBean程序代码 数据库系统必须支持用户能够读取表结构，并识别关键字段。利用面向对象快速开发工具，如PowerBuilder、Delphi等，自行开发一个java代码自动生成程序。在该程序中读取数据库表的结构：字段名、数据类型、

数据长度，自动生成一个JavaBean代码。在该代码中定义与表中字段对应的同名变量，建立所有变量的setValue和getValue方法，建立insert、0update、0delete函数分别处理insert、0update、0delete的SQL语句生成和执行。在表单提交的数据处理页面中，编写如下代码，将表单数据存储到JavaBean中：

```
<jsp:useBean id="table" class="table1_bean" /> <jsp:setProperty name="table" property="*" />
```

(注：table1_bean为上述自动生成的对应某一个表的JavaBean) 然后调用table1_bean中insert、0update、0delete函数完成数据表存储，并返回执行结果。如：

```
<%boolean success =table.insert(); %>
```

第一种方法简单直观，但对每一表单都需要编写相应的数据处理程序。对稍微大一点的应用，表单数量可能很多，开发工作量很大，开发工作效率低。表结构变动如增加、减少字段时，需修改相应的数据处理程序。第二种方法相对第一种简便得多，每一数据表的数据处理由对应的JavaBean实现，JavaBean自动生成，不需编写，表结构变动时只需重新生成新的JavaBean，经java编译后覆盖原java类即可。但该方法需要开发JavaBean自动生成程序，表结构变动时JavaBean需要重新生成和编译。介绍一种简便通用的方法实现表单数据存储

在WEB应用开发中，很多表单在经过前台浏览器端简单的数据校验后，提交后台服务器，服务器对数据不用作任何处理直接将数据存储到一个数据表中。对这种情况，我们可以只编写一个程序，对这些表单统一处理，将数据存储到相应的一个数据表中。该方法同样要求数据库系统支持表结构读取和关键字段识别。我们采用JSP技术编写该程序，程序文件取名为DbdataStore.jsp。

- 1、调用格式 在网页中表单的Action调

用方法如下：`< Form Name=Frm1 Method=Post
Action="DBdataStore.jsp? tablename=table1& OperType=..."
>` table1为数据将要存储的数据库表的表名，OperType操作类型分为三种：insert，0update，0delete。表单中的`< input
type=text name=... >`，`< textarea name=... >` `< 0select name=...
>`等中的name值应与数据表的字段名相同，DBdataStore.jsp中逐一提取表单提交的对应字段名的数据值，若表单中未定义输入，得到的值为空值，则对该字段不作处理。

2、以oracle为例的视图定义

1) 建立表各列数据类型视图

```
CREATE OR REPLACE VIEW v_dbstru AS SELECT table_name
, column_name , data_type , data_length , data_precision
, data_scale , column_id FROM all_tab_columns WHERE
owner=user1 ; //user1为数据表的属主。
```

2) 建立表的关键列视图

```
CREATE OR REPLACE VIEW v_pkey_column AS SELECT
b.table_name , b.column_name , b.position FROM all_constraints
a , all_cons_columns b WHERE a.owner=b.owner AND
a.constraint_name=b.constraint_name AND a.owner=user1 AND
a.constraint_type=P ;
```

3、主要程序代码

1) 程序初始化

```
String tablename=request.getParameter("tablename") ; //提取表名
String OperType=request.getParameter("OperType") ; //提取操作类型
String sFieldValue="" ; //存放表单提交的字段数据值
String fieldname="" , Datatype="" //存放字段名，字段数据类型
int iFieldvalue=0 ;
String 0updateSql="" , whereSql=" where "
, insSql1="" , insSql2="" , opSql="" , strSql="" ;
ResultSet rs1=null , rs2=null ;
insSql1="insert into " + tablename + "(" ;
insSql2="values(" ;
```

2) 生成sql语句关键字段部分 生成insert语句

关键字段部分，如：insert into table1(id 和 values(100))；只使用关键字段生成update，delete语句where部分，如：where id=100；在操作类型为update时，网页form表单中不对关键字段的数据进行修改。rs1=Stmt.executeQuery("SELECT column_name FROM v_pkey_column WHERE table_name="tablename");

100Test 下载频道开通，各类考试题目直接下载。详细请访问 www.100test.com