作业1： Web开发环境的配置 JDBC的使用

姓名：王帆

学号：20152180

天津理工大学

Web应用程序设计与开发

目录

[第一部分：Web开发环境的部署与配置 2](#_Toc510283495)

[1.下载Tomcat 8.0，安装并配置环境变量 2](#_Toc510283496)

[2.安装并配置MyEclipse 5](#_Toc510283497)

[第二部分：JDBC的使用 11](#_Toc510283498)

[1.设计工具类DBUtil.java，实现对数据库数据的增删改 11](#_Toc510283499)

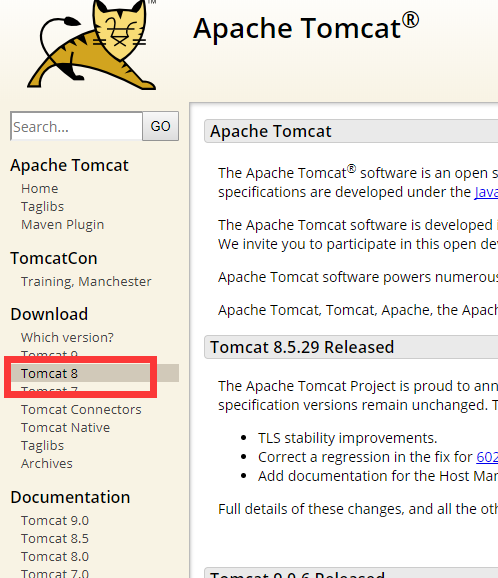
[2.调用实现的工具类，对数据库进行操作 14](#_Toc510283500)

[3.在SSMS或者其他数据库管理系统中检查是否完成操作 14](#_Toc510283501)

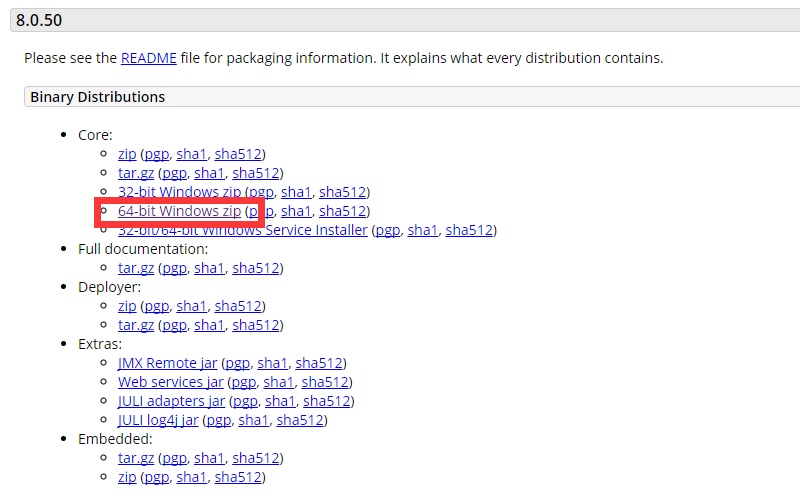
第一部分：Web开发环境的部署与配置

## 1.下载Tomcat 8.0，安装并配置环境变量

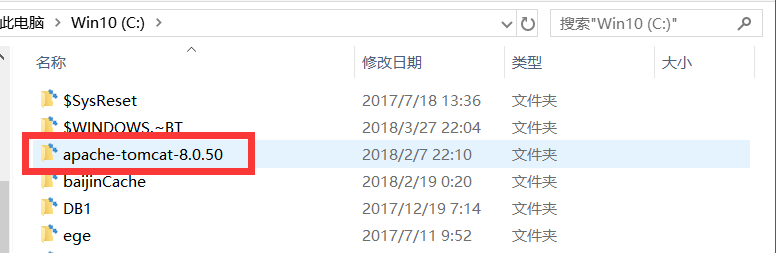
1.登录tomcat.apache.org， 找到Tomcat 8.0版本，下载



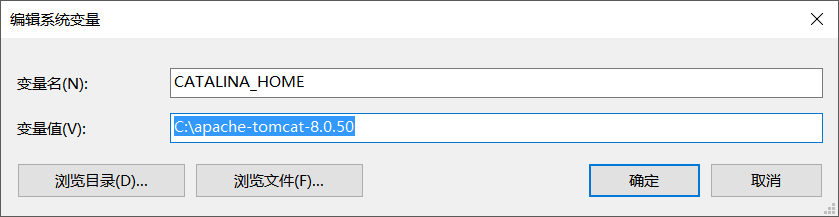
下载64-bit（或32-bit）Windows zip（压缩包版）



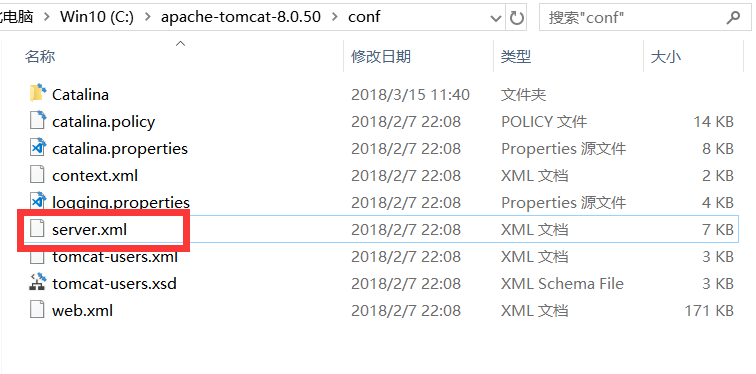
将下载后的压缩包解压至目标安装目录，如C:\Tomcat8



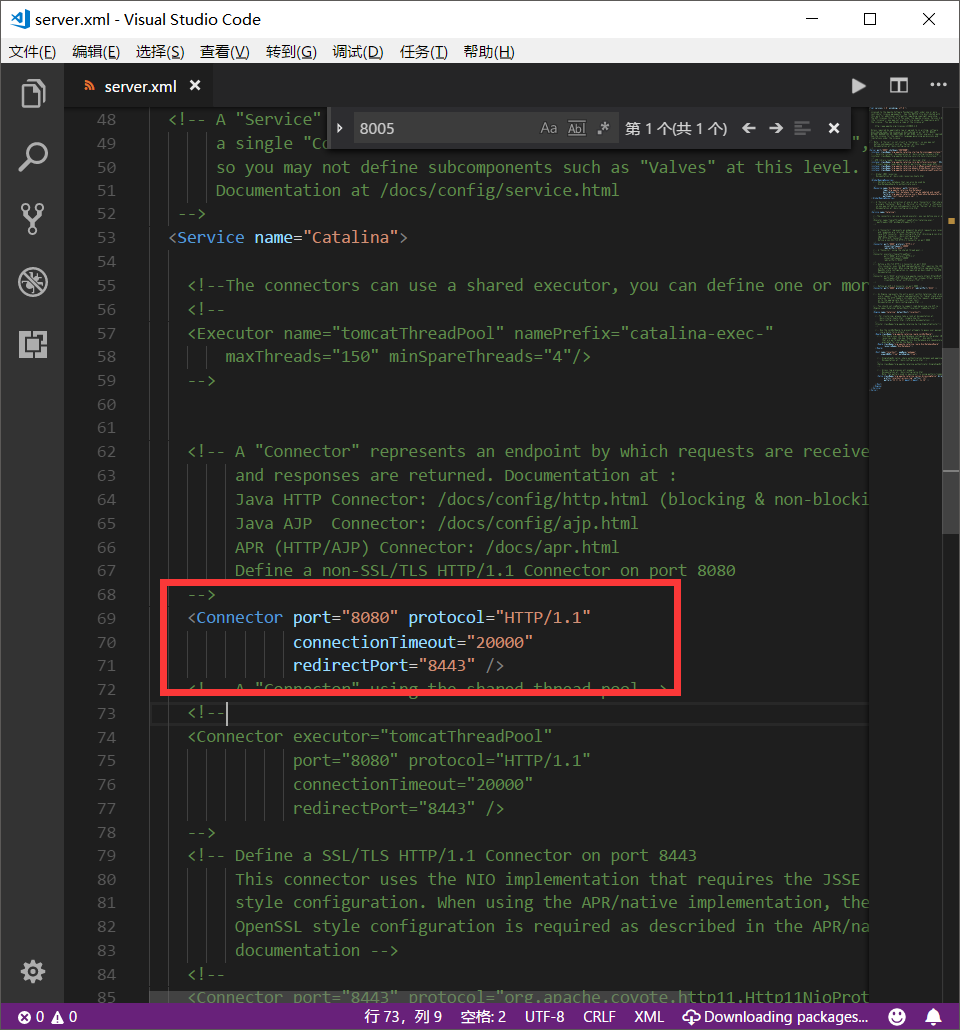
3.在系统-高级设置-环境变量中，新建一条系统变量，名为CATALINA\_HOME，变量值为Tomcat安装目录，如“C:\Tomcat8”。

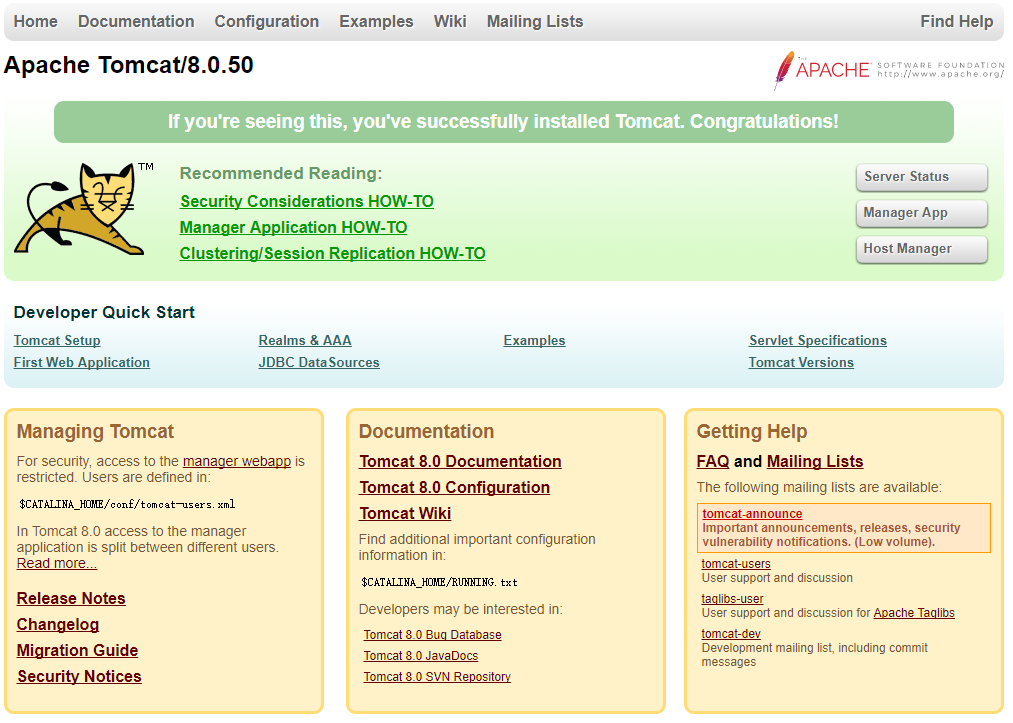


4. 在server.xml文件中，配置Tomcat服务器的端口号和首页等属性。



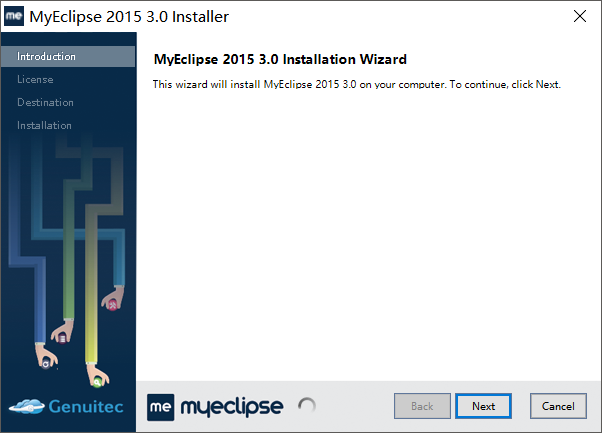
其中，port为访问端口号，默认8080。



5. 在浏览器中输入<http://localhost:8080>，打开Tomcat默认首页

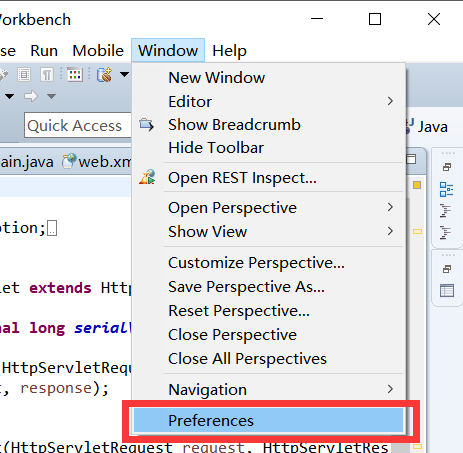
2.安装并配置MyEclipse

1.打开MyEclipse安装文件，一路next，最后安装成功。

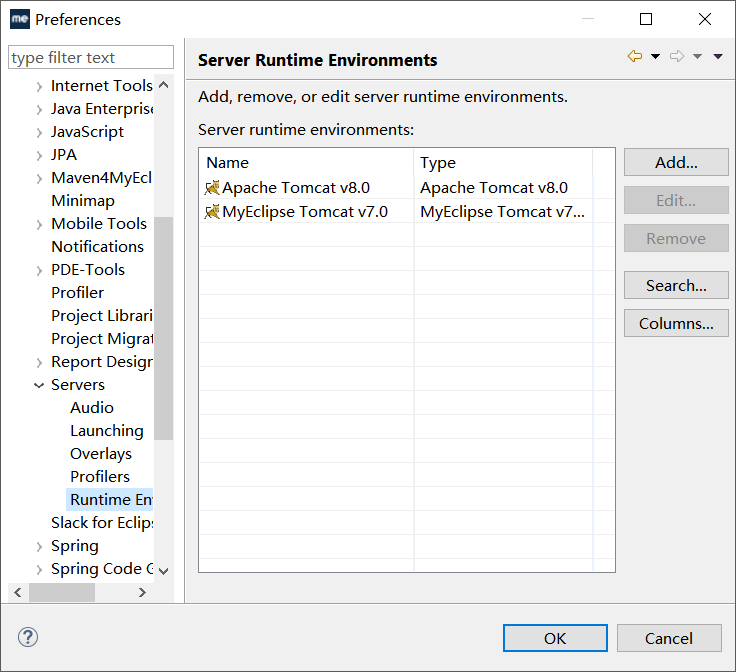


2.在MyEclipse环境配置刚刚安装的Tomcat 8.0。

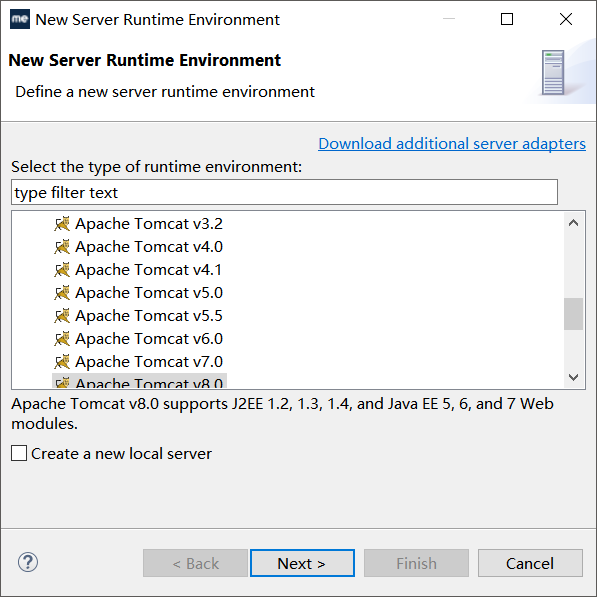
选择菜单栏Window - Preferences



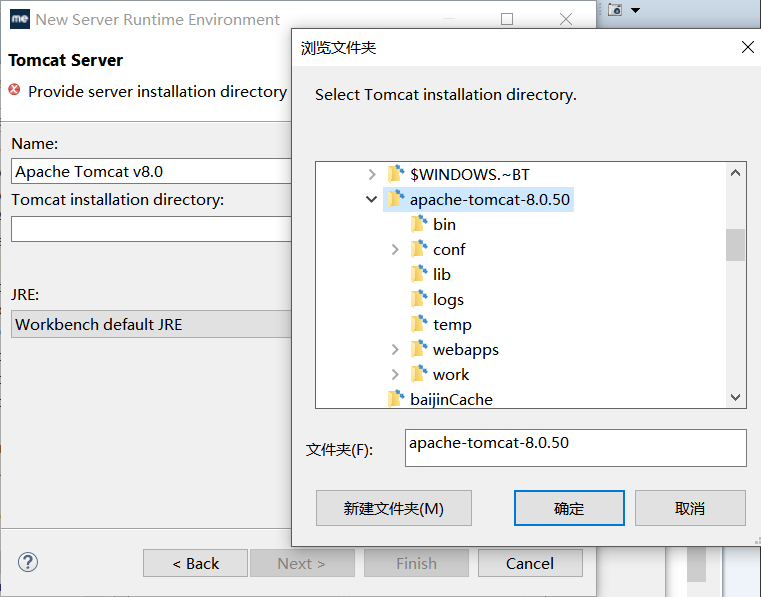
选择Servers – Runtime Environments，进入下图界面。



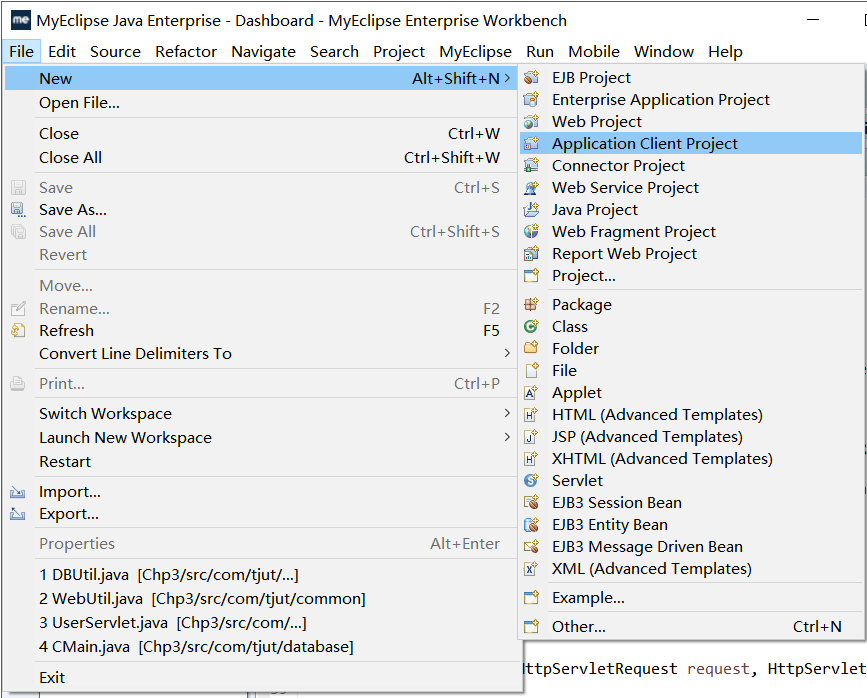
单击Add，在弹出的“New Server Runtime Environment”对话框中选择安装的Apache Tomcat v8.0



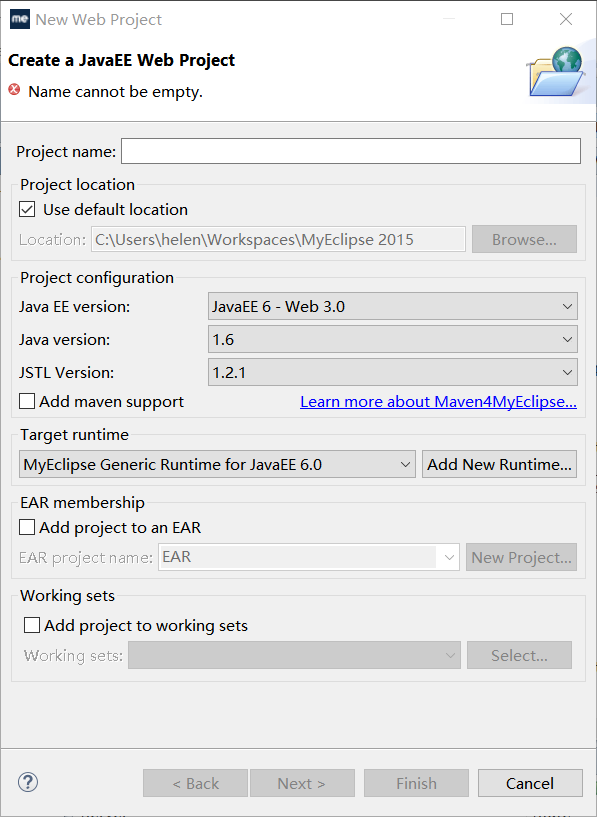
选择Tomcat安装目录，更改名称。，点击finish即可。



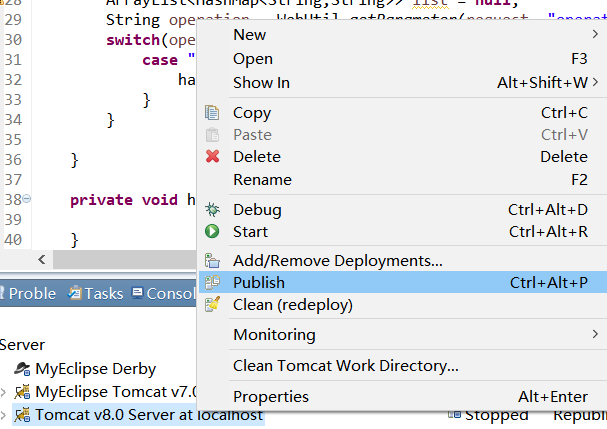
3.创建一个Web工程，并将该工程部署到Web服务器。



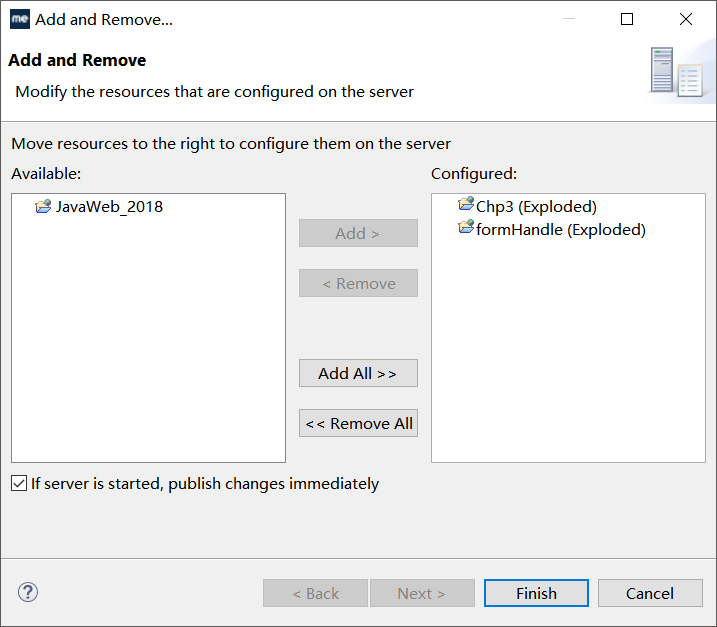
选择Web Project，在弹出的新建项目对话框中，填写项目名，选择JavaEE版本，以及目标运行时（Web容器），点击finish。



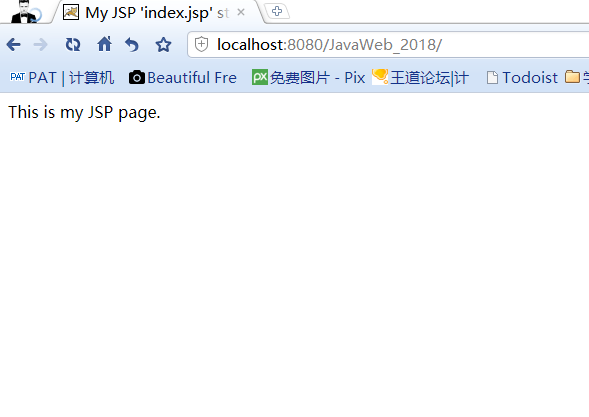
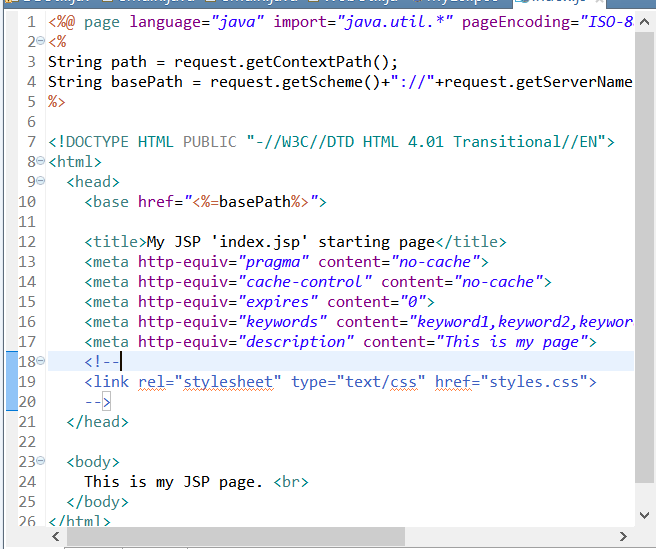
项目建立并设置完成后，右键点击刚刚建立的Tomcat服务器，选择Add/Remove Deployments。



在弹出的对话框中，将Available栏的准备发布的项目添加到Configured中，点击Finish



4.在Web工程内新建一个jsp文档，在jsp文档中实现任意功能（如 Hello World），并通过浏览器成功访问。



## 第二部分：JDBC的使用

## 1.设计工具类DBUtil.java，实现对数据库数据的增删改

|  |
| --- |
| //数据库操作工具类  **package** com.tjut.database;  **import** java.sql.Connection;  **import** java.sql.DriverManager;  **import** java.sql.ResultSet;  **import** java.sql.ResultSetMetaData;  **import** java.sql.SQLException;  **import** java.sql.Statement;  **import** java.util.ArrayList;  **import** java.util.HashMap;  **public** **class** DBUtil {  //1.定义并声明常用字段  **private** **static** String *url* =  "jdbc:sqlserver://127.0.0.1:1433;databasename";  **private** **static** String *user* = "sa";  **private** **static** String *pwd* = "sa";  //2.定义并声明SQL操作对象  **private** **static** Connection *conn* = **null**;  **private** **static** Statement *st* = **null**;  **private** **static** ResultSet *rs* = **null**;  //3.获取连接  **private** **static** Connection getConn() {  **try** {  Class.*forName*("com.microsoft.sqlserver.jdbc.SQLServerDriver");  *conn*=DriverManager.*getConnection*(*url*,*user*,*pwd*);  } **catch** (Exception e) {  e.printStackTrace();  }  **return** *conn*;  }  //4.定义容器方法,查询并返回ArrayList  //ArrayList<HashMap<String,String>>  **public** **static** ArrayList<HashMap<String,String>> getDataSet(String sql){  HashMap<String,String> hash = **new** HashMap<>();  ArrayList<HashMap<String,String>> list = **new** ArrayList<>();  ResultSetMetaData rsma = **null**;    **try** {  *conn* = DBUtil.*getConn*();  *st* = *conn*.createStatement();  *rs* = *st*.executeQuery(sql);  rsma = *rs*.getMetaData();  **int** columncount = 0;  **while**(*rs*.next()){  columncount = rsma.getColumnCount();  **for**(**int** i=0; i<columncount; i++){  hash.put(rsma.getColumnName(i), *rs*.getString(i));  }  list.add(hash);  }  } **catch** (SQLException e) {  e.printStackTrace();  }**finally**{  *finallyHandle*(*conn*, *st*, *rs*);  }  **return** list;  }  //5.执行SQL语句(INSERT/UPDATE/SELECT)  **public** **static** **boolean** executeBatch(ArrayList<String> list){  **boolean** flag = **true**;  **try** {  *conn* = *getConn*();//调用getConn()方法，初始化数据库连接  *conn*.setAutoCommit(**false**);  *st* = *conn*.createStatement();  **for**(**int** i=0;i<list.size();i++){  *st*.addBatch(list.get(i));  }  *st*.executeBatch();  *conn*.commit();  *conn*.setAutoCommit(**true**);  } **catch** (Exception ex) {  **try** {  *conn*.rollback();  } **catch** (SQLException e) {  e.printStackTrace();  }  flag = **false**;  ex.printStackTrace();  }**finally**{  *finallyHandle*(*conn*, *st*, *rs*);  }  **return** flag;    }  //6.最终处理（关闭数据库连接）  **private** **static** **void** finallyHandle (Connection conn, Statement st, ResultSet rs) {  **try** {  **if**(rs!=**null**){  rs.close();  rs=**null**;  }  **if**(st!=**null**){  st.close();  st=**null**;  }  **if**(conn!=**null**){  conn.close();  conn=**null**;  }  } **catch** (Exception ex) {  ex.printStackTrace();  }  }  } |

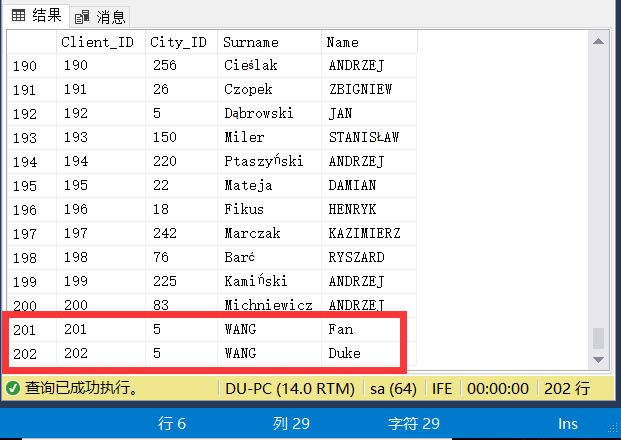
## 2.调用实现的工具类，对数据库进行操作

|  |
| --- |
| //数据库业务类  **package com.tjut.database;**  **import java.util.ArrayList;**  //导入数据库工具类  **import com.tjut.database.DBUtil;**  **public** **class** DBMain {  **public** **static** **void** main(String[] args) {  ArrayList<String> list = **new** ArrayList<>();  String sql = "";  **try**{  //测试增加条目  sql = "INSERT INTO Clients  (Client\_ID, City\_ID, Surname, Name) VALUES(201, 5, 'WANG', 'Fan')";  list.add(sql);  sql = "INSERT INTO Clients  (Client\_ID, City\_ID, Surname, Name) VALUES(202, 5, 'WANG', 'Duke')";  list.add(sql);  //测试修改条目  sql = "UPDATE Clients SET City\_ID = 20 WHERE Name ='Fan'";  list.add(sql);  //测试删除条目  sql = "DELETE FROM Clients WHERE Name = 'Fan' AND Surname = 'WANG'";  list.add(sql);    **boolean** flag = DBUtil.*executeBatch*(list);  **if**(flag){  System.***out***.println("OK!");  }  }**catch**(Exception e){  System.***out***.println("ERROR!");  e.printStackTrace();  }  }  } |

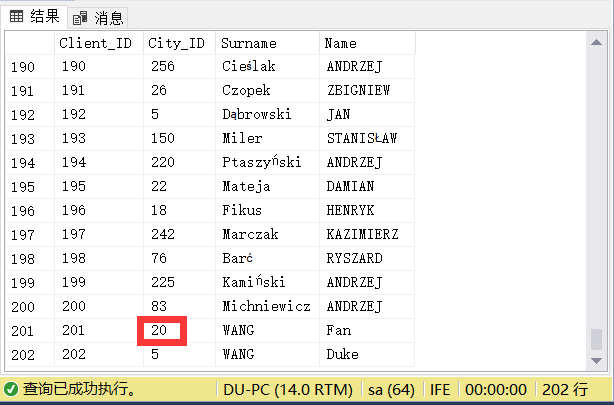
## 3.在SSMS或者其他数据库管理系统中检查是否完成操作

分别执行增加/修改/删除命令，并在SSMS中查看结果：

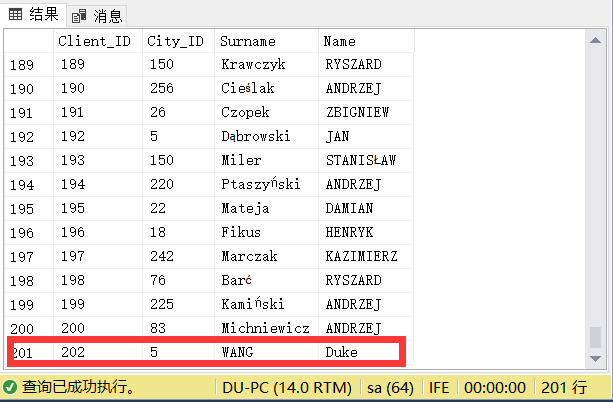
增加：



修改：



删除：



参考资料：

1. win10中打开SQL Server配置管理器方法 - He桃 - 博客园<https://www.cnblogs.com/He-tao-yuan/p/6744412.html>
2. java通过JDBC链接SQLServer2017 - CSDN博客<https://blog.csdn.net/rebekahq/article/details/78691343>
3. W3School：SQL 教程

<http://www.w3school.com.cn/sql/index.asp>

1. Tomcat7.0/8.0 详细安装配置图解，以及UTF-8编码配置 - CSDN博客

<https://blog.csdn.net/chenchunlin526/article/details/54705951>

1. MyEclipse中文教程连载 | 视频教程 | 开发文档 - MyEclipse官方中文网

<http://www.myeclipsecn.com/learningcenter/>