

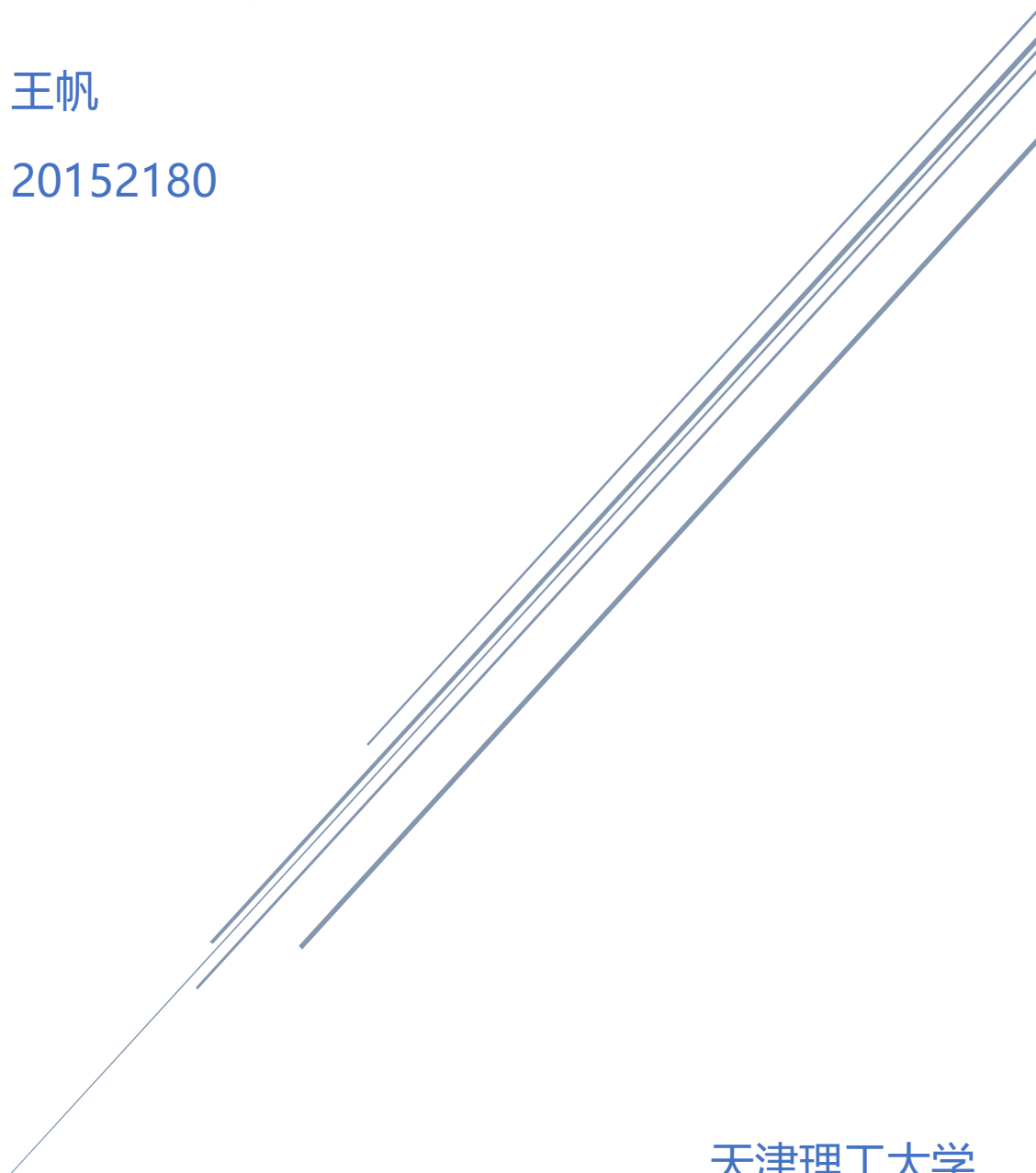
作业 1:

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

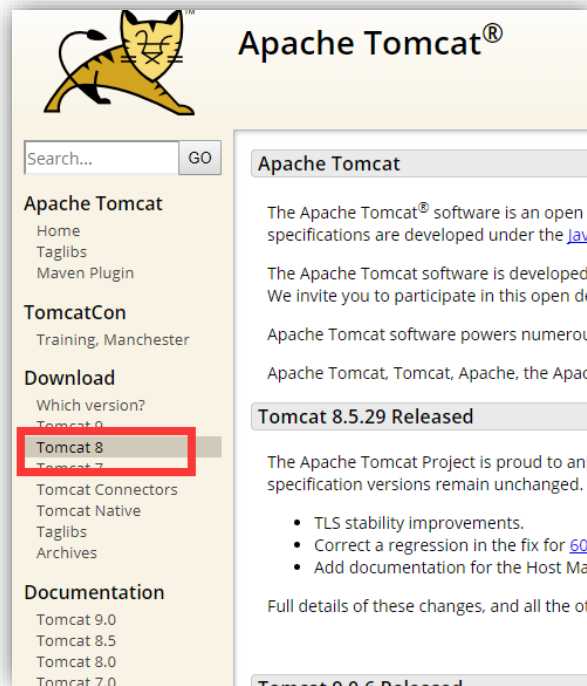
目录

第一部分：Web 开发环境的部署与配置	2
1. 下载 Tomcat 8.0，安装并配置环境变量	2
2. 安装并配置 MyEclipse	5
第二部分：JDBC 的使用	11
1. 设计工具类 DBUtil.java，实现对数据库数据的增删改	11
2. 调用实现的工具类，对数据库进行操作	14
3. 在 SSMS 或者其他数据库管理系统中检查是否完成操作	15

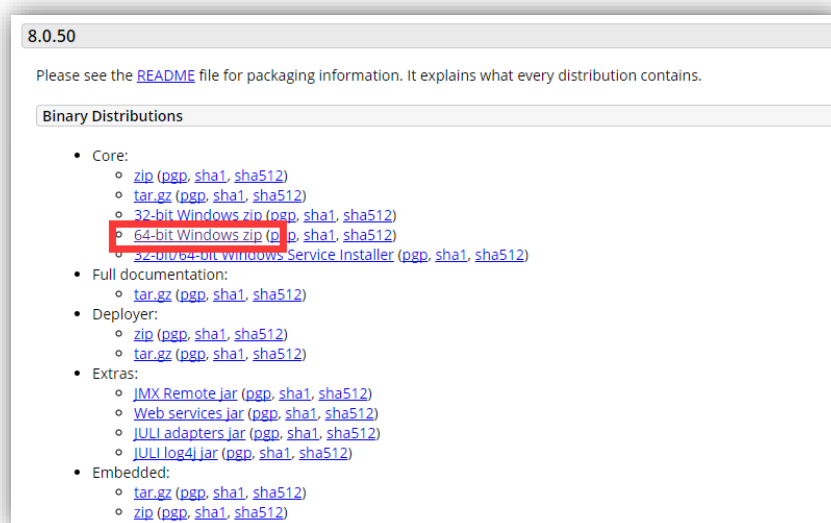
第一部分：Web 开发环境的部署与配置

1. 下载 Tomcat 8.0，安装并配置环境变量

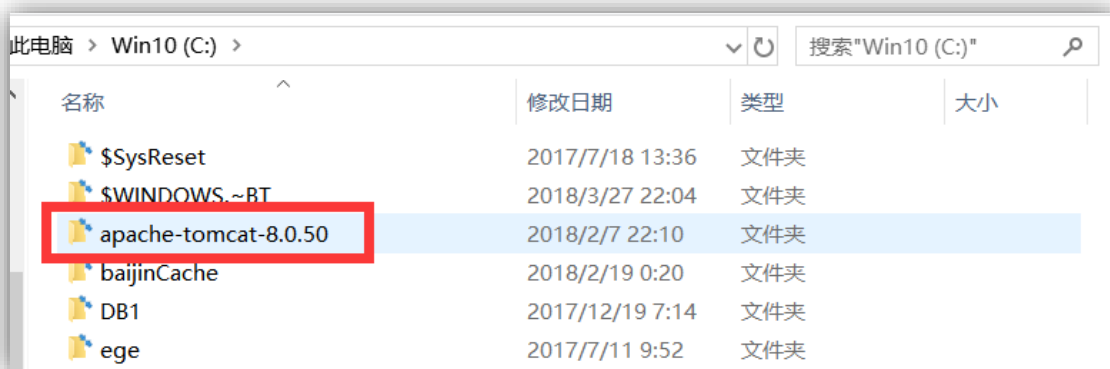
1. 登录 tomcat.apache.org，找到 Tomcat 8.0 版本，下载



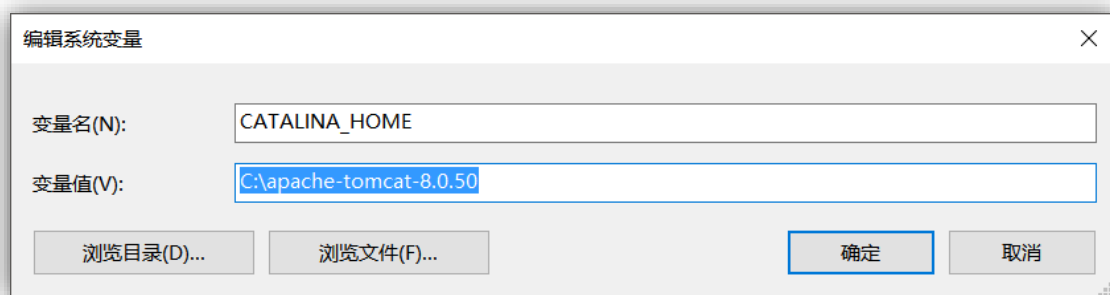
下载 64-bit (或 32-bit) Windows zip (压缩包版)



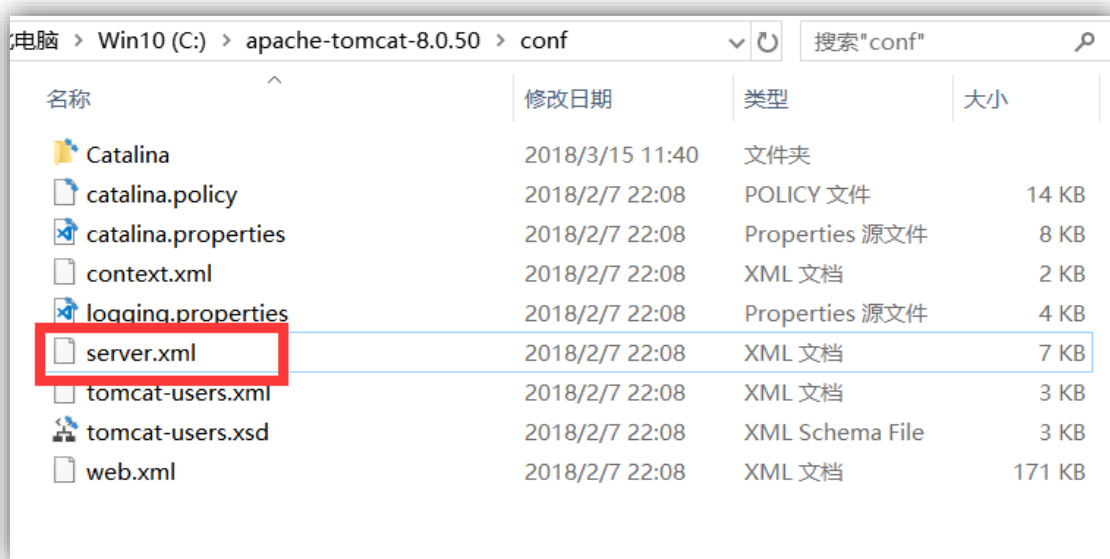
将下载后的压缩包解压至目标安装目录，如 C:\Tomcat8



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”。



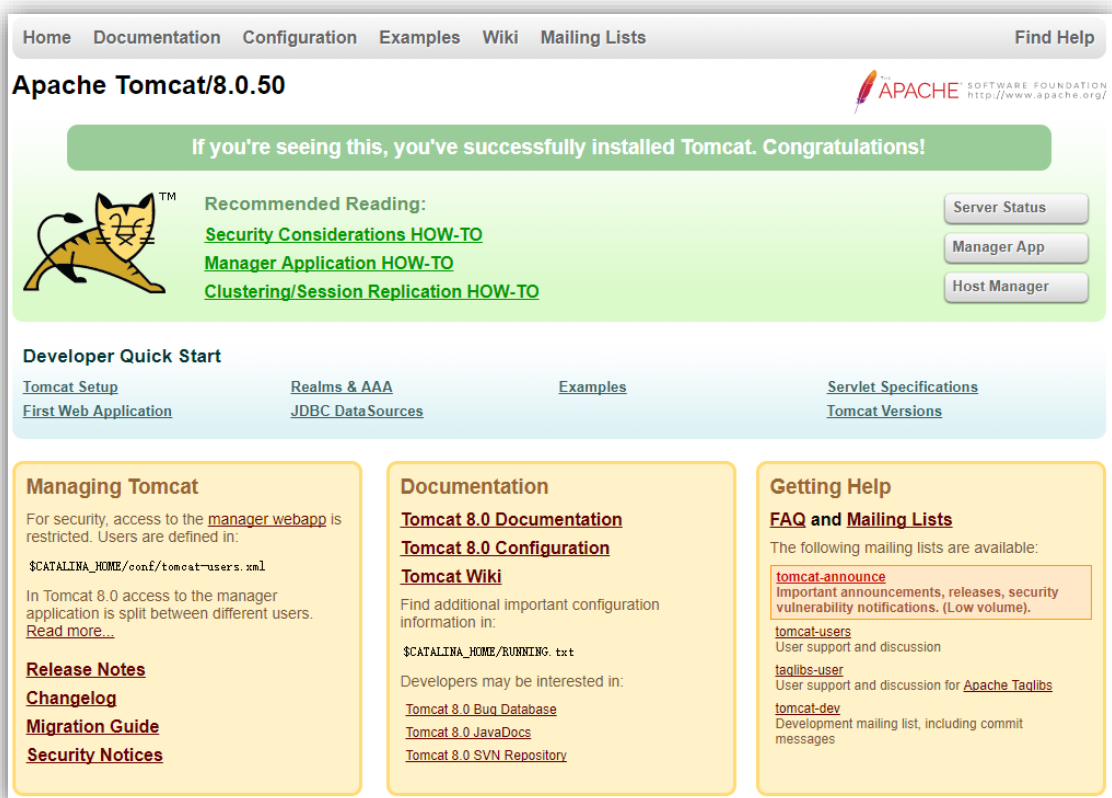
4. 在 server.xml 文件中，配置 Tomcat 服务器的端口号和首页等属性。



其中，port 为访问端口号，默认 8080。

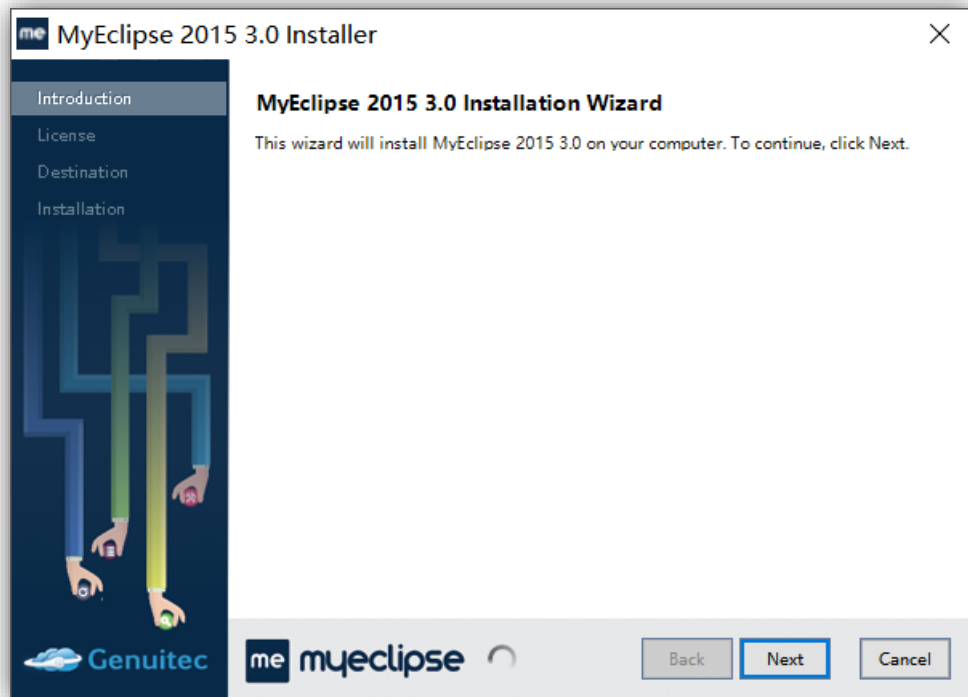
```
66 | APR (HTTP/AJP) Connector: /docs/apr.html
67 | Define a non-SSL/TLS HTTP/1.1 Connector on port 8080
68 |
69 | <!-->
70 | <Connector port="8080" protocol="HTTP/1.1"
71 |           connectionTimeout="20000"
72 |           redirectPort="8443" />
73 | <!-- A "Connector" using the shared thread pool -->
74 | <!-->
```

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



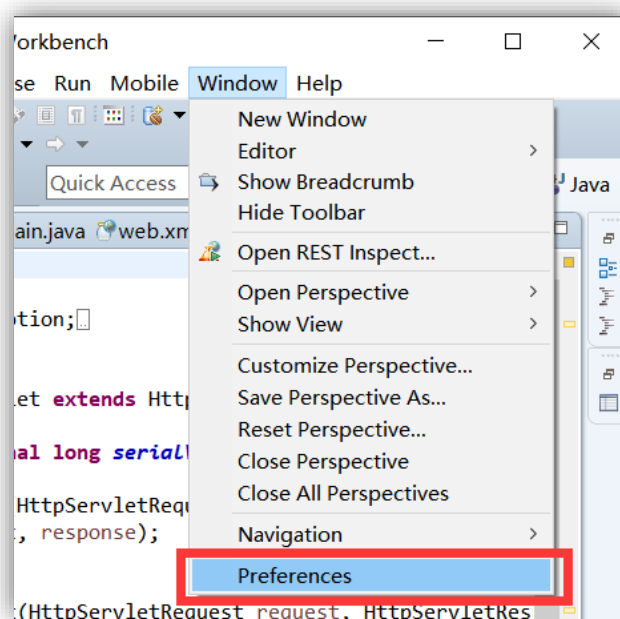
2.安装并配置 MyEclipse

1.打开 MyEclipse 安装文件，一路 next，最后安装成功。

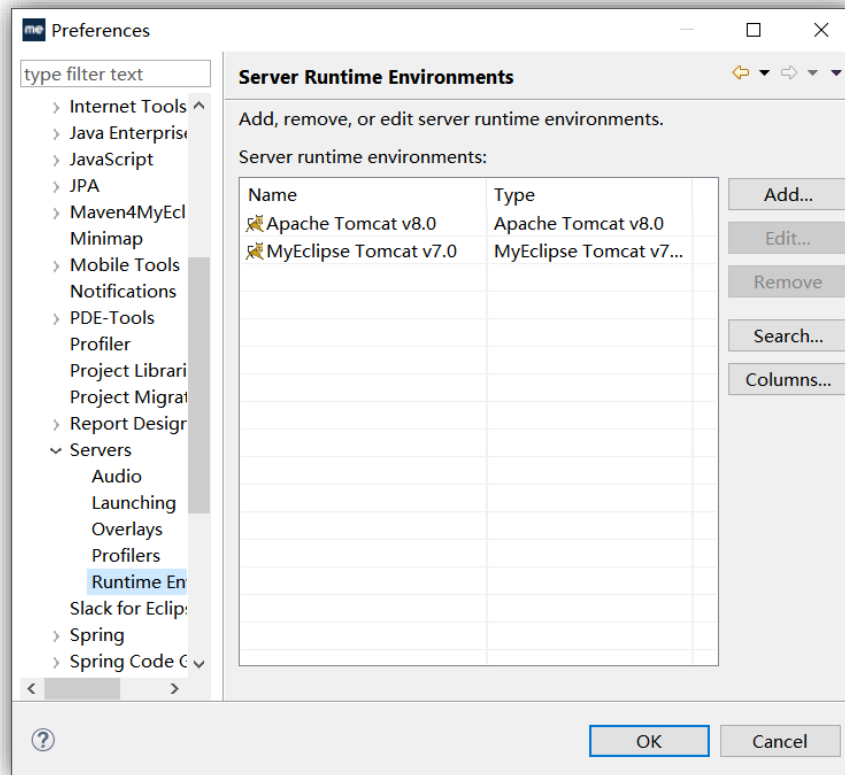


2.在 MyEclipse 环境配置刚刚安装的 Tomcat 8.0。

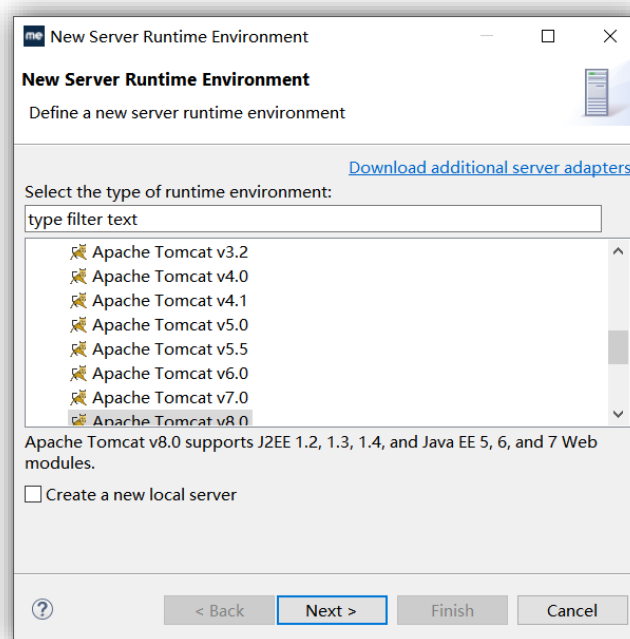
选择菜单栏 Window - Preferences



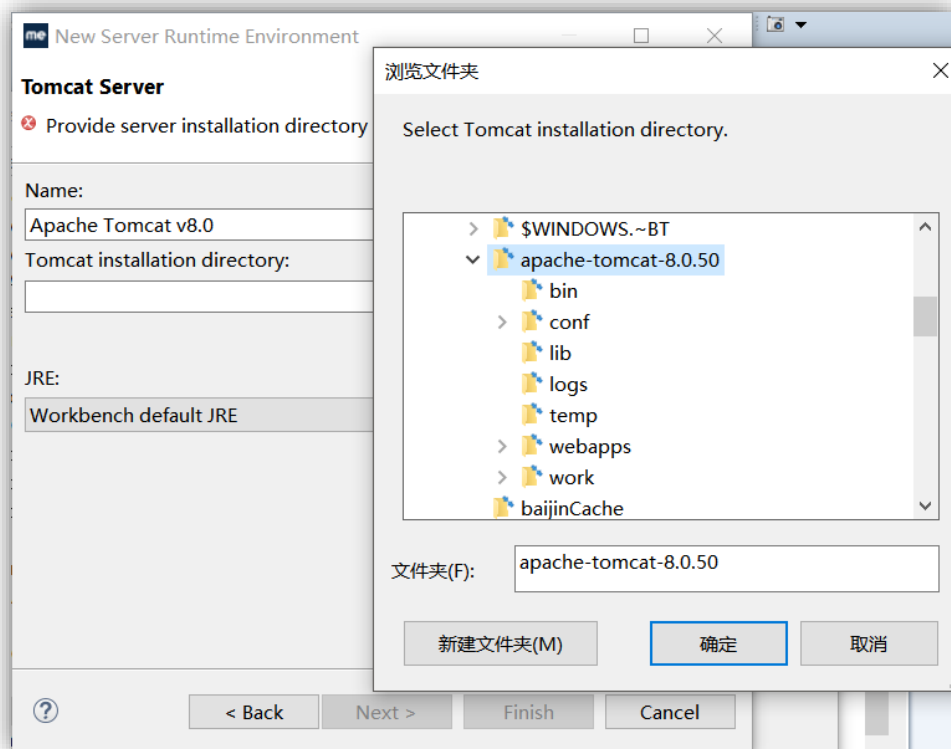
选择 Servers – Runtime Environments, 进入下图界面。



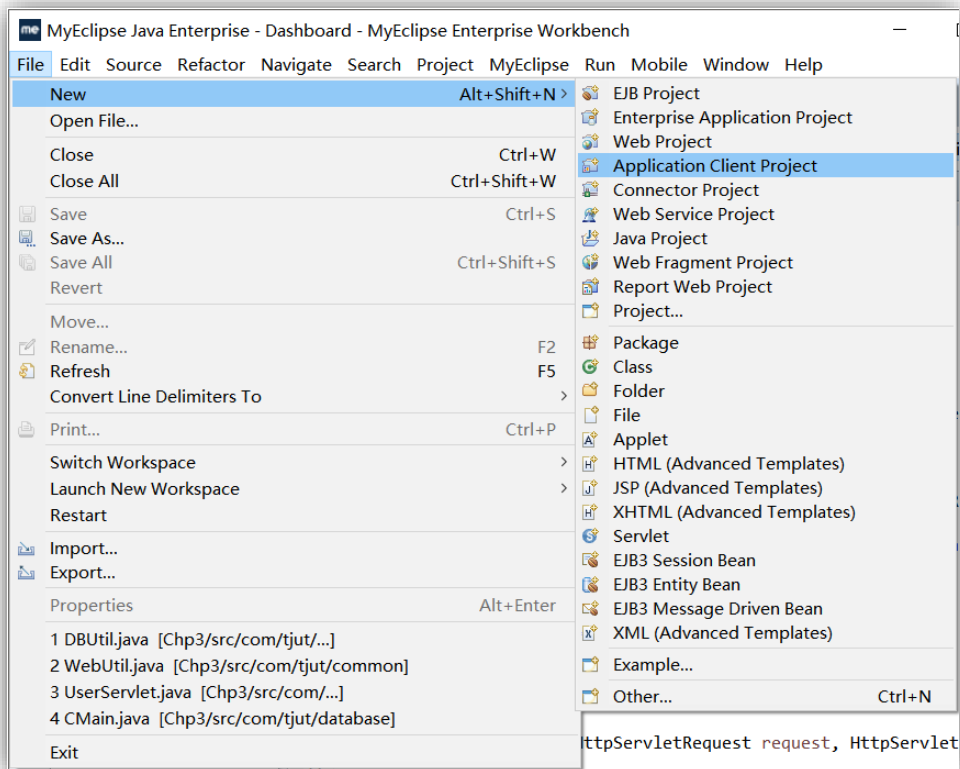
单击 Add, 在弹出的 “New Server Runtime Environment” 对话框中选择安装的 Apache Tomcat v8.0



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。

New Web Project

Create a JavaEE Web Project

Name cannot be empty.

Project name:

Project location

☒ Use default location

Location:

Project configuration

Java EE version:

Java version:

JSTL Version:

☐ Add maven support [Learn more about Maven4MyEclipse...](#)

Target runtime

EAR membership

☐ Add project to an EAR

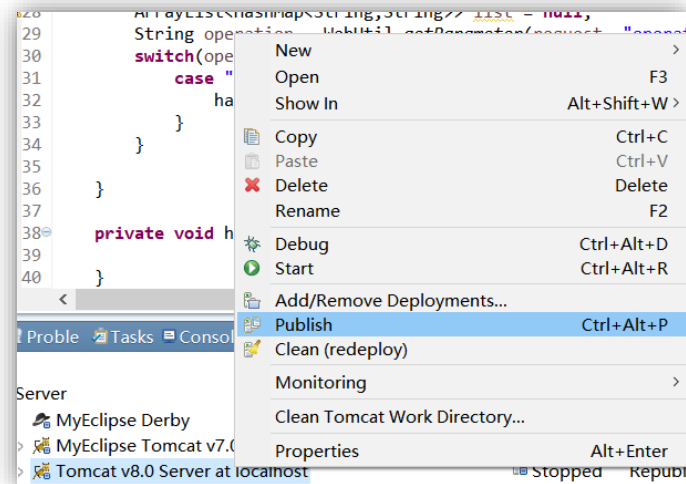
EAR project name:

Working sets

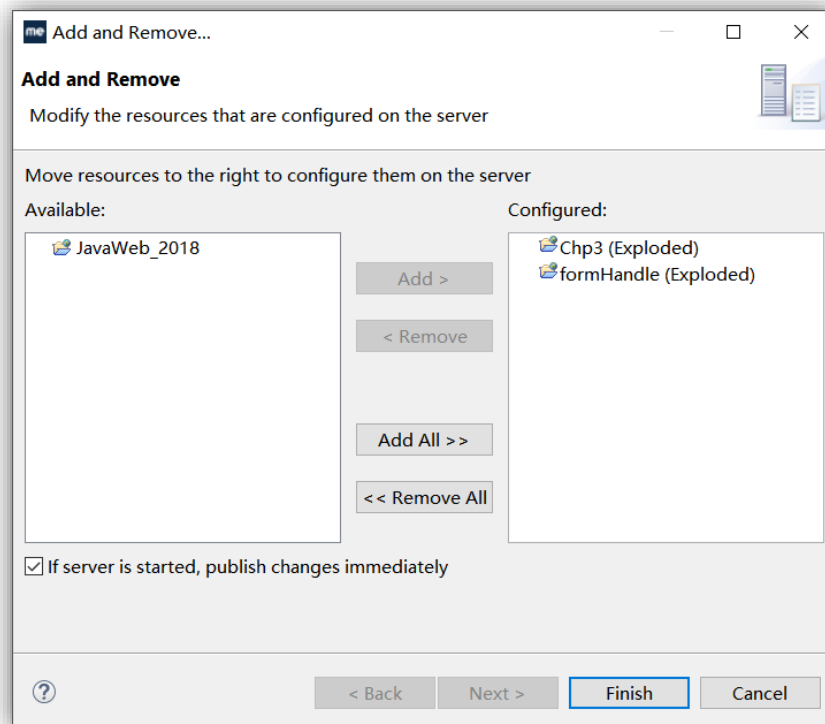
☐ Add project to working sets

Working sets:

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。

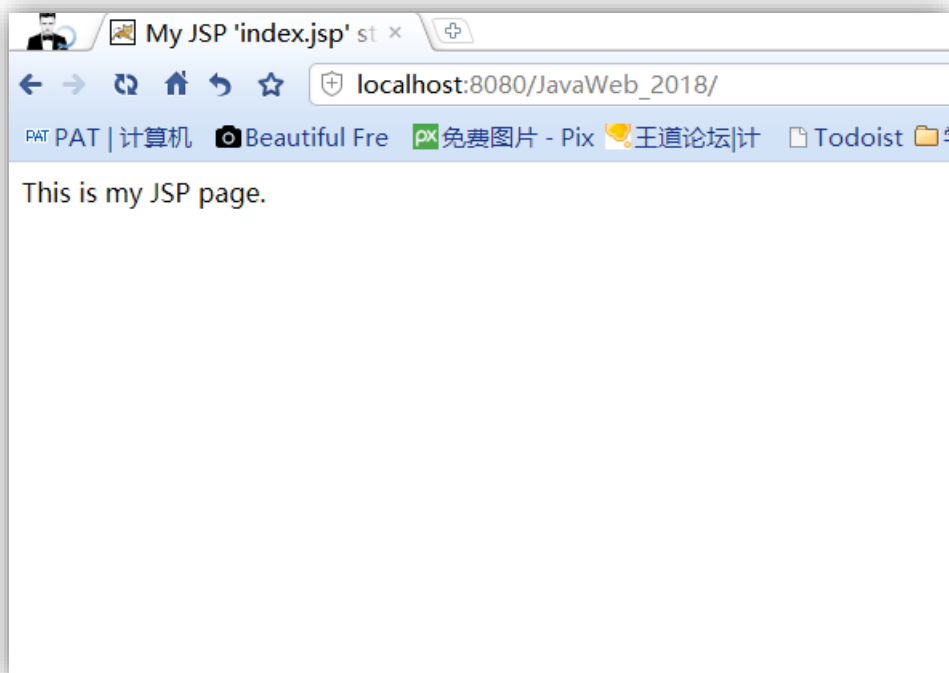


在弹出的对话框中，将 Available 栏的准备发布的项目添加到 Configured 中，点击 Finish



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。

```
1 <%@ page language="java" import="java.util.*" pageEncoding="ISO-8859-1"%>
2 <%
3 String path = request.getContextPath();
4 String basePath = request.getScheme()+ "://" + request.getServerName() +
5 %>
6
7 <!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN">
8 <html>
9 <head>
10 <base href="<%=basePath%>">
11
12 <title>My JSP 'index.jsp' starting page</title>
13 <meta http-equiv="pragma" content="no-cache">
14 <meta http-equiv="cache-control" content="no-cache">
15 <meta http-equiv="expires" content="0">
16 <meta http-equiv="keywords" content="keyword1,keyword2,keyword3">
17 <meta http-equiv="description" content="This is my page">
18 <!--|
19 <link rel="stylesheet" type="text/css" href="styles.css">
20 --|
21 </head>
22
23 <body>
24 This is my JSP page. <br>
25 </body>
26 </html>
```



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

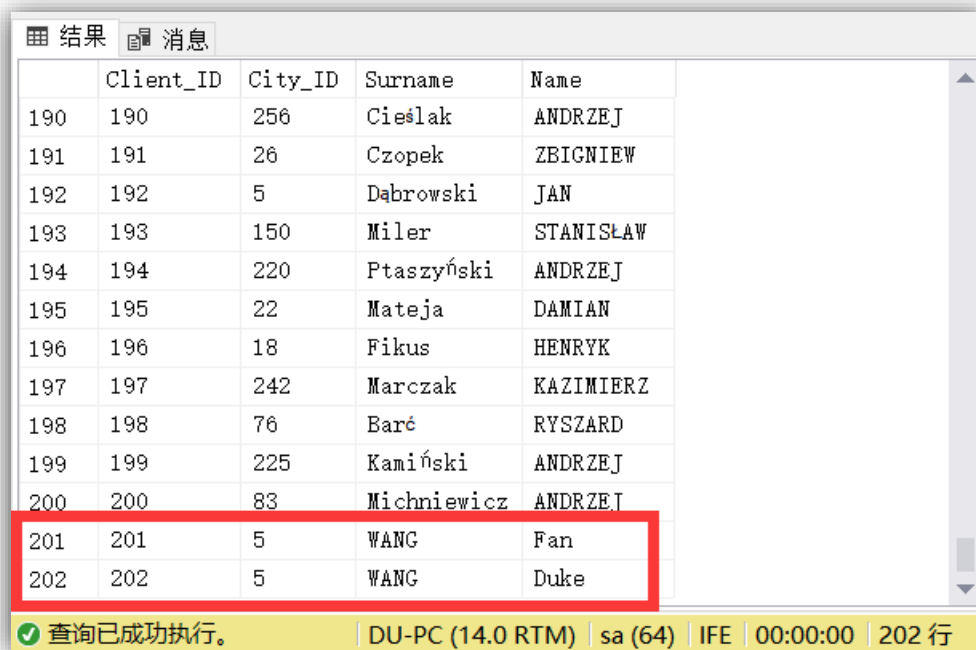
            sql = "DELETE FROM Clients WHERE Name = 'Fan' AND Surname =
'WANG'";
            list.add(sql);

            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：

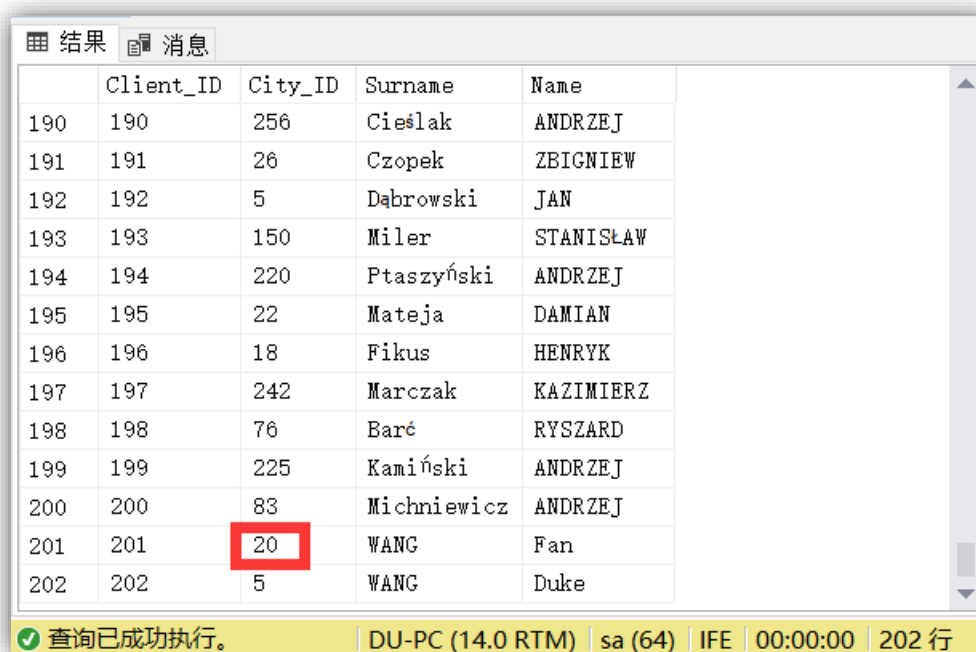
增加：



	Client_ID	City_ID	Surname	Name
190	190	256	Cieślak	ANDRZEJ
191	191	26	Czopek	ZBIGNIEW
192	192	5	Dabrowski	JAN
193	193	150	Miler	STANISŁAW
194	194	220	Ptaszyński	ANDRZEJ
195	195	22	Mateja	DAMIAN
196	196	18	Fikus	HENRYK
197	197	242	Marczak	KAZIMIERZ
198	198	76	Barć	RYSZARD
199	199	225	Kamiński	ANDRZEJ
200	200	83	Michniewicz	ANDRZEJ
201	201	5	WANG	Fan
202	202	5	WANG	Duke

✓ 查询已成功执行。 | DU-PC (14.0 RTM) | sa (64) | IFE | 00:00:00 | 202 行

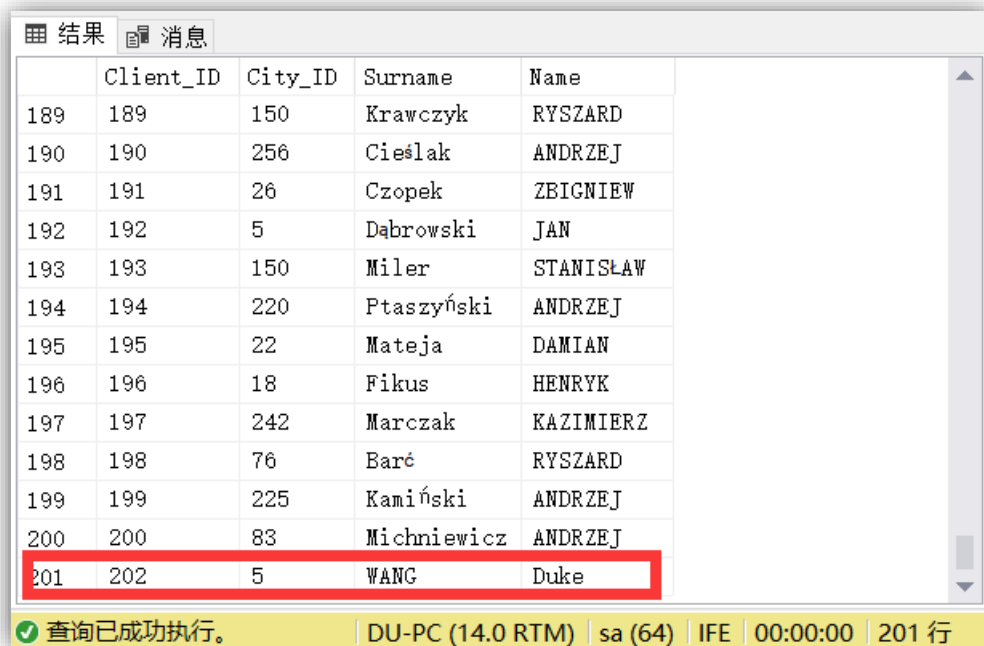
修改：



	Client_ID	City_ID	Surname	Name
190	190	256	Cieślak	ANDRZEJ
191	191	26	Czopek	ZBIGNIEW
192	192	5	Dabrowski	JAN
193	193	150	Miler	STANISŁAW
194	194	220	Ptaszyński	ANDRZEJ
195	195	22	Mateja	DAMIAN
196	196	18	Fikus	HENRYK
197	197	242	Marczak	KAZIMIERZ
198	198	76	Barć	RYSZARD
199	199	225	Kamiński	ANDRZEJ
200	200	83	Michniewicz	ANDRZEJ
201	201	20	WANG	Fan
202	202	5	WANG	Duke

✓ 查询已成功执行。 | DU-PC (14.0 RTM) | sa (64) | IFE | 00:00:00 | 202 行

删除:



	Client_ID	City_ID	Surname	Name	
189	189	150	Krawczyk	RYSZARD	
190	190	256	Cieślak	ANDRZEJ	
191	191	26	Czopek	ZBIGNIEW	
192	192	5	Dąbrowski	JAN	
193	193	150	Miler	STANISŁAW	
194	194	220	Ptaszyński	ANDRZEJ	
195	195	22	Mateja	DAMIAN	
196	196	18	Fikus	HENRYK	
197	197	242	Marczak	KAZIMIERZ	
198	198	76	Barć	RYSZARD	
199	199	225	Kamiński	ANDRZEJ	
200	200	83	Michniewicz	ANDRZEJ	
201	202	5	WANG	Duke	

✓ 查询已成功执行。 | DU-PC (14.0 RTM) | sa (64) | IFE | 00:00:00 | 201 行

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>
4. 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>
5. 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/>