数据库访问

本教程首先您已经了解了 JDBC 应用程序的工作方式。在您开始学习 Servlet 数据库之前，请访问 Java MySQL 连接来设置相关驱动及配置。

注意：

*你可以下载本站提供的 jar 包：**mysql-connector-java-5.1.39-bin.jar***

在 java 项目中，只需要在 Eclipse 中发布 mysql-connector-java-5.1.39-bin.jar 就可以运行 java 项目。但是在 Eclipse web 项目中，当执行 Class.forName（“ com.mysql.jdbc.Driver”）;时不会去查找驱动的。所以本实例中我们需要把 mysql-connector-java-5.1.39-bin .jar 拷贝到 tomcat 下一个 lib 目录。

从基本概念下手，让我们来创建一个简单的表，并在表中创建几条记录。

创建测试数据

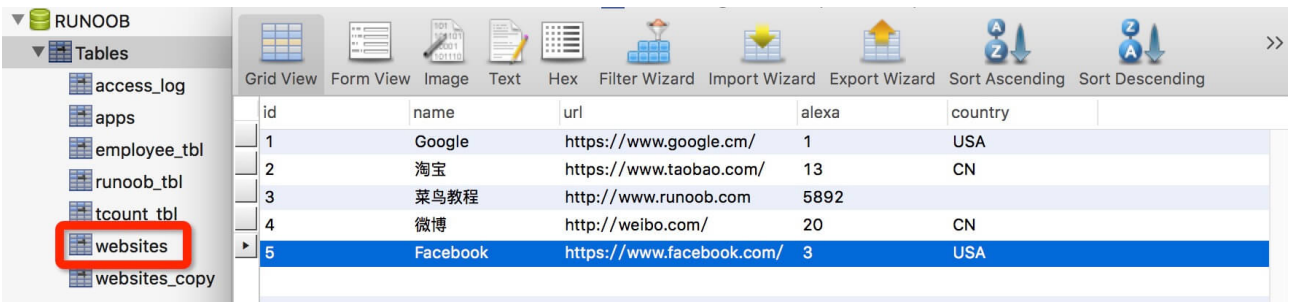
接下来我们在 MySQL 中创建 RUNOOB 数据库，并创建 website 数据表，表结构如下：

```
CREATE TABLE `websites` （`id` INT （11 ）NOT NULL AUTO_INCREMENT ，`name` 炭（20 ）NOT NULL DEFAUL
T '' COMMENT '站点名称' ，`url` VARCHAR （255 ）NOT NULL DEFAULT '' ，`Alexa 的` int （11 ）非空默认值
'0' 注释'Alexa 排名' ，`country char （10 ）非空默认值' 注释



  '国家' ，
 主键（`id` ））ENGINE = InnoDB AUTO_INCREMENT = 10 DEFAULT CHARSET = utf8 ;
```

插入一些数据：

```
插入“网站” 值（“ 1” ，“谷歌” ，“ https://www.google.cm/” ，“ 1” ，“美国” ），（“ 2” ，“淘宝” ，“
https: / /www.taobao.com/ ' '13' ，'CN' ），（ '3' ，'菜鸟教程' ，'http://www.runoob.com' ，
'5892' ，'' ），（ ' 4” ，“微博” ，“ http://weibo.com/” ，'20' ，'CN' ），（'5'  ，“
Facebook” ，“ https://www.facebook.com/” ，“ 3” ，“美国” ）；
```

数据表显示如下：



访问数据库

下面的实例演示了如何使用 Servlet 访问 RUNOOB 数据库。

```
包 com 。runoob 。测试;

导入 java 。io 。IOException ; 导入 java 。io 。PrintWriter ; 导入 java 。sql 。*;



导入 javax 。servlet 。ServletException ; 导入 javax 。servlet 。注解。WebServlet ; 导入 javax 。servl
et 。http 。HttpServlet ; 导入 javax 。servlet 。http 。HttpServletRequest ; 导入 javax 。servlet 。h
ttp 。HttpServletResponse ;




/ **
 * Servlet 实现类 DatabaseAccess
 * / @WebServlet （“ / DatabaseAccess” ）公共类 DatabaseAccess 扩展 HttpServlet { private static fin
al long serialVersionUID = 1L ; // JDBC 驱动程序名和数据库 URL 静态最终字符串 JDBC_DRIVER = “ com.mysq
l.jdbc.Driver” ; 静态最终字符串 DB_URL = “ jdbc：mysql：// localhost：3306 / RUNOOB” ;




    //数据库的用户名与密码，需要根据自己的设置 static final String USER = “ root” ; 静态最终字符串 PASS
= “ 123456” ; / **


     * @see HttpServlet＃HttpServlet（）
     * / public DatabaseAccess （）{ 超级（）; // TODO 自动生成的构造函数存根}
```

```
    / **
     * @see HttpServlet＃doGet（HttpServletRequest 请求，HttpServletResponse 响应）
     * / 受保护的 void doGet （HttpServletRequest 请求，HttpServletResponse 响应）抛出 ServletExcepti
on ， IOException { Connection conn = null ; 语句 stmt = null ; //设置响应内容类型
        response 。setContentType （“ text / html; charset = UTF-8” ）; PrintWriter out = 响应。get
Writer （）; 字符串标题=
```

```
“ Servlet Mysql 测试-菜鸟教程” ; 字符串 docType = “ <! DOCTYPE html> \ n” ; 出来。println （d
ocType + “ <html> \ n” + “ <head> <title>” + 标题+ “ </ title> </ head> \ n” + “ <body bgcolor = \”
＃f0f0f0 \“> \ n “ + ” <h1 align = \“ center \”>“ + 标题+ ” </ h1> \ n“ ）; 尝试{ // XML JDBC 驱动器
类。forName （“ com.mysql.jdbc。
```

```
            //：一个连接
            conn = DriverManager 。getConnection （DB_URL ， USER ， PASS ）;

            //执行 SQL 查询
            stmt = conn 。createStatement （）; 字符串 sql ;
            sql = “选择 ID，名称，来自网站的网址” ; ResultSet rs = stmt 。executeQuery （sql ）;
```

```
            //展开结果集数据库而（RS 。下一个（））{ //通过字段检索 INT ID   = RS 。getInt （“ id” ）;
字符串名称= rs 。getString （“ name” ）; 字符串 url = rs 。getString （“ url” ）;
```

```
                //输出数据出来。println （“ ID：” + id ）; 出来。println （“，站点名称：” + name ）;
出来。println （“，站点 URL：” + url ）; 出来。println （“ <br />” ）; } 出。println （“ </ body> </ h
tml>” ）;









            //完成后关闭
            rs 。close （）;
            stmt 。close （）;
            康涅狄格州。close （）; } catch （SQLException se ）{ //处理 JDBC 错误
            se 。printStackTrace （）; } catch （Exception e ）{ //处理 Class.forName 错误
            e 。printStackTrace （）; } 最后{ //最后是用于关闭资源的块尝试{ 如果（









                stmt ！= null ）
                stmt 。close （）; } catch （SQLException se2 ）{ } 尝试{ if （conn ！= null ）
                conn 。close （）; } catch （SQLException se ）{
                se 。printStackTrace （）; } }









    }

    / **
     * @see HttpServlet＃doPost（HttpServletRequest 请求，HttpServletResponse 响应）
     * / 受保护的 void doPost （HttpServletRequest 请求，HttpServletResponse 响应）抛出 ServletExcept
ion ，IOException { // TODO 自动生成的方法 stub
        doGet （请求，响应）; } }
```

现在让我们来编译上面的 Servlet，并在 web.xml 文件中创建以下压缩：

```
.... 的<servlet> < servlet 的- 名称> DatabaseAccess </ servlet 的- 名称> < servlet 的- 类> COM 。runoo
b 。测试。DatabaseAccess </ servlet 的- 类> </ servlet 的> < servlet 的- 映射> < servlet 的- 名称> Data
baseAccess </ servlet 的- 名称> < URL




        - 图案> / TomcatTest / DatabaseAccess </ URL - 图案> </ servlet 的- 映射> ....
```

现在调用这个 Servlet，输入链接：http：// localhost：8080 / TomcatTest / DatabaseAccess，将显示以下响应结果：