

Day07_JDBC

JDBC概念

Java Database Connective
其实就是使用java语言来操纵数据库
是sun公司定义的一套接口，用于操作数据库

java连接数据库

JDBC快速入门

1. 导入驱动jar包
2. 注册驱动
`Class.forName("com.mysql.jdbc.Driver");`
3. 获取数据库连接对象Connection
`Connection conn = DriverManager.getConnection(URL, username, password);`
4. 获取执行sql语句的对象
也就是传说中的sql执行者对象
`Statement stmt = conn.createStatement();`
5. 创建sql
`String sql = ...;`
6. 执行sql
`int count = stmt.excute(sql);`
7. 处理返回结果
`System.out.println(count);`
8. 释放资源
`stmt.close();`
`conn.close();`

`com.mysql.jdbc.Driver`
在这个类中有一个静态代码库

当类一加载进内存，就会自动执行
👉 拓展： 这句话在mysql5之后的驱动jar 可以不写

JDBC详解每一个类

DriverManager

获取数据库连接

驱动管理类

注册驱动

`Class.forName("com.mysql.jdbc.Driver");`

Connection: 数据库连接对象

`Connection conn = DriverManager.getConnection(URL, User, Password);`

URL

URL组成部分

jdbc: 主协议
mysql:// 子协议
localhost 主机名
ip
:3306 端口号
/stu 数据库名

URL连接的是本机的数据库时，可简写为/// `jdbc:mysql:///stu`

User

用户名

Password

密码

获取执行sql语句的对象

`Statement createStatement()` 获取statement对象

`PrepardStatement prepareStatement(String sql)` 获取preparedStatement对象
预处理sql语句的对象(防止sql注入)

管理事务

开启

`setAutoCommit(false)`

提交

`commit()`

回滚

`rollback()`

Statement: 执行sql语句对象

`boolean execute(String sql)` 执行所有sql

`int executeUpdate(String sql)` 执行 增，删，改 的sql
返回值int 代表该条sql影响的行数

`ResultSet executeQuery(String sql)`

执行查询语句

返回值

ResultSet

结果集对象
`next()` 判断有没有下一条记录
如果有，游标向下移动一位

获取数据
`getXXX()` 参数
int 列的编号
从1开始
String 列名
Subtopic

preparedStatement

执行sql语句对象。预编译sql语句对象

防止sql注入

提高效率

1. 定义sql sql语句中的参数使用? 作为占位符

2. 创建preparedStatement对象 需要传入sql语句

3. 给占位符? 赋值 `pstmt.setXxx(参数1, 参数2)`
参数1 ? 出现的位置
从1开始
参数2 具体的值

4. 执行sql 执行的方法不需要再传递sql语句了