第三天

了解Servlet运行和tomcat容器创建，关于xml语法的讲解。

```
//写在<web-app>中。
 <servlet>
                <servlet-name>hello</servlet-name>
                 //servlet名
                <servlet-class>web.HelloServlet</servlet-class>
                 //反射的Java类，指定到本地文件位置。
</servlet>
 <servlet-mapping>
                <servlet-name>hello</servlet-name>
                <url-pattern>/hello</url-pattern>
                 //导向到对应网址下
 </servlet-mapping>

```

接受前端浏览器传递的参数：

```
request.getParameter()//request对象，请求对象
```

参数传递的方式，浏览器传递给服务器。

通讯协议

> 协议：TCP、IP、UDP

http协议(超文本传输协议)--应用层协议，定义了web和服务器之间的通讯。

数据包的结构：请求数据包结构（浏览器->服务器）

1. 请求行（数据包第一行）
    a）请求方式（get/post） 刷题记录（和之前的CTF-web有关联）
    b）请求资源路径
    c）协议的类型和版本、

2.若干消息头（www命名的一些有特殊含义的键值对如：content-type= text/html）服务器和浏览器都会遵守这些消息头，消息头一般由浏览器和服务器自动生成，也可以手动编程。

响应数据包结构：第一部分：状态行、协议的类型与版本、状态码（数字）404、200、500.

第二部分：若干消息头

第三部分：实体内容，服务器返回给浏览器的处理结果。

输出传输中的中文乱码问题（重点）

（ASCII）当表单出现乱码时，在html文档中添加，背后的原因是因为编码方式不同。

```
<meta http-equiv="content-type"
content="text/html;charset-utf-8">
```

在servlet添加:

```
request.setCharaterEncoding("utf-8")
```

Java连接数据库类型

```
Statement
PreparedStatement
//实战PreparedStatment比Statment更安全可以有效的防止SQl注入攻击
//而且PreparedSatment比Statment运行的效率
```

Statement实例:

```
import java.sql.Connection;
import java.sql.ResultSet;
import java.sql.Statement;
import java.util.Scanner;

public class DBUtilTest {
   public static void main(String[] args) {
      //System.out.println(DBUtil.getConnection());
      //DBUtil是JDBC中的一个工具
      Scanner sc=new Scanner(System.in);
      //获取用户名
      String name=sc.nextLine();
      //获取用户密码
      String pwd=sc.nextLine();
      //System.out.println(name+":"+pwd);
      //获取数据库链接
      Connection conn=DBUtil.getConnection();
      try {
```

```
            //获取数据库语句操作对象
            Statement state=conn.createStatement();
            //定义sql语句
            String sql="select * from emp where uname='"+name+"' and pwd='"+pwd+"'";
            //让Statment对象向数据库发送上面的sql语句 并获取查询语句 查询的结果集对象
            ResultSet rs=state.executeQuery(sql);
            //从结果及中查询出对应的数据信息
            //判断是否查询到了结果
            while(rs.next()){
               System.out.println("有结果");
            }
        } catch (Exception e) {
            // TODO Auto-generated catch block
            e.printStackTrace();
        }
    }
}
```

PreparedStatement实例

```
import java.sql.Connection;
import java.sql.PreparedStatement;
import java.sql.ResultSet;
import java.sql.SQLException;
import java.util.Scanner;

public class DBUtilTest2 {
    public static void main(String[] args) {
        Scanner sc=new Scanner(System.in);
        //获取用户名
        String name=sc.nextLine();
        //获取用户密码
        String pwd=sc.nextLine();
        //System.out.println(name+":"+pwd);
        //获取数据库链接
        Connection conn=DBUtil.getConnection();
        //定义sql语句
        String sql="select * from emp where uname=? and pwd=?";
        //获取数据库语句操作对象PreparedStatment
```

```
20          try {
21              PreparedStatement prep=conn.prepareStatement(sql);
22              //为两个占位符传递参数
23              prep.setString(1, name);
24              prep.setString(2, pwd);
25              //PreparedStatement 向数据库发送sql语句并执行  获取返回的结果集对象
26              ResultSet rs=prep.executeQuery();
27              while(rs.next()){
28                  System.out.println("有数据");
29              }
30          } catch (Exception e) {
31              // TODO Auto-generated catch block
32              e.printStackTrace();
33          }
34      }
35  }
```