第七章 Jmeter 高级编程

一、 JMeter 内置函数

- 以两个下划线开头。
- 函数区分大小写。
- ${__char(ascii1,ascii2,...)}
  - ✓ 返回指定 ascii 的字符
- ${__machineIP(存入变量名)}
  - ✓ 返回本机 IP
  - ✓ 若省略变量名，则直接输出 IP
- ${__threadNum}
  - ✓ 返回当前线程号
  - ✓ 函数后的括号可以省略
- ${__time(格式，存入变量)}
  - ✓ 直接使用返回 1970/1/1 至今的秒数
  - ✓ 获取日期时间，Y 年，MM 月，dd 日，hh，mm，ss
    - ✧ 格式不必加引号
- ${__UUID}
  - ✓ 生成一个唯一的字符串
- ${__Random(初值，终值，存入变量名)}
  - ✓ 生成随机数
- ${__RandomString(length,seed,variable)}
  - ✓ 用于生成随机字符串。
  - ✓ length
    - ✧ 指定字符串长度。
  - ✓ seed
    - ✧ 字符串种子（基于这些字符自由组合成将来的字符串）。
  - ✓ variable
    - ✧ 生成的字符串存入此变量。

二、 Jmeter 访问 MySQL 数据库

- 加载数据库驱动包
  - ✓ 点击测试计划-->点击中间底部"浏览"-->选中 mysql 驱动 jar 包-->打开
- 配置数据库连接参数
  - ✓ 配置元件→JDBC Connection Configuration
    - ✧ 通常加到线程组前面
    - ✧ Variable Name
      - ● 输入数据库连接名
    - ✧ Validation Query
      - ● Select 1
        - ■ 表示检查 select 语法
    - ✧ Database URL

- jdbc:mysql://localhost:3306/数据库名
  - JDBC Driver class
    - com.mysql.jdbc.Driver
  - ✓ 不同数据库的 URL 和驱动程序不同。
- ➢ 添加 JDBC Request
  - ✓ Variable Name
    - 即前面的数据库连接名
  - ✓ Query Type
    - select 用于查询，update 用于插入和更新（含删除）
  - ✓ 其它设置保持默认
- ➢ Prepared（预编译查询）
  - ✓ 在 sql 语句中使用“?”代替实际数据，将来使用参数数据替换“?”
  - ✓ Parameter values
    - 参数值，多个用逗号间隔，将来替换 sql 语句中的“?”
  - ✓ Parameter types
    - 参数的类型
    - 必填，且与参数值个数要一致
  - ✓ Variables names
    - 省略时，与表中列名相同
    - 后续若要使用参数名，则不能省略，以后可以使用${变量名_1}、${变量名_2}等访问，数字表示行号，不需要记录集的名字
  - ✓ Result variable name
    - 结果集的名字
    - 访问：vars.getObject("rs").get(0).get("uname")
      - rs 表示记录集名称
      - 0 表示第 1 行
      - uname 表示列名

三、 测试 Java 程序

1 编译软件

- ➢ Jmeter 没有自带编译器，需要借助第三方编译器才能实现。
- ➢ Eclipse 导入包
  - ✓ \apache-jmeter-3.0\lib
    - avalon-framework-4.1.4.jar
    - jorphan.jar
    - logkit-2.0.jar
  - ✓ \apache-jmeter-3.0\lib\ext
    - ApacheJMeter_core.jar
    - ApacheJMeter_java.jar
  - ✓ 以上包需要全部导入

2 被测程序

```
public class Add{
    public int sum(int a,int b)
    {
        return a+b;    r
    }

    public static void main(String args[]){
        Add math = new Add();
        System.out.println(math.sum(1, 2));
    }
}
```

3 测试程序

```
public class JavaTest extends AbstractJavaSamplerClient {
    private Add test=null;
    private String a;
    private String b;
    private String result;

    //定义参数
    public Arguments getDefaultParameters() {
        Arguments params = new Arguments();
        params.addArgument("num1", "");//参数 num1，初值为空，jmeter 可更改此值
        params.addArgument("num2", "");
        return params;
    }

    //每个线程测试前执行一次，做一些初始化工作，取参数值
    public void setupTest(JavaSamplerContext arg0) {
        test = new Add();
        a = arg0.getParameter("num1");//获取参数值
        b = arg0.getParameter("num2");
    }

    //开始测试，从 arg0 参数可以获得参数值
    public SampleResult runTest(JavaSamplerContext arg0) {
        SampleResult sr = new SampleResult();//取样器
        sr.setSamplerData("请求参数 num1："+a+"\n 请求参数 num2："+b);//发送请求
        try {
            sr.sampleStart();//类似于事务，开始统计响应时间
            result = String.valueOf(test.sum(Integer.parseInt(a), Integer.parseInt(b)));
```

```
            if (result != null && result.length() > 0) {
                //下面的操作将响应数据输出到 Jmeter 的察看结果树中
                sr.setResponseData("结果是："+result, "utf-8");
                sr.setDataType(SampleResult.TEXT);
                sr.setResponseCode("200");
                sr.setResponseOK();
            }
            sr.setSuccessful(true);
        } catch (Throwable e) {
            sr.setSuccessful(false);
            e.printStackTrace();
        } finally {
            sr.sampleEnd();     //结束统计响应时间标记
        }
        return sr;
    }

    //测试结束时调用，只执行一次
    //public void teardownTest(JavaSamplerContext arg0) {
    //}
    //必须加入下面的 main 方法编译，导出代码时，必须禁用 main
    public static void main(String[] args) {
        Arguments pms = new Arguments();
        pms.addArgument("num1", "1");//设置参数，赋值 1
        pms.addArgument("num2", "2");//设置参数，赋值 2
        JavaSamplerContext arg0 = new JavaSamplerContext(pms);
        JavaTest test = new JavaTest();
        test.setupTest(arg0);
        test.runTest(arg0);
        test.teardownTest(arg0);
    }
}
```

4 导出包

- 在测试程序中加入 main 方法，然后运行正确，否则不能导出包。
- 务必禁用测试程序中的 main 方法，然后保存代码。
- 右击项目→导出→Java→可运行的 JAR 文件（导出时要关 JMeter）
  - ✓ 启动配置
    - ✧ 选择测试的类名
  - ✓ 导出目标
    - ✧ 必须放在\apache-jmeter-3.0\lib\ext 目录中，名字可与类名不同
  - ✓ 忽略错误
    - ✧ 找不到 main 方法

- Jmeter 操作
  - ✓ 导入包
  - ✓ 测试计划→Add directory or jar to classpath 处：浏览
  - ✓ 添加线程组
    - ✧ 添加 Java 请求
      - ● 在类名称中选择上面导出的包
      - ● 为参数赋值
    - ✧ 添加查看结果树并运行

四、 测试 BeanShell 请求

- 添加 BeanShell Sampler

```
public static String test(){
        vars.put("username","zhsan");       //保存参数
        vars.put("password","123456");
        return "success";
}
test();      //必须调用函数
```

- 添加 Debug Sampler
  - ✓ 可以尾部添加 ${username} 查看到变量
  - ✓ 在 Debug Sampler 的响应数据中可以看到 BeanShell 中的变量名与值
- 添加结果树看结果