

Jmeter 函数介绍

Jmeter函数是什么？

Jmeter中函数，就相当于我们小时候学的未知数X，在jmeter中函数的作用是用来替代某个一 需要计算的不确定的值，它可以用在几乎任何地方。在Jmeter中函数的表示方式如下：

`${_函数名(参数1,参数2,...参数n)}`

Jmeter有哪些函数

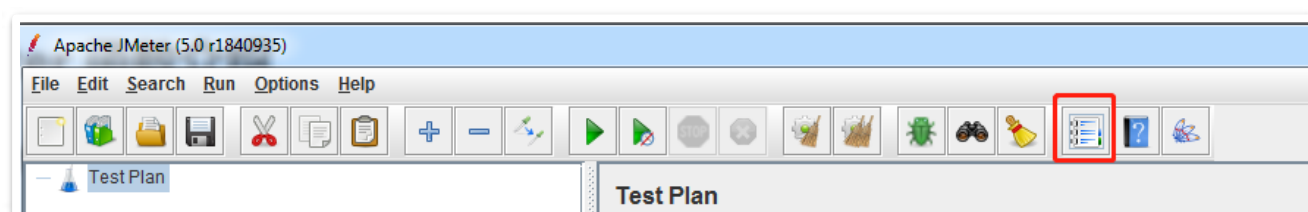
Jmeter内置了48个函数，以我们当前业务场景来说，使用到内置函数的频率不高，目前常用的可能就2个。一个是__time,另一个就是__digest,详细的函数列表可以到[函数列表](#)查询。

- __time用于获得当前时间，可以通过SimpleDateFormat来指定输出结果的格式，如返回年/月/日则函数书写为 `${_time(YYYY/MM/DD,)}` 在 脚本运行时 将得到 2019/01/04
- __digest用于字符加密，支持多种加密算法，如，对支付Abc进行MD5加密则书写格式为：`${_digest(MD5,Abc,,)}` 在 脚本运行时 将得到结果 35593b7ce5020eae3ca68fd5b6f3e031

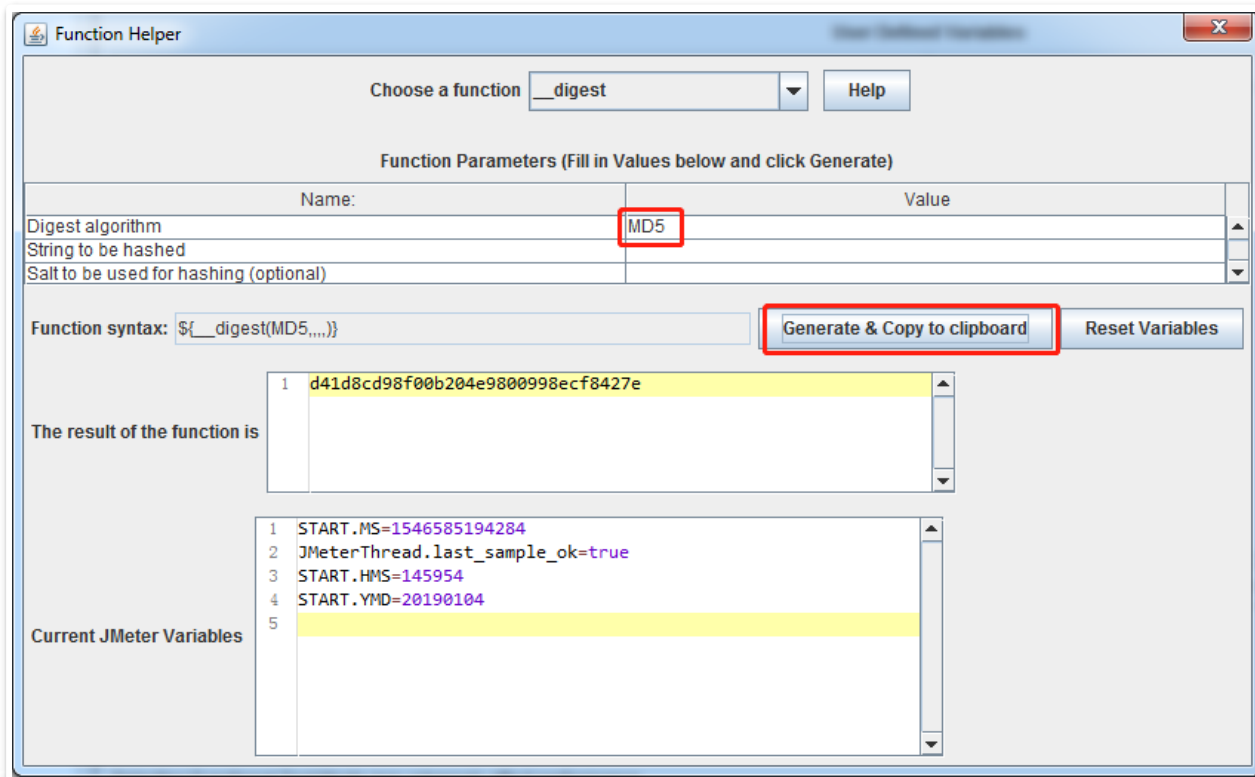
Jmeter函数如何使用？

函数在Jmeter中使用方式，就和普通值一样，只需要按照规定格式填写即可，下面举几个例子来进行简单说明。

- 在使用函数前，我们先简单了解下函数的含义。建议去查询[函数列表](#),明白函数的基本含义后，我们可以使用Jmeter内置的函数助手来进行函数的编写。



- 在函数助手中选择所需要的函数，并根据要求填写函数所需要的参数信息。



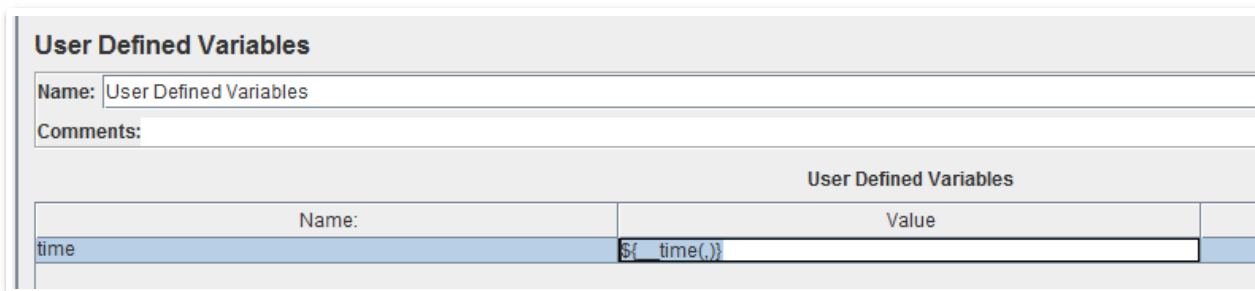
- 在脚本中使用需要的函数,使用方式如同Jmeter变量一样，在需要的地方直接填写即可，示例如下：

```

1  {"type":${type},
2  "timestamp": "${__time(,)}"
3  }

```

- 在用户定义变量中使用



- 示例脚本下载

内置函数不够了怎么办？

在有些时候，Jmeter内置函数可能是无法满足我们的业务需求，这个时候，我们就需要自己去编写函数，来简化脚本的编写。自己编写的函数和内置函数在用法上都是一样的。

Jmeter编写自定义函数

在自己编写Jmeter函数时需要注意以下几点：

- 自定义函数所在的包名应包括 `.functions` 且不包括 `.gui`
- 函数名需要以 `_` 开头
- 函数类需要继承AbstractFunction并实现其方法
 - getArgumentDesc-- 参数描述信息
 - execute-- 函数主体部分
 - setParameters-- 函数的参数
 - getReferenceKey-- 函数的名字

Tips：如果需要修改默认函数的识别方式（Classpath finder）可以到 `jmeter.properties` 去修改，但可能会导致其他第三方的插件失效，建议保持默认

```
1  # Classpath finder
2  # =====
3  # The classpath finder currently needs to load every single JMeter class to
   find
4  # the classes it needs.
5  # For non-GUI mode, it's only necessary to scan for Function classes, but all
   classes
6  # are still loaded.
7  # All current Function classes include ".function." in their name,
8  # and none include ".gui." in the name, so the number of unwanted classes
   loaded can be
9  # reduced by checking for these. However, if a valid function class name does
   not match
10 # these restrictions, it will not be loaded. If problems are encountered,
   then comment
11 # or change the following properties:
12 classfinder.functions.contain=.functions.
13 classfinder.functions.notContain=.gui.
14
```

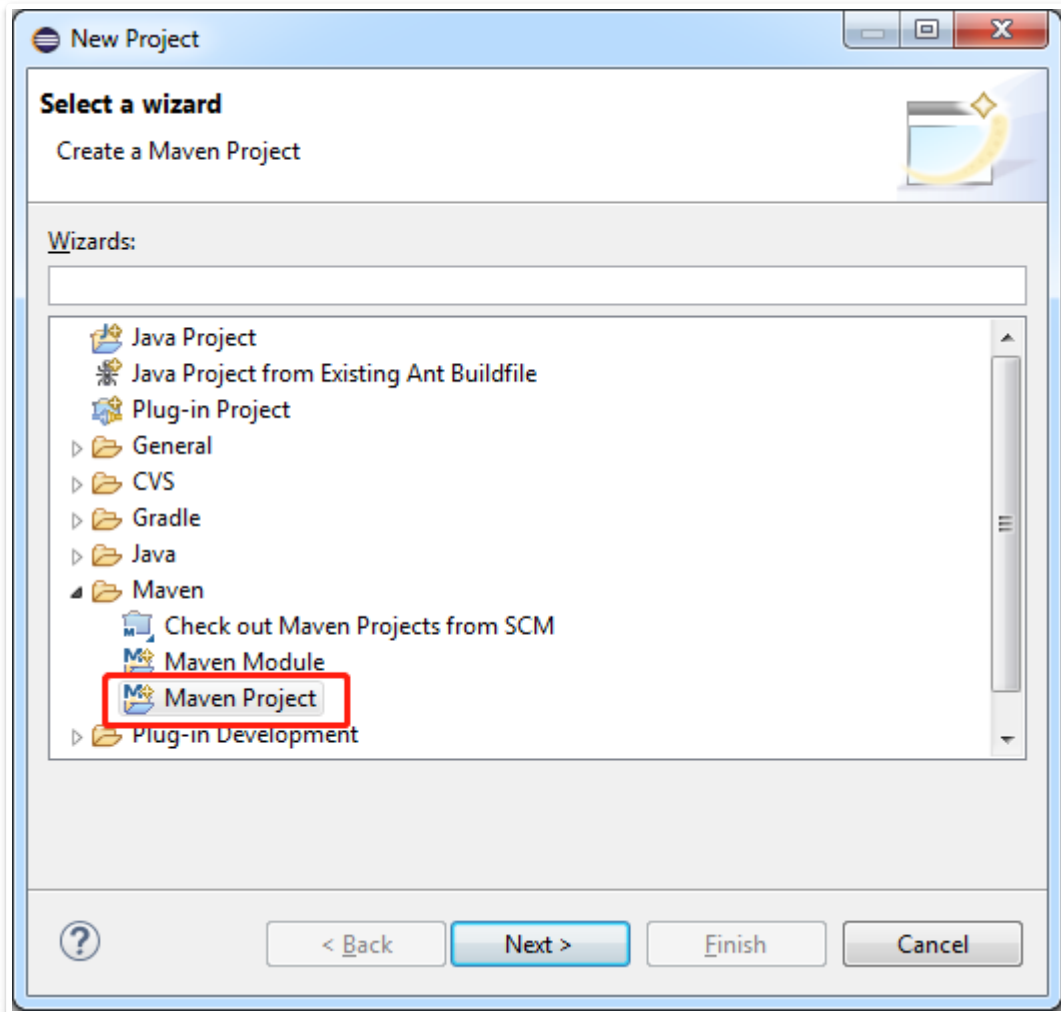
如何编写自定义函数

工具准备

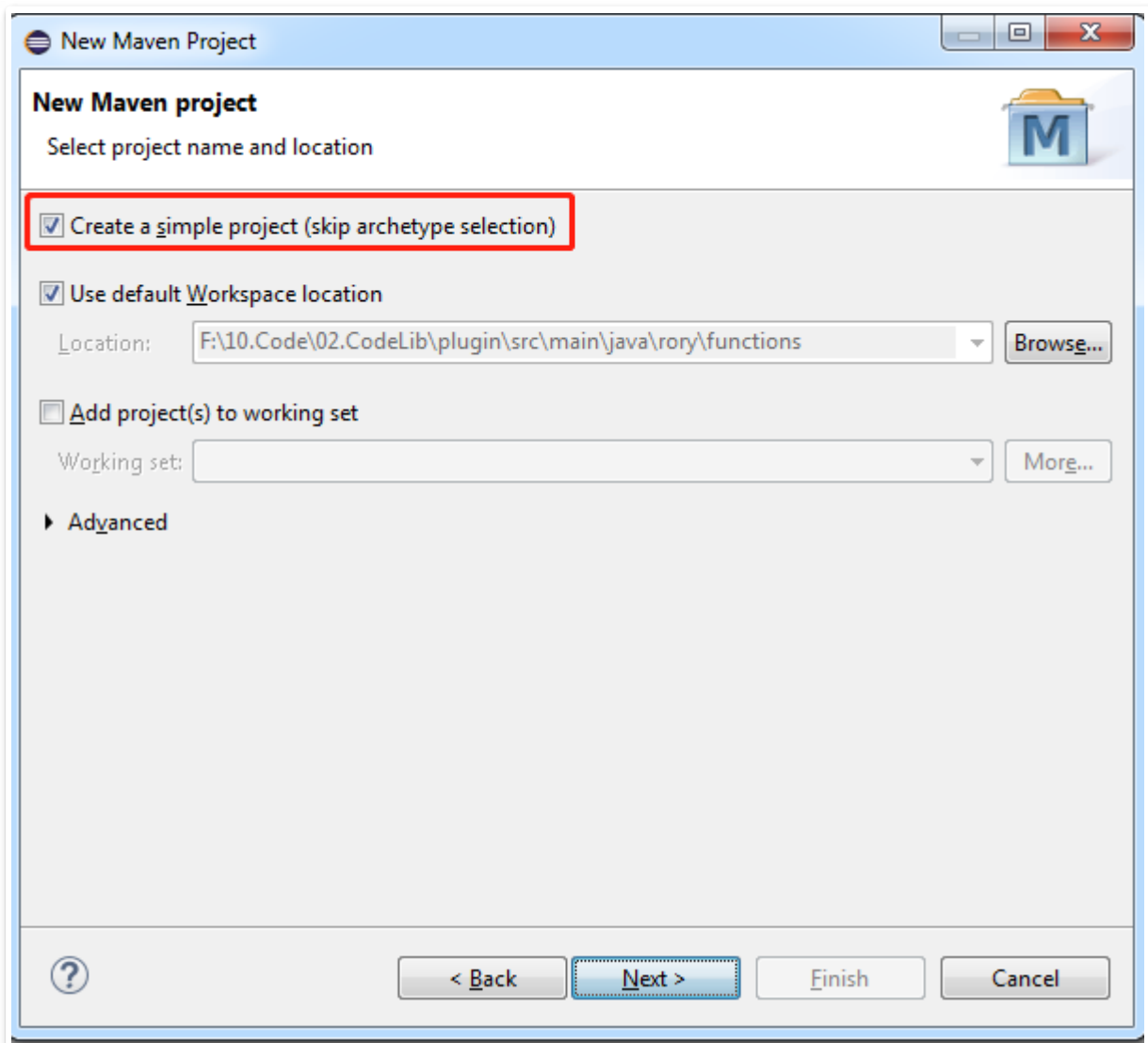
在开始编写自定义函数前，你需要先下载一个JAVA IDE，这里使用Eclipse

新建工程

- 在Eclipse中新建一个Maven工程File-New-Project...



- 勾选Create a simple project



- 输入Group Id和Artifact id 然后点击Finish

New Maven Project

Configure project

Artifact

Group Id: demo

Artifact Id: demo

Version: 0.0.1-SNAPSHOT

Packaging: jar

Name:

Description:

Parent Project

Group Id:

Artifact Id:

Version:

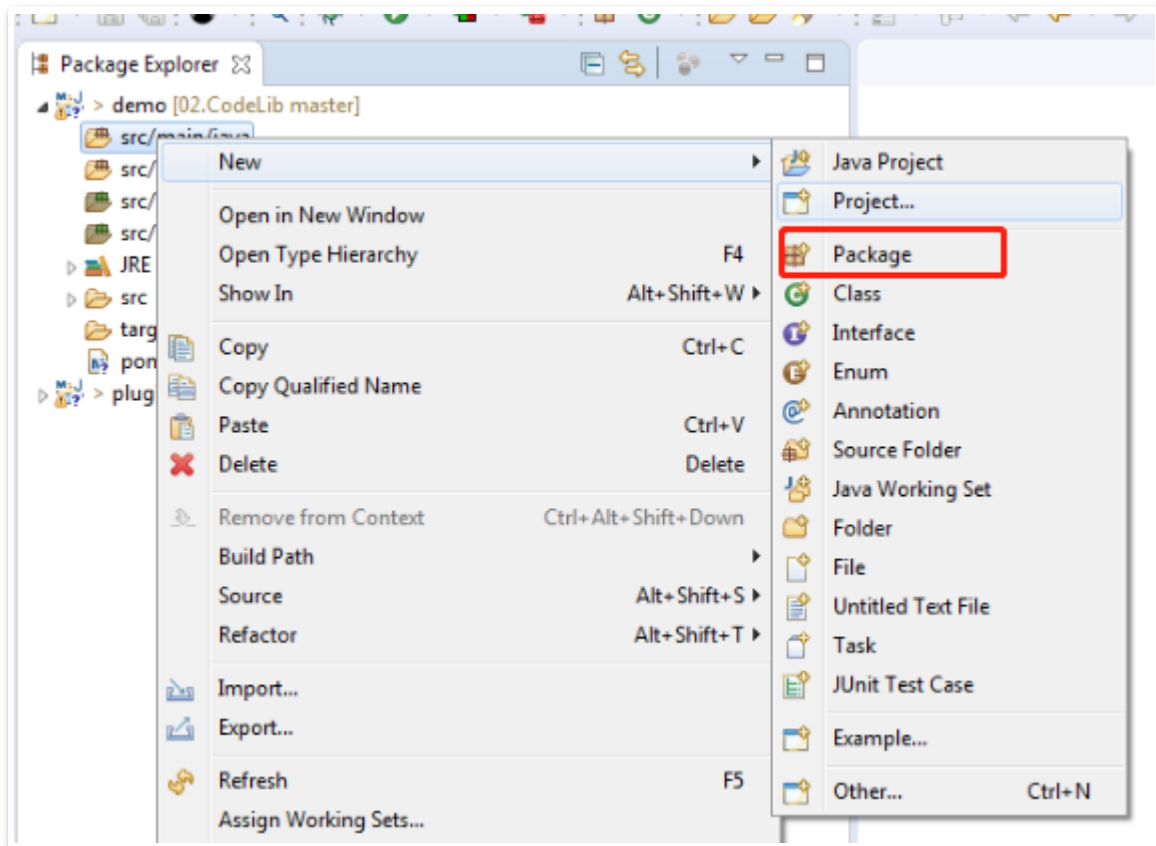
Browse... Clear

Advanced

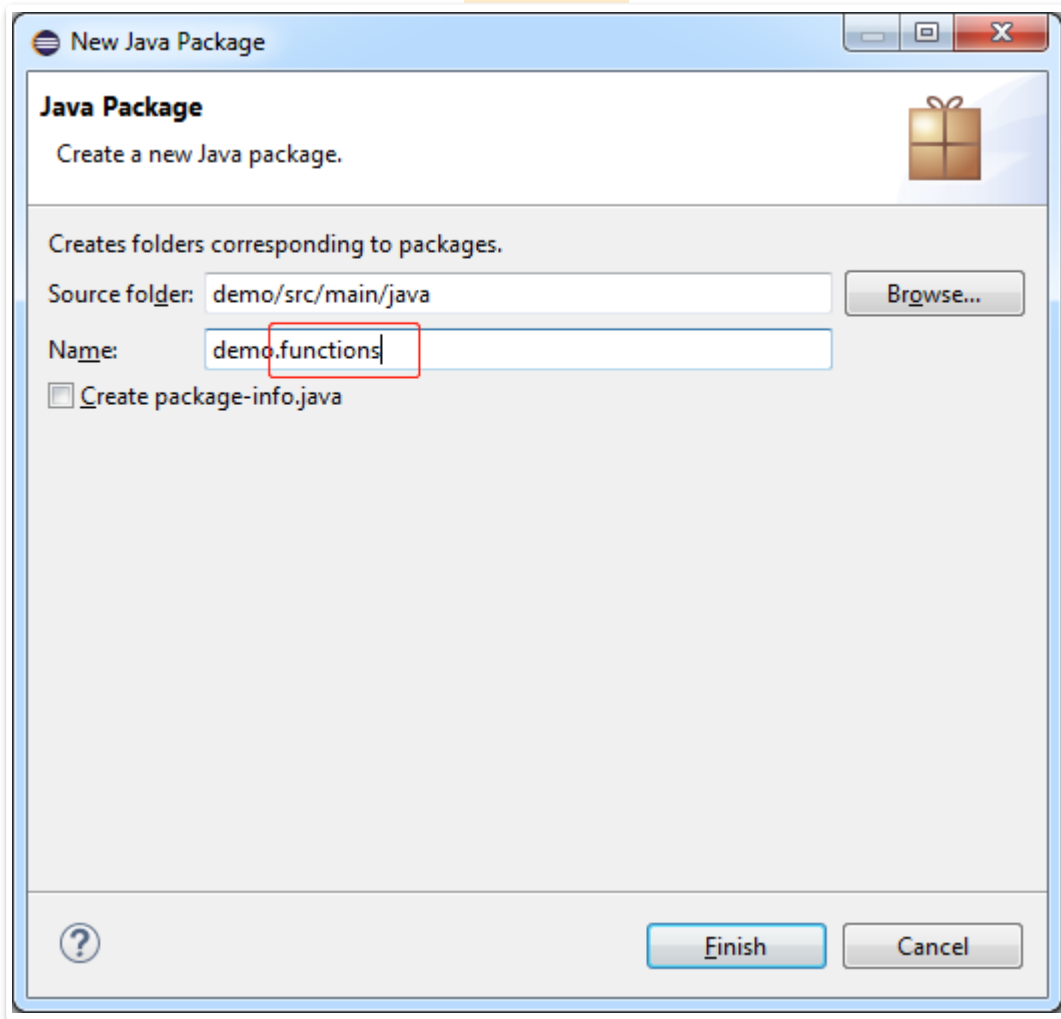
< Back Next > Finish Cancel

新建包

- 在src/main/java 上点击右键，然后选择new，并选择package



- 包名随意，但包名中一定要包含 `.functions`。否则jmeter将不会识别该函数。



添加依赖包

在开始编写自己的函数时，我们需要先添加Jmeter的依赖包。本例中使用到了Maven，因此只需要编辑项目里的 `pom.xml` 文件即可。在自己的文件中添加如下内容，并点击保存 (Eclipse会联网进行下载所需要的jar,需要等一段时间)


```
1  <dependencies>
2    <dependency>
3      <groupId>org.apache.jmeter</groupId>
4      <artifactId>ApacheJMeter_core</artifactId>
5      <version>5.0</version>
6    </dependency>
7    <dependency>
8      <groupId>org.apache.jmeter</groupId>
9      <artifactId>ApacheJMeter_components</artifactId>
10     <version>5.0</version>
11  </dependency>
12 </dependencies>
```

完成的配置文件如下：

```
1  <project xmlns="http://maven.apache.org/POM/4.0.0"
   xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
   xsi:schemaLocation="http://maven.apache.org/POM/4.0.0
   http://maven.apache.org/xsd/maven-4.0.0.xsd">
2    <modelVersion>4.0.0</modelVersion>
3    <groupId>demo</groupId>
4    <artifactId>demo</artifactId>
5    <version>0.0.1-SNAPSHOT</version>
6    <dependencies>
7      <dependency>
8        <groupId>org.apache.jmeter</groupId>
9        <artifactId>ApacheJMeter_core</artifactId>
10       <version>5.0</version>
11      </dependency>
12      <dependency>
13        <groupId>org.apache.jmeter</groupId>
14        <artifactId>ApacheJMeter_components</artifactId>
15        <version>5.0</version>
16      </dependency>
17    </dependencies>
18  </project>
```

新建类

在创建了包之后，就到了函数编写的步骤了。让我们一起来完成我们的第一个函数吧。

- 新建类
 - 类名命名和函数名一致（建议）
 - 继承 `org.apache.jmeter.functions.AbstractFunction` 类（必须）
- 实现AbstractFunction类的抽象方法
- 完整代码如下：

```
1 package demo.functions;
2
3 import java.text.SimpleDateFormat;
4 import java.util.Collection;
5 import java.util.Date;
6 import java.util.LinkedList;
7 import java.util.List;
8
9 import org.apache.jmeter.engine.util.CompoundVariable;
10 import org.apache.jmeter.functions.AbstractFunction;
11 import org.apache.jmeter.functions.InvalidVariableException;
12 import org.apache.jmeter.samplers.SampleResult;
13 import org.apache.jmeter.samplers.Sampler;
14
15 public class getTime extends AbstractFunction {
16     // Define the function name,the name must be start with__.
17     private static final String KEY = "__getTime";
18
19
20
21     // Define the description about the parameter,this text will help user
22     // understand it.
23     private static final List<String> desc = new LinkedList<String>();
24     static {
25         desc.add("Formt the time(optional)");
26     }
27
28     // The object array use for store the user input parameter.
29     private Object[] values;
30
31     // Get the description and show in jmeter helper.
32     public List<String> getArgumentDesc() {
33
34         return desc;
35     }
36
37 }
```

```

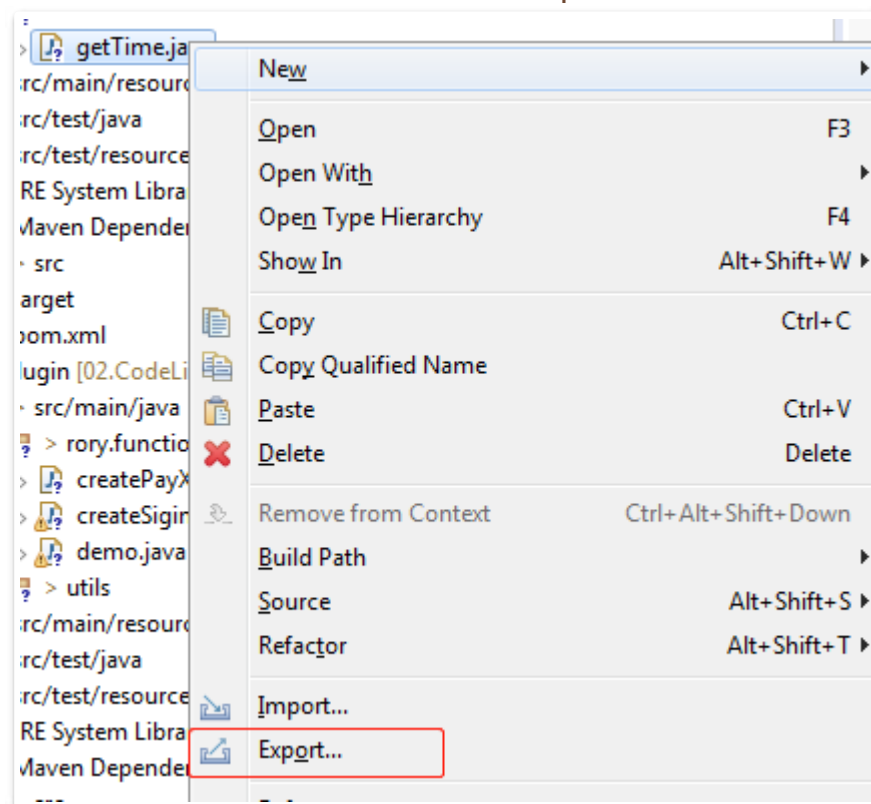
38         //Logic,The result will return jmeter.
39         @Override
40         public String execute(SampleResult previousResult, Sampler currentSampler)
throws InvalidVariableException {
41
42             Date currentDate=new Date();
43             String currentdate=currentDate.toString();
44             //If the parameter is empty will default value.
45             if(values.length>0) {
46                 SimpleDateFormat df = new SimpleDateFormat(((CompoundVariable)
values[0]).execute().trim());
47                 currentdate=df.format(currentDate);
48             }
49
50             return currentdate;
51         }
52
53         //Covent the parameters to array.
54         @Override
55         public void setParameters(Collection<CompoundVariable> parameters) throws
InvalidVariableException {
56
57             values = parameters.toArray();// save parameter.
58
59         }
60
61         // Get define function name and display in jmeter helper.
62         @Override
63         public String getReferenceKey() {
64
65             return KEY;
66         }
67
68     }
69

```

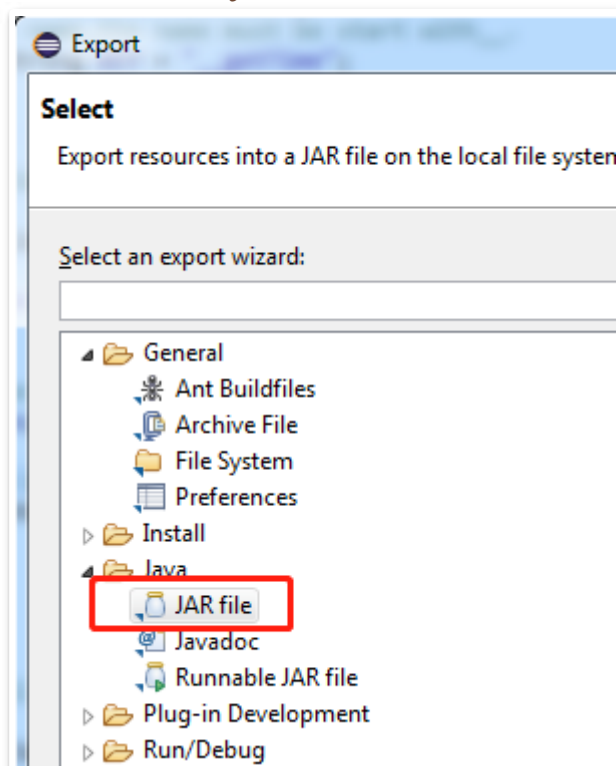
导出与安装

- 把代码导出为jar文件。

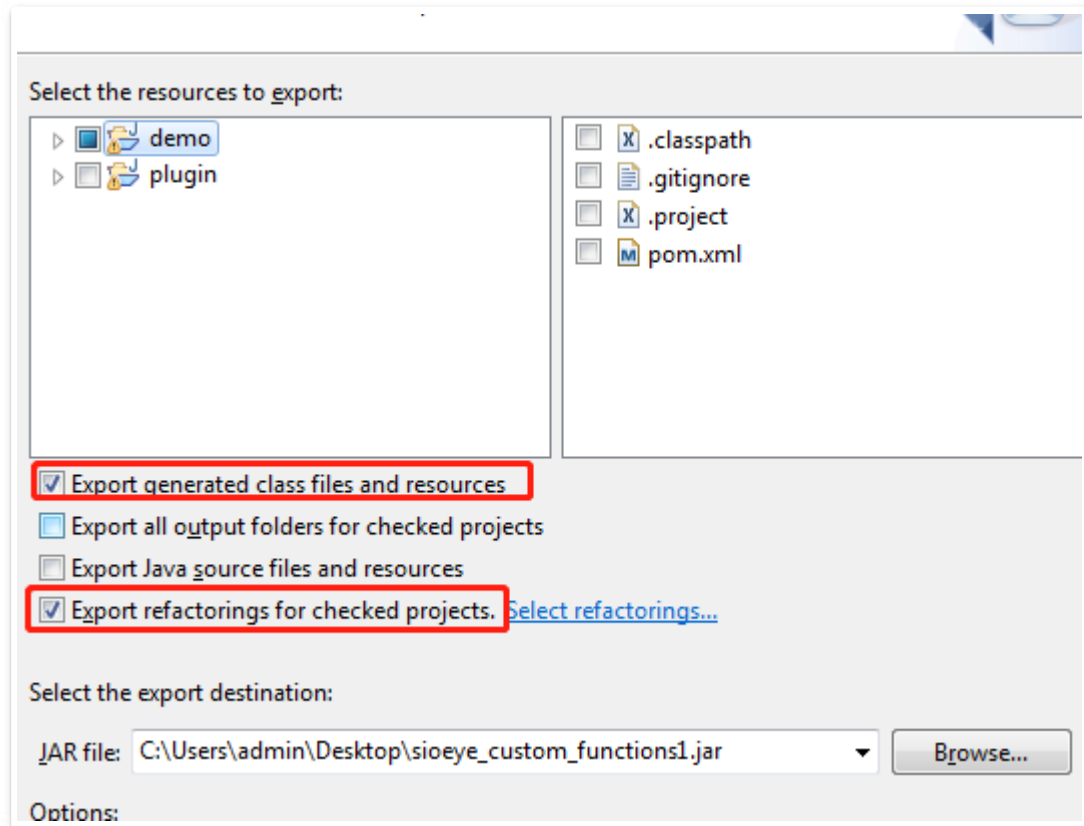
- 在要导出的类或包上点击右键，选择export



- 找到Java并选择jar



- 指定存放路径,导出类文件和资源,然后点击Finish



- 把导出的jar文件放到Jmeter安装路径的 **lib/ext** 目录下
- 重启jmeter就可以了效果如下：

