

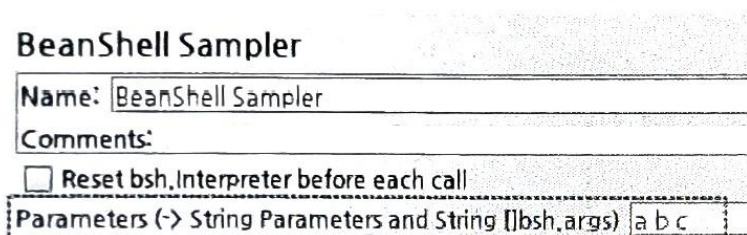
4.10.5 미리 정의된 변수

BeanShell을 사용하는 Element마다 미리 정의된 변수가 있다. 이는 BeanShell이 호출될 때 JMeter의 Context에서 전달받은 변수들이라고 보면 된다. 이를 이용해서 변수를 설정하거나 로그 쓰기 등의 작업을 수행할 수 있다.

BeanShell Sampler

- **Parameters, bsh.args** : 스크립트에 전달되는 파라미터에 접근하는 변수다. BashShell Sampler나 Pre/PostProcessor의 Parameters 부분에 a b c라고 설정하고 스크립트 영역에 [그림 4-77]의 코드를 입력하면 jmeter.log의 결과는 [그림 4-78]과 같다. 즉, Parameters는 Parameters 필드에 입력된 내용 전체가 되고, bsh.args은 Parameters 필드의 내용을 하나의 공백 문자로 분리한 String 배열이 저장된다.

[그림 4-76] Parameter 설정



BeanShell Sampler

Name: BeanShell Sampler

Comments:

☐ Reset bsh.interpreter before each call

Parameters (-> String Parameters and String []bsh.args) a b c

[그림 4-77] Parameter 사용 코드

```
log.info("bsh.args[1]="+bsh.args[1]);  
log.info("Parameters="+Parameters);
```

[그림 4-78] Parameter 사용 결과 로그

```
2014/03/13 19:28:17 INFO - jmeter.util.BeanShellTestElement: bsh.args[1]=b  
2014/03/13 19:28:17 INFO - jmeter.util.BeanShellTestElement: Parameters=a b c
```

- **SampleResult** : 현재 Sampler(BeanShell Sampler)의 SampleResult를 나타낸다. (Class : org.apache.jmeter.samplers.SampleResult)

- **ResponseCode** : 응답코드를 설정하거나 읽어 들일 수 있는 변수(String)다. (예 : "200")
- **ResponseMessage** : 응답 메시지를 설정하거나 읽어 들일 수 있는 변수(String)다. (예 : "OK")
- **IsSuccess** : Sampler의 성공과 실패를 설정하거나 읽어 들일 수 있는 변수(Boolean)다. (예: true/false)
- **Label** : Sampler의 Label을 설정하거나 읽어 들일 수 있는 변수(String)다.
- **FileName** : 스크립트 파일에 입력된 파일명을 읽어 들일 수 있는 변수(String)다.
- **ctx** : JMeter의 Context에 접근할 수 있게 하는 변수다. Thread, Thread Group 관련 정보와 Loop, JMeterEngine에 접근할 수 있다. (Class : org.apache.jmeter.threads.JMeterContext)
- **vars** : 다른 Element에서 설정되거나 사용될 변수를 설정하고 읽어 들일 수 있게 하는 변수다. (Class: org.apache.jmeter.threads.JMeterVariables)
- **props** : 설정 정보에 접근할 수 있게 하는 변수다. (Class: java.util.Properties)
- **log** : jmeter.log 파일에 로그를 쓸 수 있게 하는 변수다. (Class: org.apache.log.Logger)

BeanShell PostProcessor

- **prev** : 바로 직전에 실행된 Sampler의 SampleResult를 나타낸다. 즉, PostProcessor가 적용되는 Sampler의 SampleResult를 의미한다. (Class: org.apache.jmeter.samplers.SampleResult)
- **data** : 현재의 PostProcessor가 적용된 Sampler의 ResponseData를 나타낸다.

- 나머지 변수는 BeanShell Sampler와 동일하다.

BeanShell PreProcessor

- **prev** : 바로 직전에 실행된 Sampler의 SampleResult를 나타낸다. (Class: org.apache.jmeter.samplers.SampleResult)
- **sampler** : 현재의 PreProcessor가 적용된 Sampler를 나타낸다. HTTPSampler라면 org.apache.jmeter.protocol.http.sampler.HTTPSamplerBase을 참조한다.
- 나머지 변수는 BeanShell Sampler와 동일하다.

4.11 Proxy 서버를 이용한 리코딩

LoadRunner와 같은 대부분의 상용 성능 테스트 솔루션에는 스크립트 레코딩 툴이 포함되어 있다. 하나하나 손으로 작성하는 것이 아니라 사용자가 웹 브라우저를 사용함과 동시에 요청 정보가 기록되어 이를 다시 재현해 줄 수 있는 툴이다.

상용 툴보다 다소 복잡하긴 하지만 Recording Controller와 HTTP(S) Test Script Recorder를 이용하면 JMeter에서도 이런 기능을 사용할 수 있다.

HTTP(S) Test Script Recorder는 일종의 프락시 서버로, 이 Element를 이용하는 요청 중 원하는 패턴의 요청을 Recording Controller에 저장하는 방식으로 동작한다. 여기에서는 Recording Controller와 HTTP(S) Test Script Recorder로 손쉽게 Test Plan을 작성하는 방법을 알아본다.

4.11.1 Proxy를 통한 레코딩 예제

이번 예제에서는 JMeter를 이용해서 <http://jmeter.apache.org> 사이트의 *.html 요청을 레코딩해 본다.