

TestExpert A sua comunidade de teste e qualidade de software

[Início](#) > [Blogs](#) > [Blog de eudescosta](#)

Ferramenta de Testes: JMeter

ter, 17/06/2008 - 08:53 — [eudescosta](#)



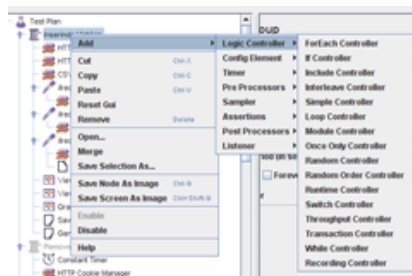
Vou falar agora um pouco sobre o [jmeter](#), uma ferramenta do grupo [apache](#), para a realização de testes de performance, carga e stress. Apesar de ser este o foco do jmeter (testes de performance, carga e stress) ele também pode ser utilizado para realizar testes em webservices, banco de dados e também automatizar alguns teste funcionais, seu uso, alias, é bastante amplo. Lógico que existem ferramentas que desempenham o mesmo papel que o Jmeter como o caso [WAST \(Web Application Stress Test\)](#), [WebLoad](#) mas vamos aborar jmeter neste momento.

Irei mostrar a seguir, o Jmeter em sua operação mais básica e nos posts decorrentes irei aprofundando na sua aplicação.

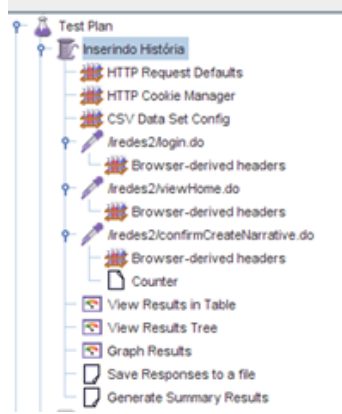
Quem vê o Jmeter pela primeira vez não deve se sentir muito confortável devido a sua relativa falta de intuitividade, mas isso logo passa depois que você entende sua estrutura, que pelo que vocês irão ver, é bem simples.

Creio que a melhor forma de entender a estrutura de elementos da árvore do Jmeter é mostrando logo de início um script do Jmeter todo pronto, pois assim fica mais prático. Irei portanto, usar como exemplo um script que escrevi para alguns testes de performance e stress, são scripts simples que mostram como é fácil usar esta ferramenta.

Antes, irei mostrar onde se encontram os elementos que serão utilizados no exemplo que irei dar, veja na imagem abaixo como chegar neles.



Bem agora vamos para o script com os elementos que utilizei.



Vamos lá, na ordem:

1. Inserindo História - Thread Group (**Add > New > Thread Group**)

Este é o começo de qualquer script (ou plano de teste, se preferir) no jmeter, este elemento inicia seu script, todos os outros elementos que serão interpretados pelo jmeter devem ficar obrigatoriamente abaixo deste thread group, você pode, se preferir, organizar seu script em vários thread groups sem problemas. É aqui no Thread Group que você irá indicar número de conexões simultâneas, Ramp-up e Tempo de Execução.

2. HTTP Request Defaults (**Thread Group > Add > Config Element > HTTP Request Defaults**)

Aqui você define o padrão de seus requests HTTP, este elemento basicamente gerencia os HTTP requests que essa thread está prestes a fazer.

O que basicamente deve estar preenchido neste elemento é o campo **Server Name or IP**.

3. HTTP Cookie Manager (**Thread Group > Add > Config Element > HTTP Cookie Manager**)

Suporte a Cookies, você pode remover este elemento, mas recomendo deixar. Este elemento garante que cada Thread vai receber seu próprio cookie.

4. CSV Data Set Config (**Thread Group > Add > Config Element > CSV Data Set Config**)

Esta aqui é uma facilidade do jmeter, em telas de login ou qualquer tela que você tenha que passar dados para o browser, você pode usar este elemento, ele funciona da seguinte forma.

Você preenche arquivo .csv com os parâmetros que você vai querer passar, conforme imagem abaixo

	A	B
1	admin,1234	
2		

Salve e depois volte ao elemento CSV Data Set Config e informe respectivamente nome para os valores acima *admin* e *1234*, conforme imagem abaixo

Variable Names (comma-delimited): USER,PASS

Neste momento, ao rodar o script o jmeter vai vincular *USER=admin* e *PASS=1234*, e você pode chamar estes valores em qualquer lugar do script assim: `${USER}` e `${PASS}`.

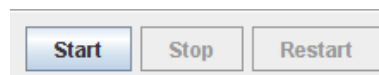
No caso deste script irei utilizar estes valores no campo abaixo.

5. ./redes2/login.do

Aqui é onde começa a interação com o browser, ou seja, onde as requisições são efetivamente realizadas, existe uma forma bem fácil de conseguir estas informações, usando uma funcionalidade chamada **HTTP Proxy Server**.



Com o **HTTP Proxy Server** você pode gravar ações no browser e o jmeter irá gravando tudo pra você, basta que você informe nas configurações de seu browser a porta 8080 (você pode alterar para qualquer porta disponível no seu pc) e pressionar PLAY



Pronto, feito isso, o jmeter deverá lhe poupar de realizar manualmente estes passos.

Como mencionado no tópico 5) irei usar os valores `${USER}` e `${PASS}` aqui.

Name:	Value
login	<code>\${USER}</code>
password	<code>\${PASS}</code>
status	1

6. Browser-derived headers

Estes controles o **HTTP Proxy Server** adiciona automaticamente, não se preocupe com eles.

7. ./redes2/viewHome.do

Mais um request gravado pelo **HTTP Proxy Server**

8. Browser-derived headers

Estes controles o **HTTP Proxy Server** adiciona automaticamente, não se preocupe com eles.

9. ./redes2/confirmCreateNarrative.do

Mais um request gravado pelo **HTTP Proxy Server**

10. Browser-derived headers

Estes controles o **HTTP Proxy Server** adiciona automaticamente, não se preocupe com eles.

11. Counter - (Thread Group > Add > Pre Processors > Counter)

Esta aqui é outra facilidade do jmeter, quando existe a necessidade

de se incluir vários registros sequencialmente no banco, a melhor forma de se fazer isso é via estes Counters, ele vai basicamente incrementar em +1 (isto no entanto é configurável)

Counter

Name:Counter

Comments:Incrementa a História +1

Start1

Increment1

Maximum100

Number format

Reference NameH

☐Track counter independently for each user

No caso acima ele vai começar com 1 incrementar de 1 em 1 até 100 e o nome de referência é H, ou seja, onde você quiser utilizar este counter, basta chamar `#{H}` onde quiser incrementar, veja imagem abaixo.

selectedContacts	
selectedContactGroups	
name	-História \${H}
narrativeDescription	task \${H} - robot running
visibilityLevel	0
tags	

12. View Results in Table - (Thread Group > Add > Listener > View Results in Table)

Bem aqui é a parte dos relatórios, este relatório exibirá os seguintes valores

Sample #	Start Time	Thread Name	Label	Sample Time(s)	Status	Bytes
----------	------------	-------------	-------	----------------	--------	-------

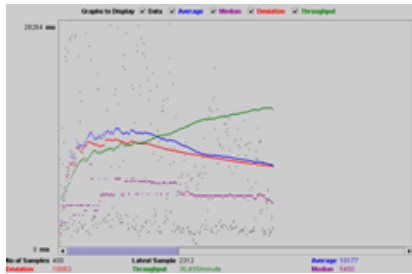
13. View Results Tree - (Thread Group > Add > Listener > View Results Tree)

Os resultados estão vazios, mas ele lista aqui todos os requests e alem disso aqui você também pode renderizar o HTML e ver exatamente o que este request fez, ótimo para debugar caso ocorra algum problema.

Filename

Sampler result

14. Graph Results - (Thread Group > Add > Listener > Graph Results)



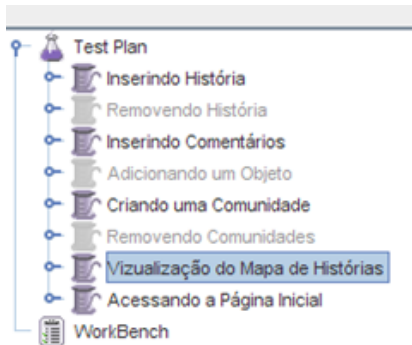
15. Save Responses to a file - (Thread Group > Add > Post Processors > Save Responses to a file)

Salva requisições em um arquivo

16. Generate Summary Results - (Thread Group > Add > Post Processors > Generate Summary Results)

Salva um sumário em um arquivo

Você pode organizar tudo em várias threads desta forma



E antes de finalizar esta parte que é a mais importante

Estas são as propriedades de uma Thread, você deve informar nela o que você vai fazer

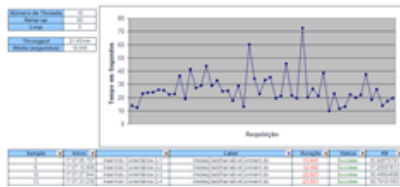
Number of Threads - Quantidade de Threads que será realizada

Ram-UP - Tempo que o jmeter para executar as threads informada acima

Loop - Quantas vezes ele irá realizar esta atividade

No caso acima 10 requisições em 60 segundos repetindo 5 vezes

No final consolida estes resultados em uma planilha



...zezologs

<http://www.zezologs.org/blog/ferramenta-de-testes-jmeter/>

[Blog de eudescosta](#) Por favor, [se logue](#) ou [se registre](#) para poder enviar comentários

[Testes de Performance](#)

Conteúdo popular

Hoje:

[Conteúdo exclusivo](#)

[Como elaborar uma análise de riscos?](#)

[Temas para TCC](#)

[Qualidade, Qualidade de Software e Garantia da Qualidade de Software São as Mesmas Coisas ?](#)

[Regras para a publicação das vagas](#)

[Modelos e Templates de Planos de Testes](#)

[Ferramentas de Teste: Testlink](#)

[Automação e Gerenciamento de Testes: Aumentando a Produtividade com as Principais Soluções Open Source e Gratuitas](#)

[LIVRO: Teste e Análise de Software](#)

[Formando Equipes Eficientes de Teste de Software](#)

[Introdução à Automação de Testes Funcionais](#)

[Modelo V de Testes – Parte 1: O Modelo](#)

[Guia Completo para Certificações em Qualidade e Teste de Software - Versão 2008](#)

[Ferramenta de Testes: JMeter](#)

[CBTM – Change-Based Test Management ou Gerenciamento em Teste Baseado em Mudanças](#)

Posts recentes no blog

[The Bug is on the Table](#)

[Curso de Teste de Performance, Carga e Stress em Curitiba \(Profissional R\\$ 350 Até 12/05/2009\)](#)

[O Paradigma do Elevador](#)

[Qualidade para a retaguarda do software](#)

[Evento: Como planejar a equipe e criar casos de testes OO \(Download do PowerPoint\)](#)

[Cobertura de Testes com Pairwise](#)

[A importância do Teste de Software](#)

[Modelo V: Parte 3 – Validação](#)

[Revisão de Testabilidade: Um caso a se pensar.](#)

[Revista Software Test & Performance](#)

[mais](#)

Novos tópicos no fórum

[Curso de Teste de Performance, Carga e Stress em Curitiba \(Profissional R\\$ 350 Até 12/05/2009\)](#)

[Ferramenta para Teste de Software](#)

[1º Encontro Mensal da ALATS São Paulo - Avaliação e Download da Apresentação
Teste...](#)

[PHP x Ruby on Rails: quem vence o duelo das linguagens?](#)

[Software Grátis para estudantes, micro empresas e para todos teste por 45 dias](#)

[AUTOMATIZADOR](#)

[Problemas com TestComplete -> TestExecute](#)

[% padrão para cobertura de teste](#)

[Test Developer - Porto Alegre/RS - 07/04/2009](#)

[ARQUITETO DE TESTES](#)

[1º Encontro Mensal da ALATS São Paulo - 16/Abr - Noturno](#)

[Gratuito - Sorocaba/SP - Palestra Carreira em Teste de Software - 13/Abr - 19:30](#)

[Vagas TI Wipro - Curitiba/PR](#)

[Treinamento de Gestão de Testes: Teoria e Prática - Porto Alegre/RS](#)

[mais](#)

TestExpert 2007-2008 | Todos os direitos reservados | [Política de Privacidade](#)