

Testes de performance

JMeter

Edjalma Queiroz

`edjalma@ambientinformatica.com.br`

`http://www.ambientinformatica.com.br`

Faculdade SENAI Universitário

04 de Dezembro de 2014

Sumário

- 1 Apresentação
- 2 Plano de Ensino

- Competências
- 3 Introdução

Apresentação

Formação

- Graduado em Análise de Sistemas pela Universidade Salgado de Oliveira.
- Especialista em Tecnologia da Informação e Negócios Eletrônicos pela Universidade Salgado de Oliveira.
- Mestre em Ciência da Computação pela UFG.

Competências

Competências

- Pesquisar, selecionar, instalar e configurar ambientes de desenvolvimento Java (E1);
- Fazer com que aplicações Java funcionem através de ambientes dinâmicos (E2)
- Interconectar aplicações Java com outros componentes necessários para o funcionamento de aplicações Java WEB (E3).
- Criar casos de Testes (Funcional).
- Executar casos de testes em Aplicações (*JMeter Tools*).

Habilidades

Habilidades

- Conhecer sobre Testes Funcional e mensurar *performance*.
- Conhecer a ferramenta JMeter
- Aplicar Testes Funcionais.

Bases Tecnológicas

Bases Tecnológicas

- Introdução ao JMeter

Metodologia de Ensino

Metodologia de Ensino

Acompanhar e executar tutoriais

- 40% teórico
- 60% prático

Introdução

JMeter

- Aplicação desktop open source.
- 100% Java
- Foi desenvolvida para testar o comportamento funcional e medir o desempenho.
- Originalmente projetado para testar aplicações web
- Com sua larga adoção, o projeto foi expandido para outras funções de teste.

Introdução

JMeter - Características

- Capacidade de testar desempenho de muitos tipos de servidor/protocolo:
 - LDAP: Permite enviar requisições para um servidor LDAP. Possui uma implementação simplificada e outra estendida.
 - Message-oriented middleware (MOM) via JMS
 - Mail - SMTP(S), POP3(S) and IMAP(S)
 - MongoDB (NoSQL)
 - Native commands or shell scripts
 - TCP

Introdução

JMeter - Características

- Capacidade de testar desempenho de muitos tipos de servidor/protocolo:
 - Web - HTTP, HTTPS: Pode incluir parâmetros ou arquivos a requisição, escolher o método usado (GET ou POST) e manipular Cookies. Este sampler possui dois tipos de implementação: Java HTTP ou Commons HttpClient.
 - SOAP: Permite enviar requisições SOAP para um Webservice, ou enviar XML-RPC através do protocolo HTTP.
 - FTP: executa o comando de retrieve em um arquivo específico.
 - Database via JDBC: Com esta requisição é possível executar queries em um banco de dados específico.

Introdução

JMeter - Características

- Completamente portavel e 100% puro java
- Suporte a multi-threading, o que permite executar testes em ambientes em que se exige a simultaneidade através de threads ou através de várias funções por grupos (segmentação).
- Interface GUI cuidadosamente desenhada permite a construção de planos de teste e depurações mais rápidas.
- Armazenamento em cache das análises feitas bem como repetição dos testes.

Test Plain

Plano de Teste

Qualquer teste executado como JMeter passa pela criação de um *Test Plan*. O plano de Teste é dividido em elementos:

- *Thread Group*: Este é ponto de começo, todos os outros elementos do Test Plan devem estar sob este. Como o próprio nome ressalta, este controla as threads que serão executadas pelo teste.
- *Listeners*: Fornecem acesso as informações obtidas pelo JMeter durante os testes.

Test Plain

Plano de Teste

Qualquer teste executado como JMeter passa pela criação de um *Test Plan*. O plano de Teste é dividido em elementos:

- *Controllers*: Estes são divididos em dois grupos Samplers e Logic Controllers.
 - *Samplers*: São controladores pré-definidos para requisições específicas. Podendo ser customizada com a inserção de configurações (Configurations), Assertions e etc.
 - *Logic Controllers*: São controladores mais genéricos. Podendo ser customizada com a inserção de outros controllers, configuration elements, assertions, etc.

Test Plain

Plano de Teste

Qualquer teste executado como JMeter passa pela criação de um *Test Plan*. O plano de Teste é dividido em elementos:

- *Timers*: Por padrão, o JMeter faz requisições sem pausas entre elas. Os timers são utilizados para incluir pausas entre as requisições.
- *Assertions*: Usado para verificar se a resposta obtida na requisição é a esperada. Podendo ser usado expressões regulares (Perl-style regular expression) na comparação.

Test Plain

Plano de Teste

Qualquer teste executado como JMeter passa pela criação de um *Test Plan*. O plano de Teste é dividido em elementos:

- *Configuration Elements*: Embora não faça requisições (exceto para HTTP Proxy Server), este elemento pode adicionar ou modificar as requisições.
- *Pre-Processor Elements*: Executa alguma ação antes de fazer a requisição. Mais usado para pré-configurações das requisições.
- *Post-Processor Elements*: Executa alguma ação depois de fazer a requisição. Mais usado para processar as respostas da requisição.

JMeter IDE

JMeter IDE

- Ir para IntroducaoJMeter.pdf