

Configuração do Job no Jenkins

Transcrição

Agora que já entendemos os elementos envolvidos em nossos testes funcionais, configurar nosso Job será uma tarefa mais fácil de ser empreendida. Clicando no Job "argentum-web", vamos na opção "Configurar". Na aba "Build", vamos trocar o goal package do Maven para `verify` :

```
clean verify
```

Agora, precisamos informar um profile na hora de executar nossas goals. Fazemos isso passando o parâmetro `-P` seguido do nome do profile que desejamos executar.

```
clean verify -Ptestes-integracao
```

Ainda falta mais uma configuração, contudo vamos salvar e ver primeiro o que acontece com apenas essa configuração. Em seguida, vamos clicar em "Construir agora". Como de costume, vamos verificar a saída do console enquanto nosso projeto é construído. Veja que ele inicializará o plugin do Maven failsafe. Veja que o tomcat é inicializado, mas há um erro.

O erro, facilmente identificado por uma mensagem amigável, sinaliza que não foi possível encontrar um navegador configurado para executar as configurações dos testes. Em nossos testes, esperamos uma propriedade de sistema que aponte para o local na qual o webdriver para o Chrome se encontra. Precisamos informar essa propriedade, caso contrário continuaremos recebendo a mensagem de erro.

No mundo Java, há várias formas de criar essa propriedade, mas como queremos manter tudo centralizado, usaremos o próprio build para isso. Na aba "Build" da configuração do nosso Job. Na linha, onde informamos as goals do maven, fazemos:

```
clean verify -Ptestes-integracao -Dwebdriver.chrome.driver=caminhoDoWebDriver
```

Rodando o build novamente, vemos que não ocorre mais o erro, pois o teste funcional foi executado com sucesso, realizou o deploy e enviou para nós um e-mail.

Uma alternativa ao Chrome ou outros navegadores é utilizar um Browser que roda em memória como PhantomJS (PhantomJS). Podemos alterar nossa configuração para utilizá-lo:

```
clean verify -Ptestes-integracao -Dphantomjs.binary.path=caminhoDoPhantomJS
```

É claro que você precisará baixá-lo no link que vimos para podermos utilizá-lo.

Rodando nosso código mais uma vez, veja que não há um browser carregado e exibido, mas sim um browser em memória que fará nossos testes.

