

Getting Started With Mammon

Informações

Resumo

Objetivo	Mostrar o processo de checkout e desenvolvimento no projeto Mammon.
Aplicação	Desenvolvedores no projeto Mammon.
Tarefas Relacionadas	

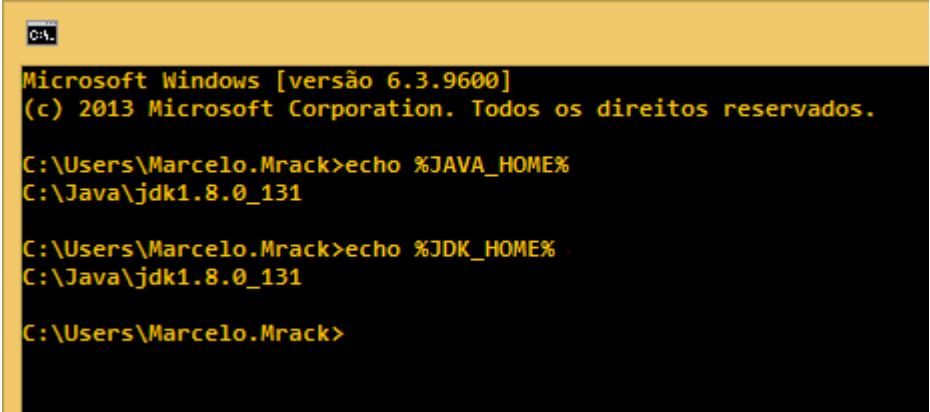
Palavras-chave

[how-to](#)

Referências

Nenhuma.

Procedimento

Cenário	Você quer desenvolver (trabalhar) no projeto Mammon e não sabe como fazer para obter o código-fonte e fazer evoluções.
Procedimento	1 - Instalar o JDK 1.8 64bits Baixar o website da Oracle, e instalar em uma pasta no seu Sistema Operacional Após isso, configurar a variável de ambiente: • JAVA_HOME = <diretório da instalação do Java> • JDK_HOME = <diretório da instalação do Java> Isso deve ficar assim: <pre>C:\Users\Marcelo.Mrack>echo %JAVA_HOME% C:\Java\jdk1.8.0_131 C:\Users\Marcelo.Mrack>echo %JDK_HOME% C:\Java\jdk1.8.0_131 C:\Users\Marcelo.Mrack></pre> Após isso, adicionar o diretório bin do JDK do Java no seu Path do Sistema Operacional, para que comandos como javac, jps e outros estejam acessíveis. Ou seja, o teste abaixo deve funcionar mostrando a versão do Java e as opções do JPS, por exemplo:

```
C:\
Microsoft Windows [versão 6.3.9600]
(c) 2013 Microsoft Corporation. Todos os direitos reservados.

C:\Users\Marcelo.Mrack>javac -version
javac 1.8.0_131

C:\Users\Marcelo.Mrack>jps -lmv

C:\Users\Marcelo.Mrack>jps -help
usage: jps [-help]
          jps [-q] [-mlvV] [<hostid>]

Definitions:
  <hostid>:      <hostname>[:<port>]

C:\Users\Marcelo.Mrack>
```

Veja seção #2 abaixo, para ver como configurar variáveis de ambiente no Windows e como colocar elas dentro do path do Sistema Operacional.

2 -Instale o Maven

Baixe e instale o Maven em um diretório local da sua máquina.



A 3layer recomenda fortemente que você baixe o SDK da Atlassian, o qual já possui um Maven pré-configurado para projetos de desenvolvimento Atlassian que usam o Mammon. Embora não seja obrigatório, isso facilita bastante quando você vai usar o Mammon para projetos de plugins Atlassian. O Maven da Atlassian é um Maven que, além de utilizar os repositórios públicos da Apache, também acessa repositórios públicos da Atlassian, para obter mais dependências. Ou seja, ele é um Maven como qualquer outro, mas com mais repositórios configurados.

Após instalado, configure as variáveis de ambiente do seu Maven no Sistema Operacional, como:

- MVN_HOME = <diretório de instalação do seu Maven>
- MAVEN_HOME = <diretório de instalação do seu Maven>

Elas devem ficar assim:

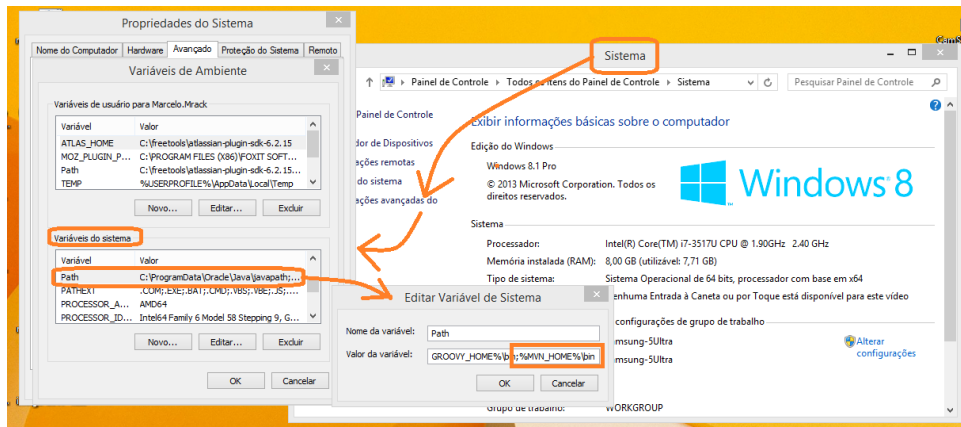
```
C:\
Microsoft Windows [versão 6.3.9600]
(c) 2013 Microsoft Corporation. Todos os direitos reservados.

C:\Users\Marcelo.Mrack>echo %MVN_HOME%
C:\freetools\atlassian-plugin-sdk-6.2.15\apache-maven-3.2.1

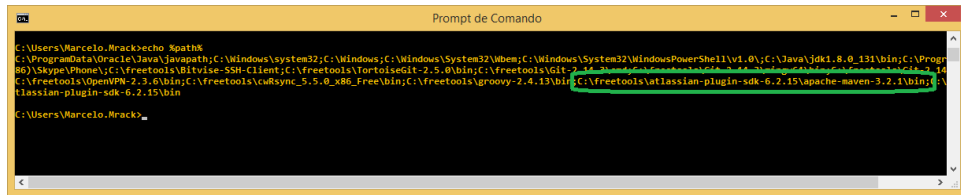
C:\Users\Marcelo.Mrack>echo %MAVEN_HOME%
C:\freetools\atlassian-plugin-sdk-6.2.15\apache-maven-3.2.1

C:\Users\Marcelo.Mrack>
```

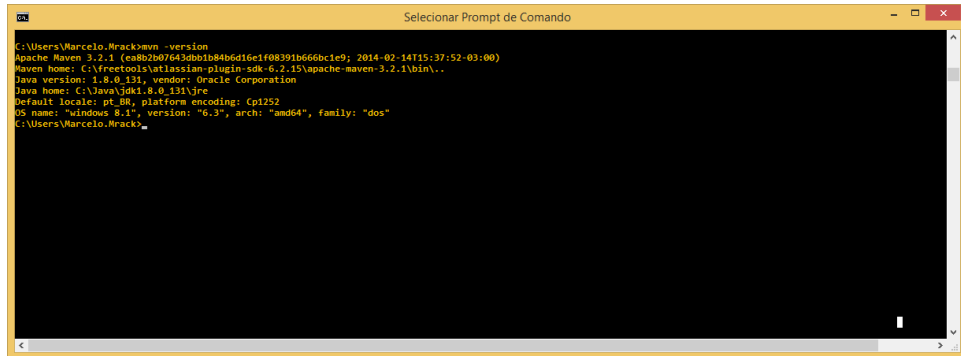
Inclua o executável do Maven na linha de comando do Sistema Operacional, colocando pelo menos uma das variáveis acima dentro do path do sistema + o diretório **bin**, como:



Isso deve ficar assim:



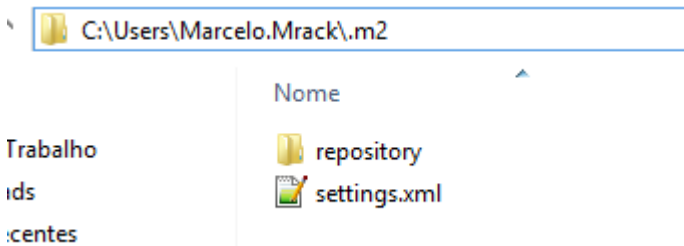
Teste então o Maven, com o comando **mvn -version**:



3 - Configure seu Maven com o Nexus da 3layer

O projeto Mammon utiliza o Nexus da 3layer para armazenamento de artefatos. Portanto, você deve fazer a seguinte configuração no seu Maven:

Baixe o arquivo [settings.xml](#) para sua máquina, e salve-lo dentro da sua pasta de configurações .m2 do Maven, como:



4 - Clonar os Repositórios do Bitbucket

O código do Mammon é público para leitura, e pode ser acessível nas URL abaixo:

<https://3layer.com.br/bitbucket/projects/MM>

Devem existir os repositórios baixo:

 [mammon-atlassian](#)

 [mammon-base](#)

 [mammon-business](#)

 [mammon-core](#)

 [mammon-message](#)

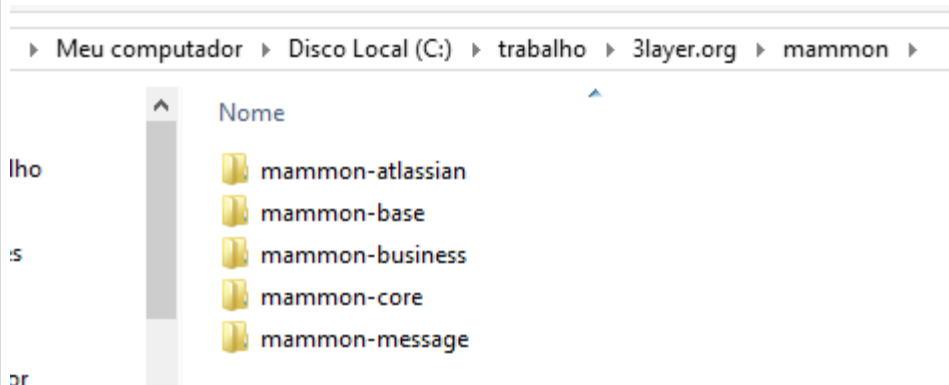
Basta realizar a clonagem do branch **Development** de cada um dos repositórios para a sua máquina local.

Se você estiver dentro de um desses grupos de usuário (veja no seu Profile de usuário quais grupos você pertence), você terá acesso para escrita nos repositórios:

- 3layer-mammon-administrators
- 3layer-mammon-coordinators
- 3layer-mammon-developers
- 3layer-mammon-managers
- 3layer-mammon-qa

Se não estiver logado, você pode fazer a clonagem, mas não conseguirá fazer *push* de volta para o repositório de origem.

O resultado das clonagens deve ser algo como isso:



Entre em cada repositório, e garanta que você está com o branch **Development** ativo, conforme exemplo abaixo:

```
Prompt de Comando

C:\trabalho\3layer.org\mammon\mammon-business>git status
On branch development
Your branch is up-to-date with 'origin/development'.

Changes to be committed:
  (use "git reset HEAD <file>..." to unstage)

    new file:   src/main/java/org/treelayer/mammon/business/service/DefaultService.java
    new file:   src/main/java/org/treelayer/mammon/business/service/HttpUtil.java

Changes not staged for commit:
  (use "git add <file>..." to update what will be committed)
  (use "git checkout -- <file>..." to discard changes in working directory)

    modified:   src/main/java/org/treelayer/mammon/business/service/DefaultService.java
    modified:   src/main/java/org/treelayer/mammon/business/service/HttpUtil.java

C:\trabalho\3layer.org\mammon\mammon-business>
```

5 - Builds Locais

Uma vez baixados os projetos, realize a o build e instalação locais, entrando em cada um deles e executando o comando **mvn install** na seguinte ordem:

- 1. mammon-base
- 2. mammon-core
- 3. mammon-message
- 4. mammon-business
- 5. mammon-Atlassian

Exemplo:

```
Prompt de Comando - mvn install

C:\trabalho\3layer.org\mammon>cd mammon-base
C:\trabalho\3layer.org\mammon\mammon-base>mvn install
[INFO] Scanning for projects...
[INFO] -----
[INFO] Reactor Build Order:
[INFO]
[INFO] 3layer Mammon Base
[INFO] 3layer Mammon Base Atlassian
[INFO]
[INFO] Using the builder org.apache.maven.lifecycle.internal.builder.singlethreaded.SingleThreadedBuilder with a thread count of 1
[INFO]
[INFO] -----
[INFO] Building 3layer Mammon Base 3.0.0-SNAPSHOT
[INFO] -----
```

6 - Entendendo os repositórios Mammon

Os repositórios Mammon constituem o arcabouço para todos os desenvolvimentos de projetos da 3layer, sendo assim organizados:

Repositório	Conteúdo
mammon-base	Contém todas configurações para os demais projetos, com controles de qualidade de código e versões de bibliotecas. É usado por herança nos outros projetos (concieto de "POM Parent") para que todos os projetos tenham sempre os mesmos padrões. Alterações nele se propagam automaticamente para todos os outros projetos. Isso pode ser bom, no sentido atualizar a versão de uma biblioteca, como poder "ruim", no sentido de alterar as rergas de tabulação de código-fonte ou Javadoc, o que obrigaria refatoração de todos os projetos derivados.
mammon-core	Contém classes, anotações e outros recursos extremamente básicos e comuns para todos os projetos que usam o Mammon como base. Exemplos são a classe Inspector , usada para operações de reflexão e a anotação PublicApi que marca objetos para reuso.
mammon-message	Contém todas as funcionalidades para trabalhar com mensagens de validação, avisos e internacionalização no Mammon. Depende do Mammon-Core e é usado pelo Mammon-Business.
mammon-business	Contém funcionalidades para presistência e serviços REST, bem como bases para criação de lógica de negócio através de classes de domínio de de serviços. No tocante à persistência, esse pacote oferece transparência em relação à tipo de banco de dados e framework de persistência e, quando não suportado, permite extensões. Persistência atual é baseada no Hibernate e Active Objects, mas outras podem ser implementadas, como No-SQL, por exemplo. No tocante ao REST, possui um serviço 100% funcional para operações CRUD seguindo o conceito de "crie um POJO e terá o CRUD automático via REST". Tal serviço é baseado em uso avançado de reflexão e é direcionado pelos princípios do framework Apache Isis , embora não utilize ele. Quando o serviço REST precisa regras de negócio, elas podem ser implementadas via services customizadas e ainda assim usar as operações CRUD genéricas do REST, que invocam as regras de negócio via modelo de delegação. Ou, se desejado, podem ser criados serviços REST "brutos" ou então estendendo o serviço padrão ou então as classes abstratas. O serviço REST é implementado pelo Jersey . Já oas services, usam o padrão de extensão e adaptação, com classes genéricas. E, as classes de domínio, representam as regras de negócio, encapsulando os objetos POJO do sistema. DTOs são usados quando necessário, e se valem da biblioteca Murdoc para copia de valores automática via anotações. Depende do mammon-core e mammon-message.
mammon-atlassian	É a base para os desenvolvimentos de plugins Atlassian. Contém funcionalidades para plugins JIRA, Confluence e outros. É ainda um projeto em formação e, pode acabar sendo desmembrado (a exemplo do Mammon Core, Business e Message) em projetos separados para representar melhor os tipos de plugin (coisas básicas para qualquer plugin Atlassian; plugins JIRA; plugins Confluence; plugins Bamboo, etc). Esse é o projeto que está mais instável ainda, no sentido de não se saber como/se as coisas estão bem organizadas ainda. Observe que este projeto herda não o mammon-base padrão, mas o módulo atlassian dentro dele.

6 - Próximos Passos

A partir daqui, recomenda-se a leitura do documento [Mammon - Padroes, Arquitetura e Framework de Desenvolvimento Java \(2012\)](#)

E em caso de dúvidas, crie uma [Question](#) com o labem **mammon**.

Observações	Nenhuma.
-------------	----------