

Tutorial de Metalsmith

Paulo Jerônimo

2016-06-30 15:19:20 BRT

Conteúdo

1. Criando o projeto	2
2. Adicionando o Metalsmith.....	4
3. Fazendo o uso de arquivos Markdown.....	6
4. Disponibilizando arquivos estáticos (imagens, CSS, etc)	8
5. Desenvolvendo o uso de layouts	10
6. Suportando o uso de múltiplas línguas	11
7. Inserindo a localização nas URLs.....	12
8. Traduzindo o layout	14
9. Traduzindo as URLs	15
10. Gerando um índice localizado	16
11. Apresentando uma história de sucesso	17

Este tutorial **(AINDA INCOMPLETO, EM DRAFT)** apresenta como criar um site estático, em JavaScript, utilizando o [Metalsmith](#), um dos [vários geradores de sites estáticos](#) existentes no mercado.

Os seguintes assuntos são explorados:

1. Organização de uma estrutura de diretórios para o projeto;
2. Utilização do [NodeJS](#) e do Metalsmith;
3. Utilização do [Git](#) para o versionamento dos fontes;
4. Utilização do [Vim](#) com *plugins* provendo *features* para *syntax highlighting* e *code completion* para linguagens como JavaScript, JSON, [Jade](#) e [Markdown](#), dentre outras;
5. Escrita de páginas utilizando a linguagem [Markdown](#);
6. Escrita de templates utilizando a linguagem [Jade](#);
7. Criação de sites apresentando conteúdos em diferentes línguas;

Nesta versão, este tutorial expande, adapta e executa, passo a passo, numa VM Linux, os comandos apresentados, de forma bastante prática prática e assertiva, nestes dois outro tutoriais:

1. [Building a static multi-language site with Metalsmith \(part I\)](#)
2. [Building a static multi-language site with Metalsmith \(part II\)](#)

Para que você possa se aprofundar ainda mais nos assuntos cobertos aqui, é recomendada a leitura desses materiais. Você pode fazer isso antes ou depois de seguir este tutorial, fica a teu critério.

1. Criando o projeto

Inicie o projeto com os seguintes comandos:

```
mkdir metalsmith-test && cd $_  
date > TIMESTAMP  
npm init
```

Abaixo são apresentadas as respostas que dou para os questionamentos feitos pelo comando acima.



Em tuas respostas, utilize o teu nome para o campo **author** e deixe as outras respostas iguais.

```
This utility will walk you through creating a package.json file.  
It only covers the most common items, and tries to guess sensible defaults.
```

```
See `npm help json` for definitive documentation on these fields  
and exactly what they do.
```

```
Use `npm install <pkg> --save` afterwards to install a package and  
save it as a dependency in the package.json file.
```

```
Press ^C at any time to quit.  
name: (metalsmith-test)  
version: (1.0.0)  
description: Metalsmith test  
entry point: (index.js)  
test command:  
git repository: .  
keywords: metalsmith javascript  
author: Paulo Jerônimo  
license: (ISC)
```

Verifique se o conteúdo do teu arquivo **package.json** é semelhante a [este](#).

Faça o teu primeiro *git commit*:

```
git init  
git add .  
git commit -m 'Criado o projeto'
```



Você receberá um erro na execução do comando `git commit` se ainda não tiver configurado teu nome e email no `git`. Isso ocorrerá, por exemplo, caso você esteja utilizando a `fedora-nodejs-box` pela primeira vez. Para corrigir esse problema, e poder reexecutar o comando, execute os comandos a seguir: *(obviamente, informe teus dados como valores para `user.name` e `user.email`)*

```
git config --global user.name "Paulo Jerônimo"
git config --global user.email "paulojeronimo@gmail.com"
```

2. Adicionando o Metalsmith

Adicione o metalsmith ao `package.json` e crie um novo arquivo HTML com os comandos abaixo:

```
npm install --save metalsmith
mkdir content && touch content/index.html
```

Observe as mudanças no repositório (`git status`).

Crie o arquivo `index.js` com o comando abaixo:

```
cat > index.js <<'EOF'
var path = require('path');

var metalsmith = require('metalsmith');

metalsmith(__dirname)
  .source('content')
  .destination('dist')
  .build(function (err) {
    if (err) { console.error(err); }
  });
EOF
```

Observe as mudanças na estrutura de diretórios do repositório:

```
tree -L 1
node index.js
find dist/
git status
```

Crie o arquivo `.gitignore` com o comando abaixo:

```
cat > .gitignore <<'EOF'
dist
node_modules
EOF
```

Faça o teu segundo commit:

```
git status
git add .
git commit -m 'Adicionado o uso do Metalsmith'
```

Verifique o log do repositório:

```
git log --name-status
```

3. Fazendo o uso de arquivos Markdown

Renomeie o arquivo `index.html` para `index.md` e altere o seu conteúdo com os seguintes comandos:

```
git mv content/index.html content/index.md
cat > content/index.md <<'EOF'
---
title: Home
---

My collection of recipes.
EOF
```

Adicione o pacote `metalsmith-markdown`:

```
npm install --save metalsmith-markdown
```

Clone o código desse tutorial e adicione seu diretório ao `.gitignore`. Estamos fazendo isso para poder utilizar alguns arquivos de seu repositório.

```
git clone https://github.com/paulojeronimo/tutorial-metalsmith
echo tutorial-metalsmith >> .gitignore
```

O conteúdo apresenta, [no formato diff](#), as mudanças que você precisará fazer no `index.js`:



as linhas com um "+" na frente representam adições realizadas.

```
diff --git a/index.js b/index.js
index eeb354b..6dac41a 100644
--- a/index.js
+++ b/index.js
@@ -1,10 +1,12 @@
  var path = require('path');

  var metalsmith = require('metalsmith');
+var markdown = require('metalsmith-markdown');

  metalsmith(__dirname)
    .source('content')
    .destination('dist')
+  .use(markdown())
    .build(function (err) {
      if (err) { console.error(err); }
    });
```


Você pode fazer essas alterações "na mão", alterando o arquivo, se quiser. Ou, você pode simplificar tua vida e executar o seguinte comando:

```
git apply tutorial-metalsmith/patches/index.js.1
```

Você observará as mudanças através do comando `git difftool`. Antes, porém, faça as seguintes configurações:

```
git config --global diff.tool vimdifff
git config --global --add difftool.prompt false
```

Agora, veja as diferenças:

```
git difftool
```

Execute o comando abaixo e veja o arquivo (`index.html`) que será gerado a partir do `index.md`:

```
node index.js
cat dist/index.html
```

Faça o teu terceiro *git commit*:

```
git status
git add .
git commit -m 'Feito o uso de arquivos Markdown'
```

4. Disponibilizando arquivos estáticos (imagens, CSS, etc)

Adicione o uso do pacote `metalsmith-assets`:

```
npm install --save metalsmith-assets
```

Crie o diretório `assets/css` e o arquivo `styles.css` dentro dele. Em seguida veja a estrutura montada.

```
mkdir -p assets/css && touch $_/styles.css
tree assets/
```

Para fazer uso de assets, mudanças precisarão ser realizadas no `index.js`, conforme o conteúdo abaixo:

```
diff --git a/index.js b/index.js
index 6dac41a..6f0288f 100644
--- a/index.js
+++ b/index.js
@@ -2,11 +2,15 @@ var path = require('path');

  var metalsmith = require('metalsmith');
  var markdown = require('metalsmith-markdown');
+var assets = require('metalsmith-assets');

  metalsmith(__dirname)
    .source('content')
    .destination('dist')
    .use(markdown())
+  .use(assets({
+    source: 'assets'
+  }))
    .build(function (err) {
      if (err) { console.error(err); }
    });
```

Altere o `index.js` executando o comando a seguir:

```
git apply tutorial-metalsmith/patches/index.js.2
```

Verifique a mudança, rode novamente o `index.js`, e veja a estrutura construída:

```
cat index.js  
node $_  
tree dist/
```

Faça o teu quarto *git commit*:

```
git status  
git add .  
git commit -m 'Disponibilizado arquivos estáticos'
```

5. Desenvolvendo o uso de layouts

```
npm install --save jade metalsmith-layouts
```

```
mkdir -p layouts && cat > layouts/default.jade <<EOF
doctype html
head
  title Cookbook - #{title}
  meta(charset='utf-8')
  link(rel='stylesheet', href='/css/styles.css')
body
  header
    h1 Cookbook
  main: article
    h1= title
    != contents
EOF
```

```
cat tutorial-metalsmith/patches/index.js.3
git apply !${
```

```
node index.js
cat dist/index.html
```

```
git status
git add .
git commit -m 'Desenvolvido o uso de layouts'
```

6. Suportando o uso de múltiplas línguas

```
git mv content/index.md content/index_en.md
```

```
cat > content/index_es.md <<'EOF'
---
title: Inicio
---

Mi colección de recetas.
EOF
```

```
npm install --save metalsmith-multi-language
```

```
git apply tutorial-metalsmith/patches/index.js.4
cat index.js
node $_
tree dist/
```

```
git status
git add .
git commit -m 'Suportado o uso de múltiplas línguas 1/2'
```

```
git apply tutorial-metalsmith/patches/layouts/default.jade.1
cat layouts/default.jade
```

```
node index.js
npm install -g http-server
(cd dist/; http-server -p 4000)
```

Abra a URL <http://localhost:4000> e teste a aplicação. Ao terminar teu teste, pressione um **Ctrl+C** para encerrar o servidor e continuar os próximos comandos.

```
git status
git commit -am 'Suportado o uso de múltiplas línguas 2/2'
```

7. Inserindo a localização nas URLs

```
npm install --save metalsmith-permalinks
```

```
mkdir content/recipes
cat > content/recipes/tortilla_en.md <<'EOF'
---
title: Spanish omelette
---

Ingredients:

- 2 potatoes
- 3 eggs
- 1 onion
- Olive oil
- Salt
EOF
cat > content/recipes/tortilla_es.md <<'EOF'
---
title: Tortilla de patatas
---

Ingredientes:

- 2 patatas
- 3 huevos
- 1 cebolla
- Aceite de oliva
- Sal
EOF
```

```
git apply tutorial-metalsmith/patches/index.js.5
git apply tutorial-metalsmith/patches/content/index_en.md.1
```

```
node index.js
(cd dist/; http-server -p 4000)
```

Abra a URL <http://localhost:4000/en/spanish-omelette/> e teste a aplicação. Ao terminar teu teste, pressione um **Ctrl+C** para encerrar o servidor e continuar os próximos comandos.

```
git status  
git add .  
git commit -m 'Inserida a localização nas URLs'
```

8. Traduzindo o layout

```
npm install --save metalsmith-i18n
```

```
mkdir -p locales
cat > locales/en.json <<'EOF'
{
  "Cookbook": "Cookbook",
  "Also available in": "Also available in",
  "es": "castellano"
}
EOF
cat > locales/es.json <<'EOF'
{
  "Cookbook": "Recetario",
  "Also available in": "Disponible también en",
  "en": "English"
}
EOF
```

```
git apply tutorial-metalsmith/patches/index.js.6
git apply tutorial-metalsmith/patches/layouts/default.jade.2
```

```
node index.js
(cd dist/; http-server -p 4000)
```

Abra a URL <http://localhost:4000/en/spanish-omelette/> e teste a aplicação. Ao terminar teu teste, pressione um **Ctrl+C** para encerrar o servidor e continuar os próximos comandos.

```
git status
git add .
git commit -m 'Traduzido o layout'
```


9. Traduzindo as URLs

```
npm install --save metalsmith-collections
```

```
git apply tutorial-metalsmith/patches/index.js.7
```

```
node index.js  
tree dist/  
(cd dist/; http-server -p 4000)
```

Abra a URL <http://localhost:4000/en/recipes/spanish-omelette/> e teste a aplicação. Ao terminar teu teste, pressione um **Ctrl+C** para encerrar o servidor e continuar os próximos comandos.

```
git status  
git add .  
git commit -m 'Traduzidas as URLs'
```

10. Gerando um índice localizado

```
git apply ../tutorial-metalsmith/patches/layouts/default.jade.3
```

```
cat > layouts/home.jade <<'EOF'
extends ./default.jade

block extra
  h2= __('Recipes')

  ul
    each recipe in collections['recipes_{{locale}}']
      li: a(href='/${recipe.path}')= recipe.title
EOF
```

```
git apply ../tutorial-metalsmith/patches/content/index_en.md.2
```

```
node index.js
git status
sed -i 's/\(: "\)Recipes/\1Recetas/g' locales/es.json
cat $_
```

```
git status
git add .
git commit -m 'Gerado um índice localizado'
```

11. Apresentando uma história de sucesso

Parabéns! Você concluiu um bom trabalho! Execute o comando git a seguir para ver toda a tua história de sucesso:

```
git log --pretty=format: '* %s %Cgreen(%cr)'
```