**Bower na prática**

Um dos grandes problemas encontrados no desenvolvimento de software é o gerenciamento de dependências. Saiba como o Bower pode ajudar.

**por**[**Diogo Beato**](https://tableless.com.br/author/diogo-beato) **15/01/2014** [**1 comentário**](https://tableless.com.br/bower-na-pratica/#disqus_thread)

A maioria das linguagens já possuem ferramentas para automatizar e facilitar esse tipo de tarefa.

Por exemplo: Java – (Maven e Ivy), Ruby – (Rubygems), Python – (pip), entre outras.

No desenvolvimento front-end, uma ferramenta que vem ganhando bastante espaço nessa área é o [Bower](http://bower.io/" \o "bower.io" \t "_blank).

Como o bower é para gerenciar componentes de front-end, pode ser adicionado em qualquer projeto, independente do seu back-end ser Java, Ruby, Python, Node, PHP, etc.

Andei pesquisando a respeito dessa ferramenta e senti a falta de artigos em português que ensinem o passo-a-passo de como incorpora-lo em nossas aplicações, por isso resolvi escrever esse tutorial, ajudar com que todos possam dar seus primeiros passos com o [Bower](http://bower.io/" \o "bower.io" \t "_blank) e melhorar o gerenciamento de dependências dos seus respectivos front-ends.

**Instalação**

Inicialmente é necessário primeiro que você tenha o [Node](http://nodejs.org/) junto com o [NPM](https://npmjs.org/) instalado na sua máquina. Caso contrário, basta acessar o [nodejs.org](http://nodejs.org/), fazer download e instalar. Bem simples!

Feito isso, vamos para o terminal/cmd e instalar o bower com o seguinte comando:

npm install -g bower

Para quem ainda não conhece, o [NPM](https://npmjs.org/)  é um gerenciador de pacotes de programas que rodam com o Node. Aqui estamos basicamente mandando ele instalar o bower no nosso computador, uma coisa que gostaria de ressaltar é a opção -g que está dizendo para o npm instalar o bower globalmente em nossa máquina, assim ele já fica nas nossas variáveis de ambiente e podemos utiliza-lo facilmente em outros projetos.

**Adicionando o Bower ao projeto**

Nesse exemplo, vamos ilustrar um projeto web simples, sem back-end, mas garanto que vai ficar fácil de entender como incorpora-lo ao seu projeto, independente da linguagem. Caso vocês tenham alguma dúvida, perguntem nos comentários que vou me esforçar para ajudar.

Nosso projeto vai se chamar **“zombie-striker”** e terá a seguinte estrutura:

|zombie-striker/

|--assets/

|----scripts/

|----styles/

|----images/

|--index.html

Para adicionarmos o bower, vamos até a pasta do projeto “/zombie-striker” e digitar o comando:

**bower** init

O bower irá iniciar um wizard para gerar o arquivo “bower.json” pedindo pra você completar as seguintes informações:

# nome do projeto

name:zombie-striker

# versão do projeto

version:0.0.1

# descrição do projeto

description: app to strike zombies with bower

# arquivo principal do seu projeto

main file: assets/scripts/main.js

# palavras-chaves

keywords: zombie striker

# autores do projeto

authors: "Diogo Vecchiati http://divecch.com"

# tipo de licença

license: MIT

#homepage do projeto

homepage: "https://github.com/diRex/zombie-striker"

# se você gostaria que o bower adicionasse os components já instalados, como dependências no arquivo json.

set currently installed components as dependencies?(y/n) n

# se você gostaria de adicionar o ignore list default do bower

add commonly ignored files to ignore list?(y/n) y

# se você gostaria de tornar esse pacote privado para que não seja acidentalmente publicado no registro de pacotes do bower.

would you like to mark this package as private which prevents it from beig accidentally published to the registry?(y/n) **y**

**Observação**: Algumas das opções acimas são válidas apenas para pacotes que vão ser distribuídos como novos componentes, por exemplo: caso você esteja criando um novo framework e queira disponibilizar aos demais através do bower .Porém, não é nosso caso, então podemos utilizar o “bower.json” gerado pelo wizard e modificar de acordo com a nossa necessidade. Caso você queira, pode pular a etapa de wizard do bower init e  criar o “bower.json” na mão com as opções que você queira.

Ao terminar o wizard, você terá um “bower.json” parecido com esse:

{

"name": "zombie-striker",

"version": "0.0.1",

"authors": [

"Diogo Vecchiati <http://divecch.com>"

],

"description": "app to strike zombies with bower",

"main": "assets/scripts/main.js",

"keywords": [

"zombie"

],

"license": "MIT",

"homepage": "https://github.com/diRex/zombie-striker",

"private": true,

"ignore": [

"\*\*/.\*",

"node\_modules",

"bower\_components",

"test",

"tests"

]

}

**Adicionando dependências**

Como de costume na maioria dos projetos front-end, vamos utilizar o JQuery como dependência. Vou mostrar duas maneiras de fazer isso.

**Editando o arquivo “bower.json”**

Você pode editar o seu arquivo “bower.json” e adicionar

...

"ignore": [

"\*\*/.\*",

"node\_modules",

"bower\_components",

"test",

"tests"

]

"dependencies": {

"jquery": "~2.0.3"

}

}

e em seguida executar:

**bower** install

Toda vez que você executa o bower install, ele verifica quais as dependências existentes no seu arquivo “bower.json” e caso elas não estejam presentes na pasta de componentes serão instaladas.

**Executando o comando bower install**

Outra maneira é executando o camando bower install <package>

bower **install** jquery --save

A opção --save serve para adicionar o componente no “dependencies” do “bower.json”.

Por padrão, o diretório que o bower utiliza pra salvar os componentes instalados é “bower\_components/”, caso você queira modificar, basta criar um arquivo chamado “.bowerrc” com o seguinte conteúdo:

{

"directory":"assets/components"

}

Depois de alterar o diretório dos componentes, o ideal é que você remova o diretório anterior, “bower\_componentes” e execute o bower install, pra ele fazer download das dependências novamente. Ou simplesmente renomeie 😛

Para importar o jquery no nosso projeto é o mesmo “arroz com feijão”, só adicionar a tag script no html:

<**script** src="assets/components/jquery/jquery.min.js"

**Pesquisando componentes**

Caso você queira pesquisar mais componentes para adicionar ao seu projeto, é só utilizar o bower search. Por exemplo, quero adicionar o bootstrap ao meu projeto, mas não sei o nome correto do pacote.

**bower** search bootstrap

Assim você consegue listar todos os pacotes que tenham relação com o bootstrap, é bem confuso de ver no terminal,como acontece na pesquisa de qualquer gerenciador de pacotes, mas da pra encontrar.

**Dicas**

**Não versione o diretório de componentes**

Não versione a pasta que vc estiver utilizando pra salvar os componentes gerenciados pelo bower, se a sua pasta for a padrão(“bower\_components/”), coloque no seu .gitignore, assim você deixa seu repositório mais leve e evita conflitos de libs que foram adicionadas por diferentes desenvolvedores.

**Atenção com as dependências**

Mantenham todas as dependências configuradas no “bower.json” algumas vezes pode acontecer de você executar um bower install jquery, esquecer da opção --save ou esquecer de adicionar manualmente no “bower.json” Quando outro desenvolvedor for participar do seu projeto e executar um bower install, o jquery não vai estar lá.

**Ferramenta visual para pesquisa de pacotes**

Outra forma de visualizar os componentes registrados no bower de maneira mais agradável é através do site: <http://sindresorhus.com/bower-components/>.

**That’s all folks**, espero que tenham gostado, que passem a utilizar o bower em seus projetos, pois facilita muito controlar quais dependências existentes no projeto. Se quiserem pesquisar mais sobre as possibilidades que o bower oferece, basta acessar o site oficial: <http://bower.io/>.

Qualquer dúvida, critica ou sugestões comentem aqui em baixo.