**O que é**

*Template engine;* (Permite manipular e organizar conteúdo HTML de forma dinâmica, renderiza dados)

1. Library javascript
2. Utiliza templates para manipular, organizar conteúdo html de forma dinâmica no DOM.
3. Diferente de outras engines : é logic-less, utiliza o menos possível de lógica na manipulação dos templates
4. Suporte a mais de 22 linguagens

Mustache is a logic-less template syntax. It can be used for HTML, config files, source code - anything. It works by expanding tags in a template using values provided in a hash or object.

We call it "logic-less" because there are no if statements, else clauses, or for loops.

Instead there are only tags. Some tags are replaced with a value, some nothing, and others a series of values. (A lógica nos templates é a mínima possível)

* Suporte à várias linguagens: 22 linguagens
  + Ruby
  + JavaScript
  + Python
  + Erlang
  + Elixir
  + PHP
  + Perl
  + Perl6
  + Objective-C
  + Java
  + C#/.NET
  + Android
  + C++
  + CFEngine
  + Go
  + Lua
  + ooc
  + ActionScript
  + ColdFusion
  + Scala
  + Clojure[Script]
  + Fantom
  + CoffeeScript
  + D
  + Haskell
  + XQuery
  + ASP
  + Io
  + Dart
  + Haxe
  + Delphi
  + Racket
  + Rust
  + OCaml
  + Swift
  + Bash
  + Julia
  + R
  + Crystal
  + Common Lisp
  + Nim
  + Smalltalk
  + Tcl
  + C
  + ABAP
  + Elm
* Editores:
  + TexteMate
  + Vim
  + Emacs
  + Coda
  + Atom

**Para que serve**

Explicar para que serve um *Template engine;*

Boa manutenibilidade:

* Fácil de usar;
* Sintaxe simples;
* Não possui lógica avançada embutida;
* Universal (rodam tanto no servidor como no browser e tem suporte à variadas linguagens)

Possibilita a implementação do seu próprio parser para os templates Mustache;

**Como funciona**

Via npm:

$ npm install mustache --save

Instale com o bower (gerenciador de pacotes para front-end):

$ bower install --save mustache

ou Demo:

<http://mustache.github.io/#demo>

Importar no código html:

</script type=”text/javascript” src=”mustache.js”></script>

<script id="template" type="x-tmpl-mustache">

Template

</script>

**JS**

Instalação Global:

$ npm install -g mustache

$ mustache dataView.json myTemplate.mustache > output.html

ou package.json

$ npm install mustache --save-dev

{

"scripts": {

"build": "mustache dataView.json myTemplate.mustache > public/output.html"

}

}

$ npm run build

**Linhas de Comando:**

Mustache.render

-p

**Exemplos:**

**Variáveis**

**Seções**

**Funções**

**Seções invertidas**

**Comentários**

**Parciais**

**Delimitadores customizados**