



Automatizando tarefas com o Grunt

O que é, o que ele faz?

O Grunt é um módulo para Node.js que tem como funcionalidade a automatização de execução de tarefas, como exemplo, deixar configurado uma tarefa de builtin, onde ele irá compilar arquivos less, minificar e concatenar css e js, rodar testes unitários, copiar arquivos de imagem para um determinado diretório, entre muitas outras coisas.

Como eu instalo?

Primeiramente, temos que baixar e instalar o Node.js, para isso, [clique aqui](#).

Após o node ser instalado, abra o terminal de comandos e execute o seguinte:

```
npm install -g grunt-cli
```

isto irá instalar grunt via linha de comando, a opção “-g” indica que é globalmente, ou seja, será adicionado a sua pasta de bibliotecas do node.js.

Como eu uso?

Após o node ser instalado, crie uma pasta para fazermos os testes, a minha ficou em “c:\teste”, após, abra o terminal dentro da pasta e execute o seguinte comando:

```
npm init
```

este comando serve apenas para criar um arquivo json de informações do seu projeto, será necessário informar alguns dados, como nome e descrição, obs: não utilize espaços no nome do projeto.

Depois disto, vamos instalar o Grunt localmente no projeto, executando o seguinte:

```
npm install grunt
```

será criada uma pasta chamada “node_modules”, esta pasta será utilizada para armazenar todos os plug-ins do seu projeto que o Grunt irá utilizar.

Agora vamos criar nossos arquivos para fazer testes

<pre>assets/my-javascript.js 1. function foo() 2. { 3. alert('bar'); 4. }</pre>	<pre>assets/my-style.css 1. #foo 2. { 3. background-color: red; 4. } 5. 6. .bar 7. { 8. color: blue; 9. }</pre>
--	--



Na raiz do projeto, crie um arquivo chamado “Gruntfile.js”, onde serão configuradas todas as tarefas que serão executadas, coloque este conteúdo como exemplo:

Gruntfile.js

```
1. module.exports = function(grunt) {
2.
3.     // Configuração do projeto
4.     grunt.initConfig({
5.         pkg: grunt.file.readJSON('package.json'),
6.
7.         // Tarefa de Concateção e Minificação de estilos
8.         cssmin: {
9.
10.            // tarefa principal, pode-se criar várias
11.            main : {
12.                files : {
13.                    // este será o arquivo de estilo de saída
14.                    'assets/production/prod.min.css' : [
15.                        'assets/my-style.css'
16.                    // pode ser adicionados vários arquivos aqui
17.                ]
18.            }
19.        },
20.
21.        // Tarefa de Concatenação, Minificação e distorção de script
22.        uglify : {
23.
24.            // Comentário opcional no script de saída
25.            options: {
26.                banner: '/*! <%= pkg.name %> - v<%= pkg.version %> -
27. ' + '<%= grunt.template.today("yyyy-mm-dd") %> */'
28.            },
29.
30.            main : {
31.                files : {
32.                    // este será o arquivo de script de saída
33.                    'assets/production/prod.min.js' : [
34.                        'assets/my-javascript.js'
35.                    // pode ser adicionado vários arquivos aqui
36.                ]
37.            }
38.        }
39.
40.    }
41.
42. });
43.
44. // Carrega os plugins de tarefas necessários
45.
46. // Plugin necessário para os estilos
47. grunt.loadNpmTasks('grunt-contrib-cssmin');
48.
49. // Plugin necessário para os scripts
50. grunt.loadNpmTasks('grunt-contrib-uglify');
51.
52. // Registra as tarefas que serão executadas
53. grunt.registerTask( 'default', [ 'cssmin','uglify' ] );
54.
55. };
```



Este arquivo está configurado para fazer o seguinte:

Juntar os arquivos de estilos, caso tivesse mais de um (`my-style.css`), minificá-los e criar um arquivo **“assets/prod/prod.min.css”** com o conteúdo desejado, o mesmo fará com os arquivos de script.

Antes de executarmos essas tarefas, é necessário instalar alguns plug-ins adicionais, são eles:

grunt-contrib-mincss : Responsável por minificar os estilos

grunt-contrib-uglify : Responsável por minificar os scripts

Execute o seguinte comando para instalar:

```
npm install grunt-contrib-cssmin grunt-contrib-uglify
```

Após instalado, basta apenas executarmos nossas tarefas, com o comando:

```
grunt
```

se tudo der certo, dois arquivos serão criados, conforme o esperado.

Como executo tarefas específicas?

O Grunt também permite a execução de tarefas específicas, como exemplo, apenas minificar os scripts ou só os estilos, para fazer isso rodamos os seguintes comandos:

```
grunt cssmin // irá executar todas as tarefas relacionadas a minificação de css
```

```
grunt cssmin:main // irá executar a tarefa “main” da minificação
```

```
grunt uglify:main cssmin // irá executar uma tarefa específica de uglify(“main”) e todas as tarefas de “cssmin”.
```

Outros exemplos podem ser encontrados no “getting-started” do Grunt:

<http://gruntjs.com/getting-started>

Outros pacotes de tarefas podem ser encontrados no link abaixo:

<https://github.com/gruntjs/grunt-contrib#overview>