



escola
britânica de
artes criativas
& tecnologia

Desenvolvedor Full Stack Python

Automação de tarefas com Gulp

Neste módulo conhecemos o Gulp, uma ferramenta utilizada para automação de tarefas, com ela podemos automatizar a compilação do SASS e compressão de arquivos.

Você pode consultar o código escrito durante o módulo clicando [aqui.](#)

Setup

Para começar a utilizar o Gulp é necessário fazer duas instalações.

Inicialmente instalamos o **CLI (Command Line Interface)** do Gulp de forma global:

gulp-cli: `npm install --global gulp-cli` ou `npm install --global gulp-cli`

E dentro da pasta do projeto precisamos inicializar um projeto Node, através do comando **npm init**.

Após isso instalamos o Gulp como dependência de desenvolvimento para o projeto:

npm install --save-dev gulp

Setup

Após a instalação precisamos criar um arquivo chamado **gulpfile.js**, é neste arquivo onde escreveremos as tarefas que serão automatizadas.

Também precisamos alterar o arquivo package.json, na seção scripts, ficando assim:

```
"scripts": {  
  "gulp": "gulp",  
  "test": "echo \"Error: no test specified\" && exit 1",  
},
```

Com isso podemos executar o comando `npm run gulp`, via terminal, desde que estejamos na pasta do projeto.

Tarefas

O Gulp funciona a base de tarefas, as tarefas podem ser privadas ou públicas.

Uma tarefa pública pode ser acessada através do comando:

npm run gulp NOME_DA_TAREFA

Já uma tarefa privada é acessível apenas por outra tarefa no Gulp file.

Ao executar o comando `npm run gulp` iremos receber uma mensagem de erro, informando que a tarefa padrão (default) não foi encontrada, isso porque no arquivo `gulpfile.js` não existe uma função para o `exports.default`.

Execução de tarefas

As tarefas do Gulp podem ser executadas de forma **paralela ou serial**.

As tarefas executadas de forma **paralela** **serão iniciadas no mesmo instante**, porém todo o fluxo de execução irá aguardar o término de todas as tarefas para concluir.

Esse tipo de execução é interessante quando temos tarefas pesadas e independentes, por exemplo, a compressão de imagens e compilação do SASS.

Comprimir imagens é mais demorada que compilar o SASS, logo essas duas tarefas que não estão relacionadas podem ser executadas de forma paralela.

Execução de tarefas

Já as tarefas executadas de forma **serial** fazem com que o Gulp **aguarda uma tarefa terminar para executar a próxima**, o que é útil quando temos um processo que depende de outro.

Por exemplo, reutilizando o exemplo da compressão de imagens, imagine que depois de comprimir as imagens temos a intenção de redimensiona-las.

Nesse caso seria necessário aguardar a compressão dos arquivos para depois redimensiona-las, afinal não teríamos muito ganho em redimensionar as imagens antes de comprimi-las.

Plugins

O Gulp possui vários plugins que nos ajudam nas tarefas do dia a dia, como compilação de pré-processadores CSS, minificação de imagens e outros arquivos.

Cada plugin é um novo pacote Node que instalamos, é importante que essa instalação seja feita utilizando o sufixo **--save-dev** para que a dependência seja utilizada apenas em ambiente de desenvolvimento.

Links úteis

- [Documentação completa do Gulp](#)
- [Lista de plugins](#)
- [Documentação do plugin Gulp-SASS](#)
- [Documentação do plugin Gulp-Sourcemaps](#)
- [Documentação do plugin Gulp-Uglify](#)
- [Documentação do plugin Gulp-Obfuscate](#)
- [Documentação do plugin Gulp-Imagemin](#)