



AULA 04 – Vídeo 05

Desenvolvimento

Até aqui, tivemos que reiniciar manualmente nosso servidor sempre que fizemos alguma alteração no código. Há um mecanismo, por padrão não disponível no node, chamado *hot reload*, que detecta tais alterações e reinicia automaticamente o servidor.

Para termos esse recurso disponível, podemos utilizar o nodemon.

O nodemon pode ser instalado tanto localmente (*i.e.*, somente em nosso projeto) quanto globalmente (*i.e.*, em toda a nossa máquina, disponibilizando-o para todos os projetos node), por meio do npm.

Para instalá-lo globalmente, utilizamos o comando:

```
npm install -g nodemon
```

Para instalá-lo localmente, primeiramente devemos acessar a pasta que contém o arquivo package.json. Na pasta, utilizamos o seguinte comando (note que aqui *não há o -g*, que indica que a instalação deve ser global):

```
npm install nodemon
```

Uma vez instalado, podemos ver que o arquivo package.json foi atualizado e contém o nodemon como dependência:

```
"dependencies": {  
  "express": "^4.18.1",  
  "nodemon": "^2.0.16"  
}
```

Para utilizarmos o nodemon, devemos alterar o arquivo package.json. Na seção 'scripts' em nosso arquivo package.json, podemos definir scripts para rodar em ambientes

diversos. Devemos criar um novo script chamado "dev", com o comando nodemon e o caminho para o arquivo index.js a partir da pasta que contém o arquivo package.json:

```
"scripts": {  
  "test": "echo \"Error: no test specified\" &&  
exit 1",  
  "dev": "nodemon src/index.js"  
},
```

Com isso, caso o nodemon esteja instalado globalmente, pode-se utilizar o comando

npm run dev

para iniciar a API.

Caso o nodemon esteja instalado localmente, deve-se acrescentar o comando 'npx' ao script:

```
"scripts": {  
  "test": "echo \"Error: no test specified\" &&  
exit 1",  
  "dev": "npx nodemon src/index.js"  
},
```

Com o 'npx' acrescentado, pode-se utilizar o mesmo comando para iniciar a API com o nodemon:

npm run dev