
MANUAL DE INSTALAÇÃO DO SERVIDOR DE SUPORTE À

Descrição

Este manual tem o propósito de mostrar como instalar e correr a API que dá suporte à aplicação UDO. A API foi desenvolvida sobre a plataforma de *NodeJS* e como base de dados foi utilizado o *MongoDB*.

O manual será apoiado com *screenshots* de uma consola em Windows mas para sistemas UNIX será análogo.

Pré-requisitos

Antes da importação da API é necessário a instalação dos três *softwares* abaixo mencionados e também ter acesso ao respositório que está alojado *online*.



<http://www.nodejs.org/>



<http://www.mongodb.org/downloads>



<http://git-scm.com/>

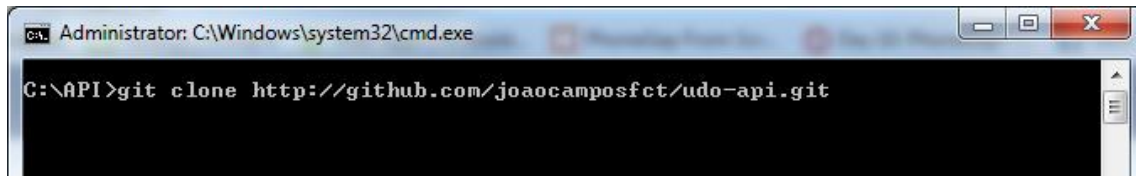
GitHub

Conta no <http://www.github.com> como colaborador no repositório da api.

Importação do repositório

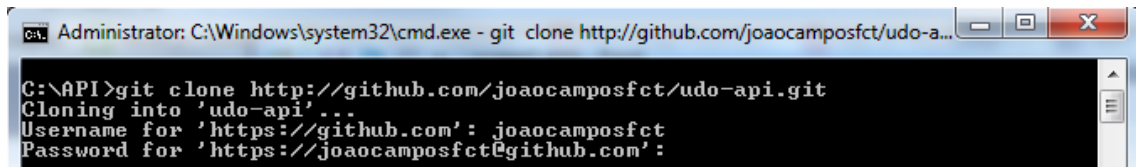
Para importar o repositório que contém a API executar o seguinte comando:

```
git clone http://github.com/joaocamposfct/udo-api.git
```



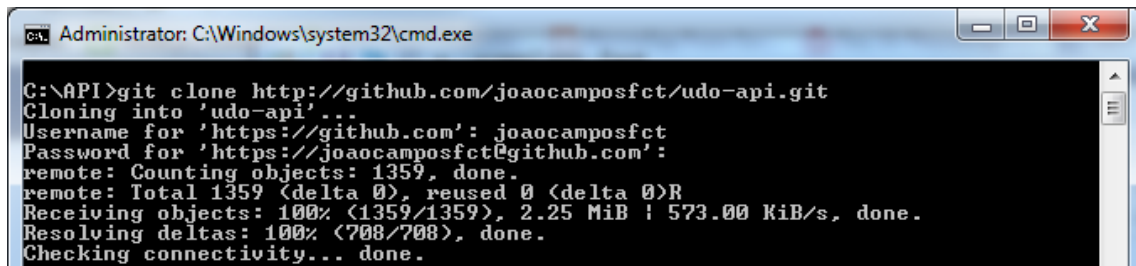
```
C:\API>git clone http://github.com/joaocamposfct/udo-api.git
```

Será requisitado o nome de utilizador e a password do github para conseguir clonar o repositório. Assim quando for necessário inserir essas informações, insira-as.



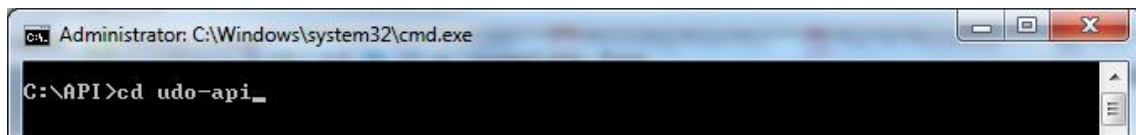
```
C:\API>git clone http://github.com/joaocamposfct/udo-api.git
Cloning into 'udo-api'...
Username for 'https://github.com': joaocamposfct
Password for 'https://joaocamposfct@github.com':
```

O comando demorará alguns segundos (dependendo da conexão) a importar o repositório, quando completar o output do comando será identico ao que se segue:

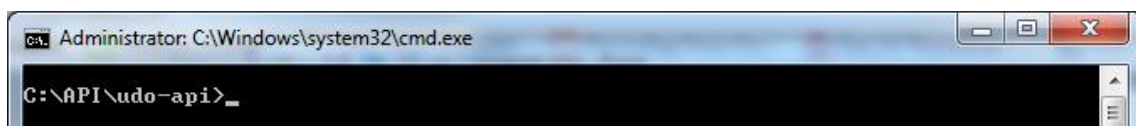


```
C:\API>git clone http://github.com/joaocamposfct/udo-api.git
Cloning into 'udo-api'...
Username for 'https://github.com': joaocamposfct
Password for 'https://joaocamposfct@github.com':
remote: Counting objects: 1359, done.
remote: Total 1359 (delta 0), reused 0 (delta 0)
Receiving objects: 100% (1359/1359), 2.25 MiB | 573.00 KiB/s, done.
Resolving deltas: 100% (708/708), done.
Checking connectivity... done.
```

Concluído o passo anterior irá reparar que dentro da pasta para onde importou o repositório estará outra pasta denominada de *udo-api*. Deverá direccionar-se para essa pasta (através do comando *cd udo-api*).



```
C:\API>cd udo-api_
```



```
C:\API\udo-api>
```

Agora que já importámos o repositório, na secção a seguir vamos instalar todos os módulos necessários para a execução da API.

Instalação dos módulos

Nesta secção iremos instalar todos os módulos essenciais à execução do servidor que sustenta a API. Para tal necessitamos de correr o comando:

npm install



Este comando irá buscar uma lista de módulos aos ficheiro *package.json* – que se encontra na raiz da pasta *udo-api* – e irá importar e instalar cada módulo localmente. Este processo poderá demorar alguns minutos dependendo da conexão que estiver a ser usada.

NOTA: PODERÁ SER NECESSÁRIO PRIVILÉGIOS DE ADMINISTRADOR PARA INSTALAR ALGUNS MÓDULOS, ASSIM, CERTIFIQUE-SE QUE TEM ESSES MESMOS PRIVILÉGIOS ANTES DE EXECUTAR O COMANDO.

Um último módulo que terá que ser instalado especificamente é o módulo *grunt-cli*. Este módulo terá que ser instalado globalmente pois o mesmo dá acesso ao comando que nos permitirá executar o servidor.

Para tal, executa-se o comando:

npm install -g grunt-cli

(A opção *-g* no comando significa que queremos a instalar o módulo globalmente.)

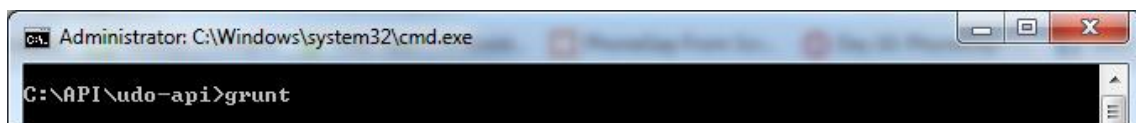


Passemos então à execução da aplicação.

Execução da API

Para executar a aplicação basta aceder à pasta *udo-api* – caso não tenha acedido previamente – e correr o comando:

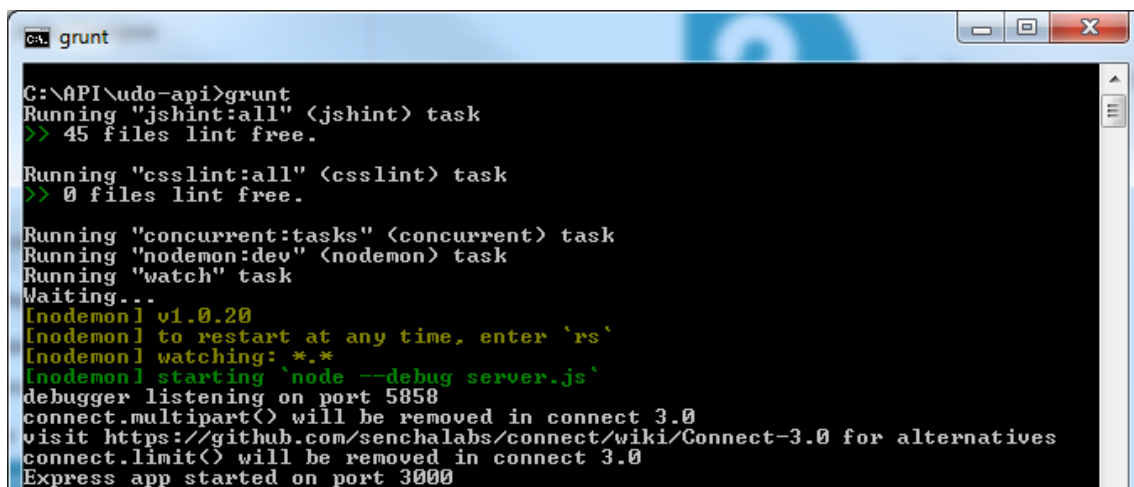
grunt



Este comando permitirá executar o servidor. Outra funcionalidade deste comando é que, enquanto está em execução, verifica se existem alterações aos ficheiros da aplicação, caso aconteça, ele reinicializará o servidor.

NOTA: CASO NÃO TENHA REINICIADO A CONSOLA ONDE ESTIVEMOS A IMPORTAR E A INSTALAR OS MÓDULOS PODERÁ SER NECESSÁRIO FAZÊ-LO CASO O COMANDO GRUNT NÃO FUNCIONE PORQUE A CONSOLA NÃO DETECTA A PRESENÇA DO MESMO. TESTE PARA VER SE O COMANDO É EXECUTADO, CASO CONTRÁRIO REINICIE A CONSOLA, ACEDA À PASTA ONDE ESTÁ A API E EXECUTE O COMANDO NOVAMENTE.

Caso o comando execute com sucesso aparecerão os seguintes outputs (ou algo similar):



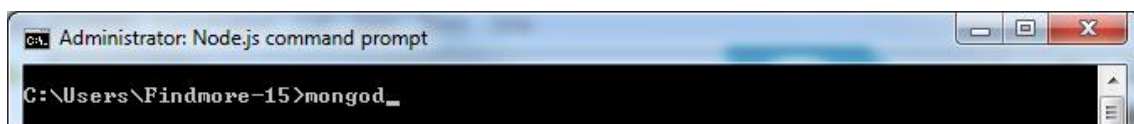
```

C:\API\udo-api>grunt
Running "jshint:all" (jshint) task
>> 45 files lint free.

Running "csslint:all" (csslint) task
>> 0 files lint free.

Running "concurrent:tasks" (concurrent) task
Running "nodemon:dev" (nodemon) task
Running "watch" task
Waiting...
[nodemon] v1.0.20
[nodemon] to restart at any time, enter `rs`
[nodemon] watching: *.*
[nodemon] starting 'node --debug server.js'
debugger listening on port 5858
connect.multipart() will be removed in connect 3.0
visit https://github.com/senchalabs/connect/wiki/Connect-3.0 for alternatives
connect.limit() will be removed in connect 3.0
Express app started on port 3000
  
```

NOTA: PARA A APLICAÇÃO ESTAR FUNCIONAL É NECESSÁRIO QUE O SERVIDOR DO MONGODB ESTEJA TAMBÉM EM EXECUÇÃO. PARA TAL EXECUTE O COMANDO MONGOD NOUTRA INSTÂNCIA DA CONSOLA OU COMO SERVIÇO.



```

C:\Users\Findmore-15>mongod_
  
```

Executar a API em ambiente de produção

O ambiente de execução por defeito é o de desenvolvimento, isto é, na consola onde a aplicação estiver a correr aparecerão mensagens de *output* com os pedidos e o código HTTP referente à resposta de cada pedido (erro, sucesso ou não modificado). A imagem seguinte representa isso mesmo:

```

C:\API\udo-api>grunt
Running "jshint:all" (jshint) task
>> 45 files lint free.

Running "csslint:all" (csslint) task
>> 0 files lint free.

Running "concurrent:tasks" (concurrent) task
Running "watch" task
Running "nodemon:dev" (nodemon) task
Waiting...
[nodemon] v1.0.20
[nodemon] to restart at any time, enter `rs`
[nodemon] watching: *.*
[nodemon] starting `node --debug server.js`
debugger listening on port 5858
connect.multipart() will be removed in connect 3.0
visit https://github.com/senchalabs/connect/wiki/Connect-3.0 for alternatives
connect.limit() will be removed in connect 3.0
Express app started on port 3000
GET / 304 614ms
GET /fights/5399bcf5f45eeee02a2f5ab7/opponents 304 48ms
GET /events/5399bcf5f45eeee02a2f5ab7/available 304 66ms
GET /friendrequests/5399bcf5f45eeee02a2f5ab7/count/received 304 68ms
GET /users/5399bcf5f45eeee02a2f5ab7/friendList 304 81ms
GET /friendrequests/5399bcf5f45eeee02a2f5ab7/count/sent 304 28ms
GET /challenges/5399bcf5f45eeee02a2f5ab7/count/pending 304 26ms
GET /fights/5399bcf5f45eeee02a2f5ab7/statistics 304 116ms
GET / 304 52ms
GET /fights/5399bcf5f45eeee02a2f5ab7/mine 200 97ms
  
```

Para correr em ambiente de produção temos que criar e afectar uma variável de ambiente denominada `NODE_ENV`, para tal, no Windows corremos o seguinte comando:

set NODE_ENV=production

```

Administrator: C:\Windows\system32\cmd.exe

C:\API\udo-api>set NODE_ENV=production_
  
```

(Para UNIX o comando seria `exports NODE_ENV=production`)

E em seguida correr novamente o comando “grunt” para executar a aplicação.

```

C:\API\udo-api>grunt
Running "jshint:all" (jshint) task
>> 45 files lint free.

Running "csslint:all" (csslint) task
>> 0 files lint free.

Running "cssmin:combine" (cssmin) task
File public/build/css/dist.min.css created: 130.91 kB → 98.18 kB

Running "uglify:production" (uglify) task
File public/build/js/dist.min.js created: 2.38 MB → 1.55 MB

Running "concurrent:tasks" (concurrent) task
Running "nodemon:dev" (nodemon) task
Running "watch" task
Waiting...
[nodemon] v1.0.20
[nodemon] to restart at any time, enter 'rs'
[nodemon] watching: *.*
[nodemon] starting 'node --debug server.js'
debugger listening on port 5858
connect.multipart() will be removed in connect 3.0
visit https://github.com/senchalabs/connect/wiki/Connect-3.0 for alternatives
connect.limit() will be removed in connect 3.0
Express app started on port 3000
  
```

Sempre que se quiser alterar o ambiente de execução basta afectar a variável `NODE_ENV` para 'development' para o ambiente de desenvolvimento e para 'production' caso queiramos ir para ambiente de produção.

NOTA IMPORTANTE: A CADA MUDANÇA DE AMBIENTE DE EXECUÇÃO A APLICAÇÃO CONECTA-SE A UMA BASE DE DADOS DIFERENTE. A "UDO-DEV" É A BASE DE DADOS DE DESENVOLVIMENTO E A "UDO" É A DE PRODUÇÃO.

E assim termina o manual de instalação do servidor de suporte à UDO.