

## Mouro

### 1. Descripción

Mouro es un gestor de base de datos, el cual se comunica con una base de datos 'sqlite' para almacenar las credenciales emitidas por el didi-server y los distintos issuers, permite el acceso a los usuarios de didi a sus credenciales y verificar el estado de los mismos.

### 2. Ejecución

#### a. Instalar Node:

Este módulo utiliza npm como gestor de paquetes, el cual viene integrado con Node: <https://nodejs.org/en/>

#### b. Instalar y ejecutar cliente de swarm (opcional):

Este módulo utiliza swarm para generar y almacenar los backups:  
[https://swarm-guide.readthedocs.io/en/latest/node\\_operator.html#download-pre-compiled-swarm-binaries](https://swarm-guide.readthedocs.io/en/latest/node_operator.html#download-pre-compiled-swarm-binaries)

#### c. Instalar dependencias:

Abrir una terminal en la carpeta principal del módulo (a la altura del archivo 'package.json') y ejecutar el comando '**npm install**'

#### d. Ejecutar mouro:

A la misma altura, ejecutar el comando '**\$ENV\_VARS npm run start**', donde \$ENV\_VARS son las distintas variables de entorno del módulo.

Ej:

```
'PORT=3001 SWARM_URL=http://localhost:8500 npm run start'
```

#### e. Variables de entorno:

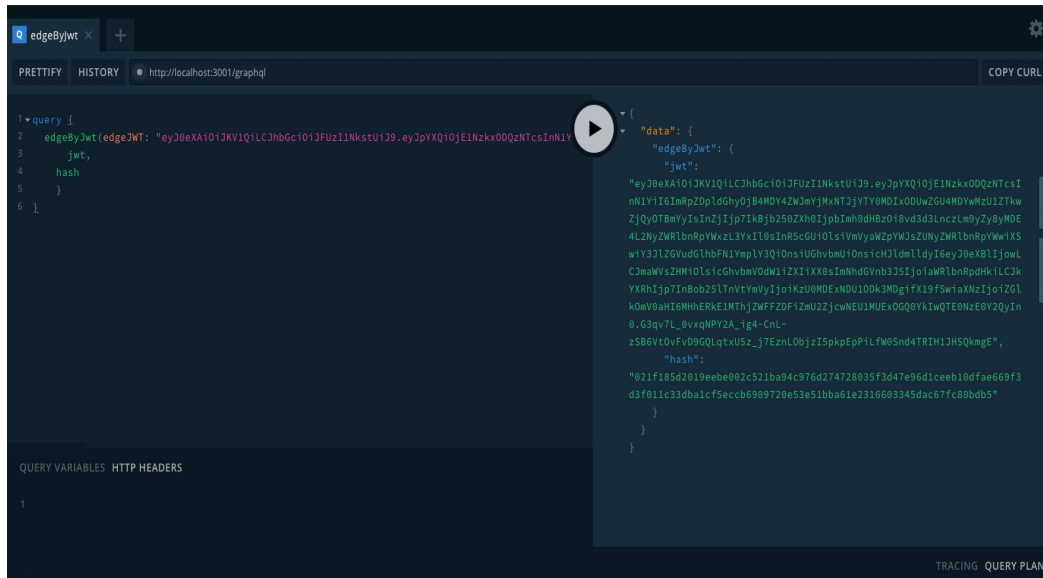
Nombre	Obligatorio	Descripción	notas	ej
DIDI_SERVER_DID	Si	Did correspondiente al didi-server	La mayor parte de los métodos solo pueden ser llamados por el didi-server	0xDFA518ceaEd1bfe6f704E51A18d4bB0A14714cd2
PORT	No	puerto en que corre	Si no está definido, se tomará el puerto 3000 o el primer puerto disponible	3001
SWARM_URL	No	dirección en la que está corriendo swarm	Si no está definido, no se generan backups	<a href="http://192.168.2.113:8500">http://192.168.2.113:8500</a>
SQLITE_FILE	No	el nombre del archivo	Si no está definido, se genera un archivo por usuario, si lo está, se guarda todo ahí	<a href="http://localhost:8500">http://localhost:8500</a>

### 3. Comandos

Una vez iniciado el programa, se mostrará la url con la que se pueden realizar queries sobre Mouro de forma directa:

## Comandos

Una vez iniciado el programa, se mostrará la url con la que se pueden realizar queries sobre Mouro de forma directa:



Los comandos existentes son los siguientes:

Nota: Algunas de las llamadas requieren de la generación de un token vacío firmado por el DID del usuario que realiza la llamada, este se ingresa en la sección “HTTP HEADERS” y tiene el siguiente formato:

```
{ "Authorization": "Bearer eyJ0eXAiOiJKV1QiLCJhbGciOiJIUzI1NiIsInR5cCI6IkpXZWQieS1ODAyMjY0NzEsImV4cCI6MTU0MDIyNjkzMS4xMTIsImZlcnl6cy96bmRpZDpldGhyOjB4REZBNTE4Y2VhRWQxYmZINmY3MDRFNTFBMThkNGJCMEExNDcxNgNkMiJ9.NdihoBy4uEMsCLaitRIETQ-fnB2SGJLyxfApekded_42T9IZHJklcGRQmxweOaer-UuG3A4R7-LOQYN76MiviQCgA" }
```

**me:** Retorna la info del usuario a partir del token.

Ej:

```
query {
  me() {
    did
  }
}
```

**findEdges:** Retorna los certificados del dueño del token.

Ej:

```
query {  
  findEdges(toDid: "did:ethr:....") {  
    jwt  
  }  
}
```

**Hash:** Retorna el hash a utilizarse para la obtención del último backup usando swarm.

Ej:

```
query {  
  hash(did: "did:ethr:....")  
}
```

**edgeByJwt:** Obtiene un certificado a partir de su jwt.

Ej:

```
query {  
  edgeByJwt(edgeJWT: "...", did: "did:ethr:....") {  
    jwt,  
    hash  
  }  
}
```

**addEdge:** Guarda un nuevo certificado.

Ej:

```
query {  
  addEdge(edgeJWT: "...", did: "did:ethr:....") {  
    jwt,  
    hash  
  }  
}
```

#### 4. Backup

Si se recibe la variable de entorno "SWARM\_URL" se generará backups en la url especificada por ella en cada modificación de la base de datos, generandose un nuevo hash cada vez que se agrega o revoca un certificado. Estos hashes se almacenan tanto en didi-server como mouro y pueden recuperarse con la query desde mouro:

```
query {  
  hash(did: "did:ethr:....")  
}
```

O sobre la base de datos de didi-server con la query:

```
db.getCollection('users').find({did: "did:ethr:..."}, {backupHash: 1})
```

Una vez obtenido ese hash, puede recuperarse el último backup utilizando el comando:

```
curl $SWARM_URL/bzz-raw:/$HASH/ >> $NOMBRE_DE_ARCHIBO
```

Donde **\$SWARM\_URL** es la url donde se encuentra corriendo swarm, **\$HASH** es el hash obtenido en el paso anterior y **\$NOMBRE\_DE\_ARCHIBO** es el nombre de archivo al que bajar el backup.