

Manual Técnico para su despliegue en Windows

1. Requerimientos técnicos

Es necesario para correr dicha aplicación los siguientes requerimientos técnicos ya instalados dentro servidor/equipo dependiendo el sistema operativo para el sistema:

- Node.js en su última versión. (Preferentemente la versión '18.12.1').
- NPM en su última versión. (Preferentemente la versión '8.19.2').
- MongoDB Compass en su última versión. (Preferentemente la versión '1.33.1').
- POSTMAN en su ultima versión.
- Consola de comando de preferencia (Preferentemente Windows Powershell actualizada).
- IDE de preferencia (Visual Studio Code , configurado en js, actualziado).

2. Clonar repositorio

Clonar el repositorio de Github: <https://github.com/TheGeneralisimo/big-headportal-transparencia>

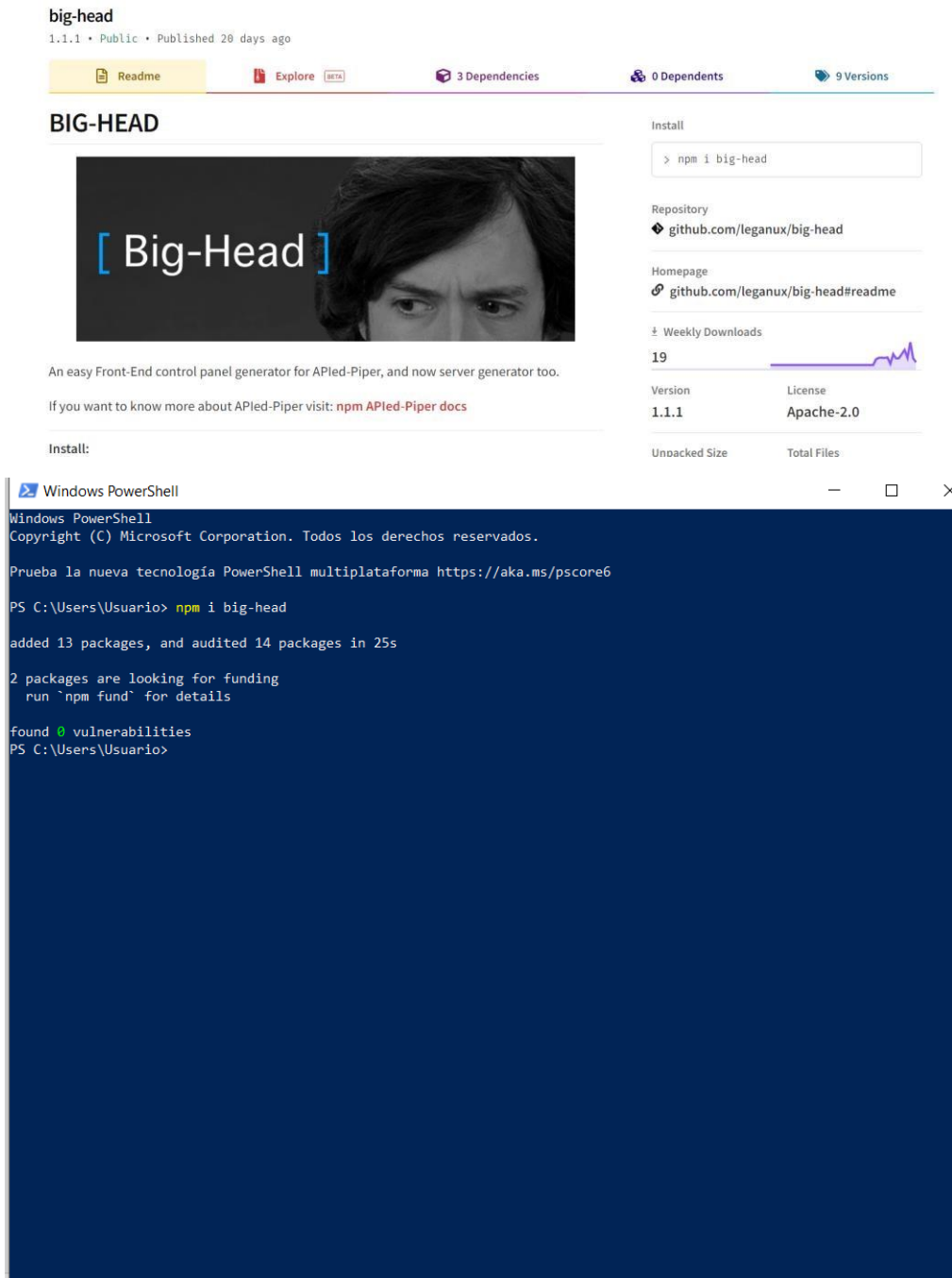
The screenshot shows the GitHub repository page for 'TheGeneralisimo / big-head-portal-transparencia'. The repository is public and has 18 commits. The file list includes:

File	Commit Message	Commit Time
.idea	cambios	yesterday
apiend_piper_server	zulema version	22 hours ago
console	new update	22 hours ago
data	zulema version	22 hours ago
def	new version	2 days ago
.gitattributes	Initial commit	6 days ago
LICENSE	new version	2 days ago
Manual Tecnico.pdf	18/11/2022	4 days ago
Manual de Usuario.pdf	Update Manual de Usuario.pdf	21 hours ago
README.md	Update README.md	4 days ago

The right sidebar shows the repository's description: 'Proyecto de prácticas profesionales para la Secretaría de Contraloría del Estado de Hidalgo para la Dirección General de Políticas de Transparencia y Anticorrupción. Basada en big-head'. It also includes links to the README, MIT license, 0 stars, 1 watching, and 1 fork. The Releases section indicates 'No releases published' and provides a link to 'Create a new release'.

3. Configuración de big-head en Node.js

Dentro de la consola de uso de node.js o cualquier consola de desarrollo, instalar el siguiente recurso npm con el comando: *"npm i big-head"*




The image shows the npm package page for **big-head** and a Windows PowerShell terminal window.

big-head
1.1.1 • Public • Published 20 days ago

Readme Explore beta 3 Dependencies 0 Dependents 9 Versions

BIG-HEAD



An easy Front-End control panel generator for APled-Piper, and now server generator too.

If you want to know more about APled-Piper visit: [npm APled-Piper docs](#)

Install:

Install

```
> npm i big-head
```

Repository
github.com/leganux/big-head

Homepage
github.com/leganux/big-head#readme

Weekly Downloads
19

Version
1.1.1

License
Apache-2.0

Unpacked Size
Total Files

Windows PowerShell

```
Windows PowerShell
Copyright (C) Microsoft Corporation. Todos los derechos reservados.

Prueba la nueva tecnología PowerShell multiplataforma https://aka.ms/pscore6

PS C:\Users\Usuario> npm i big-head

added 13 packages, and audited 14 packages in 25s

2 packages are looking for funding
  run `npm fund` for details

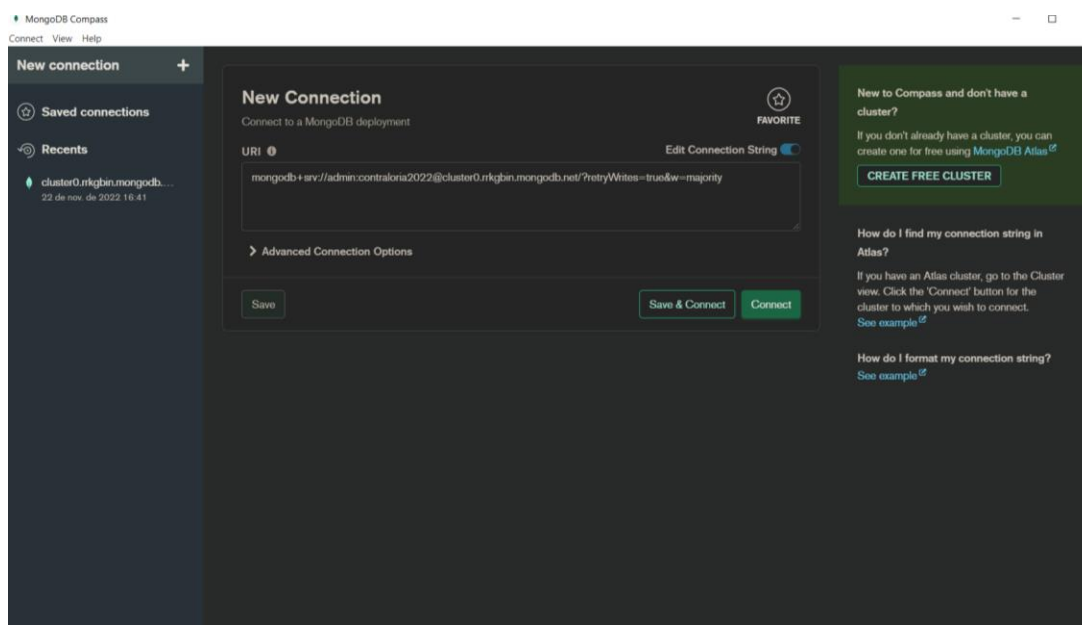
found 0 vulnerabilities
PS C:\Users\Usuario>
```

4. Configuración de URI MongoDB Compass

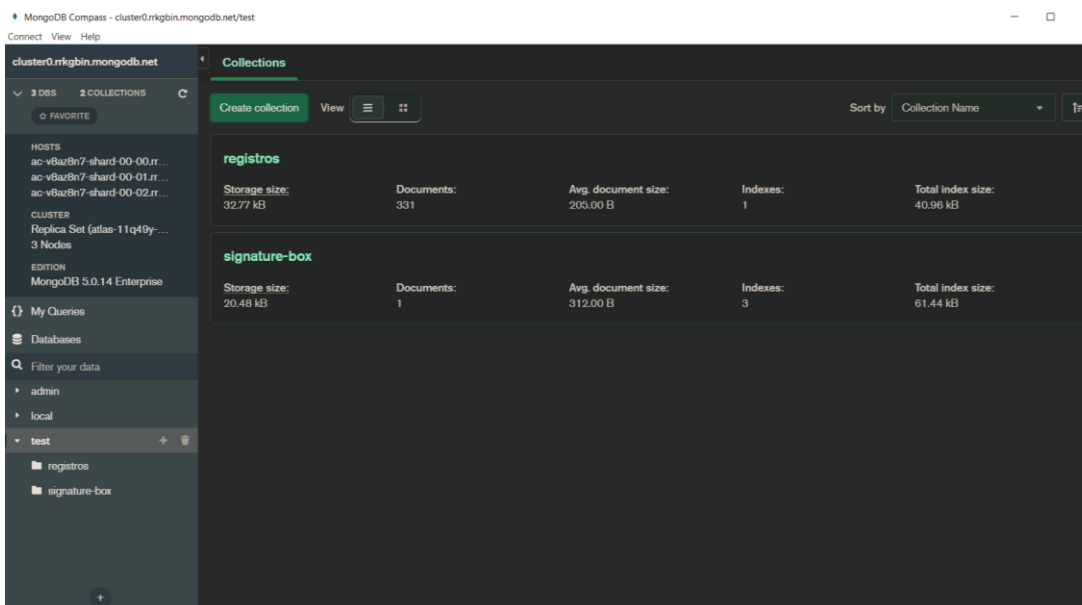
Dentro del panel de administración de MongoDB Compass realizar una conexión con el siguiente URI:

`"mongodb+srv://admin:contraloria2022@cluster0.rkgbin.mongodb.net/?retryWrites=true&w=majority"`

Donde se alberga la base de datos en un clúster en línea.

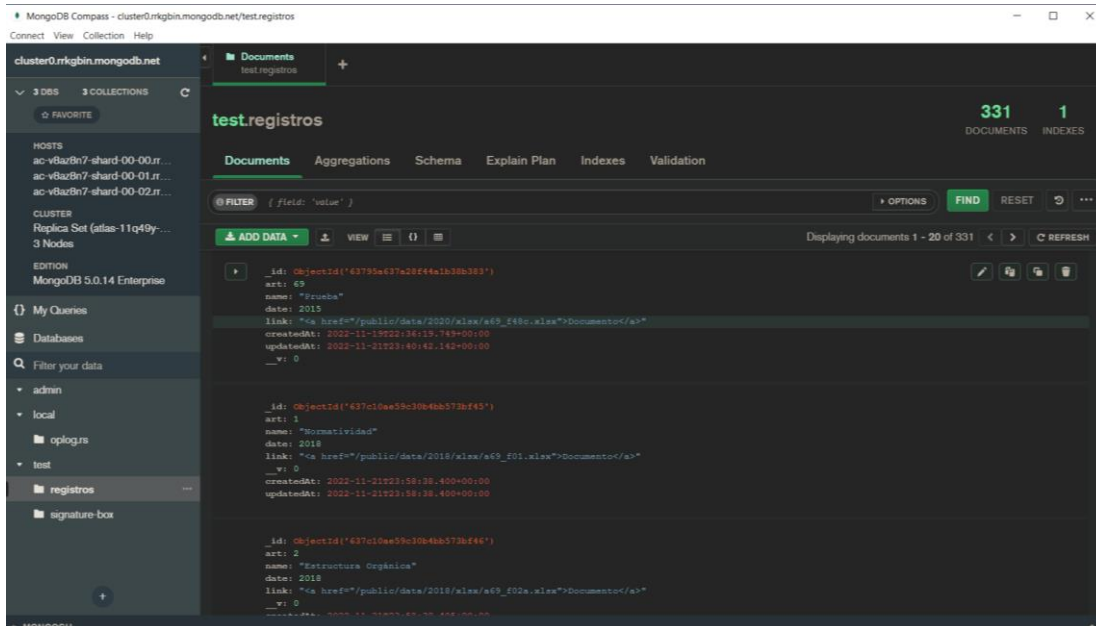


En caso de existir un error en la conexión del clúster, se proporciona el **usuario:admin** y las **password:contraloria2022**

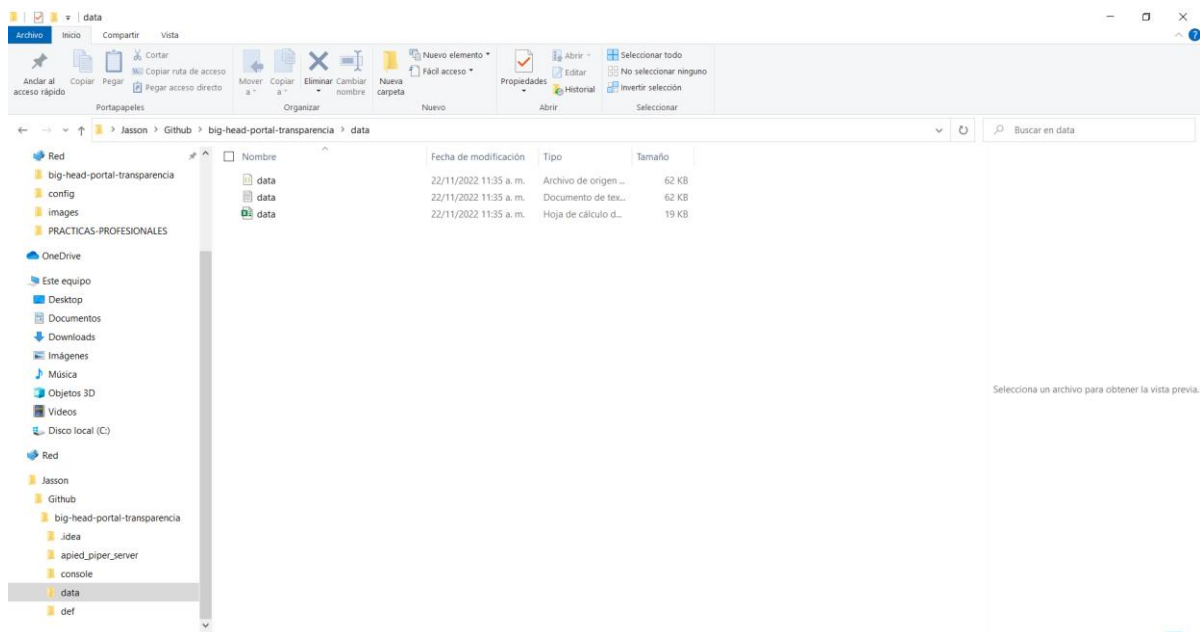


5. Carga de base de datos MongoDB Compass a través de POSTMAN en formato JSON

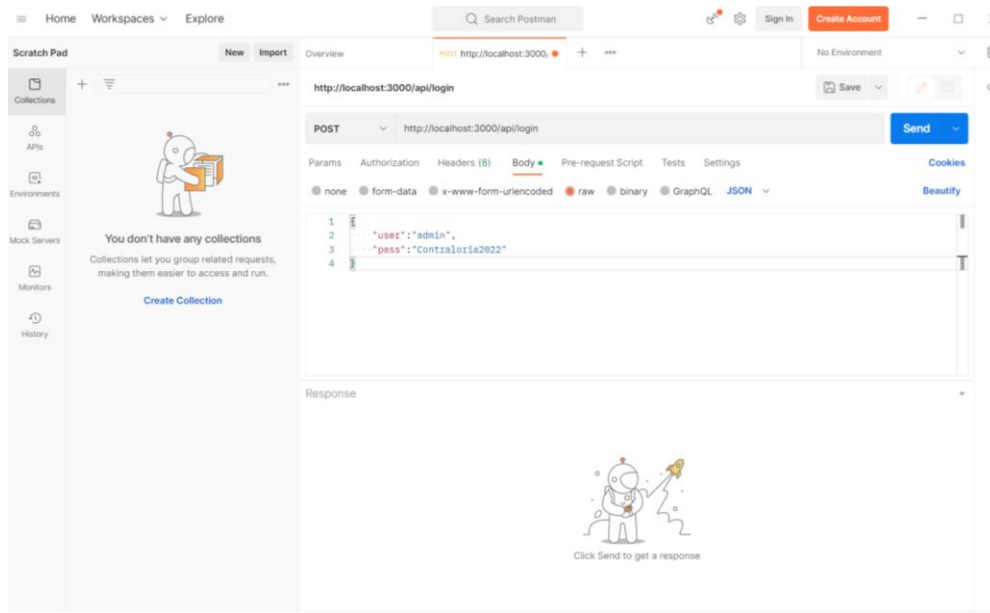
Actualmente esta cargada la información ya preestablecida dentro de la base de datos de Mongo. Se puede visualizar en la base “test/registros”.



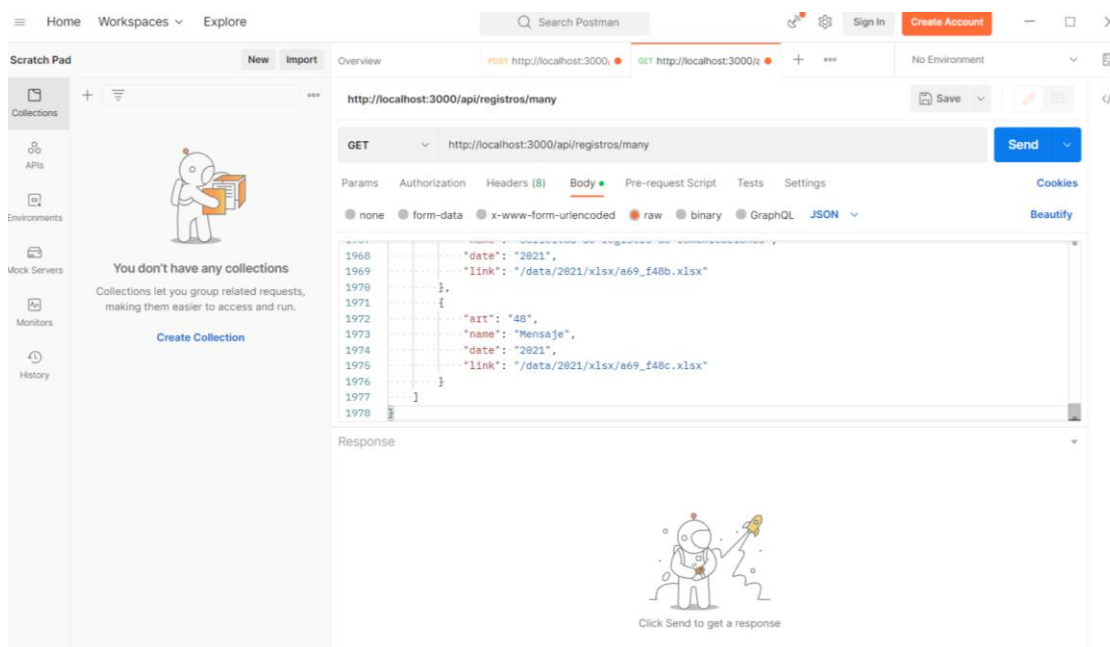
En caso de realizar una alta masiva Dentro del repositorio “big-head-portal-transparencia/data” está presente en documento JSON los registros de ruta de información en el archivo “data.json” que debe ser cargado utilizando la herramienta de POSTMAN.



Dentro de la herramienta POSTMAN con la aplicación levantada se debe. Acceder en el LOGIN, redactar en etiquetas en tipo raw, tipo JSON. La carga de user:"admin" y pass:"Contraloria2022" para adquirir el token de autenticación cifrada. Con la sentencia POST `http://localhost:3000/api/login`.

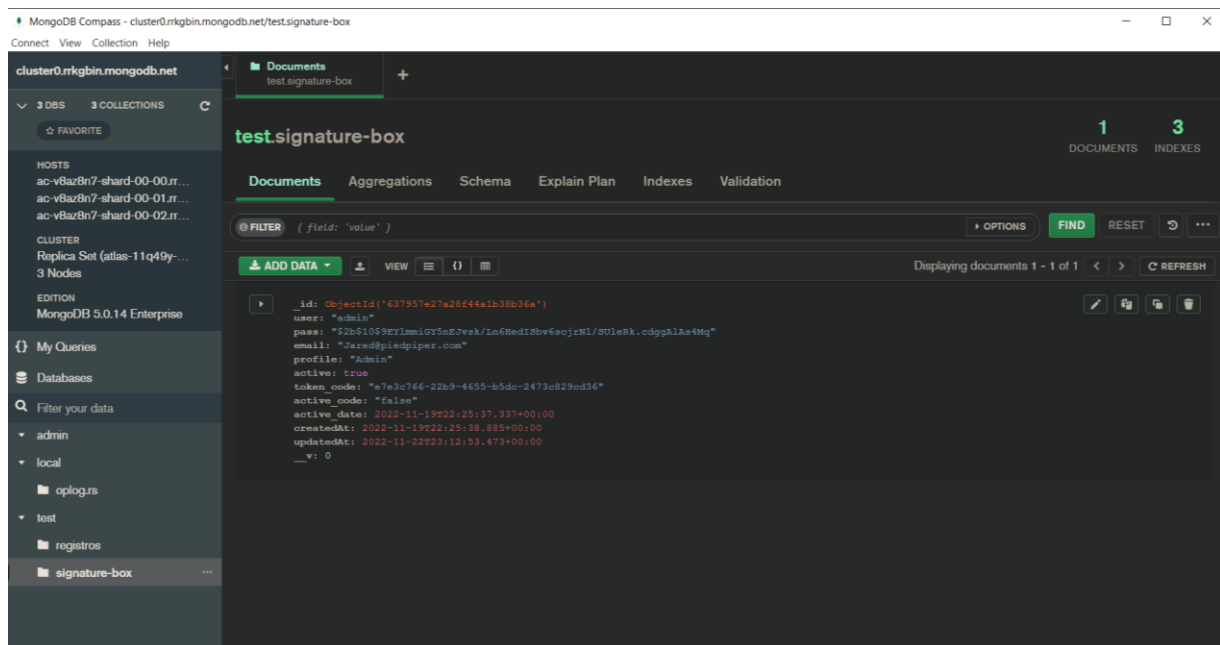


Posteriormente se debe redactar en la etiqueta de Autenticación el token cifrado de tipo "Bearer token"; posteriormente cargar en el raw, tipo JSON, el documento `data.json` bajo la sentencia

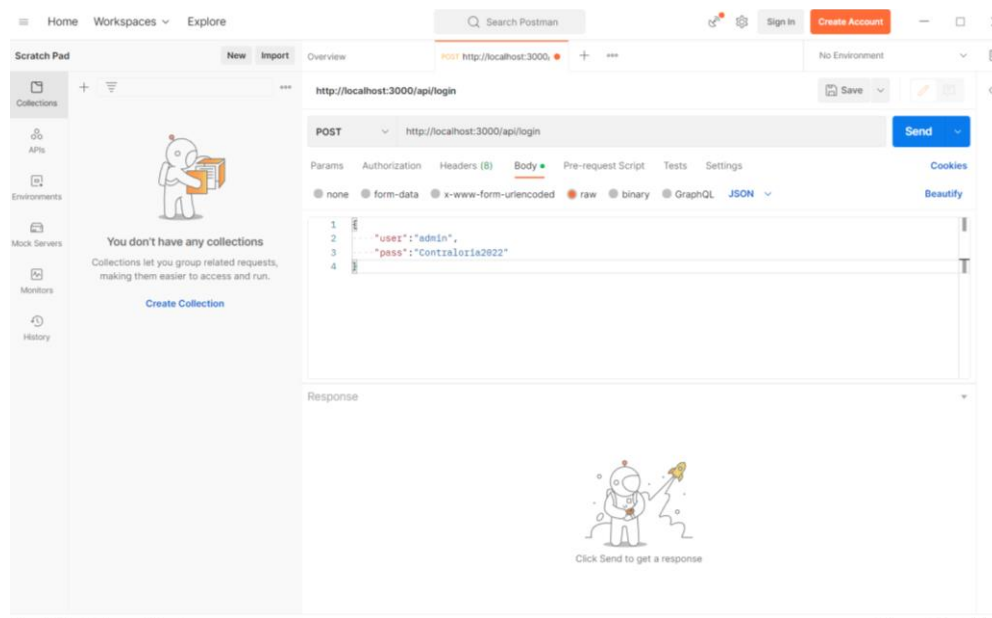


6. Carga de usuarios en MongoDB Compass a través de POSTMAN en formato JSON

Actualmente está cargada la información ya preestablecida dentro de la base de datos de Mongo. Se puede visualizar en la base “test/signatura-box.



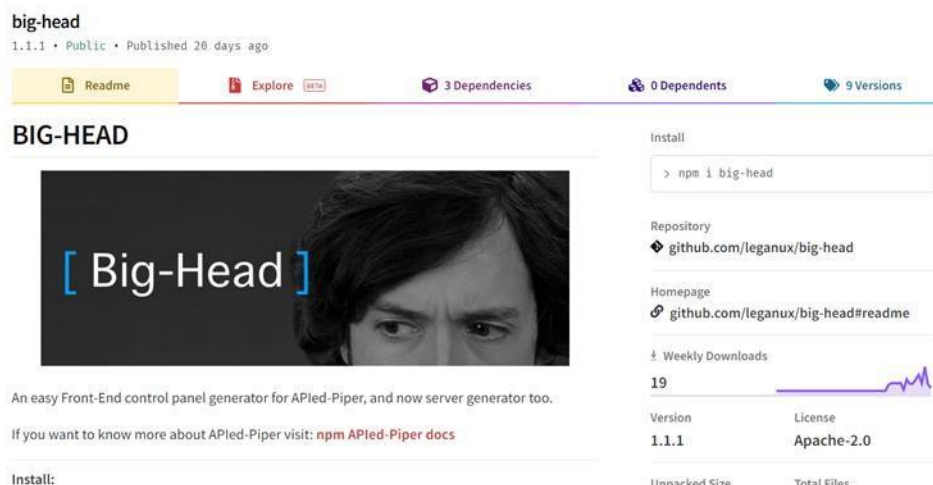
Dentro de la herramienta POSTMAN con la aplicación levantada se debe. Acceder en el LOGIN, redactar en etiquetas en tipo raw, tipo JSON. La carga de user:"admin" y pass:"Contraloria2022" para adquirir el token de autenticación cifrada. Con la sentencia POST <http://localhost:3000/api/login>.



Posteriormente se debe redactar en la etiqueta de Autenticación el token cifrado de tipo “Bearer token”; posteriormente cargar en el raw, tipo JSON, el documento data.json bajo la sentencia

7. Documentación de big-head

Dicho proyecto se basa en el paquete big-head para la creación de un panel de control de información. Favor de visualizar en el sitio <https://www.npmjs.com/package/big-head> para mayor documentación.



8. Configuración de apied_piper_server

En caso de algún error, duda, favor de revisar la documentación en el sitio <https://www.npmjs.com/package/big-head> para mayor documentación.

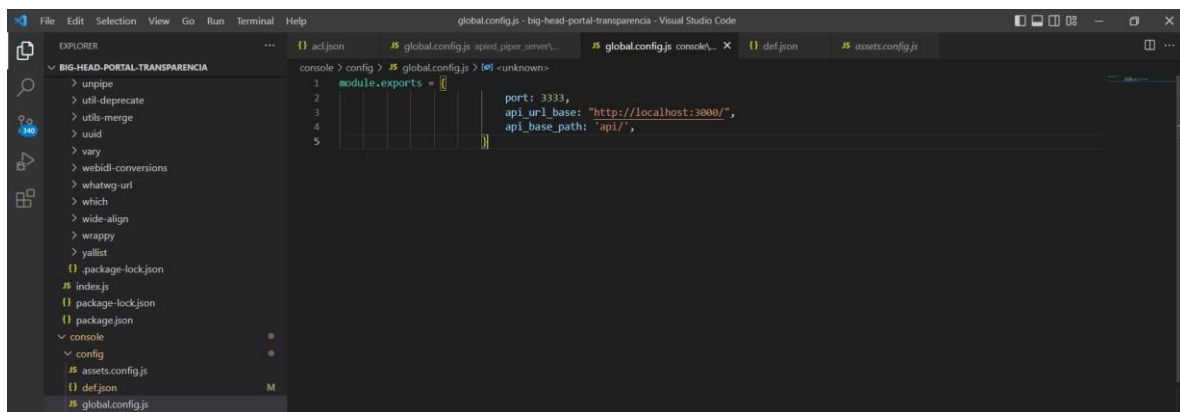
Dentro del portafolio “apied_piper_server” entrar a la carpeta “\big-head-portaltransparencia\apied_piper_server\config” , y seleccionar el archivo “global.config” para realizar las configuraciones correspondientes al uso del puerto establecido por el backend del api. Al igual que usuarios de administradores, creación de usuarios, cambio de servidores de bases de datos.

```
File Edit Selection View Go Run Terminal Help
global.config.js - Visual Studio Code
global.config.js
1 module.exports = {
2   port: "3000",
3   base_path: "/api/",
4   db_uri: "mongodb+srv://admin:contraloria2022@cluster0.rkghin.mongodb.net/portaltransparencia?retryWrites=true&w=majority",
5   user: 'admincontraloria',
6   pass: 'Contraloria2022',
7 }
```

9. Configuración de console (big-head console)

En caso de algún error, duda, favor de revisar la documentación en el sitio <https://www.npmjs.com/package/big-head> para mayor documentación.

Dentro del portafolio “console” entrar a la carpeta “\big-head-portal-transparencia\console\config”, y seleccionar el archivo “global.config” para realizar las configuraciones correspondientes al uso del puerto establecido por el frontend del panel.



10. Despliegue de apied_piper_server

En caso de algún error, duda, favor de revisar la documentación en el sitio <https://www.npmjs.com/package/big-head> para mayor documentación.

Dentro de la carpeta “apied_piper_server” ya configurada, se debe ejecutar desde terminal el comando “npm start”. En caso de existencia de algún error dentro de la actualización de los “node_modules” favor de eliminar la carpeta “node_modules” ya configurada al igual que el archivo “package.lock.json”; posteriormente debe ejecutar la reinstalación de las dependencias con el comando “npm install”

11. Despliegue de console

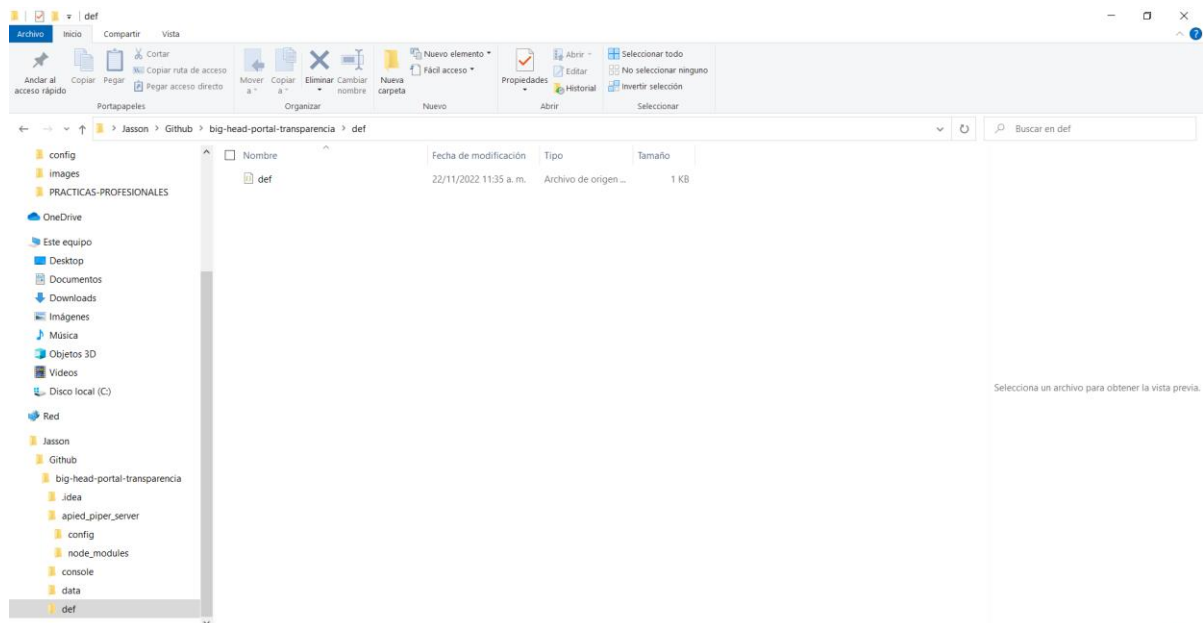
En caso de algún error, duda, favor de revisar la documentación en el sitio <https://www.npmjs.com/package/big-head> para mayor documentación.

Dentro de la carpeta “console” ya configurada y el server apied_piper_server corriendo, se debe ejecutar desde terminal el comando “npm start”. En caso de existencia de algún error dentro de la actualización de los “node_modules” favor de

eliminar la carpeta “node_modules” ya configurada al igual que el archivo “package.lock.json”; posteriormente debe ejecutar la reinstalación de las dependencias con el comando “npm install”

12. Identificación de archivo de definición

Primero necesita crear un archivo de definición, estos archivos lo ayudarán a construir su servidor y consola. para obtener más información, el documento de definición se encuentra en la carpeta “big-head-portal-transparencia\def”



```
1 {
2   "registros": {
3     "operation": {
4       "all": true
5     },
6     "definition": {
7       "art": {
8         "type": "number"
9       },
10      "name": {
11        "type": "string"
12      },
13      "date": {
14        "type": "number"
15      },
16      "link": {
17        "type": "string"
18      }
19    },
20    "datatable_search_fields": [
21      "name"
22    ]
23  }
24 }
25
```

13. Construcción de apied_piper_server

En caso de algún error, duda, favor de revisar la documentación en el sitio <https://www.npmjs.com/package/big-head> para mayor documentación.

Para la construcción del servidor de microservicios, se debe seguir el siguiente comando: “bighead service” con las siguientes opciones:

*-d, --definition <definición...> La carpeta de archivo completa y la ruta de la definición JSON. Obligatorio ***

-l, --location <ubicación...> La carpeta de archivo completa y la ruta donde se creará el proyecto. Defecto: ./

-b, --base <base...> Servidor de ruta base APled-Piper. Predeterminado: /api/

-p, --port <puerto...> Puerto donde se ejecutará Big-Head. Predeterminado: 3000

-m, --mdb_uri <mdb_uri...> El URI de MongoDB para almacenar datos. Predeterminado: mongodb://localhost:27017/apied_piper

-u, --user <usuario...> Usuario administrador por defecto. Predeterminado: PIED

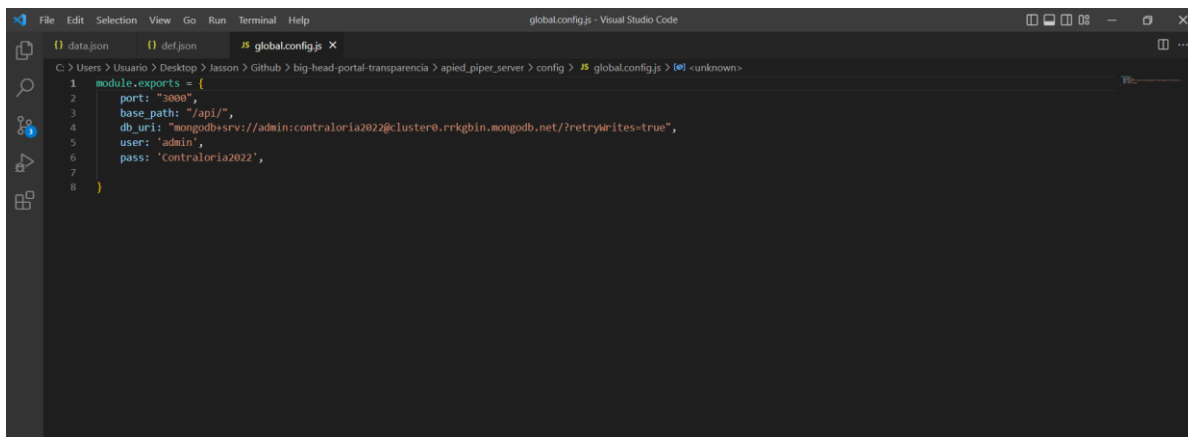
-pw, --password <contraseña...> Contraseña de administrador por defecto. Valor predeterminado: HBO_Silicon33

-h, --help muestra ayuda para el comando

Por lo tanto, para realizar la construcción del servidor se debe llegar el siguiente comando:

```
bighead service -d "big-head-portal-transparencia/def/def.json" -m  
"URI de la base de datos MONGO"
```

Posteriormente a la creación se debe cambiar la configuración de usuario y password. Dentro del archivo global.config.



En caso de existencia de algún error, revisar la instalación de la dependencia “big head”; o dentro de la actualización de los “node_modules” favor de eliminar la carpeta “node_modules” ya configurada al igual que el archivo “package.lock.json”; posteriormente debe ejecutar la reinstalación de las dependencias con el comando “npm install” .

14. Construcción de console

En caso de algún error, duda, favor de revisar la documentación en el sitio <https://www.npmjs.com/package/big-head> para mayor documentación.

Para la construcción de la consola, se debe seguir el siguiente comando: “bighead console” con las siguientes opciones:

*-d, --definition <definición...> La carpeta de archivo completa y la ruta de la definición JSON. Obligatorio ***

-l, --location <ubicación...> La carpeta de archivo completa y la ruta donde se creará el proyecto. Defecto: ./

-u, --uri <uri...> APled-Piper url server. Predeterminado: http://localhost:3000/

-b, --base <base...> Servidor de ruta base APled-Piper. Predeterminado: API/

-p, --port <puerto...> Puerto donde se ejecutará Big-Head. Predeterminado: 3333

-h, --help muestra ayuda para el comando

Por lo tanto, para realizar la construcción del servidor se debe llegar el siguiente comando:

bighead console -d "big-head-portal-transparencia/def/def.json"

En caso de existencia de algún error, revisar la instalación de la dependencia "big head"; o dentro de la actualización de los "node_modules" favor de eliminar la carpeta "node_modules" ya configurada al igual que el archivo "package.lock.json"; posteriormente debe ejecutar la reinstalación de las dependencias con el comando "npm install" .