**School Teachers and Students Distribution (STSD) API**

STSD API dibuat menggunakan *framework* Node.js dan beberapa *library/module* utama seperti:

1. Express (*web framework* untuk Node.js)
2. jsonwebtoken (Aplikasi JSON Web Token untuk Node.js)
3. mysql (*driver* Node.js untuk MySQL)
4. Plotly (*graphing library* untuk Node.js)

STSD API menyediakan layanan pengambilan dataset dalam format JSON dari berbagai *website* Open Data yang ada. Pada dokumentasi ini digunakan dataset yang bersumber dari NYC Open Data. Selain itu API ini juga menyediakan layanan visualisasi bagi data yang terdapat di server.

----------------------------------------------------------------------------------------------------------------

**Authentication**

*Authentication* dilakukan secara sederhana menggunakan *token* yang dihasilkan dari jsonwebtoken menggunakan *user/client id* dan *secret* yang terdapat di *server. Token* ini kemudian akan disimpan di *database* dan dikirimkan sebagai *response.* Setiap kali *user* memanggil API *Endpoints* **/api/data** dan **/api/visual** *user* harus menyertakan *token* yang sudah diterima sebelumnya pada *header request* sebagai **x-access-token**. *Token* akan berganti setiap kali *login.*

----------------------------------------------------------------------------------------------------------------

**Responses**

Beberapa kode *response* yang ada ketika suatu *request* dilakukan.

**200 OK**

*request* berhasil dilaksanakan.

**204 No Content**

*request* telah dilakukan, tetapi *client* tidak harus berganti dari *current page.* Hal ini terjadi jika pada saat *login email* dan *password* tidak cocok, atau *email* tidak terdaftar di *database.*

**400 Bad Request**

*server* tidak dapat memproses *request* karena terdapat *syntax* yang tidak valid. Hal ini terjadi jika format *request* tidak sesuai dengan seharusnya, misalnya dalam melakukan registrasi atau *login* pada *request body* tidak terdapat *email* atau *password.*

**401 Unauthorized**

I*request* yang dilakukan membutuhkan otentikasi dan *user* tidak *login* terlebih dahulu atau memberikan *credentials* yang dibutuhkan. Hal ini terjadi ketika mencoba melakukan *request* ke **/api/data** atau **/api/visual** tanpa menyertakan *token.*

**500 Internal Server Error**

*server* tidak dapat memproses *request* untuk alasan yang tidak diketahui. Contohnya ketika melakukan *request* ke **/api/data** atau **/api/visual** dengan menyertakan *token,* tetapi untuk alasan yang tidak diketahui *server* tidak dapat mengotentikasi *token* tersebut.

----------------------------------------------------------------------------------------------------------------

**API Endpoints**

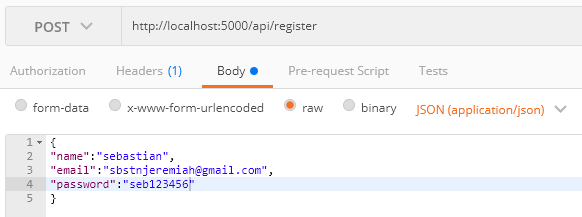
**/api/register**

melakukan registrasi untuk menerima token agar dapat menggunakan *resource* yang disediakan.

HTTP Method: **POST**

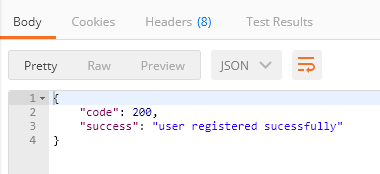
Header: -

Body:



Response:

(contoh jika berhasil)

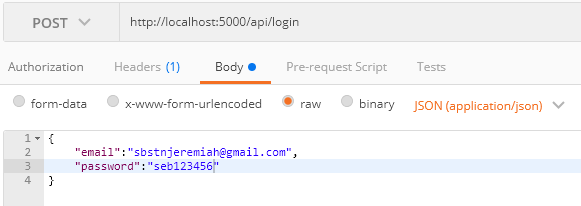


**/api/login**

HTTP Method: **POST**

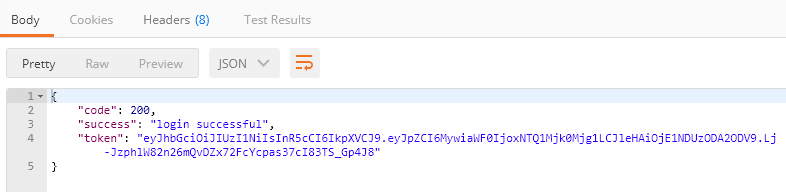
Header: -

Body:



Response:

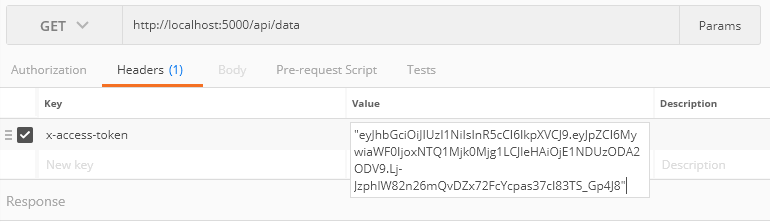
(contoh jika berhasil)



**/api/data**

HTTP Method: **GET**

Header:



Body: -

Untuk saat ini data yang diambil masih tertentu, ditentukan langsung melalui URL yang dimasukkan dalam *source code.* Untuk kedepannya, akan digunakan URL dari *request body* pada HTTP *request*, tetapi perlu diperhatikan URL tersebut harus merupakan URL dari dataset dalam format JSON (untuk lebih jauh mungkin dapat digunakan *converter* ke JSON, seperti csvtojson untuk Node.js). Juga dalam pemrosesan ini dataset akan disimpan dalam *server,* tetapi untuk nama *file* sendiri masih belum ditentukan oleh *user*.

Jadi akan terdapat dua atribut dalam *request* body, yaitu:

* **url**, yang berisi URL dataset yang diinginkan
* **filename**, yang berisi nama *file* untuk dataset tersebut dalam *server*

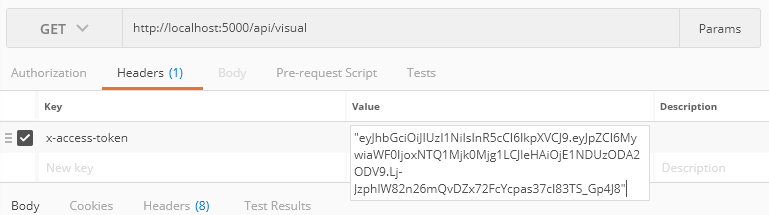
Response:



**/api/visual**

HTTP Method: **GET**

Header:



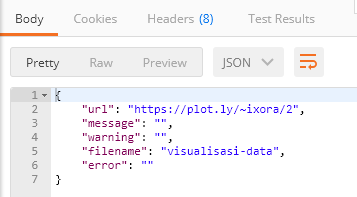
Body: -

Untuk saat ini, dataset yang digunakan untuk visualisasi masih merupakan *dummy* yang terdapat di dalam *folder* proyek. Rancangan sebenarnya adalah ketika *user* melakukan *request* ke **/api/data**, dataset yang di-*request* akan disimpan dalam sebuah file JSON di *server* dengan nama *file* ditentukan *user* (Untuk penyimpanan dataset dalam *file* sendiri sudah dapat dilakukan). Setelah itu dalam *request* body akan ada atribut dimana *user* menentukan dua dataset yang akan menjadi perbandingan dalam visualisasi, serta nama *file* hasil visualisasi.

Jadi nantinya akan ada tiga atribut dalam *request body*, yaitu:

* **filename1**, berisi nama file pertama
* **filename2**, berisi nama file kedua
* **vfilename,** berisi nama file hasil visualisasi

Response:



Atribut **url** berisi URL hasil visualisasi dataset.

---------------------------------------------------------------------------------------------------------------------------

**Usage**

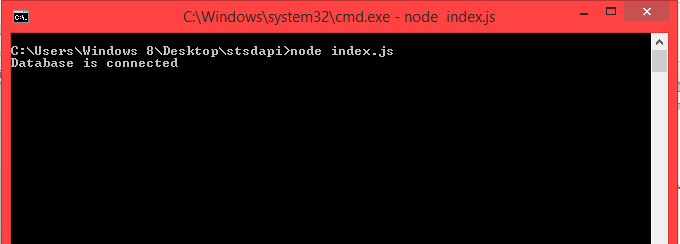
Hingga saat ini STSD API masih dijalankan di *localhost.* Pengujian API dilakukan melalui Postman.

**Getting Started**

Karena API ini masih dijalankan di localhost, untuk menjalankan *download* terlebih dahulu proyek ini melalui <https://github.com/sebastianjer/apiproject>.

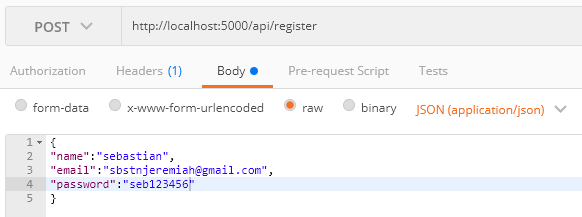
Jalankan Apache dan MySQL pada XAMPP, lalu *import* file auth.sql yang ada pada folder proyek.

Kemudian pada folder proyek buka *Command Prompt* (Windows) atau *Terminal* (Linux) dan ketikkan **node index.js** seperti gambar di bawah ini.



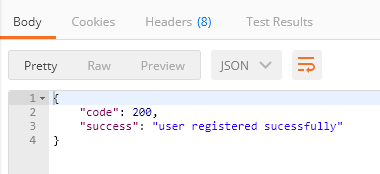
Selanjutnya dilakukan pengiriman HTTP *request* melalui Postman.

**Sign Up**

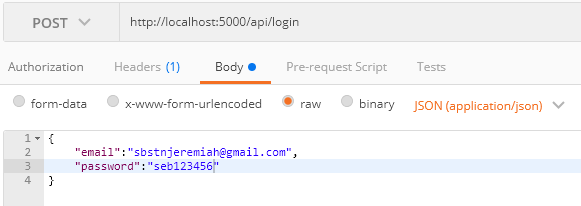


*User* perlu terlebih dahulu mendaftarkan diri melalui http://localhost:5000/api/register dengan mengirimkan HTTP POST *Request* yang berisikan *name, email,* dan *password* dalam format JSON seperti gambar di atas.

Jika berhasil *user* akan menerima *response* yang menunjukkan bahwa pendaftaran berhasil dilakukan dan informasi *user* sudah tersimpan di *database,* seperti di bawah ini.

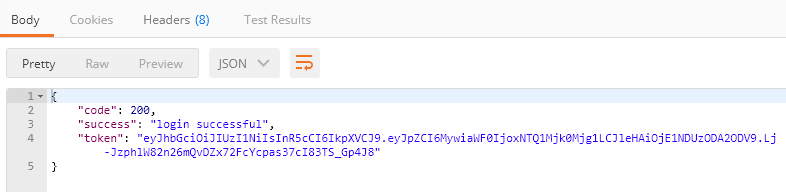


**Login**



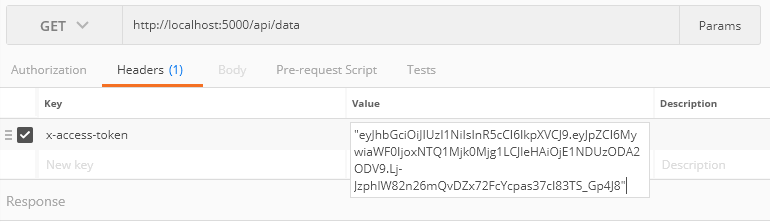
Setelah *user* terdaftar, *user* dapat melakukan *login* melalui http://localhost:5000/api/login dengan mengirimkan HTTP POST *Request* yang berisikan *name, email,* dan *password* dalam format JSON seperti gambar di atas.

Jika berhasil *user* akan menerima *response* yang berisi kode dan pesan yang menunjukkan *login* berhasil dilakukan serta *token* yang nantinya akan digunakan untuk setiap *request* berikutnya, seperti pada gambar di bawah ini.



**Get Data**

*Token* yang telah diterima sebelumnya akan disertakan dalam HTTP GET *Request* http://localhost:5000/api/data pada atribut x-access-token di *Header.* Contohnya seperti di bawah ini.

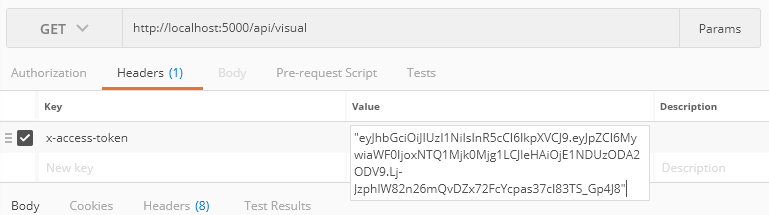


Jika *token* berhasil di-otentikasi, *request* akan diproses, dan jika berhasil akan memberikan *response* dataset dalam format JSON sebagai berikut.

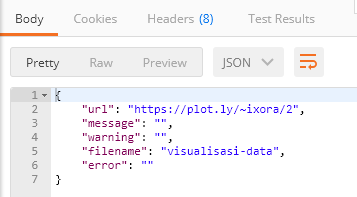


**Get Visualization**

Sama seperti **Get Data**, untuk melakukan HTTP GET *Request* pada http://localhost:5000/api/visual perlu disertakan *token* pada atribut x-access-token di *Header.* Contohnya seperti di bawah ini.



Jika berhasil, akan dikirimkan *response* dalam format JSON yang berisi URL hasil visualisasi, *message, warning, filename,* dan pesan *error* (jika terjadi *error*). Contohnya seperti berikut ini.



Untuk mendapatkan hasil visualisasi, akses URL yang diberikan pada *response* melalui *browser.*

Contoh hasil pada <https://plot.ly/~ixora/2> :

