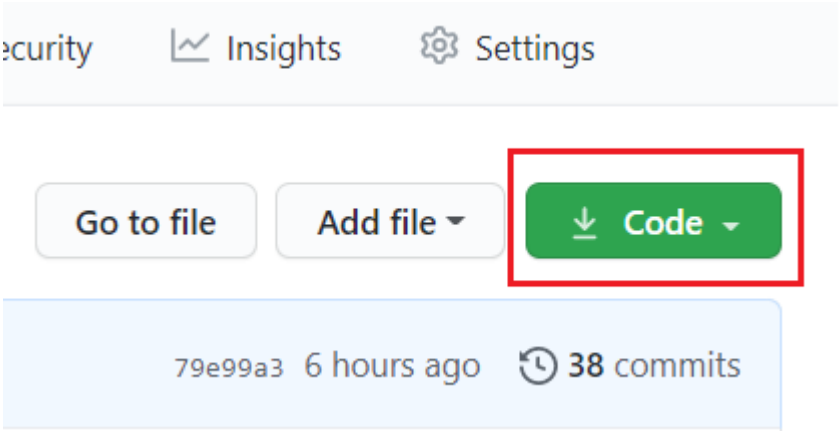
Table Of Content

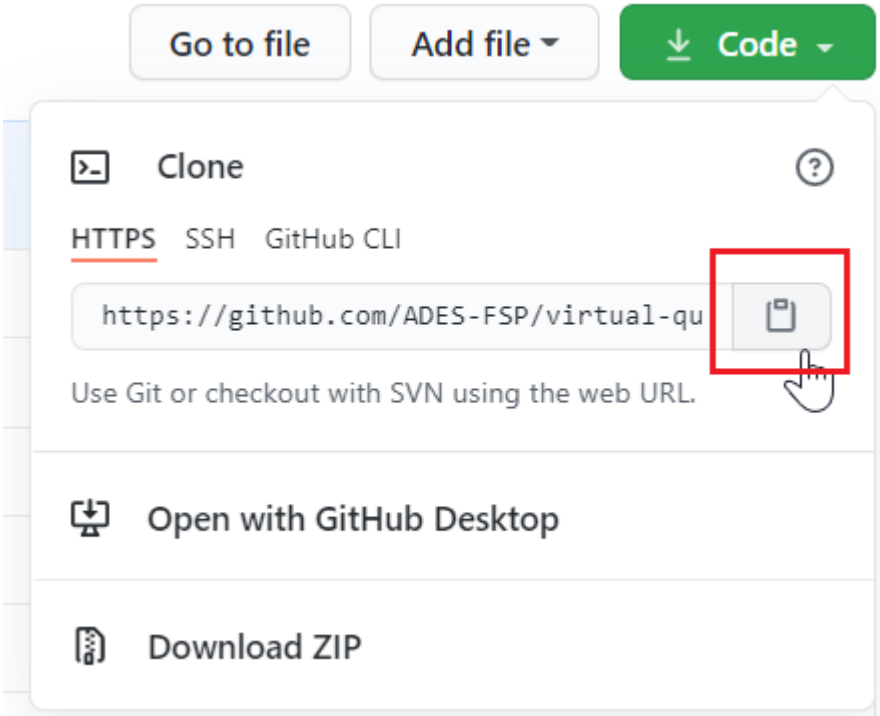
1 | Cloning Repository

1.1 | Cloning a repository using the command line

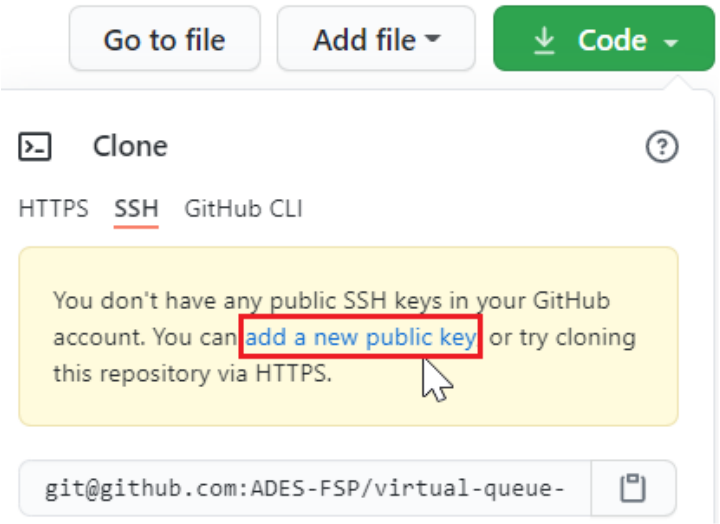
1. Navigate to the main page of the repository on GitHub.
2. Below the navigation bar, click download Code.



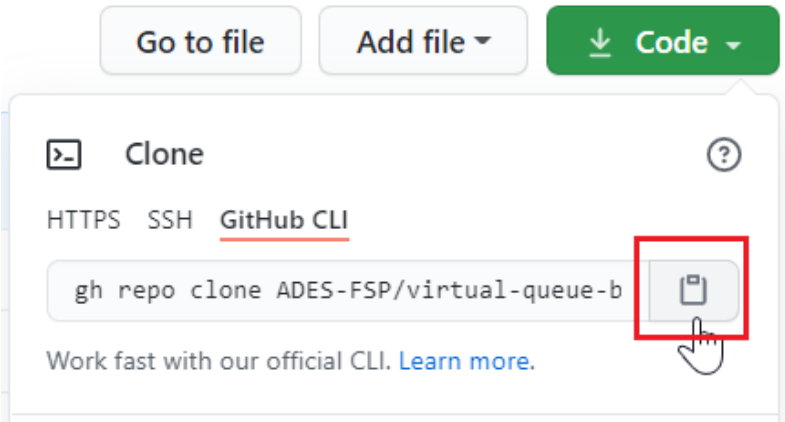
3. To clone the repository by using HTTPS, click under **HTTPS**.



To clone the repository by using an SSH key, click Use SSH, then click .
If you do not have any public SSH keys in your GitHub account. You can add a new public key by clicking on add a new public key.



To clone a repository by using GitHub CLI, click **GitHub CLI**, then click .



4. Open Git Bash.

5. On Git Bash, change the current directory to the location where you want to store the cloned directory.

6. Type `git clone`, then paste the URL that copied earlier.

   `$ git clone (URL)`

   An example is shown below:

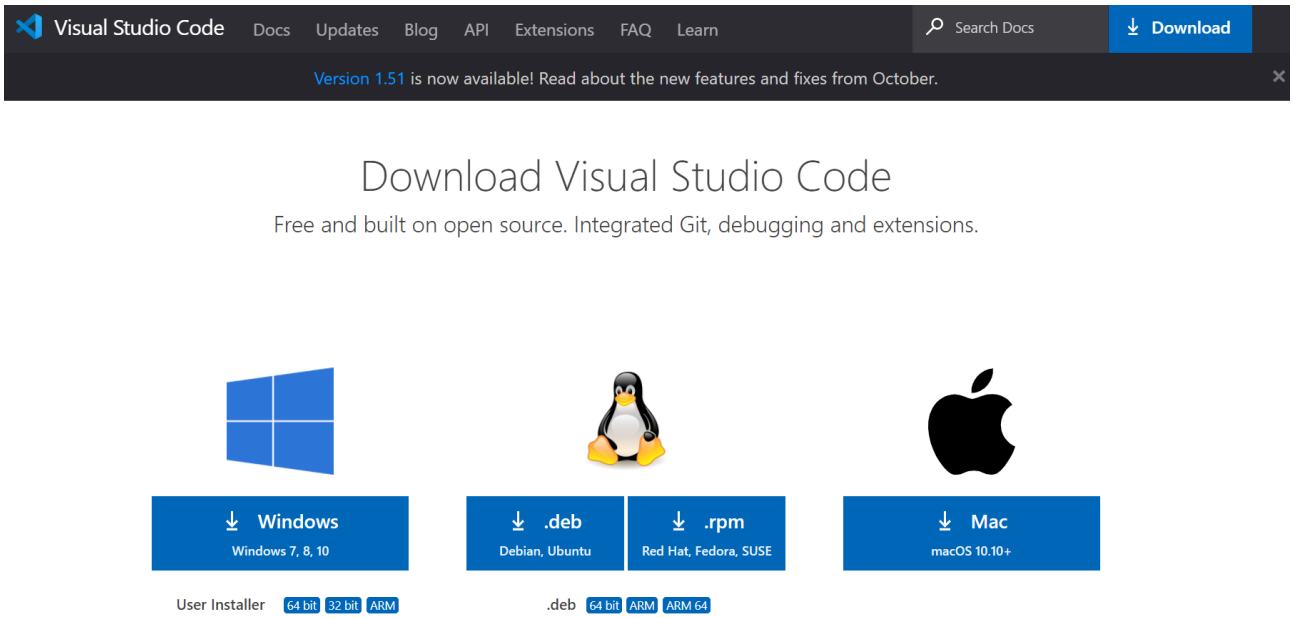
   `$ git clone gh repo clone ADES-FSP/virtual-queue-backend-2b03-adesca1`

7. Hit Enter to create a clone on your local host.

2 | Software Installation

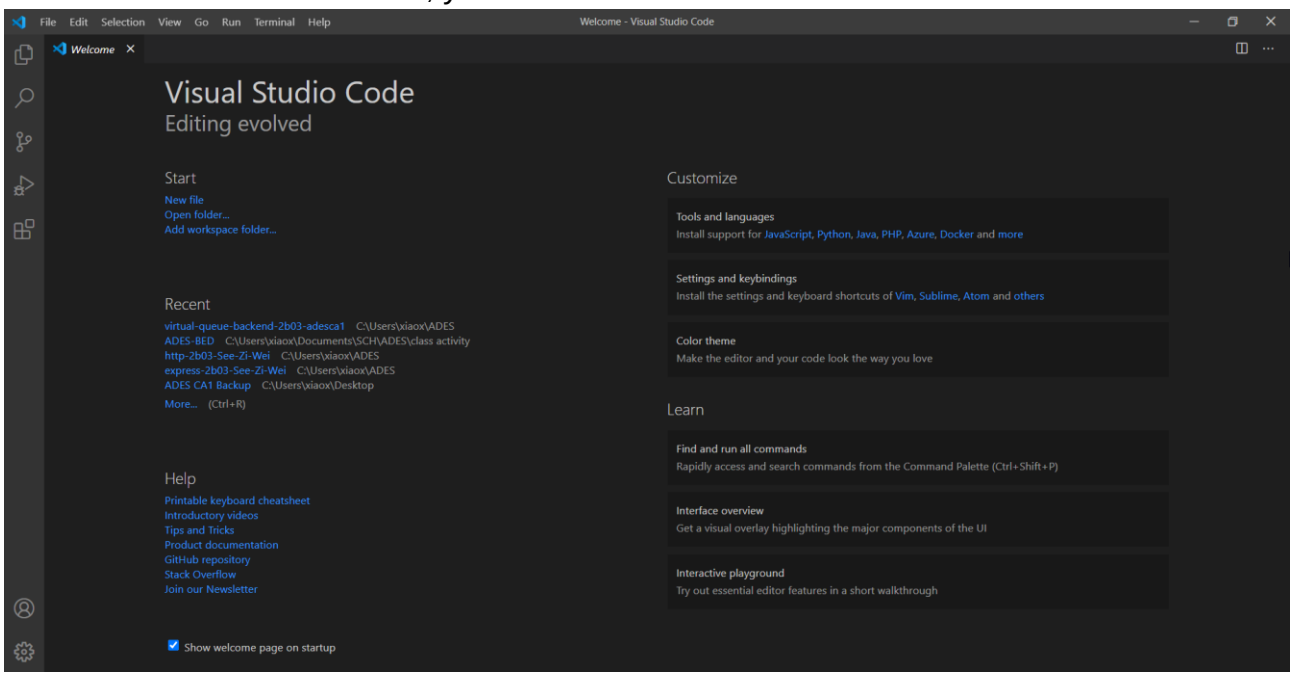
2.1 | Download and Install Visual Studio Code

1. Navigate to the website: https://code.visualstudio.com/download to download and install Visual Studio Code if you do not have it.



2. Once the exe file is downloaded, open the exe file to install it on your system.

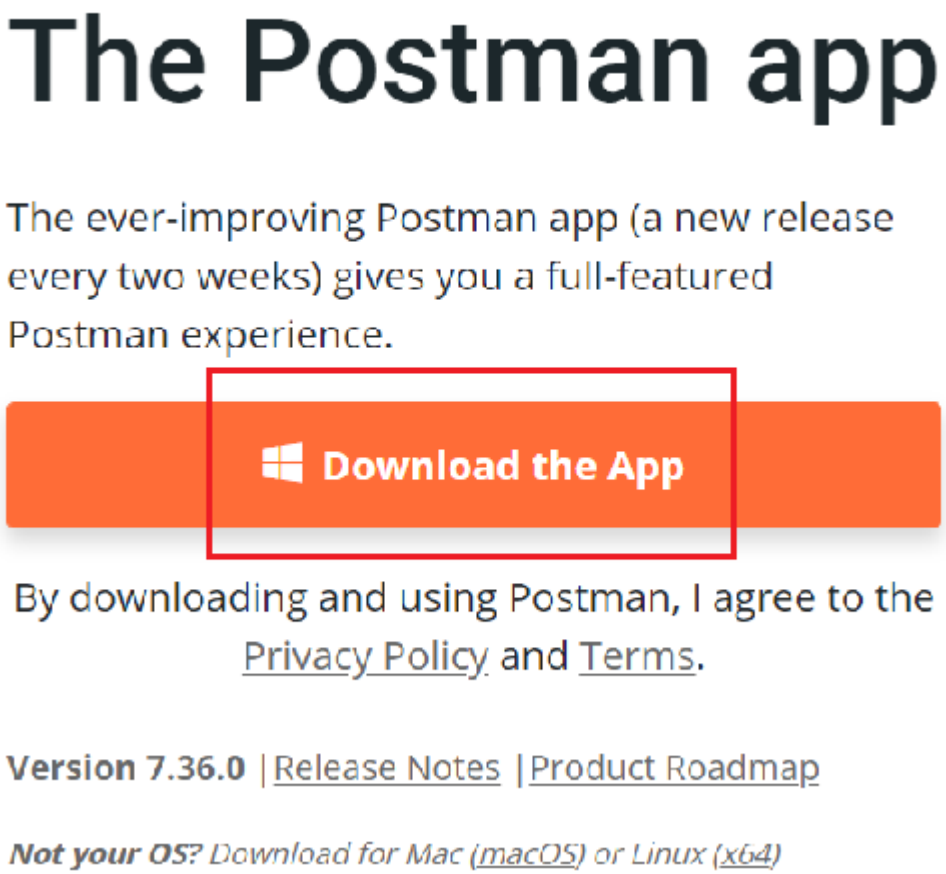
3. After successful installation, you will see the below Visual Studio Code toolkit.



2.2 | Download and Install Postman

We will use Postman to test if you have successfully set up the application. Below are the guides on how to download Postman.

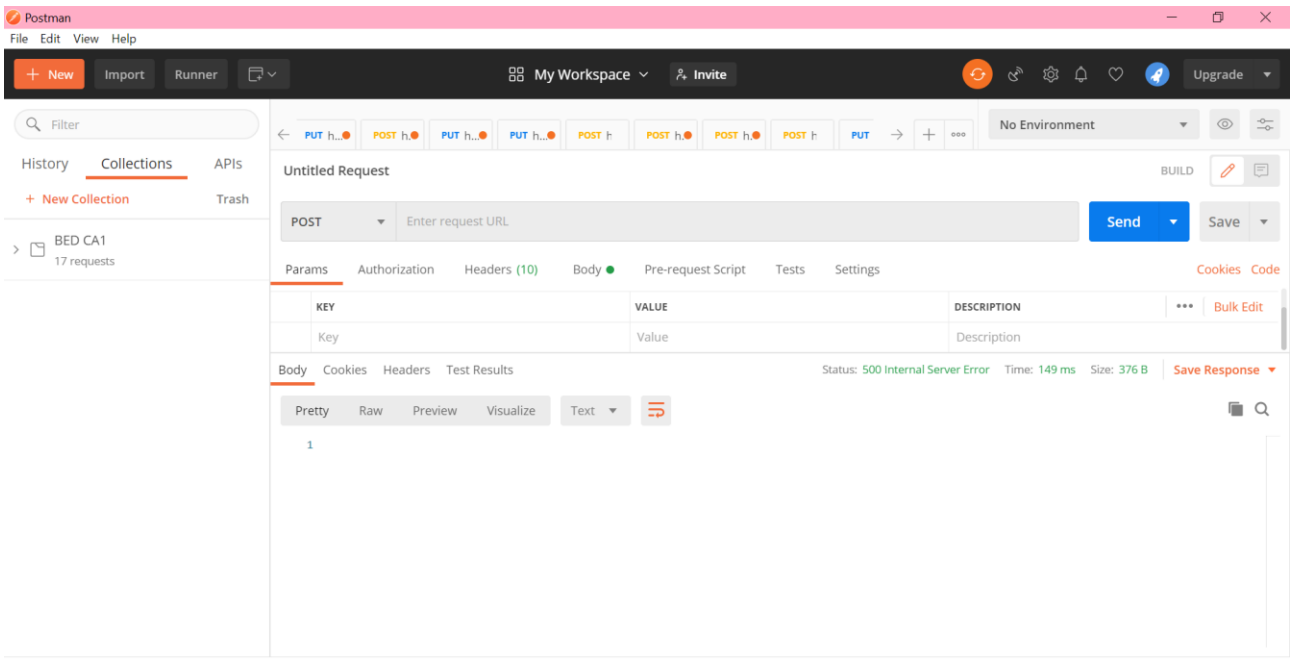
1. To install Postman, go to this link: https://www.postman.com/downloads/ and click Download the App.



2. Once the exe file is downloaded, open the exe file to install it on your system.

3. Once installation of Postman is completed, you will be asked to create an account. You can create an account on Postman now or create an account later by clicking on “Take me straight to the app, I’ll create an account another time” link.

4. After successful installation and registration, you will see the below Postman toolkit.

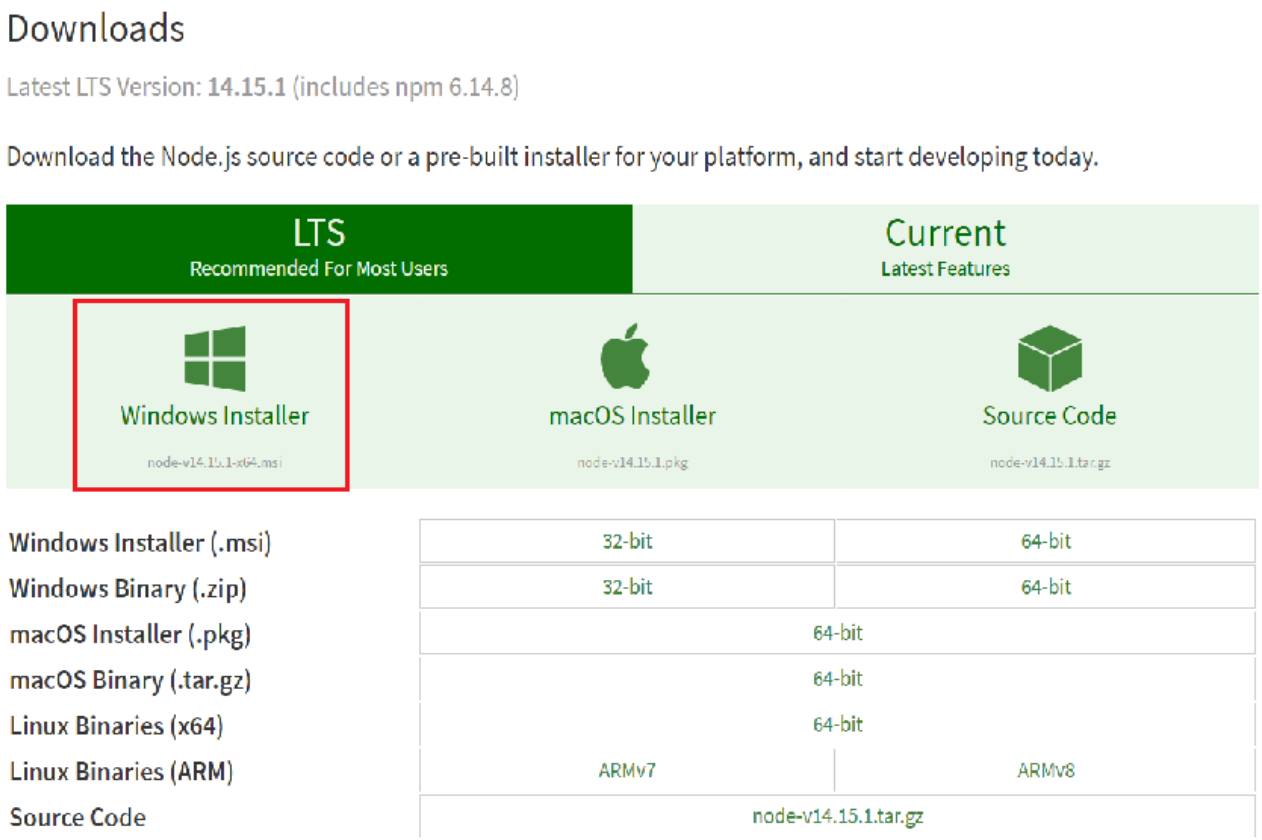

Postman
File Edit View Help
New
Import
Runner
My Workspace
Invite
Upgrade
Filter
History
Collections
APIs
New Collection
Trash
BED CA1
17 requests
No Environment
Untitled Request
BUILD
POST
Enter request URL
Send
Save
Params
Authorization
Headers (10)
Body
Pre-request Script
Tests
Settings
Cookies
Code
KEY
VALUE
DESCRIPTION
Bulk Edit
Key
Value
Description
Body
Cookies
Headers
Test Results
Status: 500 Internal Server Error
Time: 149 ms
Size: 376 B
Save Response
Pretty
Raw
Preview
Visualize
Text
1

2.3 | Install Node.js and NPM on Windows

2.3.1 | Download Node.js Installer

1. Navigate to https://nodejs.org/en/download/.

2. Click onto the Windows Installer button.
(Node.js installer includes the NPM package manager.)



2.3.2 | Install Node.js and NPM from browser

1. After the exe file of the windows installer is downloaded, double-click onto the exe file to launch it.

2. Click **Run** when the system asks if you want to run the software.

3. Click **Next** when the system shows welcome to the Node.js Setup Wizard.

4. The next screen will be reviewing the license agreement. Click **Next** if you agree to the terms.

5. When the system prompts you for the installation location, you can leave the default location or change it to the location that you want to install it. Then, click **Next**.

6. After that, the system will ask you to select the components to include or remove from the installation. You can accept the defaults by clicking **Next** unless you have a specific need.

7. Click **Install** to run the installer.

8. Click **Finish** when it finishes installing.

2.3.3 | Verify Installation

1. Open a command prompt, type: `node –v` and hit Enter.
The system should display the Node.js version installed on your system.

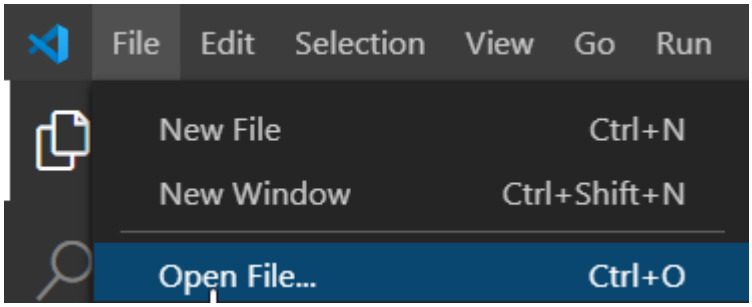
2. Type: `npm –v` and hit Enter.
The system should display the npm version installed on your system.

3 | Setup Command

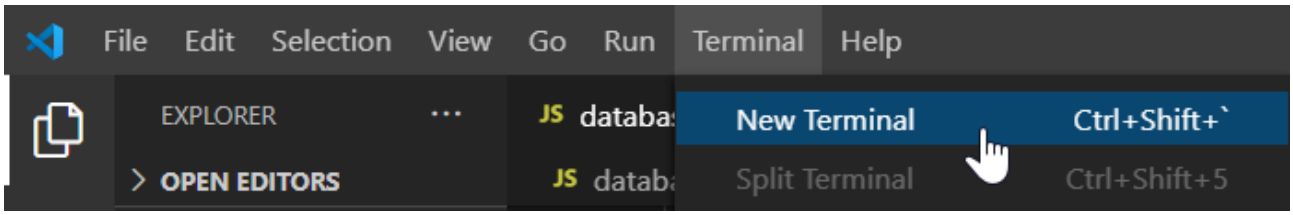
3.1 | Install jsonschema package

1. Open Visual Studio Code.

2. Under File, click on **Open File** to open the project you stored previously.



3. Under Terminal, click on **New Terminal**.



4. Type npm install jsonschema and hit Enter.

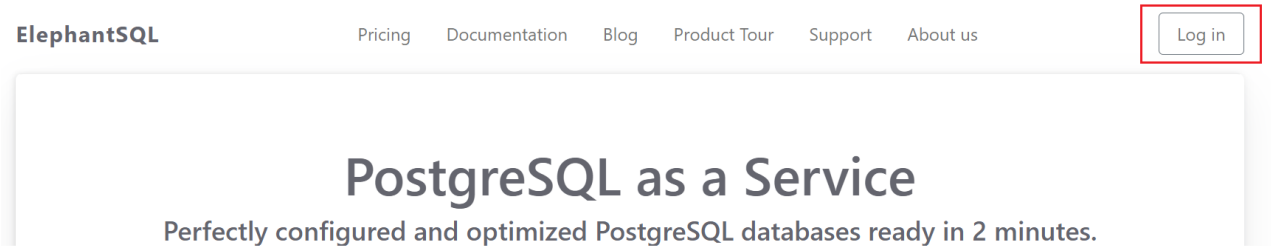
4 | Setup Database

4.1 | Setup Database on ElephantSQL

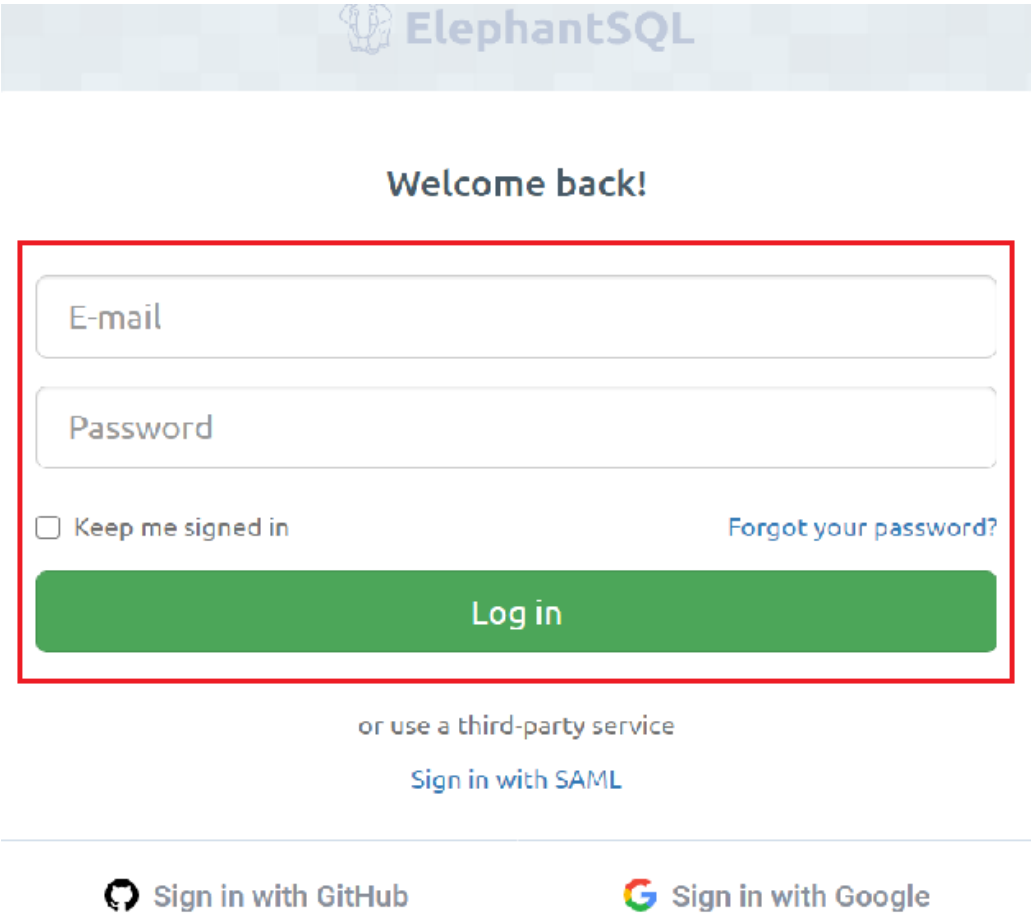
4.1.1 | Create a new instance

1. Navigate to the website: https://www.elephantsql.com/

2. Click on the Login button.

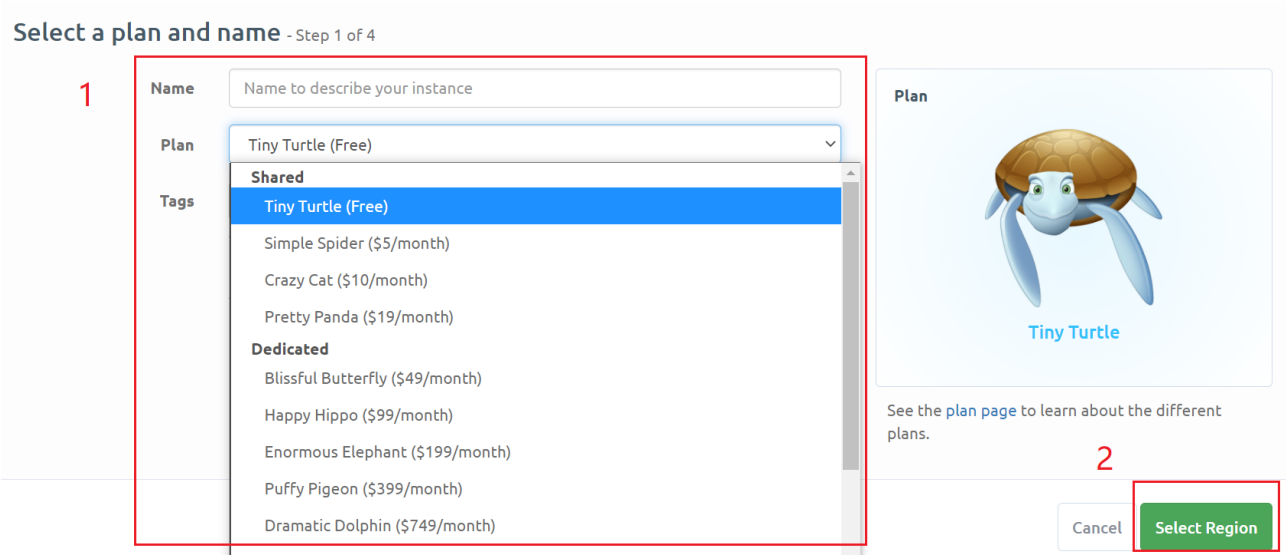


3. Key in your account credential to login or click the button **Sign in with GitHub/ Sign in with Google**.

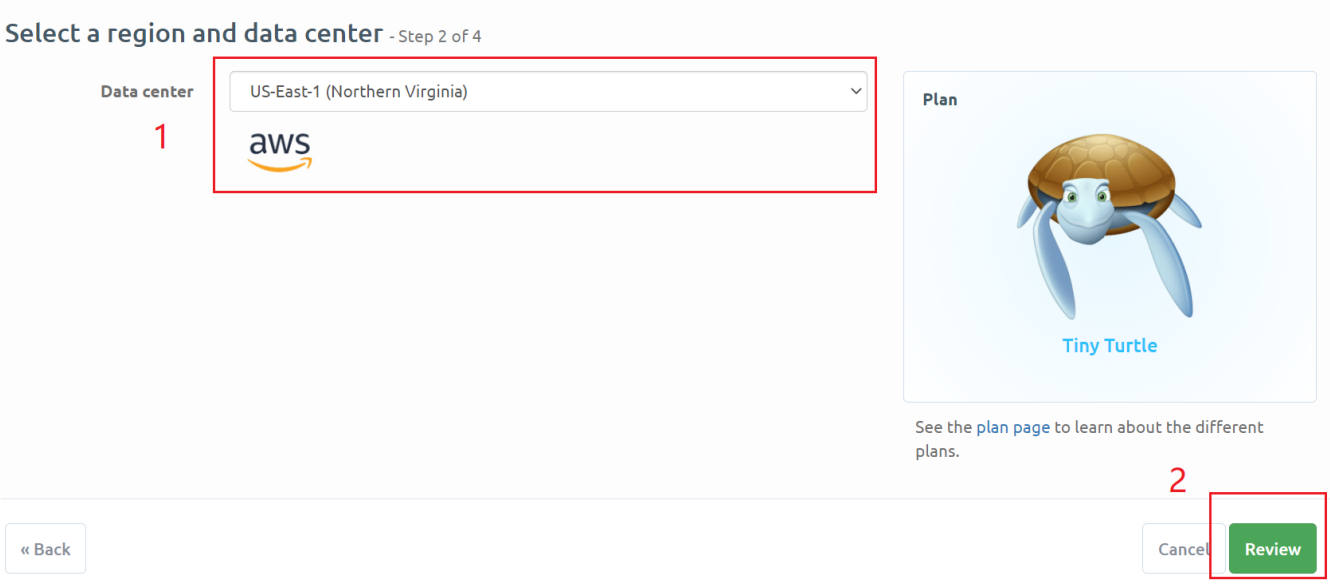


4. Click + Create New Instance on top right of the page.

5. Fill in the **Name** for your instance and select a **Plan**. You can use the **Tiny Turtle** plan because it's free, but you can also choose the other plan. Then, click on the Select Region button.



7. Select a region and data centre for your instance and click Review.



8. Once all the information is filled out, click Create instance on the bottom right of the page.

Confirm new instance - Step 4 of 4

Plan

Tiny Turtle

Total: Free

Name:	2
Provider:	Amazon Web Services
Region:	US-East-1 (Northern Virginia)

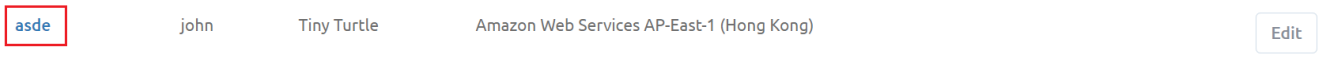
« Back

Cancel

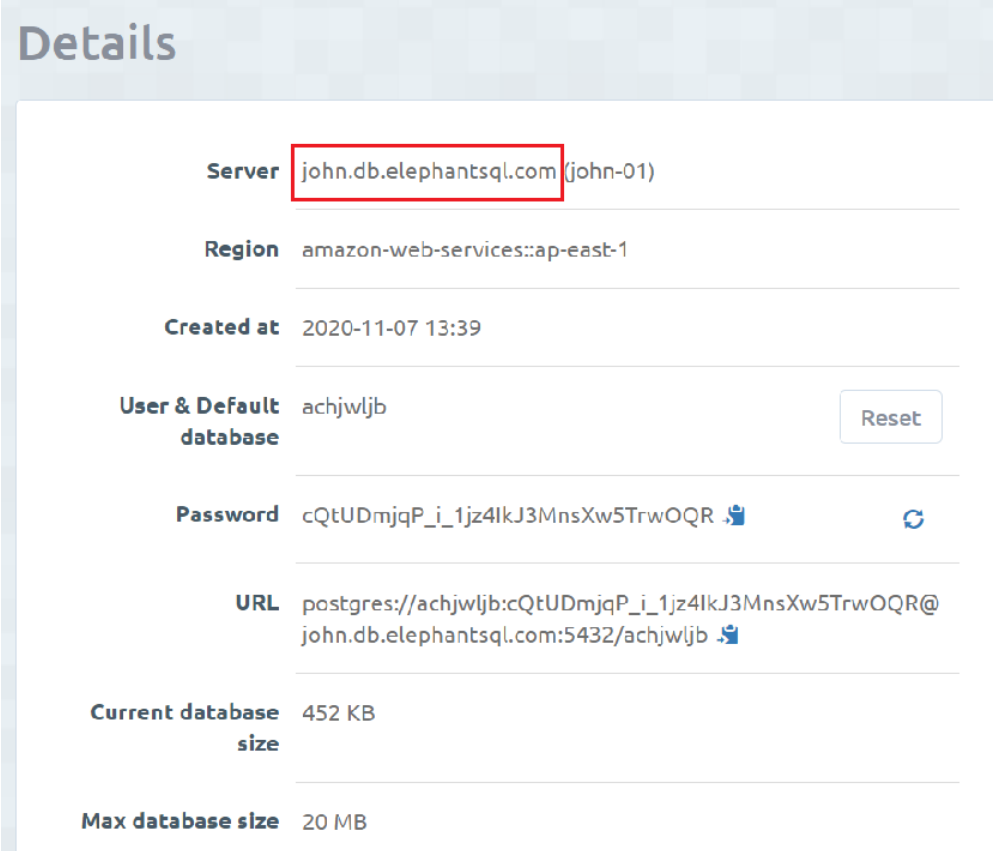
Create instance ✔

4.1.2 | Connect application to database

1. Click on the instance you just created to view its details. For example:

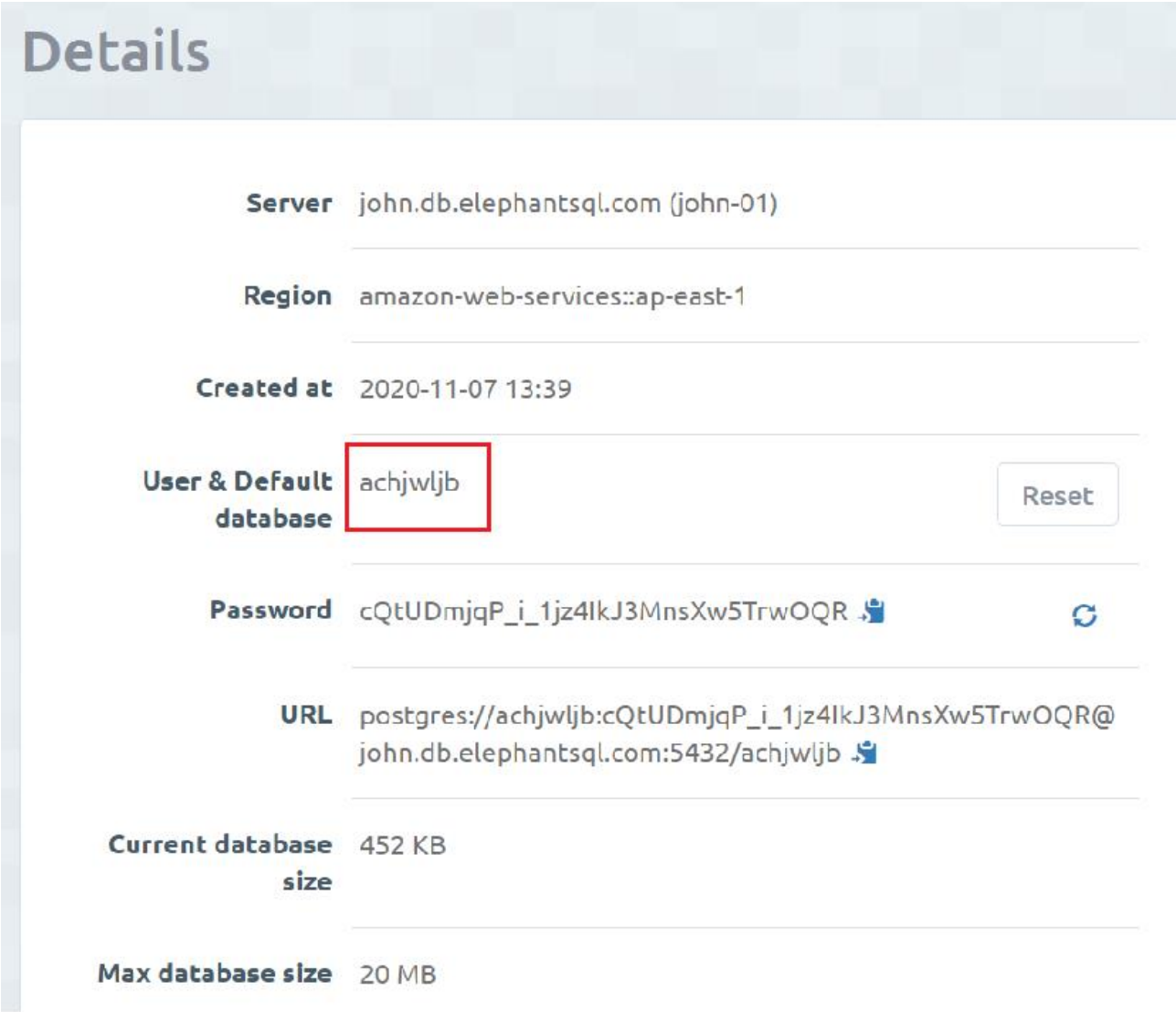


2. Copy the information on **Server.**

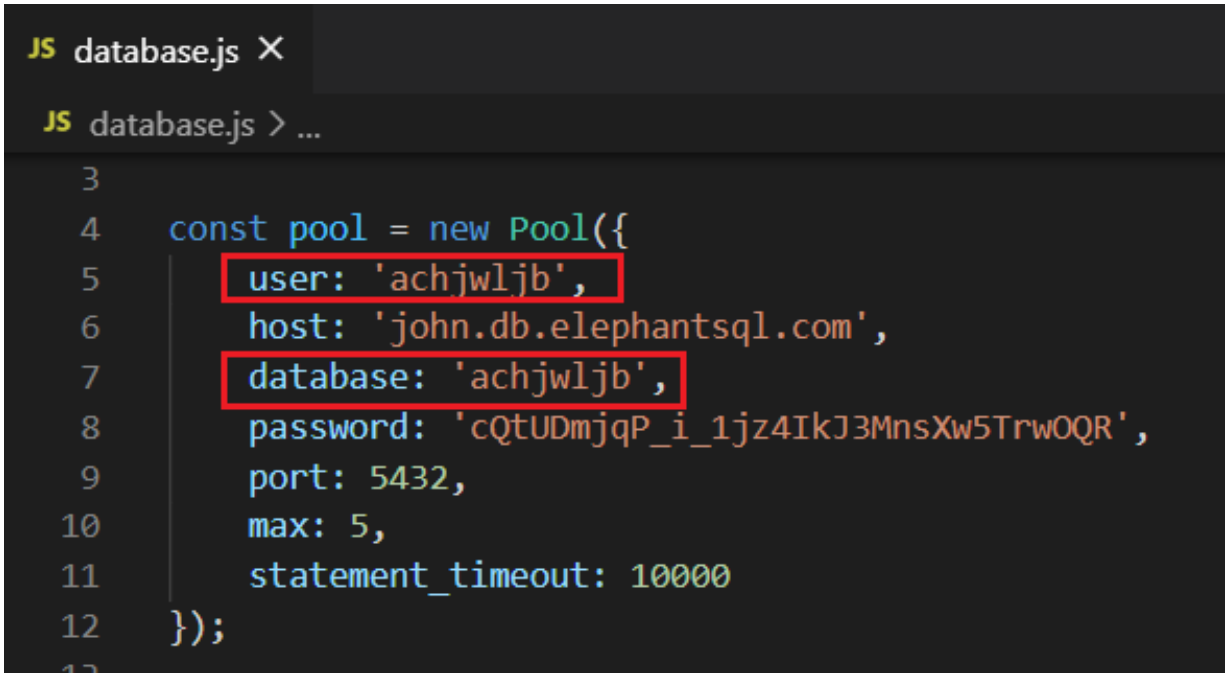


3. Go to database.js, paste the information copied on the **host** section.

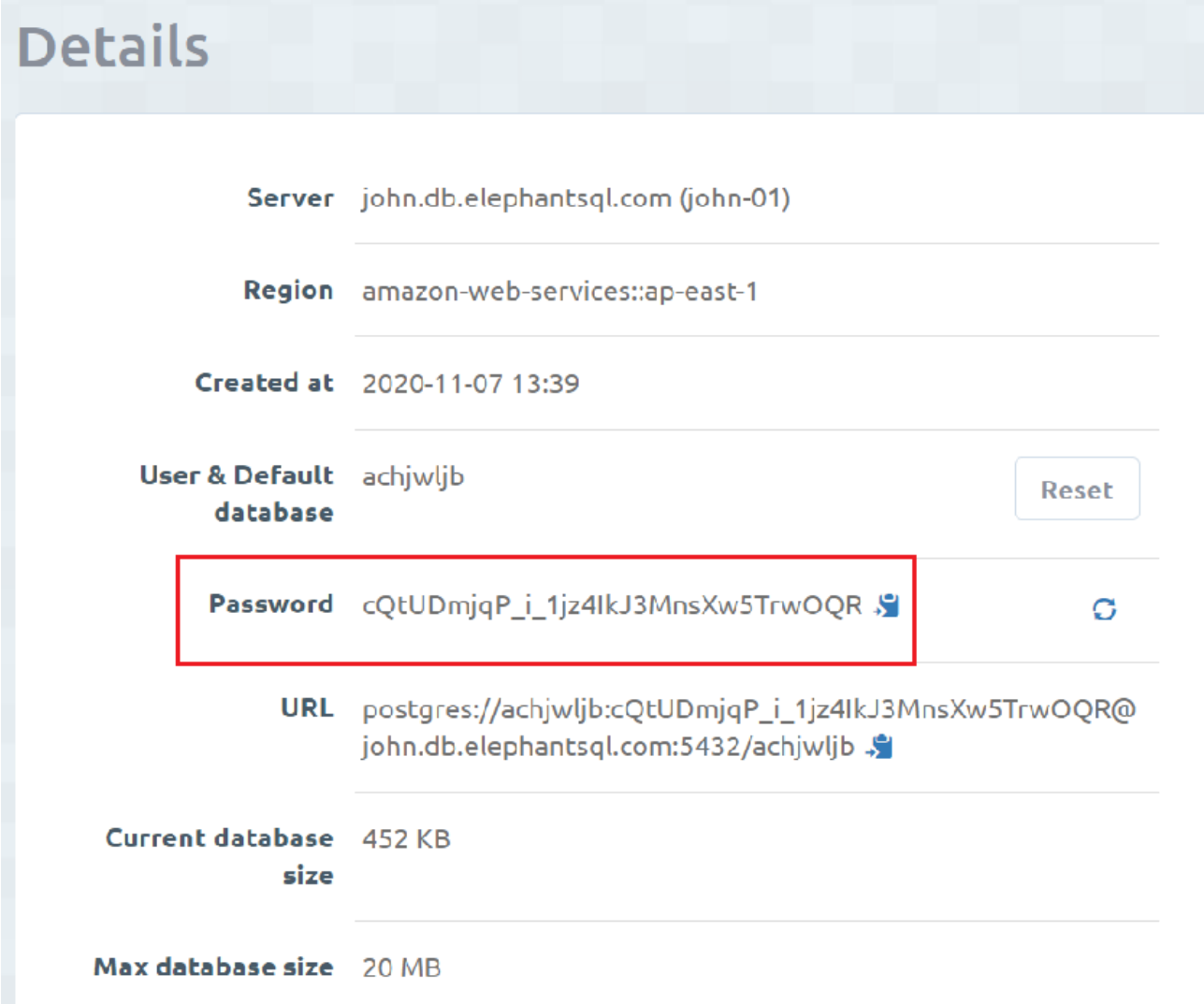
4. Copy the **User & Default database.**



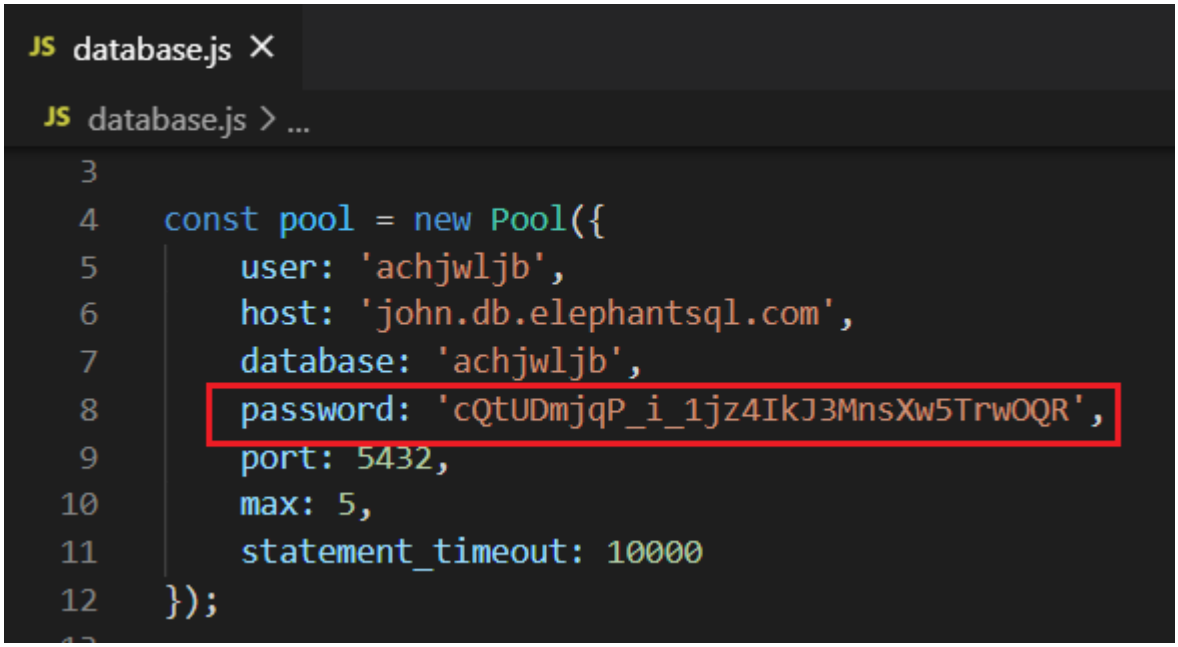
5. Go to database.js, paste the information copied on **user** and **database**.



6. Copy the **Password.** You will need to click on the little eye icon to expand the full URL.



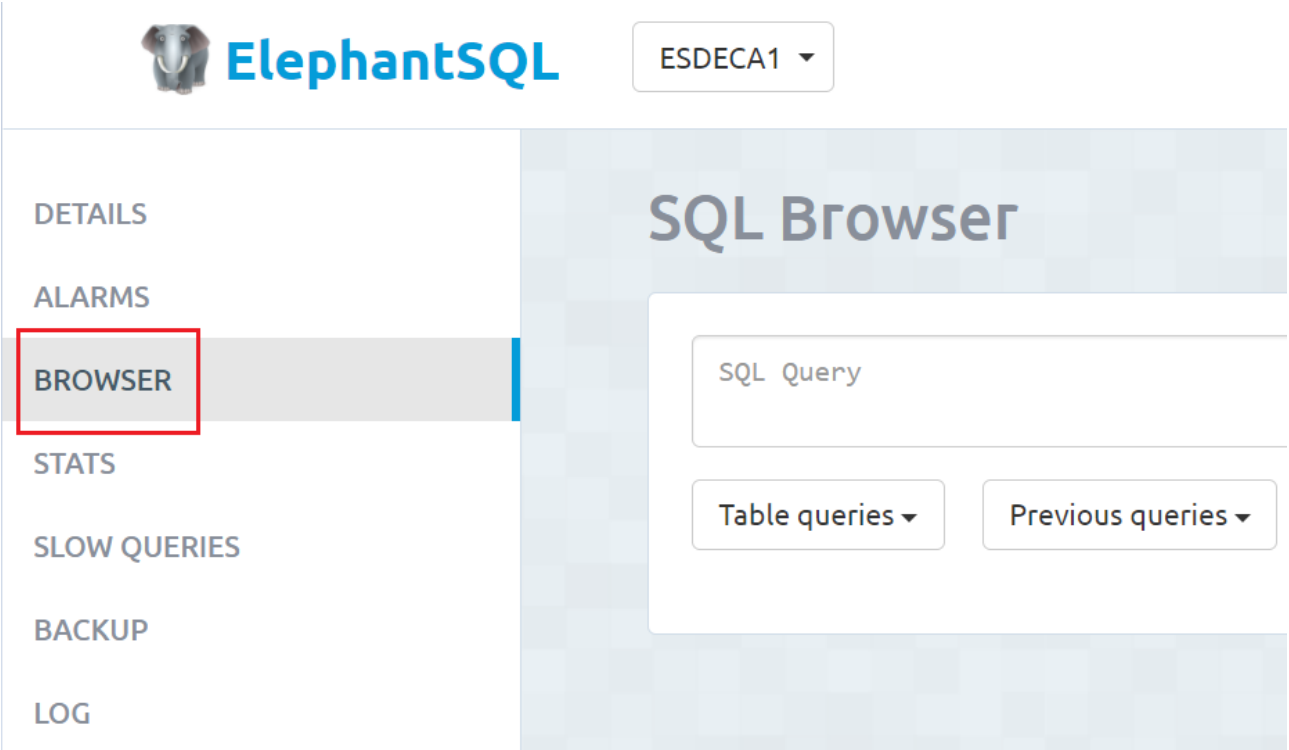
7. Go to database.js, paste the information copied on **password**.



```
JS database.js ×
JS database.js > ...
3
4   const pool = new Pool({
5       user: 'achjwljb',
6       host: 'john.db.elephantsql.com',
7       database: 'achjwljb',
8       password: 'cQtUDmjqP_i_1jz4IkJ3MnsXw5TrwOQR',
9       port: 5432,
10      max: 5,
11      statement_timeout: 10000
12  });
```

4.1.3 | Create table on database created on ElephantSQL

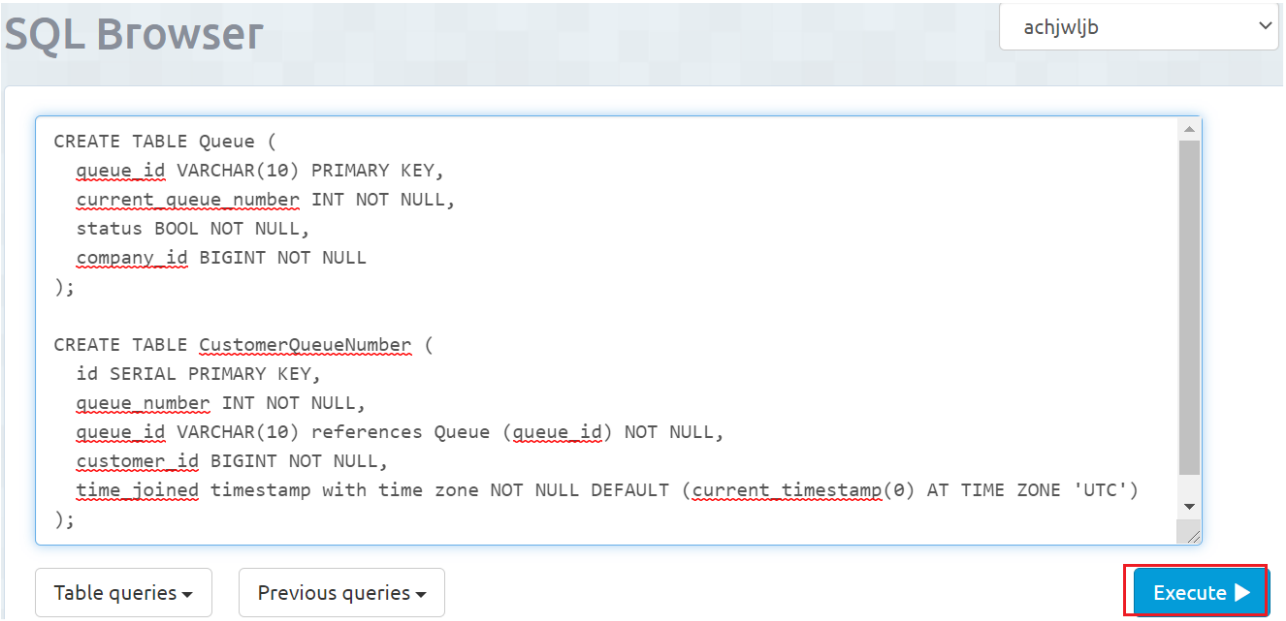
1. Go to the browser on ElephantSQL.



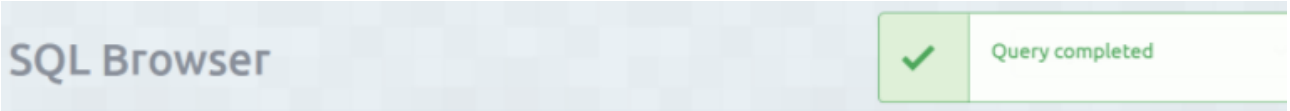
2. Copy the SQL statements below.

```
CREATE TABLE Queue (
  queue_id VARCHAR(10) PRIMARY KEY,
  current_queue_number INT NOT NULL,
  status BOOL NOT NULL,
  company_id BIGINT NOT NULL
);

CREATE TABLE CustomerQueueNumber (
  id SERIAL PRIMARY KEY,
  queue_number INT NOT NULL,
  queue_id VARCHAR(10) references Queue (queue_id) NOT NULL,
  customer_id BIGINT NOT NULL,
  time_joined timestamp with time zone NOT NULL DEFAULT (current_timestamp(0) AT
TIME ZONE 'UTC')
);
```

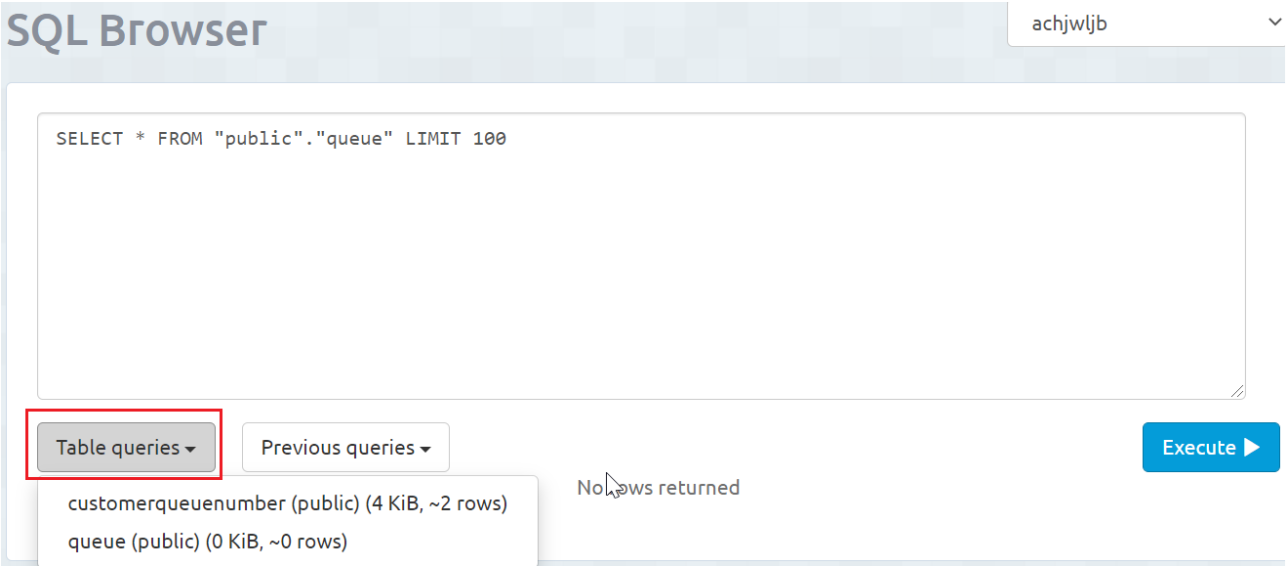
3. Paste the SQL statements in SQL Browser and hit Execute



4. A prompt message: Query completed indicates that you successfully launch the sql statement.



5. Click on Table queries and tables called customerqueuenumber and queue should be created.



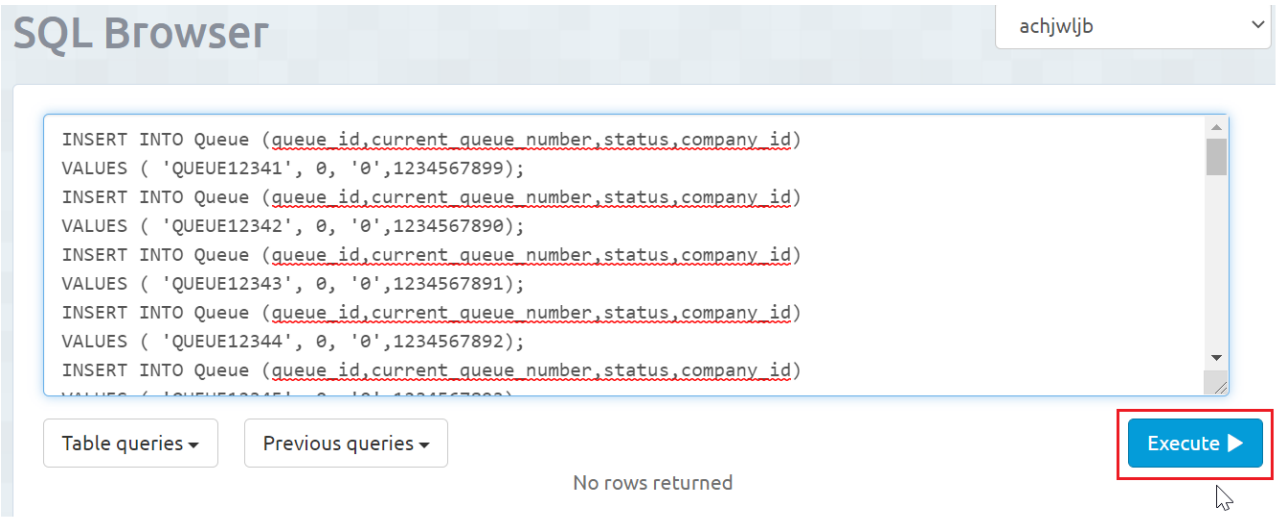
4.1.4 | Insert records into tables

1. Copy the sample SQL statements below. (you can create your own records and insert.)

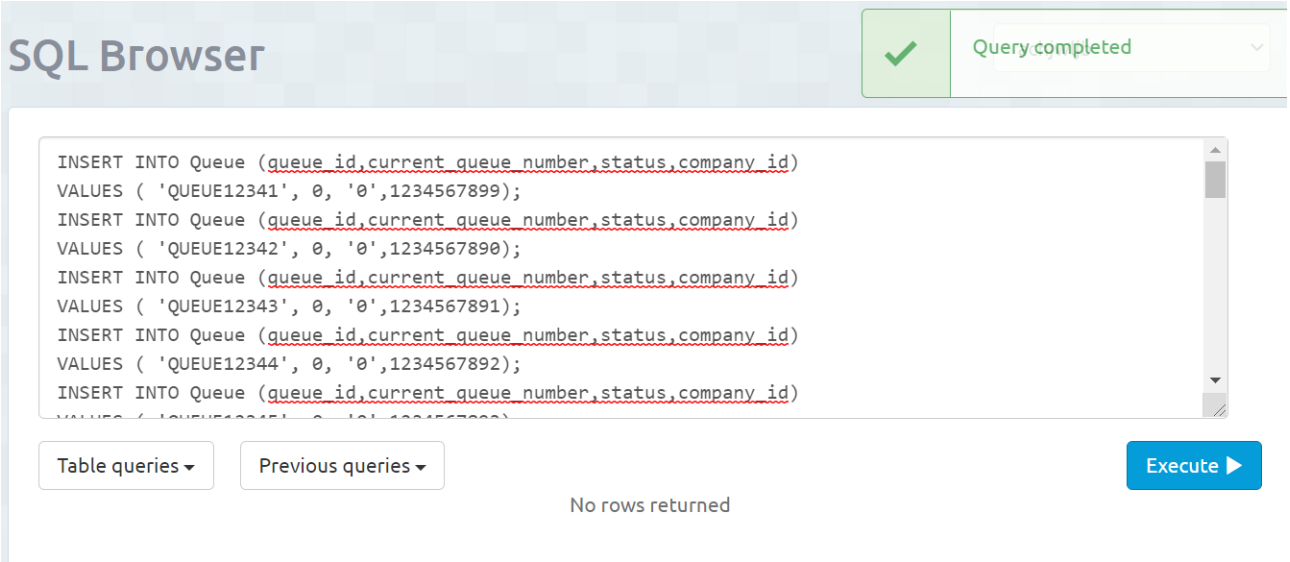
```
INSERT INTO Queue (queue_id,current_queue_number,status,company_id)
VALUES ( 'QUEUE12341', 0, '0',1234567899);
INSERT INTO Queue (queue_id,current_queue_number,status,company_id)
VALUES ( 'QUEUE12342', 0, '0',1234567890);
INSERT INTO Queue (queue_id,current_queue_number,status,company_id)
VALUES ( 'QUEUE12343', 0, '0',1234567891);
INSERT INTO Queue (queue_id,current_queue_number,status,company_id)
VALUES ( 'QUEUE12344', 0, '0',1234567892);
INSERT INTO Queue (queue_id,current_queue_number,status,company_id)
VALUES ( 'QUEUE12345', 0, '0',1234567893);
INSERT INTO Queue (queue_id,current_queue_number,status,company_id)
VALUES ( 'QUEUE12346', 0, '0',1234567894);
INSERT INTO Queue (queue_id,current_queue_number,status,company_id)
VALUES ( 'QUEUE12347', 0, '0',1234567895);
INSERT INTO Queue (queue_id,current_queue_number,status,company_id)
VALUES ( 'QUEUE12348', 0, '0',1234567896);

INSERT INTO CustomerQueueNumber (queue_number,queue_id,customer_id)
VALUES ( 1, 'QUEUE12341', 1134567899);
INSERT INTO CustomerQueueNumber (queue_number,queue_id,customer_id)
VALUES ( 1, 'QUEUE12342', 1234567899);
INSERT INTO CustomerQueueNumber (queue_number,queue_id,customer_id)
VALUES ( 1, 'QUEUE12343', 1334567899);
INSERT INTO CustomerQueueNumber (queue_number,queue_id,customer_id)
VALUES ( 2, 'QUEUE12343', 1434567899);
INSERT INTO CustomerQueueNumber (queue_number,queue_id,customer_id)
VALUES (3, 'QUEUE12343', 1534567899);
INSERT INTO CustomerQueueNumber (queue_number,queue_id,customer_id)
VALUES ( 2, 'QUEUE12341', 1634567899);
INSERT INTO CustomerQueueNumber (queue_number,queue_id,customer_id)
VALUES ( 3, 'QUEUE12341', 1734567899);
INSERT INTO CustomerQueueNumber (queue_number,queue_id,customer_id)
VALUES ( 2, 'QUEUE12342', 1834567899);
INSERT INTO CustomerQueueNumber (queue_number,queue_id,customer_id)
VALUES ( 4, 'QUEUE12343', 1934567899);
INSERT INTO CustomerQueueNumber (queue_number,queue_id,customer_id)
VALUES ( 5, 'QUEUE12343', 1114567899);
INSERT INTO CustomerQueueNumber (queue_number,queue_id,customer_id)
VALUES ( 1, 'QUEUE12344', 1124567899);
INSERT INTO CustomerQueueNumber (queue_number,queue_id,customer_id)
VALUES ( 1, 'QUEUE12345', 1134567899);
```

```
INSERT INTO CustomerQueueNumber (queue_number,queue_id,customer_id)
VALUES ( 2, 'QUEUE12344', 1144567899);
INSERT INTO CustomerQueueNumber (queue_number,queue_id,customer_id)
VALUES ( 3, 'QUEUE12344', 1154567899);
INSERT INTO CustomerQueueNumber (queue_number,queue_id,customer_id)
VALUES ( 2, 'QUEUE12345', 1164567899);
INSERT INTO CustomerQueueNumber (queue_number,queue_id,customer_id)
VALUES ( 1, 'QUEUE12346', 1174567899);
INSERT INTO CustomerQueueNumber (queue_number,queue_id,customer_id)
VALUES ( 2, 'QUEUE12346', 1184567899);
INSERT INTO CustomerQueueNumber (queue_number,queue_id,customer_id)
VALUES ( 3, 'QUEUE12346', 1194567899);
```

2. Paste the SQL statements into SQL Browser and hit Execute.

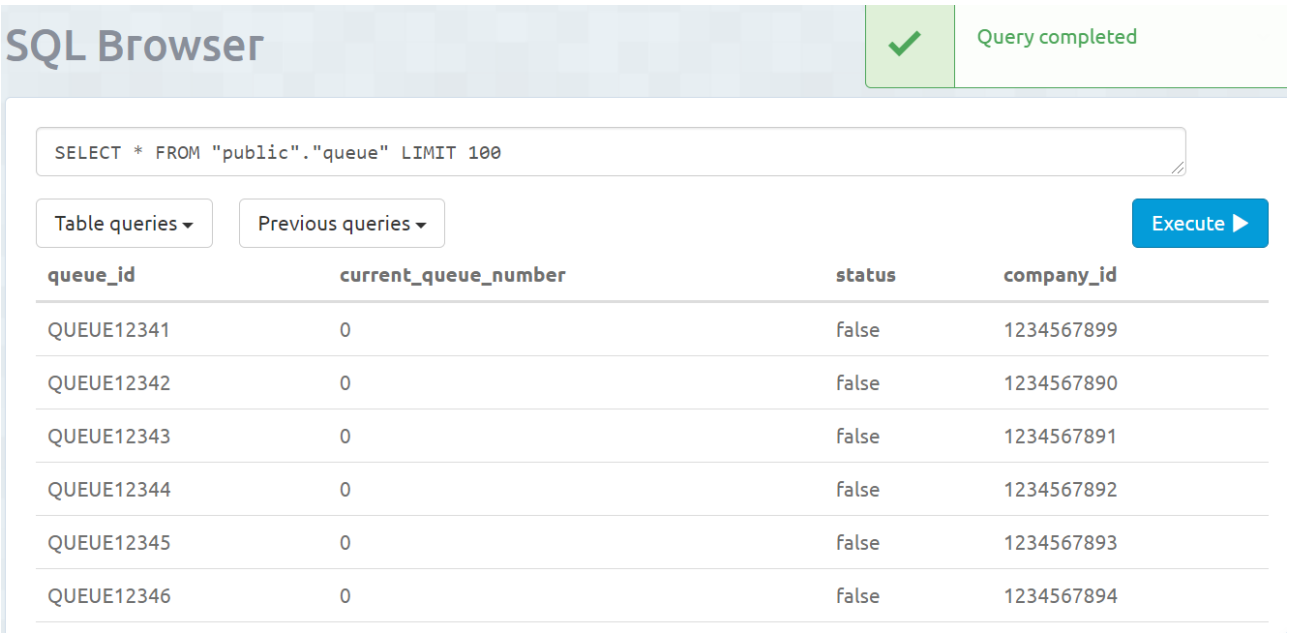


3. A prompt message: Query completed indicates that you successfully launch the sql statement.



4. Click on Table queries and click on one of the tables created called queue. Then, hit Execute.

5. You should see the records created in the table.

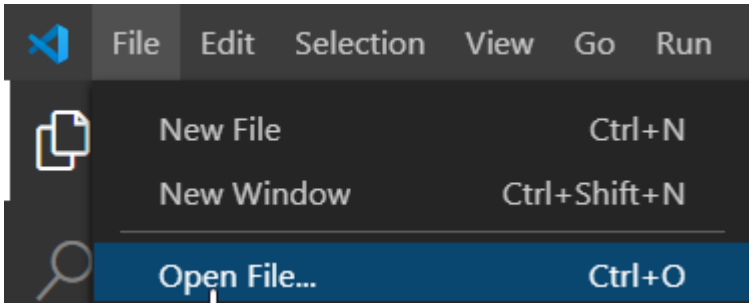


5 | Start the Application

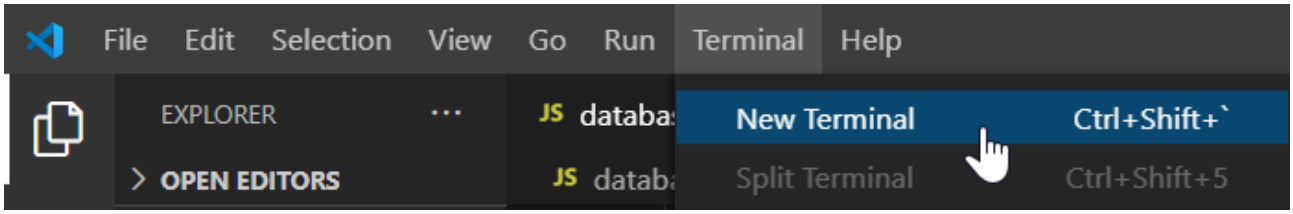
5.1 | node server.js

1. Open Visual Studio Code.

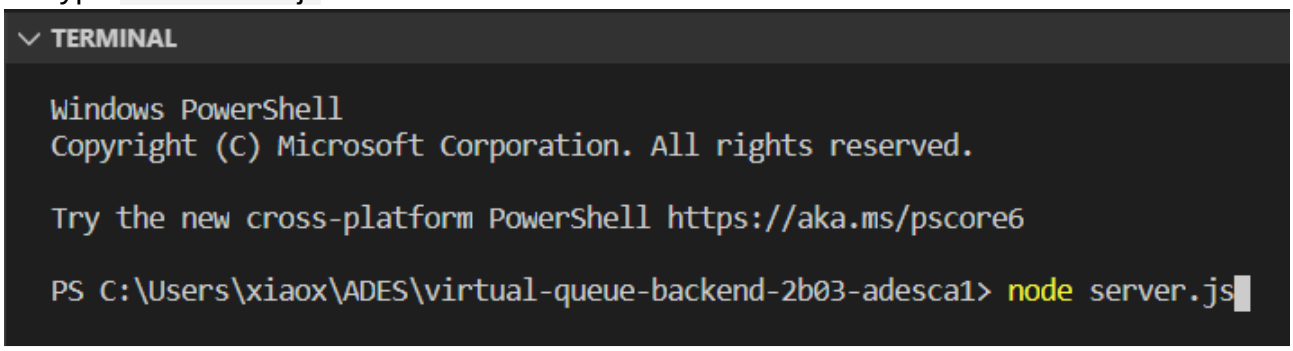
2. Under File, click on **Open File** to open the project you stored previously.



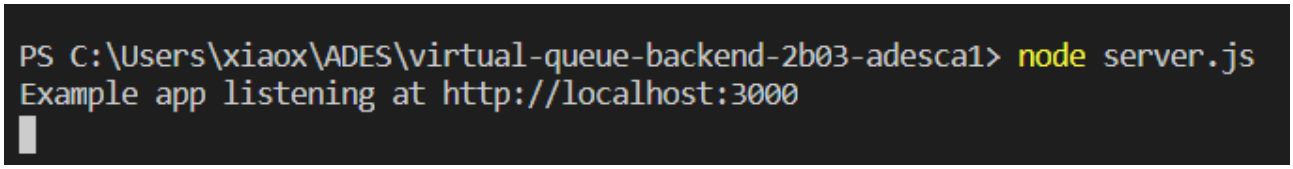
3. Under Terminal, click on **New Terminal**.



4. Type node server.js and hit Enter.



5. You will see a url like example below, it means you have successfully started the app.



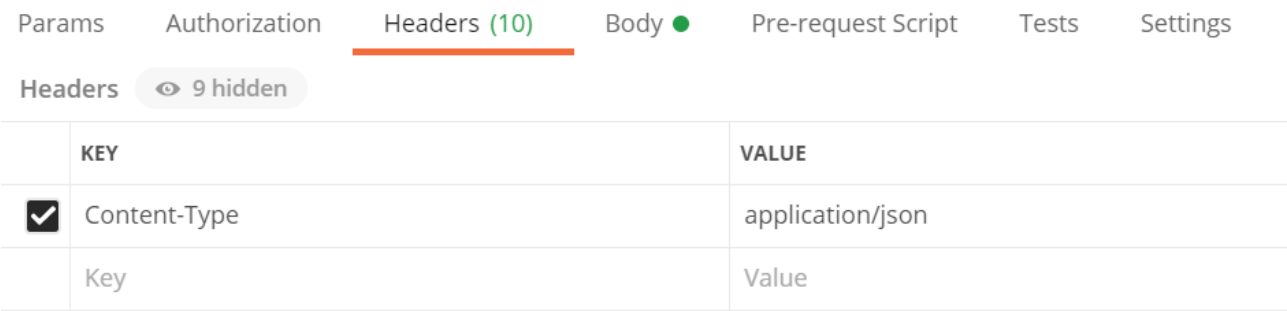
6 | Testing the Project

6.1 | Test using Postman

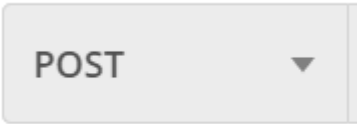
We will test one of the APIs which is Create Queue API. A successful request made to the server should create a new entry in the queue id with the correct parameters.

1. Check your database, tables should be created with data inserted. (Can refer to 4.1.3 & 4.1.4) Queue Id (“QUEUE33333”) does not already exist in the database.

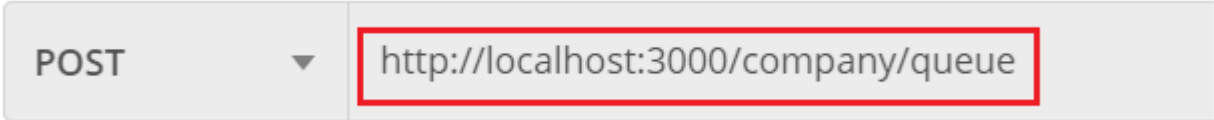
2. Go to Postman, under Headers, set Content-Type to application/json.



3. Choose **POST** for the method.

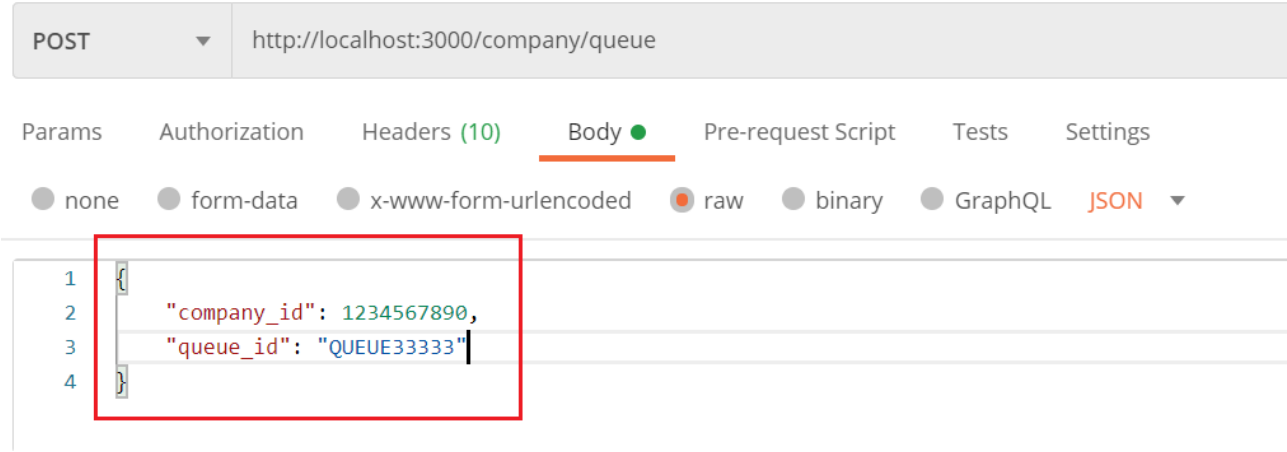


4. Enter request url: http://localhost:3000/company/queue.

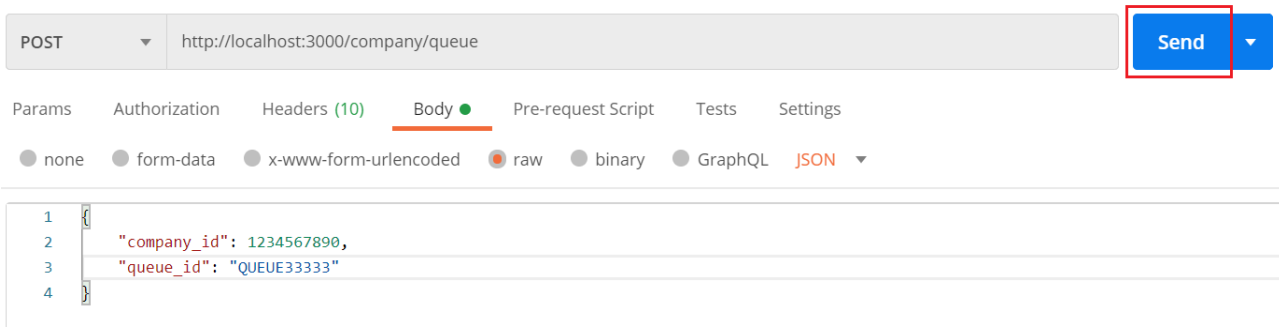


5. Under **Body**, key in information below:

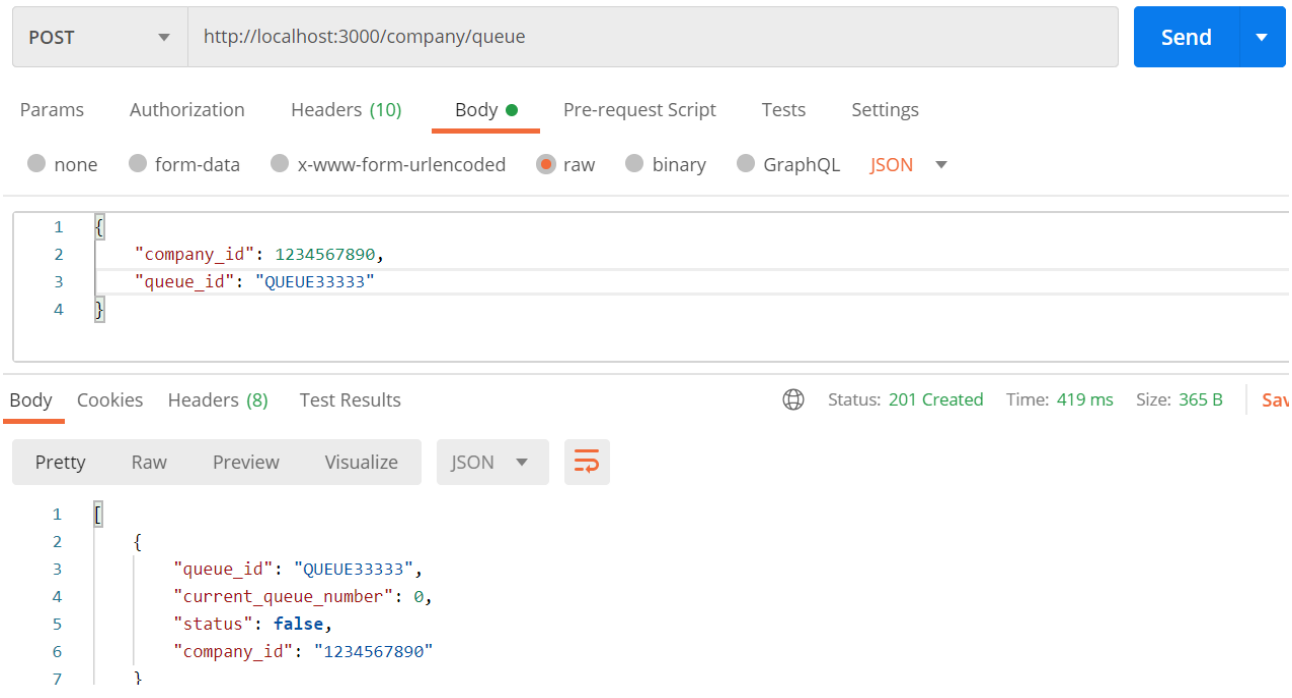
```
{
“company_id”: 1234567890,
“queue_id”: “QUEUE33333”
}
```



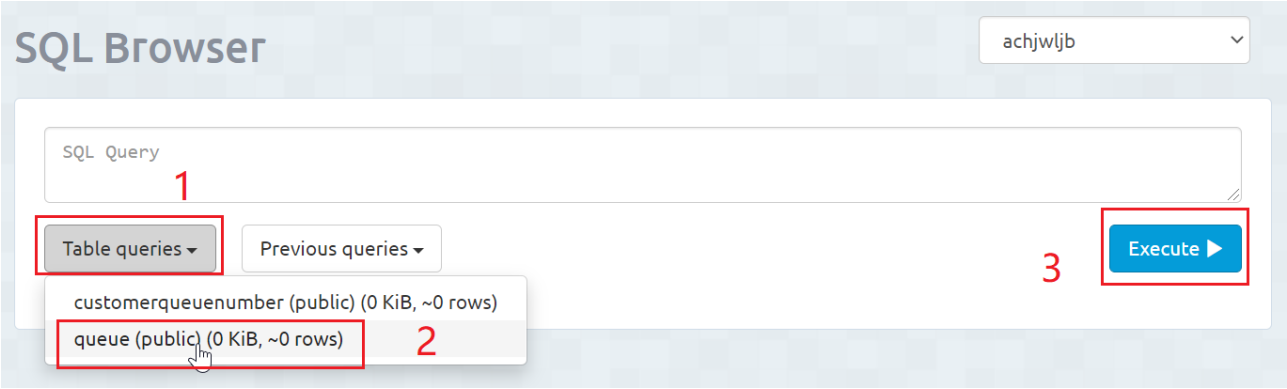
6. Then, Hit Send.



7. You will receive a 201 Created response from Postman.



8. Go to SQL Browser on ElephantSQL, Click on Table queries. Then, click on one of the tables (I will click on the table queue in this example). After that, hits Execute.



9. You should see a prompt message: Query completed indicates successfully executed the sql statement. You should also see that queue_id(QUEUE33333) is created. It means the Create Queue API works. (which means you have successfully set up the app!)