Create a Power BI Dashboard with the Stack Overflow API

Connect the Stack Overflow API and Power BI Dashboard to create charts and reports from your Teams data.

Document generated 12/06/2024

PDF VERSION

Tags | **Reporting** |

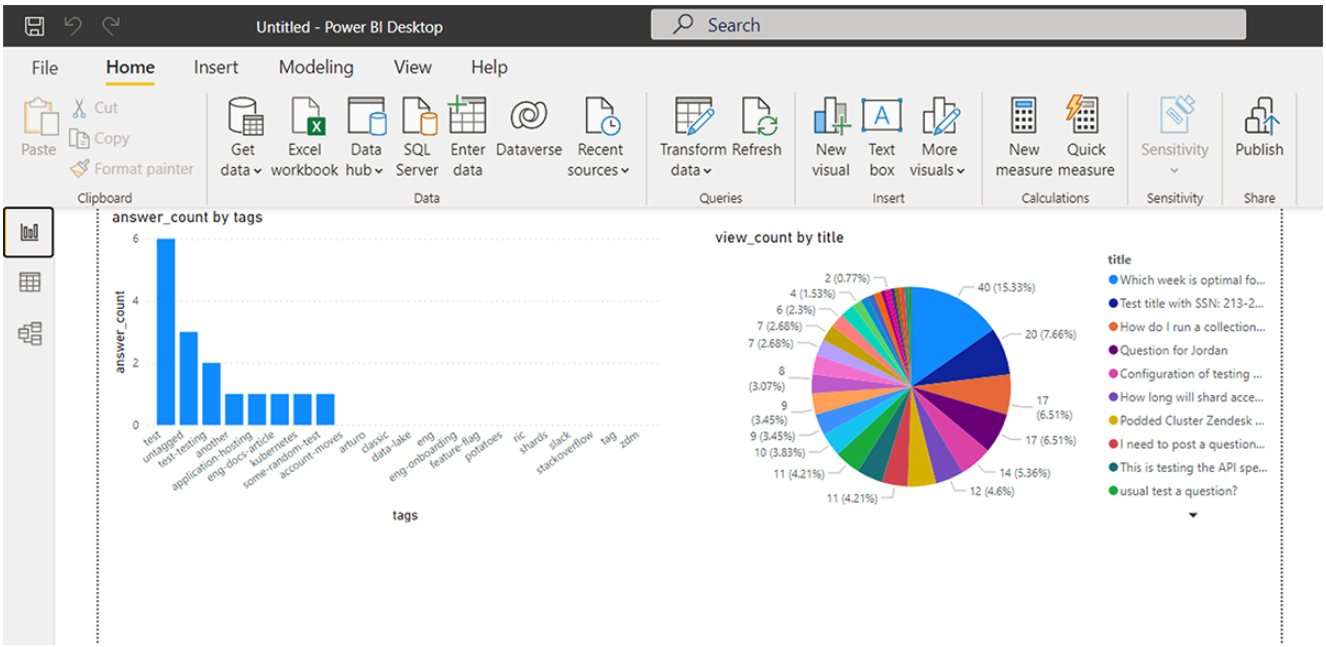
ADMIN PRIVILEGES REQUIRED

Applies to: Free | **Basic** | **Business** | Enterprise

Enterprise users can access their documentation here. Find your plan.

Overview

Microsoft Power BI (business intelligence) is a data visualization product that allows you to create interactive charts and reports.



There are many ways to connect Stack Overflow to Power BI to create a data dashboard from your Stack Overflow for Teams data. In this example, you'll use the Stack Overflow API, an automation job, Node.js, and Power BI datasets.

***NOTE:** This document demonstrates just one way to create a Power BI data dashboard with **Stack Overflow for Teams API v2.3**. Because of the many variables involved, Stack Overflow **does not** offer support for connecting your Teams API to Power BI.*

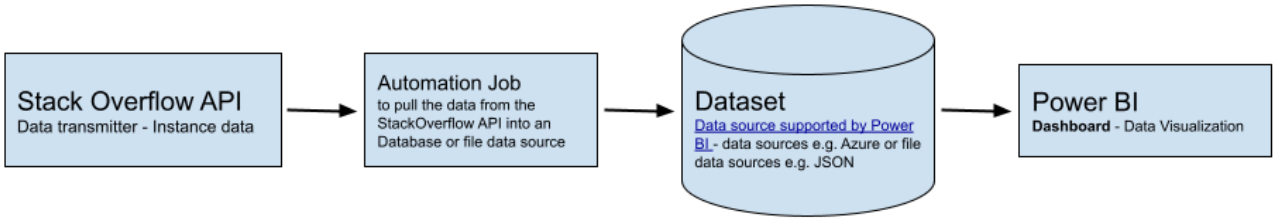
THIS ARTICLE APPLIES TO STACK OVERFLOW FOR TEAMS BASIC AND BUSINESS ONLY.
Stack Overflow for Teams Enterprise users should read this article instead. Find your plan.

Components of the data dashboard

A Stack Overflow data dashboard is useful for teams who want to monitor and improve their site's engagement. The dashboard is a quick and easy way for teams to see how many new questions are asked, how many remain unanswered, etc. There are many insights to be gleaned from your Stack Overflow data.

StackOverflow offers an API for data access, but Power BI doesn't connect directly to APIs. Instead, you'll use an automation job to pull data from the StackOverflow API into a data source. You'll then connect the Power BI dashboard to that data source.

Below is an overview of how the Stack Overflow API connects to the Power BI dashboard.



These instructions demonstrate just one of the many ways you can connect your Stack Overflow for Teams API to Power BI. For this example, you'll use the following components:

1. **StackOverflow for Teams API**
   You'll access a Stack Overflow for Teams API endpoint with your personal access token (PAT) to programmatically query question data, specifically the number of unanswered questions. (These questions may have responses, but none of the answers are marked as accepted.)
2. **Data Source**
   Because Power BI can't access the Stack Overflow API directly, you'll store dashboard data in a different data source. Power BI supports many data sources. In this example, you'll store the data in a local JSON file for simplicity.
3. **Automation Job**
   You'll use scheduled automation (a cron job) in a Node.js application to pull data from the Stack Overflow API and push it to the data source.
4. **Power BI**
   You'll use DataSets and Reports in Power BI Desktop to create your dashboard. The automation cron job will update the data in your file daily.

Install and configure the Node.js application

NOTE: *This document assumes you already have Node.js installed locally.*

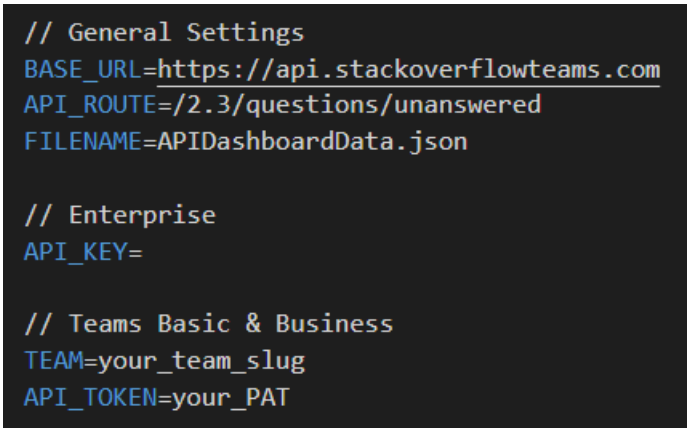
Download and unzip the Node.js cron application from this Google drive. Install the dependencies using the `npm ci` command. Learn more about npm from the official npm docs.

The cron application contains only a few files, and you'll change just one: the .env.template configuration file. Copy this file, and name the new version **.env**. Set the following configuration variables in the new .env file:

```
API_TOKEN=[your_PAT]
BASE_URL=https://api.stackoverflowteams.com
API_ROUTE=/2.3/questions/unanswered
FILENAME=APIDashboardData.json
TEAM=[team_slug]
```

For more information about API PATs, go to https://stackoverflowteams.help/en/articles/4385859.

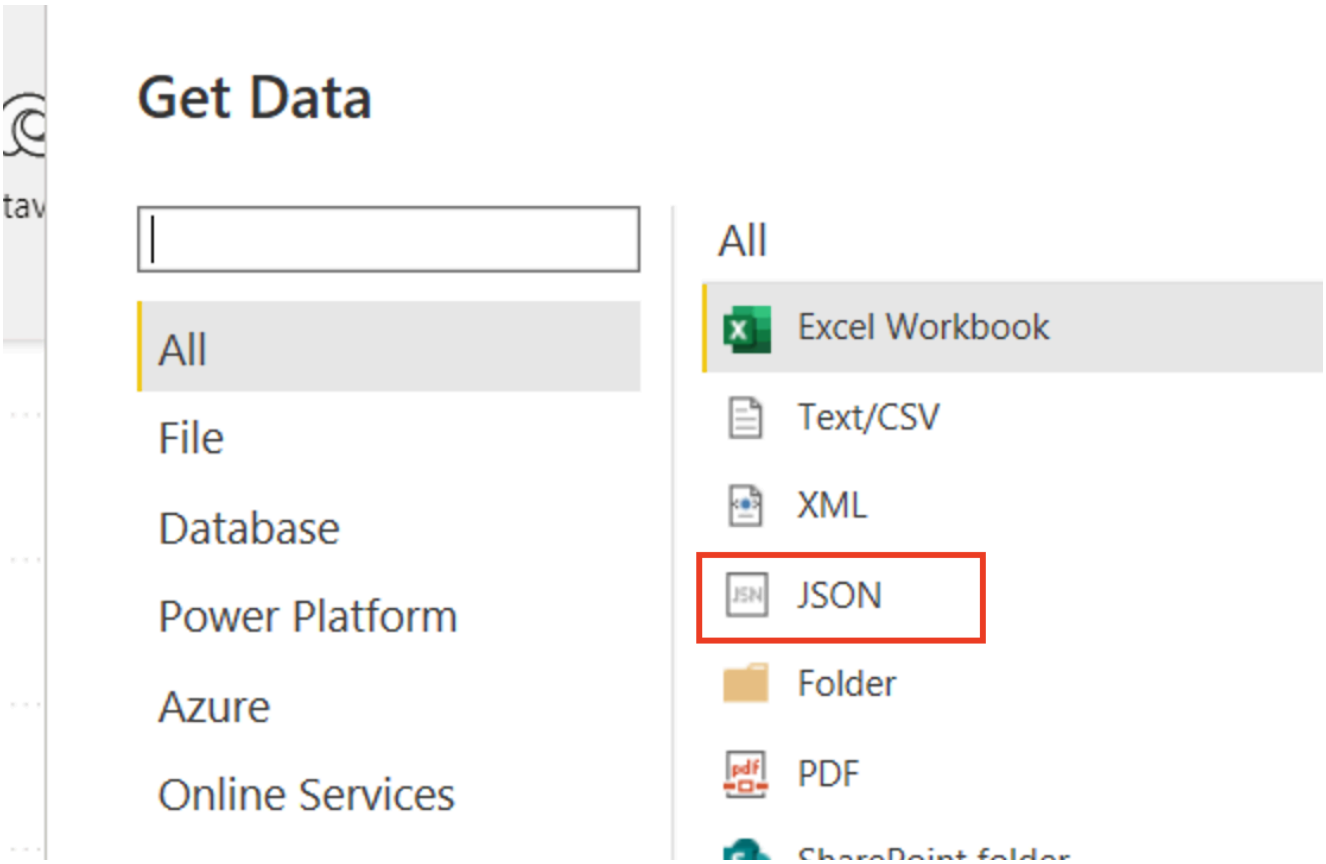
When you're done editing the .env file, it should look like this:



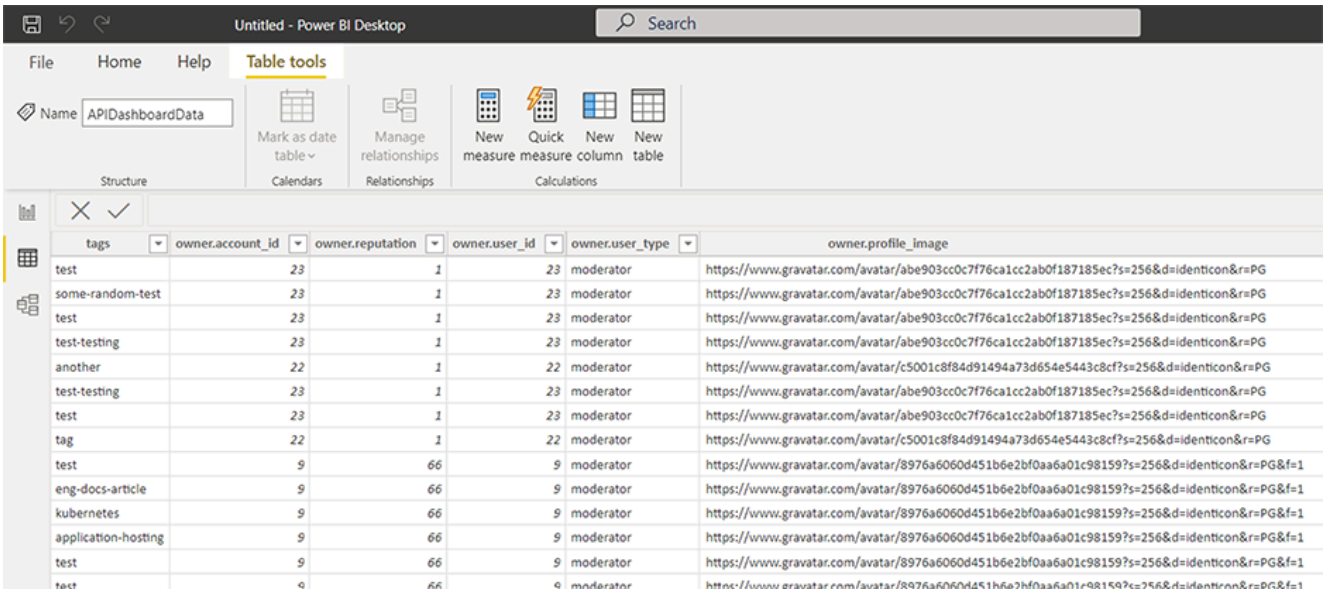
After setting the .env configuration variables, run the program using the `node index.js` command. The program will continue to run until it's stopped. You can stop running the script and cron job at any time by pressing CTRL-C.

When the cron job runs the first time, it will create a JSON file with your Stack Overflow for Teams data. To pull the data into Power BI:

1. Click **Get data - more**.
2. Click **JSON**.

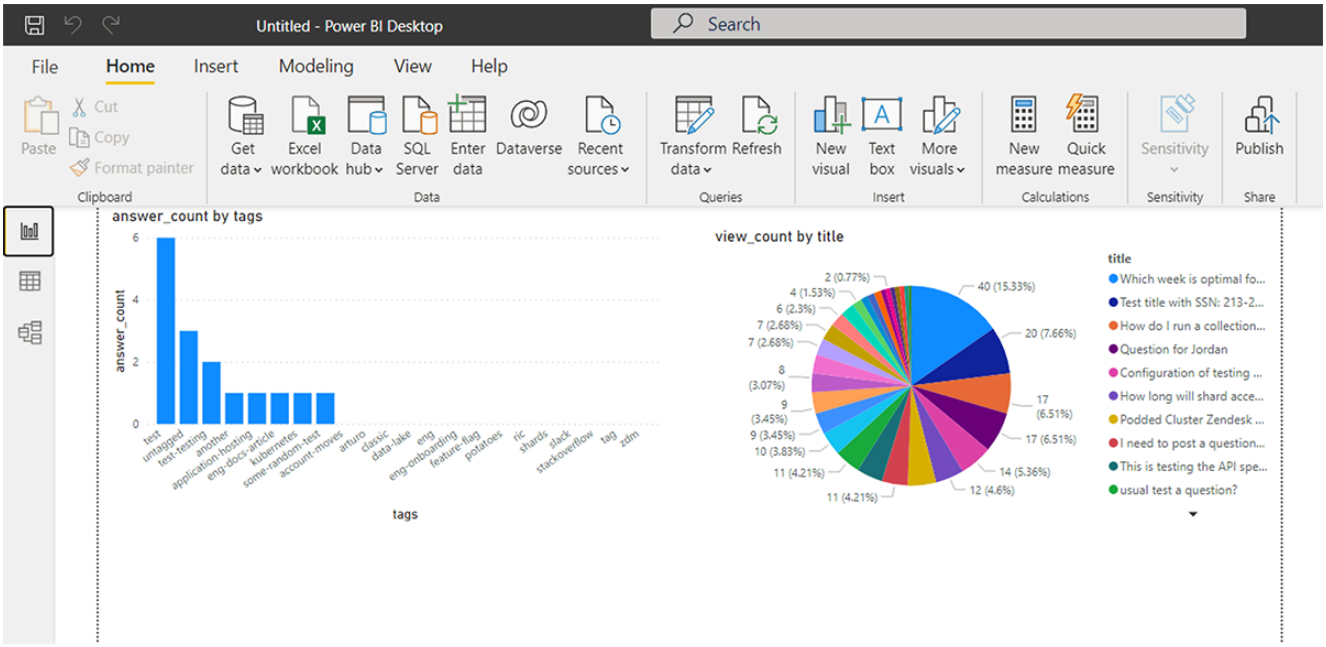


Power BI will pull the data in and show it in a table.



tags	owner.account_id	owner.reputation	owner.user_id	owner.user_type	owner.profile_image
test	23	1	23	moderator	https://www.gravatar.com/avatar/abe903cc0c7f76ca1cc2ab0f187185ec?s=256&d=identicon&r=PG
some-random-test	23	1	23	moderator	https://www.gravatar.com/avatar/abe903cc0c7f76ca1cc2ab0f187185ec?s=256&d=identicon&r=PG
test	23	1	23	moderator	https://www.gravatar.com/avatar/abe903cc0c7f76ca1cc2ab0f187185ec?s=256&d=identicon&r=PG
test-testing	23	1	23	moderator	https://www.gravatar.com/avatar/abe903cc0c7f76ca1cc2ab0f187185ec?s=256&d=identicon&r=PG
another	22	1	22	moderator	https://www.gravatar.com/avatar/c5001c8f84d91494a73d654e5443c8cf?s=256&d=identicon&r=PG
test-testing	23	1	23	moderator	https://www.gravatar.com/avatar/abe903cc0c7f76ca1cc2ab0f187185ec?s=256&d=identicon&r=PG
test	23	1	23	moderator	https://www.gravatar.com/avatar/abe903cc0c7f76ca1cc2ab0f187185ec?s=256&d=identicon&r=PG
tag	22	1	22	moderator	https://www.gravatar.com/avatar/c5001c8f84d91494a73d654e5443c8cf?s=256&d=identicon&r=PG
test	9	66	9	moderator	https://www.gravatar.com/avatar/8976a6060d451b6e2bf0aa6a01c98159?s=256&d=identicon&r=PG&f=1
eng-docs-article	9	66	9	moderator	https://www.gravatar.com/avatar/8976a6060d451b6e2bf0aa6a01c98159?s=256&d=identicon&r=PG&f=1
kubernetes	9	66	9	moderator	https://www.gravatar.com/avatar/8976a6060d451b6e2bf0aa6a01c98159?s=256&d=identicon&r=PG&f=1
application-hosting	9	66	9	moderator	https://www.gravatar.com/avatar/8976a6060d451b6e2bf0aa6a01c98159?s=256&d=identicon&r=PG&f=1
test	9	66	9	moderator	https://www.gravatar.com/avatar/8976a6060d451b6e2bf0aa6a01c98159?s=256&d=identicon&r=PG&f=1

You can now build reports and visualize your Stack Overflow for Teams data. Read this Microsoft Power BI quickstart guide for more information on creating reports and charts.



NOTE: *You'll need to perform a one-time, manual refresh in Power BI Desktop by clicking* ***Refresh*** *on the "Home" ribbon. When you click* ***Refresh****, the data in the file's model is refreshed with updated data from the original data source.*

How it works

Understanding how the Node.js code works will help you make changes to better meet your own reporting needs. We'll explore the index.js file in depth below to highlight the flexibility (and limitations) of this Power BI implementation.

NOTE: *This example shows just one of many possible ways to build a Power BI Dashboard using the Stack Overflow API. This script is flexible, and you can modify it to connect to other visualization tools.*

Build the API request

```
require('dotenv').config();
```

The function `buildRequest()` builds the API request using the `config()` install dependencies.

```
//Build API request
function buildRequest() {
  const {API_KEY, API_ROUTE, BASE_URL, TEAM, API_TOKEN} = process.env;

  //Teams request
```

The script differentiates between the Teams and Enterprise product environments based on the variables populated in the .env file. In this example, it will make an API call to your Teams data.

```
// Teams request
if (API_TOKEN && TEAM) {
  const url = new URL(API_ROUTE, BASE_URL);
  const params = new URLSearchParams({
    "team" : TEAM
  });

  url.search = params.toString();

  return {
      method: 'GET',
      url: url.toString(),
      headers: {
        'X-API-Access-Token': API_TOKEN
      }
  };
}
```

You'll set the `API_Route` variable to /2.3/questions/unanswered in this example to pull the number of unanswered questions. You could pull different data by changing the `API_Route` variable in the .env config file (see Install and configure the Node.js application above). Learn more about the available API routes at https://api.stackoverflowteams.com/docs.

NOTE: *API filters and pagination are beyond the scope of this example. You can set custom filters and pagination by changing the* `const url` *and* `headers` *variables. Learn more at https://api.stackoverflowteams.com/docs.*

Make the API call

```
//API request function
async function makeRequest() {
  const config = buildRequest();

  const res = await axios(config);

  console.log(res.status);

  return res.data.items;
}
```

The function `makeRequest()` uses the axios HTTP client to make an asynchronous API call.

```
const axios = require('axios');
```

Axios uses the data property (resource URL) and automatically stringifies data when sending JavaScript objects to the API. This makes it easy to serialize the data into a JSON string.

Write data to the data source

The function `writeDataToFile()` writes the API data to a JSON file. This is the data you'll upload daily into Power BI.

```
//Write data to file
function writeDataToFile (filename, data) {
  try {
```

If you choose a different file or database, consider the following:

- Your script shouldn't create a new file or database table every time the cron runs. Use the same file or database table each time.
- Your script shouldn't simply append data to the file or table, as the size of the data source will grow without limit. Instead, your script should overwrite data in the file or database with new data from the API call.

To accomplish this, the example script does the following in the `try` statement:

1. Combines new API data with data from the previous script execution (if any)
2. Scans the data set to eliminate duplicates based on identical `question_id` values
3. Overwrites the JSON data file with the new deduplicated data
4. Turns the data into `previousData` for the next script execution

```
function writeDataToFile (filename, data) {
  try {

    const combinedData = previousData.concat(data);
    const dedupedData = dedupe(combinedData, value => value.question_id);
    const stringifiedData = JSON.stringify(dedupedData);

    fs.writeFileSync(filename, stringifiedData);

    previousData = dedupedData;
```

In this example, you used the `question_id` value to deduplicate the data. Depending on what API route you use, and the data it returns, you may need to set a different unique identifier. If you query articles, for example, you'd likely use the `article_id` value to eliminate duplicate data.

Run the script

The cron job calls a function to execute the script. That function is `fetchAndWriteData()`, which combines the `makeRequest()` and `writeDataToFile()` functions. When first called, the `fetchAndWriteData()` function waits for the

results of the API call in the `makeRequest()` function. It then uses the `writeDataToFile()` function to prepare (deduplicate) the data and write it to the data file.

```
//Combine API request and add to file
async function fetchAndWriteData() {
  try {
    const filedata = await makeRequest();

    writeDataToFile(process.env.FILENAME, filedata);
```

Build an automation job

The `const nodeCron` command loads the node-cron library, which is required to schedule an automation job.

```
const nodeCron = require('node-cron');
```

You'll then schedule a cron job with nodeCron's `schedule()` function. In this example, the cron job executes the `fetchandWriteData()` function every day at midnight.

```
//Schedule a job to run every day at midnight
const job = nodeCron.schedule("0 0 * * *", fetchAndWriteData);
```

You can use other automation tools to create cron jobs, such as Azure's WebJob when using an Azure SQL database.

More on data sources

This example uses a simple JSON file as the database. We recommend JSON (or another file format) over a database when the data set is relatively small and you can store it in the application cache. For larger data sets, use a database. To use a database (Azure, for example), you'll need to prepare SQL tables for the data and change the function to include SQL queries.

Power BI also connects to workbooks on OneDrive or SharePoint Online, automatically checking hourly for changes to the data. If it detects that the data has changed, Power BI will refresh the data set and reports in the Power BI service.

Need help? Submit an issue or question through our support portal.