**Using the Fitbit Web API with Python, MongoDB**

**and Streamlit**

Diagram

Description automatically generatedA Beginner's Guide for Storing and Visualizing FitBit data

*Photo from* [*FitBit*](https://www.fitbit.com/dev) *site*

In today's data-driven world, the ability to collect, store, and analyze data has become more critical than ever before. Public APIs offer an excellent way to acquire data from various sources, and the Fitbit Web API is a perfect example of this.

It provides access to a wealth of fitness-related data, including step counts, heart rate, and sleep patterns, making up a rich source of information for developers, data scientists, and fitness enthusiasts.

In this tutorial, we will explore how to use the Fitbit Web API to acquire data, store it in a NoSQL database and visualize it on a web platform using [Streamlit](https://streamlit.io/), an open-source Python library for building beautiful custom web apps for machine learning and data science.

With this tutorial, you will learn how to extract insights from your fitness data and create a custom dashboard that displays your progress over time. Whether you are a fitness enthusiast or a data scientist, this tutorial is a valuable resource for anyone interested in working with Fitbit Web API data.

**Setup Fitbit Account**

The wearable that was used to collect fitness data from Fitbit Web API is **Fitbit Sense**, a health and fitness-based smartwatch.

Before starting data collection, we need to follow some steps, which are based on this [tutorial](https://towardsdatascience.com/using-the-fitbit-web-api-with-python-f29f119621ea).



*Fitbit Sense 2*

Firstly, we must create a Fitbit account from [here](https://accounts.fitbit.com/signup?targetUrl=https%3A%2F%2Fwww.fitbit.com%2Fglobal%2Fus%2Faccount). Just use a valid email and a password.

Graphical user interface, text, application, email

Description automatically generated

After you successfully create an account, go to [dev.fitbit.com](https://dev.fitbit.com/getting-started/) and from the upper right corner of the page, select **Manage/Register An App.**

Graphical user interface, application

Description automatically generatedYou will see a page like the below one (taken from the tutorial we attached before):

You need to specify **Personal** Application Type to be able to ask to download intraday data. The callback URL is http://127.0.0.1:8080 because the Python API we will use has it as the default redirect URL.

Next steps are pretty straightforward, so we will not mention them here.

**Data Collection**

**Data Storage**

**Data Visualization**

*I hope you find this tutorial useful. Please let us know if you have any thoughts or concerns.*

*Thanks for reading!*