|  |
| --- |
|  |
| Electricity Bill Management System API project documentation |
|  |

|  |
| --- |
| ZOUHIR KHALIL  07/12/2023 |

1. My contact
   * + Full name : ZOUHIR KHALIL
     + Email : [khalilz@hotmail.fr](mailto:khalilz@hotmail.fr)
     + LinkedIn : /khalil-zouhir-92382b236
     + GitHub : /KhalilZouhir98
     + Phone Number : 0642660196
2. Introduction

In the realm of data engineering, Application Programming Interfaces (APIs) play a pivotal role in facilitating the seamless exchange of data between diverse systems. APIs act as bridges, allowing different software applications to communicate, share information, and execute functions. This interconnectedness is especially crucial for data engineers, as it enables them to access, manipulate, and transport data efficiently. Recognizing the paramount importance of APIs in modern data ecosystems, I have chosen to embark on a small project to kickstart my career in data engineering.

The doc is structured into distinct sections, each contributing to a comprehensive understanding of the project.

* **Project Presentation and Goals:**
  + This section provides an overview of the project, outlining its objectives and the problems it aims to address.
* **Planning and Modeling Tools:** 
  + Here, we dive into the tools selected for project planning and modeling. From project management tools to data modeling software, this section outlines the strategic choices made to ensure the project's success.
* **Database Design and Modeling:**
  + This step is crucial for creating a robust foundation for data storage and retrieval. In this section, I discuss the chosen database model, schema design, and the rationale behind these decisions.
* **Project Implementation and Demonstration:**
  + this section showcases the chosen tools and technologies used for creating this API and it features screens of the requests made to the API and the corresponding responses.

This doc wont contain the api endpoints and [Environment Setup](https://chat.openai.com/c/1ee6f4ea-8ec8-4817-8ec5-e7c2b63e2f2a#environment-setup) to learn abou these things please check the flask api documentation .

1. Project presentation
   1. The objective of the project

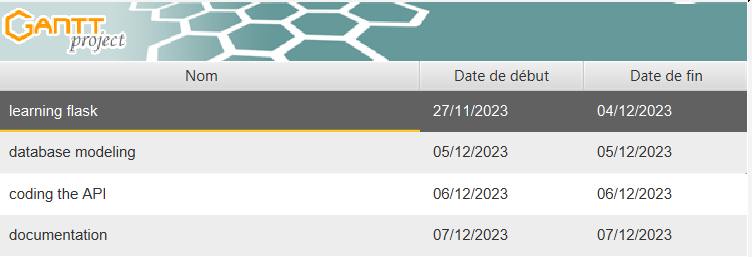
The primary objective of this project is to design and implement a robust Flask API for an Electricity Bill Management System, empowering users to perform essential operations seamlessly on the database . The core functionalities of the API revolve around the standard HTTP methods GET, POST, PATCH, and DELETE .

* List of operations :
* Create users, bills, meters and electricity readings of the meter .
* Retrieve users, bills, meters and electricity readings informations
* Update users, bills, meters and electricity readings informations
* Delete records

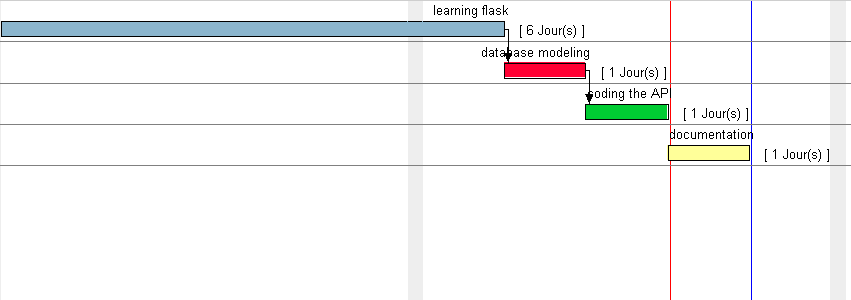
1. Planning and modeling tools
   1. Gantt chart

A Gantt chart is a visual project management tool that provides a quick and easy way to illustrate project timelines and tasks.

* **Tasks :**



* **Chart :**



* 1. Database modeling with merise

Merise is a data modeling methodology used in information system development. It provides a structured approach to analyze, design, and implement databases. This methodology emphasizes the organization of data and the relationships between different elements in a concise and systematic manner.



The modelisation tool we used to showacase our MCD and MLD is Draw.io its a free and web-based diagramming application used for creating flowcharts, diagrams, and other visual representations of ideas or processes.

1. database Design and modeling

Our database structure revolves around four primary entities: Users, Meters, Bills and electricity reading .

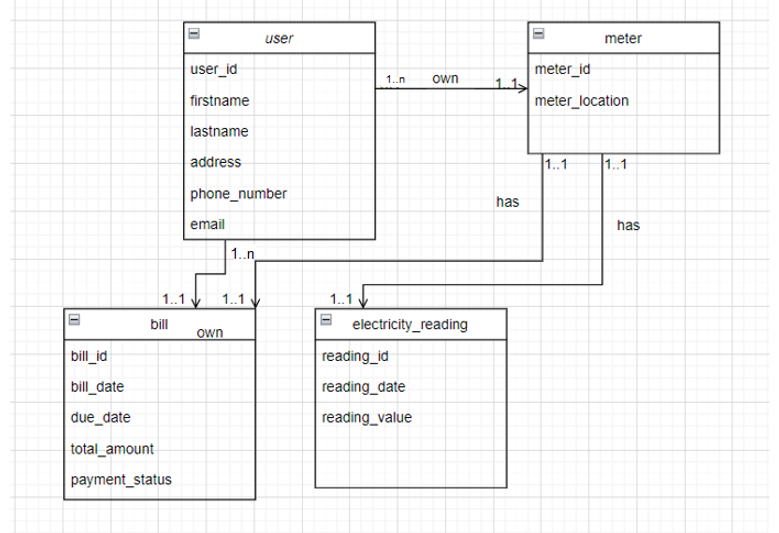
**User Model:** Each user is uniquely identified and characterized by basic information such as first name, last name, email address, and phone number.

**Meter Model:** Users can be associated with one or more meters, each defined by an ID and location.

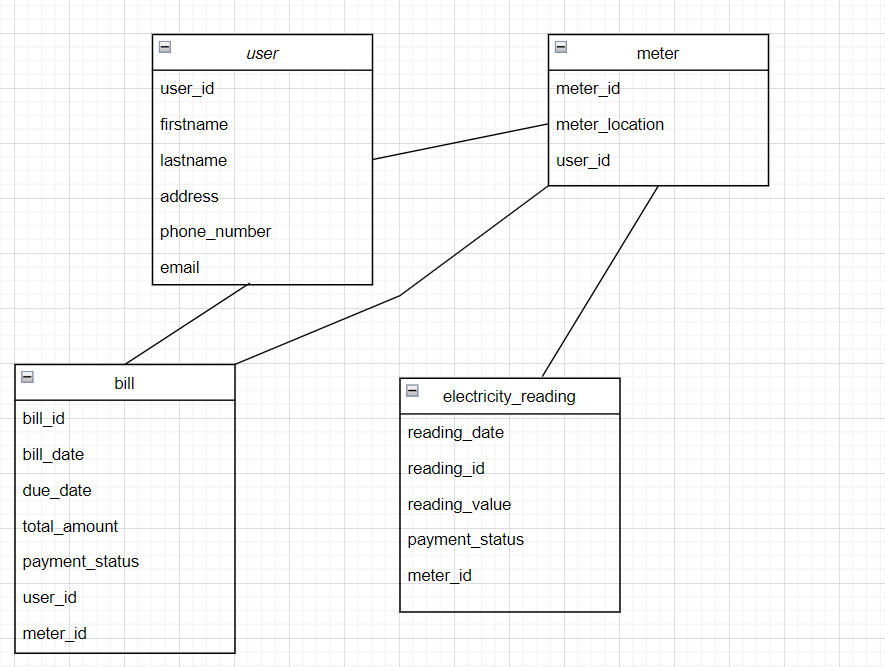
**Billing Model:** Bills are linked to specific meters, containing details like total value, date, due date, and payment status.

**Electricity reading :** each reading got an id a value and a date

* 1. MCD



* 1. MLD



1. Project implementation and Demonstration
   1. Programming languages and frameworks
      * 1. FLASK

Flask, a lightweight Python web framework, excels not only in building web applications but also in crafting robust APIs. Its simplicity and modularity make it an ideal choice for developers seeking a streamlined approach to API development. Flask's minimalistic design allows for quick setup, and its intuitive routing system makes defining API endpoints straightforward. With Flask, creating RESTful APIs becomes an effortless process, enabling developers to focus on designing and implementing the logic behind their endpoints without unnecessary complexities.



* + - 1. SQL

SQL, or Structured Query Language, is a powerful programming language designed for managing and manipulating relational databases. It serves as the standard interface for interacting with relational database management systems (RDBMS) and is employed across a wide spectrum of applications.



* + - 1. Python

Python is a high-level, versatile programming language known for its readability and simplicity. Widely used for web development, data analysis, artificial intelligence, and more, Python's syntax emphasizes code readability, making it an excellent choice for both beginners and experienced developers.

Obviously u cant code with flask without having basic knowledge with python



* 1. Development tools
     + 1. Xampp

XAMPP is a free, open-source cross-platform software stack that facilitates local web development. The name "XAMPP" stands for cross-platform, Apache, MariaDB/MySQL, PHP, and Perl. It includes an Apache web server, MariaDB/MySQL database, PHP, and Perl, providing a convenient environment for testing and developing web applications on a local machine.

* + - 1. MySQL

MySQL is a popular open-source relational database management system (RDBMS). Developed by Oracle Corporation, MySQL is known for its reliability, performance, and ease of use. It supports the structured query language (SQL) and is widely used for managing and manipulating databases in various applications, ranging from simple web applications to large-scale enterprise systems.

* + - 1. Apache

Apache, when utilized in the XAMPP stack, refers to the Apache HTTP Server—an open-source web server software. In the context of XAMPP, Apache serves as the web server component, handling HTTP requests and facilitating the deployment of web applications. Paired with MySQL and other tools in XAMPP, Apache provides a comprehensive local development environment, enabling users to build, test, and deploy dynamic websites and web applications efficiently.

* + - 1. PhpMyadmin

phpMyAdmin is a web-based graphical user interface (GUI) for managing MySQL and MariaDB databases. Developed in PHP, phpMyAdmin allows users to interact with their databases through a web browser, providing a user-friendly interface for tasks such as database creation, modification, and data manipulation. It simplifies the administration of MySQL databases, making it a popular tool for developers and administrators working with MySQL-based systems.

* + - 1. Pycharm

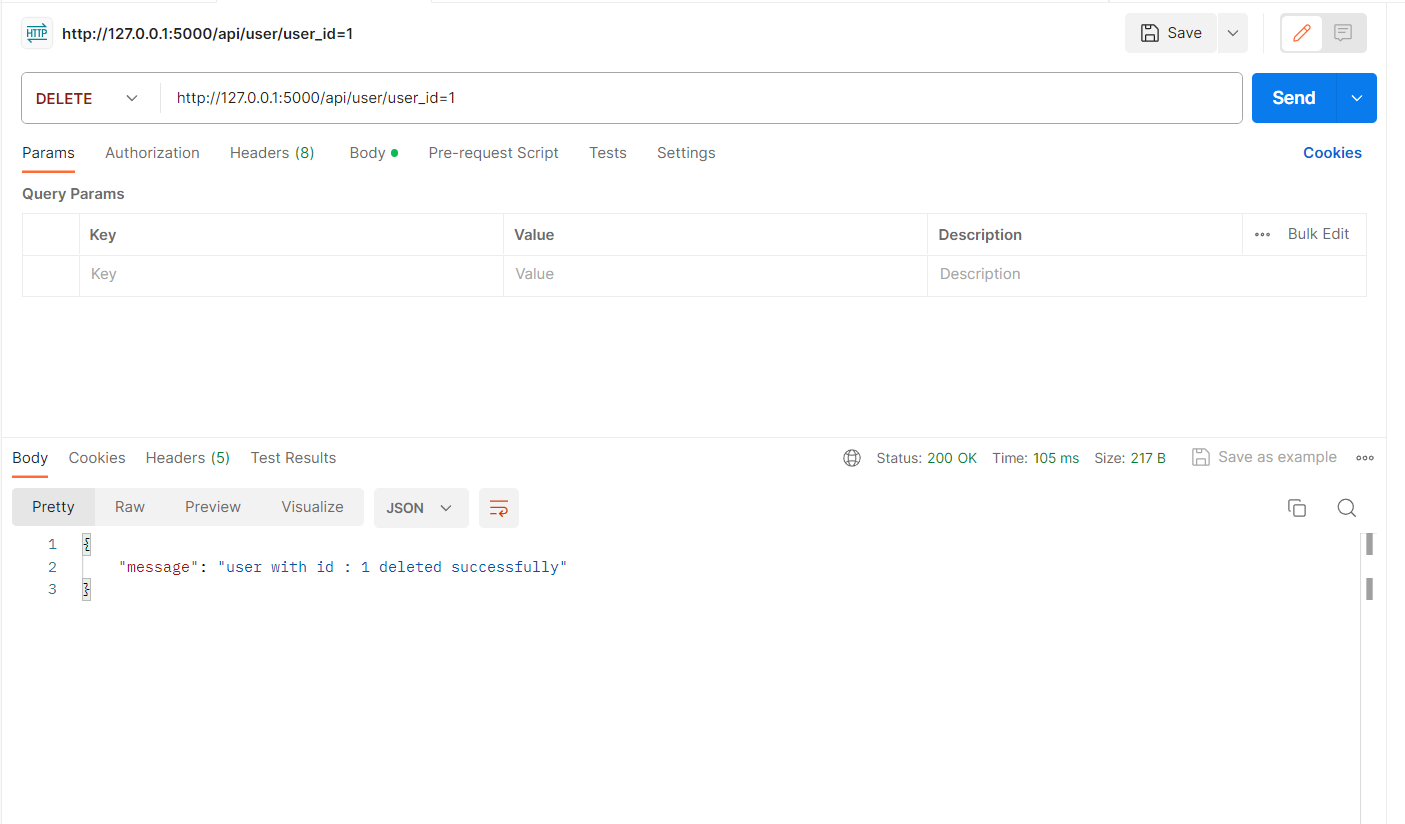
PyCharm is a powerful integrated development environment (IDE) specifically designed for Python. Developed by JetBrains, PyCharm provides a comprehensive set of tools for Python developers, including code analysis, intelligent code completion, debugging, and testing features. With its user-friendly interface and robust capabilities, PyCharm enhances productivity and streamlines the development workflow, making it a popular choice among Python developers.

* + - 1. Postman

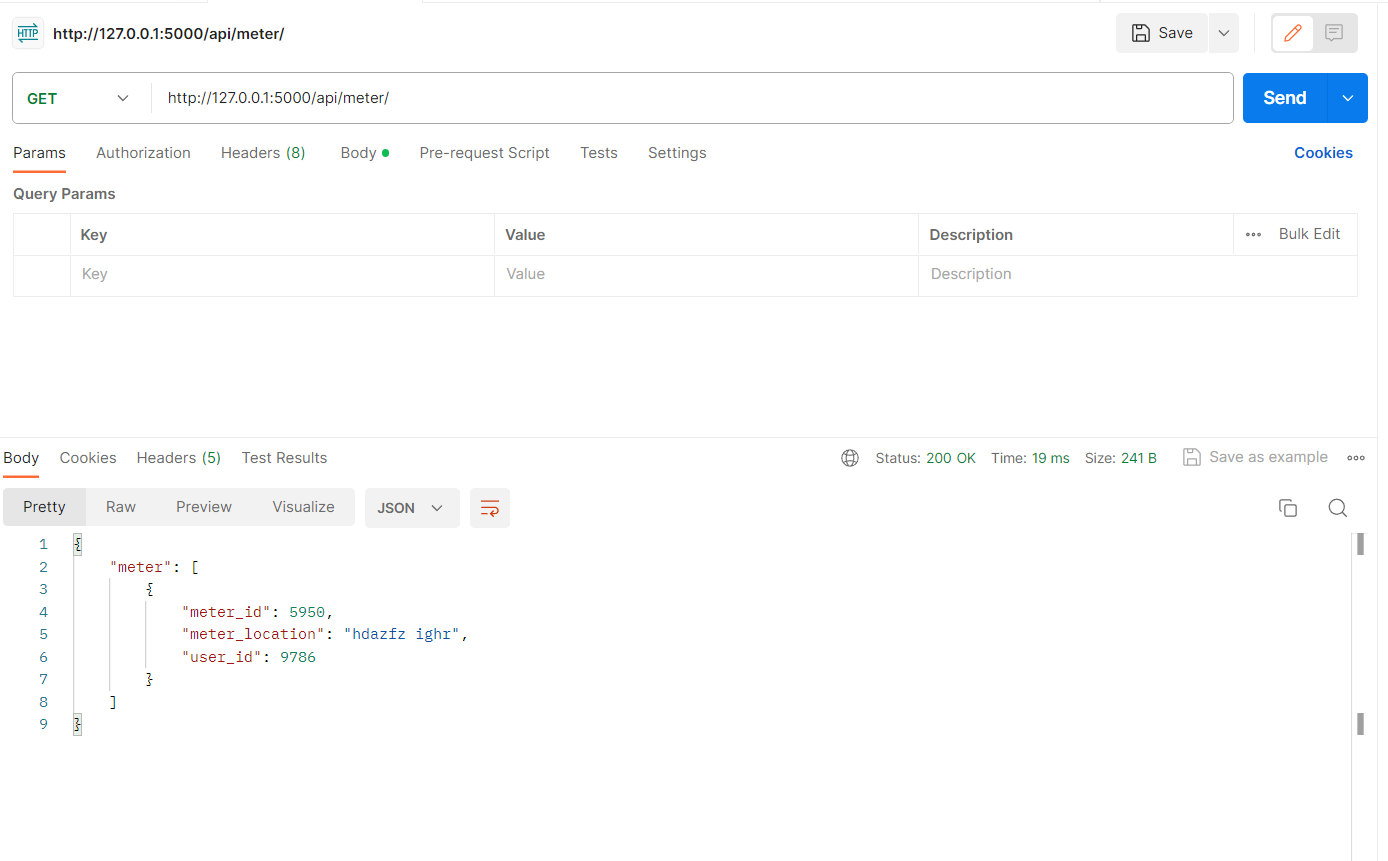
Postman is a widely-used collaboration platform for API development. Originally known as a simple HTTP client for testing APIs, Postman has evolved into a comprehensive toolset that simplifies the entire API development process. It allows developers to design, test, and document APIs efficiently. With features such as request building, testing, and automated workflows, Postman facilitates seamless communication and collaboration among developers, testers, and other stakeholders involved in API development.



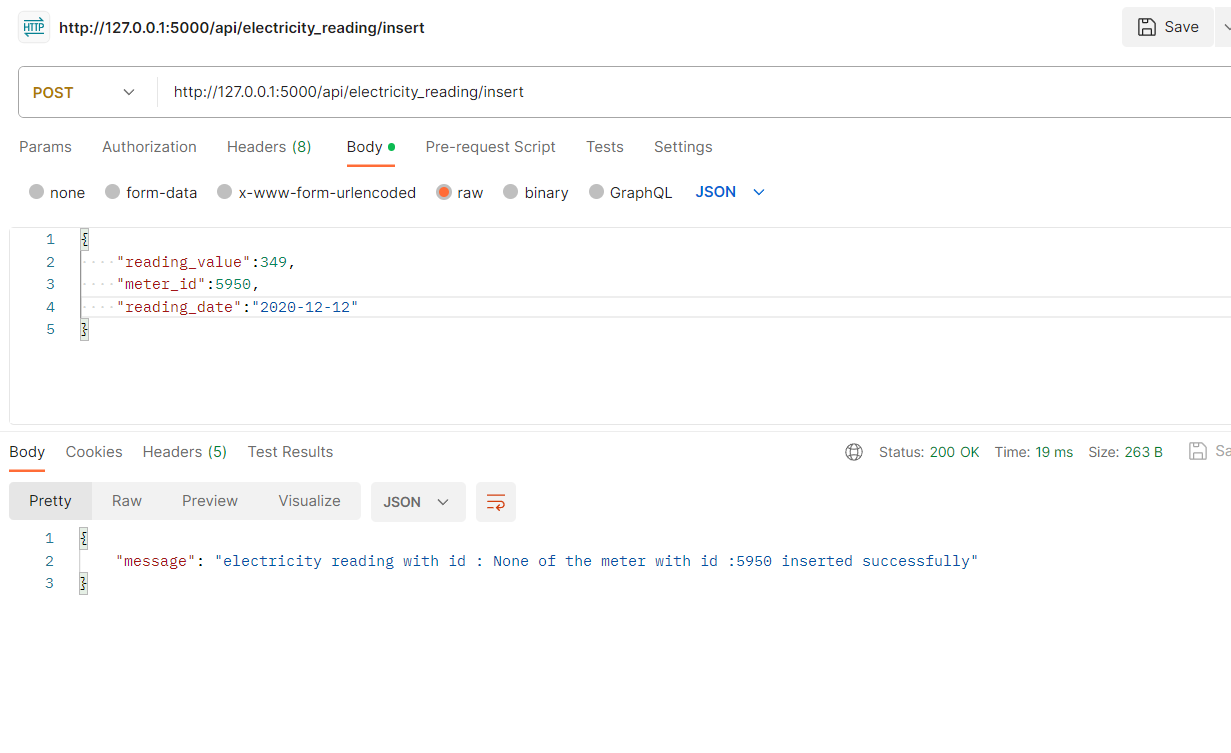
* 1. Demonstration
* Delete



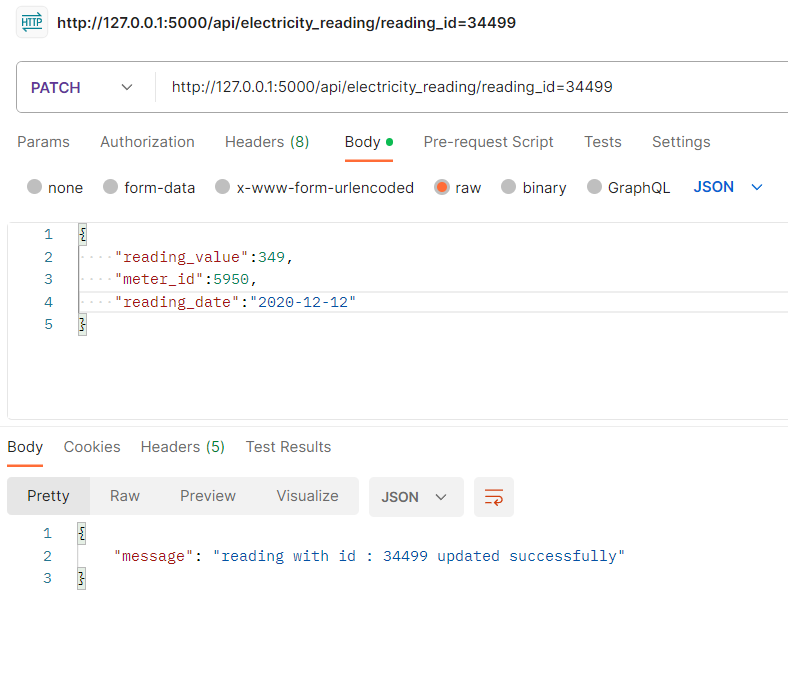
* Get



* Post



* Patch



For more infos about the api endpoints and [Environment Setup](https://chat.openai.com/c/1ee6f4ea-8ec8-4817-8ec5-e7c2b63e2f2a#environment-setup)  please check the flask api documentation in the repo.