

Explaining CRUD in the Patient Management API

CRUD stands for the four basic functions of persistent storage: **Create**, **Read**, **Update**, and **Delete**. In an API, these functions are typically mapped to HTTP methods. Here's how they apply to our `patient_management_api.py` file.

1. Create => POST

- **Endpoint:** POST `/patients/`
- **Purpose:** To add a new patient record to our system.
- **How it Works:** The client (like a web frontend) sends a JSON object containing the new patient's details (name, age, sex, illness) in the request body. Our API receives this data, assigns a new unique id to it, and saves it in our `patients_db` list. It then returns the complete patient record, including the new id, back to the client.

2. Read => GET

- **Endpoint 1:** GET `/patients/`
 - **Purpose:** To retrieve a list of all patient records.
 - **How it Works:** When a client makes a GET request to this URL, the API simply returns the entire `patients_db` list as a JSON array.
- **Endpoint 2:** GET `/patients/{patient_id}`
 - **Purpose:** To retrieve the details of one specific patient.
 - **How it Works:** The client includes the unique id of the patient they want in the URL path. The API searches the `patients_db` for a patient with that matching id. If found, it returns that patient's data. If not, it returns a 404 Not Found error.

3. Update => PUT

- **Endpoint:** PUT `/patients/{patient_id}`
- **Purpose:** To modify or edit the details of an existing patient.
- **How it Works:** The client specifies which patient to update by putting their id in the URL. The new information for the patient is sent in the request body. The API first finds the existing patient. If the patient exists, it updates their details with the new data provided and returns the updated record. If the patient doesn't exist, it returns a 404 Not Found error.

4. Delete => DELETE

- **Endpoint:** DELETE `/patients/{patient_id}`
- **Purpose:** To permanently remove a patient record from the system.
- **How it Works:** The client sends the id of the patient they want to remove in the URL path. The API finds that patient in the `patients_db` and removes them. If the patient can't be found, it returns a 404 Not Found error. If successful, it returns a confirmation message.