**Challenge Problem: Go Engineer API Development**

*Functional Requirements:*

1. Create Contact Endpoint:
   * Method: POST
   * Path: /contacts
   * Request Body: JSON object representing the contact
   * Response: JSON object of the created contact with an assigned ID
2. Get Contact by ID Endpoint:
   * Method: GET
   * Path: /contacts/{id}
   * Path Parameter: id (contact ID)
   * Response: JSON object of the contact with the provided ID
3. Delete Contact Endpoint:
   * Method: DELETE
   * Path: /contacts/{id}
   * Path Parameter: id (contact ID)
   * Response: No content (204 status code) upon successful deletion
4. Update Contact Endpoint:
   * Method: PUT
   * Path: /contacts/{id}
   * Path Parameter: id (contact ID)
   * Request Body: JSON object representing the updated contact
   * Response: JSON object of the updated contact

*Non-functional Requirements:*

* Document the source code thoroughly using comments to explain the functionality and any important design decisions.
* Store contact data in a memory list for simplicity; persistence is not required.
* Do not implement any authorization logic (e.g., token-based authentication).
* Adhere to Go best practices for naming conventions, code structure, and error handling.
* **Utilize code generation tools like Copilot, GPT, or others as much as possible to expedite development.**
* Provide an estimate of the time it took to complete the project.
* Include a POSTMAN collection with examples of how to call each endpoint.

This challenge problem aims to assess your ability to develop a basic RESTful API using Go while adhering to best practices and documenting your code effectively. Good luck!