**Theory: RESTful API Design**

When designing a RESTful API, there are several principles and best practices that help make the API scalable, maintainable, and easy to use. These principles are commonly used in applications that follow the REST (Representational State Transfer) architecture style.

**Key REST Principles:**

1. **Statelessness:**
   * Each request from a client to a server must contain all the necessary information to understand and process the request. The server does not store anything about previous client requests.
   * This means that each API request is independent and self-contained.
2. **Resource-Based URLs:**
   * In REST, everything is considered a resource, and these resources are identified by URLs. A good URL design is essential for creating RESTful APIs.
   * Example: /doctors/ would represent a collection of doctor profiles, and /doctors/{id}/ would represent a specific doctor by their unique ID.
3. **Using HTTP Methods for CRUD Operations:**
   * RESTful APIs leverage standard HTTP methods to perform CRUD (Create, Read, Update, Delete) operations:
     + **GET**: Retrieve data from the server (Read).
     + **POST**: Send data to the server to create a new resource (Create).
     + **PUT/PATCH**: Update an existing resource on the server (Update).
     + **DELETE**: Remove a resource from the server (Delete).