# API Design

□□□□□□□□□□□□□□□-□□□□□□□□□□□□□□ □□□□□□□□□□□□:

• Nouns in URLs:  Use descriptive nouns (e.g., /□□□□□□□□□□, /□□□□□□□□□□□□□□□□□□) to represent resources, not verbs indicating actions.

• HTTP Verbs:  Leverage HTTP verbs (GET, POST, PUT, DELETE) to denote actions on resources (GET: retrieve,  POST: create, PUT: update, DELETE: remove).

• Plural Nouns for Collections:  Identify collections of resources with plural nouns (e.g., /□□□□□□□□□□, not /□□□□□□□□□).

• Nested Resources:  Model hierarchical relationships with nested URLs (e.g., /□□□□□□□□□□/: □□□□□□□□□□□□/□□□□□□□□□□).

□□□□□□□□ □□□□□□□□□□□□□□□ & □□□□□□□□□□ □□□□□□□□□□□□□□□□□□:

• Standardized Formats:  Opt for industry-standard data formats like JSON or XML for request and response payloads.

• Descriptive Error Codes:  Utilize HTTP status codes (e.g., 200: success, 400: bad request, 404: not found) and provide clear error messages for troubleshooting.

• Validation:  Implement robust input validation on the server-side to prevent malformed requests.

□□□□□□□□□□□□□□□□□□□□□□□□ & □□□□□□□□□□□□□□□□□□□□□□□□:

• Caching:  Utilize caching mechanisms to reduce server load and improve response times for frequently accessed data.

• Pagination:  Enable result pagination (e.g., limit, offset parameters) to handle large datasets efficiently.

• Rate Limiting:  Implement rate limiting to prevent abuse and ensure fair access for all users.

□□□□□□□□□□□□□□□□:

• Authentication & Authorization:  Enforce proper authentication and authorization mechanisms to control access to sensitive data and functionalities.

• HTTPS:  Always enforce HTTPS for secure communication and data encryption.

□□□□□□□□□□□□□□□□□□ □□□□□□□□□□□□□□□□□□□□□□□□□□□□ :

• Versioning:  Implement API versioning to manage changes and ensure compatibility with existing integrations.

• Documentation:  Provide comprehensive API documentation covering endpoints, parameters, responses, and code samples.

• Testing:  Write thorough unit and integration tests to ensure API functionality and maintainability.

By adhering to these technical best practices, you can create a robust, well-documented, and secure API that empowers developers and fosters a thriving developer ecosystem.

---

Revision #2
Created 14 April 2024 17:17:58 by sedawk
Updated 10 February 2025 17:19:34 by sedawk