

Understanding APIs and RESTful Architecture

1.3 HTTP Methods and Status Codes

Learning objectives

- ▶ What are HTTP methods?
- ▶ How methods map to CRUD operations
- ▶ Safe and idempotent methods
- ▶ Common HTTP status codes

What Is HTTP?

- ▶ HyperText Transfer Protocol
- ▶ Foundation for communication between client and server
- ▶ Every API request uses HTTP to send and receive data
- ▶ Methods define what action the client wants to perform
- ▶ Responses include status codes and data

Common HTTP Methods

Method	Description	Typical Use
GET	Retrieve data	Reading resources
POST	Send new data	Creating resources
PUT	Replace existing data	Updating or replacing a resource
PATCH	Partially update data	Updating one or more fields
DELETE	Remove data	Deleting resources

Mapping CRUD to HTTP Methods

CRUD Operation	HTTP Method	Example Action
Create	POST	Add a new user
Read	GET	Retrieve a list of books
Update	PUT or PATCH	Modify existing order
Delete	DELETE	Remove a customer record

Method and Endpoint

- ▶ Create → POST /api/books
- ▶ Retrieve one → GET /api/books/{id}
- ▶ Retrieve many → GET /api/books
- ▶ Update → PUT or PATCH /api/books/{id}
- ▶ Delete → DELETE /api/books/{id}

Safe and Idempotent Methods

- ▶ Safe methods: Do not change data
 - ▶ Example: GET – only reads data
- ▶ Idempotent methods: Multiple identical requests have the same effect
 - ▶ Example: PUT and DELETE
- ▶ POST is neither safe nor idempotent
- ▶ Important for designing predictable APIs

HTTP Status Codes (1/2)

- ▶ 200 OK: Successful request
- ▶ 201 Created: Resource created
- ▶ 204 No Content: Successful, no body returned
- ▶ 400 Bad Request: Invalid input
- ▶ 401 Unauthorized / 403 Forbidden: Auth issues

HTTP Status Codes (2/2)

- ▶ 404 Not Found: Resource doesn't exist
- ▶ 409 Conflict: Duplicate or invalid state
- ▶ 500 Internal Server Error: Generic server failure
- ▶ 503 Service Unavailable: Temporary issue
- ▶ Always choose the most specific status code

Status Code Categories

Range	Category	Meaning
1xx	Informational	Request received, processing
2xx	Success	Request succeeded
3xx	Redirection	Further action needed
4xx	Client Error	Bad request from client
5xx	Server Error	Server failed to fulfill request

Demo: Observe Methods and Status Codes

- ▶ Goal: See real examples of HTTP methods and status codes in action
- ▶ Show:
 - ▶ GET <https://jsonplaceholder.typicode.com/posts/1> → returns 200 OK
 - ▶ POST <https://jsonplaceholder.typicode.com/posts> with dummy JSON → returns 201 Created
 - ▶ Invalid URL → 404 Not Found

Lab: Match HTTP Methods to Actions

- ▶ Goal: Practice identifying which HTTP methods and status codes fit common API operations.
- ▶ Instructions:
 - ▶ For each scenario, choose the most appropriate HTTP method.
 - ▶ Predict what status code the server should return.
 - ▶ Write your answers in your notes (scenario table on next slide)

Lab: Match HTTP Methods to Actions

Scenario	Method?	Status Code?
Retrieve all books	?	?
Add a new order	?	?
Delete a customer	?	?
Update a task's title	?	?
Request a non-existing product	?	?

Lab - Example Solutions

Scenario	Method	Status Code
Retrieve all books	GET	200 OK
Add a new order	POST	201 Created
Delete a customer	DELETE	204 No Content
Update a task's title	PUT or PATCH	200 OK
Request a non-existing product	GET	404 Not Found

Lab - Reflection

- ▶ Why are some methods safe while others are not?
- ▶ What's the difference between 200 and 201?
- ▶ Why is idempotency important for reliability?
- ▶ Which codes would you show to users in an app?
- ▶ How do these methods relate to CRUD operations?

Key Takeaways

- ▶ HTTP defines standard actions for APIs
- ▶ CRUD maps directly to GET, POST, PUT, DELETE
- ▶ Safe methods don't modify data
- ▶ Idempotent methods give predictable results
- ▶ Status codes communicate outcome of requests
- ▶ Together, these form the foundation of REST APIs