Chapter 3 Understanding RESTful API Design

Lecture Duration: 90 Minutes

🕒 *Goal:* Grasp HTTP concepts and REST principles to prepare for API programming in the next workshop.

UbiN3$
Ubiquitous Networked Embedded System

1 Fundamentals of HTTP

What is HTTP?

- **Hypertext Transfer Protocol** — the foundation of web communication.
- Client–server model:
    - **Client:** sends requests (browser, Postman, app)
    - **Server:** returns responses (data, HTML, JSON)
- Stateless: Each request is **independent**.

🧠 *Key Idea:* HTTP enables interaction between frontend and backend systems.

HTTP Message Structure

Request

```
GET /students HTTP/1.1
Host: api.example.com
Accept: application/json
```

Response

```
HTTP/1.1 200 OK
Content-Type: application/json

[{ "id": 1, "name": "Alice" }]
```

2 HTTP Methods & Status Codes

Common HTTP Methods

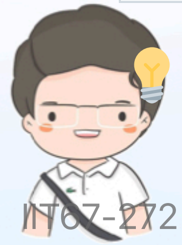
Method	Description	Example
GET	Retrieve data	`/students`
POST	Create new data	`/students`
PUT	Update entire record	`/students/1`
PATCH	Update partial record	`/students/1`
DELETE	Remove data	`/students/1`

🧩 *Mnemonic*: CRUD → **Create, Read, Update, Delete** maps to **POST, GET, PUT/PATCH, DELETE**.

UbiN3$
Ubiquitous Networked Embedded System

HTTP Status Codes

Code	Meaning	Usage
200 OK	Success	Data retrieved
201 Created	Resource created	After POST
400 Bad Request	Invalid input	Validation error
401 Unauthorized	Authentication required	Login needed
404 Not Found	Resource missing	Wrong URL or ID
500 Internal Server Error	Server failure	Unexpected error



Tip: Always return clear and consistent status codes in your API design.

3 Request–Response Cycle

Client–Server Interaction

```
+------------+          +-------------------+
|   Client   |  --->    |  HTTP Request     |
| (Browser)  |           -----------------
|            |  <---    |  HTTP Response    |
+------------+          +-------------------+
```

- The **client** sends HTTP requests.
- The **server** processes and returns a **response**.
- Each request includes:
    - **Method**
    - **URL**
    - **Headers**

Example API Call (JSON)

Request

```
POST /students
Content-Type: application/json

{
  "name": "Bob",
  "email": "bob@university.edu"
}
```

Response

```
HTTP/1.1 201 Created
{
  "id": 1,
  "name": "Bob",
  "email": "bob@university.edu"
}
```

4 RESTful API Principles

What is REST?

- **Representational State Transfer**
- Architectural style for designing networked APIs
- Key principles:
    1. **Stateless communication**
    2. **Uniform interface (HTTP methods)**
    3. **Resource-based URLs**
    4. **Representation via JSON or XML**
    5. **Client–server separation**

UbiN3$
Ubiquitous Networked Embedded System

RESTful Resource Design

Resource	Endpoint	Example
Student	`/students`	List all students
Student (by ID)	`/students/{id}`	`/students/5`
Course	`/courses`	`/courses/3`

💡 *Guideline:* Use **nouns**, not verbs, for endpoint names.

UbiN3S
Ubiquitous Networked Embedded System

Example CRUD Endpoints for “Student Management”

Operation	HTTP Method	Endpoint	Description
Create Student	POST	`/students`	Add new student
Retrieve All	GET	`/students`	List all students
Retrieve by ID	GET	`/students/{id}`	Find student
Update Student	PUT	`/students/{id}`	Modify data
Delete Student	DELETE	`/students/{id}`	Remove student

UbiN3$ Ubiquitous Networked Embedded System

Data Transfer Objects (DTOs)

- DTO = **Data Transfer Object**
- Used to send or receive data between client and server.
- Ensures only necessary data is exposed.

```
{
  "id": 3,
  "name": "Sara",
  "email": "sara@uni.edu"
}
```

🧠 *Concept:* DTOs protect your internal model and simplify validation.

UbiN3$
Ubiquitous Networked Embedded System

Validation Concept

- Ensures correctness of data before processing.
- Example: Using annotations in backend frameworks
  - `@NotNull`, `@Email`, `@Size`
- Prevents invalid API requests.

5 Testing APIs with Postman

What is Postman?

- A popular API testing and debugging tool.
- Allows sending HTTP requests manually.
- Useful for testing:
    - Methods (GET, POST, PUT, DELETE)
    - Headers and tokens
    - JSON bodies and responses

UbiN3$ Ubiquitous Networked Embedded System

Postman Interface Overview

- **Request Builder:** Choose HTTP method and endpoint.
- **Headers Tab:** Define authentication or content type.
- **Body Tab:** Provide JSON for POST/PUT.
- **Response Panel:** Inspect returned data and status.

🧠 *Activity:* Test `/students` endpoints using different HTTP methods.

UbiN3$ Ubiquitous Networked Embedded System

Example Postman Test

1. POST Request

```
POST /students
Body:
{
  "name": "Lina",
  "email": "lina@uni.edu"
}
```

2. GET Request

```
GET /students
```

3. DELETE Request

```
DELETE /students/1
```

Summary

✅ Learned:

- HTTP fundamentals
- Methods and status codes
- Request–response flow
- RESTful API design principles
- API testing with Postman

🚀 *You are now ready to build a CRUD API in the next workshop.*

UbiN3$
Ubiquitous Networked Embedded System

Thank You!

Questions?

💬 *Instructor:* Asst. Prof. Chanankorn Jandeng

🏫 *Walailak University – IIT67-272*

UbiN3S
Ubiquitous Networked Embedded System