REST API Workshop Curriculum: From Concepts to Code

DAY 1 (2 hours): Foundation & Design Principles

Opening (10 minutes)

- Welcome & objectives
- Audience poll: experience levels
- Workshop roadmap & connection to future MCP Server work

Module 1: Quick Networking Fundamentals (20 minutes)

- HTTP request/response cycle (5 min)
- JSON data format basics (5 min)
- Client-server architecture (5 min)
- Status codes overview (5 min)

Module 2: REST Philosophy & Principles (35 minutes)

- **What is REST?** (10 min)
    - REpresentational State Transfer
    - Why REST over RPC or GraphQL for industrial systems
    - Stateless, cacheable, uniform interface
- **Resource-Centric Thinking** (15 min)
    - Resources vs. actions
    - Example: Library system (Books, Authors, Members)
    - Hospital system (Patients, Appointments, Departments)
    - *Pseudo-code examples throughout*
- **REST Constraints & Benefits** (10 min)
    - Scalability in manufacturing environments
    - Interoperability with existing systems

Module 3: Resource Modeling & URL Design (40 minutes)

- **Identifying Resources** (15 min)
    - Library example: **/books**, **/authors**, **/members**
    - School example: **/students**, **/courses**, **/grades**
    - Manufacturing teaser: **/machines**, **/sensors**, **/production-orders**
- **Naming Conventions Best Practices** (15 min)
    - Nouns not verbs: **/books** not **/getBooks**
    - Plural vs singular: **/books/123**
    - Hierarchical relationships: **/books/123/reviews**
    - Query parameters for filtering: **/books?author=smith&year=2023**
- **URL Structure Workshop** (10 min)
    - Interactive exercise: Design URLs for a hospital system

Module 4: HTTP Methods & Status Codes (30 minutes)

- HTTP Response Codes Descriptions
- **HTTP Verbs Deep Dive** (20 min)
    - GET: Retrieving resources
    - POST: Creating new resources
    - PUT: Updating entire resources
    - PATCH: Partial updates
    - DELETE: Removing resources
    - *When to use each in manufacturing contexts*
- **Status Codes That Matter** (10 min)
    - 2xx Success family
    - 4xx Client errors (400, 401, 404, 422)
    - 5xx Server errors (500, 503)

Day 1 Wrap-up (5 minutes)

- Key takeaways

- Preview of Day 2 hands-on coding

DAY 2 (2 hours): Implementation & Best Practices

Warm-up & Day 1 Recap (10 minutes)

- Quick review of REST principles
- Setup check for hands-on coding

Module 5: FastAPI Hands-on Implementation (70 minutes)

Part A: Project Setup (15 min)

python Collapse Run Save Copy

```python
# Live coding with LLM assistance
# Creating a simple Library Management API
from fastapi import FastAPI, HTTPException
from pydantic import BaseModel
from typing import List, Optional
```

Part B: Basic CRUD Operations (25 min)

- **Models with Pydantic**

python Collapse Run Copy

```python
class Book(BaseModel):
    id: int
    title: str
    author: str
    isbn: Optional[str] = None
```

- **GET endpoints**

python Collapse Run Copy

```python
@app.get("/books")
@app.get("/books/{book_id}")
```

Part C: Creating and Updating Resources (20 min)

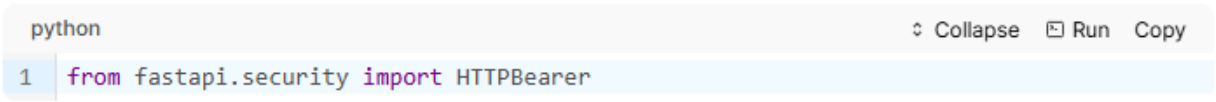
- **POST for creation**
- **PUT for updates**
- **Error handling with HTTPException**

Part D: Manufacturing Context Application (10 min)

- Quick pivot: "How would we model a **/machines** endpoint?"
- Discussion of real-world considerations

Module 6: Security & Authentication (25 minutes)

- **Security Fundamentals** (15 min)
    - API Keys vs. JWT tokens
    - HTTPS in production
    - Input validation with Pydantic
    - Rate limiting considerations
- **FastAPI Security Features** (10 min)
    - Built-in authentication
    - Dependency injection for security



Module 7: API Evolution & Best Practices (20 minutes)

- **Versioning Strategies** (10 min)
    - URL versioning: **/v1/books**
    - Header versioning
    - When and how to deprecate
- **Documentation & OpenAPI** (10 min)
    - FastAPI's automatic docs
    - Customizing OpenAPI specs
    - Why documentation matters for integration

Module 8: Manufacturing Applications & Next Steps (10 minutes)

- **Industrial Use Cases Review**

    - Machine data APIs
    - Production monitoring endpoints
    - Integration with OPC-UA/MQTT data
- **Bridge to MCP Server Workshop**
    - How REST principles apply to Model Context Protocol
    - Preview of future integration patterns

Closing & Q&A (5 minutes)

- Key takeaways checklist
- Resources for continued learning
- Next workshop preview

Workshop Materials Provided:

1. VS Code setup guide with FastAPI
2. Complete code examples and templates
3. Postman collection for testing APIs
4. Manufacturing-specific API design patterns document
5. Security checklist for industrial environments

Success Metrics:

Participants can:

✅ Explain REST principles and benefits
✅ Design resource-based URLs following conventions
✅ Choose appropriate HTTP methods and status codes
✅ Build a basic CRUD API with FastAPI
✅ Identify security considerations for their environment
✅ Plan API versioning strategy