5-Day API Course - 100/100 Implementation Guide

🎯 **Core Philosophy:** "Show, Practice, Apply, Scale" **80% Hands-on, 20% Theory + Real-World Production Context**

DAY 1: Foundation & Design - "Build the Blueprint"

Focus: Domain-driven API architecture with production mindset

10:00-10:15 AM: Intros & Reality Check (15 min)

- 🎯 **Activity:** Everyone shares their worst API integration experience (2 min each)
- Environment verification + mindset setting: "We're building for production"

10:15-10:45 AM: Module 1 - API Experience & Types (30 min)

- 🎯 **API Treasure Hunt** (10 min): 5 public APIs, find the good and ugly
- 💬 **Real-World Discussion** (10 min): "What makes an API production-ready?"
- 🎯 **API Classification Challenge** (10 min): REST vs GraphQL vs gRPC - when to use what?

10:45-11:00 AM: Break

11:00-12:00 PM: Module 2A - Domain-Driven TaskFlow Architecture (1 hour)

- **Your Demo First** (10 min): Complete TaskFlow API walkthrough
- 💬 **DDD Fundamentals** (15 min): "How do you identify business domains?"
- 🎯 **Domain Modeling Lab** (20 min): Map TaskFlow domains and boundaries
- 💬 **Microservices Reality** (15 min): "How do domain boundaries influence deployment?"

12:00-1:00 PM: Lunch

1:00-1:45 PM: Module 2B - API Design + Documentation (45 min)

- 🎯 **Domain-to-REST Mapping** (15 min): Transform domains into URIs

- 💬 **Cross-Domain Operations** (10 min): "How do you handle operations that span domains?"
- 🎯 **OpenAPI Contract Design** (10 min): Write specs that reflect domain language
- 💬 **Documentation Standards** (10 min): "How do you keep docs synchronized with code?"

1:45-2:30 PM: Module 2C - Error Handling & Edge Cases (45 min)

- 🎯 **Status Code Scenarios** (15 min): 15 real scenarios, choose correct codes
- 💬 **Production Error Handling** (15 min): "How do you handle cascading failures?"
- 🎯 **Error Response Design** (15 min): Design professional error responses

2:30-2:45 PM: Break

2:45-3:30 PM: Module 2D - gRPC vs REST Deep Dive (45 min)

- **Your Demo** (15 min): Live gRPC service comparison with REST
- 💬 **Internal vs External APIs** (15 min): "When do you choose gRPC over REST?"
- 🎯 **Protocol Buffer Design** (15 min): Design .proto for TaskFlow notifications

3:30-4:00 PM: NEW Module 2E - Integration Reality Check (30 min)

- 🎯 **Legacy API Challenge** (15 min): Given ugly legacy responses, design clean facades
- 💬 **Integration War Stories** (15 min): "How do you handle unreliable external APIs?"

Day 1 Outcome: Complete API architecture blueprint ready for production

---

DAY 2: Implementation - "Code Like Production"

Focus: Domain-driven FastAPI with production patterns

10:00-10:30 AM: Take-home Review + Production Setup (30 min)

- 🎯 **Structure Showcase:** Everyone presents their domain structure
- **Your Demo:** Production-grade project organization

10:30-10:45 AM: Break

10:45 AM-12:00 PM: Module 3A - Domain-Driven FastAPI (1h 15min)

- **Your Demo** (10 min): Complete domain-layered structure

- 💬 **Architecture Discussion** (10 min): "How do you separate business logic from API logic?"
- 🎯 **Code-Along: User Service** (45 min): Domain entities, repositories, services
- 🎯 **Mini-Challenge:** Add domain validation (10 min)

12:00-1:00 PM: Lunch

1:00-2:15 PM: Module 3B - Team Management Implementation (1h 15min)

- **Your Demo** (15 min): Team relationships and hierarchies
- 🎯 **Code-Along: Team Endpoints** (45 min): Build together
- 🎯 **Challenge:** Team member management with permissions (15 min)

2:15-2:30 PM: Break

2:30-3:15 PM: Module 3C - Independent Project Coding (45 min)

- 🎯 **Solo Challenge:** Build Project endpoints (30 min)
- 🎯 **Peer Code Review** (15 min): Share and critique solutions

3:15-4:00 PM: ENHANCED Module 3D - Production Testing (45 min)

- **Your Demo** (10 min): Complete test suite with domain testing
- 💬 **Testing Strategy** (10 min): "How do you test external dependencies?"
- 🎯 **Testing Lab** (15 min): Write comprehensive endpoint tests
- 🎯 **Contract Testing** (10 min): Introduction to API contract validation

Day 2 Outcome: Working TaskFlow API with bulletproof testing

DAY 3: Security - "Lock It Down"

Focus: Enterprise security with incident response

10:00-10:30 AM: Security Audit + Threat Modeling (30 min)

- 🎯 **Vulnerability Hunt:** Find security holes in yesterday's code
- 💬 **Threat Modeling:** "What are the biggest API security risks?"

10:30-10:45 AM: Break

10:45 AM-12:00 PM: Module 5A - Authentication Implementation (1h 15min)

- **Your Demo** (20 min): Complete JWT auth system
- 🎯 **Code-Along: Auth System** (45 min): Implement together
- 🎯 **Security Challenge:** Password policies and MFA (10 min)

12:00-1:00 PM: Lunch

1:00-2:15 PM: Module 5B - Authorization + Enterprise Patterns (1h 15min)

- **Your Demo** (10 min): Role-based access control
- 💬 **Enterprise Authorization** (15 min): "How do you handle permissions at scale?"
- 🎯 **Permission Matrix Design** (20 min): Design TaskFlow permissions
- 🎯 **Code-Along: RBAC Implementation** (30 min)

2:15-2:30 PM: Break

2:30-3:15 PM: Module 5C - Security Controls (45 min)

- 🎯 **Rate Limiting Lab** (15 min): Implement smart rate limiting
- 🎯 **Input Validation Lab** (15 min): Bulletproof validation
- 🎯 **Security Headers Lab** (15 min): CORS, CSP, HSTS configuration

3:15-4:00 PM: NEW Module 5D - Security Incident Simulation (45 min)

- 🎯 **Live Security Drill** (25 min): Simulated API breach - identify, contain, fix
- 💬 **Incident Response** (10 min): "What's your security incident process?"
- 🎯 **Compliance Quick Check** (10 min): GDPR/SOC2 requirements for APIs

Day 3 Outcome: Enterprise-grade secured API with incident response skills

---

DAY 4: Operations & Scale - "Make It Production-Ready"

Focus: Deployment, monitoring, and performance optimization

10:00-10:30 AM: Monitoring + Health Checks (30 min)

- **Your Demo:** Professional logging and monitoring setup
- 🎯 **Health Check Lab:** Add comprehensive health endpoints

10:30-10:45 AM: Break

10:45 AM-12:00 PM: ENHANCED Module 6A - Containerization Workshop (1h 15min)

- **Your Demo** (15 min): Multi-stage Dockerfile + docker-compose
- 🎯 **Docker Lab** (45 min): Containerize TaskFlow with optimization
- 🎯 **Docker Challenge:** Multi-environment configuration (15 min)

12:00-1:00 PM: Lunch

1:00-2:00 PM: NEW Module 4B - Performance & Database Optimization (1 hour)

- 🎯 **N+1 Query Hunt** (20 min): Find and fix performance issues
- 🎯 **Caching Strategy Lab** (25 min): Implement Redis caching
- 💬 **Scaling Discussion** (15 min): "How do you handle database bottlenecks?"

2:00-3:15 PM: ENHANCED Module 6B - Production Deployment (1h 15min)

- **Your Demo** (15 min): Complete production deployment pipeline
- 🎯 **Deployment Marathon** (45 min): Deploy to cloud with load balancing
- 🎯 **API Gateway Setup** (15 min): Configure rate limiting and routing

3:15-3:30 PM: Break

3:30-4:00 PM: Module 6C - CI/CD + Monitoring (30 min)

- **Your Demo:** GitHub Actions workflow with testing
- 🎯 **Pipeline Setup:** Basic CI/CD with automated testing
- 🎯 **Monitoring Dashboard:** Set up API analytics

Day 4 Outcome: Production-deployed API with full operational monitoring

---

DAY 5: Advanced Features & Business Impact - "Think Like a Lead"

Focus: Advanced patterns and business-driven API development

10:00-10:30 AM: Performance Showcase + Advanced Patterns (30 min)

- **Performance Review:** Showcase optimization results
- **Your Demo:** Advanced caching and async patterns

10:30-10:45 AM: Break

10:45 AM-12:00 PM: Module 2 (gRPC) - Internal Services (1h 15min)

- **Your Demo** (20 min): Complete gRPC service with streaming
- 💬 **gRPC Production Discussion** (15 min): "How do you handle gRPC service discovery?"
- 🎯 **gRPC Lab** (30 min): Implement real-time notification service
- 💬 **Protocol Comparison** (10 min): When to use each approach

12:00-1:00 PM: Lunch

1:00-2:00 PM: Module 7A - Architecture & Case Studies (1 hour)

- 🎯 **Architecture Analysis** (20 min): Compare TaskFlow to Slack/Stripe APIs
- 💬 **Production Lessons** (20 min): "What would you change in TaskFlow for 1M users?"
- 🎯 **Scaling Design Challenge** (20 min): Design TaskFlow for enterprise scale

2:00-2:15 PM: Break

2:15-3:00 PM: NEW Module 7B - API Business Impact (45 min)

- 🎯 **API Monetization Workshop** (20 min): Design pricing tiers for TaskFlow
- 💬 **Business Metrics Discussion** (15 min): "How do you measure API success?"
- 🎯 **Contract Evolution Challenge** (10 min): Make breaking changes safely

3:00-3:30 PM: Module 7C - Portfolio Enhancement (30 min)

- 🎯 **Final Feature Sprint** (20 min): Add one advanced feature
- 🎯 **Documentation Polish** (10 min): Professional README creation

3:30-4:00 PM: Demo Day + Industry Insights (30 min)

- 🎯 **Final Presentations** (20 min): Everyone demos their complete system
- 💬 **Industry Roundtable** (10 min): "What would you build differently now?"

Day 5 Outcome: Portfolio-worthy API with business understanding and leadership perspective

---

🎯 Your Prepared Assets (Achievable & Realistic)

Day 1 Prep:

- **TaskFlow OpenAPI spec** with domain language examples
- **Domain modeling templates** (printable worksheets)
- **gRPC .proto examples** for common patterns
- **Legacy API response samples** for integration challenge

Day 2 Prep:

- **FastAPI project template** with domain structure
- **Working domain models** for User, Team, Project
- **Test suite examples** with different testing strategies
- **Code snippets** for common patterns

Day 3 Prep:

- **Complete JWT auth system** (copy-paste ready)
- **RBAC decorator examples** with different permission levels
- **Security breach scenario** (1-page description with clues)
- **Security checklist** (printable reference)

Day 4 Prep:

- **Production Dockerfile** with multi-stage build
- **docker-compose.yml** with Redis, PostgreSQL
- **Deployment scripts** for major cloud providers
- **N+1 query examples** in TaskFlow code

Day 5 Prep:

- **gRPC notification service** (working implementation)
- **API comparison matrix** (Slack, Stripe, GitHub analysis)
- **Monetization examples** from real APIs
- **Scaling architecture diagrams**

🔧 Tools & Setup Requirements

Required Tools:

- VS Code, Python 3.8+, Node.js (for demos)
- Postman/Insomnia, Docker Desktop
- Redis (Docker), PostgreSQL (Docker)
- Git, GitHub account

Optional but Recommended:

- Railway/Render account for deployment
- Redis Cloud free tier
- Postman Team workspace

📊 What Makes This 100/100

Technical Excellence (40 points):

- Domain-driven architecture ✓
- Production-grade patterns ✓
- Comprehensive testing ✓
- Security best practices ✓

Real-World Readiness (30 points):

- Legacy integration challenges ✓
- Security incident response ✓
- Performance optimization ✓
- Production deployment ✓

Business Context (20 points):

- API monetization understanding ✓
- Scaling decision frameworks ✓
- Cross-functional collaboration ✓
- Contract evolution strategies ✓

Teaching Excellence (10 points):

- 80/20 hands-on ratio ✓
- Progressive complexity ✓
- Peer learning opportunities ✓
- Portfolio-building focus ✓

🎯 Success Metrics

Day 1: Students can design API architecture for any domain **Day 2:** Students can implement production-grade FastAPI services **Day 3:** Students can secure APIs against common threats **Day 4:** Students can deploy and monitor APIs in production **Day 5:** Students can make architectural decisions with business context

Final Outcome: Students leave with a complete, deployed, portfolio-worthy API and the confidence to lead API projects in their organizations.

Key Implementation Note: Every addition is achievable within your existing time structure. The "new" modules replace or extend existing content without adding time pressure. Focus on the security simulation and API business workshop - these give maximum impact for minimal prep effort.