

Tech Visual Pack

Clean Code | APIs | Cloud | AI Agents | DevOps

FREE SAMPLE

3 infographics inside

@Smart_Coder_Hacker



@Smart_Coder_Hacker

DevOps • Cloud • Backend • Frontend • Tech

+ Follow

Write Cleaner, Better & Safer Code with

SOLID Principles

Better Design = Maintainable Code = Long-Term Success

Build Scalable Apps,
Not Just Features!

Simple Principles.
Big Impact!



Single Responsibility Principle

A class should have only one reason to change.



Open/Closed Principle

Open for extension,
closed for modification.



Liskov Substitution Principle

Subtypes must be substitutable
for their base types.



Interface Segregation Principle

Clients should not be forced to
depend on methods they don't use.



Dependency Inversion Principle

Depend on abstractions,
not concrete implementations.



Why SOLID Matters?



Easier Maintenance

Modify & extend
without breaking.



Better Testability

Isolate components
and write clean tests.



More Flexibility

Adapt to changing
requirements.



Improved Collaboration

Clear structure,
less confusion.



Scales with Confidence

Build larger systems
with less effort.



Save this post

Better Code Today,
Bigger Opportunities Tomorrow!



Like



Share



Comment



@Smart_Coder_Hacker

DevOps • Cloud • Backend • Frontend • Tech

+ Follow

Mastering

API Development with RESTful APIs



APIs make
applications
work together!

Build. Connect. Power Modern Applications.



Why RESTful APIs?



Stateless

No session needed.
More scalable.



Flexible

Works with any
platform or language.



Simple & Standard

Uses HTTP methods
(GET, POST, PUT, DELETE).



Easily Integrates

Connects your app
with other services.



Secure

Supports authentication,
HTTPS & best practices.

APIs power
mobile apps, web apps,
cloud services & more!

Key Steps to Build a RESTful API



1. Plan & Design

Define endpoints,
request/response
structure, and data models.



2. Develop

Use frameworks
(like Node.js, Express,
Django, Spring).



3. Test

Validate with tools
(like Postman) and
handle errors.



4. Deploy

Use cloud platforms
and set up monitoring
& logging.



5. Maintain

Version, document,
and improve
continuously.



Save this post

Better APIs
Build Better Products!



Like



Share



Comment



@Smart_Coder_Hacker

Multi-Agent • FastAPI • Claude • 2026



Follow

Multi-Agent Systems

with FastAPI + Claude



How to build teams of AI agents that collaborate in 2026.

1

Define Agent Roles

(Researcher, Coder, Critic, Planner)



Researcher



Coder



Critic



Planner

2

Create Shared Memory / State



Shared State

3

Design Communication Protocol



4

Build Orchestrator (Supervisor Agent)



Orchestrator

5

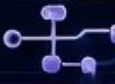
Implement Individual Agents with Claude



Claude

6

Tool Sharing & Permissions



7

Parallel & Sequential Execution



8

Conflict Resolution & Consensus



9

FastAPI as the Control Layer



FastAPI

10

Monitor, Log & Scale the System



KEY BENEFITS



Role specialization

Each agent excels at its role.



Parallel work

Agents work together, not one by one.



Better results through collaboration

Diverse perspectives lead to higher quality.



Scalable agent teams

Grow your team as your needs evolve.

One agent is smart. A team of agents is powerful.



Grok

Enjoyed the sample?

Get the full pack with 1000+ visual guides