

Building a REST API with Python 3

Starting the Project



Reindert-Jan Ekker

@rjekker | <https://nl.linkedin.com/in/rjekker>



Version Check



This version was created by using:

- Python 3.11
- Poetry 1.4
- FastApi 0.103
- sqlalchemy 2.0



Version Check



This course is 100% applicable to:

- Python 3.10+
- Poetry 1.4+
- Fastapi 0.103+
- SQLAlchemy 2.x



Overview



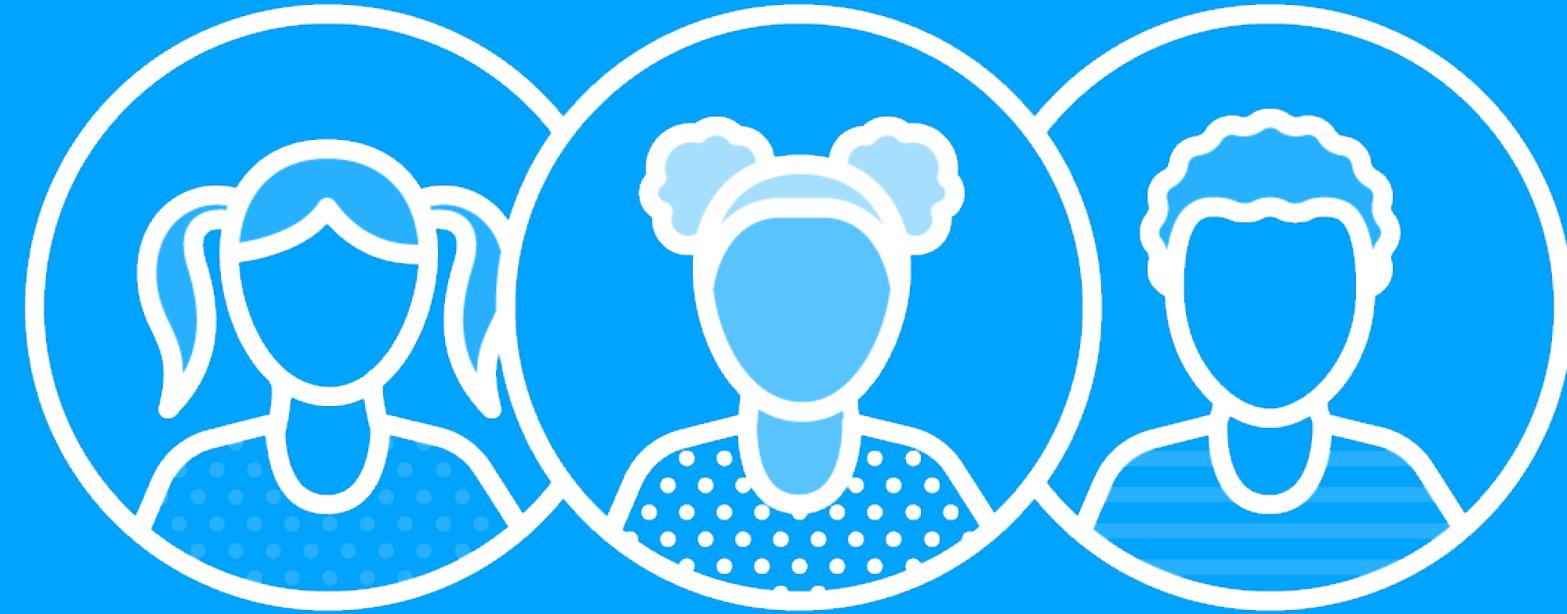
Project: building a REST API with Python 3

FastAPI, sqlalchemy

Best practices

Aim to approach a real world project



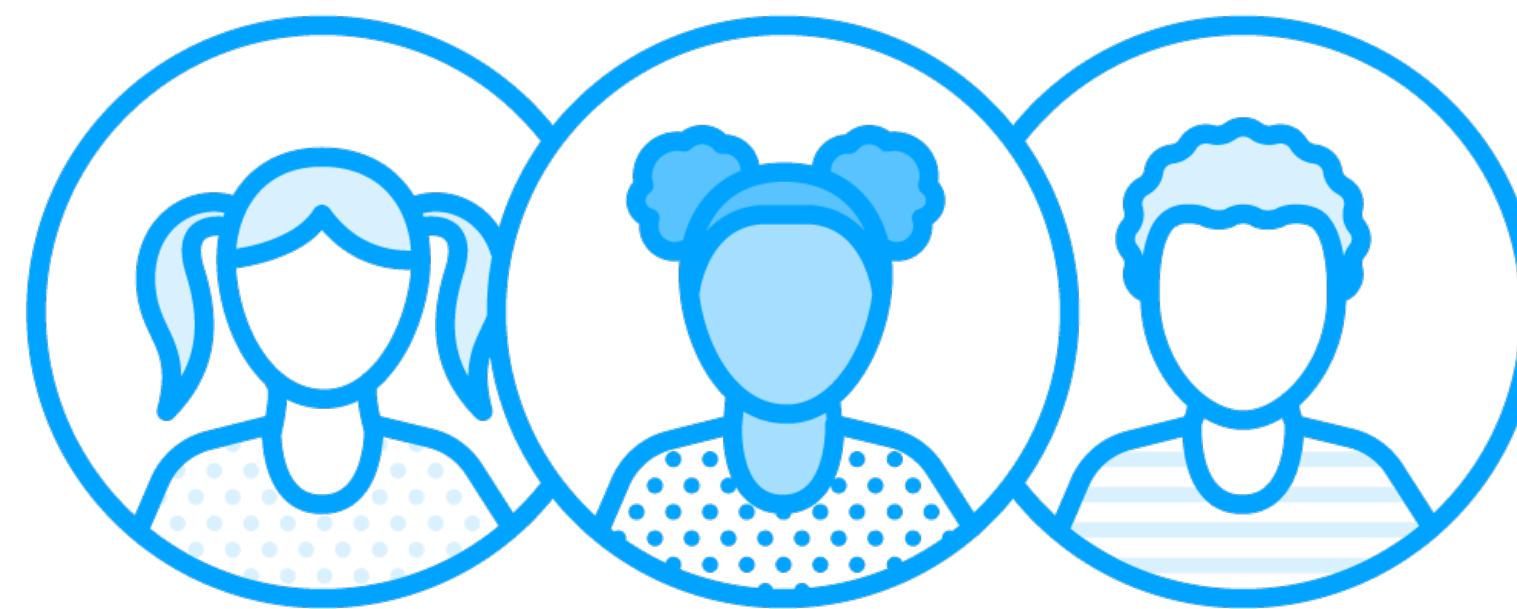


Audience and Prerequisites

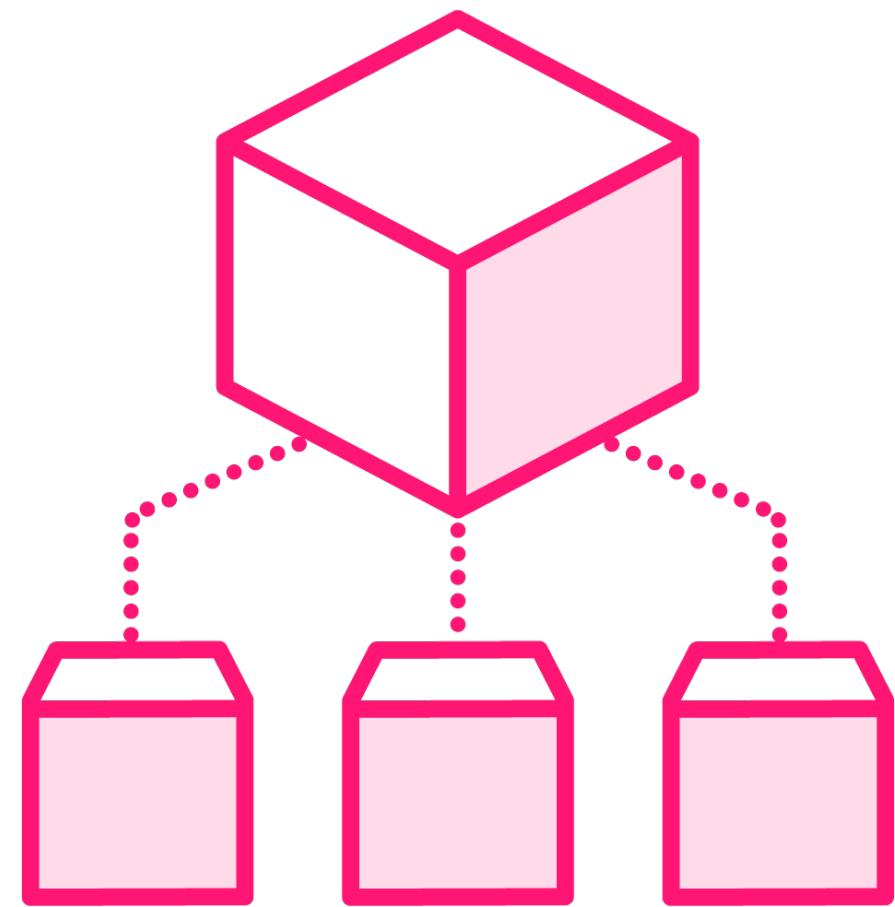
**Who should watch this course?
What should I already know?**



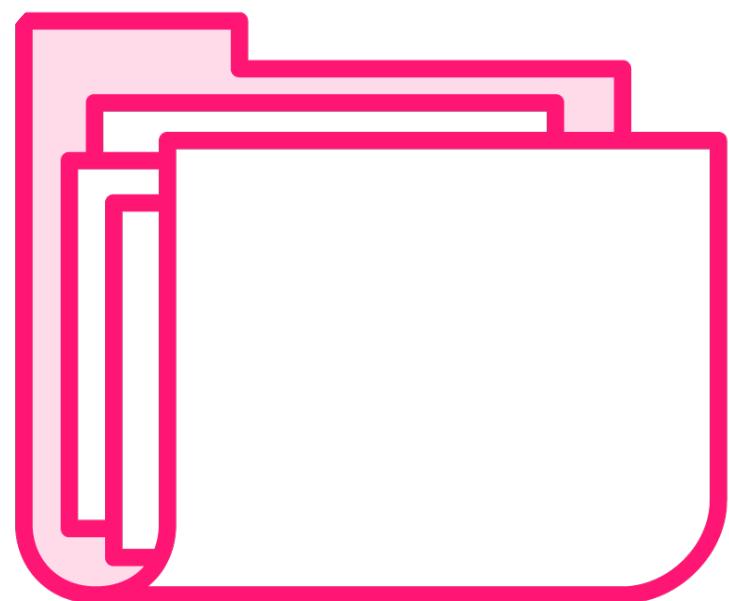
Intermediate Python
**Familiar with the following
topics**



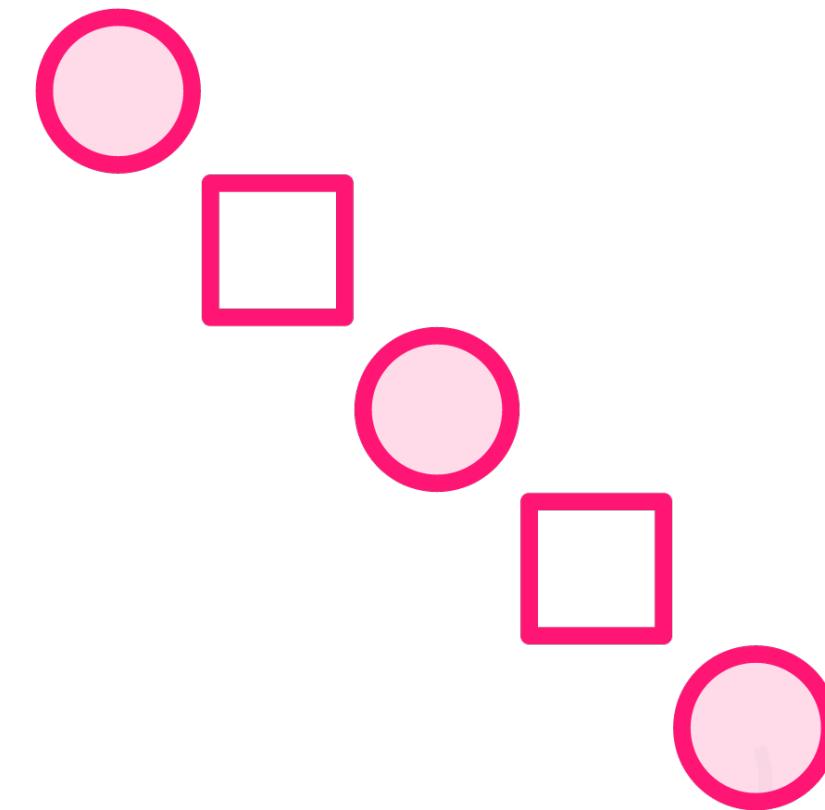
Topics Applied in the Project: Python



**OOP/Classes
Inheritance
Properties**



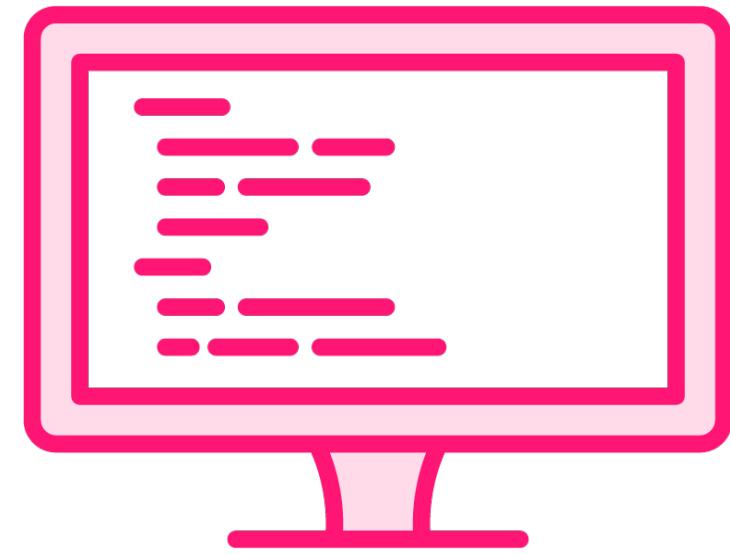
Files/Pathlib



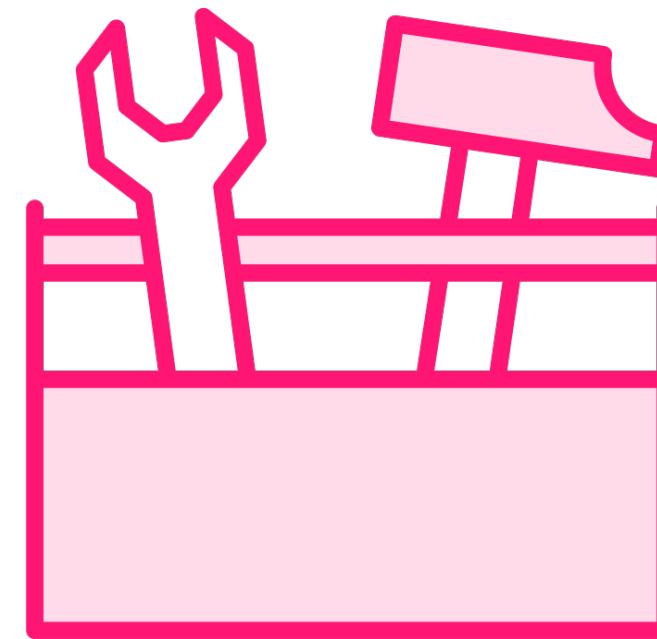
Regular Expressions



Topics Applied in the Project: Best Practices



Code Style
Pep 8/flake8
Type hints/mypy



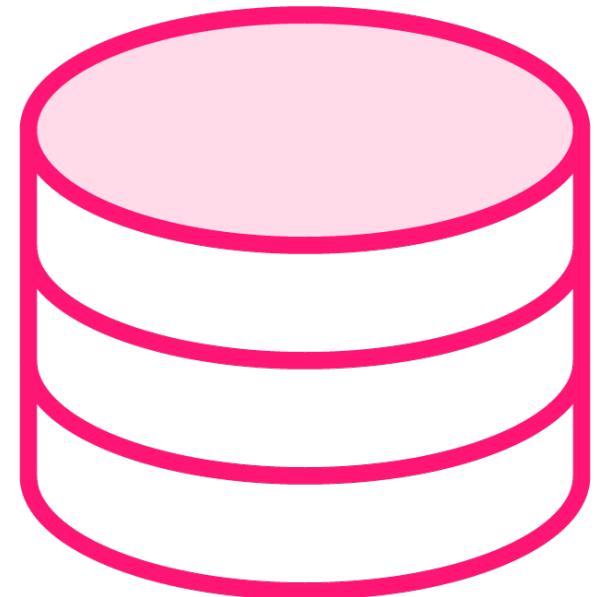
Environment
Poetry



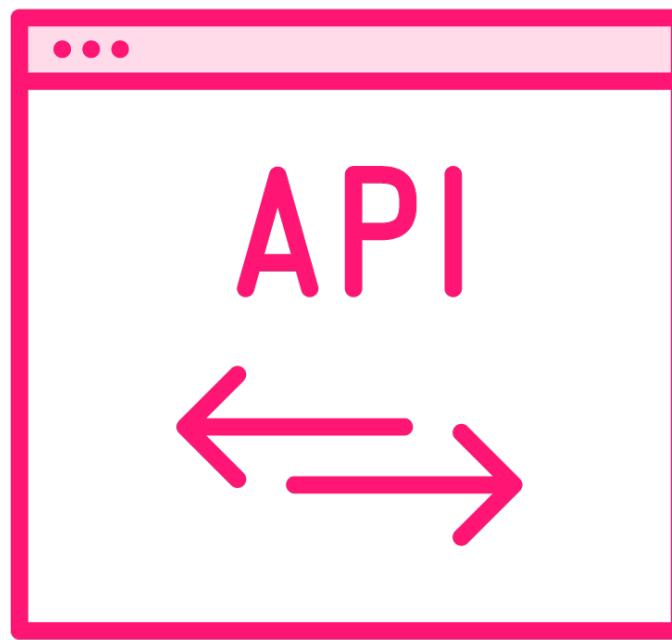
Unit Testing
Pytest



Topics Applied in the Project: Libraries



Database
Sqlalchemy



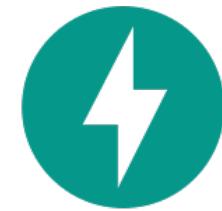
REST
client: requests/htpx



Unit Testing
Pytest



Also in the Project



FastAPI to create REST API



pyyaml to parse YAML data



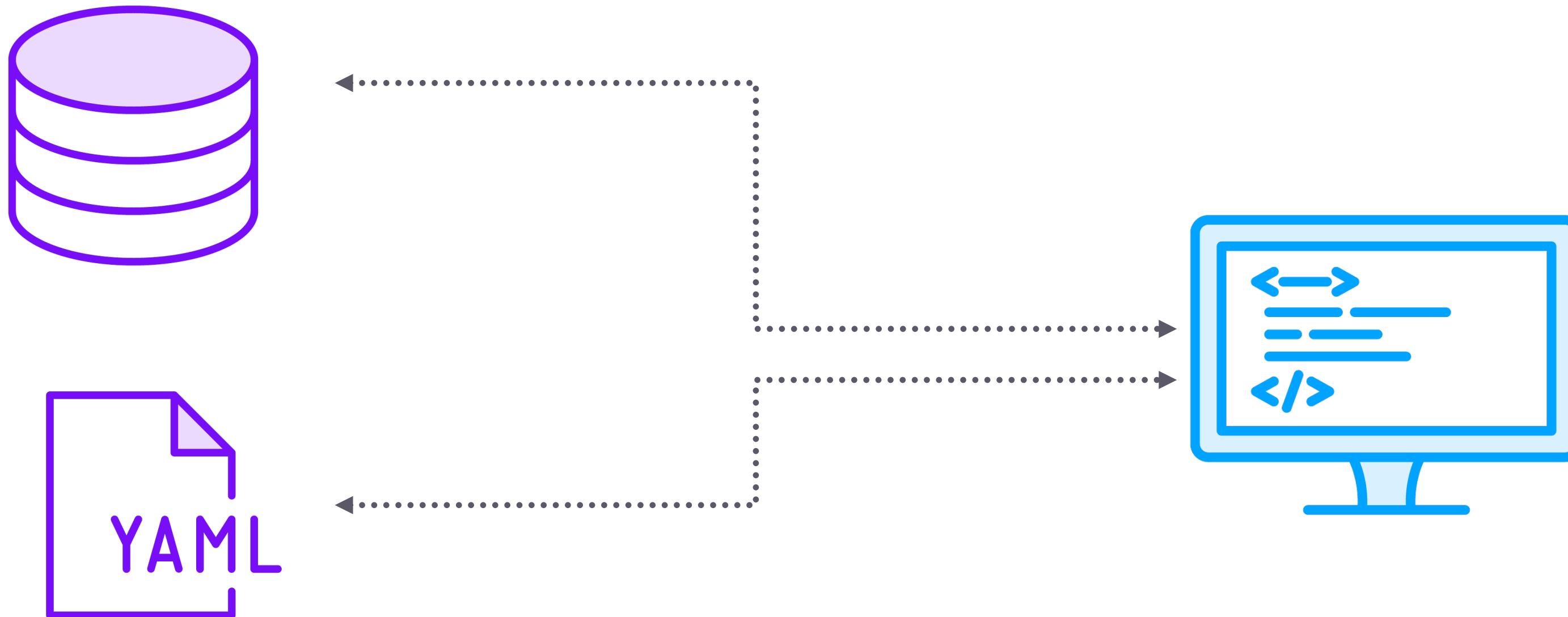
precommit framework to run linters



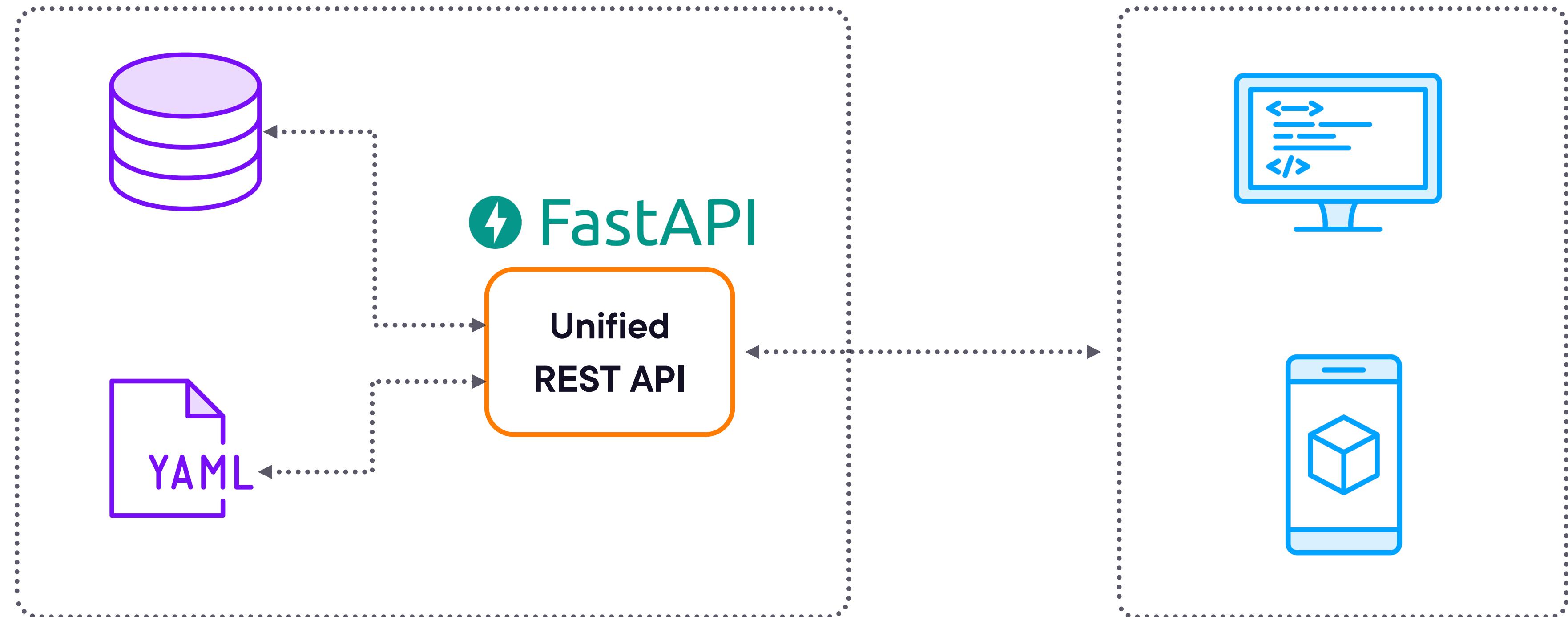
python-dotenv for project configuration



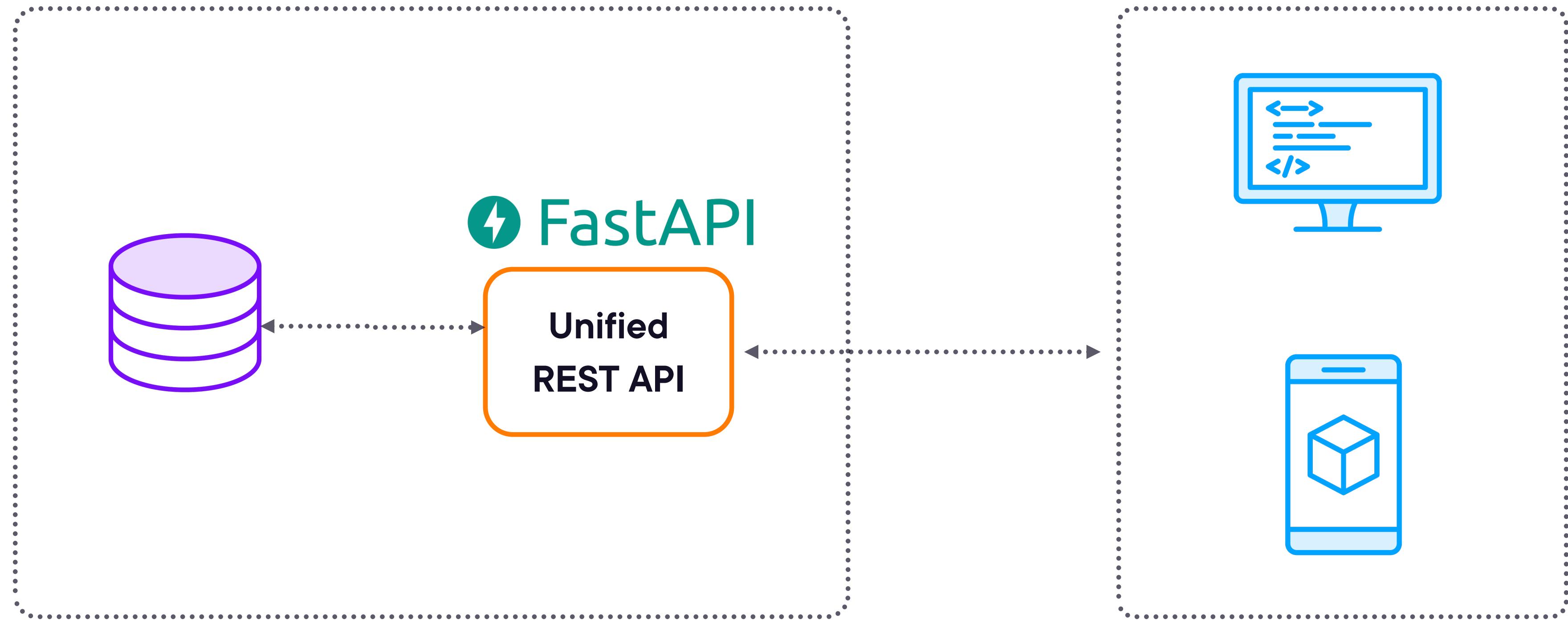
Current situation



Our project



Possible future?

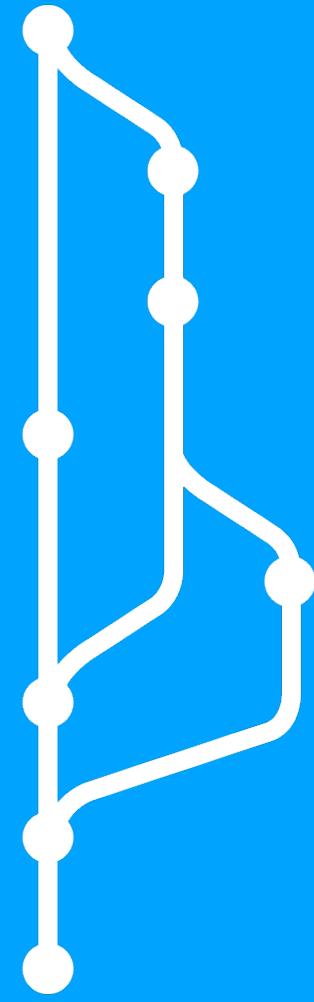


Demo



- Getting the project
- Install Poetry
- Run project
- Open in Pycharm and VS Code





<https://bit.ly/python3-rest-api>

Use this URL to clone or download the demo code.





Installing Poetry

<https://bit.ly/installing-poetry>



Summary



Project overview

Download/clone

poetry install

poetry run python runserver.py

Pre-commit framework



Up Next:

Creating a datamodel

