

Lesson 42 Summary: FastAPI

Data Science with Python – Key Concepts

Data Science Program

FastAPI for ML Deployment

FastAPI

Modern Python API
Auto docs (Swagger)
Type hints

REST Endpoints

GET: read data
POST: send data
`@app.get/post`

Pydantic

Data validation
Type checking
BaseModel class

ML Integration

Load model at start
Predict endpoint
Return JSON

Testing

TestClient
pytest integration
`/docs` for manual

Running

`uvicorn main:app`
`--reload` for dev
localhost:8000

Why FastAPI?

Key advantages:

- **Fast:** High performance, async support
- **Auto docs:** Swagger UI at /docs
- **Type hints:** Automatic validation

Production-ready with minimal code

HTTP methods:

- **GET:** Retrieve data (`@app.get`)
- **POST:** Send data (`@app.post`)
- **ML:** POST for predictions (JSON input/output)

RESTful design for ML inference APIs

Data validation:

- **BaseModel:** Define input/output schemas
- **Type checking:** Automatic validation
- **Errors:** Clear validation messages

Type-safe request/response handling

Pattern for ML APIs:

- **Startup:** Load model once at app start
- **Endpoint:** Accept features, return prediction
- **Response:** JSON with prediction results

Load once, predict many

Essential Commands:

| Task | Code |
|---------------|--|
| Create app | <code>app = FastAPI()</code> |
| GET endpoint | <code>@app.get("/path")</code> |
| POST endpoint | <code>@app.post("/predict")</code> |
| Run server | <code>uvicorn main:app --reload</code> |
| View docs | <code>localhost:8000/docs</code> |

FastAPI is the standard for ML REST APIs