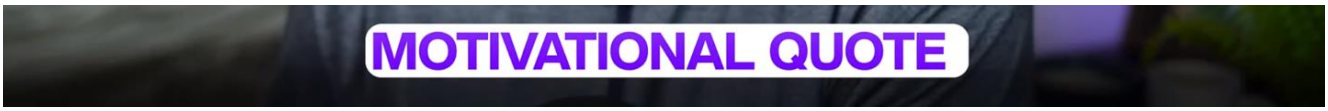
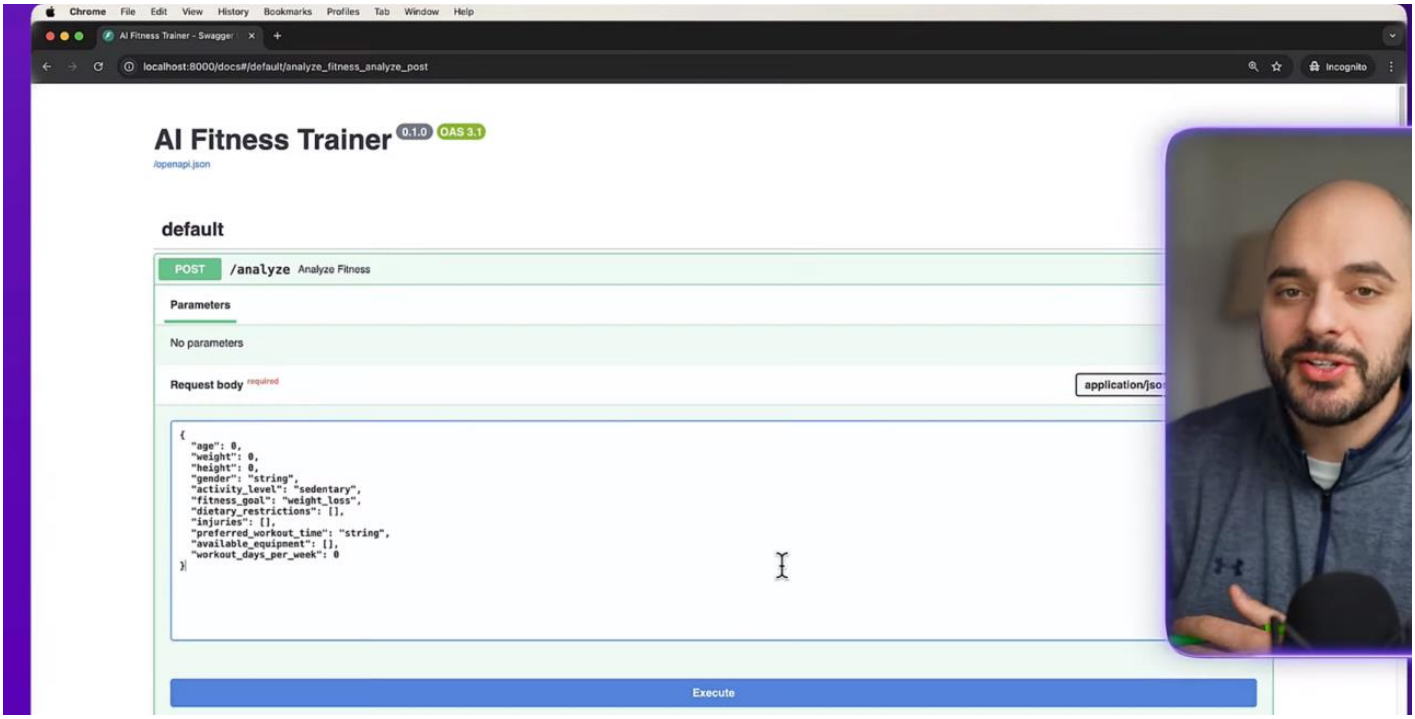
Build Your Own POWERFUL Multi-Agent AI (FastAPI & PydanticAI)

Eric Roby
23.6K subscribers

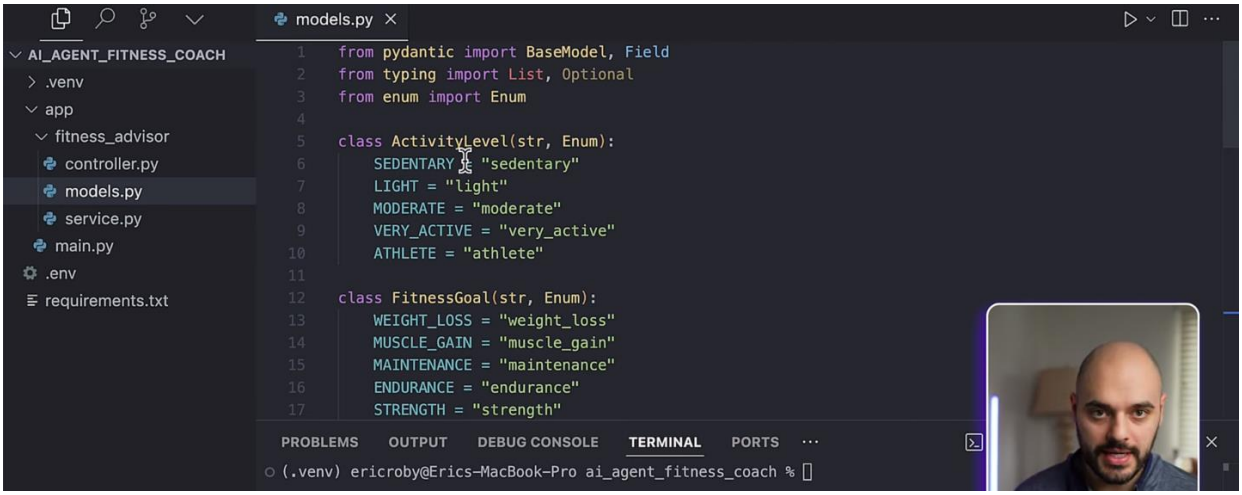
Join | Subscribe | 169 | Share | Ask | Download | ...

4,921 views Apr 10, 2025 #ai #python #codingwithroby
PydanticAI is really awesome. When you use FastAPI with PydanticAI you can create Multi AI Agents super easy that can make AI automate a lot of your daily life. This Python AI Agent video is focused on teaching you how to do this.





We have FastAPI and uvicorn installed already


models.py
AI_AGENT_FITNESS_COACH
.venv
app
fitness_advisor
controller.py
models.py
service.py
main.py
.env
requirements.txt
from pydantic import BaseModel, Field
from typing import List, Optional
from enum import Enum
class ActivityLevel(str, Enum):
SEDENTARY = "sedentary"
LIGHT = "light"
MODERATE = "moderate"
VERY_ACTIVE = "very_active"
ATHLETE = "athlete"
class FitnessGoal(str, Enum):
WEIGHT_LOSS = "weight_loss"
MUSCLE_GAIN = "muscle_gain"
MAINTENANCE = "maintenance"
ENDURANCE = "endurance"
STRENGTH = "strength"
PROBLEMS
OUTPUT
DEBUG CONSOLE
TERMINAL
PORTS
(.venv) ericroby@Erics-MacBook-Pro ai_agent_fitness_coach %

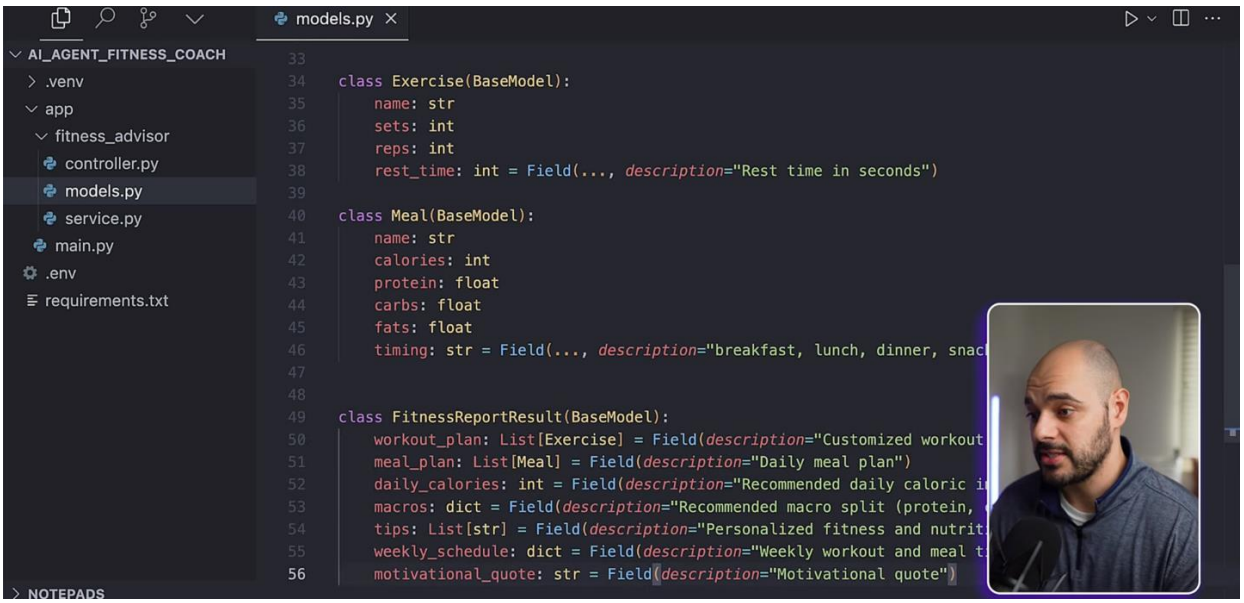

models.py
AI_AGENT_FITNESS_COACH
.venv
app
fitness_advisor
controller.py
models.py
service.py
main.py
.env
requirements.txt
SEDENTARY = "sedentary"
LIGHT = "light"
MODERATE = "moderate"
VERY_ACTIVE = "very_active"
ATHLETE = "athlete"
class FitnessGoal(str, Enum):
WEIGHT_LOSS = "weight_loss"
MUSCLE_GAIN = "muscle_gain"
MAINTENANCE = "maintenance"
ENDURANCE = "endurance"
STRENGTH = "strength"
class Exercise(BaseModel):
name: str
sets: int


models.py
AI_AGENT_FITNESS_COACH
.venv
app
fitness_advisor
controller.py
models.py
service.py
main.py
.env
requirements.txt
MUSCLE_GAIN = "muscle_gain"
MAINTENANCE = "maintenance"
ENDURANCE = "endurance"
STRENGTH = "strength"
class Exercise(BaseModel):
name: str
sets: int
reps: int
rest_time: int = Field(..., description="Rest time in seconds")
class Meal(BaseModel):
name: str
calories: int
protein: float
carbs: float
fats: float

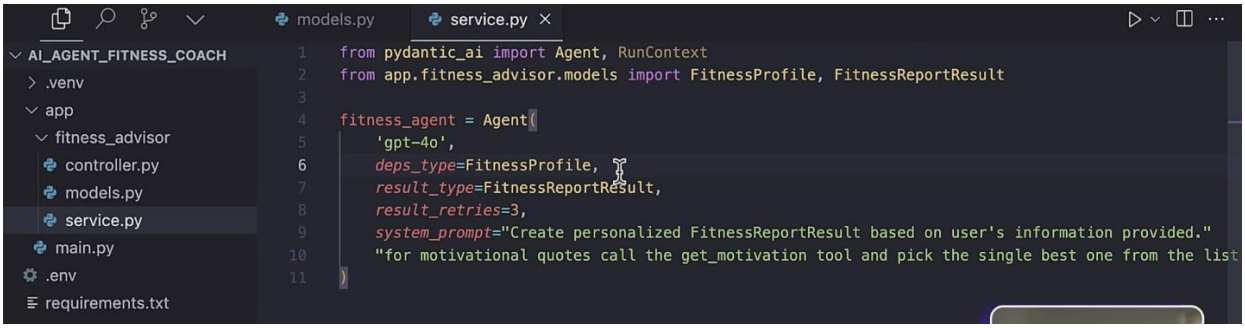

models.py
AI_AGENT_FITNESS_COACH
.venv
app
fitness_advisor
controller.py
models.py
service.py
main.py
.env
requirements.txt
name: str
calories: int
protein: float
carbs: float
fats: float
timing: str = Field(..., description="breakfast, lunch, dinner, snack")
class FitnessProfile(BaseModel):
age: int
weight: float
height: float
gender: str
activity_level: ActivityLevel
fitness_goal: FitnessGoal
dietary_restrictions: List[str] = []
injuries: List[str] = []


models.py
AI_AGENT_FITNESS_COACH
.venv
app
fitness_advisor
controller.py
models.py
service.py
main.py
.env
requirements.txt
STRENGTH = "strength"
class FitnessProfile(BaseModel):
age: int
weight: float
height: float
gender: str
activity_level: ActivityLevel
fitness_goal: FitnessGoal
dietary_restrictions: List[str] = []
injuries: List[str] = []
preferred_workout_time: str
available_equipment: List[str] = []
workout_days_per_week: int


models.py
AI_AGENT_FITNESS_COACH
injuries: List[str] = []
preferred_workout_time: str
available_equipment: List[str] = []
workout_days_per_week: int
class Exercise(BaseModel):
name: str
sets: int
reps: int
rest_time: int = Field(..., description="Rest time in seconds")
class Meal(BaseModel):
name: str
calories: int

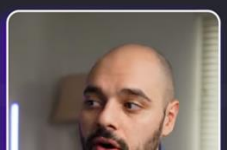

models.py
AI_AGENT_FITNESS_COACH
class Exercise(BaseModel):
name: str
sets: int
reps: int
rest_time: int = Field(..., description="Rest time in seconds")
class Meal(BaseModel):
name: str
calories: int
protein: float
carbs: float
fats: float
class FitnessReportResult(BaseModel):
workout_plan: List[Exercise] = Field(description="Customized workout
meal_plan: List[Meal] = Field(description="Daily meal plan")
daily_calories: int = Field(description="Recommended daily caloric i
macros: dict = Field(description="Recommended macro split (protein,
tips: List[str] = Field(description="Personalized fitness and nutrit
weekly_schedule: dict = Field(description="Weekly workout and meal t
motivational_quote: str = Field(description="Motivational quote")
NOTEPADS


PROBLEMS 1
OUTPUT
DEBUG CONSOLE
TERMINAL
PORTS
zsh
(.venv) ericroby@Erics-MacBook-Pro ai_agent_fitness_coach % pip install pydantic-ai


models.py
service.py
AI_AGENT_FITNESS_COACH
from pydantic_ai import Agent, RunContext
from app.fitness_advisor.models import FitnessProfile, FitnessReportResult
fitness_agent = Agent(
'gpt-4o',
deps_type=FitnessProfile,
result_type=FitnessReportResult,
result_retries=3,
system_prompt="Create personalized FitnessReportResult based on user's information provided."
"for motivational quotes call the get_motivation tool and pick the single best one from the list
)

models.py service.py

```python
from pydantic_ai import Agent, RunContext
from app.fitness_advisor.models import FitnessProfile, FitnessReportResult

fitness_agent = Agent(
    'gpt-4o',
    deps_type=FitnessProfile,
    result_type=FitnessReportResult,
    result_retries=3,
    system_prompt="Create personalized FitnessReportResult based on user's information provided."
    "for motivational quotes call the get_motivation tool and pick the single best one from the list you receive."
)

motivational_agent = Agent(
    'gpt-4o',
    result_type=list[str],
    system_prompt="Give motivational quotes based on the user's fitness goals and current status.",
)
```



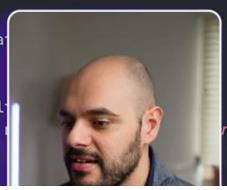
models.py service.py

```python
    result_retries=3,
    system_prompt="Create personalized FitnessReportResult based on user's information provided."
    "for motivational quotes call the get_motivation tool and pick the single best one from the list you receive."
)

motivational_agent = Agent(
    'gpt-4o',
    result_type=list[str],
    system_prompt="Give motivational quotes based on the user's fitness goals and current status.",
)

@fitness_agent.system_prompt
async def add_user_fitness_data(ctx: RunContext[FitnessProfile]) -> str:
    fitness_data = ctx.deps
    return f"User fitness profile and goals: {fitness_data!r}"


@fitness_agent.tool
async def get_motivation(ctx: RunContext) -> list[str]:
    return await motivational_agent.run(
        f"Please generate 5 motivational quotes about working out and eating healthy.")
```

AI_AGENT_FITNESS_COACH
- .venv
- app
  - fitness_advisor
    - controller.py
    - models.py
    - service.py
  - main.py
- .env
- requirements.txt

models.py service.py controller.py

```python
    system_prompt="Give motivational quotes based on the user's fitness goals and current status.",
)

@fitness_agent.system_prompt
async def add_user_fitness_data(ctx: RunContext[FitnessProfile]) -> str:
    fitness_data = ctx.deps
    return f"User fitness profile and goals: {fitness_data!r}"


@fitness_agent.tool
async def get_motivation(ctx: RunContext) -> list[str]:
    return await motivational_agent.run(
        f"Please generate 5 motivational quotes about working out and ea

async def analyze_profile(profile: FitnessProfile) -> FitnessReportResul
    result = await fitness_agent.run("Create a personalized fitness and
    return result.data
```

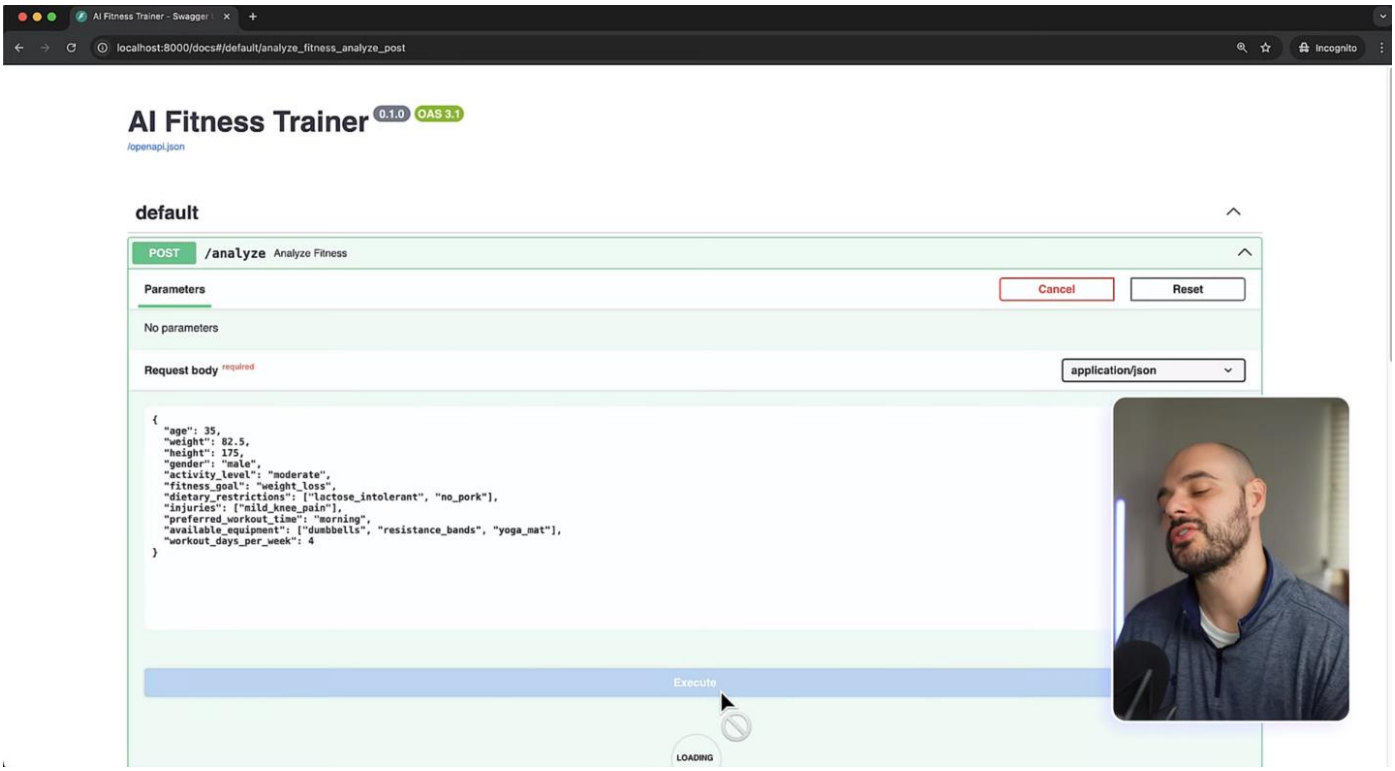


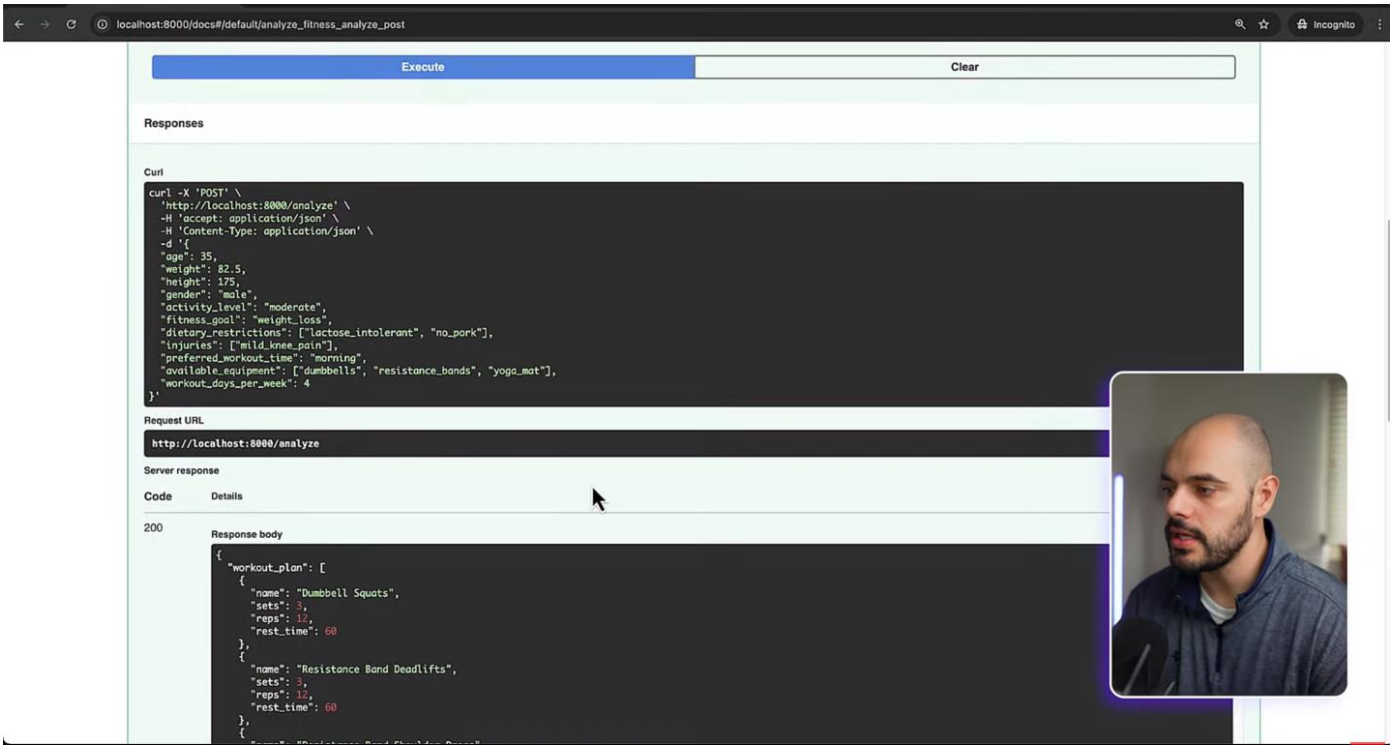
AI_AGENT_FITNESS_COACH
- .venv
- app
  - fitness_advisor
    - controller.py
    - models.py
    - service.py
  - main.py
- .env
- requirements.txt

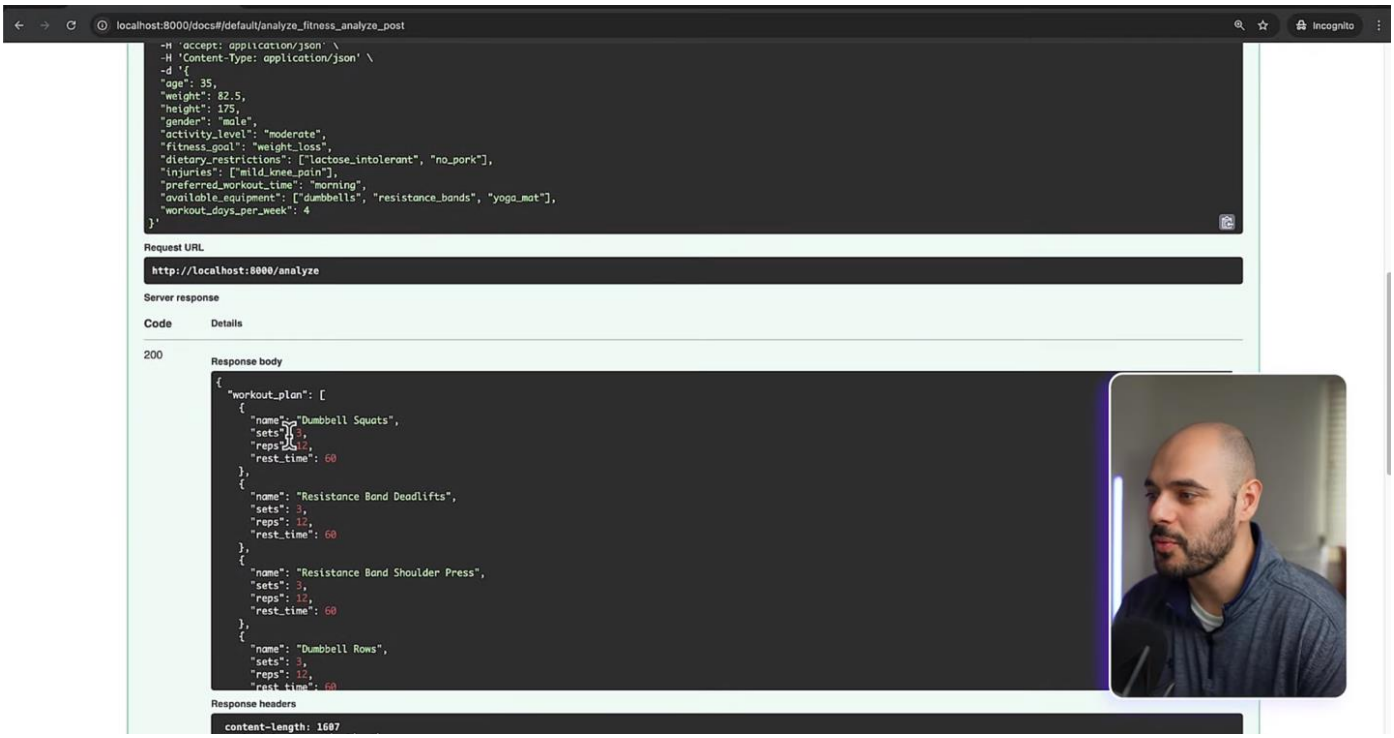
models.py service.py controller.py

```python
from fastapi import APIRouter
from app.fitness_advisor.models import FitnessProfile
from app.fitness_advisor.service import analyze_profile


router = APIRouter()


@router.post("/analyze")
async def analyze_fitness(fitness_profile: FitnessProfile):
    return await analyze_profile(fitness_profile)
```

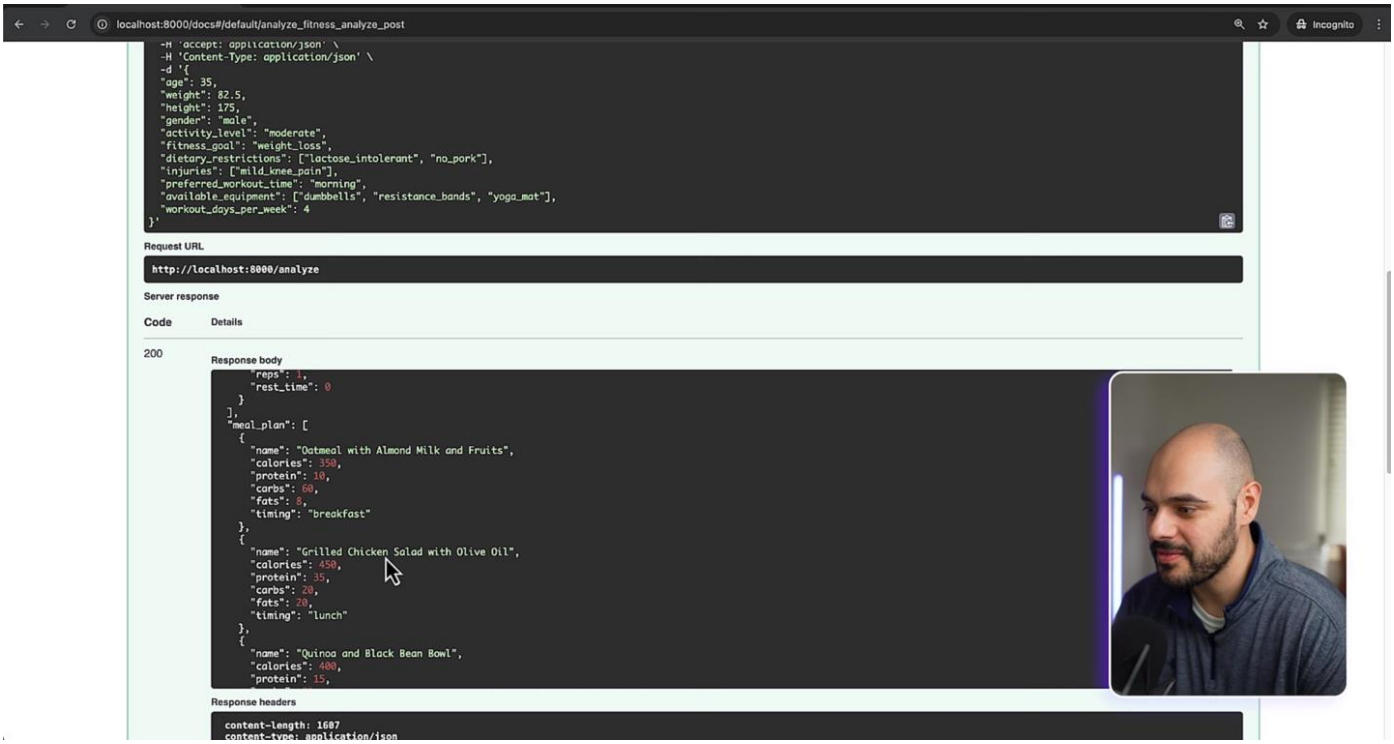

models.py
service.py
controller.py
AI_AGENT_FITNESS_COACH
.venv
app
fitness_advisor
controller.py
models.py
service.py
main.py
.env
requirements.txt
system_prompt="Give motivational quotes based on the user's fitness goals and current status.",
)
@fitness_agent.system_prompt
async def add_user_fitness_data(ctx: RunContext[FitnessProfile]) -> str:
fitness_data = ctx.deps
return f"User fitness profile and goals: {fitness_data!r}"
@fitness_agent.tool
async def get_motivation(ctx: RunContext) -> list[str]:
return await motivational_agent.run(
async def analyze_profile(profile: FitnessProfile) -> FitnessReportResult:
PROBLEMS
OUTPUT
DEBUG CONSOLE
TERMINAL
PORTS
POSTMAN CONSOLE
Python
(.venv) ericroby@Erics-MacBook-Pro ai_agent_fitness_coach % uvicorn app.main:app --reload
INFO: Will watch for changes in these directories: ['/Users/ericroby/Desktop/misc/YouTube/march/ai_agent_fitness_coach']
INFO: Uvicorn running on http://127.0.0.1:8000 (Press CTRL+C to quit)
INFO: Started reloader process [48204] using StatReload

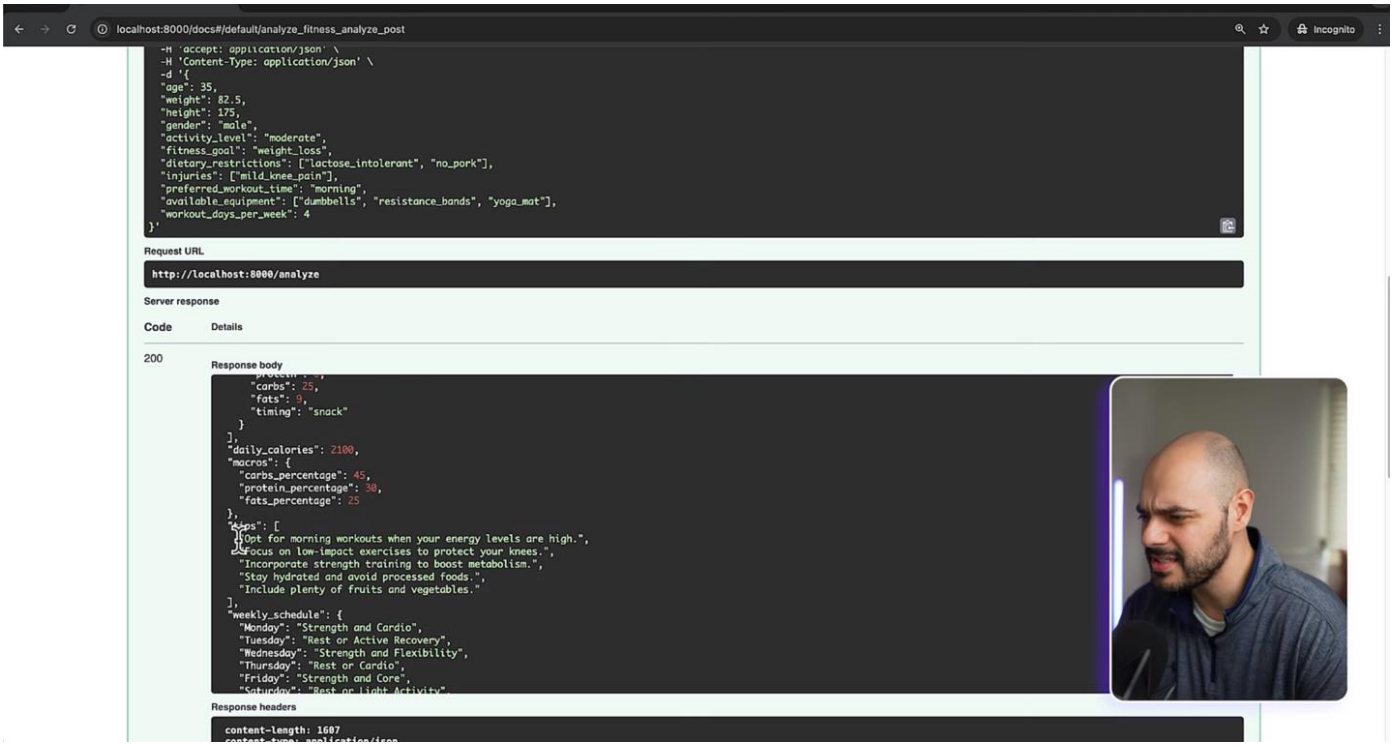

AI Fitness Trainer - Swagger
localhost:8000/docs#/default/analyze_fitness_analyze_post
Incognito
AI Fitness Trainer
0.1.0
OAS 3.1
/openapi.json
default
POST
/analyze
Analyze Fitness
Parameters
Cancel
Reset
No parameters
Request body
required
application/json
{
"age": 35,
"weight": 82.5,
"height": 175,
"gender": "male",
"activity_level": "moderate",
"fitness_goal": "weight_loss",
"dietary_restrictions": ["lactose_intolerant", "no_pork"],
"injuries": ["mild_knee_pain"],
"preferred_workout_time": "morning",
"available_equipment": ["dumbbells", "resistance_bands", "yoga_mat"],
"workout_days_per_week": 4
}
Execute
LOADING

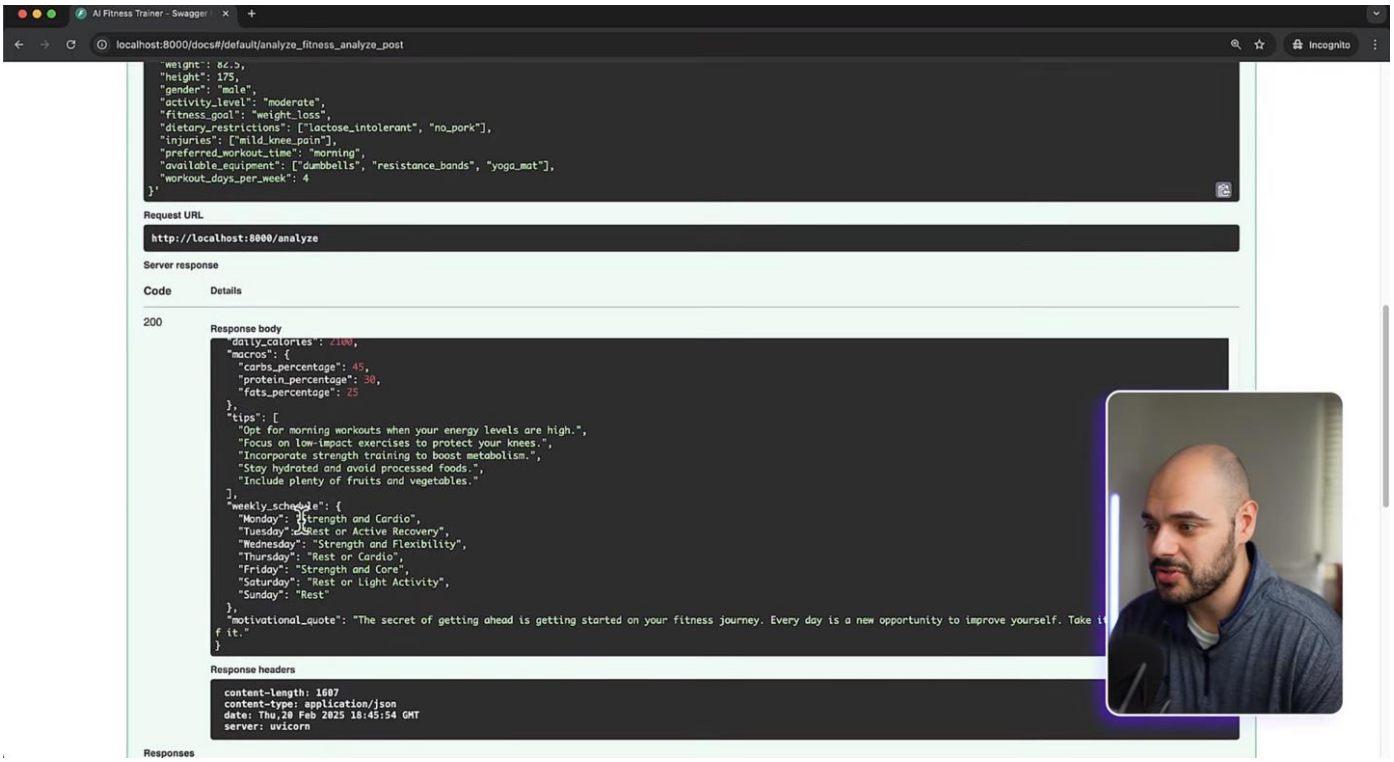

localhost:8000/docs#/default/analyze_fitness_analyze_post
Incognito
Execute
Clear
Responses
Curl
curl -X 'POST' \
'http://localhost:8000/analyze' \
-H 'accept: application/json' \
-H 'Content-Type: application/json' \
-d '{
"age": 35,
"weight": 82.5,
"height": 175,
"gender": "male",
"activity_level": "moderate",
"fitness_goal": "weight_loss",
"dietary_restrictions": ["lactose_intolerant", "no_pork"],
"injuries": ["mild_knee_pain"],
"preferred_workout_time": "morning",
"available_equipment": ["dumbbells", "resistance_bands", "yoga_mat"],
"workout_days_per_week": 4
}'
Request URL
http://localhost:8000/analyze
Server response
Code
Details
200
Response body
{
"workout_plan": [
{
"name": "Dumbbell Squats",
"sets": 3,
"reps": 12,
"rest_time": 60
},
{
"name": "Resistance Band Deadlifts",
"sets": 3,
"reps": 12,
"rest_time": 60
},


localhost:8000/docs#/default/analyze_fitness_analyze_post
Incognito
-H 'accept: application/json' \
-H 'Content-Type: application/json' \
-d '{
"age": 35,
"weight": 82.5,
"height": 175,
"gender": "male",
"activity_level": "moderate",
"fitness_goal": "weight_loss",
"dietary_restrictions": ["lactose_intolerant", "no_pork"],
"injuries": ["mild_knee_pain"],
"preferred_workout_time": "morning",
"available_equipment": ["dumbbells", "resistance_bands", "yoga_mat"],
"workout_days_per_week": 4
}'
Request URL
http://localhost:8000/analyze
Server response
Code
Details
200
Response body
{
"workout_plan": [
{
"name": "Dumbbell Squats",
"sets": 3,
"reps": 12,
"rest_time": 60
},
{
"name": "Resistance Band Deadlifts",
"sets": 3,
"reps": 12,
"rest_time": 60
},
{
"name": "Resistance Band Shoulder Press",
"sets": 3,
"reps": 12,
"rest_time": 60
},
{
"name": "Dumbbell Rows",
"sets": 3,
"reps": 12,
Response headers
content-length: 1607


"workout_days_per_week": 4
}'
Request URL
http://localhost:8000/analyze
Server response
Code
Details
200
Response body
"name": "Resistance Band Shoulder Press",
"sets": 3,
"reps": 12,
"rest_time": 60
"name": "Dumbbell Rows",
"sets": 3,
"reps": 12,
"rest_time": 60
"name": "Yoga Stretch Routine",
"sets": 1,
"reps": 1,
"rest_time": 0
"meal_plan": [
"name": "Oatmeal with Almond Milk and Fruits",
"calories": 350,
"protein": 10,
"carbs": 60,
Response headers
content-length: 1607
content-type: application/json


localhost:8000/docs#/default/analyze_fitness_analyze_post
Incognito
-H 'accept: application/json' \
-H 'Content-Type: application/json' \
-d '{
"age": 35,
"weight": 82.5,
"height": 175,
"gender": "male",
"activity_level": "moderate",
"fitness_goal": "weight_loss",
"dietary_restrictions": ["lactose_intolerant", "no_pork"],
"injuries": ["mild_knee_pain"],
"preferred_workout_time": "morning",
"available_equipment": ["dumbbells", "resistance_bands", "yoga_mat"],
"workout_days_per_week": 4
}'
Request URL
http://localhost:8000/analyze
Server response
Code
Details
200
Response body
"reps": 1,
"rest_time": 0
"meal_plan": [
"name": "Oatmeal with Almond Milk and Fruits",
"calories": 350,
"protein": 10,
"carbs": 60,
"fats": 8,
"timing": "breakfast"
"name": "Grilled Chicken Salad with Olive Oil",
"calories": 450,
"protein": 35,
"carbs": 20,
"fats": 20,
"timing": "lunch"
"name": "Quinoa and Black Bean Bowl",
"calories": 400,
"protein": 15,
Response headers
content-length: 1607
content-type: application/json


localhost:8000/docs#/default/analyze_fitness_analyze_post
Incognito
-H 'accept: application/json' \
-H 'Content-Type: application/json' \
-d '{
"age": 35,
"weight": 82.5,
"height": 175,
"gender": "male",
"activity_level": "moderate",
"fitness_goal": "weight_loss",
"dietary_restrictions": ["lactose_intolerant", "no_pork"],
"injuries": ["mild_knee_pain"],
"preferred_workout_time": "morning",
"available_equipment": ["dumbbells", "resistance_bands", "yoga_mat"],
"workout_days_per_week": 4
}'
Request URL
http://localhost:8000/analyze
Server response
Code
Details
200
Response body
"carbs": 25,
"fats": 9,
"timing": "snack"
"daily_calories": 2100,
"macros": {
"carbs_percentage": 45,
"protein_percentage": 30,
"fats_percentage": 25
"tips": [
"Opt for morning workouts when your energy levels are high.",
"Focus on low-impact exercises to protect your knees.",
"Incorporate strength training to boost metabolism.",
"Stay hydrated and avoid processed foods.",
"Include plenty of fruits and vegetables."
"weekly_schedule": {
"Monday": "Strength and Cardio",
"Tuesday": "Rest or Active Recovery",
"Wednesday": "Strength and Flexibility",
"Thursday": "Rest or Cardio",
"Friday": "Strength and Core",
Response headers
content-length: 1607


AI Fitness Trainer - Swagger
localhost:8000/docs#/default/analyze_fitness_analyze_post
Incognito
"weight": 82.5,
"height": 175,
"gender": "male",
"activity_level": "moderate",
"fitness_goal": "weight_loss",
"dietary_restrictions": ["lactose_intolerant", "no_pork"],
"injuries": ["mild_knee_pain"],
"preferred_workout_time": "morning",
"available_equipment": ["dumbbells", "resistance_bands", "yoga_mat"],
"workout_days_per_week": 4
}'
Request URL
http://localhost:8000/analyze
Server response
Code
Details
200
Response body
"daily_calories": 2100,
"macros": {
"carbs_percentage": 45,
"protein_percentage": 30,
"fats_percentage": 25
"tips": [
"Opt for morning workouts when your energy levels are high.",
"Focus on low-impact exercises to protect your knees.",
"Incorporate strength training to boost metabolism.",
"Stay hydrated and avoid processed foods.",
"Include plenty of fruits and vegetables."
"weekly_schedule": {
"Monday": "Strength and Cardio",
"Tuesday": "Rest or Active Recovery",
"Wednesday": "Strength and Flexibility",
"Thursday": "Rest or Cardio",
"Friday": "Strength and Core",
"Saturday": "Rest or Light Activity",
"Sunday": "Rest"
"motivational_quote": "The secret of getting ahead is getting started on your fitness journey. Every day is a new opportunity to improve yourself. Take it
f it."
Response headers
content-length: 1607
content-type: application/json
date: Thu,20 Feb 2025 18:45:54 GMT
server: uvicorn
Responses