

Creating a personalized running plan with Elasticsearch + Agno

By Jessica Garson



Can I run a faster 5k?



**I use a lot of different
apps to track my
workouts**





Everything syncs to Apple Health ❤

How I built this solution

Python + Agno + Elasticsearch + Notion = Faster 5k times???

Step 1

Get your data from Apple health in an XML file

Step 2

Parse the workout data from Apple Health from the last three months and sending it to Elasticsearch

Step 3

Create a script that connects to Elasticsearch using Agno to create a personalized workout plan as a markdown file

Step 4

Send your markdown file to Notion to track your path to a faster 5k 

How I built this solution

Python + Agno + Elasticsearch + Notion = Faster 5k times???



apple_health_export/
export.xml

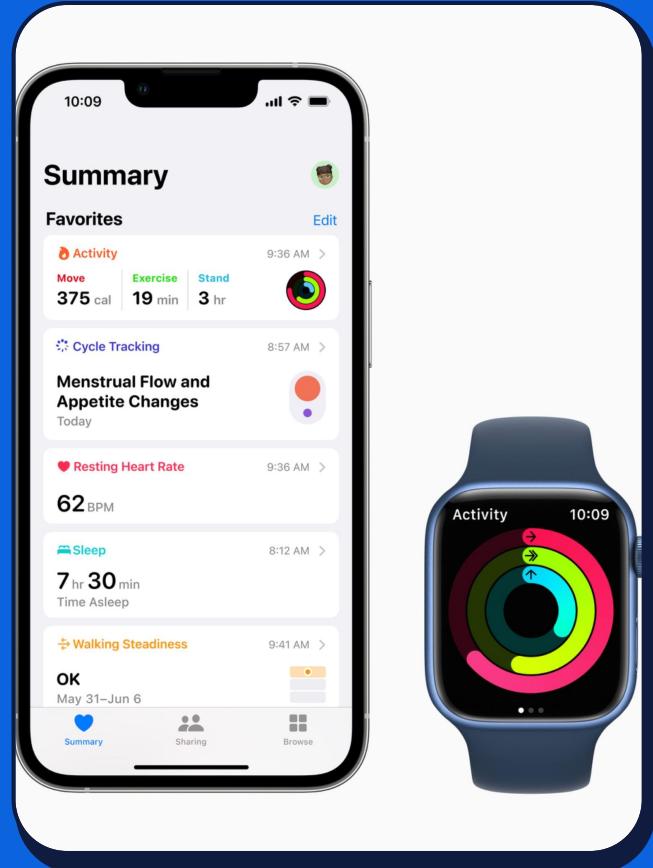
parse_data.ipynb

plan.py

notion.ipynb

Getting workout data from Apple health

1. Go to the Health app on your iPhone.
2. Tap your picture or initials at the top right.
If you don't see your picture or initials, tap Summary or Browse at the bottom of the screen, then scroll to the top of the screen.
3. Tap Export All Health Data, then choose a method for sharing your data.



```
for workout in root.findall('Workout'):
    start_date_str = workout.attrib['startDate']
    start_date = datetime.strptime(start_date_str[:19], "%Y-%m-%d %H:%M:%S")
    if start_date >= three_months_ago:
        distance_miles = 0.0
        calories = 0.0
        for stat in workout.findall('WorkoutStatistics'):
            if stat.attrib['type'] ==
"HKQuantityTypeIdentifierDistanceWalkingRunning":
                distance_miles = float(stat.attrib.get('sum', 0))
            elif stat.attrib['type'] ==
"HKQuantityTypeIdentifierActiveEnergyBurned":
                calories = float(stat.attrib.get('sum', 0))
        doc = {
            'workout_type': workout.attrib.get('workoutActivityType', 'Unknown'),
            'start_time': workout.attrib['startDate'],
            'end_time': workout.attrib['endDate'],
            'distance_miles': distance_miles,
            'distance_km': distance_miles * 1.60934,
            'calories_burned': calories,
            'device': workout.attrib.get('sourceName', 'Unknown')
        }
        docs.append(doc)
```

Getting data from
the past 3 months
into the right
format

Sending your workout data to Elastic

parse_data.ipynb

Python

```
if docs:
    actions = [
        {
            "_index": "apple-health-workouts",
            "_source": doc
        }
        for doc in docs
    ]
    helpers.bulk(es, actions)
    print(f"Successfully indexed {len(docs)} workouts into Elasticsearch!")
else:
    print("No workouts found in the last 3 months.")
```

How can I turn this data into a fitness plan?

I wanted to use an Python framework for creating agents



What is Agno?

Agno is a streamlined Python library designed for developing agents.

What are AI agents?

AI agents can engage with their surroundings, gather information, and utilize that data to independently accomplish tasks and reach objectives. This autonomy allows them to make decisions and take actions without external guidance.

Message

Here are my recent workouts:

Workout on 2025-04-01 06:47:00 -0400: 0.0 km

(HKWorkoutActivityTypePreparationAndRecovery)

Workout on 2025-04-02 07:19:00 -0400: 0.0 km

Based on this, create a personalized 4-week running plan for me to run faster.

Response (15.4s)

Based on your recent workouts, it looks like you've been integrating various activities such as running, walking, yoga, high-intensity interval training, and preparation and recovery sessions. Your running distances in the past month have been quite varied, from shorter runs (1.88 km) to longer runs (up to 7.38 km).

Here's a personalized 4-week running plan to help you run faster, while gradually building up your strength and endurance:

Week 1:

Monday:

- Easy Run: 3 km

Getting the data from Elasticsearch

plan.py

Python

```
class SimpleWorkoutKnowledge:
    def __init__(self, es_client, index_name):
        self.es = es_client
        self.index_name = index_name

    def query(self, query_text):
        query_body = {
            "_source": ["workout_type", "start_time", "distance_km"],
            "query": {
                "match_all": {}
            },
            "size": 500
        }
        results = self.es.search(index=self.index_name, body=query_body)
        return [hit["_source"] for hit in results["hits"]["hits"]]

es = Elasticsearch(
    getpass("Host: "),
    api_key=getpass("API Key: "),
```



Defining an agent

plan.py

Python

```
agent = Agent(  
    model=OpenAIChat(id="gpt-4o"),  
    description="Personal Running Coach",  
    instructions=[  
        "Review the user's past running workouts.",  
        "Create a running plan based on past distances and frequency.",  
        "If there are gaps or missed days, add easier re-entry runs.",  
    ],  
    knowledge=SimpleWorkoutKnowledge(es, index_name="apple-health-workouts"),  
    markdown=True  
)
```

Creating your markdown file

plan.py

Python

```
recent_workouts = agent.knowledge.query("recent workouts")

workouts_text = "\n".join([
    f"Workout on {w['start_time']}: {w['distance_km']} km ({w['workout_type']})"
    for w in recent_workouts
])

final_prompt = (
    f"Here are my recent workouts:\n\n{workouts_text}\n\n"
    "Based on this, create a personalized 4-week running plan for me to run
faster."
)

run_response = agent.run(final_prompt, stream=True)
full_text = "".join([chunk.content for chunk in run_response])

with open("running_plan.md", "w") as f:
    f.write(full_text)
```

Based on your recent workouts, it appears that you have been mixing different activities with running, including walking, HIIT, yoga, and cycling. In terms of your running, there is some variability in the distances, with a longer run recorded on April 10th and some shorter runs on other days.

4-Week Running Plan to Improve Speed

Week 1

- **Monday**: Easy run - 2 km at a comfortable pace. Focus on form.
- **Tuesday**: Rest or cross-training (yoga or cycling)
- **Wednesday**: Tempo run - 5 min warm-up, 1 km at a quicker pace, 5 min cool down.
- **Thursday**: Rest or light walking
- **Friday**: Intervals - 4x400m with 1:30 rest in between; warm-up and cool down for 1 km.
- **Saturday**: Rest or preparation/recovery activities
- **Sunday**: Long run - 3 km at a comfortable pace, focusing on maintaining good form.

Setting up to parse the markdown file

notion.ipynb

Python

```
notion = Client(auth=getpass("Notion auth token: "))
PARENT_PAGE_ID = getpass("Parent page id: ")

FILE_PATH = "running_plan.md"

with open(FILE_PATH, 'r', encoding='utf-8') as file:
    markdown_content = file.read()

title_match = re.search(r'^# (.+)$', markdown_content, re.MULTILINE)
title = title_match.group(1) if title_match else "Running Plan"
```

Parsing the markdown file

notion.ipynb

Python

```
def parse_markdown_text(text):

    bold_pattern = r'\*\*(\^\*)+\*\*'
    parts = re.split(r'(\*\*\^\*)+', text)
    rich_text = []

    for part in parts:
        if re.match(bold_pattern, part):
            rich_text.append({
                "type": "text",
                "text": {"content": part[2:-2]}, # strip the "**"
                "annotations": {"bold": True}
            })
        elif part:
            rich_text.append({
                "type": "text",
                "text": {"content": part}
            })

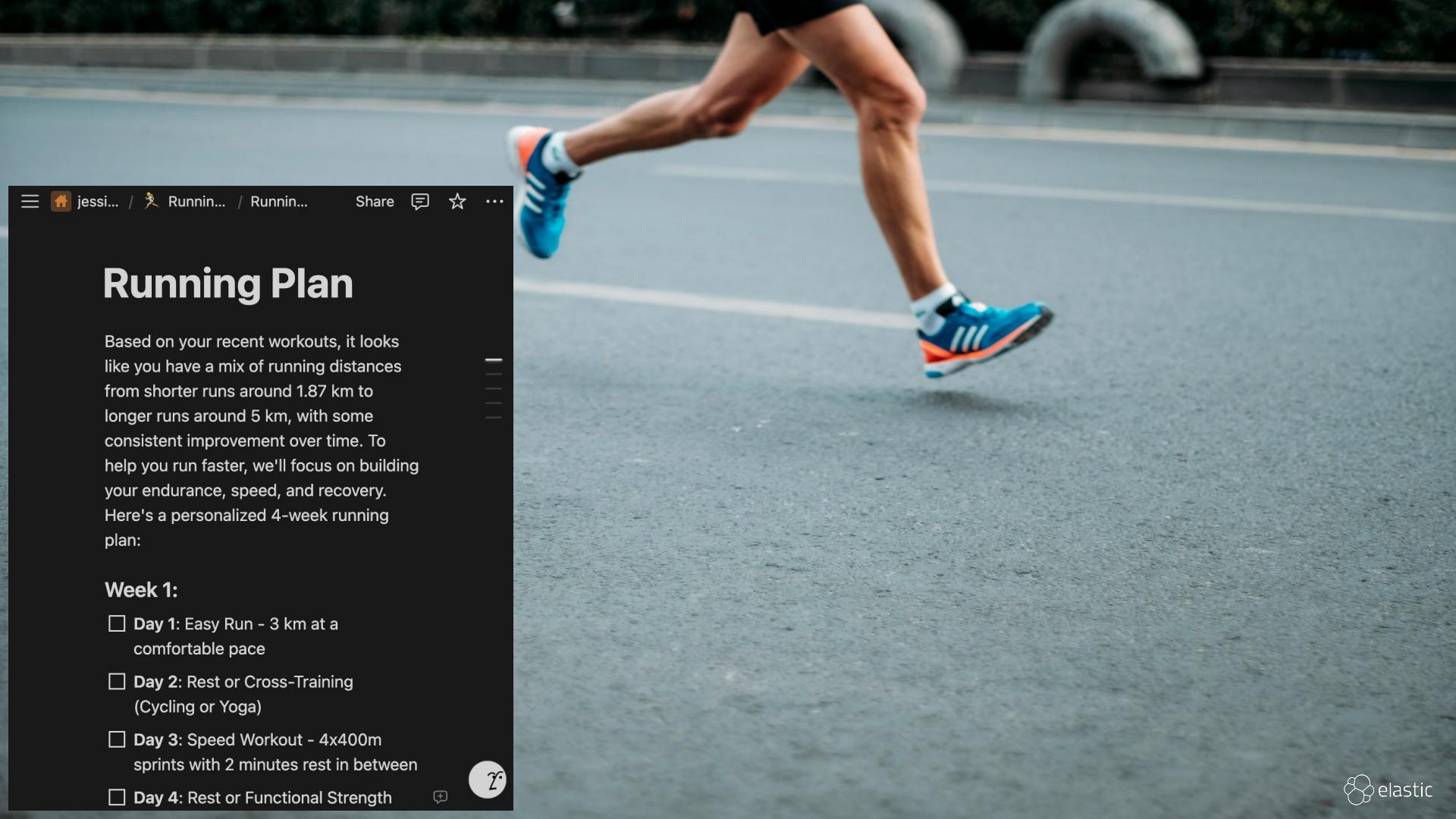
    return rich_text
```

A large pile of colorful LEGO bricks of various shapes and sizes, including plates, beams, and connectors, scattered across the entire background.

Parsing block by block

```
def create_notion_page():
    try:
        page = notion.pages.create(
            parent={"page_id": PARENT_PAGE_ID},
            properties={
                "title": {
                    "title": [{"text": {"content": title}}]
                }
            }
        )
        notion.blocks.children.append(
            block_id=page["id"],
            children=blocks
        )
        print(f"✓ Created Notion page: {title}")
        print(f"🔗 URL: {page.get('url')}")
        todo_count = sum(1 for block in blocks if block["type"] == "to_do")
        checked_count = sum(1 for block in blocks if block["type"] == "to_do" and
block["to_do"]["checked"])
        print(f"📝 To-do items: {todo_count} ({checked_count} completed)")
    return page
except Exception as e:
    print(f"✗ Error: {str(e)}")
return None
```

Sending the Markdown file to Notion



Running Plan

Based on your recent workouts, it looks like you have a mix of running distances from shorter runs around 1.87 km to longer runs around 5 km, with some consistent improvement over time. To help you run faster, we'll focus on building your endurance, speed, and recovery. Here's a personalized 4-week running plan:

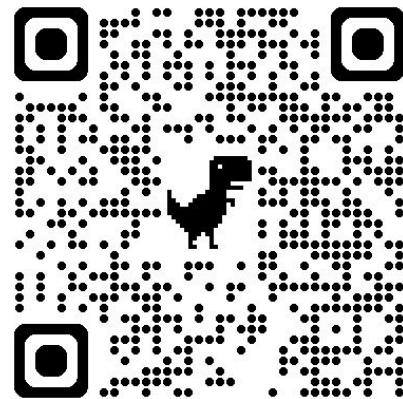
Week 1:

- Day 1:** Easy Run - 3 km at a comfortable pace
- Day 2:** Rest or Cross-Training (Cycling or Yoga)
- Day 3:** Speed Workout - 4x400m sprints with 2 minutes rest in between
- Day 4:** Rest or Functional Strength



Check out the code:

github.com/JessicaGarson/Create-a-Personalized-Running-Plan



Article

allthingsopen.org/articles/step-by-step-guide-python-elasticsearch-agno-agenetic-ai-create-running-plan



Thank you!

