

Profile Matcher API

Written by: Ramin Orak - 14 Feb 2025 – Montreal, QC

Hello,

A FastAPI-based service that retrieves player profiles, matches them with active campaigns, and updates the player's data accordingly.

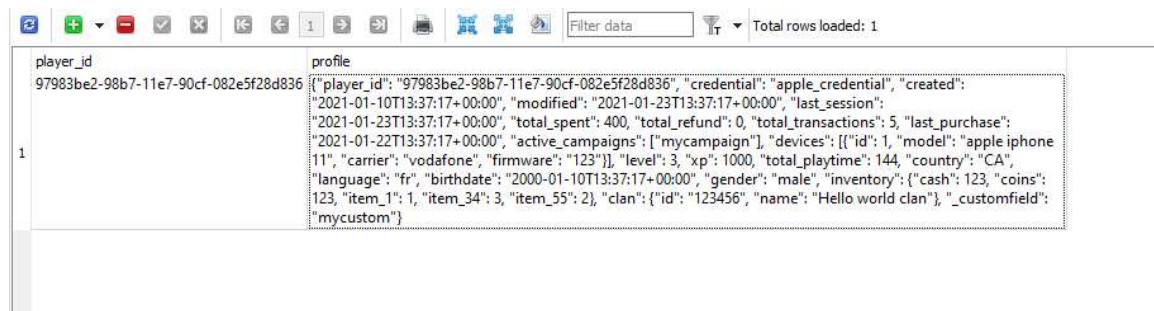
This project contains these files:

📁 gameloft_project

- └ 📄 gameloft.py # Main code FastAPI
- └ 📄 test_gameloft.py # Test API with pytest
- └ 📄 requirements.txt # all dependencies need
- └ 📄 player.db # database
- └ 📄 README.md # document
- └ 📄 ProfileMatcherSwaggerUI.pdf # SwaggerUI Documents

After running **gameloft.py** database will create and initialize with sample player profile.

For the first time this database create and Mock player profile in the database (I choose SQLite for simplicity)



The screenshot shows a data table with two columns: 'player_id' and 'profile'. The 'player_id' column contains the value '97983be2-98b7-11e7-90cf-082e5f28d836'. The 'profile' column contains a JSON object representing a player's profile. The table has a filter bar at the top with the text 'Filter data' and a dropdown arrow. The total rows loaded is 1.

player_id	profile
97983be2-98b7-11e7-90cf-082e5f28d836	{ "player_id": "97983be2-98b7-11e7-90cf-082e5f28d836", "credential": "apple_credential", "created": "2021-01-10T13:37:17+00:00", "modified": "2021-01-23T13:37:17+00:00", "last_session": "2021-01-23T13:37:17+00:00", "total_spent": 400, "total_refund": 0, "total_transactions": 5, "last_purchase": "2021-01-22T13:37:17+00:00", "active_campaigns": ["mycampaign"], "devices": [{"id": 1, "model": "apple iphone 11", "carrier": "vodafone", "firmware": "123"}], "level": 3, "xp": 1000, "total_playtime": 144, "country": "CA", "language": "fr", "birthdate": "2000-01-10T13:37:17+00:00", "gender": "male", "inventory": {"cash": 123, "coins": 123, "item_1": 1, "item_34": 3, "item_55": 2}, "clan": {"id": "123456", "name": "Hello world clan"}, "_customfield": "mycustom"} }

1- Readme.md

```
1  # 🎮 Profile Matcher API
2
3  ## 🚀 Features
4  ✓ Fetches player profiles from SQLite
5  ✓ Matches players with active campaigns
6  ✓ Updates player profiles dynamically
7  ✓ Includes automated API tests
8
9  ## 🛠 Setup & Installation
10
11  pip install -r requirements.txt
12  python gameloft.py
13
```

2- For install dependencies - Requirements.txt
Pip install requirements.txt

fastapi
uvicorn
sqlalchemy
pydantic
requests
pytest

3- python gameloft.py

In the swagger (127.0.0.1:8000/doc)

Request URL

http://127.0.0.1:8000/get_client_config/97983be2-98b7-11e7-90cf-082e5f28d836

Result is :

Response body

```
{
  "player_id": "97983be2-98b7-11e7-90cf-082e5f28d836",
  "credential": "apple_credential",
  "created": "2021-01-10T13:37:17+00:00",
  "modified": "2021-01-23T13:37:17+00:00",
  "last_session": "2021-01-23T13:37:17+00:00",
  "total_spent": 400,
  "total_refund": 0,
  "total_transactions": 5,
  "last_purchase": "2021-01-22T13:37:17+00:00",
  "active_campaigns": [],
  "devices": [
    {
      "id": 1,
      "model": "apple iphone 11",
      "carrier": "vodafone",
      "firmware": "123"
    }
  ],
  "level": 3,
  "xp": 1000,
  "total_playtime": 144,
  "country": "CA",
  "language": "fr",
  "birthdate": "2000-01-10T13:37:17+00:00",
  "gender": "male",
  "inventory": {
    "cash": 123,

```

Because I check the Date validity and expire date this sample could not be added but if for example I change "end_date": "2022-02-25 00:00:00Z" to "end_date": "2025-02-25 00:00:00Z" in the Mock Campaign It will add to active campaign for this player.

The screenshot shows a VS Code editor with a file explorer on the left containing files like `__pycache__`, `gameloft.py`, `parts.DBF`, `players.db`, `profile_matcher.db`, `profilematcher.py`, `showdbf.py`, `ShowExcel.py`, `showword.py`, and `test.py`. The main editor displays the `get_active_campaigns()` function in `gameloft.py`, which defines a campaign object with attributes like `name`, `priority`, `matchers`, `start_date`, `end_date`, `enabled`, and `last_updated`. The bottom panel shows the terminal output, indicating the application is running on `http://127.0.0.1:8000` and responding to a GET request with a 200 OK status.

Result will be this:

```
{
  "player_id": "97983be2-98b7-11e7-90cf-082e5f28d836",
  "credential": "apple_credential",
  "created": "2021-01-10T13:37:17+00:00",
  "modified": "2021-01-23T13:37:17+00:00",
  "last_session": "2021-01-23T13:37:17+00:00",
  "total_spent": 400,
  "total_refund": 0,
  "total_transactions": 5,
  "last_purchase": "2021-01-22T13:37:17+00:00",
  "active_campaigns": [
    "mycampaign"
  ],
  "devices": [
    {
      "id": 1,
      "model": "apple iphone 11",
      "carrier": "vodafone",
      "firmware": "123"
    }
  ],
  "level": 3,
  "xp": 1000,
  "total_playtime": 144,
  "country": "CA",
  "language": "fr",
  "birthdate": "2000-01-10T13:37:17+00:00",
  "gender": "male",
}
```

You can find the resources in this address:

<https://github.com/raminok/python/tree/main/interview/gameloft>