Thông tin cần lưu khi nhập điện thoại

Thought for 17s

- I need to create documentation for the new API endpoint, including basic details like base URL and endpoints: /health (GET) and /chat (POST). I'll cover request/response schemas, examples using curl, httpie, JS fetch, and Python requests. I also need to include error codes, session handling, and details on response content.
- The user suggests including a note that the answer may contain HTML, especially since HTML will be returned. We'll document that HTML strings are allowed and mention if they're sanitized. OpenAPI documentation will automatically be generated, including JSON schema snippets and field descriptions. I'll also mention CORS setup and session constraints.
- For

Done

Dưới đây là tài liệu tích hợp API cho chatbot "**phone-agent01**".

1) Tổng quan

- **Mục đích:** nhận câu hỏi của người dùng về điện thoại, tra cứu dữ liệu (qua Pinecone) và trả về câu trả lời gợi ý/tư vấn.
- **Kiến trúc:** FastAPI, có lưu lịch sử hội thoại theo `session_id` (in-memory).
- **Base URL (ví dụ khi chạy cục bộ):** `http://localhost:8000`

Bạn có thể xem OpenAPI UI tại:

- Swagger UI: `GET /docs`
- ReDoc: `GET /redoc`

> *Lưu ý: Trường `answer` có thể là **HTML** (phía client hiển thị bằng `innerHTML`) nếu bạn nhắc mô hình trả về HTML trong prompt. Nếu không, đó là plain text.*

2) Xác thực & CORS

- **Auth:** Không yêu cầu (tích hợp mạng nội bộ). Nếu cần, hãy bổ sung lớp bảo vệ ở tầng reverse proxy / gateway.
- **CORS:** cấu hình qua biến môi trường `CORS_ALLOW_ORIGINS` (mặc định `"*"`). Bên tích hợp cần chạy origin hợp lệ hoặc bật `*` trong môi trường dev.

3) Endpoint

3.1. Healthcheck

GET `/health`

Mục đích: kiểm tra dịch vụ sẵn sàng.

Response 200

```
{ "status": "ok" }
```

3.2. Chat

POST `/chat`

Mục đích: gửi câu hỏi → hệ thống trích xuất sản phẩm liên quan → truy vấn Pinecone → sinh câu trả lời (có thể là HTML).

Request body (JSON)

```
{
  "session_id": "string - bắt buộc",
  "question": "string - bắt buộc"
}
```

- **`session_id`**: định danh phiên để **lưu lịch sử hội thoại**. Dùng cùng một **`session_id`** cho nhiều lượt để có ngữ cảnh.
- **`question`**: câu hỏi của người dùng.

Response body (JSON)

```
{
  "answer": "string - có thể là HTML",
  "products": [
    {
      "name": "string",
      "description": "string",
      "full_information": "string (thường là markdown do dữ liệu nguồn)"
    }
  ],
```

```
  "session_id": "string"
}
```

- `answer`: câu trả lời đã tổng hợp. Có thể là HTML nếu bạn định hướng prompt.
- `products`: danh sách sản phẩm liên quan tìm được từ Pinecone (đã lọc trùng theo `product_id` nội bộ).
  - `name`: tên sản phẩm
  - `description`: mô tả ngắn
  - `full_information`: khối thông tin chi tiết (định dạng markdown/text trong dữ liệu nguồn)
- `session_id`: echo lại `session_id` request.

Mã lỗi

- **400 Bad Request**
  - Thiếu `question` hoặc `session_id`.
  - Ví dụ:

    ```
    { "detail": "Missing question" }
    ```
- **500 Internal Server Error**
  - Lỗi pipeline:
    - `Intent error: ...` (lỗi trích xuất sản phẩm)
    - `Pinecone query error: ...` (lỗi truy vấn vector DB)
    - `Answer error: ...` (lỗi sinh câu trả lời)

4) Ví dụ tích hợp

4.1. cURL

```
curl -X POST "http://localhost:8000/chat" \
  -H "Content-Type: application/json" \
  -d '{
    "session_id": "web-visitor-001",
    "question": "Mình phân vân iPhone 14 Pro và S23 Ultra, ưu tiên camera và pin."
  }'
```

Response mẫu (rút gọn)

```
{
  "answer": "<h2>Gợi ý cho nhu câù camera & pin</h2><ul><li><strong>iPhone 14 Pro</strong> ...</li><li><strong>Galaxy S23 Ultra</strong> ...</li></ul>",
  "products": [
    {
      "name": "iPhone 14 Pro",
      "description": "iPhone 14 Pro ...",
      "full_information": "- **Thông tin cơ bản** ..."
    },
    {
      "name": "Samsung Galaxy S23 Ultra",
      "description": "Galaxy S23 Ultra ...",
      "full_information": "- **Thông tin cơ bản** ..."
    }
  ],
  "session_id": "web-visitor-001"
}
```

4.2. JavaScript (fetch)

```
const API = "http://localhost:8000/chat";

async function ask(question, sessionId) {
  const res = await fetch(API, {
    method: "POST",
    headers: { "Content-Type": "application/json" },
    body: JSON.stringify({ session_id: sessionId, question })
  });
  if (!res.ok) throw new Error(`HTTP ${res.status}`);
  const data = await res.json();
  // data.answer có thể là HTML -> dùng innerHTML để hiển thị
  answerContainer.innerHTML = data.answer || "<p>(Không có nội dung)</p>";
  console.log(data.products);
}
```

4.3. Python (requests)

```
import requests
```

```
API = "http://localhost:8000/chat"
payload = {
    "session_id": "partner-user-123",
    "question": "Mình cần máy pin trâu, quay 4K/60fps, tầm 20 triệu."
}
r = requests.post(API, json=payload, timeout=60)
r.raise_for_status()
data = r.json()
print("Answer:", data["answer"])        # có thể là HTML
print("Products:", data["products"])    # list sản phẩm
```

5) Hướng dẫn dùng `session_id`

- Tạo một **`session_id`** (ví dụ UUID) cho mỗi phiên chat trên website đối tác.
- **Tái sử dụng** **`session_id`** cho các lượt gửi tiếp theo trong cùng cuộc trò chuyện để hệ thống hiểu ngữ cảnh (vì server lưu: **`[{user, assistant}, ...]`**).
- Khi người dùng “Làm mới”/“Bắt đầu cuộc chat mới”, hãy **tạo** **`session_id`** **mới**.

6) Lưu ý vận hành

- **Trả về HTML trong** **`answer`:** Nếu UI phía đối tác cần hiển thị đẹp, hãy định hướng mô hình sinh **HTML** (ở tầng prompt của bạn). Phía client hiển thị bằng **`innerHTML`**.
- **Kích thước câu hỏi:** Không đặt giới hạn cứng, nhưng nên giữ < 2–3 KB để thời gian sinh câu trả lời nhanh và ổn định.
- **Hiệu năng:** Mỗi lần gọi **`/chat`** sẽ chạy 3 bước: trích xuất → truy vấn Pinecone → sinh câu trả lời. Tùy mô hình và dữ liệu, thời gian phản hồi có thể khác nhau.
- **CORS:** đảm bảo domain của đối tác nằm trong **`CORS_ALLOW_ORIGINS`** nếu gọi trực tiếp từ trình duyệt.

7) Phiên bản & thay đổi

- **`v1.0.0`**: Bản đầu, endpoint **`/health`**, **`/chat`**. Response **`answer`** cho phép HTML.

Nếu bạn cần **header xác thực**, **giới hạn tốc độ**, hoặc **webhook** nhật ký hội thoại, mình có thể bổ sung nhanh vào tài liệu và mã nguồn.