

# Hướng dẫn sử dụng API

## Auth

- Hầu hết API chatbot yêu cầu header `Authorization: Bearer <token>`.
- Token lấy từ API đăng nhập web chính: <https://live-code-be-2.ript.vn/auth/login>
- Ngoại lệ: các API review code không cần token.

## Khái niệm Session & Topic

- **Session**: chuỗi hội thoại lưu toàn bộ message (user/assistant). Khi không truyền `session_id`, API QA sẽ tự tạo session mới. Với chatbot Topic, bắt buộc phải truyền `session_id` đã có topic.
- **Topic**: chủ đề gắn với session (lưu `topic_id` và lấy chi tiết qua `/topics/{id}`). Với chatbot Topic, topic được lấy từ session, client không gửi topic trong request.

## Review code

- Không cần Authorization.
- POST `/review_stream`
  - Body

```
{
  "inputs": {
    "purpose": "string - mô tả đề bài",
    "example_code": "string | null - code mẫu (optional)",
    "user_code": "string - mã cần review"
  }
}
```

- Output: `text/plain` stream (nội dung review).
- POST `/review_non_stream`
  - Body như trên.
  - Output JSON:

```
{ "status": "success", "data": "<nội dung review>" }
```

# Chatbot QA (theo đề bài/code)

Ngữ cảnh gồm đề bài ( question ) và code mẫu/đáp án ( answer ). Mỗi câu hỏi ( user\_question ) được lưu vào session.

- Input chung ( POST /chatbot\_qa và /chatbot\_qa\_non\_stream )

```
{
  "question": "string - đề bài",
  "answer": "string - code mẫu/đáp án",
  "user_question": "string - câu hỏi của sinh viên",
  "session_id": "string | null - optional, để tiếp tục session cũ"
}
```

- Nếu không truyền session\_id : API tự tạo session mới (cần token).
- Output
  - /chatbot\_qa : text/plain stream.
  - /chatbot\_qa\_non\_stream : JSON { "status": "success", "data": "<câu trả lời>" } .

# Chatbot theo Topic (kịch bản cố định của Session)

Session đã có topic\_id ; server sẽ lấy chi tiết topic qua /topics/{id} . Client **không** gửi topic hay đề bài/code, chỉ cần session\_id và câu hỏi.

- Input chung ( POST /chatbot\_topic\_stream và /chatbot\_topic\_non\_stream )

```
{
  "session_id": "string - bắt buộc, session đã gắn topic",
  "user_question": "string - câu hỏi"
}
```

- Ràng buộc & lỗi
  - Thiếu session\_id → 400.
  - session\_id không tồn tại → 404.
  - Session chưa có topic\_id → 400.
  - Topic không tồn tại ( /topics/{id} trả not found ) → 404.
- Output
  - /chatbot\_topic\_stream : text/plain stream.
  - /chatbot\_topic\_non\_stream : JSON { "status": "success", "data": "<câu trả lời>" } .

# Tạo Session & luồng gọi nhanh

- Tạo session (nếu cần chủ động tạo trước): POST /sessions

```
{
  "session_name": "optional",
  "question_id": "optional",
  "question_content": "optional",
  "topic": "optional - nếu muốn gán topic ngay (topic_id do backend quyết định)"
}
```

- Với chatbot QA: có thể bỏ qua bước này, API sẽ tự tạo session nếu không gửi session\_id .
- Với chatbot Topic: cần một session đã gán topic\_id (được backend lưu). Nếu chưa có, hãy tạo session qua backend trước, bảo đảm topic đã được set.
- Luồng chat QA (đề bài/code)
  - (Tuỳ chọn) tạo session trước, hoặc bỏ qua để API tự tạo.
  - Gọi /chatbot\_qa (stream) hoặc /chatbot\_qa\_non\_stream (non-stream) với session\_id nếu muốn nối tiếp cuộc trò chuyện.
  - API lưu message user/assistant vào session.
- Luồng chat Topic
  - Tạo session và gán topic (backend phải lưu topic\_id cho session).
  - Gọi /chatbot\_topic\_stream hoặc /chatbot\_topic\_non\_stream với session\_id đó.
  - API tự tra topic từ session → /topics/{id} → chat theo topic, lưu message user/assistant.