

Renflair AI — opaiv2 API Documentation

Final tested endpoint: <https://www.renflairai.com/api/opaiv2.php>

1) Overview

This endpoint provides a simple JSON POST API to get AI-generated answers scoped to a business account. The older `admin_id` parameter has been replaced with `secret_id` (a server-issued secret) for authentication/tracking.

Successful calls return a short answer and a `response_code` (a unique identifier you can use to trace the request in your logs).

2) Base URL

<https://www.renflairai.com/api/opaiv2.php>

3) HTTP Method

POST

4) Headers

- Content-Type: application/json
 - Use HTTPS for all requests.
-

5) Request Body (JSON)

```
{
  "secret_id": XXXX585,
  "question": "how can i earn huge money in 1 yr"
}
```

Parameters

- `secret_id` (integer|string) — **required**. The server-issued secret that identifies and authenticates the business account. Do NOT expose this in public client-side source; store securely.
 - `question` (string) — **required**. The user's question (text) to be answered by the AI.
-

6) Example cURL (quick test)

```
curl -X POST https://www.renflairai.com/api/opaiv2.php \
-H "Content-Type: application/json" \
```

```
-d '{
  "secret_id": 7564513,
  "question": "how can i earn huge money in 1 yr"
}'
```

7) Example PHP client (server-side)

```
<?php
$url = "https://www.renflairai.com/api/opaiv2.php";
$data = [
  "secret_id" => 7564513,
  "question" => "how can i earn huge money in 1 yr"
];
$ch = curl_init($url);
curl_setopt_array($ch, [
  CURLOPT_RETURNTRANSFER => true,
  CURLOPT_HTTPHEADER => ["Content-Type: application/json"],
  CURLOPT_POST => true,
  CURLOPT_POSTFIELDS => json_encode($data),
  CURLOPT_TIMEOUT => 60
]);
$response = curl_exec($ch);
curl_close($ch);
echo $response;
?>
```

8) Successful Response

Content-Type: application/json

Example (actual tested output):

```
{
  "status": "success",
  "response_code": "ed60eb69a54959ebcf252533b4feb0a0",
  "answer": "Consider investing in stocks, starting a successful business, or
taking on high-paying freelance projects. Remember, financial success usually
requires long-term planning and dedication."
}
```

Response fields

- status — success or error (string).
- response_code — string: a unique identifier for this response (hash/uuid-like). Use this to look up logs/transactions on the server. Note: the value returned in examples may be a generated placeholder in test responses; in production this should be a real, unique tracking id.
- answer — string: AI-generated reply.

9) Error Responses (examples)

- Missing required fields

```
{ "status": "error", "message": "Missing or invalid parameters" }
```

- Invalid secret_id

```
{ "status": "error", "message": "Invalid secret_id" }
```

- No balance left (wallet exhausted)

```
{ "status": "error", "message": "No balance left" }
```

- OpenAI / upstream error

```
{ "status": "error", "message": "Upstream API error: <details>" }
```

Note: The API currently returns JSON objects describing the error. You may want to standardize HTTP status codes (e.g. 400 Bad Request, 401 Unauthorized, 402 Payment Required, 500 Internal Server Error) when you integrate further.

10) Security & Best Practices

- **Never** send secret_id from public client code (mobile/web front-end). Keep it on your server or exchange for a short-lived token.
 - Use **HTTPS** only.
 - Rotate secret_id periodically and support revocation.
 - Rate-limit requests per secret_id to avoid runaway costs.
 - Log request/response metadata (IP, timestamp, prompt size, tokens) for auditing.
 - Sanitize and validate question input length (e.g. max 2,000 characters) to avoid extremely large prompts.
-

11) Recommended client-timeout & retries

- Timeout: 60 seconds (server calls OpenAI and may take time).
 - Retries: 1 retry with exponential backoff on network/timeouts. Avoid retrying on No balance left or 4xx errors.
-

12) Versioning & changelog

- Current: opaiv2.php (uses secret_id)
 - When changing response shape, increment endpoint version (e.g. /opaiv3.php) and keep backward compatibility if possible.
-

15) Contact & support

If you need changes to the response format or want to add features (e.g. `history=true` to include conversation history in the reply), update the server-side script and communicate new fields to clients.

Document created for Renflair ai — opaiv2 endpoint.