Api Troubleshooting

Deep API Troubleshooting Guide

Goal

API troubleshooting is about finding **where the failure is happening**:

```
Client/App → Network → API Gateway → Authentication → Backend Service → Database/External System
```

A good implementation engineer isolates the layer before escalating.

---

1. Start With 4 Basic Questions

Question	Why It Matters
What is expected to happen?	Defines success
What actually happens?	Defines the failure
Is it reproducible?	Separates one-time issue from systemic issue
Who is affected?	Helps identify scope

Example:

Expected: CRM ticket is created after chat ends.

Actual: Chat ends, but no CRM ticket appears.

Scope: One customer environment, all agents.

---

2. Identify the API Direction

Direction	Meaning	Example
Inbound API	Customer system calls platform API	CRM calls Glia API
Outbound API	Platform calls customer system	Glia calls CRM API
Webhook	Event notification sent automatically	Chat ended → CRM ticket
Callback	Response sent after async process	Payment status update

This matters because it tells you **which side to check first**.

---

3. Check the HTTP Method

Method	Troubleshooting Focus
GET	Is the resource ID correct?
POST	Is the payload valid?
PUT	Are all required fields included?
PATCH	Are updated fields allowed?
DELETE	Does the user/token have delete permission?

Example issue:

```
GET /api/create-ticket
```

But the API expects:

```
POST /api/create-ticket
```

Likely result:

405 Method Not Allowed

4. Check the Endpoint

Validate:

- Base URL
- API version
- Path
- Resource ID
- Query parameters
- Environment

Example:

```
https://api.vendor.com/v1/customers/12345
```

Common mistakes:

Problem	Example
Wrong environment	Using sandbox instead of production
Wrong API version	`/v1/` instead of `/v2/`
Typo in endpoint	`/costumers/` instead of `/customers/`
Missing resource ID	`/customers/` instead of `/customers/12345`

5. Check Headers

Important headers:

Header	Purpose
Authorization	Sends bearer token/API key
Content-Type	Format of request body
Accept	Format expected in response

Header	Purpose
x-api-key	API key authentication
Correlation-ID	Request tracking

Example:

```
Authorization: Bearer eyJhbGc...
Content-Type: application/json
Accept: application/json
```

Common issue:

```
Content-Type missing
```

API may not understand the JSON body.

---

6. Check Authentication

Most API failures happen here.

Error	Meaning	Common Cause
401	Not authenticated	Missing/expired/invalid token
403	Authenticated but blocked	Missing permission/scope/role

401 Checklist

Check:

```
Is Authorization header present?
Is token expired?
Is token copied correctly?
Is token for the right environment?
Was token revoked?
```

403 Checklist

Check:

```
Does token have correct scope?
Does user/app have permission?
Is IP allowlisting required?
Is this endpoint restricted?
```

---

7. Check OAuth Scope

Scopes define what the token can do.

Example:

```
read:customers
write:tickets
admin
```

If token only has:

```
read:customers
```

but you call:

```
POST /api/tickets
```

You may get:

```
403 Forbidden
```

---

8. Check the JSON Payload

For POST/PUT/PATCH, inspect payload carefully.

Example:

```
{
  "customerId": "12345",
  "channel": "chat",
  "authenticated": true
}
```

Validate:

Item	Example Problem
Syntax	Missing comma
Required fields	Missing customerId
Data type	`"true"` instead of `true`
Field name	`customerID` instead of `customerId`
Nesting	Object expected, string sent
Array format	Single value sent instead of list

---

9. Understand Status Codes by Category

Code Range	Meaning
2xx	Success
3xx	Redirect
4xx	Client/request issue
5xx	Server/backend issue

Important Codes

Code	Meaning	First Place To Look
200	Success	Validate response body

Code	Meaning	First Place To Look
201	Created	Confirm resource ID returned
204	No Content	Success, but no body
400	Bad Request	Payload/parameters
401	Unauthorized	Authentication
403	Forbidden	Permissions/scopes
404	Not Found	Endpoint/resource
405	Method Not Allowed	HTTP method
408	Timeout	Network/backend delay
409	Conflict	Duplicate/existing resource
415	Unsupported Media Type	Content-Type
422	Validation error	Field validation
429	Rate limit	Too many requests
500	Server error	Backend logs
502	Bad gateway	Proxy/upstream
503	Unavailable	Outage/maintenance
504	Gateway timeout	Backend too slow

10. Troubleshooting by Error Code

400 Bad Request

Likely:

> Invalid payload, missing fields, wrong query parameter

Check:

JSON syntax
Required fields
Field names
Data types
API documentation

401 Unauthorized

Likely:

Authentication failed

Check:

Token present
Token expired
Correct auth method
Correct environment

403 Forbidden

Likely:

Permission issue

Check:

User role
OAuth scopes
API permissions
IP allowlist

404 Not Found

Likely:

Wrong endpoint or resource does not exist

Check:

URL path
API version
Resource ID
Environment

---

409 Conflict

Likely:

Duplicate or conflicting resource

Example:

Trying to create a user that already exists

---

415 Unsupported Media Type

Likely:

Wrong or missing Content-Type

Check:

Content-Type: application/json

---

422 Validation Error

Likely:

Payload is valid JSON but fails business rules

Example:

Phone number format invalid

Required field has invalid value

---

429 Too Many Requests

Likely:

Rate limit exceeded

Check:

Retry-After header

API rate limits

Looping automation

Retry logic

---

500 Internal Server Error

Likely:

Backend application error

Action:

Collect request ID, timestamp, payload sample, and escalate

---

502 / 503 / 504

Likely:

Gateway, service availability, or timeout issue

Check:

- Load balancer
- API gateway
- Backend health
- Maintenance window
- Network latency

---

11. Network Layer Checks

APIs depend on network reachability.

Check:

- DNS resolution
- Firewall
- Proxy
- Port 443
- TLS certificate
- IP allowlisting
- VPN/private connectivity

Common symptoms:

Symptom	Possible Cause
Timeout	Firewall, proxy, backend delay
TLS error	Certificate or TLS mismatch
DNS failure	Wrong hostname
Connection refused	Service down or port blocked

---

12. TLS / Certificate Checks

For HTTPS APIs, validate:

```
Certificate not expired
Hostname matches certificate
Certificate chain trusted
TLS 1.2 or 1.3 supported
```

Common failures:

```
certificate expired
self-signed certificate
hostname mismatch
TLS handshake failed
```

---

13. Rate Limiting

Rate limiting protects APIs from overload.

Common response:

```
429 Too Many Requests
```

Check headers:

```
Retry-After: 60
```

Troubleshooting:

```
Reduce request frequency
Add backoff/retry logic
Check if automation is looping
Review API quota
```

---

14. Timeouts

Timeouts can occur at multiple layers:

```
Client timeout
API gateway timeout
Load balancer timeout
Backend service timeout
Database timeout
```

Ask:

```
How long before failure?
Does it fail always or intermittently?
Is payload large?
Is backend dependency slow?
```

---

15. Correlation IDs

A correlation ID tracks one request across systems.

Example:

```
X-Correlation-ID: abc-123
```

Use it to trace:

```
API Gateway → Auth Service → Backend Service → Database
```

For escalation, always include:

```
correlation ID
timestamp
endpoint
HTTP method
response code
```

---

16. Logs To Review

Log Type	What It Shows
API Gateway logs	Request entry, routing, response code
Auth logs	OAuth/token failures
Application logs	Backend errors
Webhook logs	Delivery attempts
Load balancer logs	Upstream health/timeouts
Firewall/proxy logs	Blocked connections
CRM logs	Customer/ticket integration failures

---

17. Good Troubleshooting Workflow

1. Confirm expected behavior
2. Reproduce issue
3. Identify endpoint + method
4. Check response code
5. Validate auth
6. Validate headers
7. Validate payload
8. Validate network/TLS
9. Check logs/correlation ID
10. Isolate failed layer
11. Escalate with evidence

---

18. Example Scenario

Issue

Chat ends but CRM ticket is not created.

Expected Flow

Chat completed → Webhook sent → CRM API creates ticket

Troubleshooting

Check:

- Was chat_completed event generated?
- Was webhook sent?
- Did CRM receive webhook?
- What HTTP response did CRM return?
- Was token valid?
- Was JSON payload accepted?
- Was ticket required field missing?

Possible findings:

Finding	Meaning
No event generated	Platform workflow issue
Webhook sent, no response	CRM endpoint/network issue
401 from CRM	Auth/token issue
400 from CRM	Payload/field mapping issue
500 from CRM	CRM backend issue

---

19. Another Scenario

Issue

Customer lookup fails during chat start.

Expected Flow

Customer enters → API calls CRM → CRM returns profile

Check:

Customer ID passed correctly?
CRM endpoint correct?
Token valid?
Customer exists?
Field mapping correct?
CRM API response code?

Likely outcomes:

Response	Likely Issue
401	Token expired
403	Missing CRM permission
404	Customer not found
400	Bad customerId format
500	CRM backend failure

---

20. What To Say In Interview

API Troubleshooting Answer

> “I start by confirming the expected workflow and reproducing the issue. Then I identify the API endpoint, HTTP method, response code, headers, authentication method, and payload. Based on the response code, I isolate whether the issue is

> authentication, permissions, request formatting, endpoint, networking, or backend related. I also check logs, timestamps, and correlation IDs before escalating with clear evidence.”

---

21. Implementation Engineer Mindset

Do not just say:

```
The API failed.
```

Say:

```
The POST request to the CRM ticket endpoint is returning 400 because the payload is missing the required caseType field.
```

That shows strong technical ownership.

Revision #1
Created 21 May 2026 00:53:44 by Cesar Gzz
Updated 21 May 2026 00:53:51 by Cesar Gzz