@demotoprod · EVALS · also LEVERS

Structured output checklist

JSON mode guarantees valid JSON and says nothing about your keys, so a required field can go missing and the output still parses. Five checks, each sized for an afternoon, put a strict schema, a stop-reason test, a validator and a retry-then-park path before your database.

1 Switch from JSON mode to strict schema decoding

One provider's guide says both modes ensure valid JSON, but only structured outputs ensure schema adherence. In its 2024 evals of complex schema following, one model with strict decoding scored 100%, an earlier model under 40%: what can happen, not every model.

- Move each call from JSON mode to a strict JSON-schema response format, one endpoint at a time.
- Replay saved production prompts through both paths and count missing keys.

2 List every key as required and close every object

In JSON Schema, properties are not required by default and extra properties are allowed, so an object missing your key is valid. One platform's guide asks for every field as required, additionalProperties false, and a null union for optional fields.

- List every key in required at every nesting level; set additionalProperties to false on every object.
- Write an optional field as a type union with null, so the key is always present.

```
{"type": "object", "required": ["id", "unit"],
 "properties": {"id": {"type": "string"}, "unit": {"type": ["string", "null"]}},
 "additionalProperties": false}
```

3 Read the stop reason before you parse

One provider's guide lists cases where output might not match the schema: a refusal, whose message takes precedence over the schema, and a response cut off at the token limit, which might be incomplete. One client library raises an error on length and content-filter stops.

- Read the refusal field and the stop reason first; parse only when the run ended normally.
- Send a refusal to a person; on a token-limit stop, retry once with a higher limit.

4 Validate every object in your own code, and reject on failure

The same guide lists minimum, maximum and string-length constraints as unsupported in schemas and does not guarantee enum capitalization; its validating helper still enforces every constraint in your code. A validation library raises an error on invalid data.

- Check every response against all constraints: required keys, ranges, lengths, enum values.
- On failure, discard the whole object; do not fill defaults, trim values or fix keys by hand.

```
data = json.loads(text)
missing = REQUIRED - data.keys()
if missing or not 0 < data["qty"] <= 1000:
    return retry_or_park(text, sorted(missing))
```

5 Retry once with the error, then park the object

One agent framework validates output, tells the model to try again on failure, and defaults its output retry budget to 1. A receiver that cannot process a message should move it to an invalid message channel.

- Retry once, adding the validation error and the failed object to the prompt; log each retry.
- After a second failure, send the object to a review queue, not the database; alert on growth.

Nitish Gautam · indica-tech.com · hello@indica-tech.com

INDICA TECH · @DEMOTOPROD

Nitish Gautam

Founder-led AI engineering. Demo to production, under NDA, in regulated industries, on fixed scope.

Paste into your project

ONE GATE PER ENDPOINT, BEFORE THE DATABASE

```
decoding: strict_json_schema       # not JSON mode
schema:
  every_key_required: true         # at every nesting level
  additional_properties: false     # on every object
  optional_field: union_with_null
before_parse: {refusal: send_to_person, token_limit: retry_once}
validate_in_code: [required_keys, ranges, lengths, enum_values]
on_failure:
  retries_with_error: 1            # one framework's default
  then: review_queue               # never the database
```

Where this list comes from

Ten-plus years taking products from zero to one across seven sectors, including defence, fintech, medical and telecom, with AI shipped into regulated and air-gapped environments. The @demotoprod channel takes one production AI failure apart each day. These checklists are the same work written down: every item is a dated failure or a documented default, with the control that stops it and the primary source.

If this list found a gap

1. ### Your demo works and production is stuck.

   Production Readiness Audit. Fixed price, two weeks. A five-dimension scorecard, cost and failure analysis, and a 90-day execution roadmap.

2. ### You need senior AI judgement without a full-time hire.

   Fractional AI Lead. Monthly: 90-day minimum, then 30 days' notice either side. Weekly 1:1 with the founder or CTO, architecture and code review, technical hiring support, investor-ready due diligence.

3. ### An AI feature has to work, by a date.

   AI Build Sprint. Fixed price, four to eight weeks. Scope signed off up front, weekly demos, CI/CD and observability included, 30-day production warranty.

4. ### A regulator, an auditor or a customer's security team is asking.

   AI Governance & Compliance. Fixed price, four weeks. ISO 27001 / SOC 2-aligned controls, a model risk framework, human-in-the-loop and audit logging, regulator-ready documentation.

BOOK A CALL
indica-tech.com/#contact

EMAIL
hello@indica-tech.com

WEBSITE
indica-tech.com

ALL CHECKLISTS
indica-tech.com/checklists/

YOUTUBE
youtube.com/@demotoprod

LINKEDIN
linkedin.com/in/startup-founderx

INSTAGRAM
instagram.com/demotoprod

TIKTOK
tiktok.com/@demotoprod

Free to share. The live list, with every source link: indica-tech.com/checklists/#structured-outputs