

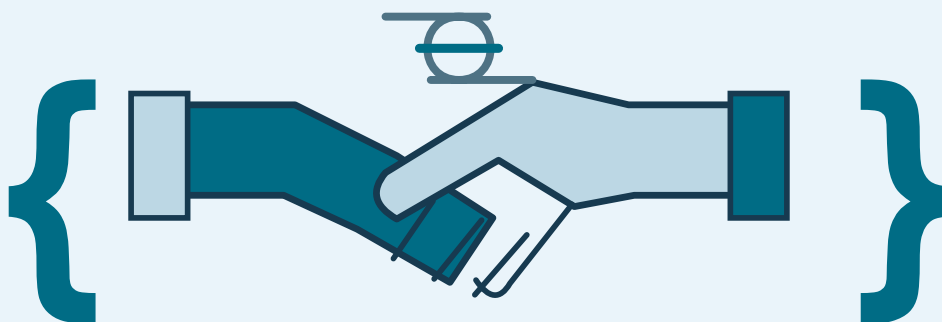


---

PYTHON AND DATA WORKFLOWS

# Design and Test API Data Contracts

Validate envelopes, preserve consumer promises, and make change policies explicit



---

**Sanjaay Gowda T S**

FREE SAMPLE / INTRODUCTION + CHAPTER 1 / EDITION 1.0.0

[sanjaaygowdatai.com](https://sanjaaygowdatai.com)

# About this free sample

## Design and Test API Data Contracts

Sanjaay Gowda T S

First edition 1.0.0, 2026-10-13

Copyright 2026 Sanjaay Gowda T S. This is a labelled free sample, not the complete ebook. You may share this sample unchanged. It contains the cover, sample permissions, contents, introduction, Chapter 1 and a closing product link. The remaining chapters, worked solutions and companion files are not included. The full ebook text, diagrams and artwork have separate single-reader terms; this sample permission does not authorize sharing the full purchased PDF or EPUB.

The companion source code, tests and fictional data are separately MIT licensed. You may adapt or share that project with its included licence notice. MIT does not apply to the ebook text or book artwork. The books and companion projects have no technical activation requirement or buyer-specific watermark.

This first edition was prepared with AI assistance. The case studies are fictional teaching examples. Any measured outputs come from the supplied examples, not production customer data. The teaching scope, assumptions and limitations are explained in the chapters. Independent human technical, editorial and assistive-technology reader review is pending. Automated tests and accessibility checks do not establish certification or suitability for every use.

Run examples in a separate working folder using the supplied synthetic data. Review inputs and calculations before adapting the techniques to real operations. Model software, service access, subscriptions, ongoing updates and consulting are not included.

Author website: [sanjaaygowdats.ai](https://sanjaaygowdats.ai)

Ebook terms: [sanjaaygowdats.ai/ebooks/license](https://sanjaaygowdats.ai/ebooks/license)

# Inside this free sample

|                                                   |   |
|---------------------------------------------------|---|
| Start here.....                                   | 4 |
| 1. Define the boundary and consumer promise ..... | 5 |
| Continue with the full guide .....                | 6 |

This free sample contains the introduction and Chapter 1 only. The closing link opens the complete guide. The companion project and worked solutions are supplied with the complete ebook.

# Start here

An API data contract describes promises between producers and consumers. It says what a message contains, what values mean, how related messages connect and which changes existing participants can tolerate. A valid JSON document is only the beginning. It can have the wrong fields, refer to the wrong request or represent an unexpected business state while remaining perfectly parseable.

This book builds an offline teaching audit for Juniper Dispatch, a fictional service accepting packet requests. Six synthetic request/response pairs exercise current and older versions, missing fields, correlation errors, unknown statuses and unsupported versions. Every organization, identity, envelope, quantity and proposed release decision is teaching material. No real API is contacted and no production reliability claim is made.

You will specify a flat envelope, validate request and response together, design useful error reports, distinguish payload checks from HTTP behavior and review a narrow compatibility policy. The companion compares a previous contract with a current contract and deliberately finds a broken response promise despite a minor-looking version increase. This makes a key lesson concrete: a version label does not make a change compatible.

The project requires Python 3.10 or later and uses only the standard library. It reads a local JSON file and prints an audit. It does not implement a web server, full JSON Schema, OpenAPI validation, authentication, authorization, persistence or retries. The limited code is useful because every rule can be inspected. A real system needs additional layers of specification, testing and operational review.

Read the nine chapters in order, complete each exercise and compare with the worked solutions. Keep the original fixture unchanged and edit copies for experiments. The final deliverables are a contract note, a fixture suite, a compatibility review and a migration checklist. Independent human technical and assistive-technology review is pending. Treat the examples as a learning method, not as a certified release gate for a consequential service.

# 1. Define the boundary and consumer promise

## Name who produces and consumes

Juniper Dispatch receives a request with version, request\_id and customer\_id. It returns a response with version, request\_id and status. The caller produces requests and consumes responses. The service consumes requests and produces responses. This direction matters because a change that is convenient for one side can break the other.

Write a consumer promise in plain language. Under the teaching policy, old callers can continue sending the fields required by their supported version, responses preserve previously promised required fields and consumers tolerate extra fields but reject unknown status values. This is a deliberately cautious policy for the fixture. Other systems may choose different rules, so compatibility must be evaluated against an explicit policy rather than a universal slogan.

## Separate contract layers

A payload contract covers shape and values. Cross-message rules cover relationships such as correlation IDs and version echo. HTTP covers methods, statuses, headers and representation handling. Business behavior covers whether the requested action actually occurs, how duplicates behave and what acceptance means. Passing the first layer cannot establish the rest.

OpenAPI provides a language-independent description of HTTP API interfaces, including request, response and schema structures [R1]. This book references a fixed specification edition, 3.1.1, rather than claiming it is the latest release. The teaching JSON contract is a smaller original format and is not an OpenAPI document. Its simplicity supports learning without implying standards conformance.

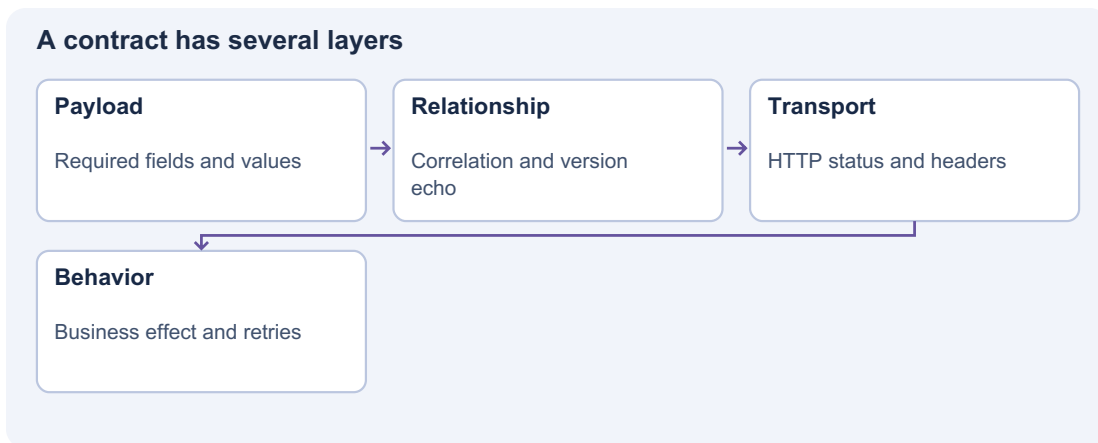


Figure 1. Payload shape, related messages, HTTP behavior and business effects require different checks.

## State the scope and exclusions

The fixture uses flat required fields whose values must be nonempty strings. Extra fields are tolerated and not deeply validated. Versions use a major.minor string without leading zeroes. The current major and current-or-older minor versions are supported. Responses must echo the request's version and `request_id`. Status must be accepted or rejected.

Write these rules beside the examples. Without them, a reader may infer behavior from one successful message and treat accidental details as promises. Also list exclusions: no nested schemas, network traffic, HTTP validation or security testing. A boundary note makes the audit useful by telling the reviewer which questions still require another tool or process.

## Exercise 1: write the promise

Write the producer and consumer roles for request and response. Draft a three-sentence contract promise covering required fields, extra fields and status values. Include one behavior that the offline audit cannot establish even if every message passes.

# Continue with the full guide

This free sample ends after the introduction and Chapter 1. The complete guide contains nine teaching chapters, worked exercise solutions, primary references and a separate Python companion project. The full PDF, reflowable EPUB and project files are supplied together.

Use the complete guide to work through the remaining exercises and inspect the supplied examples. The companion code and fictional data are separately MIT licensed; the complete ebook has the single-reader terms described on its product page.

The examples are teaching fixtures, not production customer results. Independent human technical, editorial and assistive-technology reader review remains pending. Software access, subscriptions, consulting and guaranteed results are not included.

[View the complete guide and purchase options.](#)

You may share this labelled free sample unchanged. Keep the full purchased PDF and EPUB private. Thank you for reading.

## Primary references

### 1. [R1] OpenAPI Specification 3.1.1

<https://spec.openapis.org/oas/v3.1.1.html>

An API description can document request, response and schema structures. This book deliberately references a fixed specification edition, not the latest release.