Guide to Implementing Criterion OpenAPI Specification

Version: **1.0**
Date: **29 11 2023**
Distribution: **Industry**
Document Name: **Guide_to_Implementing_Criterion_OpenAPI_Specification.pdf**

DISCLAIMER

Criterion believes it has employed personnel using reasonable skill and care in the creation of this document. However, this document is provided to the reader 'as is' without any warranty (express or implied) as to accuracy or completeness. Criterion cannot be held liable for any errors or omissions in this document or any losses, damages or expenses arising consequent to the use of this document by the reader.

CHANGE HISTORY

DATE	VERSION	STATUS
29 November 2023	1.0	

CONTENTS

1 HOW TO USE CRITERION OPEN API SPECIFICATION.

Criterion do not host services. The host of these services will be defined as 'service provider' and the parties calling the end point will be defined as 'service client'. Middle third parties are seen as being both service provider and service client. The added value of these third parties is generally out of scope for Criterion. However, Criterion does recognise the value there.

SERVICE PROVIDER

The OpenAPI Specification is designed to be downloaded, edited and then hosted by the service provider.

The service provider must:

- Keep reference to Criterion, including links and copyright information;
- The Criterion version of the resources must be included alongside each endpoint.

The service provider should:

- Use the version number in the server section to reflect the service provider's internal versioning;
- Keep the Criterion tag names;
- Not change any of the data structures or their meaning (see section *TRADING PARTNER SPECIFIC DATA* below);
- Send a copy of the OAS (OpenAPI Specification) to Criterion to help aid future development (under NDA if required);
- Inform Criterion of any problems found;
- Make their versions of adapted standards available to their clients (under limitations set out in the licensing contract);
- Use the hypermedia links where provided as a primary way to extend and layout the offering;
- Where REST IDs are defined, these should be used as surrogates to human identifiable identifiers.

The service provider could:

- Add in digital signatures;
- Add additional commentary to descriptions;
- Set up an OAS SwaggerUI (or other) test interface for use by their clients (under limitations set out in the licensing contract);
- Provide examples of usage (under limitations set out in the licensing contract);
- Integrate the Criterion OAS with other OAS in the same domain when appropriate.

SERVICE CLIENT

The client should:

- Dominantly use the OAS of the service providers not Criterion's OAS directly in implementation;
- Use Criterion's OAS as a Rosetta stone when dealing with multiple service providers;
- Use Criterion's OAS when initially evaluating the need for the Standard;
- Use Criterion's OAS prior to service provider engagement;
- Use Criterion's OAS prior to service provider implementation;
- Request changes to be made by Criterion where appropriate, rather than directly with multiple service providers;
- Use the hypermedia links where provided, rather than statically storing URLs.

2 OPENAPI DEFINITION

OpenAPI definitions within Criterion Standards (as specified in accordance with the OpenAPI Specification https://swagger.io/docs/specification/about/) provide a machine-readable API definition, allowing quicker service implementations. OpenAPI definitions consist of the components detailed in the table below. The "prescriptive" column indicates if the content supplied by Criterion for this Standard is prescriptive or not.

OPENAPI COMPONENT	DETAILS	PRESCRIPTIVE
openapi	The OpenAPI specification version used. See https://swagger.io/specification/#appendix-a-revision-history	Yes
info	High-level description of the Criterion Standard: • Title; • Description; • version (specified in Criterion Standards Versioning Policy terms, e.g. v1.0.)	Yes
security	The security scheme for securing the service must be agreed between the trading partners and specified accordingly here.	No
paths	Contains the paths/operations available in the API: • defining individual endpoints in the API, and the HTTP methods supported by these endpoints; • includes all the details of the message exchange patterns and refers to the input/output data structures for each message; • defines the handled return code values (via RFC 7807); • can support semantic versioning (https://semver.org/). The service provider should remove any endpoints they do not wish to implement. The service provider should produce specific error messages including when functionality that is not supported is used. The service provider should consult with Criterion before extending into new end points, to aid in a universal approach.	No
server	The service provider must specify the API server and base URL. More than one server can be defined, for example one for production and one for a sandbox server. All paths are relative to server URLs. This cannot be prescriptive for obvious reasons. e.g. https://api.serviceprovider.com/	No
components/ schemas	Defines common data structures (schemas) referred to elsewhere in the API definition e.g., the paths/operations can refer to schema components for definitions of the message structures.	Yes
components/ securitySchemes	Defines common security schemes that can be used by the API's operations.	No

OpenAPI components that are NOT prescriptive can be customised for the specific requirements of an implementation. The API definition supplied as part of the Criterion Standard can be used as a basis for implementing a service conforming to the Criterion Standard. There will be no specific entries relating to securing the service or semantic versioning of the service operations/paths.

Trading partner specific HTTP headers can be supplied as part of the customisation of the OpenAPI definition.

3 GENERAL NOTES

EMPTY DATA ITEMS

The schema validation rules ensure that no empty data items are sent.

TRADING PARTNER SPECIFIC DATA

If trading partners which to exchange data which is not included in the OpenAPI spec, Criterion have facilitated the use of trading partner specific data containers (called tpsData).

By using a named tpsData container, it is obvious where trading partner specific data is being deployed. Trading partners can use this extensibility feature in order to exchange information which is sensitive in competitive terms, or which has not been supported in the definition of this version of the Standard yet.

Some specs already have tpsData objects included as optional objects spread throughout the business data payload of the request and response messages. However, users of Standards are welcome to insert tpsData objects as required if not already present. These tpsData containers should appear as the last property in a particular object definition.