Ostia
Your API Strategy Partner

No-code API/AI Sandboxing Introduction

This product is designed to increase the speed of the development of APIs or MCP Tools but is also focussed on improving the development experience by promoting reuse of resources, standardization of names and strictly enforcing those standards. We clarify why this is important and how APIs are then created and quickly available for use.

Standardization

This is a key part of the product as it promotes:

- Consistency of usage and rules.
- Reuse of resources.
- Easier documentation.
- Better understanding for internal and external users of the services.

Field Names

Often organizations talk about a Taxonomy which, in simple words, is what things are called. At the base level in IT, a unit of data is identified at the Field or Property level with a 'name'.

In the past, developers have picked the name they prefer, which is fine for an individual developer, however, developers differ and the result is that multiple developers use different names for the same field. This can be as simple as using different cases in the name (e.g. AccountNumber vs accountnumber) or even different names (e.g. AccountNumber vs ACCT).

This results in confusion for others looking at different implementations and in the same validation or data generation routines being created. This is addressed by creating 'Field Templates' within the product. These Field Templates have the following characteristics:

- They have a Unique name to identify a specific unit of data.
- They have specific validation rules to validate data provided for each field.
- They have a specific data generation rule to generate validate data for that field when required.

This means that a field is defined and configured once and then reused across the Section thus avoiding duplication or confusion.

Schema Names

The product supports the creation of multiple Schemas which are created based on the Field Templates. In the same way as duplication occurs in the naming and usage of Schemas, creating Schema templates ensures the consistent use and, more importantly, reuse of Schemas that represent the same object. This also ensures that a single change to one Schema will be reflected everywhere that is used.

Data Object Schemas

The product uses a specific type of Schema for Data Objects. For most APIs, it will be designed to Get, Create, Update or Delete some data as part of its process. This is managed by creating specific Data Object Schema Templates that can then be used across Digital Capabilities and can ensure that every business object can be created in a standard way to promote reuse across the Eco-system.

Error Schema

Good error processing and clear error messages are key to creating APIs that can be used by internal and external developers. Different organization have their own views on what information should be contained in an error message. The product allows the creation of a standard Error Schema for error messages based on the users' requirements but then all error messages must follow this format for consistency.

Product Usage

The product requires some initial setup for the first APIs to be created but once completed, the creation of APIs will become simpler and faster as each additional API creation effort will build on the previous work.

Section Creation

A Section must be created and then the following activities must be completed:

1. Review and update the Error Schema to create the error message structure required for this project.
2. Add any additional Error Templates based on this schema. (Note, further error templates may be added at any time so these can be added as needed also).
3. Review and update the JWT Security Schema to include any specific custom claims that may be needed by the API.

This must only be completed once per Section. Ostia recommend creating a single Taxonomy for an Organization and thus managing everything in one Section. Other

Sections should be added only if Taxonomy conflicts between projects cannot be resolved or for test purposes.

Field Template, Schema and Data Object Creation

This creates the basis of your Taxonomy for this Section and, if possible, for your organization. It is possible to create these from the following:

- Swagger/OpenAPI Documents
- COBOL Copybooks: No schemas are created from this.
- XSDs: No schemas are created from this.
- Manual: All resources are added manually.

This must be done once per project; however, it should become a quicker process for each project as many Field Templates, Schemas and/or Data Objects will already exist.

Business Capability Creation

This is where the APIs or Digital Capabilities are created for a given project. A Business Capability is a collection of related APIs/Digital Capabilities that are then available for use. Ostia recommend that APIs/Digital Capabilities are simple, single function services that do one specific thing and should only be associated with one primary Data Object Schema but may refer to others. The base line Digital Capabilities create an expected 'Happy Path' and can be created in two ways:

- From the Swagger/OpenAPI document where one Digital Capability per endpoint will be created.
- Based on an existing Data Object Schema. The process allows you to select what HTTP Method(s) to create as base line Digital Capabilities.

Once created, each Digital Capability can be configured to include error paths, business rules related to data access or other functional requirements.

Conclusion

Standardization in the product intentionally creates guard rails around naming and configuration to force developers to focus on the genuine business requirements rather than the naming of different resources. This focus can ensure that each newer set of APIs will be built more quickly due to the work that has gone before. In addition, as resources are reused, they are further tested and issues fixed and more importantly, remain fixed going forward.