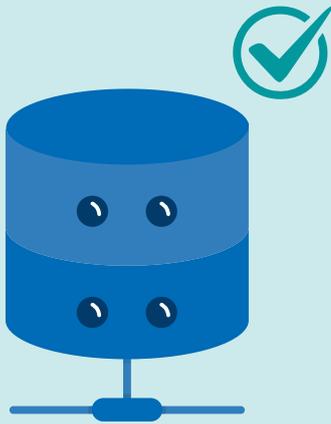


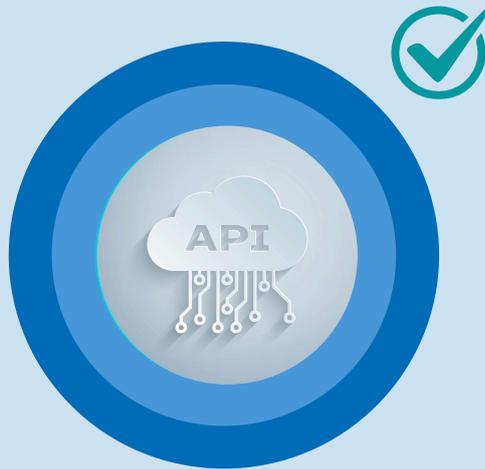
N-CAPIE

NO-CODE API ECOSYSTEM
DESIGN & DEPLOYMENT

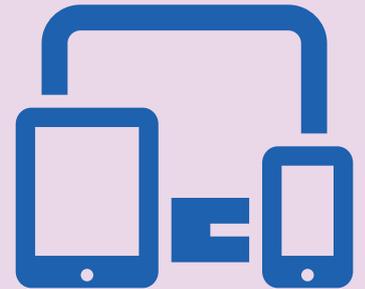
Servers & Databases



APIs & Business Logic



Frontend Applications



Simplify and Improve

WELCOME MESSAGE

N-CAPIE is a standards based No-Code SaaS service designed to simplify and improve Enterprise Level API Ecosystems.

Key Benefits:

- **Collaboration:** Enhances collaboration across teams and partners
- **Innovation:** Improves capacity for innovation
- **Speed:** Ensures quicker deployment/Go-To-Market times
- **Strategy:** Facilitates creation and implementation of API strategy
- **User Experience:** Improves partner and developer experience
- **Auditing:** Simplifies internal and external auditing
- **Resource Optimisation:** Optimises resource allocation and use
- **Consistency:** Uses identical configuration across all Environments
- **One-To-Many:** One backend to service many frontends
- **Cost:** Greatly reduce the cost of API Ecosystem design and deployment

We are committed to helping you improve your API Ecosystem design and deployment.

Best Regards,

John Power

CEO

Leo Cullen

Head of Business Development

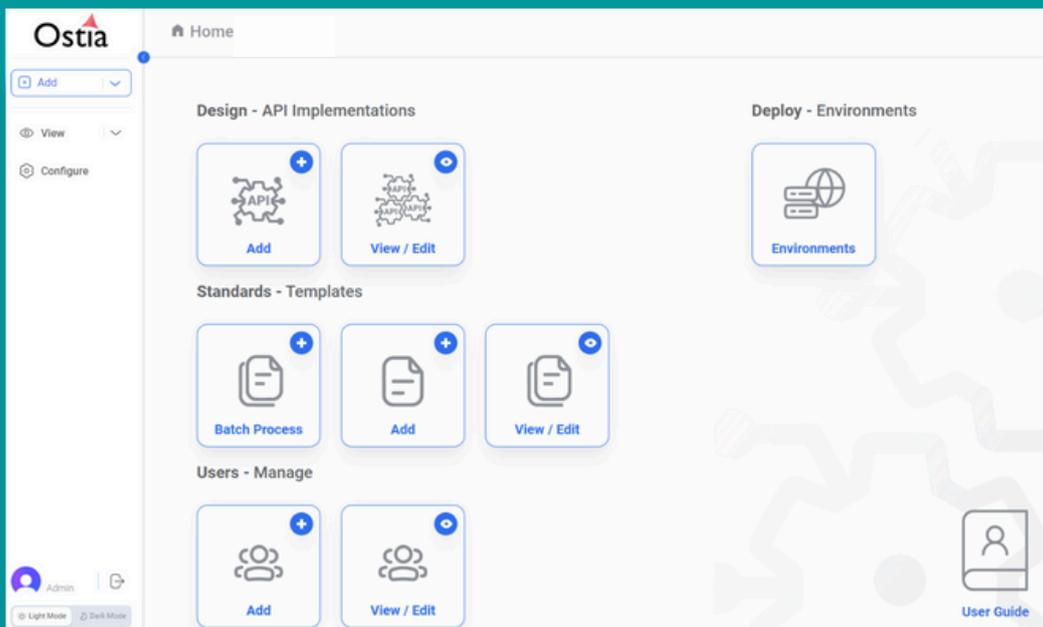
TABLE OF CONTENTS

N-CAPIE

01	Using the Service	>	1
02	Design Section	>	4
03	Deploy Section	>	21
04	Standards Section	>	23
05	User Management	>	27
06	Technology & Security	>	31

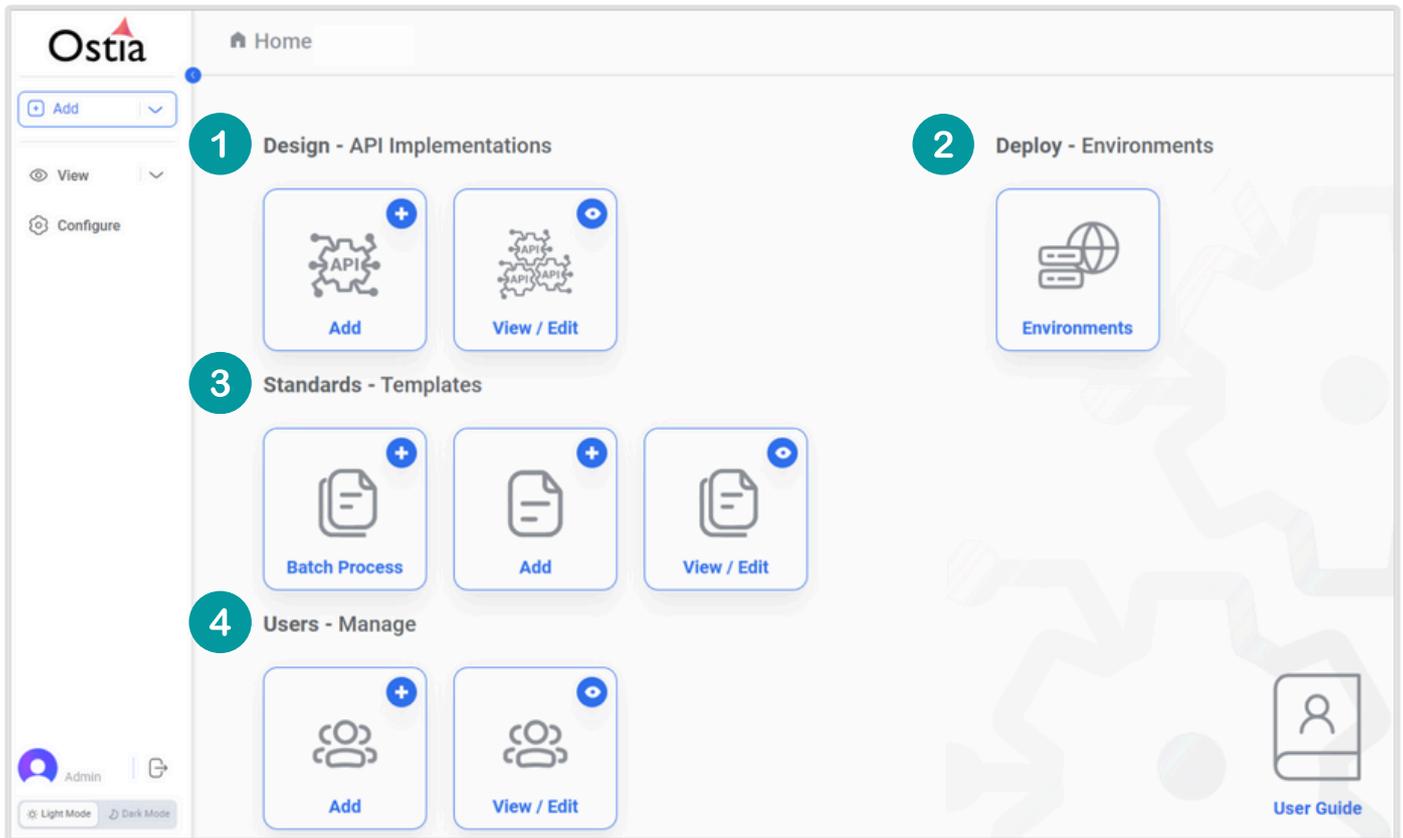
Using the Service

A simple and intuitive interface makes the design and deployment of your API Ecosystem as straightforward as possible.



Contents and Navigation

The Home page offers easy access to the Design, Deploy, Standards and User Management sections of the service.



1 Design - API Implementations

The Design section enables you to Add, View and Edit API Implementations: sets of Methods/Endpoints that offer specific business capabilities.

2 Deploy - Environments

The Deploy section enables you to sequentially promote completed API Implementation Designs into Sandbox, Test and Production Environments.

3 Standards - Templates

The Standards section enables you to configure templates for use across all projects: the templates are the core building blocks of your API Ecosystem.

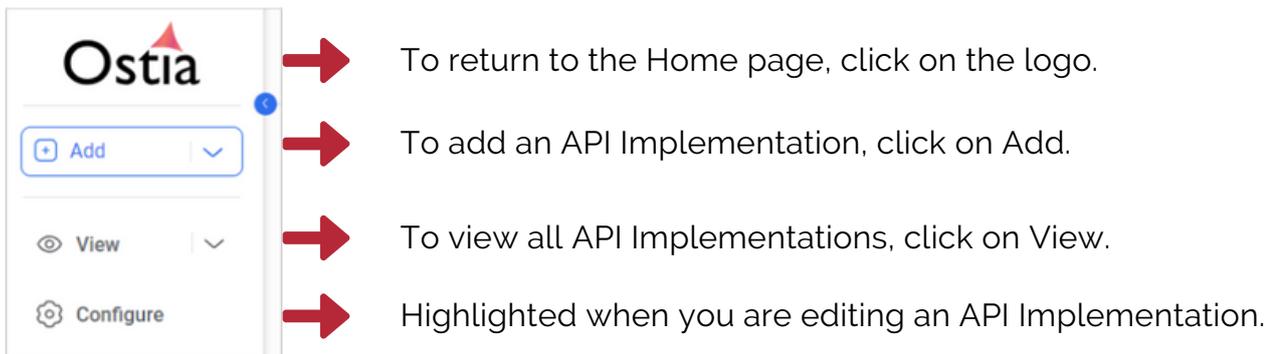
4 Users - Manage

The User Management section allows you to add and manage Teams and Users within the service.

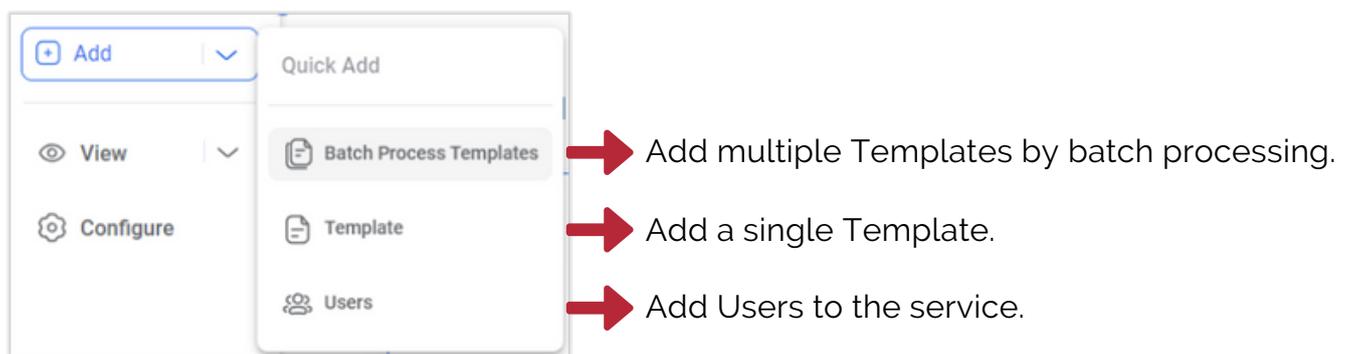
USING THE SERVICE

Side Menu

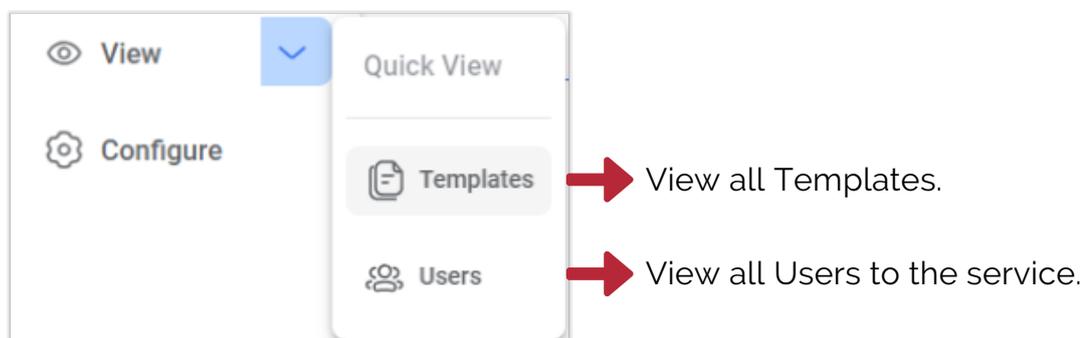
Add/View and Edit functionality is also available using the collapsible side menu.



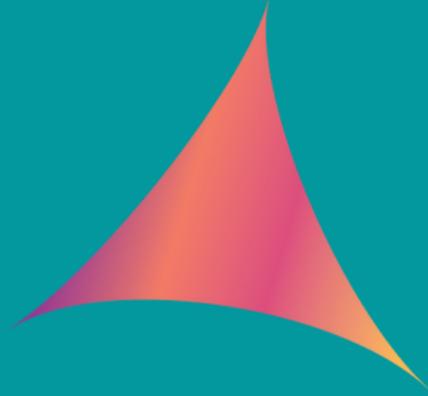
There is additional 'Quick Add' functionality available through the dropdown on the Add button.



Similarly, there is additional 'Quick View' functionality available through the dropdown on the View button.



DESIGN SECTION



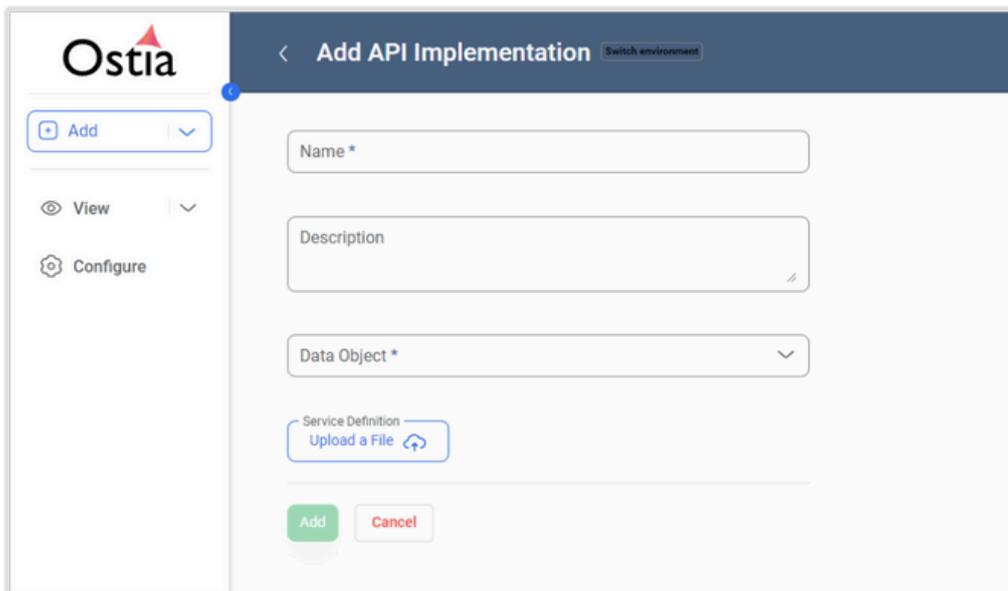
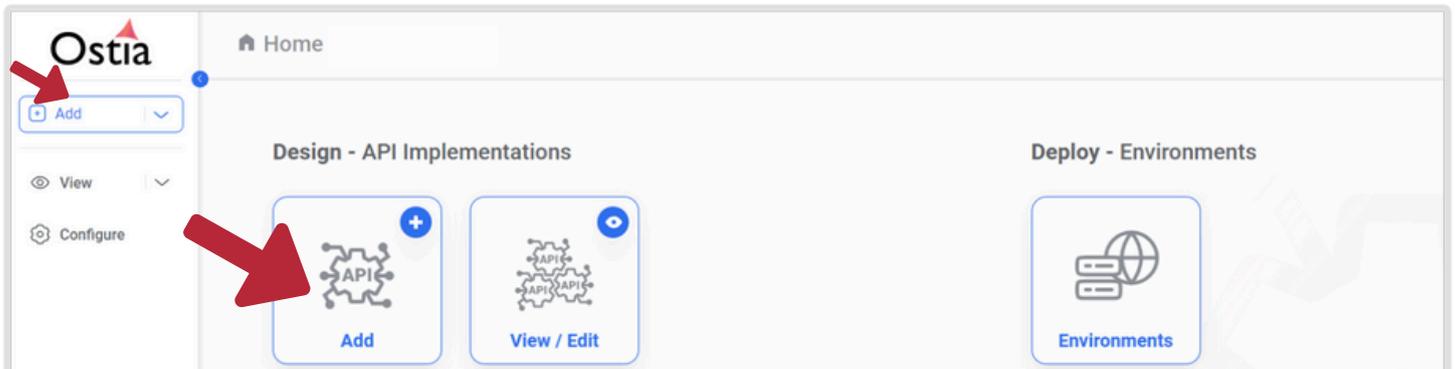
Our No-Code configuration allows you to quickly design API Implementations that meet all of your business logic requirements.

The screenshot displays the Ostia API design interface. At the top, it shows 'API Implementation: Parameters' and 'Methods: 6'. A search bar is present with the text 'Search Method'. On the left, there are navigation options: 'Add', 'View', and 'Configure'. The main area contains a table of API methods with columns for Name, Description, Type, Endpoint, and Steps. The table lists six methods: getPartnerLimit, putAccountLimit, putPartnerLimit, postAccountLimit, postTransferAcceptorLimit, and putTransferAcceptorLimit. Each method row includes a checkbox, a description, a type (GET, PUT, POST), an endpoint, and a number of steps. At the bottom of the table, there are navigation buttons for 'Previous' and 'Next', and a 'List Mode' dropdown set to 'Paginated'. The bottom left corner shows a user profile for 'Admin' and toggle switches for 'Light Mode' and 'Dark Mode'.

#	Name	Description	Type	Endpoint	Steps
<input type="checkbox"/>	getPartnerLimit	Retrieve the limits configurations and daily usage information for a partner.	GET	/[partner_id]/limits	5
<input type="checkbox"/>	putAccountLimit	Update account limits for a partner.	PUT	/[partner_id]/limits/account-limits/{account_limit_reference}	7
<input type="checkbox"/>	putPartnerLimit	Update the daily limits and alert settings for a partner.	PUT	/[partner_id]/limits/daily-limits/{partner_limit_reference}	7
<input type="checkbox"/>	postAccountLimit	Add account limits for a partner.	POST	/[partner_id]/limits/daily-limits/account-limits	4
<input type="checkbox"/>	postTransferAcceptorLimit	Add daily limits and alert settings for a Transfer Acceptor.	POST	/[partner_id]/limits/daily-limits/transfer-acceptor-limits	4
<input type="checkbox"/>	putTransferAcceptorLimit	Update daily limits and alert settings for a Transfer Acceptor.	PUT	/[partner_id]/limits/transfer-acceptor-limits/{transfer_accep...	7

Add API Implementation

API Implementations are added within the service using an OpenAPI Specification document, as follows:



1. Name the API Implementation you are creating (maximum 20 characters).
2. Provide a succinct description of its purpose (maximum 100 characters).
3. Choose the 'Data Object' relating to the API Implementation.
4. Select the requisite OpenAPI Specification document (aka Swagger).
5. Click Add.
6. When the operation is processed you will be ready to edit the 'baseline API Implementation'.

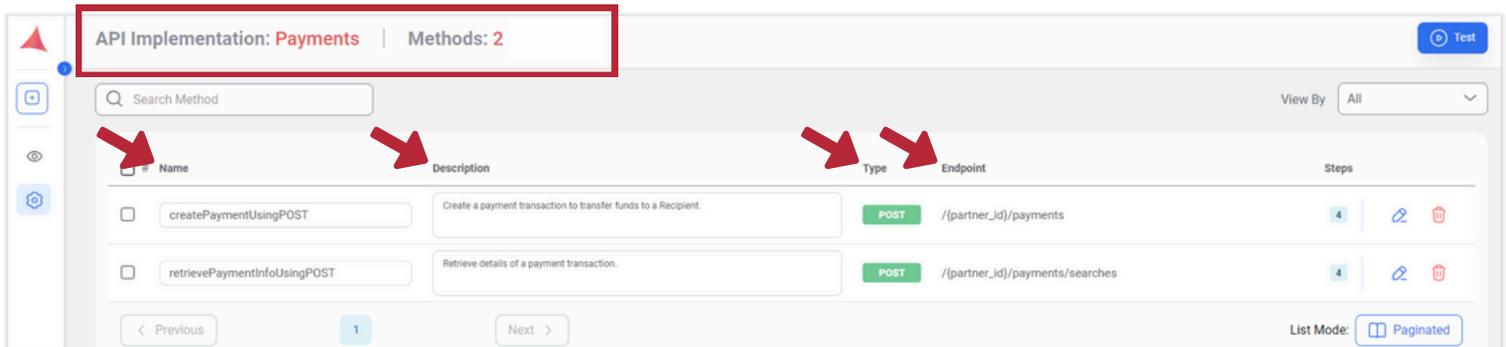
NB: Onboarding must be carried to enable creation of API Implementations. To ensure standardised design across Enterprise organisations, Field Rule and Data Object Templates must be configured before API Implementations can be created / edited.

DESIGN SECTION

Edit API Implementation

When the OpenAPI Specification document has been processed, at the top of the screen you see the Name of the API Implementation and the number of Methods it contains.

In the example illustrated here, you will see that the API Implementation is called "Payments" and it contains 2 'Post' Methods.



The information provided, includes:

1. The **Name** of each Method ([editable](#)).
2. The **Description** of each Method ([editable](#)).
3. The **Type** of Method - Post, Put, Patch, Get or Delete.
4. The **Endpoint** to which the Method relates.
5. The number of '**Steps**' taken to configure the Actions/Conditions of the Method.
The number of Steps is an indication of the relative complexity of configuration.

After reviewing/completing this information, you can then choose the Method you wish to edit.

Tools:

Test: Allows you to test the Endpoints of the Methods.

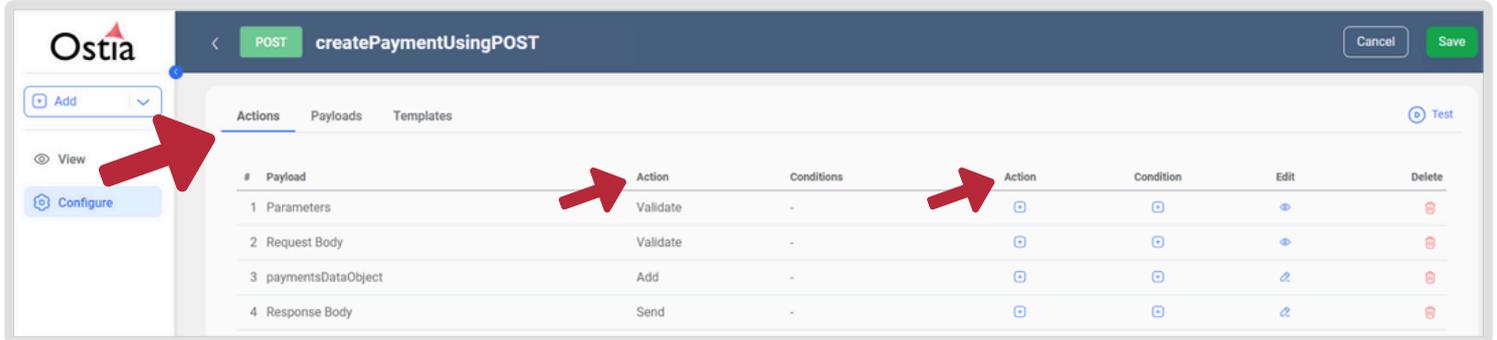
Search: Allows you to quickly search for Methods.

View By: Allows you to view selected Methods.

DESIGN SECTION

Edit Method - Actions

When you select the Method that you wish to edit, you see three tabs: Actions, Payloads and Templates.



The Actions tab allows you to configure all of the Actions and Conditions related to the Payloads within the selected Method.

The first time that you enter into this section, you will see the baseline configuration for the Method. You can then add further Actions and Conditions to build out the business logic, as required.

Add an Action

To add an Action to a Payload, you will choose where you wish to add the Action and click the '+' symbol. When you do so, an edit window appears to the left of the screen.

Each Payload has the following actions available to it:

Parameters: [Validate](#) (Field Data)

Request Body: [Validate](#) (Field Data)

Response Body: [Send](#) (Response)

Errors: [Send](#) (Error)

Data Object: [Add](#), [Update](#), [Get](#),
 [Get Multiple](#), [Delete](#)

NOTE: A Data Object is a business object that you wish to add to, update, retrieve from, delete from the database. You will note that Data Objects have 5 related Actions, while the other payloads have 1 each.

Add an Action (Cont'd)

There are 2 further 'Payloads' that can be selected when configuring Actions, these are:

1. Organisation Data Objects
2. External APIs

(These Payloads are configured/created in the 'Templates Section' of the service and enable access to data sources both within and outside of your organisation.)

Organisation Data Objects

Organisation Data Objects allow access to data sources from within your organisation that can be used in the configuration of the business logic and functionality of a Method.

External APIs

External APIs are pre-defined API calls that allow you to access data and functionality from third-party providers, or to access data sources within 'legacy systems' in your own organisation (legacy system access requires an application installation on your mainframe).

When you select either of these 2 payloads, you will see an additional dropdown box, from which to select the requisite Data Object or External API before configuring the associated Action or Method.

These 2 Payloads have the following Actions available to them:

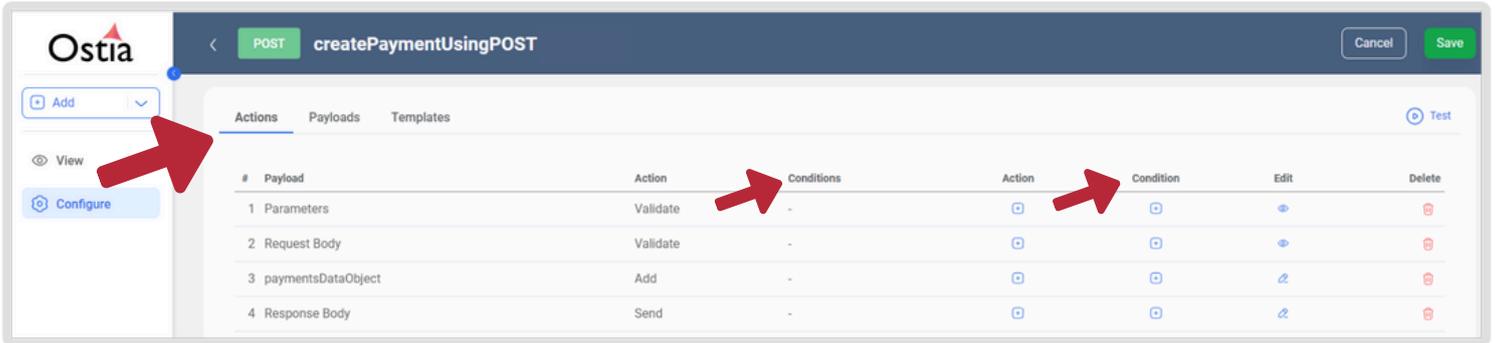
Organisation Data Object: [Add](#), [Update](#), [Get](#), [Get Multiple](#), [Delete](#)

External API: (The Pre-defined Method: [Post](#), [Put](#), [Patch](#) [Get](#), [Delete](#))

A large, light gray rectangular area containing the word "Placeholder" in a large, white, sans-serif font, centered within the box.A large, light gray rectangular area containing the word "Placeholder" in a large, white, sans-serif font, centered within the box.

DESIGN SECTION

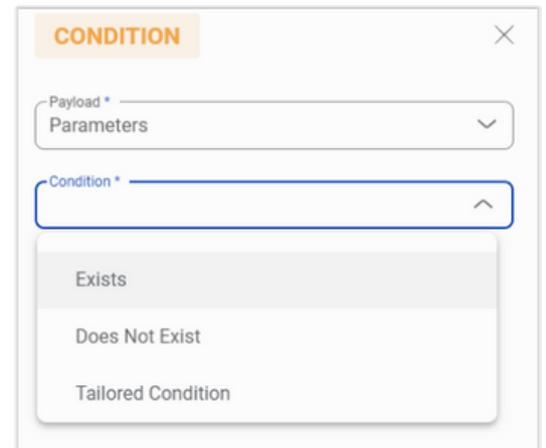
Add a Condition



To add a **Condition** to a Payload, click the '+' symbol where you wish the Condition to be added. When you do so, an edit window appears to the right of the screen.

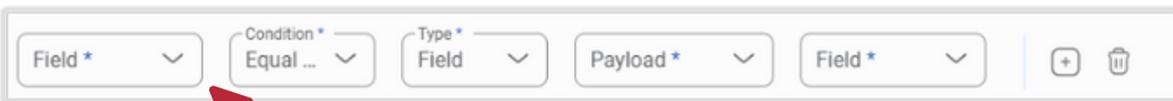
3 'Condition' options exist for each Payload:

- Exists (relates to the Payload)
- Does Not Exist (relates to the Payload)
- Tailored Condition (relates to a Field in the Payload)



Tailored Condition

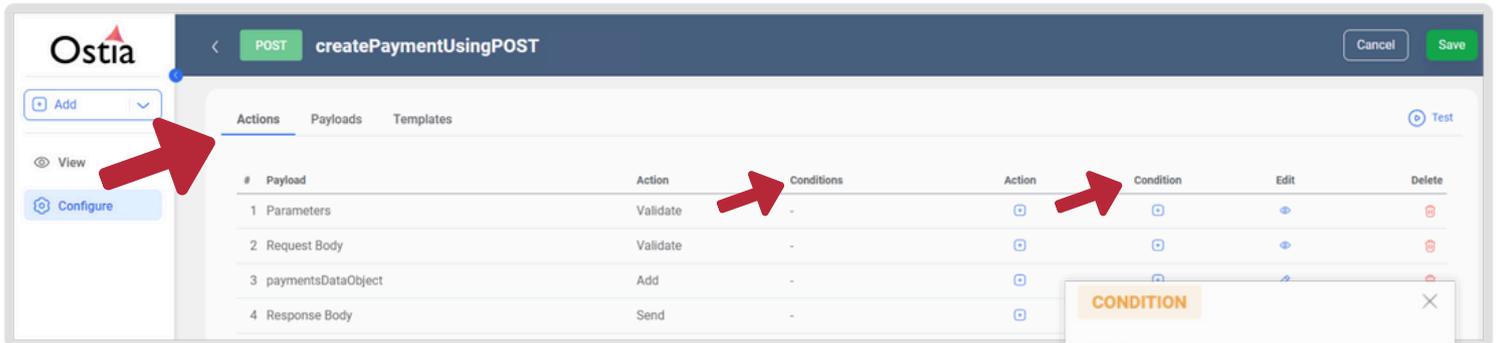
When you select Tailored Conditions, the Fields and Conditions that become available to you are as follows:



Equal To:	Value is equal to the comparative value
Not Equal To:	Value is not equal to the comparative value
Less Than:	Value is less than the comparative value
Less Than or Equal To:	Value is less than or equal to the comparative value
Greater Than:	Value is greater than the comparative value
Greater Than or Equal To:	Value is greater than or equal to the comparative value
Starts With:	Value starts with the comparative value
Ends With:	Value ends with the comparative value
Contains:	Value contains with the comparative value
Within List:	Value is in the list provided as the comparative value
Not Within List:	Value is not in the list provided as the comparative value
Not Provided:	No value provided
Provided:	Value provided

DESIGN SECTION

Add a Condition (Cont'd)

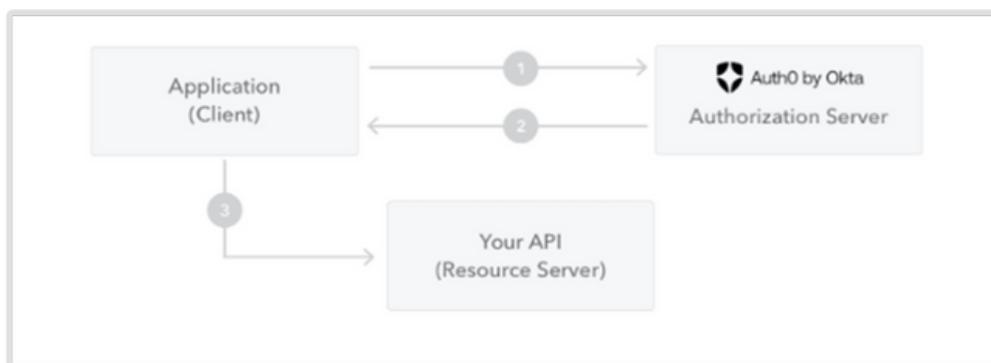


Within the Tailored Condition configuration, there is an additional Payload that becomes available for use as a data source: **Security - JWT**. There is no Action related to this Payload; it is present to allow you to select the Payload and related Field for comparative purposes.



This Payload relates to JSON Web Tokens (JWTs): a compact, URL-safe means of representing claims to be transferred between two parties. JWTs are very commonly used for User Authorization to govern access to resources.

They work as follows:



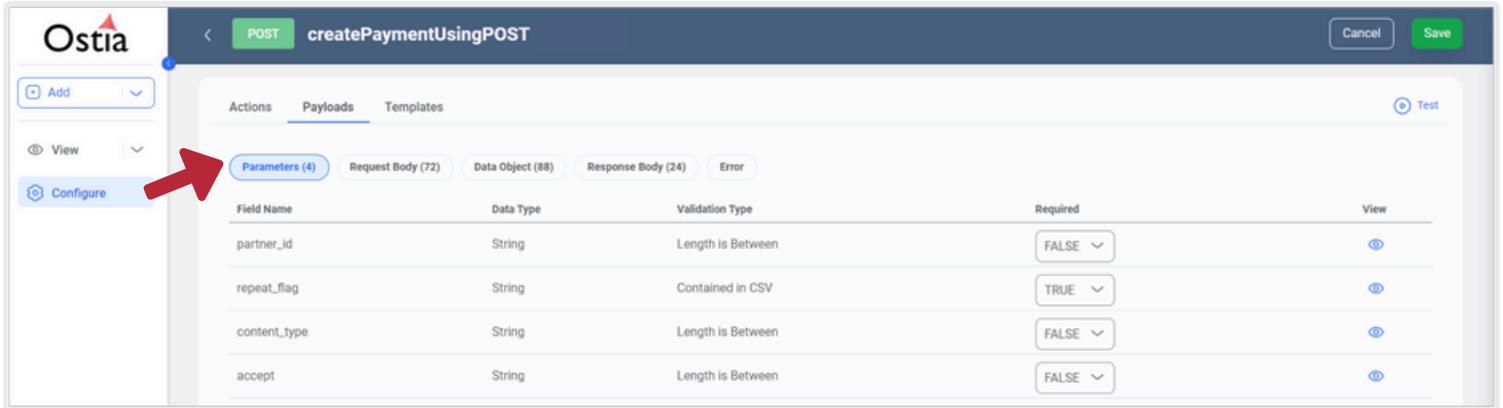
There are various Authentication/Authorisation service providers (including Okta as indicated here) - all can be used with this service.

The Security - JWT payload is created in the Standards section which we will review later in this document.

DESIGN SECTION

Edit Method - Payloads

The Payloads tab gives you access to the Method's Parameters, Request Body, Data Object, Response Body and Errors - and the number of Fields is indicated for each, where relevant. If a Payload is not present within a Method, its tab will be inactive.



Parameters

You can quickly see the number of Fields, and the content of the Parameters payload under the following headings:

- Field Name
- Data Type
- Validation Type

You can configure whether or not any field is required by selecting True/False (the only configurable item in Parameters).

There is also a 'read only' view of the 'Data Validation' and 'Data Generation' configuration of each Field (as illustrated to the right here).

partner_id

Description

Data Validation

Data 1

Data Type
String

Validation type
Length is Between

Minimum
32

Maximum
40

Error Type
Custom_Error

Custom Message

Data Generation

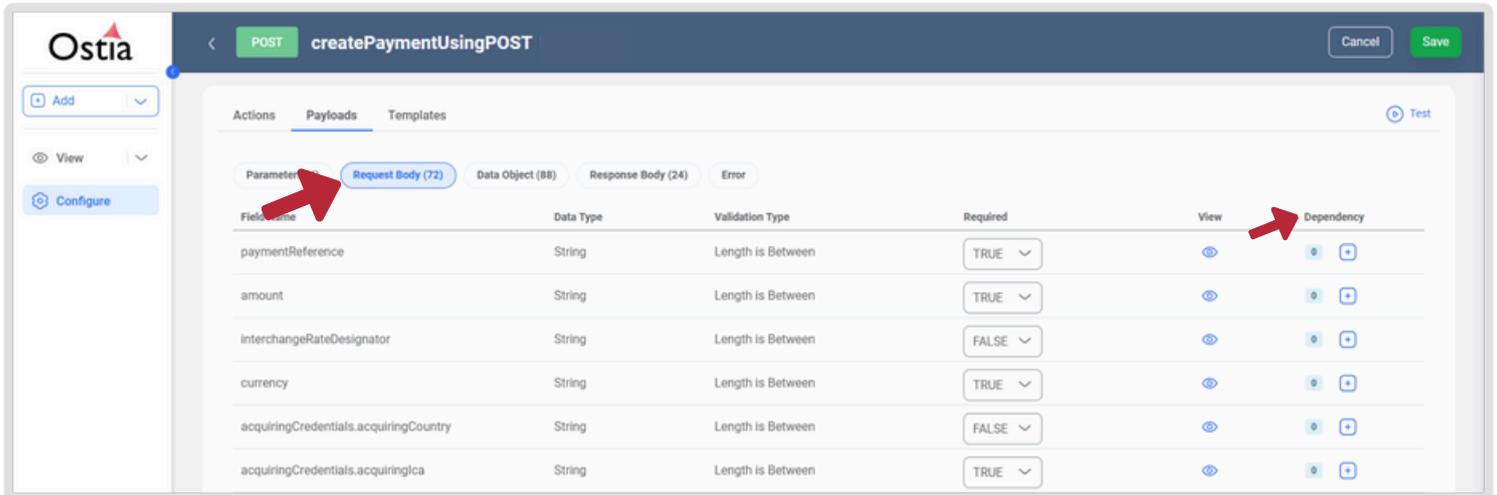
Data Type

Cancel Save

DESIGN SECTION

Request Body

The format of the Request Body Payload is identical to the Parameters Payload - and contains Field Name, Data Type and Validation Type, However, the Request Body typically has many more Fields and there is additional functionality to configure Dependencies, as required.



Dependencies

Dependencies are configured in the following manner:

1. Select the Field to which you wish to add a Dependency(ies) and click on the '+' symbol. An Edit Window will appear with the following fields for completion:

IF

THEN

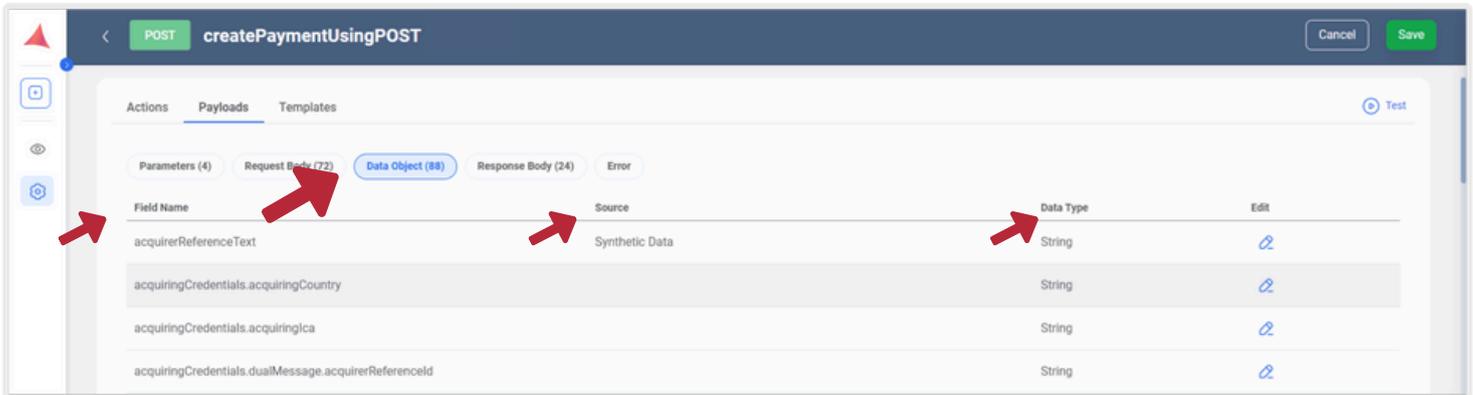
2. Complete the 'If Statement' by selecting the appropriate condition and stipulating the requisite Field or specifying the required Value amount.
3. Then choose the dependent Field(s) and configure the Data Validation requirements as appropriate. Within the configuration of the Field, there is also the option to select where the Field is Optional, Mandatory or Prohibited.

It is possible to add as many dependent Fields as required.

DESIGN SECTION

Data Object

The Data Object is a business object that you wish to add to, update, delete or retrieve from the database; and relate to POST, PUT, PATCH, DELETE or GET Methods respectively. Data Objects are created/configured in the Standards Section.



You can quickly see the number of Fields, and the content of the Data Object payload under the following headings:

- Field Name
- Source
- Validation Type

While the previous Payloads we have examined (Parameters and Request Body) are populated as part of the Request being made under the Method, the Data Object Fields' are populated from various Sources, namely:

1. **Parameters** (Fixed Field Value)
2. **Request Body** (Fixed Field Value)
3. **Synthetic Data** (Used where no Fixed Value is available)
4. **Organisation Data Object** (Fixed Field Value)
5. **External API Response** (Fixed Field Value)
6. **Multiple (Conditional)** (Fixed Field Value - subject to multiple conditions)

When an API Implementation is initially added to the service, the Source of each Data Object Field will be from (1-3 above) Parameters, Request Body or Synthetic Data.

In order for a Field to have the Source 'Organisation Data Object', 'External API' or 'Multiple (Conditional)', it must be specifically configured as such. (When Organisation Data Objects or External APIs are selected as part of the configuration in the Actions Tab, their constituent Fields become available to be used as a Source.)

DESIGN SECTION

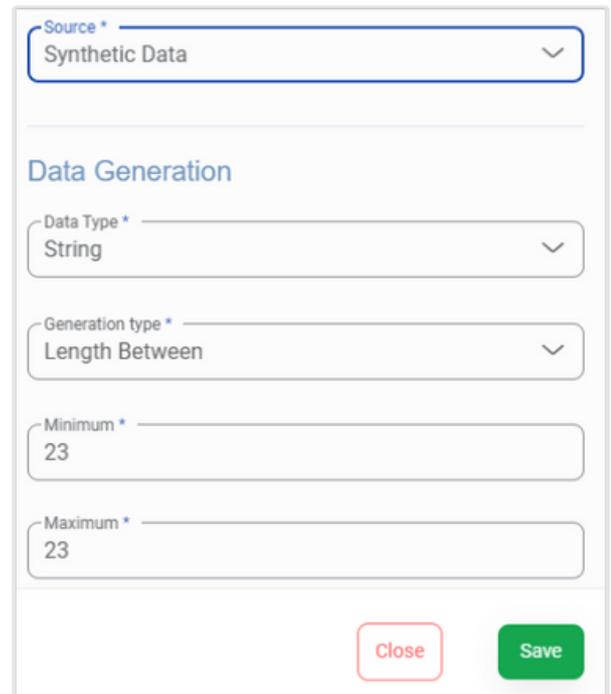
Data Object (Cont'd)

The Source of all Fields within the Data Object can be edited by clicking on the relevant Edit Button.

When Parameters, Request Body, Organisation Data Objects or External APIs are selected, it is a simple matter of selecting the Field that you wish to be the Source.

Where Synthetic Data is selected as the Source (as illustrated here), you will select the 'Data Type' and complete the Fields that are presented based on that selection.

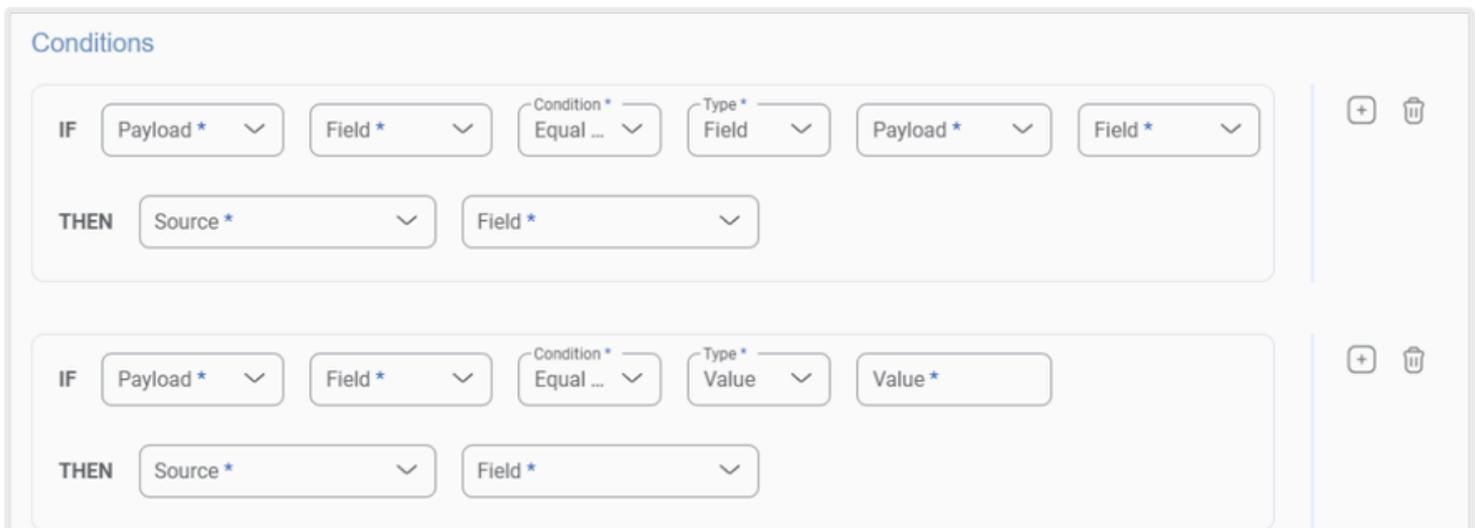
The Data Type options available for selection are: String, Date, Decimal, Numeric and Record.



The screenshot shows a configuration window for Synthetic Data. At the top, there is a dropdown menu labeled 'Source *' with 'Synthetic Data' selected. Below this is a section titled 'Data Generation'. It contains three dropdown menus: 'Data Type *' set to 'String', 'Generation type *' set to 'Length Between', and 'Minimum *' set to '23'. There is also a text input field for 'Maximum *' with '23' entered. At the bottom right, there are two buttons: a red 'Close' button and a green 'Save' button.

Multiple (Conditional) Source

If 'Multiple (Conditional)' is chosen as the Source, an Edit Window will appear with the following Fields to be completed.



The screenshot shows a configuration window titled 'Conditions'. It contains two rows of configuration options. Each row starts with 'IF' followed by a 'Payload *' dropdown, a 'Field *' dropdown, a 'Condition *' dropdown (set to 'Equal ...'), and a 'Type *' dropdown (set to 'Field'). The first row also has a second 'Payload *' dropdown and 'Field *' dropdown. The second row has a 'Value *' dropdown. Below each row is a 'THEN' section with a 'Source *' dropdown and a 'Field *' dropdown. On the right side of each row, there are '+' and trash icons.

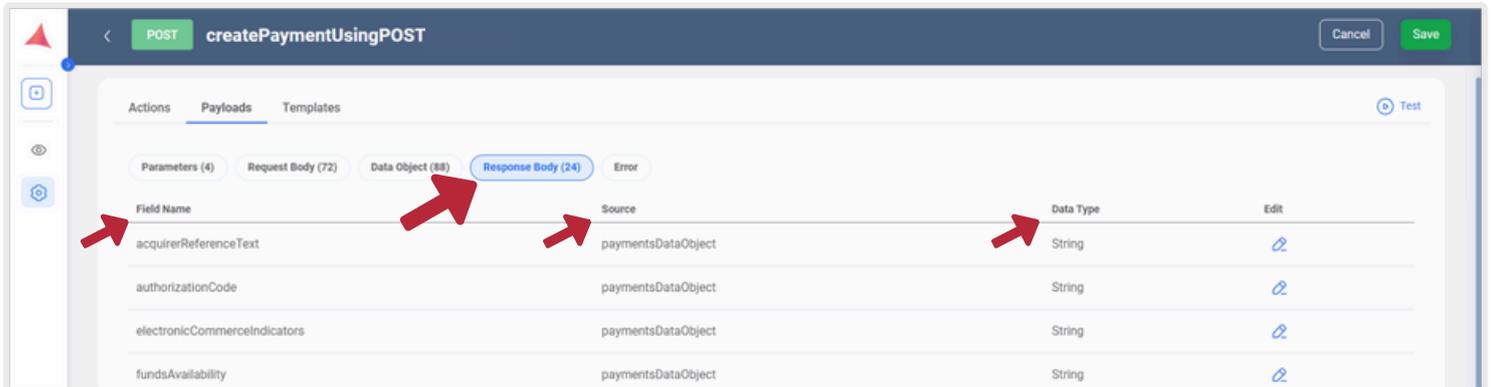
As the name suggests, the Source of the Field will be based upon the multiple conditions (at least 2) that are configured.

This Source-type covers situations where there are multiple options for the source of the Field data based on the data present in the Parameters and Request Body.

DESIGN SECTION

Response Body

The format, content and configuration of the Response Body mirrors the Data Object.



Again, you can quickly see the number of Fields, and the content of the Response Body payload under the following headings:

- Field Name
- Source
- Validation Type

While all of the data sources are still available for use, the Source of Fields in the Response Body will predominantly be the relevant Fields from the Data Object payload.

1. **Parameters** (Fixed Field Value)
2. **Request Body** (Fixed Field Value)
3. **Synthetic Data** (Used where no Fixed Value is available)
4. **Data Object** (Fixed Value - Predominantly Used)
5. **Organisation Data Object** (Fixed Field Value)
6. **External API Response** (Fixed Field Value)
7. **Multiple (Conditional)** (Fixed Field Value - subject to multiple conditions)

DESIGN SECTION

Errors

The Errors Payload displays the Errors that (can) relate to the specific Method type being viewed / edited.

In this example, we are looking at 2 of the relevant Errors for a **POST** Method:

1. Validation Error
2. Already Exists Error

As you can see, some Fields are editable and some are not.

The Fields that are editable have been created using Synthetic Data and can be edited on a per Method basis.

Errors are created and configured in the Standards section, which we will examine later. In this way, there is a Standard Error format for all projects across your organisation, which provides a consistent experience for users.

(The format and content of Errors is different from organisation-to-organisation; they are defined and configured within the service when a client is being onboarded.)

The screenshot displays two sections of an error configuration interface. The top section is titled 'Validation Error' and contains a table with the following fields: Field Name, Source, Error ID, Unique ID, Message, Fixed Value, Errors, Error Code, Generation Rule, Field Name, Generation Rule, and Error, Generation Rule. The bottom section is titled 'Already Exist' and contains a table with the following fields: Field Name, Source, Error ID, Unique Id, Message, Already Exists, Errors, Error Code, Field Name, Field Name, Error, and Resource already exists. Red arrows point to the 'Already Exists', 'Error Code', 'Field Name', and 'Resource already exists' input fields, indicating they are editable.

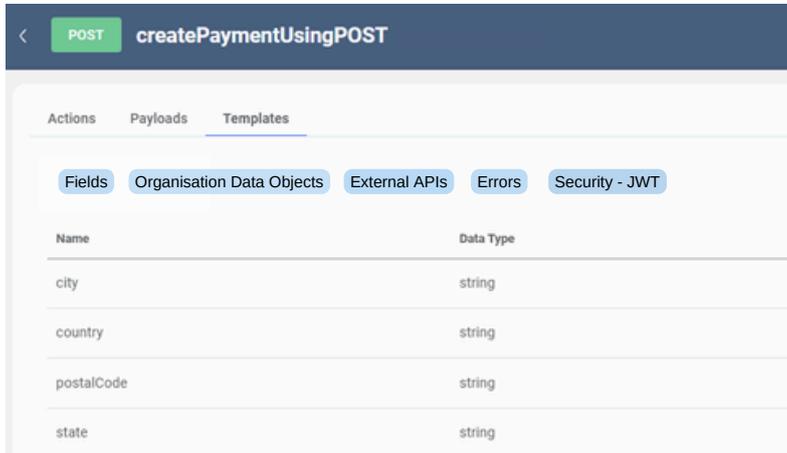
Field Name	Source
Error ID	Unique ID
Message	Fixed Value
Errors	
Error Code	Generation Rule
Field Name	Generation Rule
Error	Generation Rule

Field Name	Source
Error ID	Unique Id
Message	<input type="text" value="Already Exists"/>
Errors	
Error Code	<input type="text"/>
Field Name	<input type="text" value="Field Name"/>
Error	<input type="text" value="Resource already exists"/>

DESIGN SECTION

Templates

The Templates tab gives you access to Method-specific Field Templates, Organisation Data Objects and External APIs (where selected for use), Error Templates and your organisation's 'Security - JWT' Template.



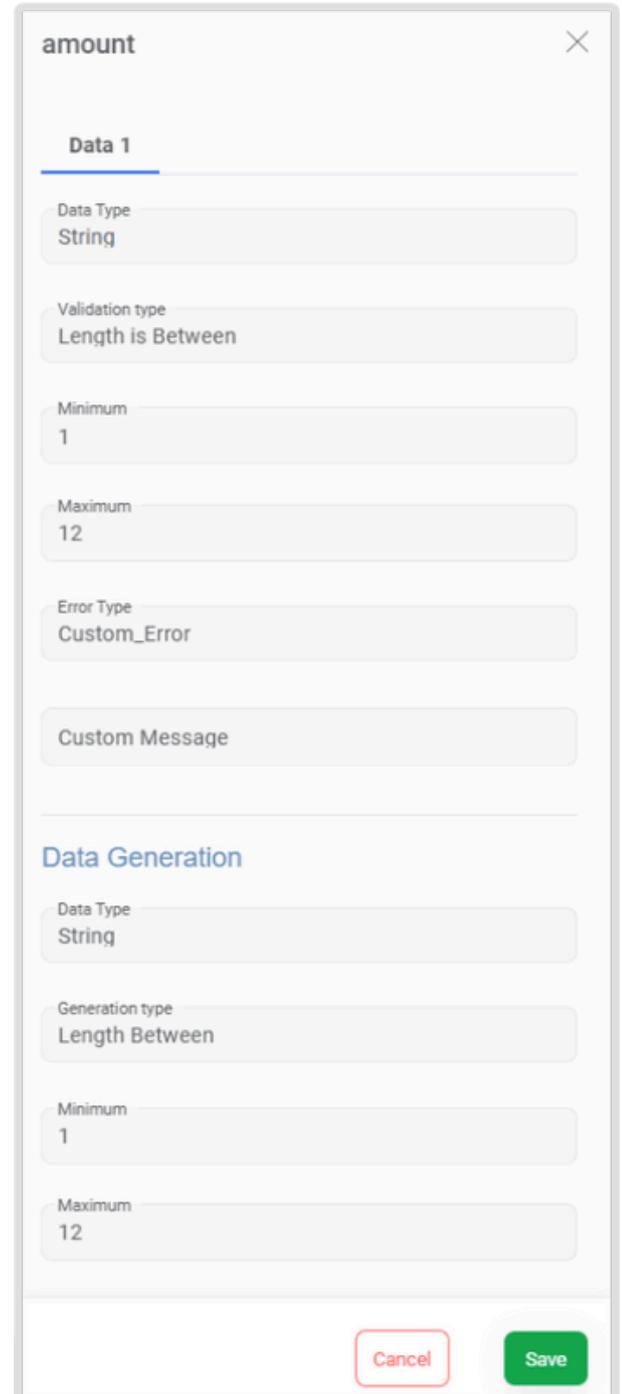
Name	Data Type
city	string
country	string
postalCode	string
state	string

Field Templates

You can quickly see what Field templates are used within the Method that you are viewing/editing.

They are arranged alphabetically by name and the Data Type and Validation Type are indicated for each Field.

By clicking on the View Button, you can view the Data Validation and Data Generation configuration for each Field Template.



amount [Close]

Data 1

Data Type: String

Validation type: Length is Between

Minimum: 1

Maximum: 12

Error Type: Custom_Error

Custom Message

Data Generation

Data Type: String

Generation type: Length Between

Minimum: 1

Maximum: 12

[Cancel] [Save]

Organisation Data Objects

When selected within the 'Actions' tab during configuration, Organisation Data Objects are displayed within the Templates tab.

Placeholder

An Organisation Data Object is configured in a similar manner to the Data Object Payload, and contains the same content:

- Field Name
- Source
- Data Type

Where it relates to an Add or Update Action (as configured in the Action Tab), the Source of each of its Fields' data is indicated and editable.

Where it relates to a Get Action (as configured in the Action Tab), there is no 'Source' configuration required and the information will be View Only - not editable.

The data from the Organisation Data Object is available for use in the Data Object payload of the Method.

There maybe several Organisation Data Objects used within the configuration of a Method; each will be displayed separately.

In fact, the same Data Object may be displayed several times, if it relates to a number of Actions (Add/Update/Get) within the same Method.

External APIs

When selected with the 'Actions' tab during configuration, External APIs are displayed within the Templates tab. They are 'view only' pre-configured API calls.

Placeholder

External APIs can relate to OpenAPI Specification (OAS) documents or Web Services Description Language (WSDL) documents as configured within the Standards section.

OAS configured External APIs will display a Parameter and Request Body; the source of each Field is configurable in a similar manner to the Data Object payload. A non-editable Response Body is also displayed.

The data from the External API Response Body is available to use as a Source in the Data Object of the Method.

WSDL configured External APIs are configured in an identical manner but rather than being displayed as Parameter, Request Body and Response Body - you will see the equivalent Payloads: SOAP Header & SOAP Request and SOAP Header and SOAP Response.

Error Templates

All Error Templates available within the service are available to view (Read Only) under the Templates tab,

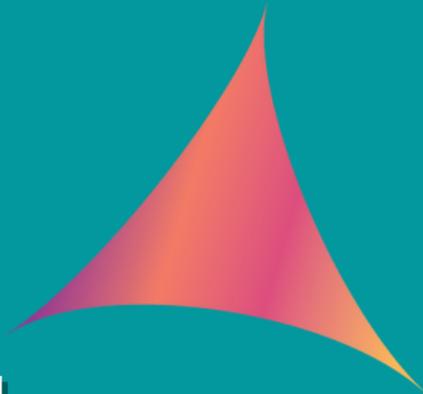
Placeholder

Security - JWT

The JSON Web Token (JWT) Template is available to view (Read Only) under the Templates tab,

Placeholder

DEPLOY SECTION

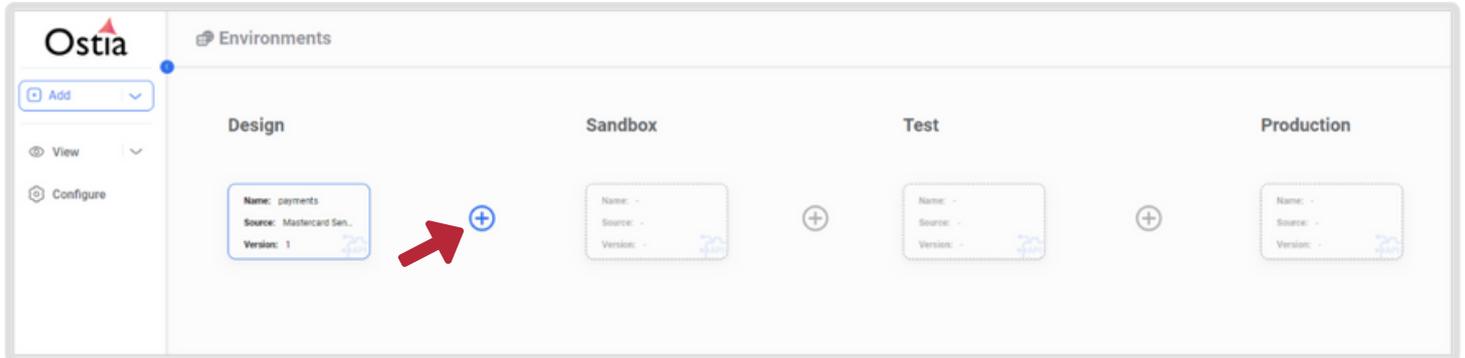


Ensure consistency of configuration and performance across your Sandbox, Test and Production Environments while simplifying delivery processes and improving partner and developer experience.

DEPLOY SECTION

Environments

Once you are finished the design of any API Implementation, you can it deploy into Sandbox, Test and Production Environments sequentially.



Sequential promotion across environments is as easy as clicking on the '+' symbol that is highlighted.

Where promotion has occurred, a blue link appears between the environments to indicate that they represent the same Version of the Design. The Version number displayed also indicates whether they represent the same or a different design. This is illustrated as follows:

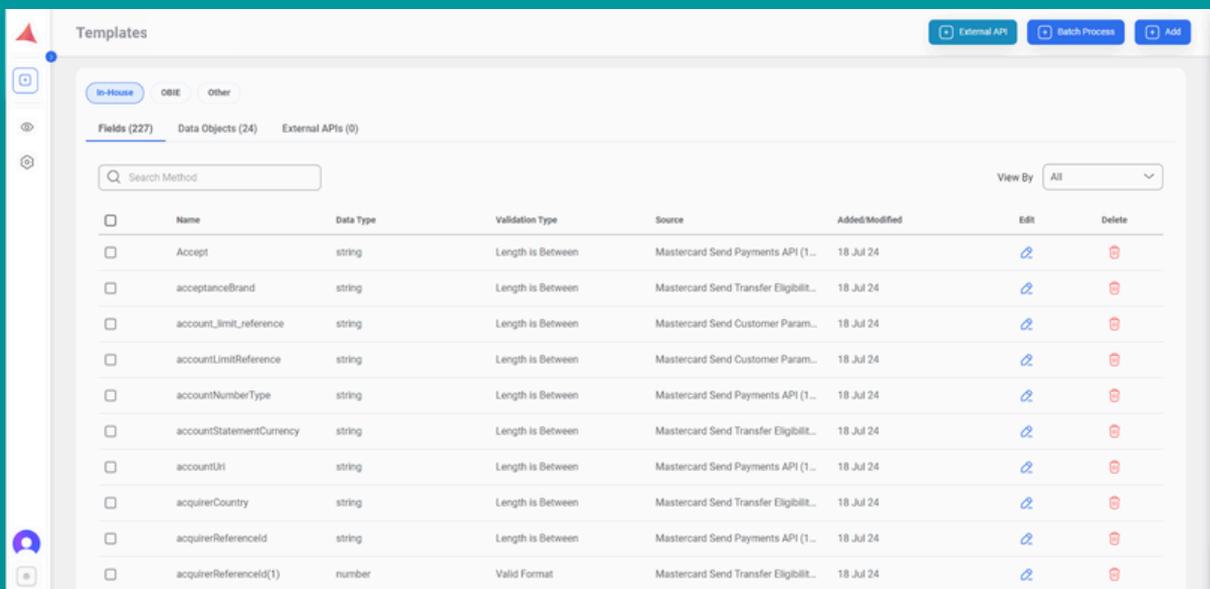
Placeholder

Where there is a difference between the Versions across the Environments it will be displayed using a red chain link and '+' symbol, as illustrated below. This difference between Versions can occur where changes are made to the Design Environment, that have yet to be promoted across all of the Environments.

Placeholder

STANDARDS SECTION

Ensure that the core building blocks of you API Ecosystem are in place in order to safeguard and optimise the interoperability and composability benefits available to your organisation,



The screenshot displays a web interface titled 'Templates' with a sidebar on the left and a main content area. The main area has tabs for 'In-House', 'OBIE', and 'other'. Below the tabs, there are filters for 'Fields (227)', 'Data Objects (24)', and 'External APIs (0)'. A search bar labeled 'Search Method' is present. A 'View By' dropdown menu is set to 'All'. The main content is a table with columns: Name, Data Type, Validation Type, Source, Added/Modified, Edit, and Delete. The table lists various API fields such as 'Accept', 'acceptanceBrand', 'accountLimitReference', 'accountLimitReference', 'accountNumberType', 'accountStatementCurrency', 'accountUri', 'acquirerCountry', 'acquirerReferenceId', and 'acquirerReferenceId(1)'. Each row includes a checkbox, a link for editing, and a red trash icon for deletion.

<input type="checkbox"/>	Name	Data Type	Validation Type	Source	Added/Modified	Edit	Delete
<input type="checkbox"/>	Accept	string	Length is Between	Mastercard Send Payments API (1...	18 Jul 24	Edit	Delete
<input type="checkbox"/>	acceptanceBrand	string	Length is Between	Mastercard Send Transfer Eligibilit...	18 Jul 24	Edit	Delete
<input type="checkbox"/>	accountLimitReference	string	Length is Between	Mastercard Send Customer Param...	18 Jul 24	Edit	Delete
<input type="checkbox"/>	accountLimitReference	string	Length is Between	Mastercard Send Customer Param...	18 Jul 24	Edit	Delete
<input type="checkbox"/>	accountNumberType	string	Length is Between	Mastercard Send Payments API (1...	18 Jul 24	Edit	Delete
<input type="checkbox"/>	accountStatementCurrency	string	Length is Between	Mastercard Send Transfer Eligibilit...	18 Jul 24	Edit	Delete
<input type="checkbox"/>	accountUri	string	Length is Between	Mastercard Send Payments API (1...	18 Jul 24	Edit	Delete
<input type="checkbox"/>	acquirerCountry	string	Length is Between	Mastercard Send Transfer Eligibilit...	18 Jul 24	Edit	Delete
<input type="checkbox"/>	acquirerReferenceId	string	Length is Between	Mastercard Send Payments API (1...	18 Jul 24	Edit	Delete
<input type="checkbox"/>	acquirerReferenceId(1)	number	Valid Format	Mastercard Send Transfer Eligibilit...	18 Jul 24	Edit	Delete

STANDARDS SECTION

In the Standards section, you are able to add and configure templates that are used in the design of API Implementations across your entire organisation.

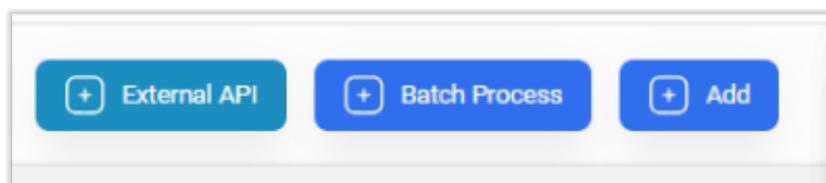
Placeholder

The types of template covered in this section:

- Fields
- Data Objects
- External APIs
- Errors
- Security - JWT

Adding Templates

Templates are added to the service in the following manner:



- Fields and Data Object templates are added to the service individually or by batch processing..
- External APIs are added individually.
- The service comes complete with pre-configured Error templates and a Security - JWT template.

During client onboarding Ostia works closely with you to ensure the effective and prompt creation of a full taxonomy of templates. This includes creation and configuration of the Field and Data Object templates and tailoring of the Error and JWT templates, where required, to fully meet your requirements.

STANDARDS SECTION

Add Field Template

To add a Field template, click on the 'Add Button'. A new Window then opens that enables the configuration of the Field template (default option).

Firstly, complete the name and description for the Field that you wish to add.

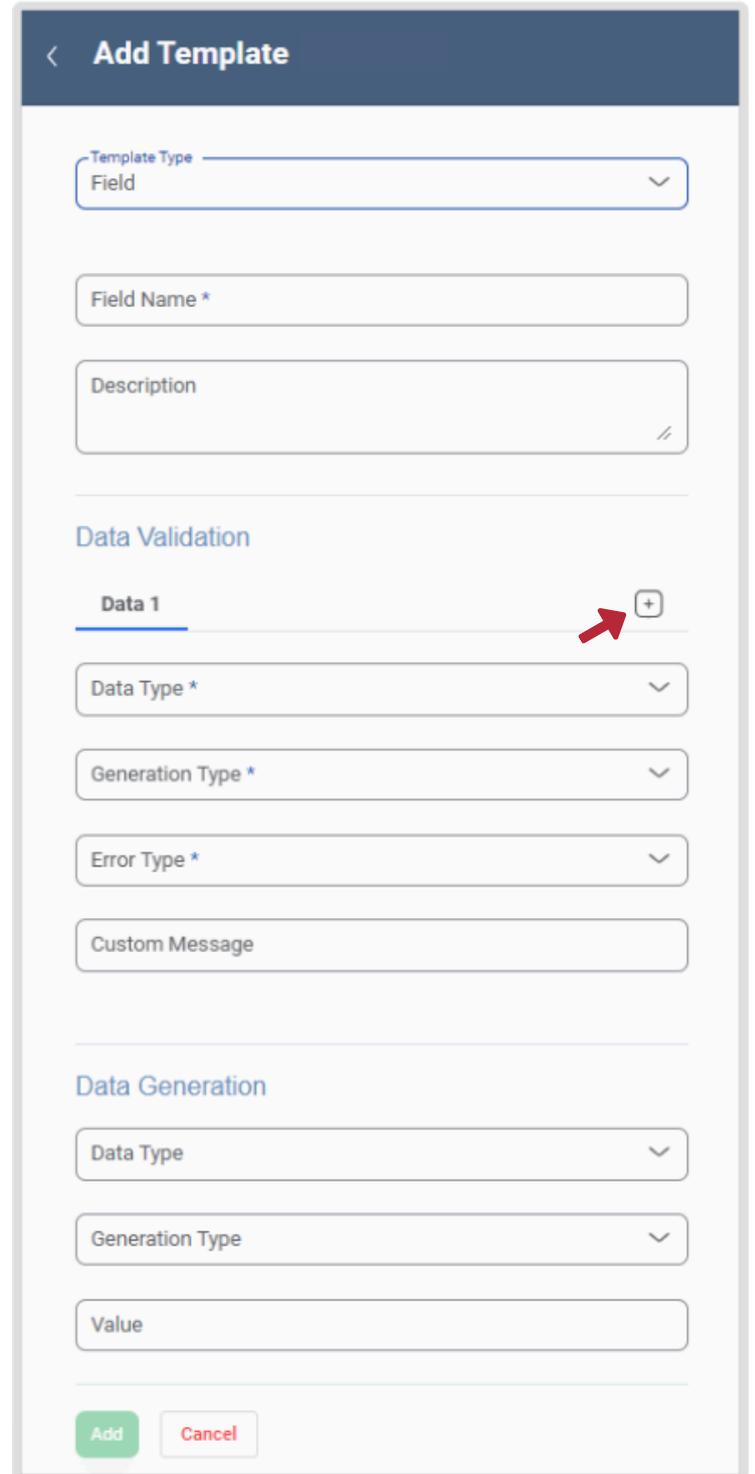
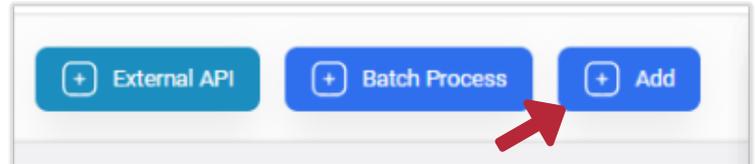
Secondly, complete the Data Validation requirements. When you select the Data Type (String, Date, Decimal, Numeric and Record.), the remaining fields will adjust to relate to that selection.

It is possible add more than one set of Data Validation requirements by clicking on the '+' symbol highlighted here.

Finally, you will complete the Data Generation fields. These should be a mirror image of the configuration in the Data Validation requirements, just completed.

The Field templates are the core building blocks of API Implementations. They will be used by default in all designs that are created within this service.

This greatly facilitates the optimisation of interoperability and composability across API Implementations and Methods within your API Ecosystem.

A screenshot of a mobile application form titled 'Add Template'. The form has a dark blue header with a back arrow and the title. Below the header, there are several sections: 'Template Type' with a dropdown menu showing 'Field'; 'Field Name *' with a text input field; 'Description' with a text area; 'Data Validation' section with a sub-header 'Data 1' and a '+' icon to its right (highlighted by a red arrow); 'Data Type *' with a dropdown menu; 'Generation Type *' with a dropdown menu; 'Error Type *' with a dropdown menu; and 'Custom Message' with a text input field. Below these is the 'Data Generation' section with 'Data Type' and 'Generation Type' dropdown menus, and a 'Value' text input field. At the bottom, there are two buttons: 'Add' (green) and 'Cancel' (white with red text).

STANDARDS SECTION

Add Data Object

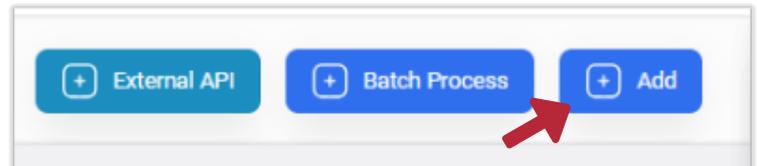
To add a Data Object template, click on the 'Add' button.

A new Window opens and when you select Data Object in the Template Type dropdown, you can configure a Data Object template.

Firstly, complete the name and description for the Data Object that you wish to add.

Then select the Fields you wish to add to the Data Object. The full list of Fields from the Field template section are available for selection.

When you have added all of the required Fields, click on the 'Add' button to create the data object.

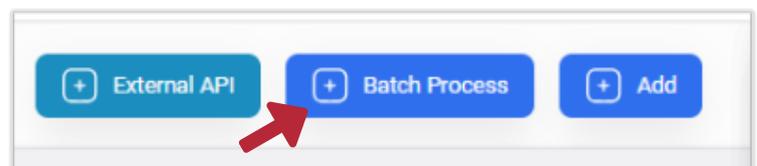
A screenshot of the 'Add Template' form. At the top is a dark blue header with a back arrow and the text 'Add Template'. Below the header is a 'Template Type' dropdown menu with 'Data Object' selected. There are two input fields: 'Field Name *' and 'Description'. Below the input fields is a table with three columns: 'Name', 'Description', and 'Delete'. The 'Name' column has a dropdown menu with 'Field Name' selected and an 'Add' button. At the bottom of the form are two buttons: 'Add' (green) and 'Cancel' (red).

Batch Processing

It is possible to batch process OpenAPI Specification (OAS) documents in order to create your taxonomy of Field and Data Object templates. Field and Data Object Templates are the core building blocks of API Implementations.

To 'Batch Process' OpenAPI Specification (OAS) documents, you simply click on the 'Batch Process' button to the top right of the screen as indicated above. This will open a new Window where you can select the OAS documents and click 'Add' to process.

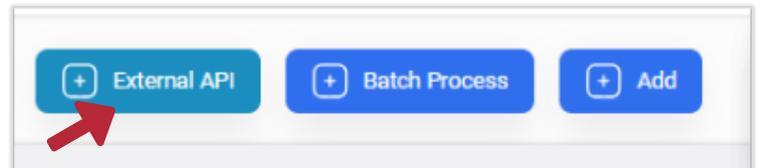
When the batch processing is complete, you will be taken to the Field template tab to review and finalise their configuration, before advancing to the Data Object tab to finalise their configuration.



STANDARDS SECTION

Add External API

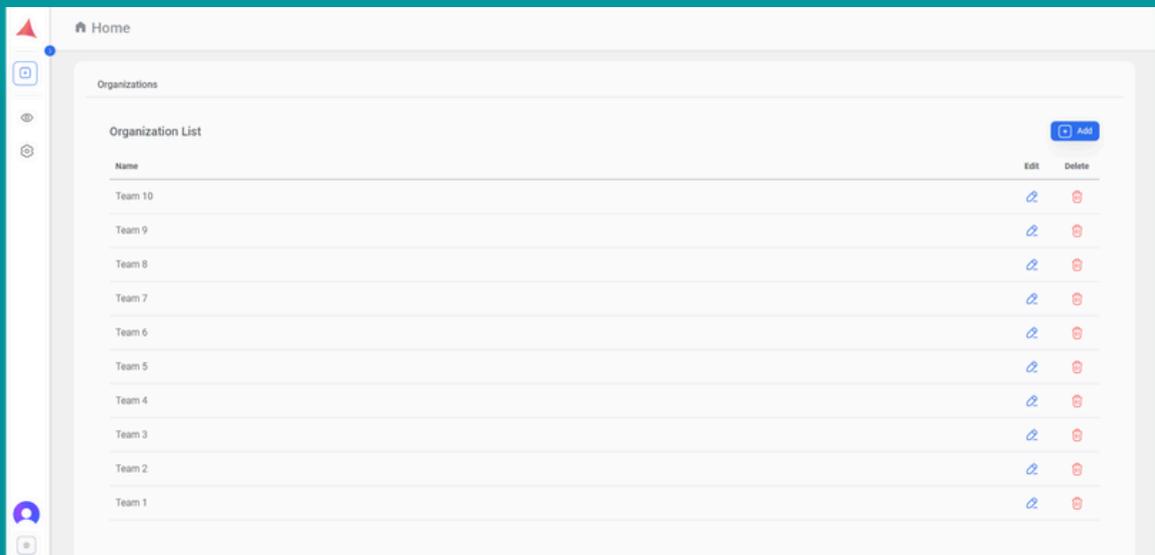
To individually add an External API, click on the 'Add External API' button.



It is then as simple as selecting the OpenAPI Specification (OAS) that relates to the External API you wish to access, and clicking the 'Add' button.

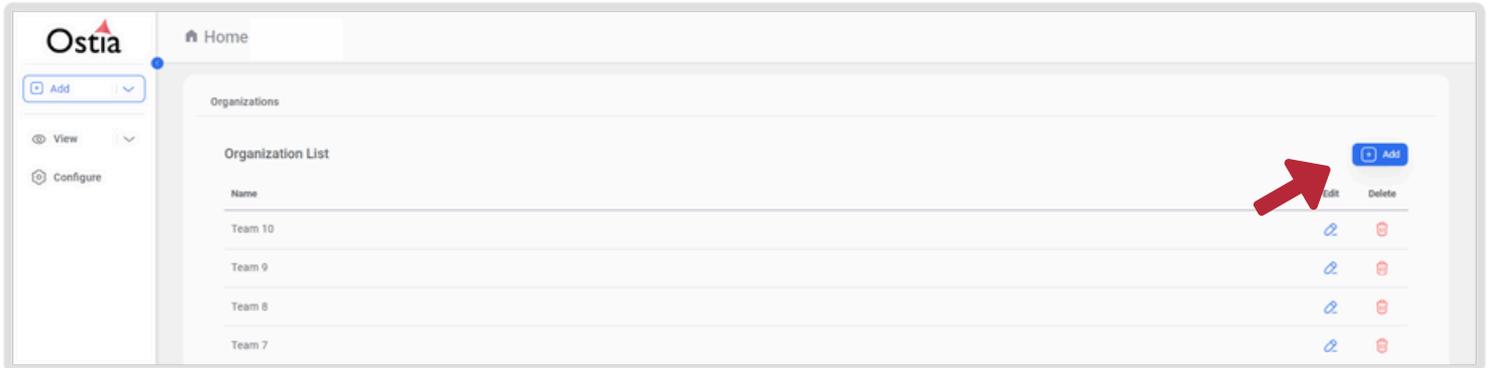
USER MANAGEMENT

People are at the core of your API Ecosystem - manage their access rights to ensure effective collaboration, innovation, design and deployment.



USER MANAGEMENT

Adding Teams and Users to the service is very straight forward.

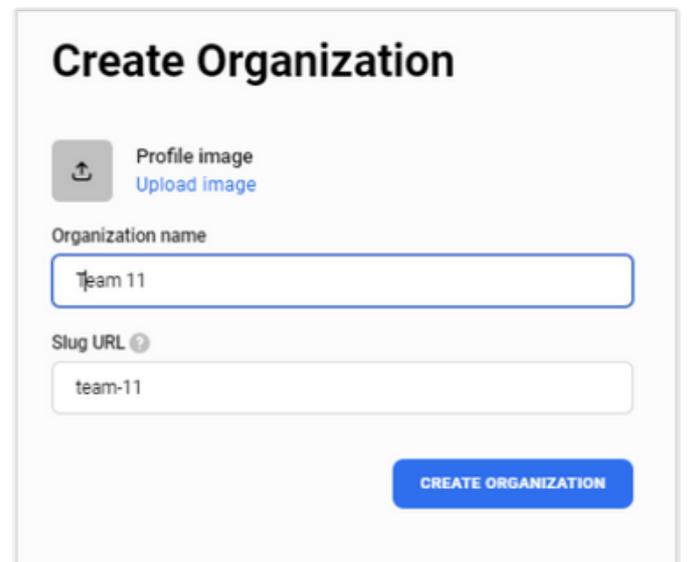


Add a Team

To add a Team to your service, click on the 'Add' button.

A new screen will appear and you will be asked to provide an Organisation name,

Complete this section and click 'Create Organisation' to add a new Team.



Add a User

To add a User to a Team, click on the Edit button related to the relevant Team, and then click on the "Invitations" tab.



Then simply click on the 'Invite Button' and enter the email of the person you wish to invite and choose the role you wish to assign to them from the available options.

USER MANAGEMENT

User Roles

It is possible to create any number of User Roles to define different access rights into the service to satisfy the requirements of different roles.

Currently, two roles are defined within the service:

- Admin
- Member

An 'Admin' user has full access to all parts of the service and can add new Teams and Users.

A 'Member' user has access to the Design Section to Add and Edit API Implementations.

The User Roles available to each organisation are created/configured during onboarding to the service to ensure the Roles meet the needs of your internal processes.



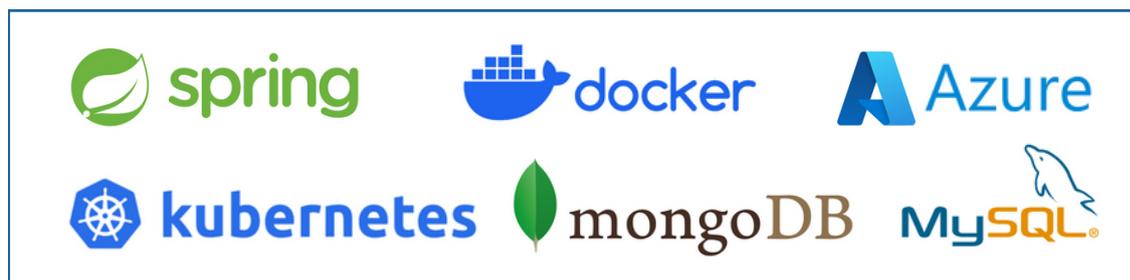
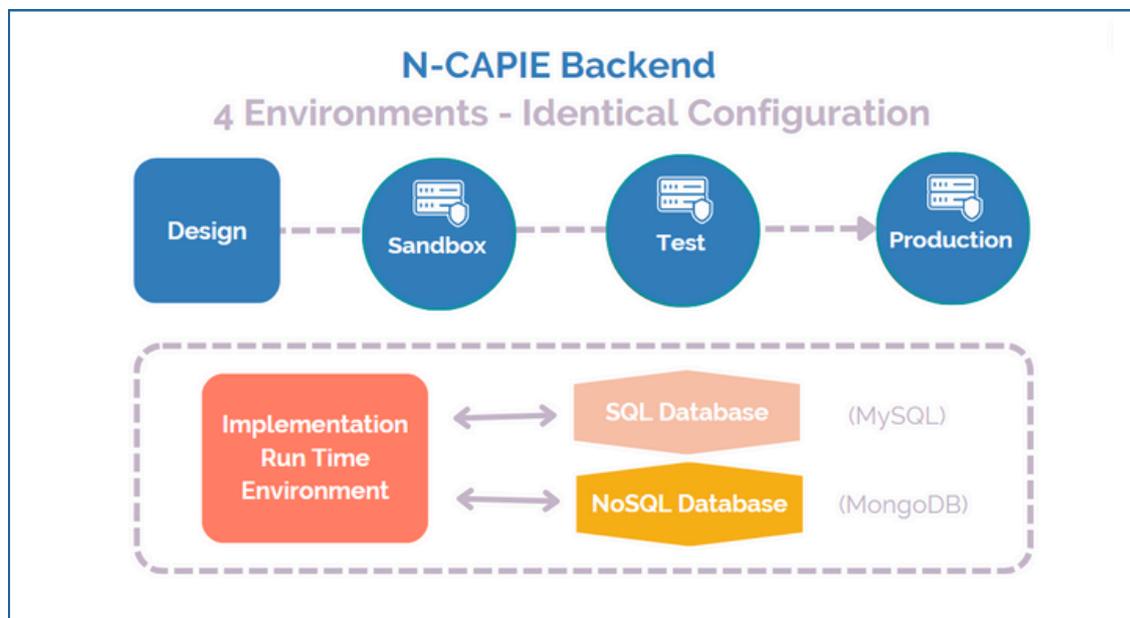
TECHNOLOGY & SECURITY

N-CAPIE offers powerful, secure and scalable infrastructure using technologies such as Spring Boot, Docker, Kubernetes, mySQL and MongoDB on Azure Cloud for a single tenant service deployment.



TECHNOLOGY & SECURITY

Each Environment within N-CAPIE is configured in an identical manner and the technologies we use ensure the provision of a secure, scalable infrastructure.



Spring Boot

The N-CAPIE Run Time Environment is developed using Spring Boot which is designed to be lightweight and standalone. This is ideal for micro-services architecture and enables the development of robust, scalable, and maintainable applications, while also simplifying deployment and operations.

Docker

N-CAPIE uses Docker to make it easier to create, deploy, and run applications through the use of containers. Containers encapsulate all dependencies, ensuring that applications run the same way regardless of the environment.

Containers also facilitate scaling and portability. They can be easily replicated and scaled horizontally to handle increased load and can run on any system that supports the Docker runtime, including different operating systems and cloud platforms.

Kubernetes

N-CAPIE uses Kubernetes - a powerful and versatile platform for managing containerized applications, driving efficiency, scalability, and reliability in modern software development and operations.

Kubernetes can automatically scale applications up or down based on traffic and resource usage, ensuring optimal performance and cost-efficiency. It efficiently distributes network traffic and workloads across multiple containers to ensure no single container is overwhelmed.

MySQL Database

N-CAPIE uses MySQL Database to store product configuration data. This enables optimal control over configuration data to ensure that any/all updates can be carried out seamlessly.

Mongo DB

N-CAPIE uses MongoDB as its document-oriented storage allows for flexible schema design, making it easier to iterate and evolve data models. Documents are stored in a JSON-like format (BSON), which aligns well with modern API payloads, reducing the need for data transformation.

Microsoft Azure Cloud

Azure's expansive global infrastructure, coupled with its focus on security and compliance makes it a strong choice for N-CAPIE..

It also provides the flexibility to scale resources up or down based on demand and to adapt quickly to changing needs.

It offers encryption, threat detection, identity management, and compliance certifications to ensure data protection and meet regulatory requirements.

Azure's redundant infrastructure ensures high availability and reliability, minimizing downtime and providing consistent access to services.

Single Tenancy

Each N-CAPIE Instance is dedicated to a single tenant which means that the environment is fully tailorable to exactly meet the client's requirements. All the server resources are dedicated to the client and all data is kept completely separate from other users.

Server Portability



Use Your Own Cloud: N-CAPIE can be deployed for use on any existing, or preferred, Cloud Provider(s). And is designed to be easily portable across all environments. On-Premise: N-CAPIE can additionally be hosted within your on-premise environment.

Integration

N-CAPIE can connect with anything that uses an API and is designed to be fully integratable within your existing CI/CD processes: Identity Provider, API Gateway, automated testing, API performance measurement and the like.

Auto-Documentation

N-CAPIE automatically documents your API Implementations using OpenAPI Specification when they are promoted to the Sandbox Environment to ensure seamless collaboration with partners and developers.

Version Control

N-CAPIE manages granular Version Control across all environments (this is possible because the same configuration is used within the Design, Sandbox, Test and Production environments).

One Back-End to Power Multiple Front-Ends

The No-Code configuration of Methods and API Implementations, means that your back-end is available to service multiple front-ends and projects through the high level of composability achieved across your API Ecosystem.

Push-Button Deployment

Once API Implementation designs are completed/approved, promotion into Sandbox, Test and Production Environments is (literally) as easy as the click of a button - server environments are pre-configured and architecture deployment is automated.

OAuth

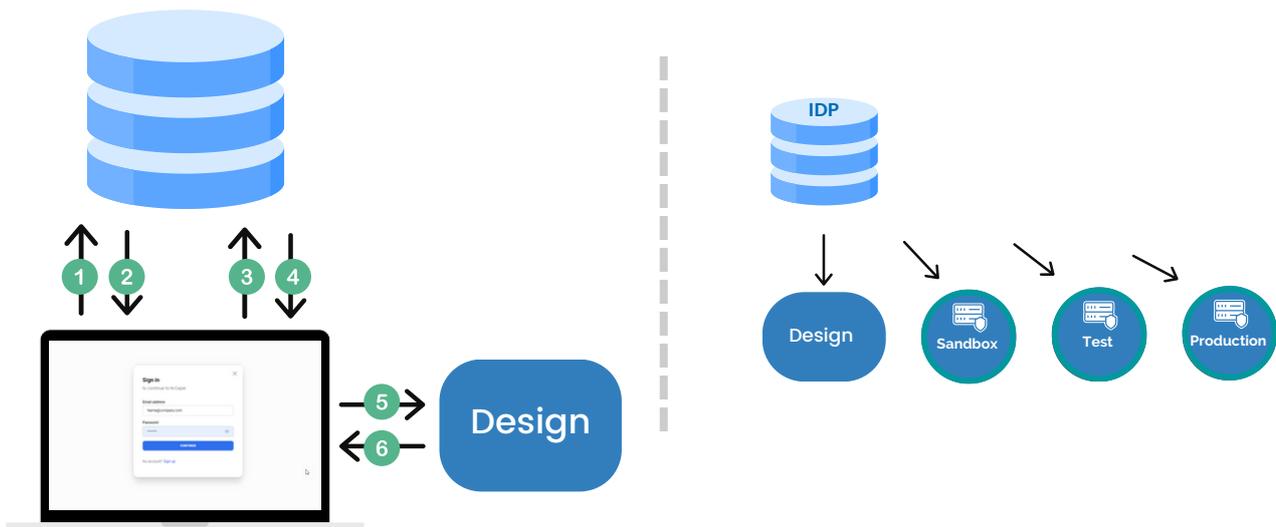
OAuth enhances security by providing a more secure and controlled way to grant and manage access to resources, reducing the risks associated with traditional authentication and authorization methods.

N-CAPIE uses OAuth 2.0 for:

1. Authentication of users and Access to Runtime Environments
2. API Accessors (applications attempting to access N-CAPIE APIs)

N-CAPIE can integrate with any Identity Provider (IDP) implementing the OAuth standard. We configure the service for each client's specific security integration requirements at the time of Onboarding.

1. Authentication of Users & Access to Runtime Environment

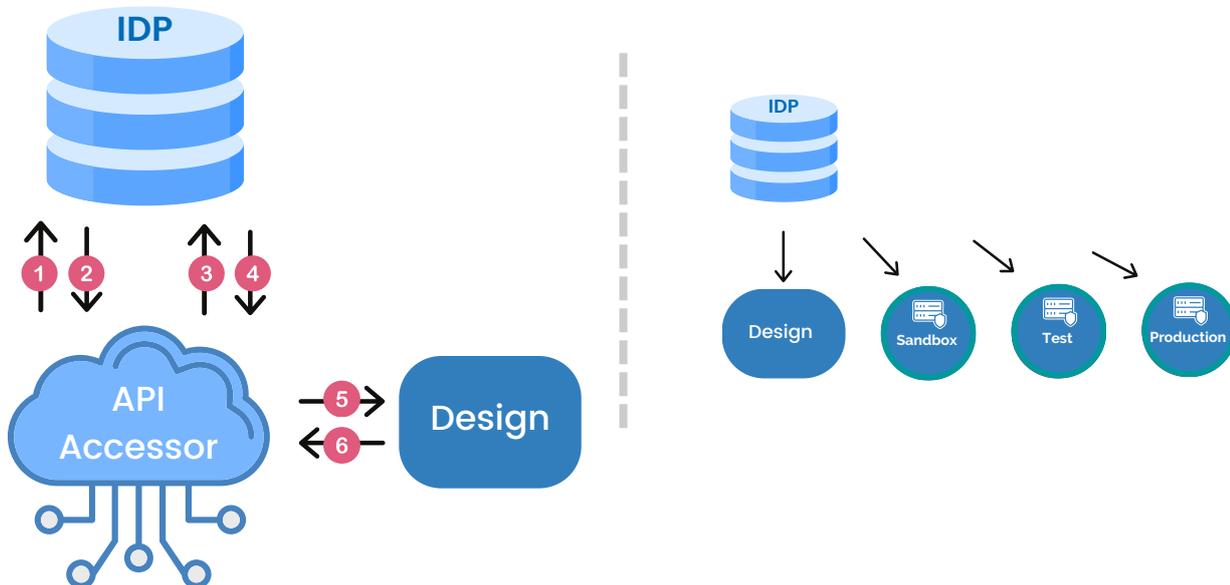


1. Users Login credentials are redirected to IDP for Authentication
2. Once Authentication occurs, the IDP sends an Authorization Code
3. N-CAPIE requests the IDP to exchange the Authorization Code for a JWT
4. IDP validates the Authorization Code and returns a valid JWT
5. N-CAPIE sets the JWT into the HTTP Authorization Header and sends requests to the Design Environment
6. The Design Environment validates that the JWT is valid, processes the API request and returns the response to N-CAPIE.

JSON Web Key Set (JWKS): When each environment is started, it downloads a JWKS from the IDP which is used to validate JWTs sent to the environment.

OAuth (Cont'd)

2. API Accessors (applications attempting to access N-CAPIE APIs)



1. The API Accessor redirects to IDP for Authentication
2. Once Authorization occurs, the IDP returns an Authorization Code
3. The API Accessor requests the IDP to exchange the Authorization Code for a JWT
4. IDP validates the Authorization Code and returns a valid JWT
5. The API Accessor then sets the JWT into the HTTP Authorization Header and sends requests to the Design Environment. (Where the UI must access the other Environments, this token must also be provided).
6. The Design Environment validates that the JWT is valid, processes the API request and returns the response to N-CAPIE.

