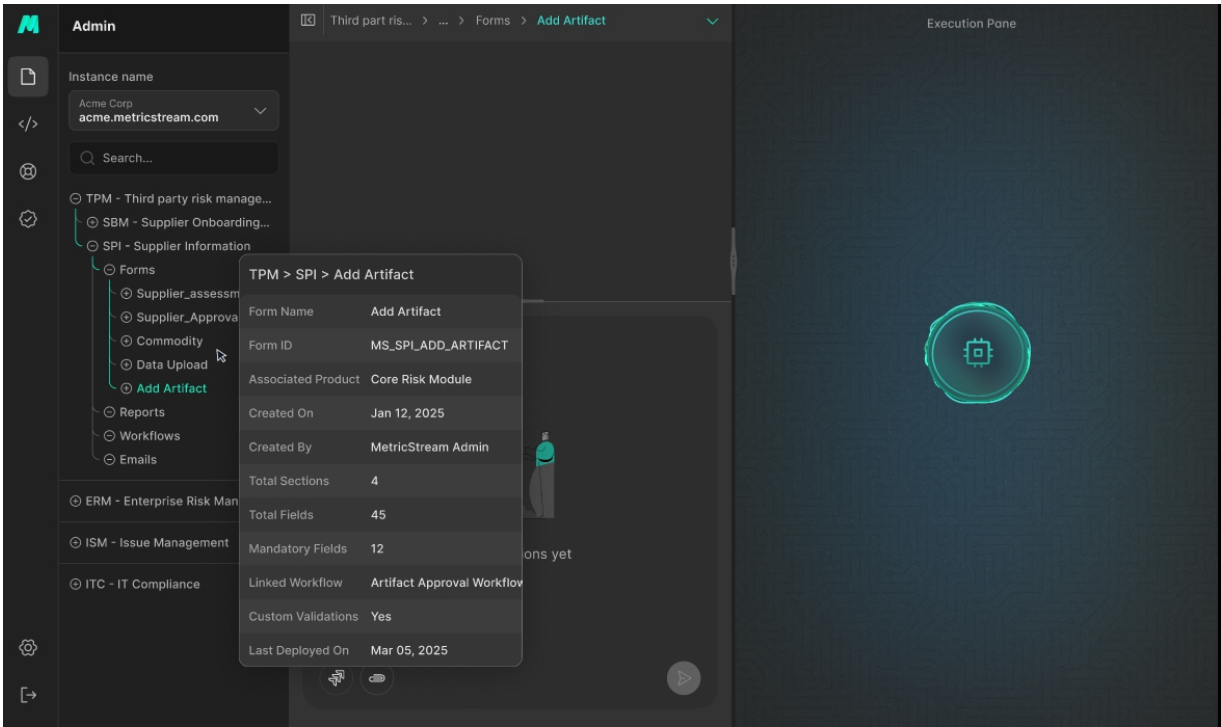# API Listing and Details

Admin UX – Module & Object Explorer Flow

⬜ Overview

As part of the **Agentic AI Webapp**, the Admin UX connects to a selected **MetricStream instance**. Upon authentication, it dynamically loads all product modules and allows users to interactively explore associated objects like Forms, Workflows, Reports, etc.

This document outlines the **user navigation flow**, required **API calls**, and object-level interaction strategy. This guide is intended for developers implementing backend logic and frontend integrations.

⬜⬜ User Flow & API Interaction

1. Instance Login

- User selects or inputs the MetricStream instance (e.g., `coditasdemo.metricstream.com`) on login.

- The app performs authentication and stores access tokens securely.

---

2. List All Modules

- After login, all modules available in the instance are listed in the left panel.

- **API Called**:

```
GET /api/modules
```

- **Response**: [
  {
  "moduleId": "IAD",
  "moduleName": "Internal Audit",
  "product": "Audit Management"
  },
  ...
  ]

- Note: Modules will get extended modules also like GRX, etc….

---

3. Expand Module to View Object Types

- When a module is expanded (e.g., `IAD`), the system fetches supported object categories.

- **API Called**:
  ```
  GET /api/modules/{moduleId}/object-types
  ```

- **Response**: ["Form", "Workflow", "Report", "DataObject", "Chart", ...]

---

4. Select Object Type (e.g., Form)

- Upon clicking an object type like **Form**, list of forms is fetched for the module.

- **API Called**:
  ```
  GET /api/modules/{moduleId}/{objects}
  ```

  - e.g. `GET /api/modules/IAD/forms`

- **Response**: [
  {
  "formId": "MS_IAD_AUDIT_PLAN",
  "formName": "Audit Plan",
  "createdOn": "2024-09-10",
  "createdBy": "admin"
  },
  ...
  ]

---

5. View Form Metadata

- When a specific form is expanded, detailed metadata is fetched including section/field info.

- **API Called**:
  ```
  GET /api/modules/{moduleId}/forms/{formId}/metadata
  ```

- e.g. GET /api/modules/IAD/forms/MS_IAD_AUDIT_PLAN/metadata

- **Response**: {
  "formName": "Audit Plan",
  "formId": "MS_IAD_AUDIT_PLAN",
  "sections": 5,

```
"fields": 42,
"mandatoryFields": 8,
"workflow": "Audit Approval Workflow",
"validations": true,
"lastDeployed": "2025-03-05",
"fieldDetails": [
{ "fieldId": "F1", "fieldName": "Audit Name", "type": "Text" },
{ "fieldId": "F2", "fieldName": "Start Date", "type": "Date" }
]
}
```

☐ Object Types Supported

The Admin UX supports exploring the following object types per module:

- Blueprint

- Form → Required for Phase 1: Conversion to BR/React with ES6 format using mPower

- Workflow → Required for Phase 1: Conversion to Emery

- Report → Required for Phase 1: Conversion to React with ES6 format using Report API

- Chart

- Dashboard

- DataObject

- DataTables

- LOVs (List of Values)

- Infolets

- Infocenters

- DataBrowsers

- Activities

- etc…..

Each object type follows a similar pattern of:

1. `GET /api/modules/{moduleId}/forms/{FOR_NAME}/`

2. `GET /api/{objectType}/get{ObjectType}Metadata?moduleId={}&id={}`

☐ API Summary

API Endpoint	Method	Description
`/api/modules`	GET	Fetch all modules in the connected instance
`/api/modules/{moduleId}/object-types`	GET	List object types (Form, Report, etc.) under a module
`/api/modules/{moduleId}/forms`	GET	List all forms in a module
`/api/modules/{moduleId}/forms/{formId}/metadata`	GET	Retrieve detailed metadata of a specific form
/api/modules/{moduleId}/{objects}	GET	Pattern for fetching all object entries
/api/modules/{moduleId}/{objects}/{objectId}/metadata	GET	Pattern for fetching metadata of a specific object

☐ Example Navigation Tree (UX Hierarchy)

(+) Modules
(+) IAD – Internal Audit
(+) Forms
(+) Audit Plan
→ Metadata
→ Field List
(+) Reports
(+) Workflows
(+) PDMS – Policy & Document Management
...

☐ Notes

- All API endpoints should be authenticated using user session/context tokens.

- Expand operations are designed to be **lazy-loaded** (on-demand) to improve performance.

- Metadata returned should be schema-aligned for consistent rendering and editing in the UI.

Analysis needed:

List of APIs

Which APIs can be used

For the usage of the above APIs what is required for Input and output

API:

Agentic AI - Phase 1 (Editor)

1. Phase 1 - Part 1 - Fr Editor requirement GET APIs (

1. List of Modules

2. List of Objects

2.1 FORM, Workflow, Blueprint, Report

2.1.3 Report

3. List of {Object} Details Note: Required for Product Development/Upgrade Cases
   3.1

4. Phase 1 - Part 2 - PUT/POST APIs

Reference: → 2025-04-24-Admin-UX-Agentic-AI-API-Design-Options.pptx