API Definition and Management

Scott Nieman
Enterprise Integration Architect
06.03.19
We operate four diversified agribusinesses, driven by insights and innovation

Dairy Foods
Milk-based products and ingredients

Animal Nutrition
Solutions that enhance performance and well-being

Crop Inputs & Insights
Agricultural products, data, technology tools and services

Sustainability
Environmental sustainability solutions
APIs are need across the Farm to Fork View

- Technology & Insights
- Crop Inputs
- Seed
- Animal Nutrition
- Primary Processing
- R&D/Manufacturing
- B2B/Industrial Marketing & Sales
- Branded Goods Marketing & Sales

Production
Consumer
API Maturity typically Limited by Middleware Capabilities

Oracle Fusion Middleware prefined REST endpoints and used BadgerFish
Level 0

Adding Intel API Gateway enabled VIP and security
Level 1
Section subhead

Current State: MuleSoft
Mulesoft Anypoint iPaaS Platform

*Anypoint Design Center*
- Studio™
- API Designer™
- Connector DevKit™

*Anypoint Management Center*
- Runtime Manager™
- API Manager™
- Analytics™
- Access management™

*Anypoint Exchange*
- API Portal™

*Mule runtime engine*

*Anypoint Connectors*
- High Availability
- Enterprise Security
- Virtual Private Cloud
- Load Balancer

*Runtime services*
- Object Store

*Hybrid cloud - CloudHub™*

*MuleSoft Runtime Options:*
- **MuleSoft Cloud - Chosen**
- Azure IaaS/On-Premise
Design & Development

API development cycle
- Integration specialist
- Adhoc integrators
- Testers

Connectivity flow development cycle
- API Designer
- Visual API Design
- API Testing
- Studio
- Flow Designer
- MUnit

Author ➔ Simulate ➔ Feedback ➔ Validate

Build ➔ Connect ➔ Test
Deploy & Operate

Promote -> Govern -> Deploy -> Monitor

Exchange
Discover and share reusable APIs, connectors, and templates.

Management Center
Deploy, operate, and analyze your APIs and applications.

Discover & share

Manage

Exchange
API Portal
Runtime Manager
API Manager
High Level Process

1. Define RAML in Design Center
   - Include schema in RAML
   - Include Manual JSON sample payload

2. Build API in Anypoint Studio
   - System APIs as Façade to application
   - Process APIs orchestrate flows
   - Experience APIs expose to consumer (internal and external)

3. Create Sample Payloads
   - Use Real Data
   - Depict more than one scenario

4. Publish to MS Exchange
   - Validate
   - Validate
   - Validate

5. Allow Client developer to Request Access
   - Download OAS 2.0
   - Swagger CodeGen
OAGIS in Design Center (RAML)

POST /wfu/labs/InspectionOrder

GET /wfu/labs/TestResults
OAGIS in Design Center

WFU Testing API V2/master

$schema: "http://json-schema.org/draft-04/schema#"
required: [ 1 item]
properties: []
additionalProperties: false
metaHeader: []
  type: object
required: [ 1 item]
properties: [ 7 keys]
  senderIdentifier: []
  receiverIdentifier: []
  intermediaryIdentifier: []
  creationDateTime: []
  correlationIdentifier: []
  messageIdentifier: []
  inspectionOrder: []
additionalProperties: false
properties: [ 4 keys]
  description: "The inspection order in this scenario is a third party lab request to test an item of interest, such as soil, or plant tissue. The type of item will determine the applicable test methods."
  type: object
additionalProperties: false
properties: [ 20 keys]
  actionCode: []
  identifier: []
  lastModificationDateTime: []
description: "This the date the inspection order was either created or modified."$ref: "#definitions/datetime"
description: []
properties: [ 2 keys]
WinField United Labs Testing API

Scenario List

The testing API provides a means to submit an inspection order for a set of samples, which when received by SureTech labs will create a lab request and corresponding sample identifiers within the LIMS system. The order provides the detail about who requested the lab request, collected the samples, account information for access and billing, the sample information including geospatial information, and the specific tests to perform. Contextual information such as sampling protocol information, and targeted product to plant and expected yield result may be provided if the test results are intended to be shared with other FMIS application for analysis or recommendations.
Requestor creates an application and requests client-id and client-secret
API Layering implements ‘separation of concerns’ architecture principle

OAGIS Proxy pass-thru, Proprietary, CSV…

Orchestration Layer, OAGIS API Calls *multiple* System APIs

DB, File, system-specific terms

OAGIS can act as the wrapper
Experience API in Exchange Portal allows other NutraBlend Customers to POST their POs

Experience API in Exchange Portal allows other Purina Vendors to GET their POs
Use of Score
Guiding Principles to use Score

- Use standards where possible; common vocabulary (semantics)
- Don’t reinvent the wheel when creating APIs
- Simplify API definition to actual business use (business context)
Types of Business Information Entities

- Canonical; multiple business contexts
- Component
- Point to Point; single business contexts
DB (srt-repo) implements ISO 15000 Part 5 Core Components

https://www.iso.org/standard/61433.html
Canonical BIE

Multiple Business Contexts

- BIE UsageA
- BIE UsageB
- BIE UsageC
<table>
<thead>
<tr>
<th>Name</th>
<th>Release</th>
<th>Context</th>
<th>Owner</th>
<th>Version</th>
<th>Status</th>
<th>Last Update Time</th>
<th>State</th>
</tr>
</thead>
<tbody>
<tr>
<td>Invoice</td>
<td>10.5</td>
<td>Create Accounts Payable Invoice Voucher</td>
<td>ScottNieman</td>
<td></td>
<td></td>
<td>2019-05-30 19:41:43</td>
<td></td>
</tr>
<tr>
<td>Journal Entry</td>
<td>10.5</td>
<td>Allocate Costs to Ledger Account</td>
<td>ScottNieman</td>
<td></td>
<td></td>
<td>2019-05-29 20:42:32</td>
<td></td>
</tr>
<tr>
<td>Location</td>
<td>10.5</td>
<td>Party Location Management and Syndication</td>
<td>ScottNieman</td>
<td></td>
<td></td>
<td>2019-05-29 15:26:50</td>
<td></td>
</tr>
<tr>
<td>Sales Order</td>
<td>10.5</td>
<td>Shipment Request and Planning</td>
<td>ScottNieman</td>
<td></td>
<td></td>
<td>2019-05-28 21:30:02</td>
<td></td>
</tr>
<tr>
<td>Purchase Order</td>
<td>10.5</td>
<td>Shipment Request and Planning</td>
<td>ScottNieman</td>
<td></td>
<td></td>
<td>2019-05-28 21:29:00</td>
<td></td>
</tr>
<tr>
<td>Carrier Route</td>
<td>10.5</td>
<td>Shipment Request and Planning</td>
<td>ScottNieman</td>
<td></td>
<td></td>
<td>2019-05-24 17:36:39</td>
<td></td>
</tr>
<tr>
<td>Shipment</td>
<td>10.5</td>
<td>Shipment Request and Planning</td>
<td>ScottNieman</td>
<td></td>
<td></td>
<td>2019-05-23 18:44:56</td>
<td></td>
</tr>
</tbody>
</table>

Nouns
## List of BIEs

<table>
<thead>
<tr>
<th>Name</th>
<th>Release</th>
<th>Context</th>
<th>Owner</th>
<th>Version</th>
<th>Status</th>
<th>Last Update Time</th>
<th>State</th>
</tr>
</thead>
<tbody>
<tr>
<td>Price List</td>
<td>10.5</td>
<td>Customer Price List</td>
<td>Scott Nieman</td>
<td></td>
<td></td>
<td>2019-01-01 19:05:03</td>
<td>Editing</td>
</tr>
<tr>
<td>WIP Status</td>
<td>10.5</td>
<td>Capture IoT Measurement</td>
<td>Scott Nieman</td>
<td></td>
<td></td>
<td>2019-02-13 04:17:08</td>
<td>Editing</td>
</tr>
<tr>
<td>Inspection Order</td>
<td>10.5</td>
<td>Checking Status of Soil, Tissue, Resin and Forage Lab Racquets</td>
<td>Scott Nieman</td>
<td></td>
<td></td>
<td>2019-01-31 14:55:34</td>
<td>Editing</td>
</tr>
<tr>
<td>Confirm BOD</td>
<td>10.5</td>
<td>Information Technology Configuration</td>
<td>Scott Nieman</td>
<td></td>
<td></td>
<td>2019-01-22 15:40:15</td>
<td>Candidate</td>
</tr>
<tr>
<td>UOM Code Conversion Rate</td>
<td>10.5</td>
<td>Information Technology Configuration</td>
<td>Scott Nieman</td>
<td></td>
<td></td>
<td>2019-01-17 19:19:48</td>
<td>Candidate</td>
</tr>
<tr>
<td>Mota Header</td>
<td>10.5</td>
<td>Information Technology Configuration</td>
<td>Scott Nieman</td>
<td></td>
<td></td>
<td>2019-01-15 18:18:12</td>
<td>Candidate</td>
</tr>
<tr>
<td>Batch Certificate Of Analysis</td>
<td>10.5</td>
<td>Requesting Third Party Lab Soil and Tissue Tests</td>
<td>Scott Nieman</td>
<td></td>
<td></td>
<td>2019-01-14 16:05:45</td>
<td>Editing</td>
</tr>
<tr>
<td>Remittance Advice</td>
<td>10.5</td>
<td>Remittance and Cash Application</td>
<td>Scott Nieman</td>
<td></td>
<td></td>
<td>2019-01-11 13:49:49</td>
<td>Editing</td>
</tr>
</tbody>
</table>
## Review BIE

<table>
<thead>
<tr>
<th>BIE Code Conversion Rate</th>
<th>UOM Code Conversion Rate</th>
</tr>
</thead>
<tbody>
<tr>
<td>Type Code</td>
<td>UOM Code</td>
</tr>
<tr>
<td>Action Code</td>
<td></td>
</tr>
<tr>
<td>Identifier</td>
<td></td>
</tr>
<tr>
<td>Identifier Set</td>
<td></td>
</tr>
<tr>
<td>From UOM Package</td>
<td></td>
</tr>
<tr>
<td>Rate Number</td>
<td></td>
</tr>
<tr>
<td>To UOM Package</td>
<td></td>
</tr>
<tr>
<td>UOM Code</td>
<td></td>
</tr>
<tr>
<td>Unit Packaging</td>
<td></td>
</tr>
<tr>
<td>Description</td>
<td></td>
</tr>
<tr>
<td>Note</td>
<td></td>
</tr>
<tr>
<td>Extension</td>
<td></td>
</tr>
</tbody>
</table>

© 2015-18 Open Application Group, Inc. All rights reserved.
Move metaHeader Routing to HTTP Headers

Moving metaHeader from HTTP payload to HTTP headers is desired

Score should be able to enable these as standard headers

```json
{
    "metaHeader": {
        "receiverIdentifier": "Catalyst",
        "senderIdentifier": "ETM",
        "correlationIdentifier": "w9r27432hd29343",
        "messageIdentifier": "w9r27432hd29343",
        "scenarioIdentifier": "INT_ETM_Catalyst_5",
        "creationDateTime": "2019-05-14T12:43:53.342-05:00"
    },
    "journalEntry": {
        "journalEntryHeader": {
            "identifier": {
                "content": "1321231",
                "typeCode": "shipment.identifier"
            },
            "lastModificationDateTime": "2019-05-14T12:43:53.342-05:00",
            "documentReference": {
                "metaHeader": "METAHEADER"
            }
        }
    }
}```
Create Extension / Submit to OAGi
Score Environment

- Laptop
  - Chrome
- Laptop
  - Chrome
- VMware Guest/Score
  - srt-web1.1.1.2d
  - srt-repo
- Docker
- CentOS
Next up for Score: Generation of Open API Spec 3.0
MuleSoft DevOps Overview

**Arch/ BA**
- Score

**Development**
- API Design Center
- Exchange
- Anypoint Studio
- Local Runtimes (Laptop)
- Testing MUnit
- Reviewer (Branch: master)
- Azure Artifacts*

**CI**
- Azure Build Pipeline
  - Build
  - Maven
  - Munit**
  - Notify Developer

**CD**
- Azure Release Pipeline
  - Download Build Artifacts
  - Release
  - Maven
  - Deploy to CloudHub
  - Execute Functional Tests**
  - Notify Developer

**Monitoring**
- Anypoint Alerts
- Jenkins
  - VPC Network Monitoring
  - Logs

**Work in Progress & Future Backlog**
MuleSoft DevOps with Score OAS3.0 support

**Arch/ BA**

**Development**

- Score
- API Design Center
- Exchange
- Anypoint Studio
- Local Runtimes (Laptop)
- Testing MUnit
- Trigger: Push
- Password State
- Azure Artifacts*

**CI**

- Azure Build Pipeline
- Build
- Maven
- Munit**
- Publish Artifacts
- Notify Developer
- Reviewer (Branch: master)

**CD**

- Azure Release Pipeline
- Download Build Artifacts
- Release
- Maven
- Deploy to CloudHub
- Execute Functional Tests**
- Notify Developer
- Operations (prod)

**Monitoring**

- Anypoint Alerts
- VPC Network Monitoring
- Logs

**Work in Progress & Future Backlog**
Thank you

Contact information here