EPICOR Insights 2024

Integrating Kinetic with Bartender REST API

Simon Hall
ERP Support Specialist
May 22, 2024

Speaker Biography

Simon Hall

ERP Support Specialist

Polyaire Pty Ltd

- Working with Epicor Products since 1998, a seasoned professional with a rich background in various Small to Medium Enterprises, specializing in Customer Support, OS, Network, Database Administration, and ERP Implementation. My recent role has been on ERP Technical Support at Polyaire Australia. My expertise extends to ERP data migrations, administering Epicor ERP on Azure IaaS, developing third-party integrations using Epicor technologies including Functions and the Epicor REST API.

Simon Hall Polyaire Australia

Self introduction details

- Involved with Epicor Products since 1998
- Worked in various industries, Distribution, Professional Services, Manufacture supporting Epicor Products and developing customizations
- Dual experience from a Vendor and customers perspectives.
- A VW Vanagon tragic….

About Polyaire : Years in business:	28
# Locations / employees:	30/250

Epicor Partnership Longevity:	2017
Current version:	2022.1
Number of users:	101

Company vision and philosophy:

At Polyaire, we are committed to excellence in air conditioning systems and solutions. Our philosophy centres on innovation, quality, and customer satisfaction. We strive to provide cutting-edge products and services that enhance comfort and efficiency. Our vision is to lead the industry through sustainable practices and continuous improvement, ensuring the highest standards of performance and reliability for our clients.

Key features used outside of core:

AMM, Phocas
FastClose
Automation Studio
Bartender Enterprise

Overview

- Different methods of Generating Bartender Labels in Kinetic
- Standard Kinetic Bartender Functionality
- On Demand Labels for non-Bartender enabled Reports. The old way
- REST Integration using Bartender REST API and Integration Webservices

What is Bartender?

- Software for designing and printing labels, barcodes, RFID tags, and more.
- Used by businesses of all sizes to improve efficiency, accuracy, and compliance in labeling and identification processes.
- Bartender offers a range of features, including an easy-to-use design tool, database connectivity, and automation capabilities, making it a powerful solution for managing labeling and identification needs.

Standard Kinetic Bartender Report Process 5 steps

We trigger an update (Std Data Directive) on

- PartTran
- JobMtl
- RMAHead
- OrderHead
- ShipHead
- RcvDtl
- PkgControlHeader

Call AutoPrint based on specific condition to execute a specified report style

- GenInv
- GenJob
- GenQA
- GenSO
- GenShip
- GenRcpt
- GenPCID

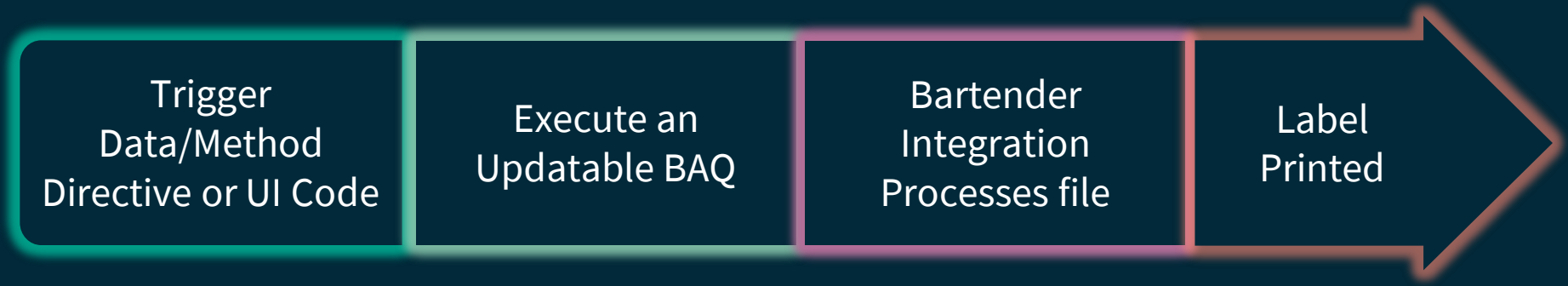
Output of the executed Report Style and Data Definition. Using stored parameters for:

- Report Location (Bartender Template)
- Output Location

Preconfigured Bartender File Integration polls the configured path output location from the report style and processes the generated file.

Success!

Kinetic Bartender – uBAQ/Func – Other Reports 4 Steps



We get to decide how, what and when to print a label

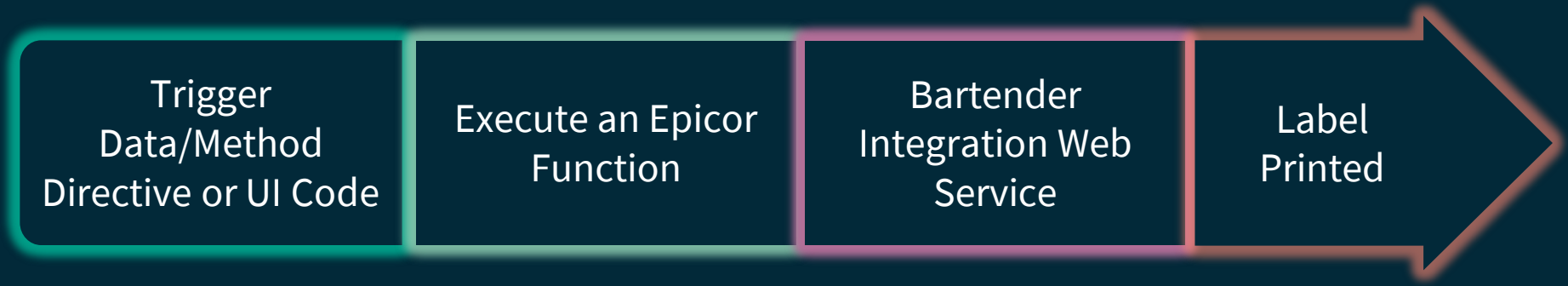
Updatable BAQ generates a file with specific Bartender command syntax and data, or call an Epicor Function

User codes could be used for specific parameters e.g. Template file, Pickup folder, filename

Preconfigured Bartender File Integration polls the configured path from the report style and processes the generated file.

Success!

Kinetic Bartender Web Svc - Function Method – Other Reports 4 Steps



We get to decide how, what and when to print a label

Generic Function called with specific parameters to call a BAQ generate data Generate a REST Request to Bartender Web Service Integration

User codes used for specific parameters e.g. Template file, Pickup folder, filename

Preconfigured Bartender Web Service is executed with the request data.

Success!

Kinetic Bartender REST API- Function Method – Other Reports 3 Steps

We get to decide how, what and when to print a label

Generic Function called with specific parameters to call a BAQ generate data Generate a REST Request to Bartender REST API

User codes used for specific parameters e.g. Template file, Pickup folder, filename

Success!

Snoozefest Alert ! - REST API Basics

Bartender Integration (WebSvc) REST API Vs BT REST API

- **Bartender Integration REST API**
  - Introduced in Bartender 2016
  - Each Web Service Integration is its own unique endpoint (URL) based on its name.
  - GET Integrations allow only a 1 label per request. Variables in Request URL
  - POST Integrations allow multiple labels per request. Variables in Body of Request (using csv)

- **Bartender REST API (Actions)**
  - Introduced in Bartender 2022. Gives another way to print documents without using an integration.
  - Can use BTXML, YAML or JSON
    - Print Actions
    - Input Actions
    - Output Action
    - Execute Actions
    - File Actions
    - Transform Actions
    - Database Actions

Refer Bartender REST API Overview

Pros and Cons Bartender Integration (Web Service)

- **Pros**
  - You have control of the resource id/URL via the integration name
  - Integration processing shows in Bartender Admin for monitoring.
  - A simple low code tool for creating complex integrations
  - No file generation needed
- **Cons**
  - Adds a layer of complexity to the solution that may not be needed
  - Can only follow workflow built into the Integration
  - GET Method 1 label
  - POST Method 1 to many labels (using CSV)
  - Limited number of characters can only be passed in the URL (GET)

Pros and Cons Bartender REST API

- **Pros**
  - No intermediary process needed
  - No file generation needed
  - One Resource/URL for all print jobs
  - Can dynamically modify workflow by using other API commands in your request payload
  - Use Action Groups to print multiple labels
- **Cons**
  - Have to roll your own monitoring solution unless you want to use History Explorer
  - Relatively New.
  - Need to use Reprint Console or bespoke reprint method to reprint labels.

The Problem

The Problem

- **Existing Legacy Requirements Needed a Bespoke Solution:**
  - Current solution was batch solution executing every 5 minutes regardless if there was anything to print. Wanted to move to a more immediate label print scenario.
  - Needed to be able to Reprint Labels without having to leave the Kinetic UI
  - Needed to be able to Reprint Labels without having to install additional software
  - Improve label print reliability. Generating files across the network was unreliable.
  - Why not use the GenJob Report style?…….The RDD did not provide the flexibility for the data needed in the specific label

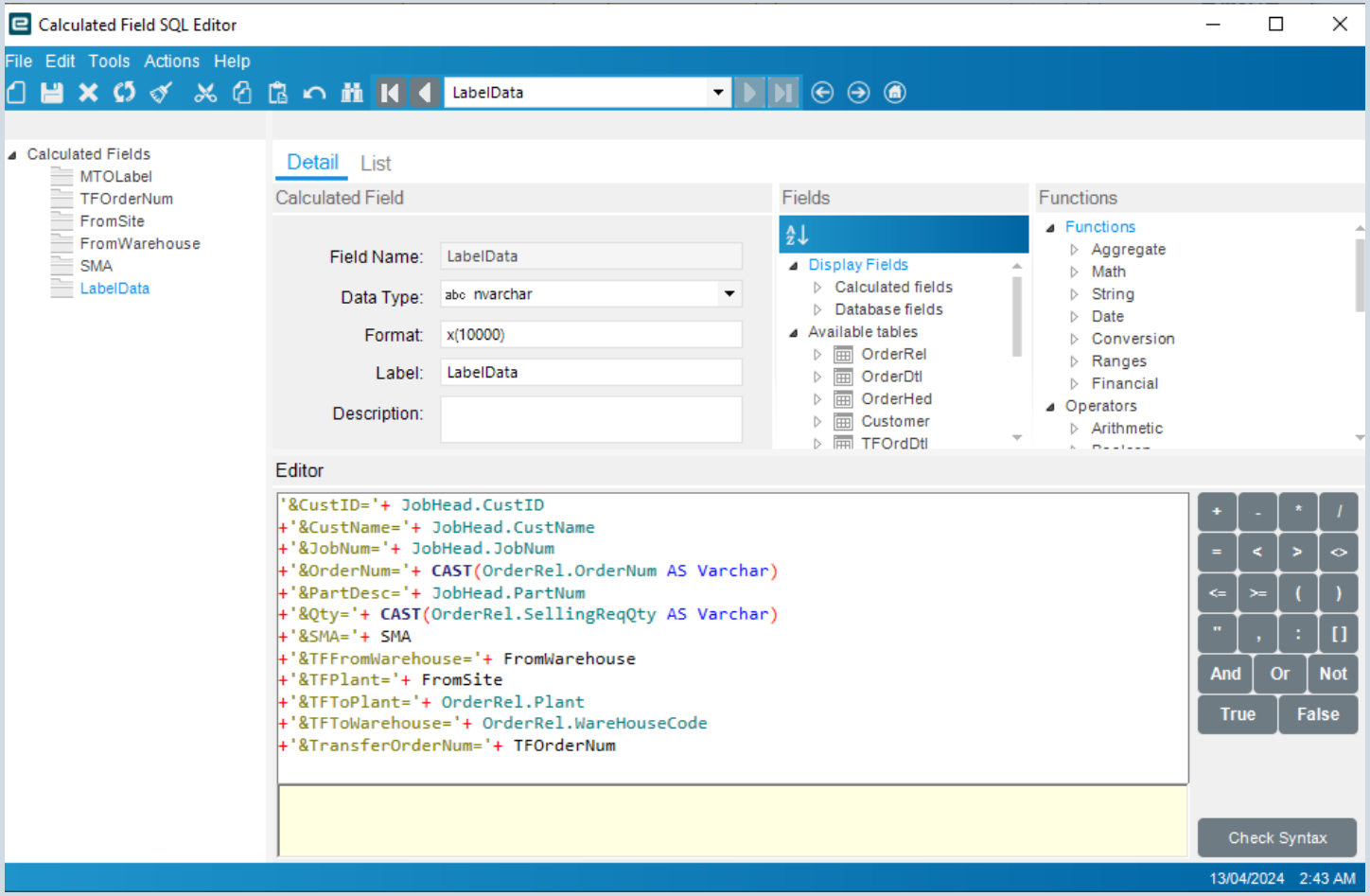
The Solution

Create Your BAQ

- Various ways
- We create a BAQ with a Calculated Field concatenating the Named Data Sources For a GET Method

OR

- We serialize the column names from BAQ Results dataset into JSON – Put in Body (POST Method)

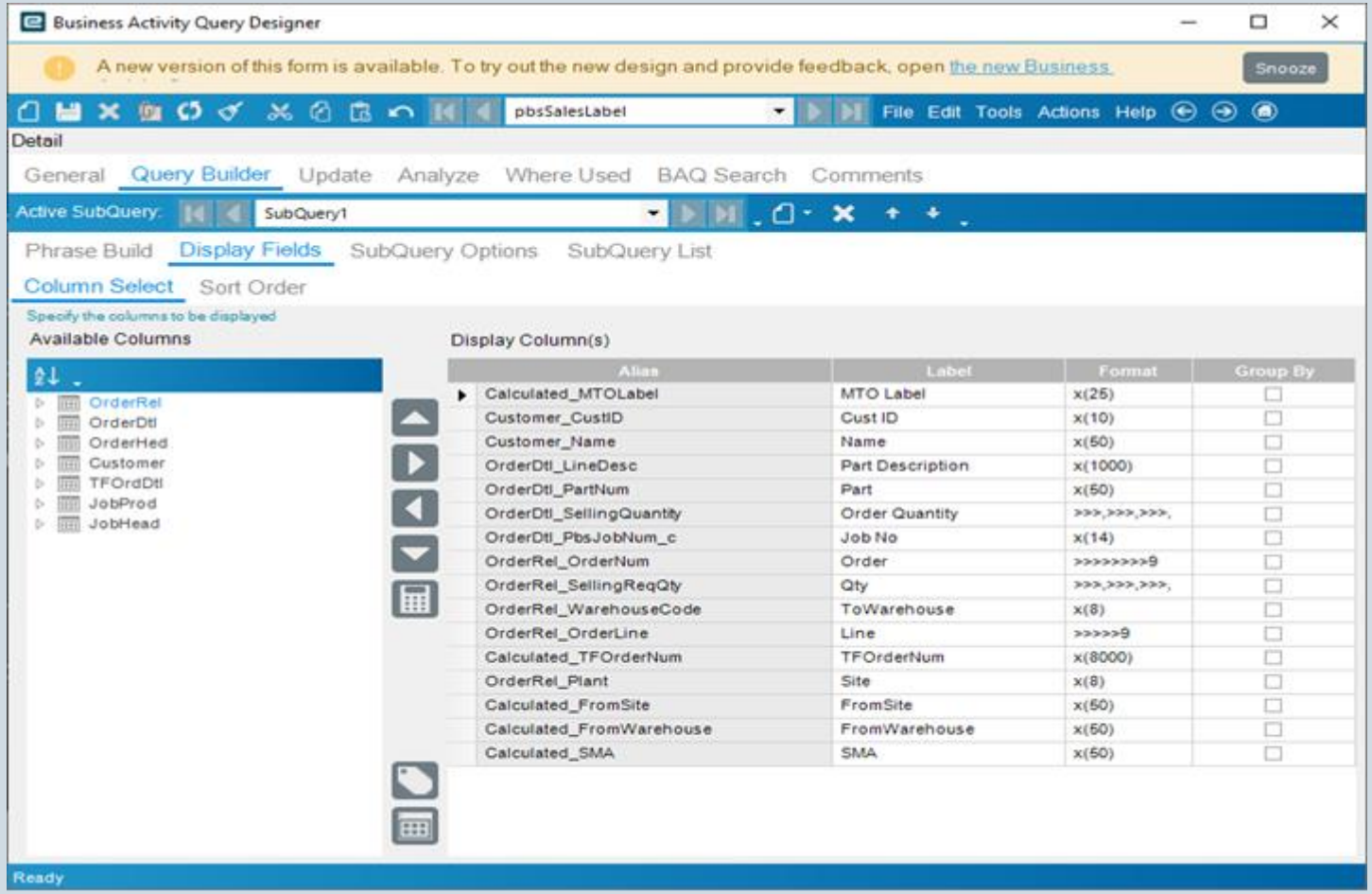


Create Your BAQ

- Various ways
- We create a BAQ with a Calculated Field concatenating the Named Data Sources For a GET type Integration

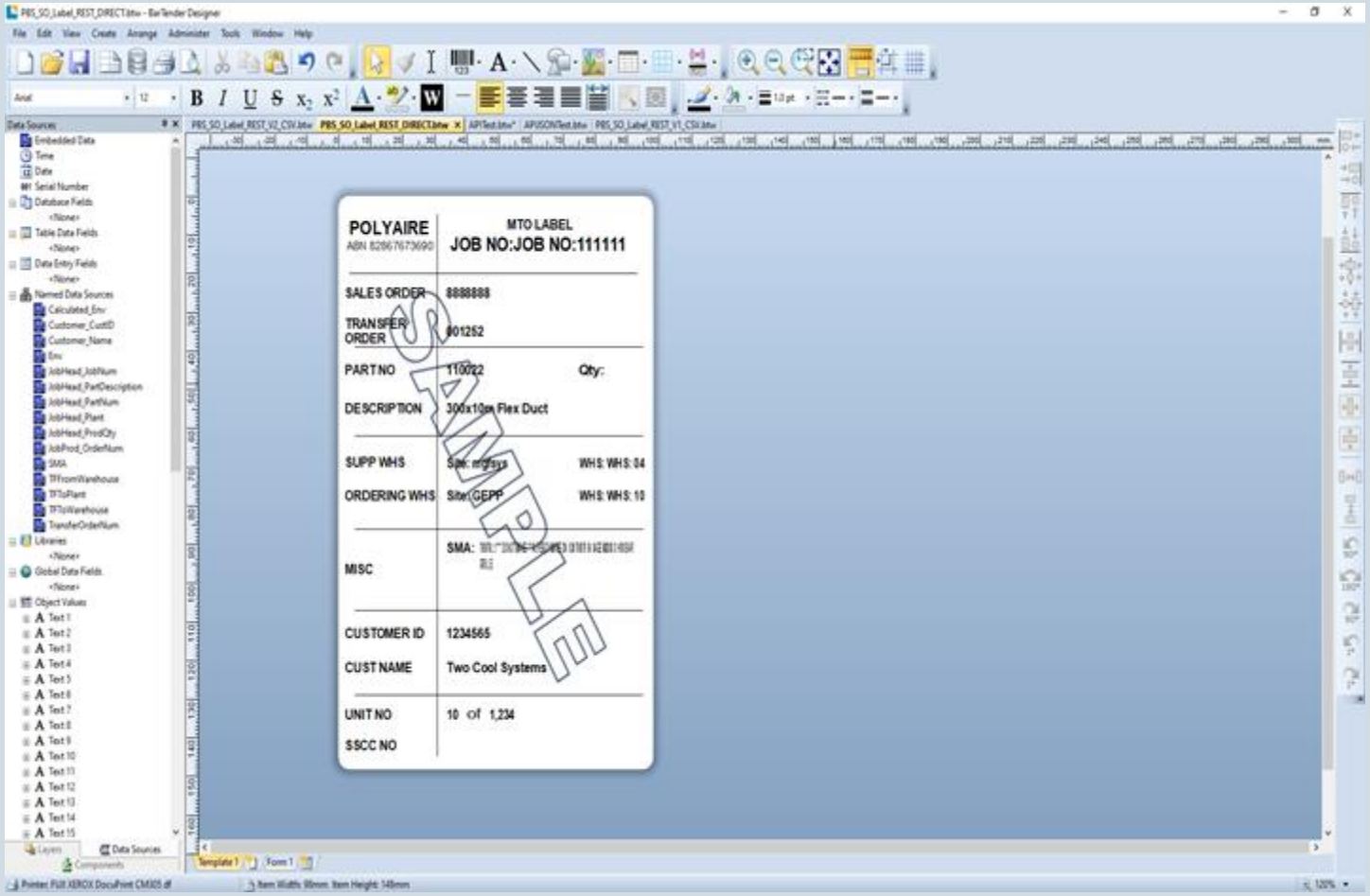
OR

- We manipulate the column names from BAQ in the Function into JSON



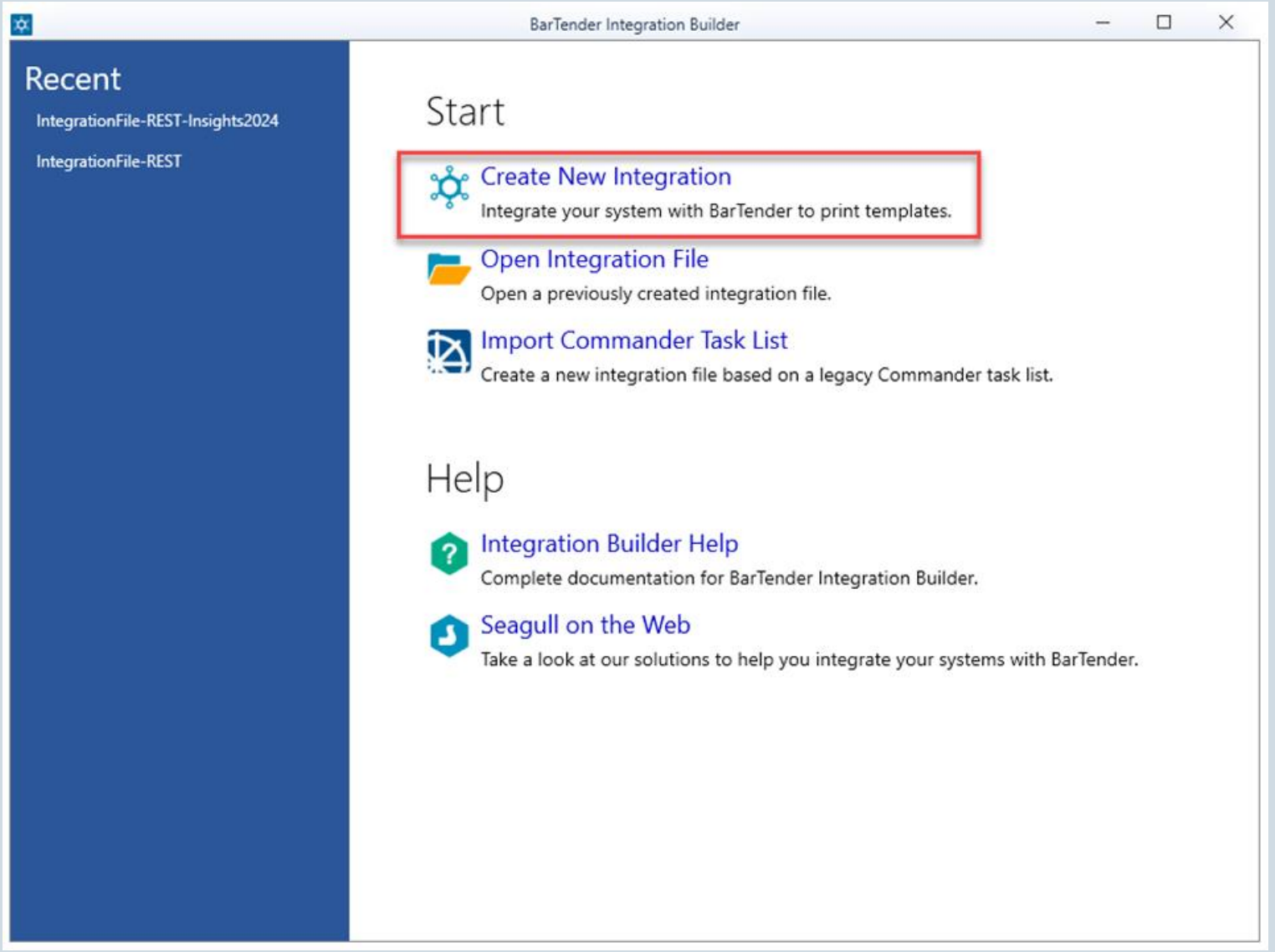
Create Bartender Label

- Use Named Data Sources rather than database fields
- Add in any other smarts to the label you need (numbering, watermarks etc.)



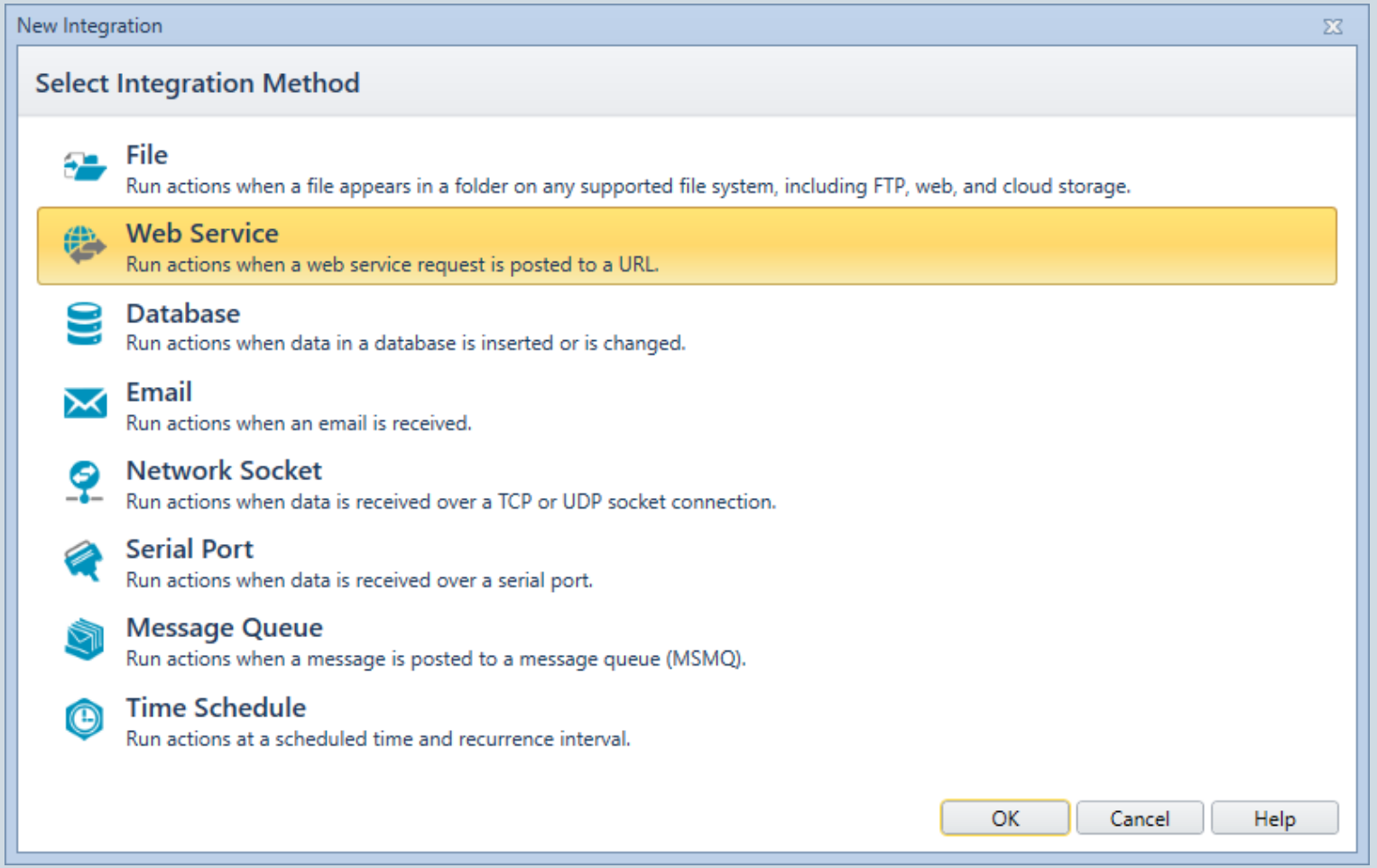
Create Bartender Integration

- Create New Integration



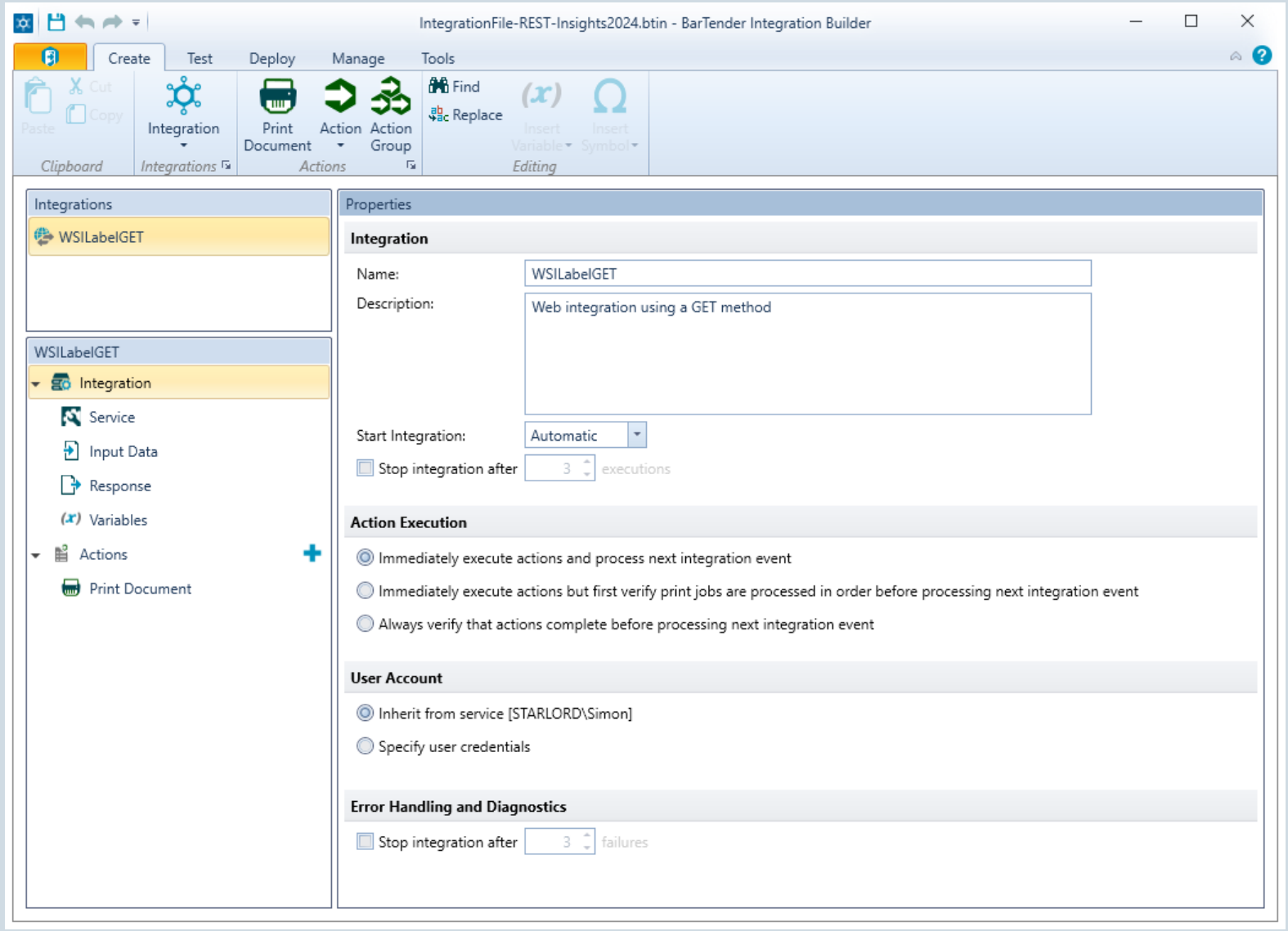
Create Bartender Integration

- Create New Integration
- Select Web Service



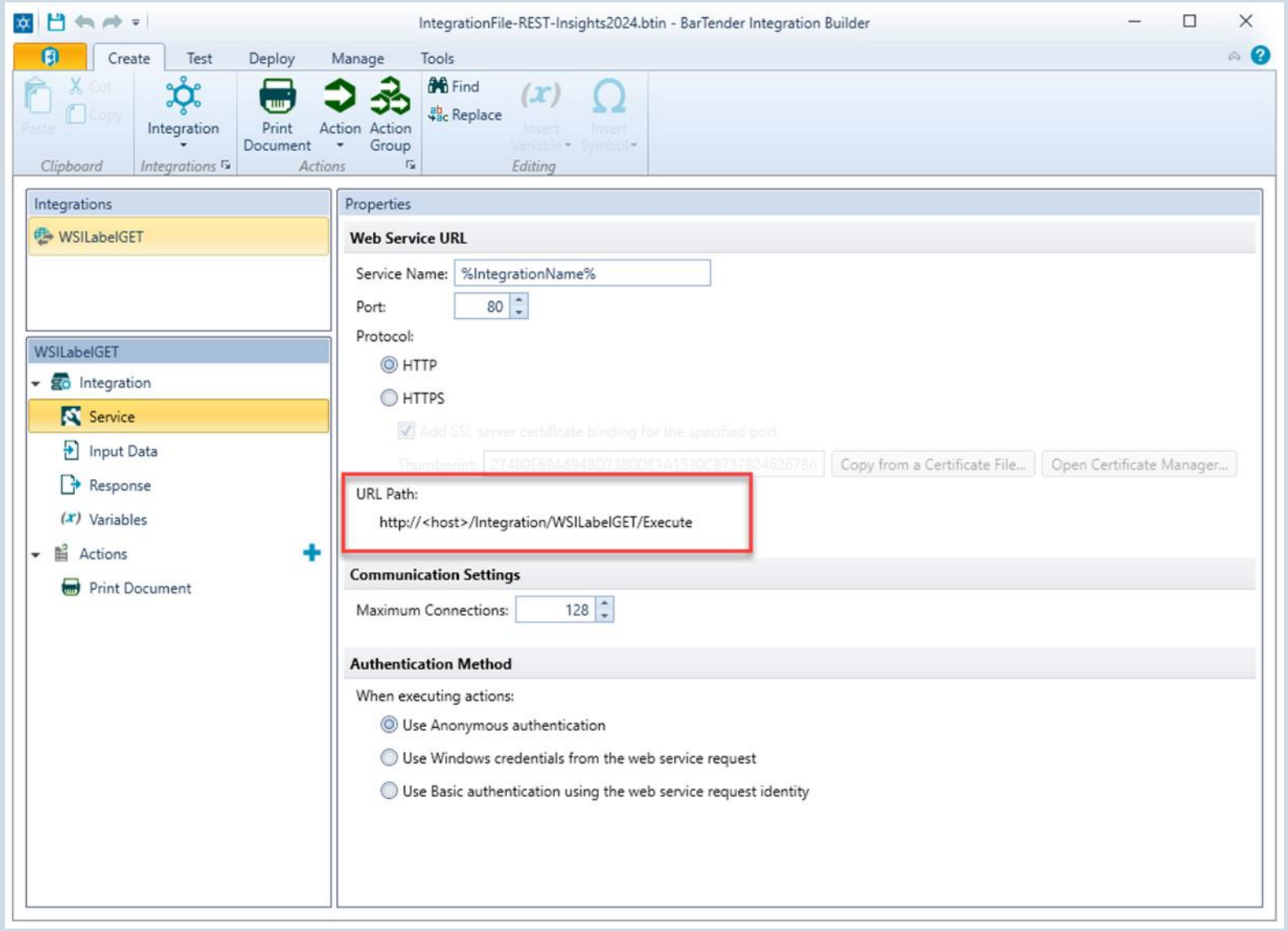
Create Bartender Integration

- Create New Integration
- Select Web Service
- Name the Integration



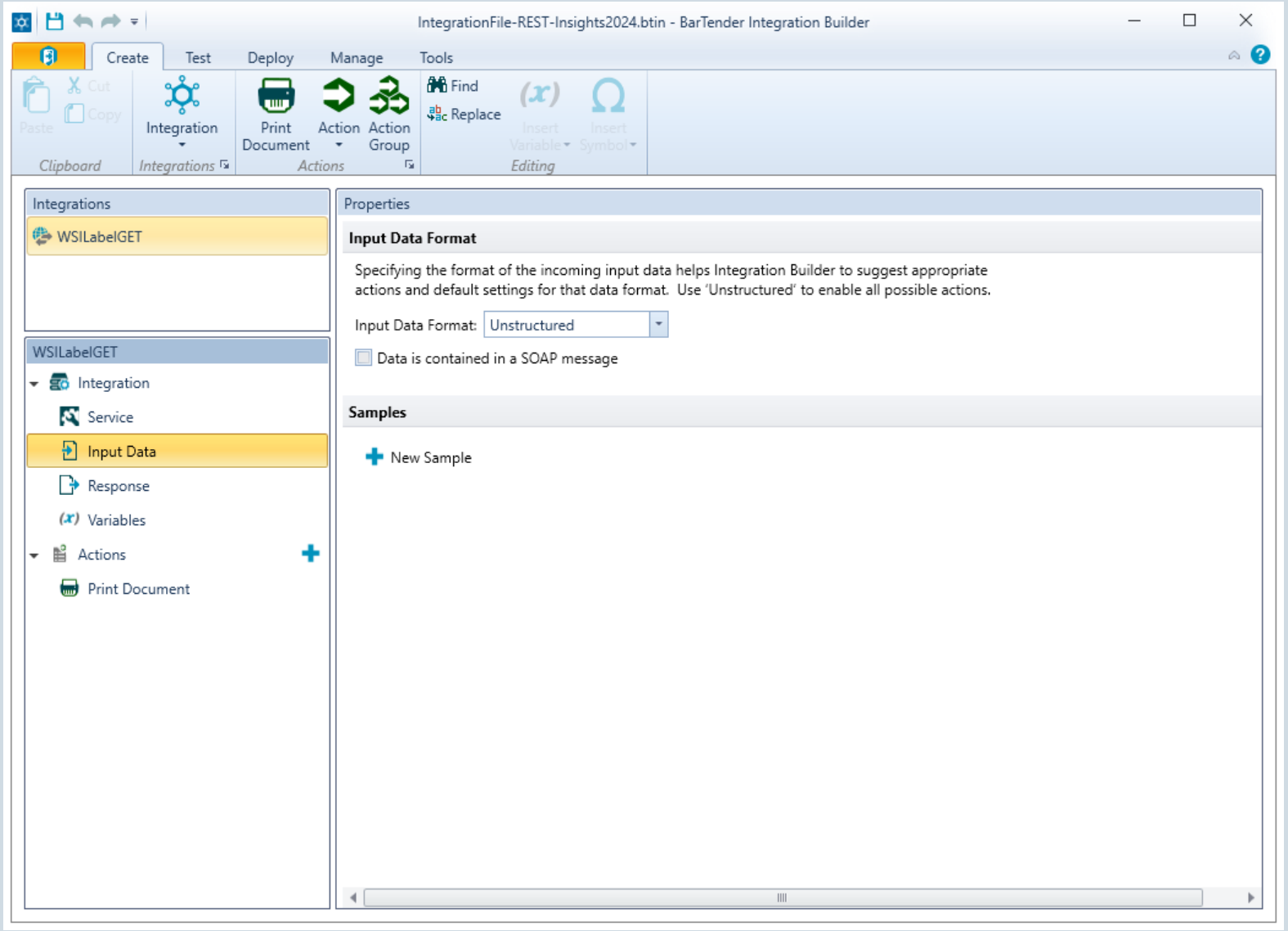
Create Bartender Integration

- Create New Integration
- Select Web Service
- Name the Integration
- Configure the Service and Security



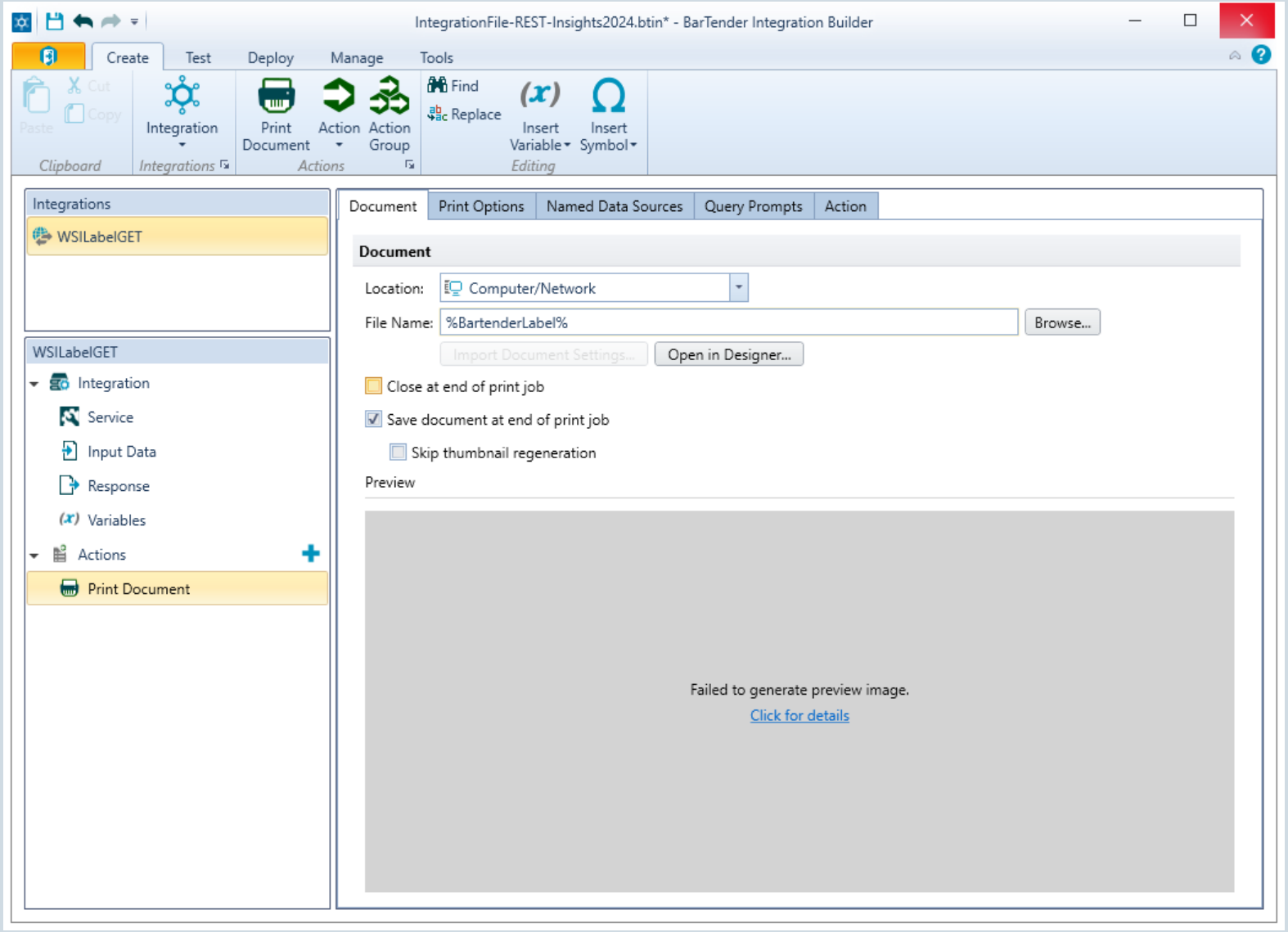
Create Bartender Integration

- Create New Integration
- Select Web Service
- Name the Integration
- Configure the Service and Security
- Select Input Data Format



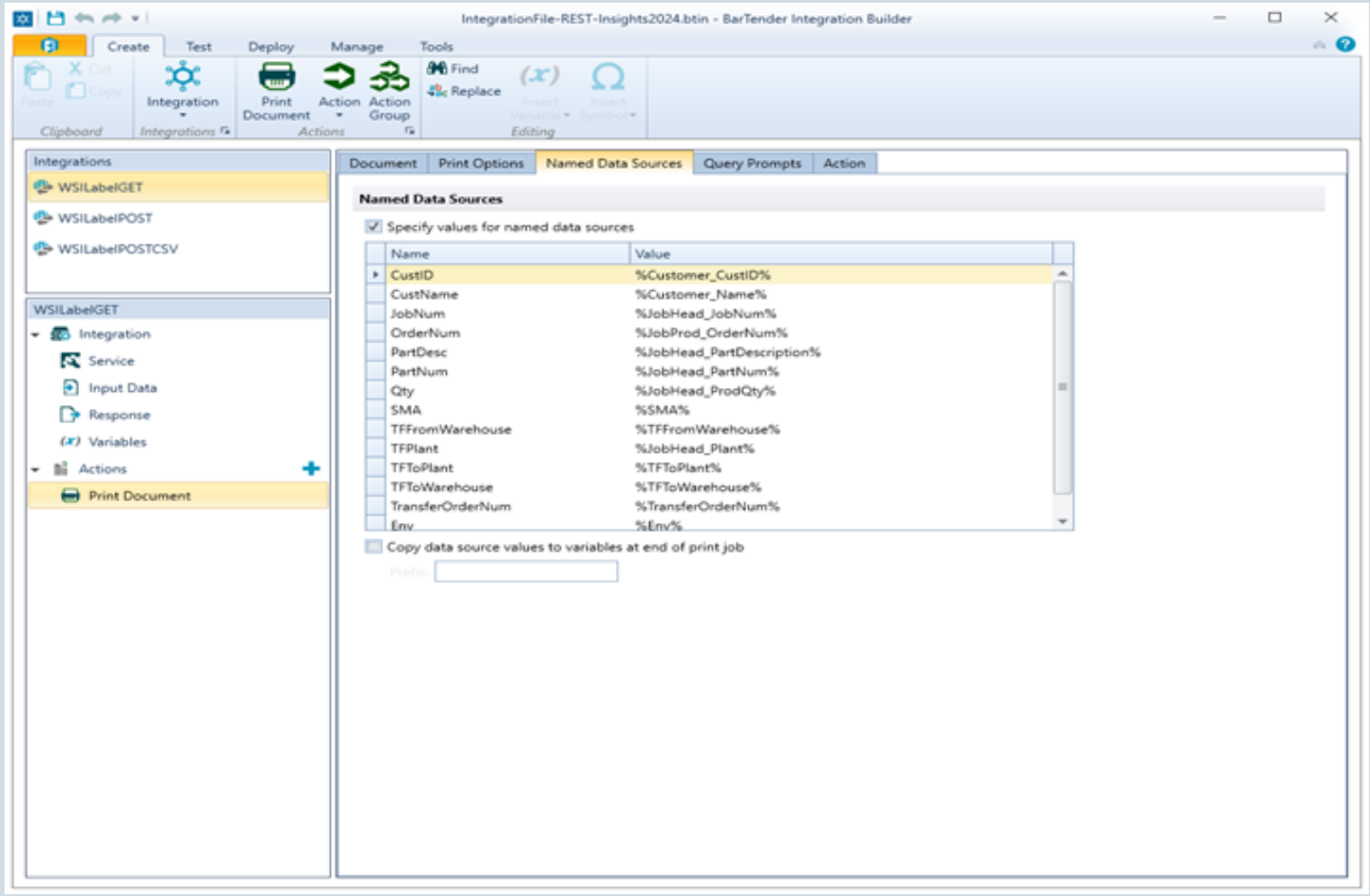
Create Bartender Integration

- Create New Integration
- Select Web Service
- Name the Integration
- Configure the Service and Security
- Select Input Data Format
- Configure Document Properties and Named Data Sources



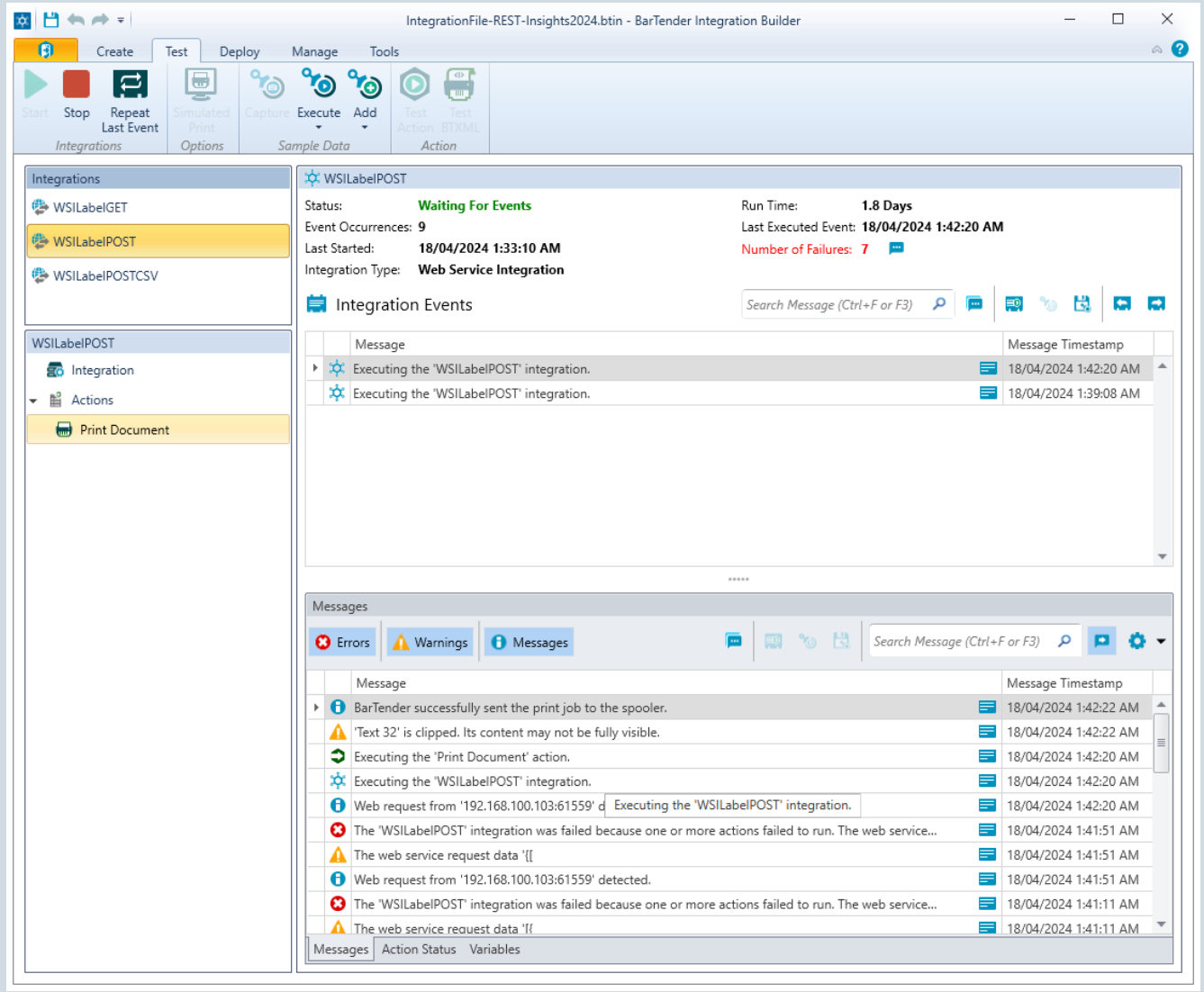
Create Bartender Integration

- Create New Integration
- Select Web Service
- Name the Integration
- Configure the Service and Security
- Select Input Data Format
- Configure Document Properties and Named Data Sources



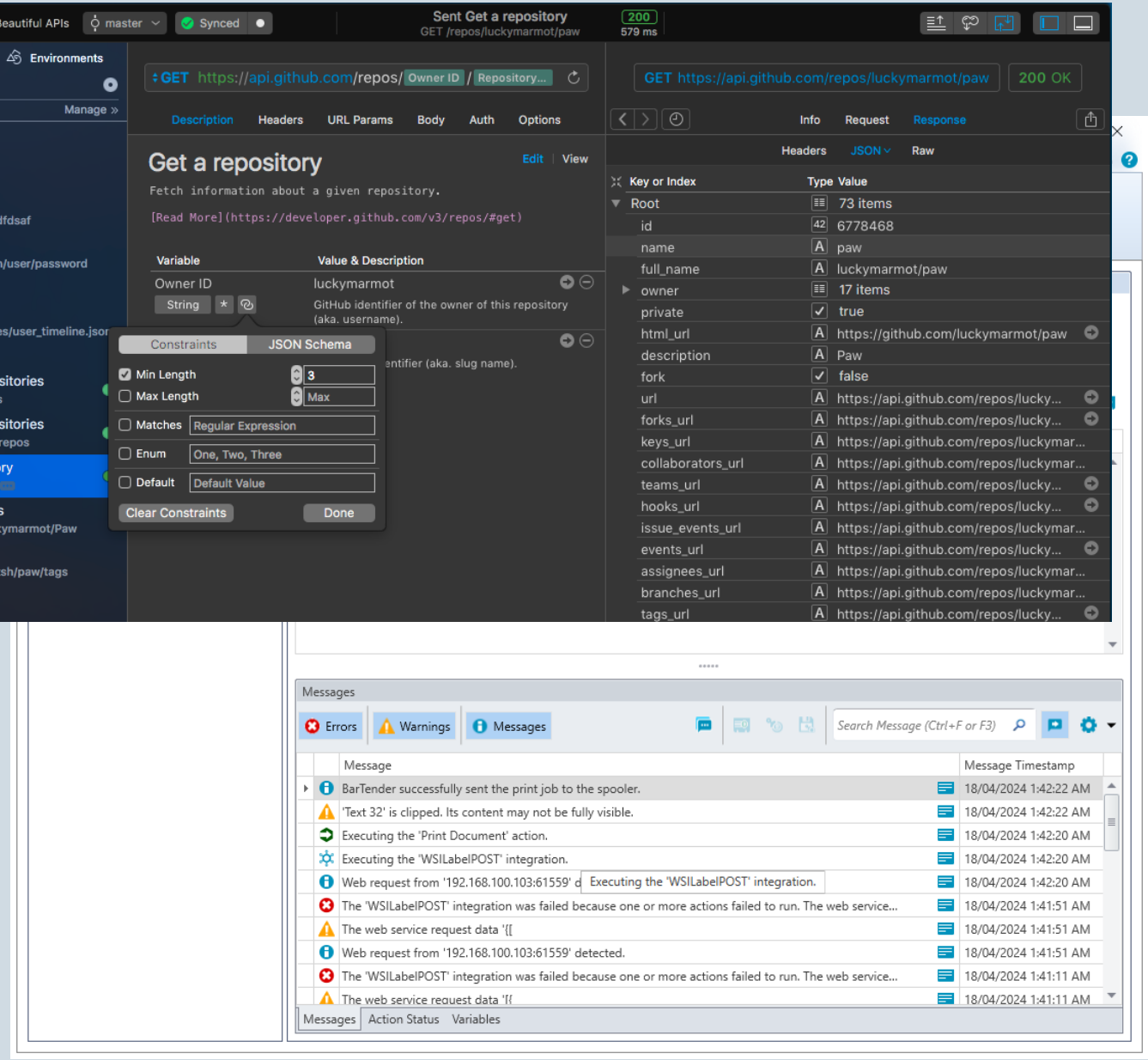
Test Bartender Integration

- Enable Test Mode in your integration
- Use your favourite REST Testing tool



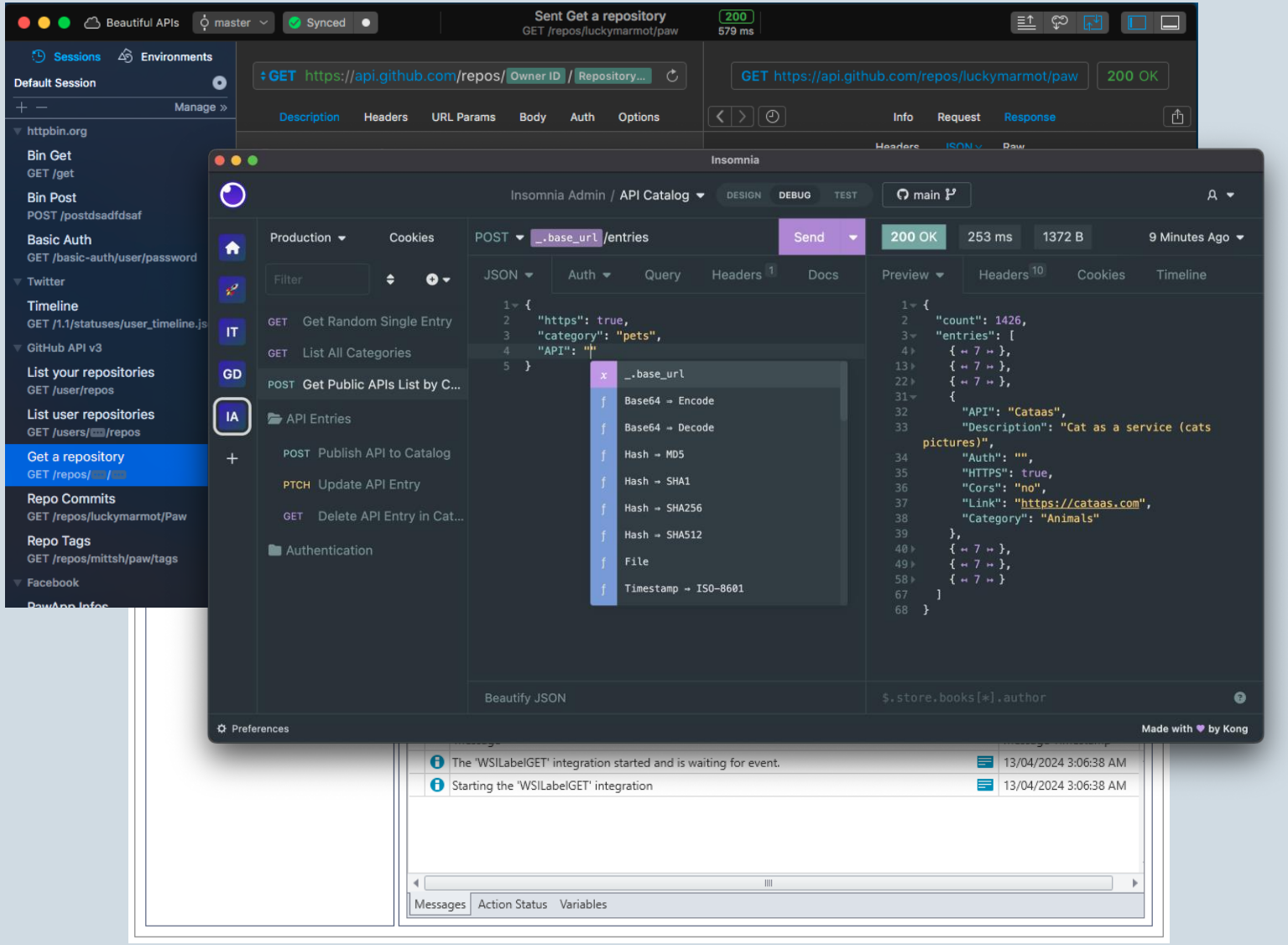
Test Bartender Integration

- Enable Test Mode in your integration
- Use your favourite REST Testing tool
- Rapid API



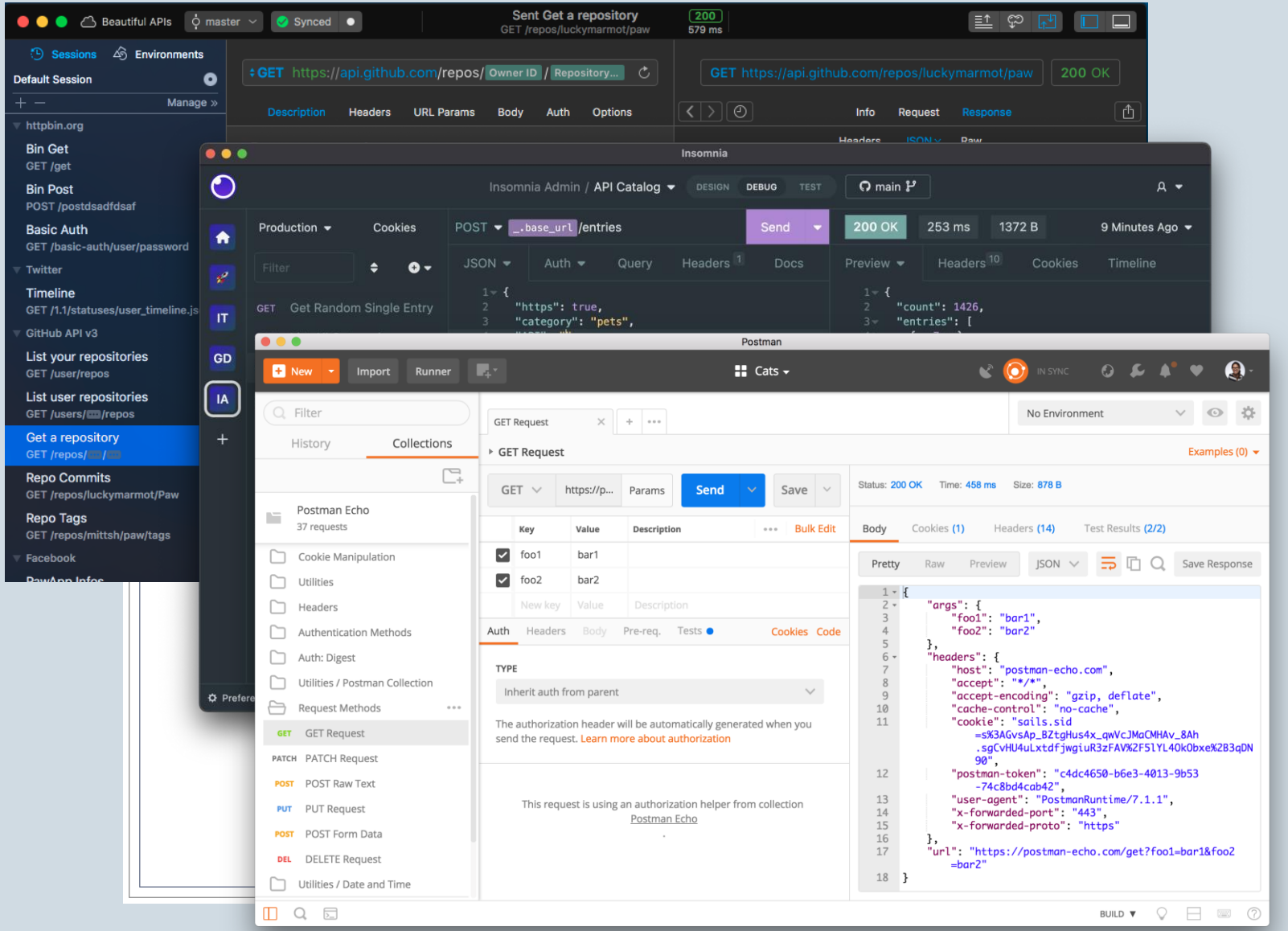
Test Bartender Integration

- Enable Test Mode in your integration
- Use your favourite REST Testing tool
- Rapid API, Insomnia



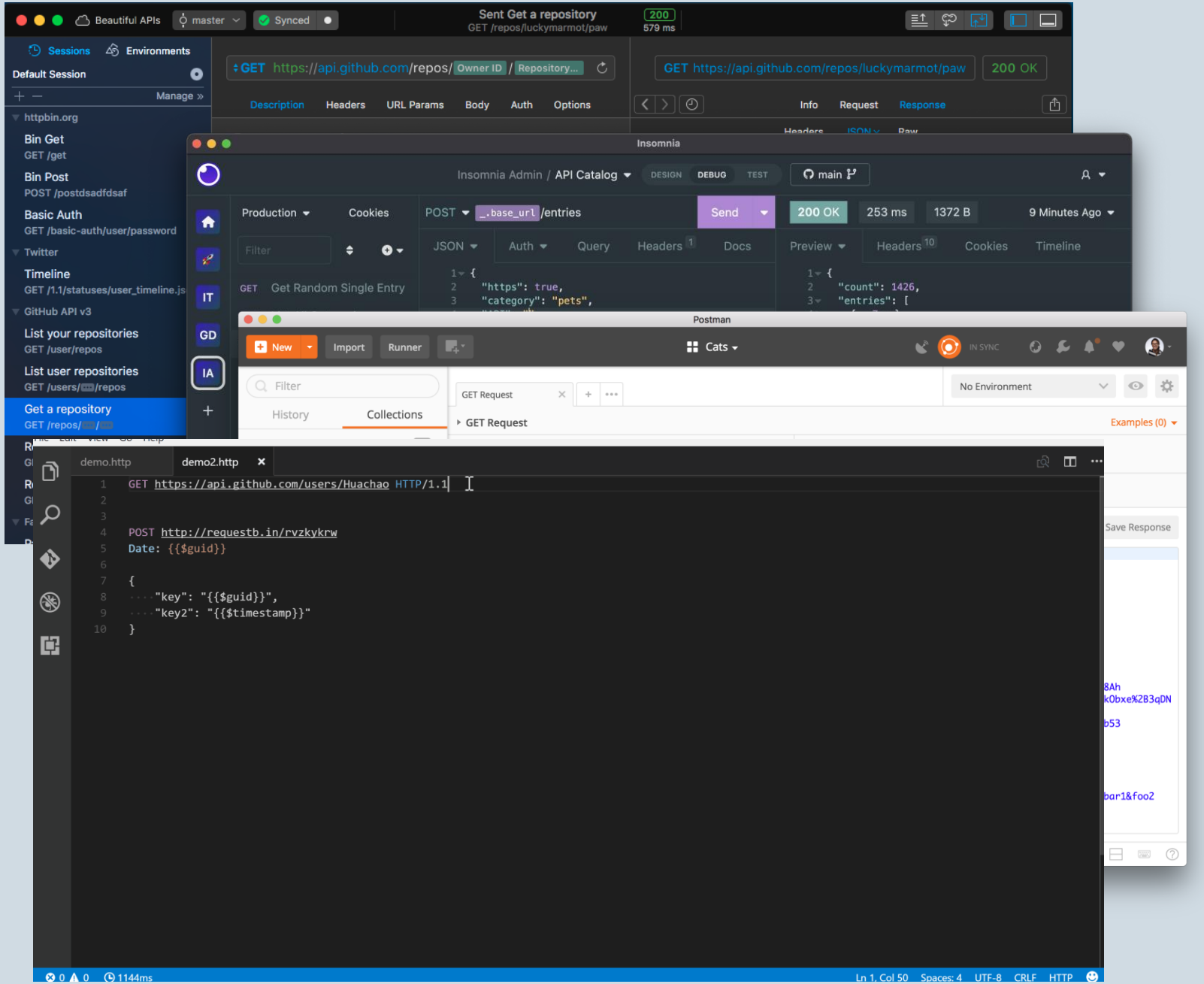
Test Bartender Integration

- Enable Test Mode in your integration
- Use your favourite REST Testing tool
- Rapid API, Insomnia, Postman,



Test Bartender Integration

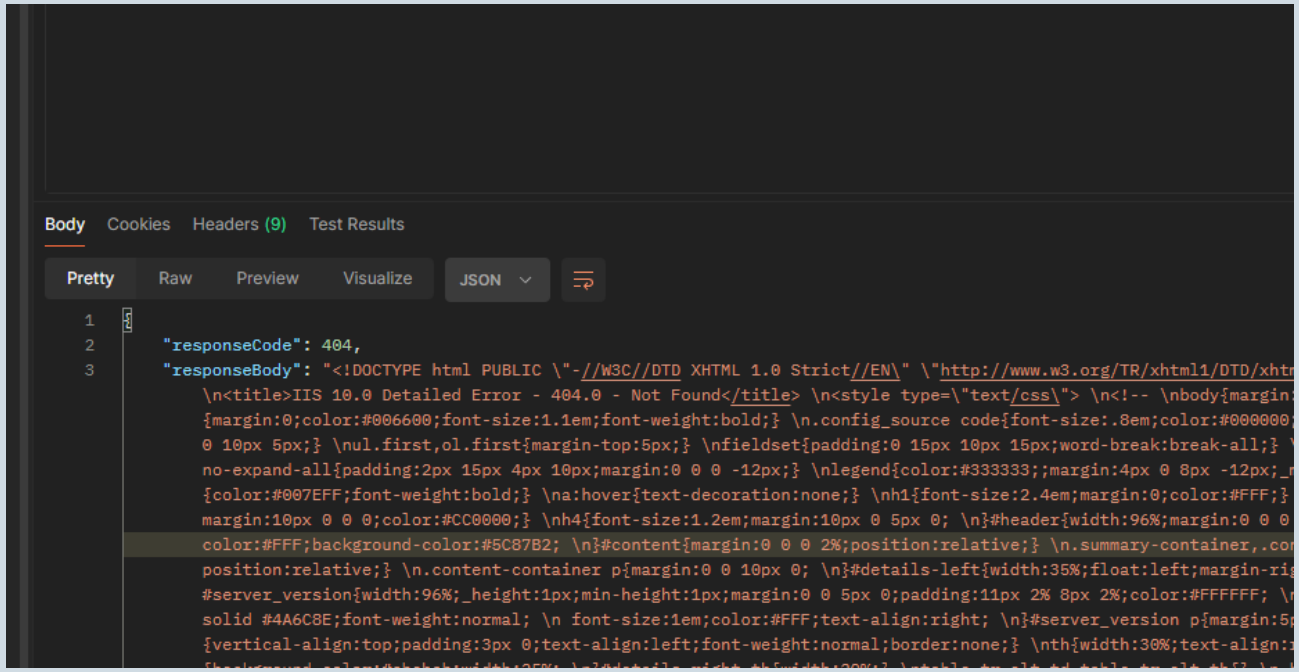
- Enable Test Mode in your integration
- Use your favourite REST Testing tool
- Rapid API, Insomnia, Postman, VSCode



Common Problems

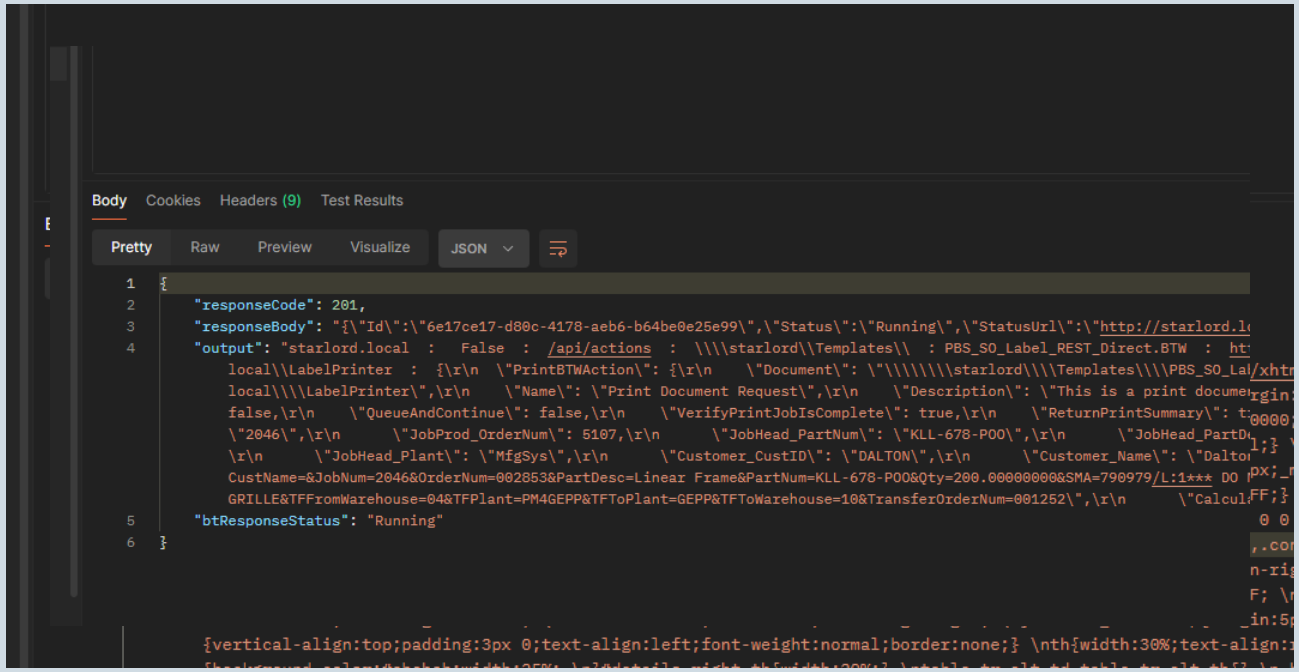
Common Problems

- 404 Error



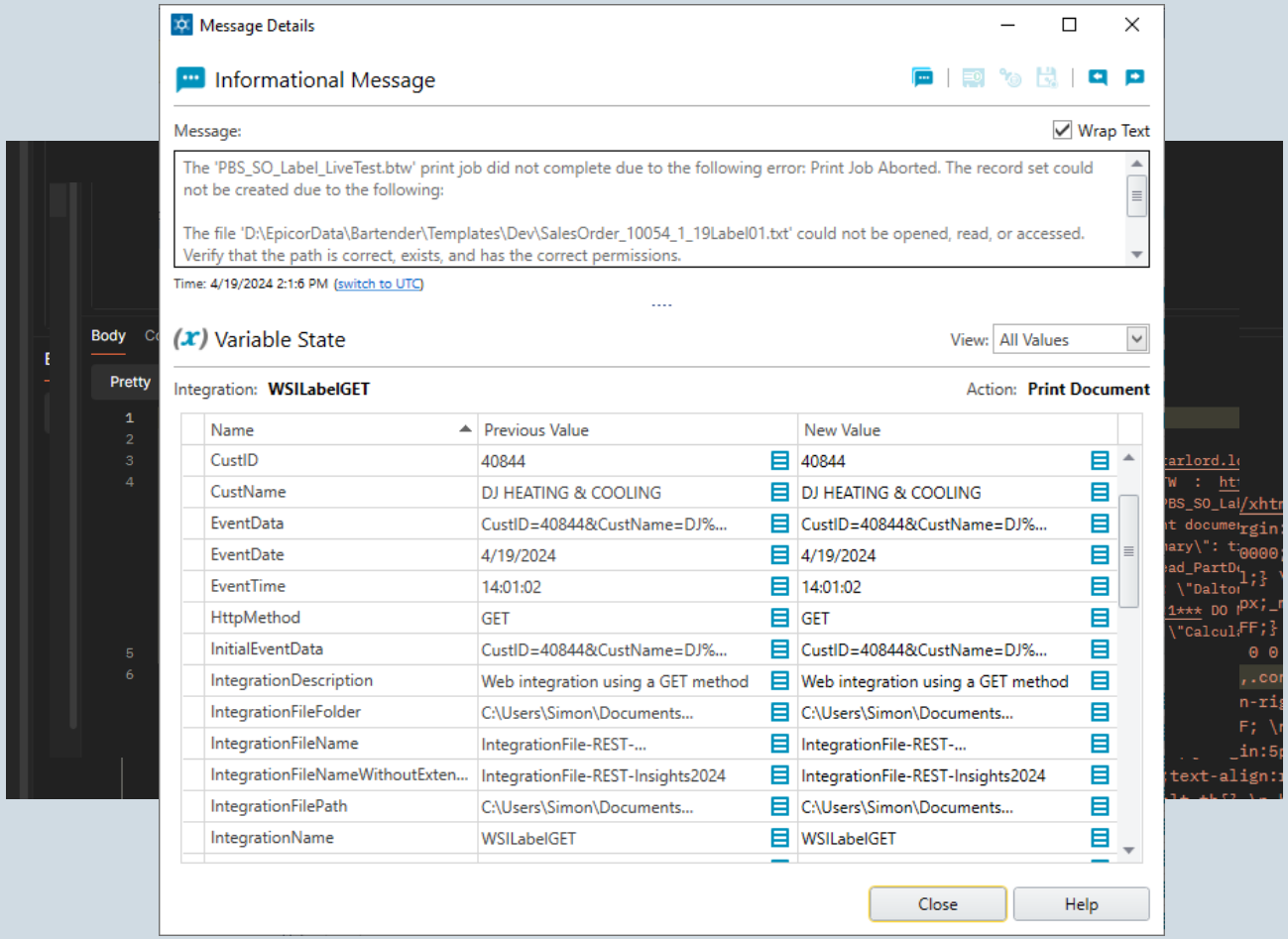
Common Problems

- 404 Error
- Expecting a 200 Ok when Bartender returns a 201 Created



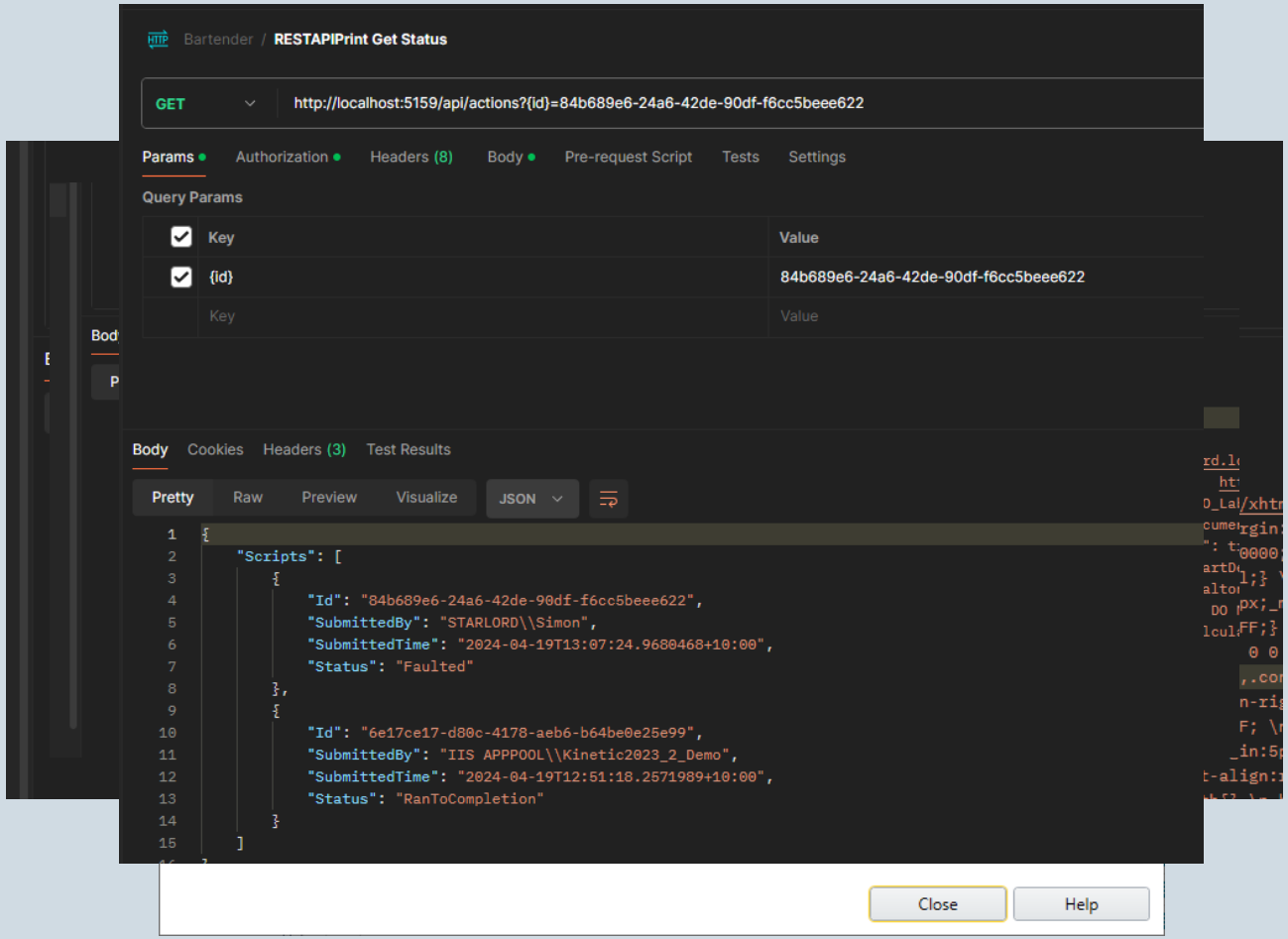
Common Problems

- 404 Error
- Expecting a 200 Ok when Bartender returns a 201 Created
- File Could not be opened, read or accessed



Common Problems

- 404 Error
- Expecting a 200 Ok when Bartender returns a 201 Created
- File Could not be opened, read or accessed
- No Label being printed. (Using REST API)



Create Your Function Both methods

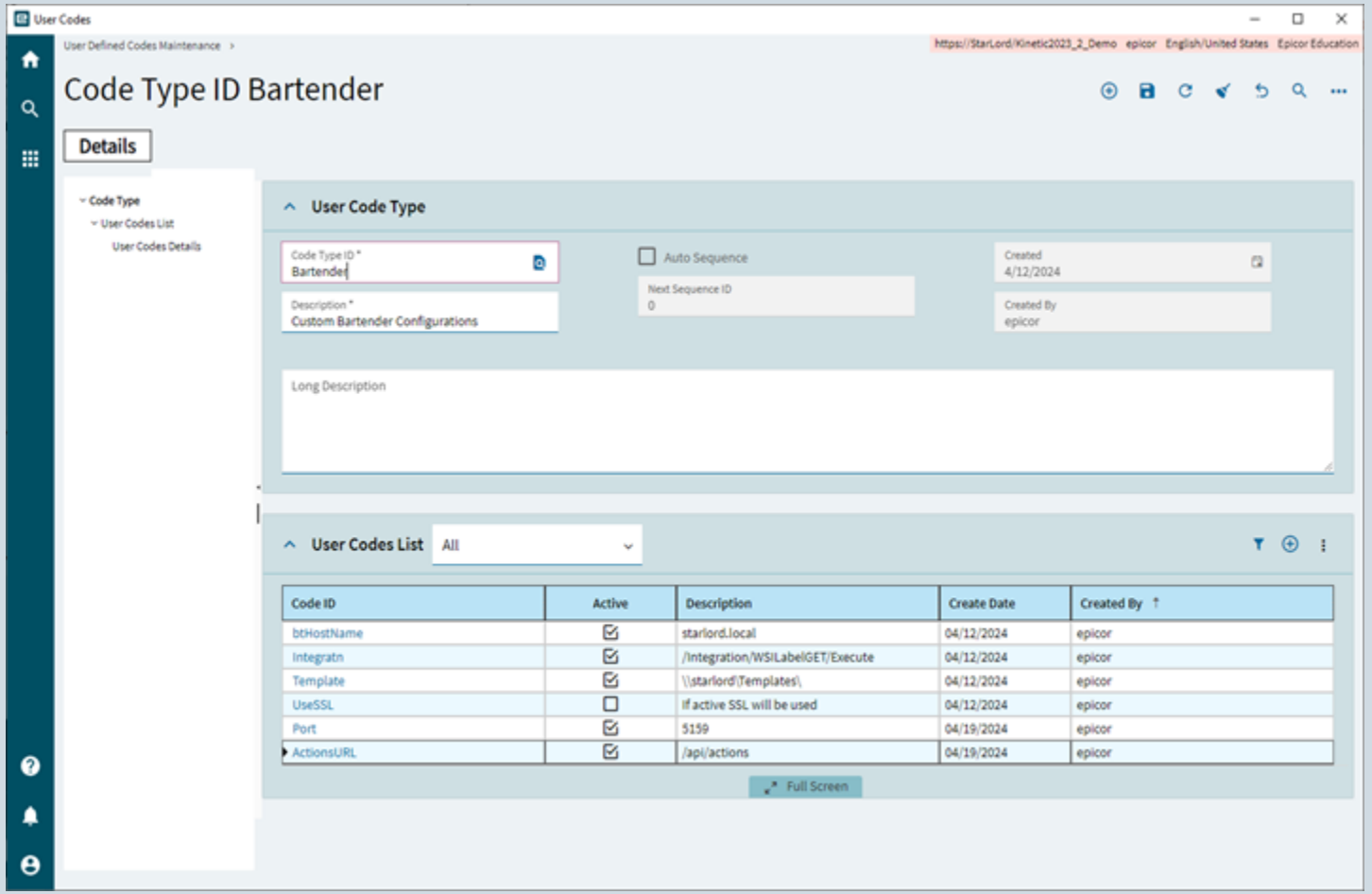
- Create for Using BT Integrations

Or

- Bartender REST API

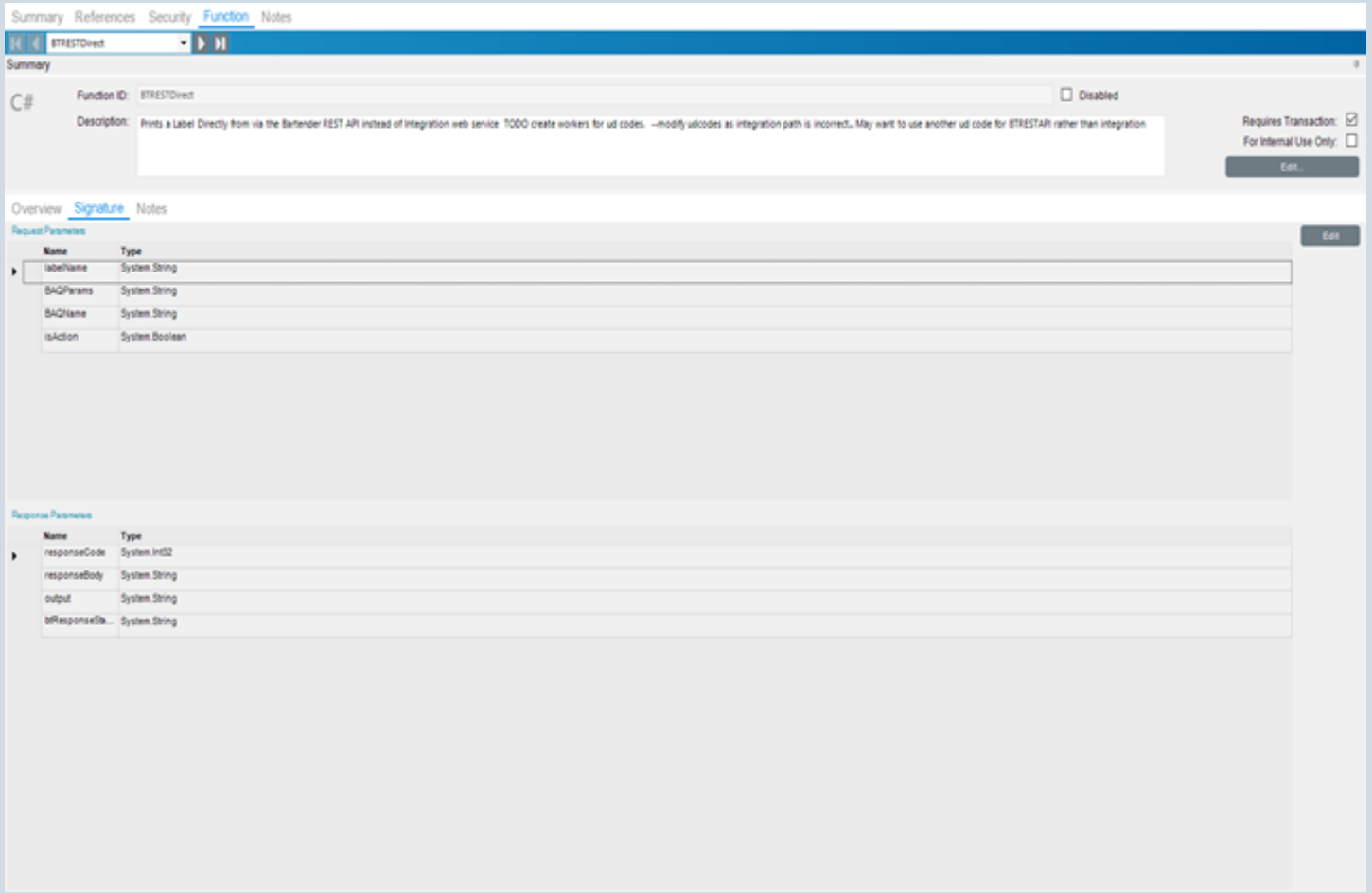
Create Your Function Both Methods

- Configure System variables as User Codes



Create Your Function Both Methods

- Configure System variables as User Codes
- Configure your signatures



Create Your Function

- Configure System variables as User Codes
- Configure your signatures
- Call the BAQ (or Not)

```
string printerName = Db.SysPrinter.Where(x => x.Company == Session.CompanyID && x.PbsSiteID_

if (btHostName == null || integrationPath == null || templatePath == null || printerName ==
{
    output = $"Bartender User Code Parameters or Printer not configured correctly, please re
    return;
}

if (useSSL)
    protocol = "https://";

// Get the Label Data and add to output
var result = this.EfxLib.Helpers_Demo.ExecuteBAQWithParams(this.BAQParams, this.BAQName);

string NamedDataSourcesjson = string.Empty;

if (result.dsResult.Tables.Contains("Results"))
{
    // Get the "Results" table
    System.Data.DataTable table = result.dsResult.Tables["Results"];

    foreach (var row in table.AsEnumerable())
    {
        // Convert the row to a dictionary
```

Create Your Function

- Configure System variables as User Codes
- Configure your signatures
- Call the BAQ (or Not)
- Manipulate BAQ Output

```
42      if (result.dsResult.Tables.Contains("Results"))
43      {
44          // Get the "Results" table
45          System.Data.DataTable table = result.dsResult.Tables["Results"];
46
47          foreach (var row in table.AsEnumerable())
48          {
49              // Convert the row to a dictionary
50              var dict = table.Columns.Cast<DataColumn>()
51                  .ToDictionary(column => column.ColumnName, column => row[column]);
52
53              // Serialize the dictionary to JSON
54              NamedDataSourcesjson = JsonConvert.SerializeObject(dict, Formatting.Indented);
55          }
```

Create Your Function

- Configure System variables as User Codes
- Configure your signatures
- Call the BAQ (or Not)
- Manipulate BAQ Output

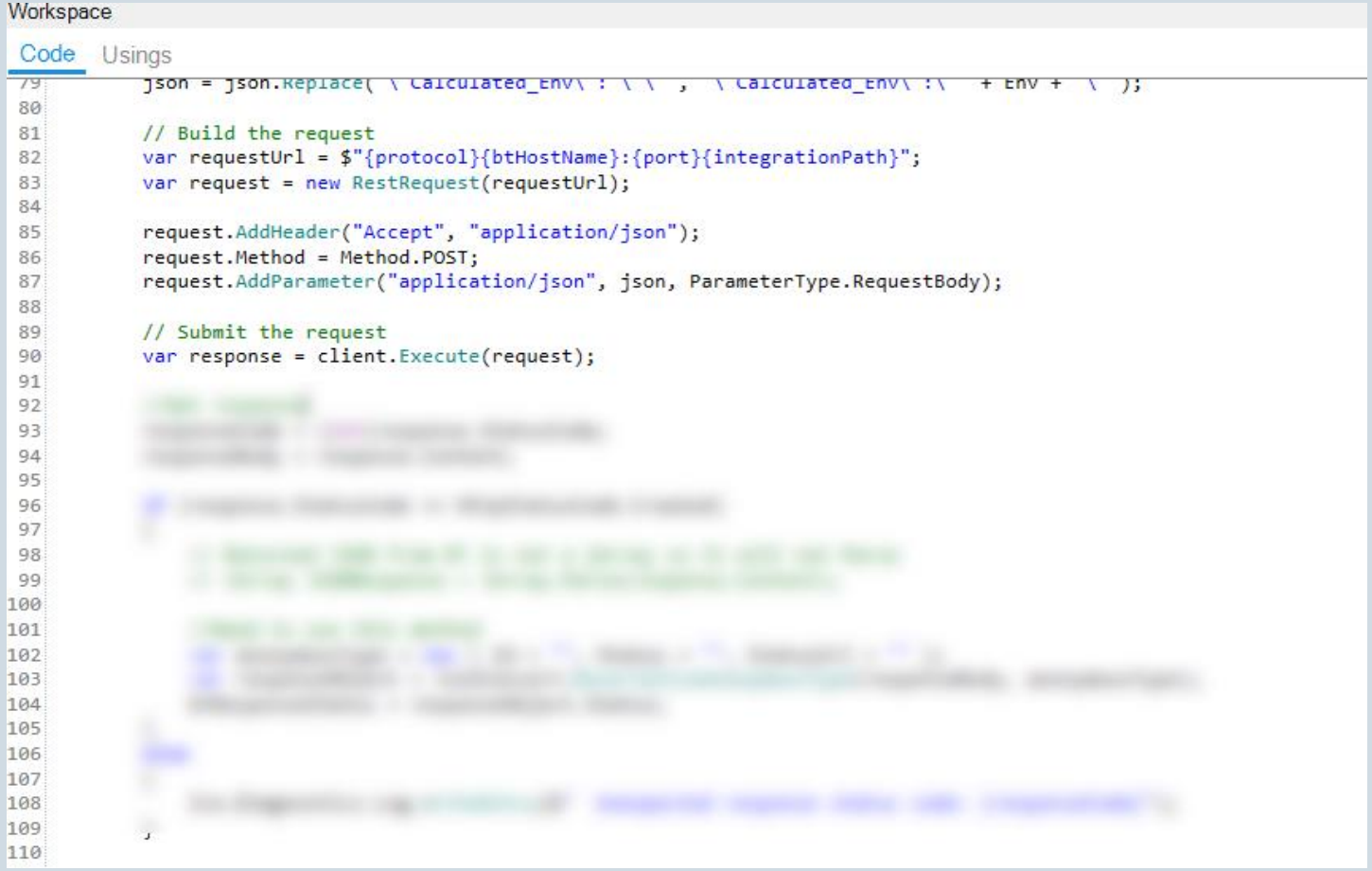
```
// Create the BT REST API Action Payload object to be serialized NamedDataSourcesjson is the list of Named D
var obj = new
{
    PrintBTWAction = new
    {
        Document = $@"{templatePath}{this.labelName}",
        Printer = printerName,
        Name = "Print Document Request",
        Description = "This is a print document request to print a barcode label.",
        SaveAfterPrint = false,
        QueueAndContinue = false,
        VerifyPrintJobIsComplete = true,
        ReturnPrintSummary = true,
        NamedDataSources = JsonConvert.DeserializeObject<Dictionary<string, object>>(NamedDataSourcesjson)

    }
};

// Serialize the object to JSON
string json = JsonConvert.SerializeObject(obj, Formatting.Indented);
```

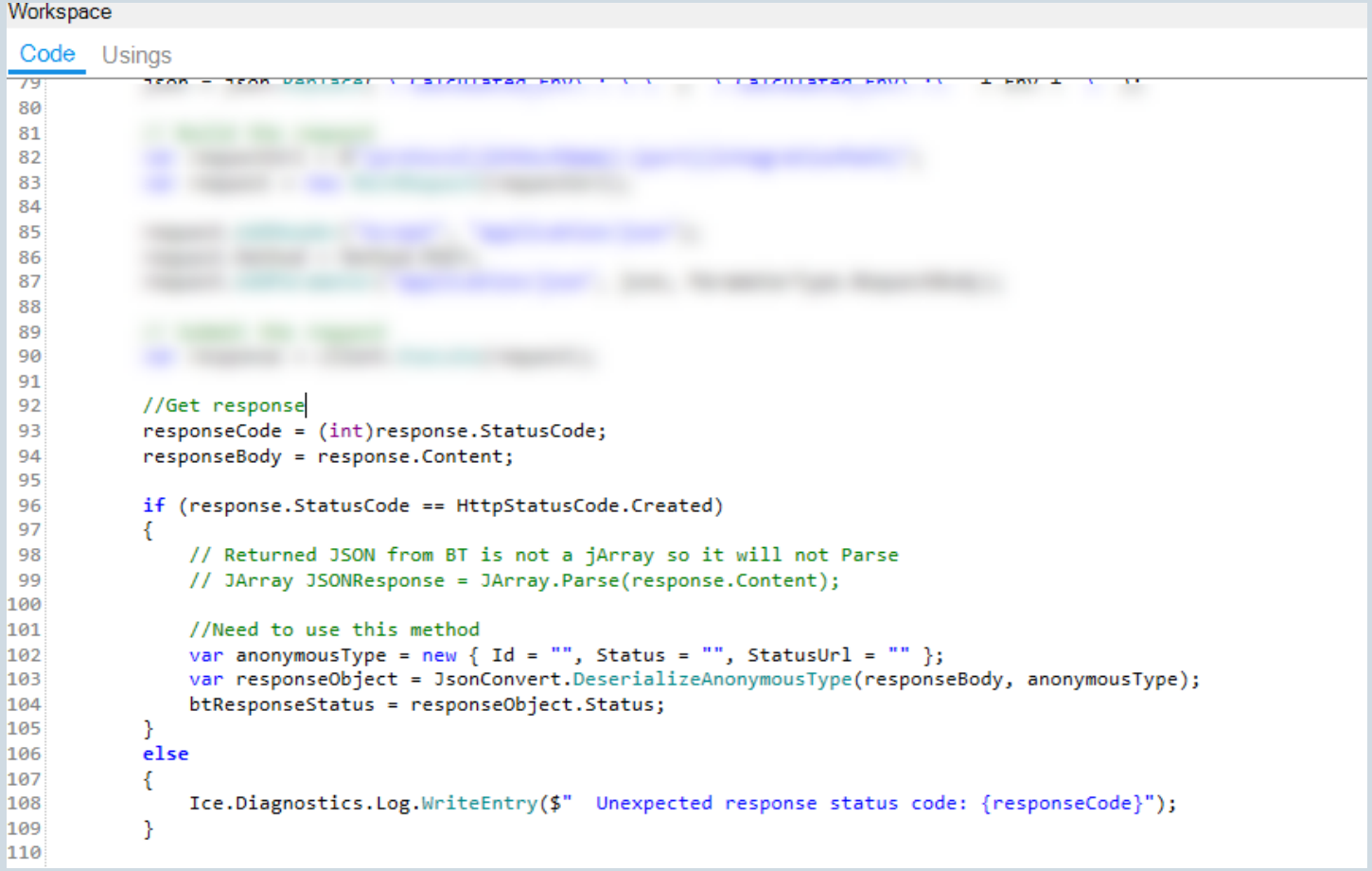
Create Your Function

- Configure System variables as User Codes
- Configure your signatures
- Call the BAQ (or Not)
- Manipulate BAQ Output
- Create and Send the REST Request



Create Your Function

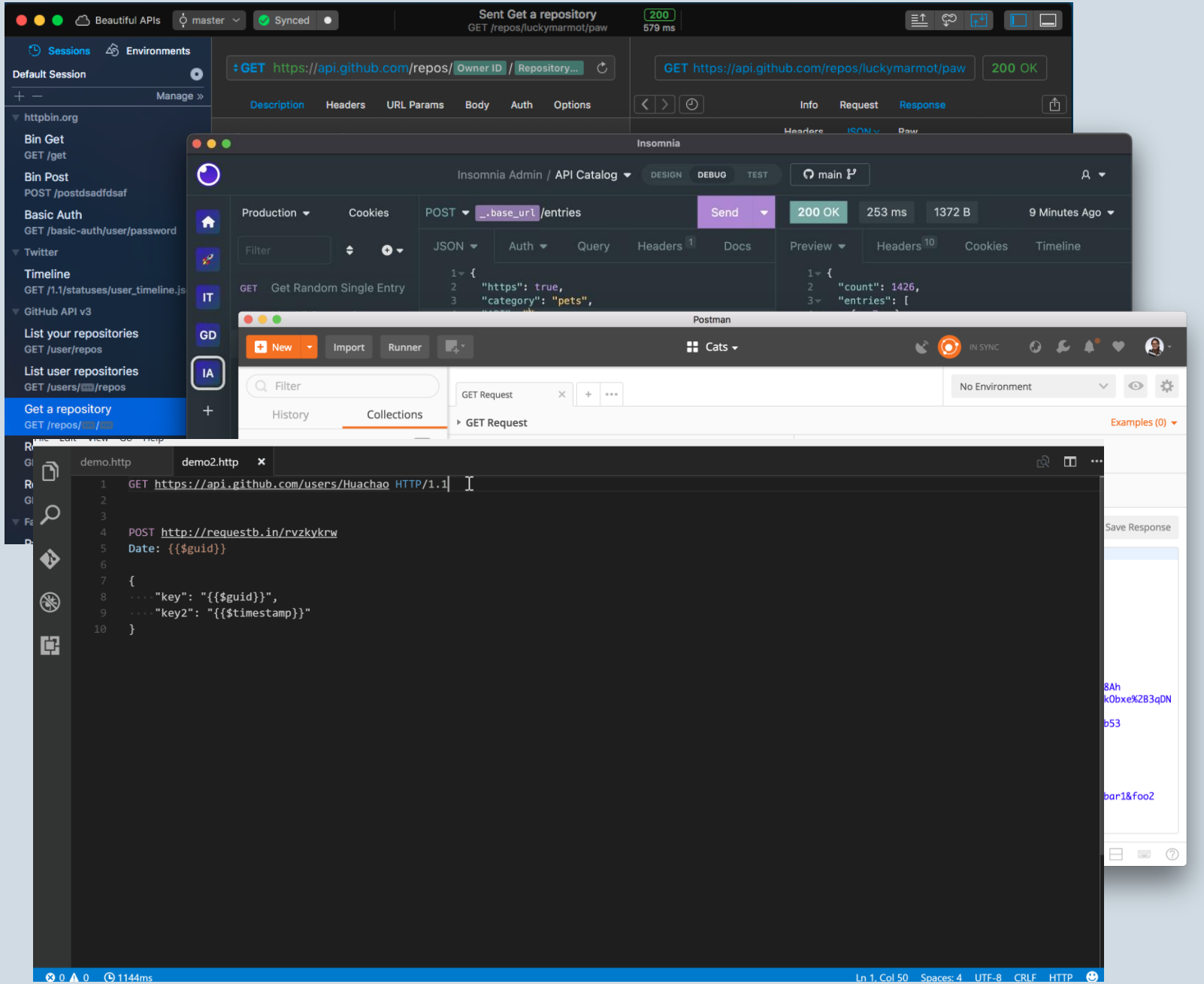
- Configure System variables as User Codes
- Configure your signatures
- Call the BAQ (or Not)
- Manipulate BAQ Output
- Create and Send the REST Request
- Analyse the Response and respond accordingly




```
92        //Get response
93        responseCode = (int)response.StatusCode;
94        responseBody = response.Content;
95
96        if (response.StatusCode == HttpStatusCode.Created)
97        {
98            // Returned JSON from BT is not a jArray so it will not Parse
99            // JArray JSONResponse = JArray.Parse(response.Content);
100
101           //Need to use this method
102           var anonymousType = new { Id = "", Status = "", StatusUrl = "" };
103           var responseObject = JsonConvert.DeserializeAnonymousType(responseBody, anonymousType);
104           btResponseStatus = responseObject.Status;
105       }
106       else
107       {
108           Ice.Diagnostics.Log.WriteEntry($"  Unexpected response status code: {responseCode}");
109       }
110
```

Test Your Function

- Configure your test tool to use the “staging URL” if you are not going to publish.
- Use your favourite REST Testing tool
- Rapid API, Insomnia, Postman, VSCode



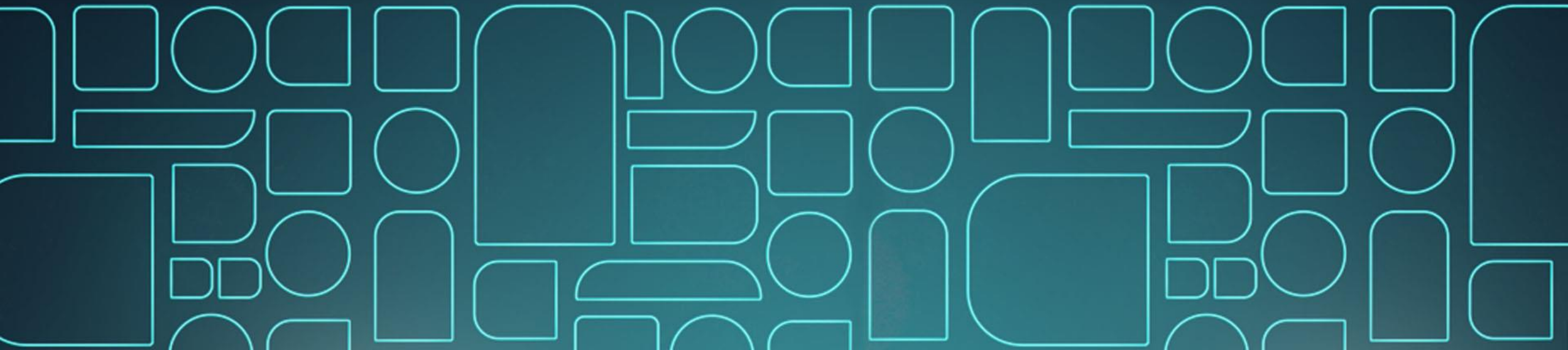
The Takeaway

Integration Webservice REST API = Build an Integration

Bartender REST API = No Integration Needed

EPICOR Insights 2024

Questions?



Resources

- Epicare Bartender Knowledge Articles - KB0041613
- BARTENDER - information on setting up autoprint BPMs needed to print standard ERP Bartender labels - KB0041613
- EpiUsers REST Overview – Novel
- Epicor Help - Search Overview of REST API v.2
- Automating Bartender
- Bartender REST API Overview
- Bartender Print Portal REST API
- Bartender Cloud REST API
- Bartender REST API Actions YAML Reference
- Seagull Software Support Knowledgebase
- Tools
  - Swagger
  - Postman
  - VS Code REST Extension
  - Insomnia
  - Rapid API

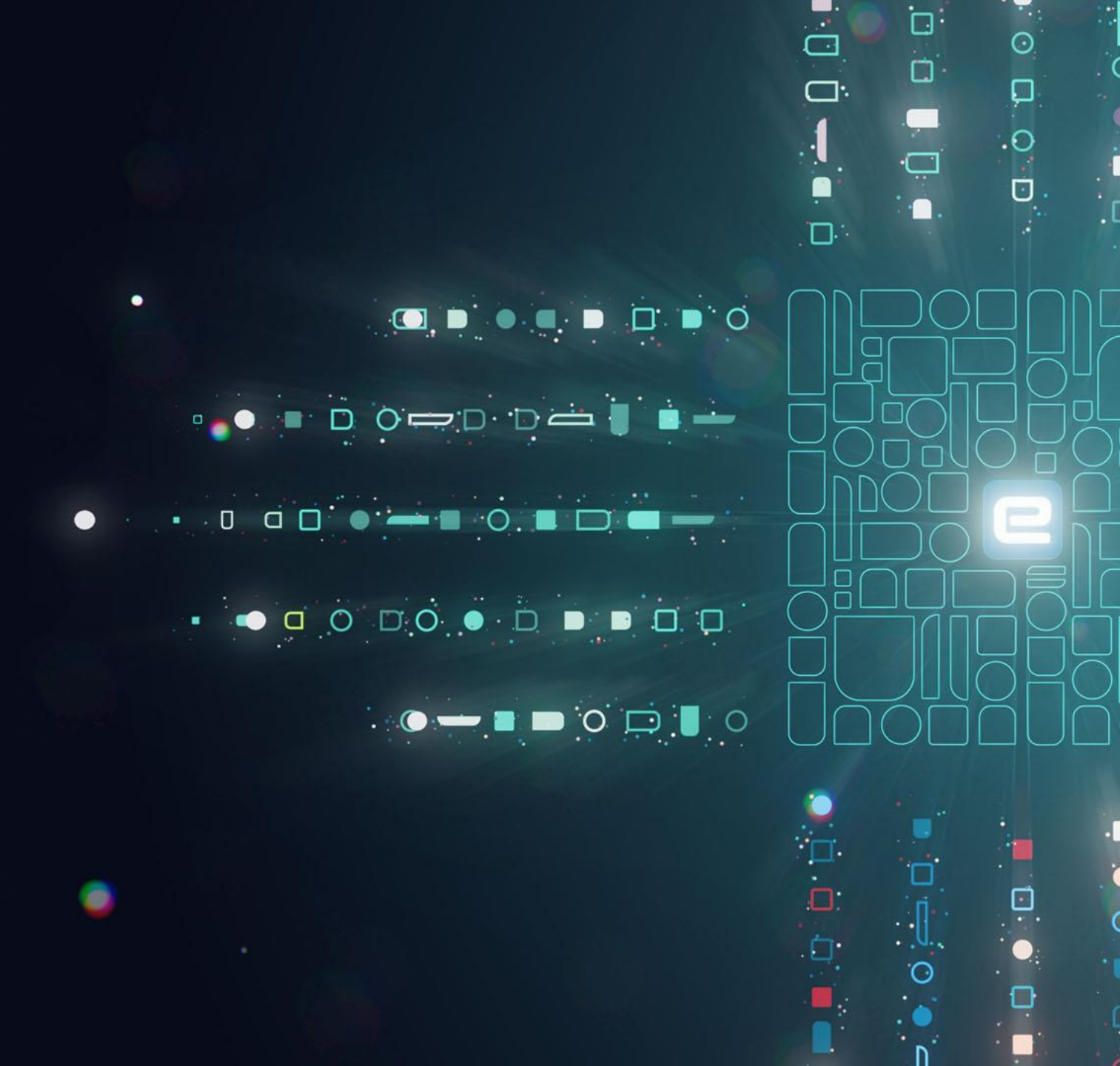
- Creating a Secure Web Integration
- Integraton Troubleshooting Guide

Give Us Feedback

EPICOR

Insights 2024

Thank you!



EPICOR
Insights 2024