



First Look: InterSystems API Manager

Version 2019.4
2019-10-31

First Look: InterSystems API Manager

InterSystems IRIS Data Platform Version 2019.4 2019-10-31

Copyright © 2019 InterSystems Corporation

All rights reserved.



InterSystems, InterSystems Caché, InterSystems Ensemble, InterSystems HealthShare, HealthShare, InterSystems TrakCare, TrakCare, InterSystems DeepSee, and DeepSee are registered trademarks of InterSystems Corporation.



InterSystems IRIS Data Platform, InterSystems IRIS, InterSystems iKnow, Zen, and Caché Server Pages are trademarks of InterSystems Corporation.

All other brand or product names used herein are trademarks or registered trademarks of their respective companies or organizations.

This document contains trade secret and confidential information which is the property of InterSystems Corporation, One Memorial Drive, Cambridge, MA 02142, or its affiliates, and is furnished for the sole purpose of the operation and maintenance of the products of InterSystems Corporation. No part of this publication is to be used for any other purpose, and this publication is not to be reproduced, copied, disclosed, transmitted, stored in a retrieval system or translated into any human or computer language, in any form, by any means, in whole or in part, without the express prior written consent of InterSystems Corporation.

The copying, use and disposition of this document and the software programs described herein is prohibited except to the limited extent set forth in the standard software license agreement(s) of InterSystems Corporation covering such programs and related documentation. InterSystems Corporation makes no representations and warranties concerning such software programs other than those set forth in such standard software license agreement(s). In addition, the liability of InterSystems Corporation for any losses or damages relating to or arising out of the use of such software programs is limited in the manner set forth in such standard software license agreement(s).

THE FOREGOING IS A GENERAL SUMMARY OF THE RESTRICTIONS AND LIMITATIONS IMPOSED BY INTERSYSTEMS CORPORATION ON THE USE OF, AND LIABILITY ARISING FROM, ITS COMPUTER SOFTWARE. FOR COMPLETE INFORMATION REFERENCE SHOULD BE MADE TO THE STANDARD SOFTWARE LICENSE AGREEMENT(S) OF INTERSYSTEMS CORPORATION, COPIES OF WHICH WILL BE MADE AVAILABLE UPON REQUEST.

InterSystems Corporation disclaims responsibility for errors which may appear in this document, and it reserves the right, in its sole discretion and without notice, to make substitutions and modifications in the products and practices described in this document.

For Support questions about any InterSystems products, contact:

InterSystems Worldwide Response Center (WRC)

Tel: +1-617-621-0700

Tel: +44 (0) 844 854 2917

Email: support@InterSystems.com

Table of Contents

First Look: InterSystems API Manager	1
1 How Does InterSystems API Manager Work?	1
2 Try it!	1
2.1 Set up the REST API in InterSystems IRIS	1
2.2 Create a Service in IAM for your REST API	2
2.3 Create a Route in IAM	3
2.4 Call Your API from a REST Client	4
2.5 Add a Rate Limiting Plugin	4
2.6 Add Your REST Specification to IAM	5
3 Learn More About InterSystems API Manager	6

First Look: InterSystems API Manager

This First Look introduces you to InterSystems API Manager (IAM), explains how it works, and gets you started for exploring its capabilities on [your own instance](#).

1 How Does InterSystems API Manager Work?

InterSystems API Manager is a component of InterSystems IRIS that allows you to take advantage of microservices and APIs that are either exposed or consumed by your InterSystems IRIS applications. Acting as an API gateway between your InterSystems IRIS servers and applications, it gives you the ability to more effectively monitor and control the traffic of calls between your server-side APIs and your client-side applications.

To learn more about InterSystems API Manager, you can view [this overview video](#).

2 Try it!

InterSystems API Manager is installed separately from an instance of InterSystems IRIS. To get your own installation kit of IAM, you can [download it](#) from the InterSystems WRC Software Distribution site, or reach out to your InterSystems sales engineer. The installation package includes a script (*iam-setup*) for setting up IAM and connecting it with your instance of InterSystems IRIS.

This exercise assumes that you have instances of InterSystems IRIS and IAM and have run the setup scripts to connect them.

2.1 Set up the REST API in InterSystems IRIS

From [this GitHub repository](#), import the following into InterSystems IRIS:

1. `/cls/cmAPI` includes three class files generated using the API management service as well as a coffeemaker object definition:
 - `impl.cls`
 - `disp.cls`
 - `spec.cls`
 - `coffeemaker.cls`
2. `/gbl/coffeemakers.gof`.

Import by going to the Management Portal and clicking **Globals > Import > My Local Machine**. Find `coffeemakers.gof`. Import to the schema `User.cmAPI.coffeemaker`.

To import, navigate in the InterSystems IRIS management portal to **System Explorer > Classes**, and then click **Import**.

System > Classes

Classes

Compile Export Import Delete Routines Globals

Lookin: Namespace %SYS

System items
 Generated items
 Mapped items

Page size: 0 Results: 1000+ Page: 1 of 1 view doc in new window

Name	Date
<input type="checkbox"/> %Api.Atelier.cls	2019-06-10 01:
<input type="checkbox"/> %Api.Atelier.v1.cls	2019-06-10 01:
<input type="checkbox"/> %Api.Atelier.v2.cls	2019-06-10 01:
<input type="checkbox"/> %Api.Atelier.v3.cls	2019-06-10 01:
<input type="checkbox"/> %Api.DemoSec.cls	2019-06-10 01:

This imports the REST API within InterSystems IRIS and the corresponding classes that will be used for applications to access the set of coffeemakers stored in the database.

Next, set up a Web Application pointing to the REST classes you just imported. Web application layers provide an additional opportunity for customization and security. In the Management Portal, select **System Administration > Security > Applications > Web Applications**. Click **Create New Web Application** and fill out the following settings:

- **Name:** /rest/coffeemakerapp
- **Namespace:** USER
- **Enable Application:** selected
- **Enable:** REST
- **Dispatch Class:** demo.disp
- **Allowed Authentication Methods:** Unauthenticated, Password

2.2 Create a Service in IAM for your REST API

As you saw in the IAM overview video, there are three major components at play within the IAM workflow —Consumers, Routes, and Services. Services in IAM are created for your APIs that exist within InterSystems IRIS. In this case, we will create a service for your Coffeemaker API.

Note: Your IAM package that you downloaded from the WRC also includes a script (iam-test) for setting up a sample service and route. We will be achieving a similar result, using the IAM portal, to add your Coffeemakers service.

test-iris

connect_timeout	60000
created_at	September 11th 2019, 1:37:07pm
host	172.16.8.86
id	c5789d3b-24ca-4a5d-bb89-ae5246304bea
name	test-iris
path	
port	51779
protocol	http
read_timeout	60000
retries	5
updated_at	January 18th 1970, 10:37:14pm
write_timeout	60000

Within IAM, navigate to the default workspace, then **API Gateway > Services**. Click **New Service**. The ensuing form will ask for a number of properties about your service. Fill in the values as follows:

- **name:** CoffeemakerService
- **retries:** 5 [this is the default]
- **connect_timeout:** 60000 [this is the default]
- **write_timeout:** 60000 [this is the default]
- **read_timeout:** 60000 [this is the default]
- **protocol:** http
- **host:** [insert the IP address of your InterSystems IRIS, which may be `localhost`]
- **port:** 52773
- **path:** /rest/coffeemakerapp
- **url:** not required

Once your service is created, you will need to create a route that maps API calls to the appropriate service.

2.3 Create a Route in IAM

When creating a route, you will need to obtain the hexadecimal ID from the CoffeemakerService that you created in the last step. Take note of this id. Then, navigate to **API Gateway > Routes** and click **New Route**. The ensuing form will ask for a number of properties about your route. Fill in the values as follows:

- **protocols:** http
- **methods:** GET, POST, PUT, DELETE
- **hosts:** *Can be left empty*
- **paths:** /rest/coffeemakerapp
- **regex_priority:** 0

Try it!

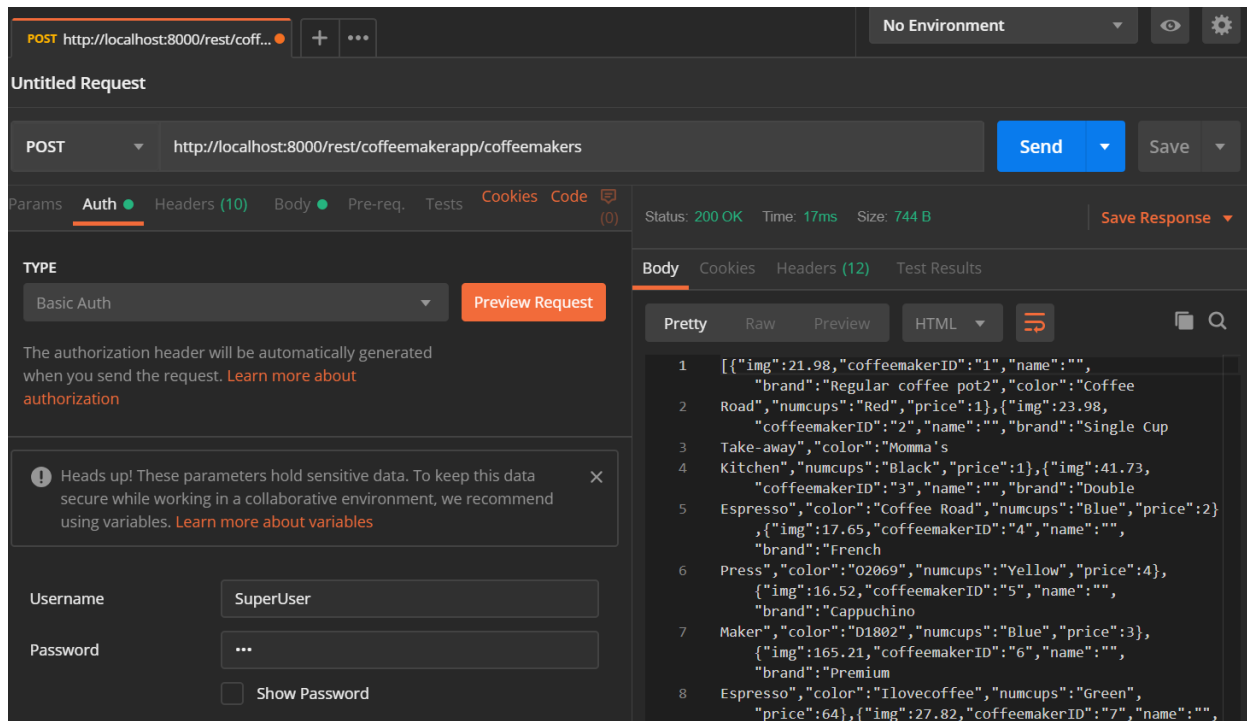
- **strip_path**: Checked [this is the default]
- **preserve_host**: Unchecked [this is the default]
- **service.id**: [copy the hexadecimal id from the CoffeemakerService you created in the last step]

Once your route is created, you will test your route and service by making a simple API call from a REST client.

2.4 Call Your API from a REST Client

Using a REST client — Postman or Advanced REST Client from Google Chrome will work here — make a simple API call to your Coffeemakers API within InterSystems IRIS. To do this, create a request URL that includes the IP address and port number of your IAM instance, as well as the path of your service and the requested endpoint. Be sure to authorize your request with a username and password. An example is shown below; note that by default, IAM handles proxy requests on port 8000. For example:

POST `http://localhost:8000/rest/coffeemakerapp/coffeemakers`

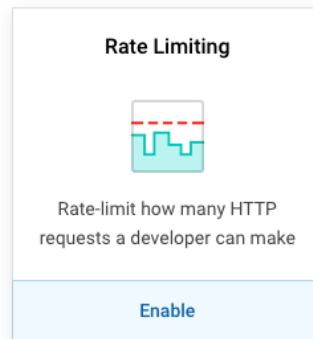


Once you have successfully made this call, you can navigate back to your IAM Dashboard to see the statistics and other information from the call that was made. From this dashboard, you can see a number of different sets of information that will help you with monitoring and controlling traffic between client applications and your services within InterSystems IRIS.

2.5 Add a Rate Limiting Plugin

Effectively monitoring your API traffic is beneficial, but only to the extent that you use that information to appropriately control and optimize your flow of traffic. IAM enables you to use a number of different plugins to boost your ability to control traffic flow between clients and APIs.

Within IAM, navigate to **API Gateway > Plugins** and click **New Plugin**. In the **Traffic Control** category, find the **Rate Limiting** plugin and click **Enable**.



On the configuration screen for this plugin, we will simply enter values into two fields: **route_id** and **config.minute**. For **route_id**, you will need to copy the ID of the route you created. To find this, you will need to look at your routes again (**API Gateway > Routes**).

Create new rate-limiting plugin

api_id

The 'id' of the API that this plugin configuration will target (optional)

service_id

The 'id' of the Service that this plugin configuration will target (optional)

route_id

The 'id' of the Route that this plugin configuration will target (optional)

In the **config.minute** field, enter 5. This will limit the rate of calls accordingly, meaning that clients can only access the API along this specified route a maximum of 5 times per minute. Click **Create** when you have finished entering fields, and your plugin has been enabled.

You can test this rate limiting plugin by returning to your REST client and making sample calls.

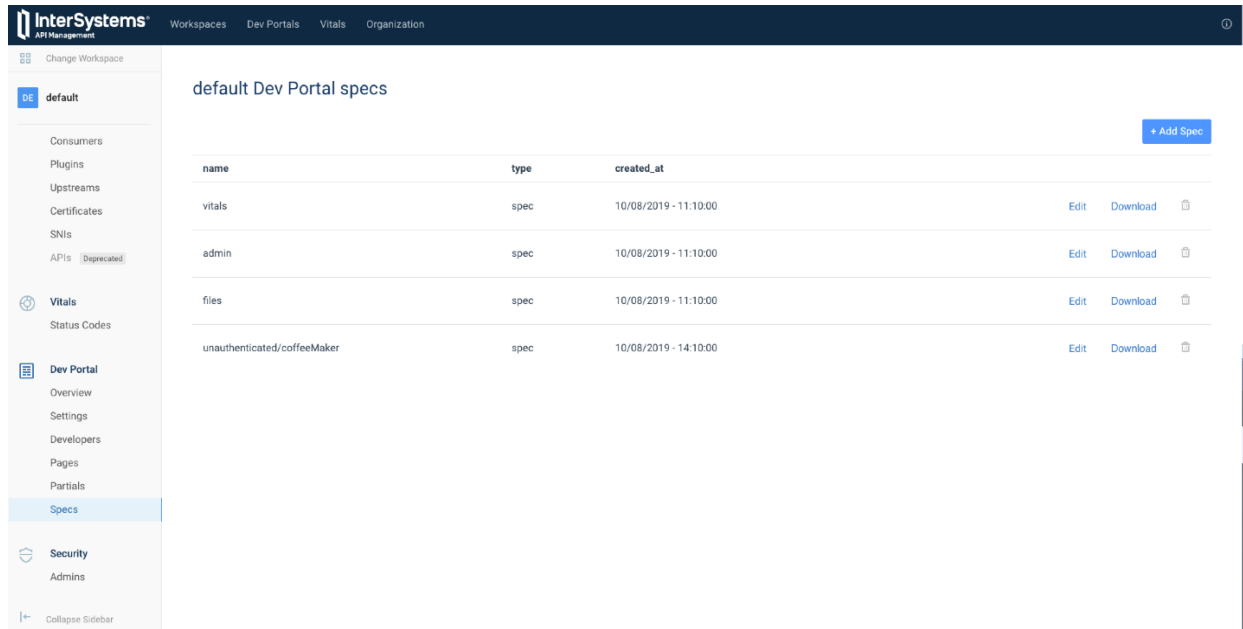
2.6 Add Your REST Specification to IAM

The best practice approach for REST development is specification-first, so it is likely that you already have your REST specification on hand. For this exercise, you can download the JSON specification from [the spec folder in the GitHub repository](#). Find the JSON spec at:

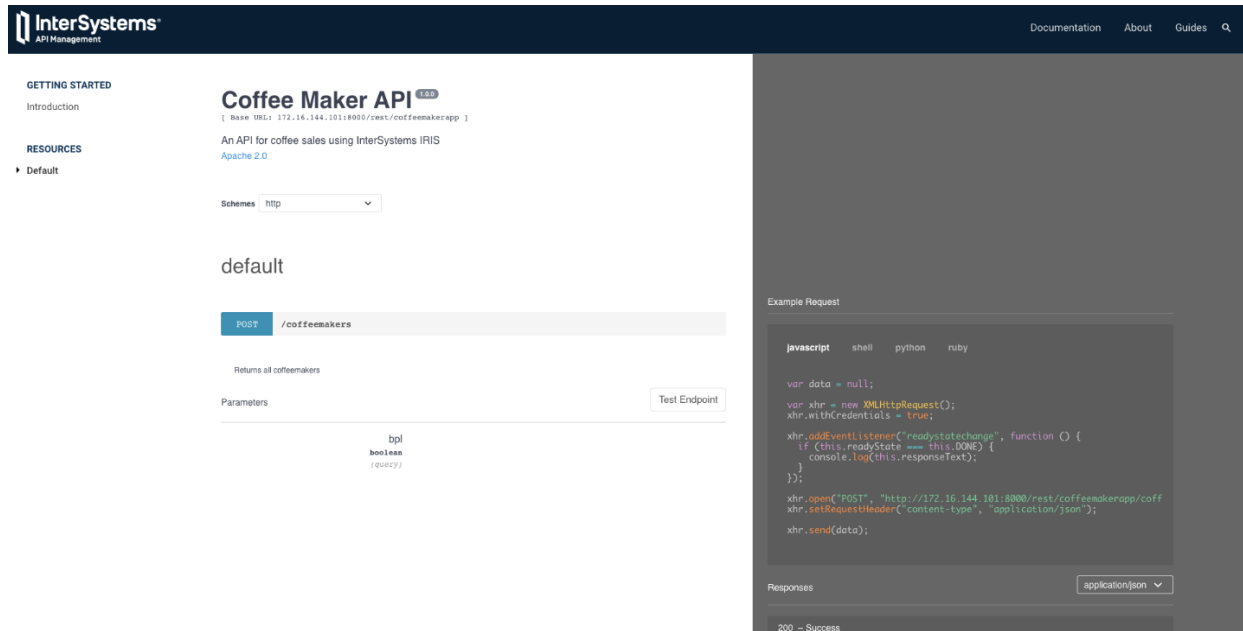
https://github.com/intersystems/FirstLook-IAM/blob/master/spec/swagger_100419.json

Note: For more information on documenting and managing your REST API specifications, read [Discovering and Documenting REST APIs](#).

Once downloaded, you are able to add your specification to IAM. Within IAM, navigate to **Dev Portal > Specs**. Here, you can click the **+Add Spec** button, as shown in the screenshot below.



By creating a name for your spec, and pasting its contents into the editor, you will be able to add the specification to IAM so that developers can easily access it in your developer portal. In your developer portal, you can see all API specs within the **Documentation** tab. Below is an example of what developers will see when viewing an API specification. This example is using the [Swagger Petstore API](#), but once added, your Coffeemaker specification should be accessible in the same way.



3 Learn More About InterSystems API Manager

InterSystems provides several resources to learn more about IAM:

- [API Manager Introduction](#) (presentation)
- [What is InterSystems API Manager?](#) (video)

- [Getting Started with API Manager](#) (documentation)
- [Administration Guide for API Manager](#) (documentation)

