
Question

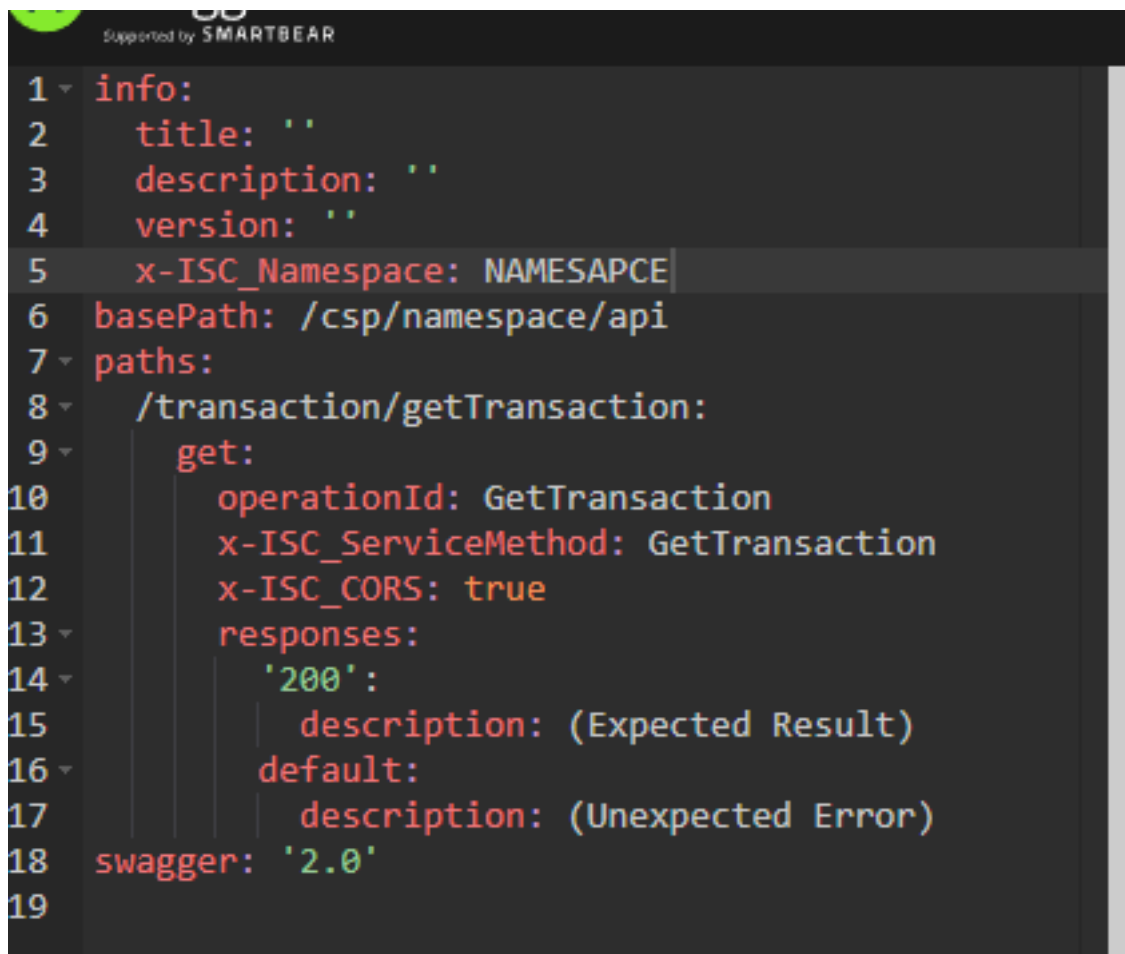
[Michael Davidovich](#) · Jan 31, 2023

Set OpenAPI Properties with GET /api/mgmt/v1/:namespace/spec/:application/

When manually coding REST services and using GET /api/mgmt/v1/:namespace/spec/:application/ to return an OpenAPI spec, how do you specify supported properties ([OpenAPI Properties in Use | Creating REST Services | InterSystems IRIS Data Platform 2021.1](#)) like responses, definitions, and information in paths like summary and description?

[GET /api/mgmt/v1/:namespace/spec/:application/ | Creating REST Services | InterSystems IRIS Data Platform 2022.2](#)

Using a very basic manual REST service that returns a JSON string I get a pretty basic spec that is not really helpful when it comes to documentation. I think the code-first approach would be useful for our purposes if we could make the resulting spec a bit more robust.

A screenshot of a code editor with a dark theme. At the top left, there is a green circular icon and the text "Supported by SMARTBEAR". The code is a JSON snippet representing an OpenAPI specification. It includes an "info" object with "title", "description", "version", and "x-ISC_Namespace". It also has a "basePath" and a "paths" object with a single path "/transaction/getTransaction" that has a "get" method. The "get" method includes "operationId", "x-ISC_ServiceMethod", "x-ISC_CORS", "responses" (with a "200" response), and "swagger" version "2.0".

```
1 info:
2   title: ''
3   description: ''
4   version: ''
5   x-ISC_Namespace: NAMESPACE
6 basePath: /csp/namespace/api
7 paths:
8   /transaction/getTransaction:
9     get:
10      operationId: GetTransaction
11      x-ISC_ServiceMethod: GetTransaction
12      x-ISC_CORS: true
13      responses:
14        '200':
15          description: (Expected Result)
16          default:
17            description: (Unexpected Error)
18 swagger: '2.0'
19
```

For context, I'm exploring the pros and cons of manual creation and spec first creation. Manual is simpler in terms of only needing a dispatch class to implement, but documentation becomes more cumbersome. Spec first creation is great on the documentation side, but when you are coding client-side it makes development process a bit more cumbersome. I have also looked at isc.rest.

[#API](#) [#CSP](#) [#REST API](#) [#InterSystems IRIS](#) [#InterSystems IRIS for Health](#)

Product version: IRIS 2022.1

\$ZV: IRIS for Windows (x86-64) 2022.1.2 (Build 574U) Fri Jan 13 2023 15:00:26 EST

Source

URL: <https://community.intersystems.com/post/set-openapi-properties-get-api/mgmt/v1/namespacespec/application>