Artículo
David Reche · 14 jun, 2019 Lectura de 5 min

La aplicación IRIS API Explorer

InterSystems IRIS 2019.2 introducirá nuevas funciones, muy interesantes. Una de las áreas con novedades que teneis que conocer es la Gestión de APIs.

Para InterSystems IRIS, se ha añadido soporte de una metodología Design-First o API-First. Este método permite realizar primero el diseño de la especificación y después generar el lado del servidor automáticamente a partir de ella.

Antecedentes

La iniciativa OpenAPI (**https://www.openapis.org/**) es una organización que da soporte a la especificación de un estándar para definir y describir APIs (**https://github.com/OAI/OpenAPI-Specification**). La especificación OpenAPI (OAS - Open API Specification) define una descripción de la interfaz (API) estándar e independiente de cualquier lenguaje de programación para crear API REST, la cual permite tanto a personas como a computadores descubrir y comprender las capacidades de un servicio sin la necesidad de acceder al código fuente, a documentos de apoyo adicionales o que inspeccionen el tráfico de la red. Cuando se define de forma apropiada mediante OpenAPI, el consumidor puede entender e interactuar con el servicio remoto, con la mínima cantidad de conocimientos que se necesitan para entender la lógica de la implementación. De forma similar a lo que se hizo en las descripciones para la interfaz de programación de bajo nivel, la especificación de la OpenAPI elimina las suposiciones cuando se llama a un servicio.

Al diseñar la API, por lo general, utilizaremos el Swagger Editor u otra herramienta similar compatible con OAS para crear y obtener la especificación OAS en un formato JSON, cada vez que sea necesario.

Una vez que se dispone de la descripción de la API en JSON OAS y está lista para implementarse, podemos generar la lógica de la API del lado del servidor utilizando la especificación OAS. En InterSystems IRIS 2019.2 podemos usar una nueva rutina **^%REST** para generar de forma automática las clases donde se colocará el código que llamará a la lógica de negocio. Los métodos de estas clases se basan en convenciones de nomenclaturas, aunque se podrá definir el método y la clase en su especificación (mediante la propiedad **operationId** dentro del Swagger).

Ejemplo del uso de la rutina REST de InterSystems IRIS:
USER>do ^%REST

REST Command Line Interface (CLI) helps you CREATE or DELETE a REST application
Enter an application name or (L)ist all REST applications (L): acmeapi
REST application not found: acmeapi
Do you want to create a new REST application? Y or N (Y):

File path or absolute URL of a swagger document.
If no document specified, then create an empty application.
OpenAPI 2.0 swagger: C:\myspec\acme.swagger.json

OpenAPI 2.0 swagger document: C:\myspec\notification.swagger.json
Confirm operation, Y or N (Y):
-----Creating REST application: acmeapi-----
CREATE acmeapi.spec
GENERATE acmeapi.disp
CREATE acmenapi.impl

REST application successfully created.

Create a web application for the REST application? Y or N (Y):
Specify web application name. Default is /csp/api/acme
Web application name: /csp/api/acme/v1

-----Deploying REST application: acmeapi-----
Application acmeapi deployed to /csp/api/acme/v1

En este momento, para crear una API REST, solo puede utilizar la **OpenAPI 2.0** de la especificación Swagger para construir la configuración de la API.

Como se puede observar, mediante esta rutina se crean tres clases:

- <application>**.spec**: esta clase es el contenedor para el swagger spec (bloque XData de OpenAPI). Esta clase es de solo lectura.
- <application>**.disp**: es una clase de Dispatch, la cual está lista para utilizarse en la aplicación CSP. Extiende la clase %CSP.REST y define el UrlMap de XData. Esta clase es de solo lectura y se marca como una clase del sistema (que se oculta de forma predeterminada en Atelier).
- <application>**.impl**: en esta clase se implementan todos los métodos necesarios. Se deberá completar esta clase para que la API realice la funcionalidad para la cual fue diseñada.

¿Qué sucede si ya desarrollé mi API?

En InterSystems IRIS 2018.1, InterSystems introdujo una funcionalidad para el descubrimiento de servicios, la cual permite a los desarrolladores explorar las API REST desplegadas de manera remota. La incorporación de Swagger también nos permite generar una especificación OpenAPI (OAS) desde la aplicación REST que ya existe. Entonces, cualquier API que modifiquemos en InterSystems IRIS puede generar automáticamente la especificación swagger.

Es posible consultar todas las API que están disponibles en el sistema mediante el administrador de la API:

HTTP GET http://<host>:<port>/api/mgmnt/

Devuelve:
HTTP GET http://<host>:<port>/api/mgmnt/

Returns:

```
[
...,
  {
  "name": "/csp/petstore/v2",
  "dispatchClass": "petstore.disp",
  "namespace": "USER",
  "resource": "",
  "swaggerSpec": "/api/mgmnt/v1/USER/spec/csp/petstore/v2",
  "enabled": true
  }
]
```

Además, la especificación Swagger de la API puede recuperarse con el método HTTP GET de la URL que se mostró mediante la propiedad swaggerSpec. Cualquier operación API que se defina por la especificación swagger original tiene una propiedad nueva definida por el nombre del método con el cual se debería implementar la acción:

Por ejemplo:

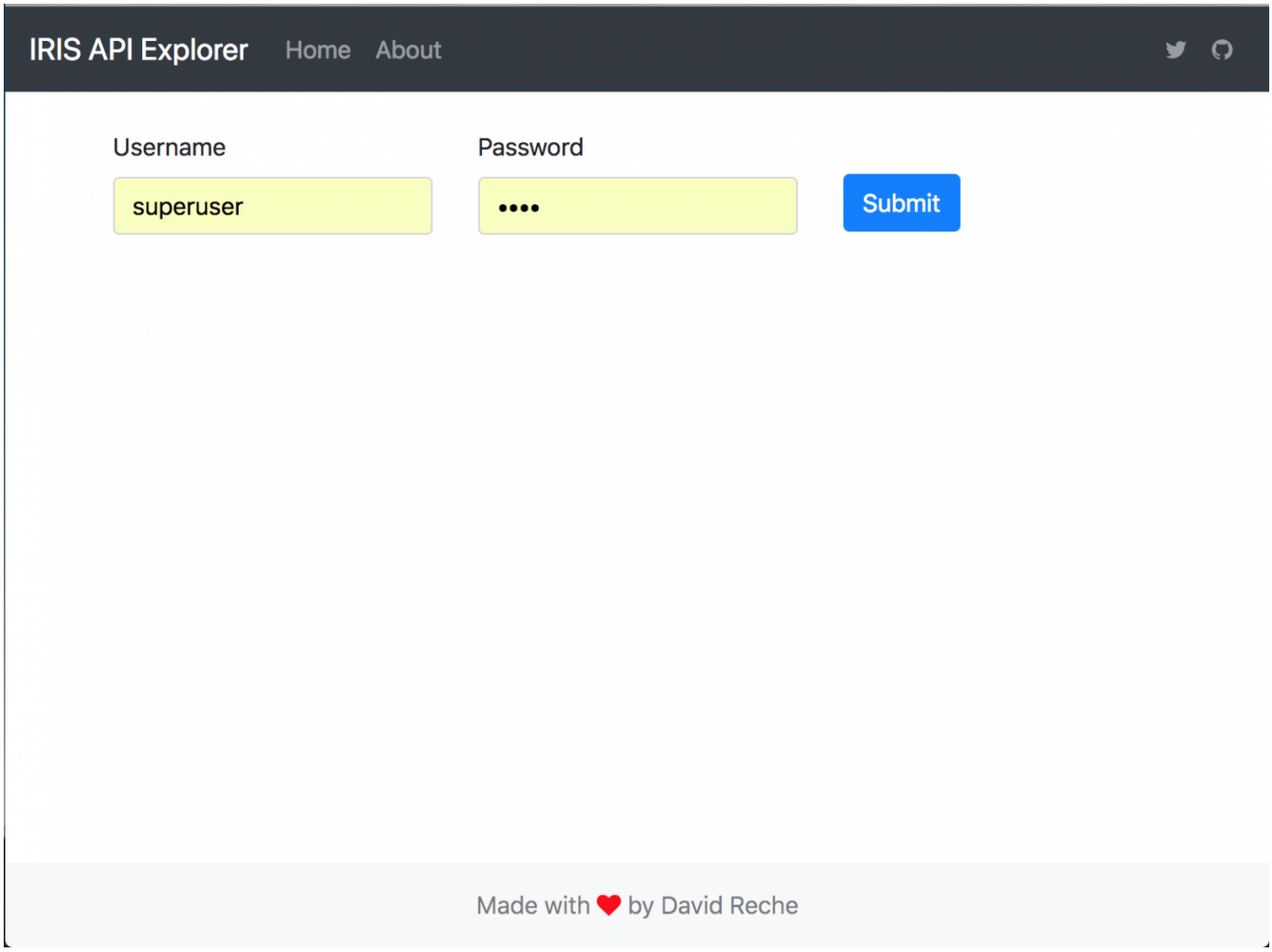
"x-ISCServiceMethod": "getPetById",

Un asunto realmente divertido es que podemos utilizar esta api/mgmnt, no solo para descubrir algo, sino también para crear/consultar/eliminar la API que está en uso
HTTP POST to /api/mgmnt/v2/<namespace>/<applicationName>
HTTP GET to /api/mgmnt/v2/<namespace>/<applicationName>
HTTP DELETE to /api/mgmnt/v2/<namespace>/<applicationName>

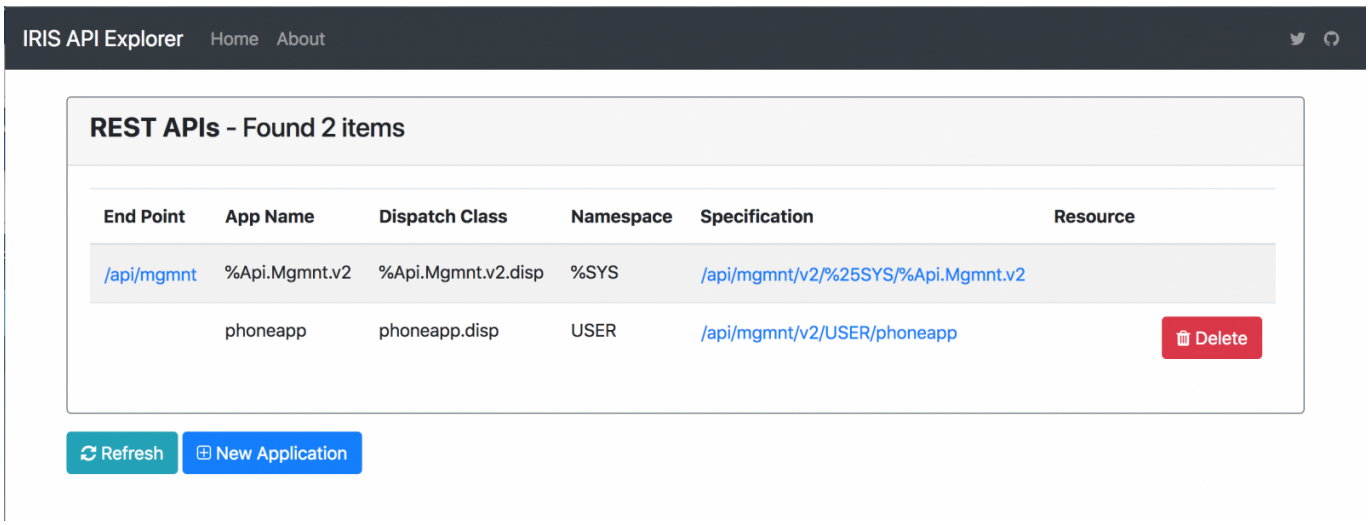
IRIS API Explorer

IRIS Explorer es una aplicación de Angular 5, que aprovecha esta API con la finalidad de proporcionar una herramienta que sea atractiva visualmente para administrar las API en IRIS. Hagamos un recorrido rápido:

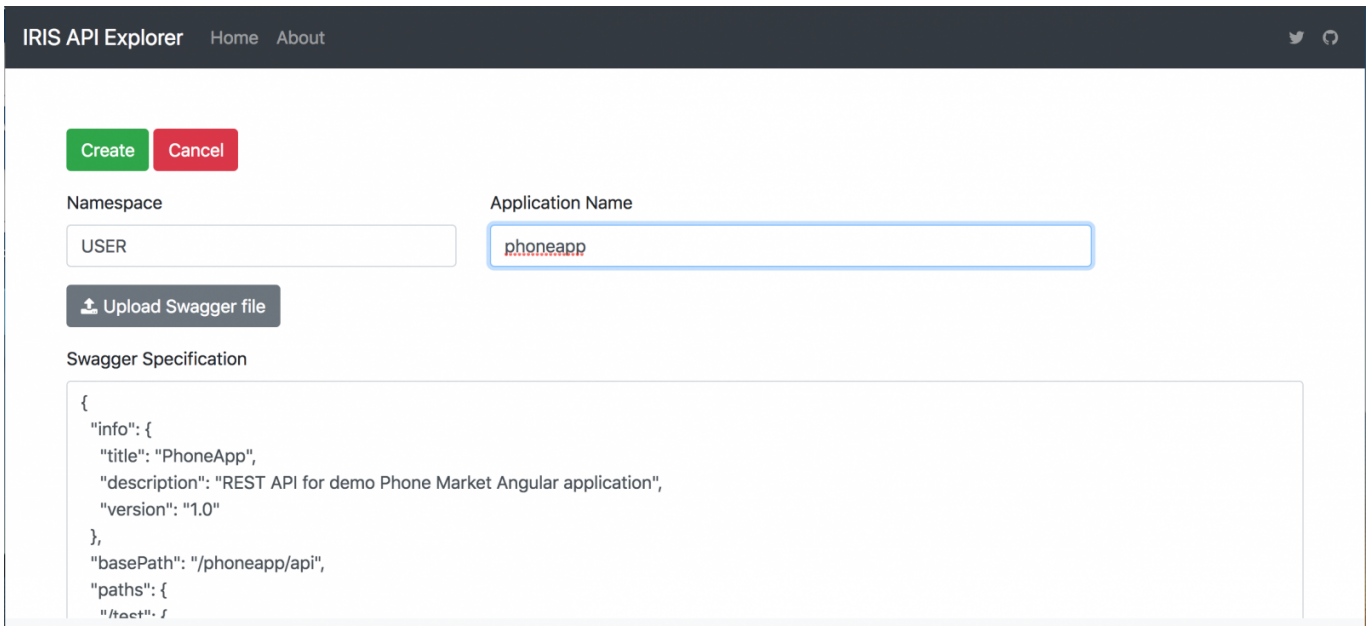
Lo primero que debemos hacer es iniciar sesión en una instancia de IRIS InterSystems (de forma predeterminada busca una instancia local en el puerto 52773):



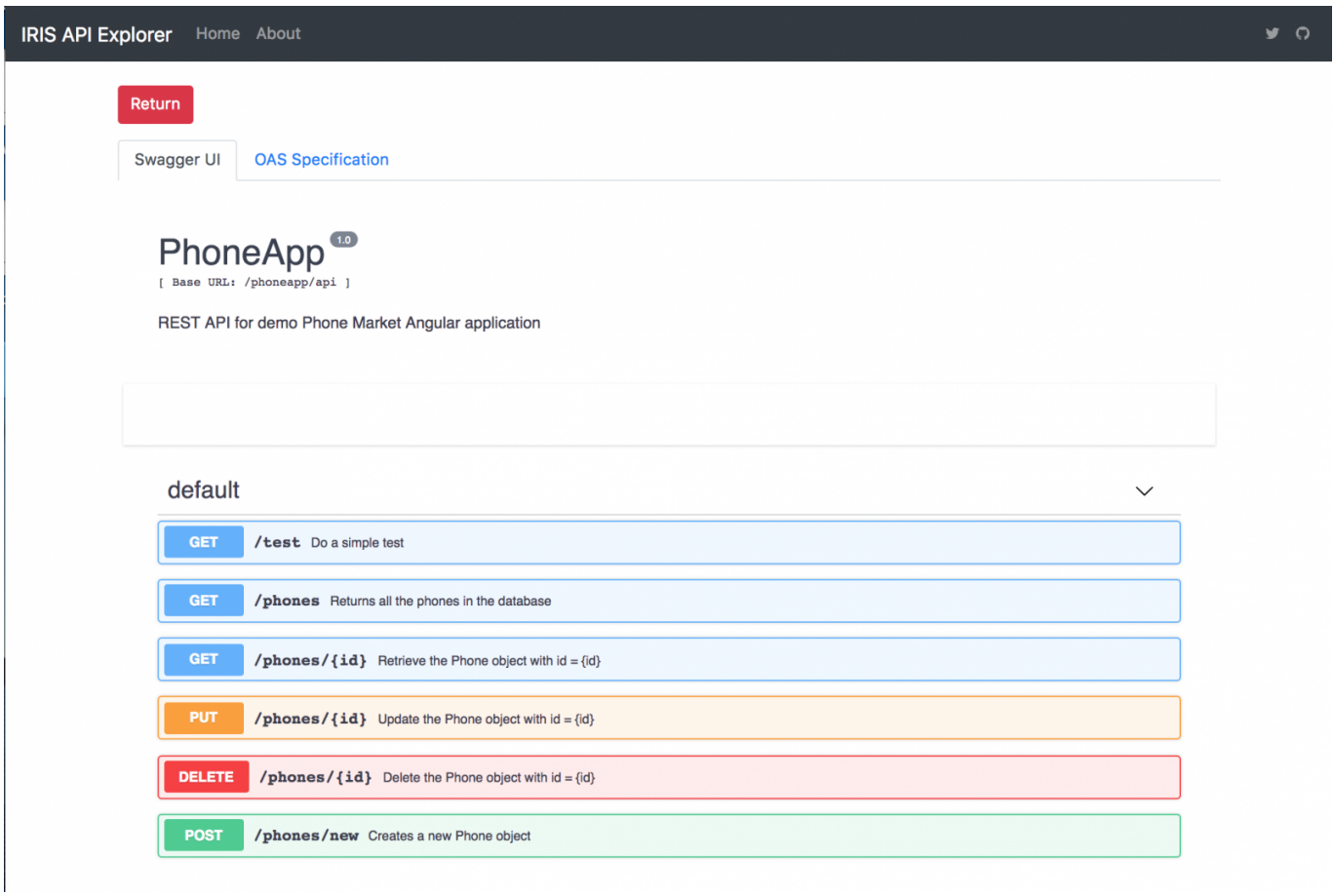
Después de iniciar sesión, la aplicación realiza una consulta para recuperar todas las API que ya existen:



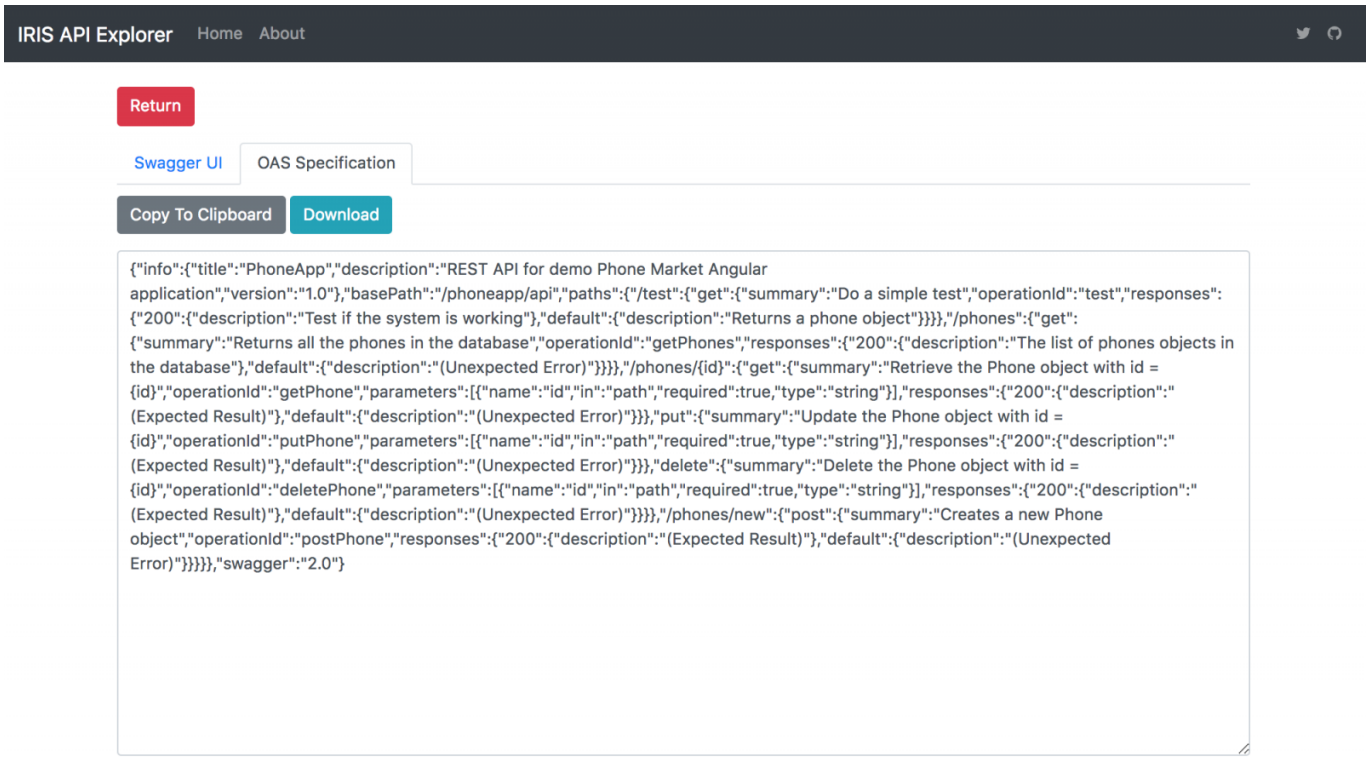
Podemos eliminar una API que ya existe o crear una nueva. Para crear una nueva aplicación necesitamos proporcionar el namespace, el nombre de la aplicación y la especificación Swagger desde un archivo .json:



Una vez que la API está creada podemos ver la especificación. Para hacer esto más divertido incorporé un Swager UI (**https://github.com/swagger-api/swagger-ui**).



Y por supuesto podemos recuperar el JSON de la especificación OAS:



Todo el código es abierto y depende de vosotros lo que useis o modifiqueis según vuestra conveniencia.

La aplicación está disponible en Open Exchange y también en Github.

¡Espero que os sea útil!

#Angular #API #API REST #InterSystems IRIS