
Artículo

[Nancy Martínez](#) · 2 feb, 2021 · Lectura de 3 min

[Open Exchange](#)

InterSystems IRIS y OData

¿Qué es OData?

OData (Open Data Protocol) es un [estándar OASIS](#), con [certificación ISO/IEC](#), que define un conjunto de prácticas recomendadas para construir y consumir APIs RESTful. OData te ayuda a enfocarte en tu lógica de negocio mientras construyes APIs RESTful, sin tener que preocuparte por los diversos enfoques para definir cabeceras de solicitud y respuesta, códigos de estado, métodos HTTP, convenciones URL, tipos de medios audiovisuales, formatos de carga, opciones de consulta, etc. OData también proporciona orientación para registrar cambios, definir funciones/acciones para llevar a cabo procedimientos reutilizables y enviar solicitudes asíncronas/por lotes (Fuente: OData.org).

Casos de uso de OData

- Desplegar datos como servicios REST con un formato interoperable, sin dificultades para desarrollarlos
- Permite que la Inteligencia empresarial (BI), visualización de datos, ERP, CRM, ESB, herramientas y motores de flujo de trabajo consuman datos utilizando REST sin esfuerzos de desarrollo
- Virtualiza datos corporativos en las herramientas de gestión de APIs
- Promueve una forma estándar para implementar las API REST, que permite funcionalidades de consulta similares a las de SQL utilizando estas APIs RESTful. OData esencialmente SQL para entornos webs, desarrollados con protocolos estándar de nivel superior, HTTP, JSON y ATOM, al mismo tiempo que se aprovecha el estilo de la arquitectura REST (progress.com).
- OData es ampliamente adoptado:

Broad Adoption for OData



- Ayuda a implementar FHIR: FHIR (Fast Healthcare Interoperability Resources) es un estándar para el intercambio electrónico de información relacionada con la atención sanitaria. Para que FHIR sea verdaderamente interoperable, [se recomienda que los sistemas utilicen las reglas establecidas por las especificaciones de OData](#) para el parámetro \$search. Además, FHIR también utiliza OAuth para establecer una relación de confianza con el cliente para una capa adicional de seguridad (progress.com).
- Soporta paginación, solicitudes por lotes y diferentes formatos como JSON, ATOM, XML, etc.

OData e InterSystems IRIS

InterSystems IRIS no es compatible con OData, pero es posible utilizar el servidor de OData para que InterSystems IRIS permita mostrar clases persistentes como REST.

Para ello, sigue estas instrucciones:

- Clonar el código fuente del servidor IRIS OData: git clone <https://github.com/yurimarx/isc-iris-odata.git>
- Ir a la carpeta: isc-iris-odata
- Ejecutar: mvnw install (MS Windows) o ./mvnw install (Linux o Mac)
- Ejecutar: docker build -t odata:1.0.0 .
- Ejecutar: docker run -p 8080:8080 odata:1.0.0. Tu servidor OData iniciado:

```

mvnw - docker run -p 8080:8080 odata:1.0.0
by root in /)
2020-06-02 17:08:03.182 INFO 1 --- [main] c.i.iris.odata.IscIrisOdataApplication : No active profile set, falling back to default profiles: default
2020-06-02 17:08:04.951 INFO 1 --- [main] .s.d.r.c.RepositoryConfigurationDelegate : Bootstrapping Spring Data JPA repositories in DEFERRED mode.
2020-06-02 17:08:05.098 INFO 1 --- [main] .s.d.r.c.RepositoryConfigurationDelegate : Finished Spring Data repository scanning in 129ms. Found 1 JPA repository interfaces
2020-06-02 17:08:06.237 INFO 1 --- [main] o.s.b.w.embedded.tomcat.TomcatWebServer : Tomcat initialized with port(s): 8080 (http)
2020-06-02 17:08:06.264 INFO 1 --- [main] o.apache.catalina.core.StandardService : Starting service [Tomcat]
2020-06-02 17:08:06.265 INFO 1 --- [main] org.apache.catalina.core.StandardEngine : Starting Servlet engine: [Apache Tomcat/9.0.35]
2020-06-02 17:08:06.399 INFO 1 --- [main] o.a.c.c.C.[Tomcat].[localhost].[/] : Initializing Spring embedded WebApplicationContext
2020-06-02 17:08:06.399 INFO 1 --- [main] o.s.web.context.ContextLoader : Root WebApplicationContext: initialization completed in 3121 ms
2020-06-02 17:08:06.556 INFO 1 --- [main] com.zaxxer.hikari.HikariDataSource : HikariPool-1 - Starting...
2020-06-02 17:08:07.071 INFO 1 --- [main] com.zaxxer.hikari.HikariDataSource : HikariPool-1 - Start completed.
2020-06-02 17:08:07.089 INFO 1 --- [main] o.s.b.a.h2.H2ConsoleAutoConfiguration : H2 console available at '/h2-console'. Database available at 'jdbc:h2:file:~/odatab
2020-06-02 17:08:07.601 INFO 1 --- [main] o.s.s.concurrent.ThreadPoolTaskExecutor : Initializing ExecutorService 'applicationTaskExecutor'
2020-06-02 17:08:07.801 INFO 1 --- [task-1] o.hibernate.jpa.internal.util.LogHelper : HH0000204: Processing PersistenceUnitInfo [name: default]
2020-06-02 17:08:07.809 WARN 1 --- [main] JpaBaseConfiguration$JpaWebConfiguration : spring.jpa.open-in-view is enabled by default. Therefore, database queries may be pe
rformed during view rendering. Explicitly configure spring.jpa.open-in-view to disable this warning
2020-06-02 17:08:08.369 INFO 1 --- [task-1] org.hibernate.Version : HH0000412: Hibernate ORM core version 5.4.15.Final
2020-06-02 17:08:08.669 INFO 1 --- [task-1] o.hibernate.annotations.common.Version : HCANN000001: Hibernate Commons Annotations {5.1.0.Final}
2020-06-02 17:08:09.129 INFO 1 --- [task-1] org.hibernate.dialect.Dialect : HH0000400: Using dialect: org.hibernate.dialect.H2Dialect
2020-06-02 17:08:10.879 INFO 1 --- [task-1] o.h.e.t.j.p.i.JtaPlatformInitiator : HH0000490: Using JtaPlatform implementation: [org.hibernate.engine.transaction.jta.p
latform.internal.NoJtaPlatform]
2020-06-02 17:08:10.904 INFO 1 --- [task-1] j.LocalContainerEntityManagerFactoryBean : Initialized JPA EntityManagerFactory for persistence unit 'default'
2020-06-02 17:08:11.549 INFO 1 --- [main] o.s.b.a.w.s.WelcomePageHandlerMapping : Adding welcome page: class path resource [static/index.html]
2020-06-02 17:08:11.764 INFO 1 --- [main] o.s.b.w.embedded.tomcat.TomcatWebServer : Tomcat started on port(s): 8080 (http) with context path ''
2020-06-02 17:08:11.766 INFO 1 --- [main] DeferredRepositoryInitializationListener : Triggering deferred initialization of Spring Data repositories...
2020-06-02 17:08:11.890 INFO 1 --- [main] DeferredRepositoryInitializationListener : Spring Data repositories initialized!
2020-06-02 17:08:11.909 INFO 1 --- [main] c.i.iris.odata.IscIrisOdataApplication : Started IscIrisOdataApplication in 9.805 seconds (JVM running for 11.031)

```

8. Inicial tu instancia de InterSystems IRIS con cualquier clase persistente

7. En navegador de acceso: <http://localhost:8080/>. Establecer los parámetros en la pantalla:

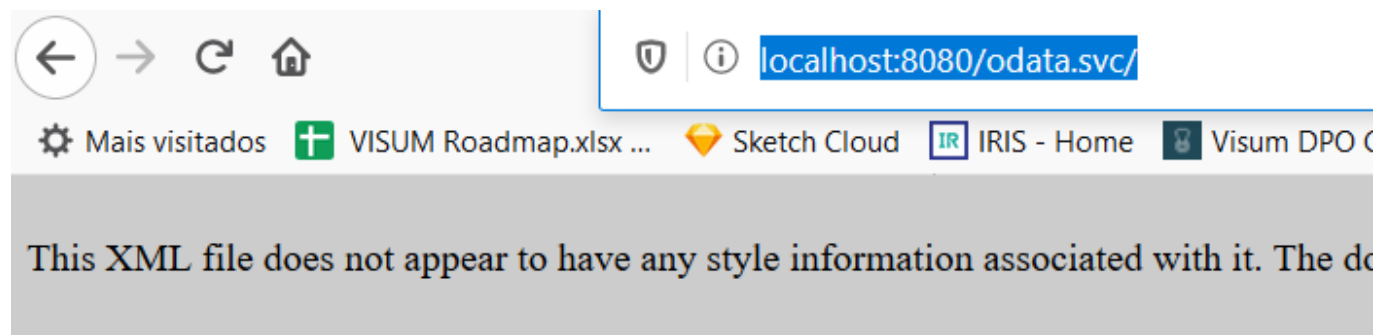
OData Server for InterSystems IRIS - Set Parameters

Host for IRIS Instance 192.168.56.1				
Host Port 9091	Namespace Contest	Schema Contest_Data	Username _SYSTEM	Password welcome1
SUBMIT				

8. Estos son los parámetros para mi instancia. Establece los parámetros en tu instancia de IRIS. En el namespace establece tu namespace de iris; en el esquema, el esquema de la tabla SQL; y en el puerto, el puerto a tu conexión con la base de datos JDBC.

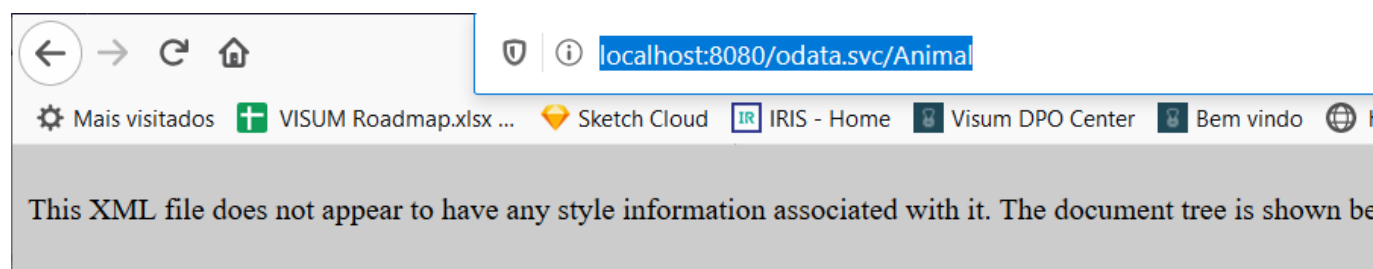
9. Hacer clic en Enviar y RECARGAR TU INSTANCIA DE DOCKER DE ODATA SERVER PARA APLICAR LOS PARÁMETROS.

10. Ir a: <http://localhost:8080/odata.svc/> para ver todas las clases persistentes en tu esquema de IRIS. En mi caso es:



```
—<app:service metadata:context="$metadata">
  —<app:workspace>
    <atom:title>Contest_Data.Contest_Data</atom:title>
    —<app:collection href="Animal" metadata:name="Animal">
      <atom:title>Animal</atom:title>
    </app:collection>
    —<app:collection href="Product" metadata:name="Product">
      <atom:title>Product</atom:title>
    </app:collection>
  </app:workspace>
</app:service>
```

11. Para navegar hacia una clase persistente, usar:
<http://localhost:8080/odata.svc/<PersistentClass>>. Por ejemplo:
<http://localhost:8080/odata.svc/Animal>
12. El servidor OData hace una lista con los datos Animales, mira:

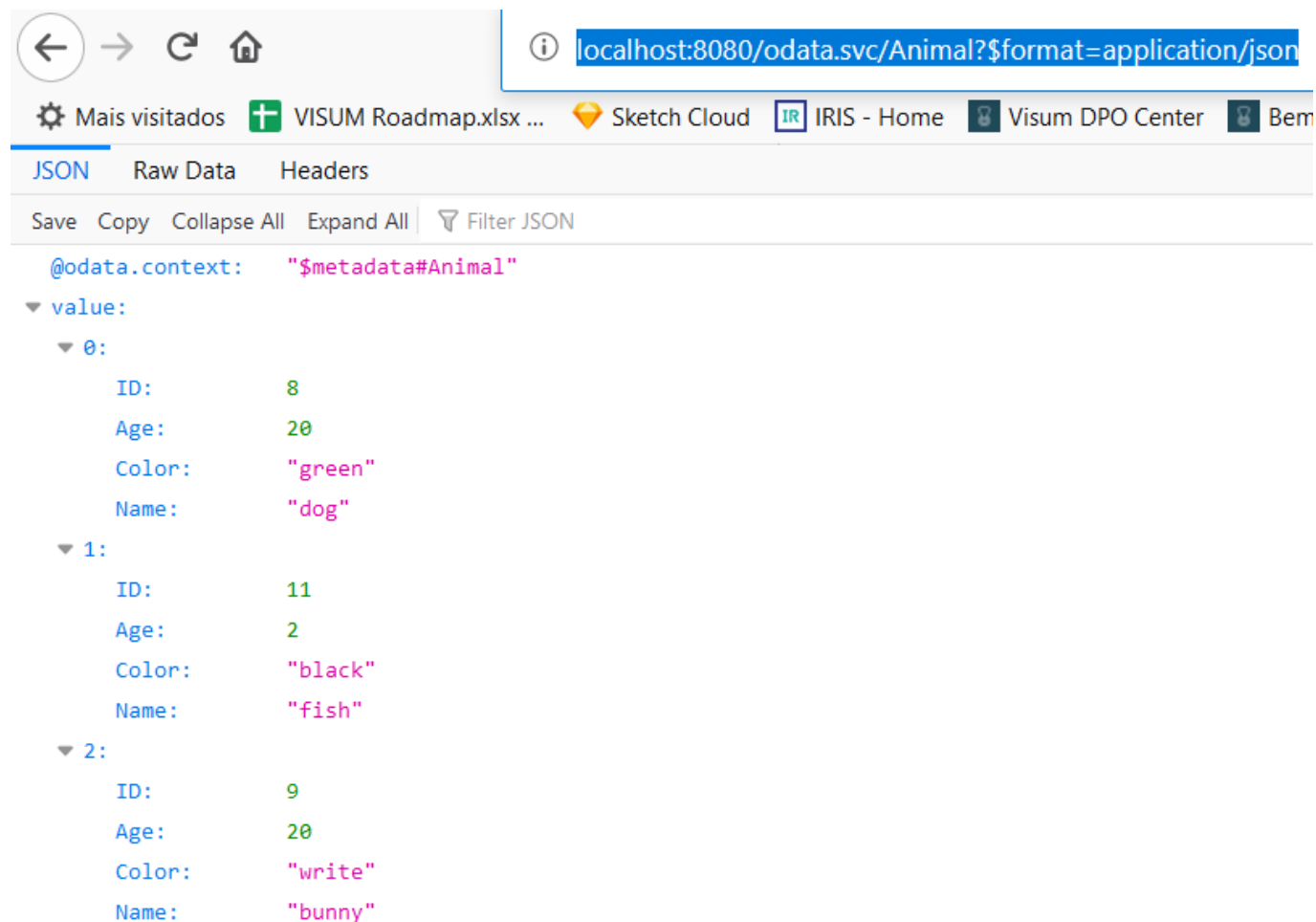


```

<a:feed m:context="$metadata#Animal">
  <a:id>http://localhost:8080/odata.svc/Animal</a:id>
  <a:entry>
    <a:id>Animal(9)</a:id>
    <a:title/>
    <a:summary/>
    <a:updated>2020-06-02T17:16:44Z</a:updated>
    <a:author>
      <a:name/>
    </a:author>
    <a:link rel="edit" href="Animal(9)"/>
    <a:category scheme="http://docs.oasis-open.org/odata/ns/scheme" term="#Contest_Data.Animal"/>
    <a:content type="application/xml">
      <m:properties>
        <d:ID m:type="Int32">9</d:ID>
        <d:Age m:type="Int32">20</d:Age>
        <d:Color>write</d:Color>
        <d:Name>bunny</d:Name>
      </m:properties>
    </a:content>
  </a:entry>
</a:entry>

```

13. Para verlo en formato JSON: [http://localhost:8080/odata.svc/Animal?\\$format=application/json](http://localhost:8080/odata.svc/Animal?$format=application/json).
Mira:



The screenshot shows a web browser window with the address bar displaying `localhost:8080/odata.svc/Animal?$format=application/json`. The browser's developer tools are open, showing the JSON response. The response is a JSON array with three objects, each representing an animal with properties ID, Age, Color, and Name.

```
@odata.context: "$metadata#Animal"
value:
  0:
    ID: 8
    Age: 20
    Color: "green"
    Name: "dog"
  1:
    ID: 11
    Age: 2
    Color: "black"
    Name: "fish"
  2:
    ID: 9
    Age: 20
    Color: "white"
    Name: "bunny"
```

14. Para ver los detalles de una fila:

[http://localhost:8080/odata.svc/Animal\(8\)?\\$format=application/json](http://localhost:8080/odata.svc/Animal(8)?$format=application/json)

15. Para eliminar, envía un DELETE en tu postman con [http://localhost:8080/odata.svc/Animal\(8\)](http://localhost:8080/odata.svc/Animal(8))

16. Para insertar, envía un POST en tu postman con <http://localhost:8080/odata.svc/Animal> y una estructura JSON con los nombres de las propiedades y pares de valores, como:

The screenshot shows a REST client interface. At the top, the method is 'POST' and the URL is 'http://localhost:8080/odata.svc/Animal'. Below this, there are tabs for 'Params', 'Authorization', 'Headers (9)', 'Body', 'Pre-request Script', 'Tests', and 'Settings'. The 'Body' tab is selected, and the content type is set to 'JSON'. The body contains the following JSON:

```
1 {  
2   "Age": 2,  
3   "Color": "black",  
4   "Name": "fish"  
5 }
```

Below the body tab, there are tabs for 'Body', 'Cookies', 'Headers (9)', and 'Test Results'. The 'Body' tab is selected, and the content is displayed in 'JSON' format. The body contains the following JSON:

```
1 {  
2   "@odata.context": "$metadata#Animal",  
3   "ID": null,  
4   "Age": 2,  
5   "Color": "black",  
6   "Name": "fish"
```

17. Así que puedes realizar todas las operaciones CRUD con tus clases persistentes.

18. Otras muchas funcionalidades estarán disponibles en el futuro, si IRIS OData Server obtiene la aceptación de la comunidad.

¡Gracias!

[#Concurso](#) [#InterSystems IRIS](#) [#InterSystems IRIS for Health](#) [#Open Exchange](#)
[Ir a la aplicación en InterSystems Open Exchange](#)

URL de fuente: <https://es.community.intersystems.com/post/intersystems-iris-y-odata>