

Article

[Yuri Marx](#) · Jun 2, 2020 3m read

[Open Exchange](#)

OData and InterSystems IRIS

What is the OData

OData (Open Data Protocol) is an [ISO/IEC approved](#), [OASIS standard](#) that defines a set of best practices for building and consuming RESTful APIs. OData helps you focus on your business logic while building RESTful APIs without having to worry about the various approaches to define request and response headers, status codes, HTTP methods, URL conventions, media types, payload formats, query options, etc. OData also provides guidance for tracking changes, defining functions/actions for reusable procedures, and sending asynchronous/batch requests (source: OData.org).

The OData use cases

- Deploy data as REST Services with an interoperable format without development effort;
- Allows BI, data visualization, ERP, CRM, ESB, Workflow tools and engines consume data using REST without development effort;
- Virtualize corporate data in API Management tools;
- OData advocates a standard way of implementing REST APIs that allows for SQL-like querying capabilities using these RESTful APIs. OData is essentially SQL for the web built n top of standard protocols – HTTP, JSON & ATOM – while leveraging the REST architecture style (progress.com);
- OData has a broad adoption, see:

Broad Adoption for OData



- OData helps to implement FHIR: FHIR, or Fast Healthcare Interoperability Resources Specification, is a standard for exchanging healthcare information electronically. In order to make FHIR truly interoperable, it is [recommended that systems use the rules specified by OData](#)

[specification](#) for the \$search parameter. Further, FHIR also uses OAuth in order to establish a trusted relationship with the client for an extra layer of security (progress.com);

- OData supports pagination, batch requests and different formats like JSON, ATOM, XML, etc.

OData and InterSystems IRIS

The InterSystems IRIS does not support OData but is possible use the OData Server for InterSystems IRIS to allows expose Persistent classes as REST.

Follow these instructions:

1. Clone the source code of the IRIS OData Server: git clone <https://github.com/yurimarx/isc-iris-odata.git>
2. Go to: isc-iris-odata folder
3. Execute: mvnw install (MS Windows) or ./mvnw install (linux or mac)
4. Execute: docker build -t odata:1.0.0 .
5. Execute: docker run -p 8080:8080 odata:1.0.0. Your OData Server started:

```

mwnw - 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:~/odatadb'
.
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 : HHH000204: 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 performed 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 : HHH000412: 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 : HHH000400: 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 : HHH000490: Using JtaPlatform implementation: [org.hibernate.engine.transaction.jta.platform.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 17:08:11.898 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)

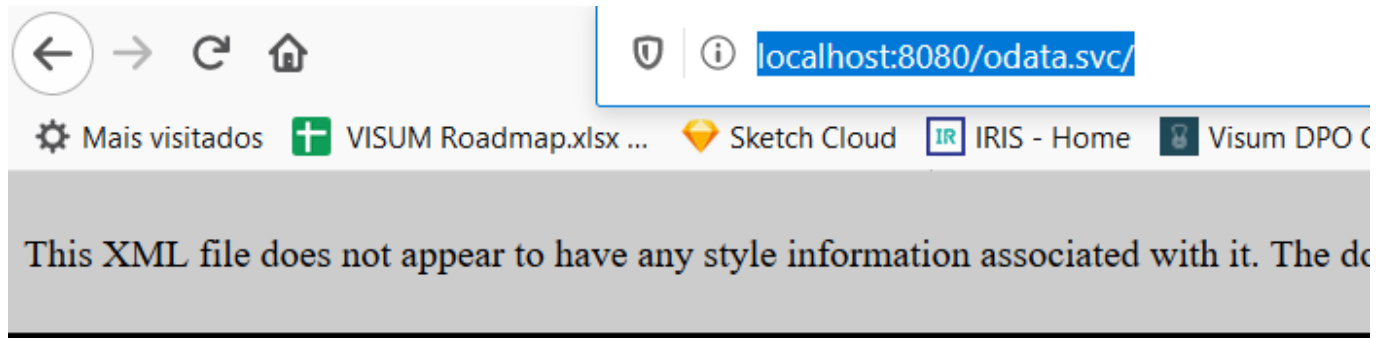
```

6. Start your InterSystems IRIS instance with any persistent class, in my case
7. In your browser access: <http://localhost:8080/>. Set parameters on the screen:

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. This is parameters to my instance. Set the parameters to your IRIS instance. In namespace set your iris namespace, in the schema, the SQL Table schema, and in the port, the port to your JDBC database connection.
9. Press submit and RELOAD YOUR ODATA SERVER DOCKER INSTANCE TO APPLY PARAMETERS.
10. Access <http://localhost:8080/odata.svc/> to see all persistent classes to your IRIS schema. In my case is:



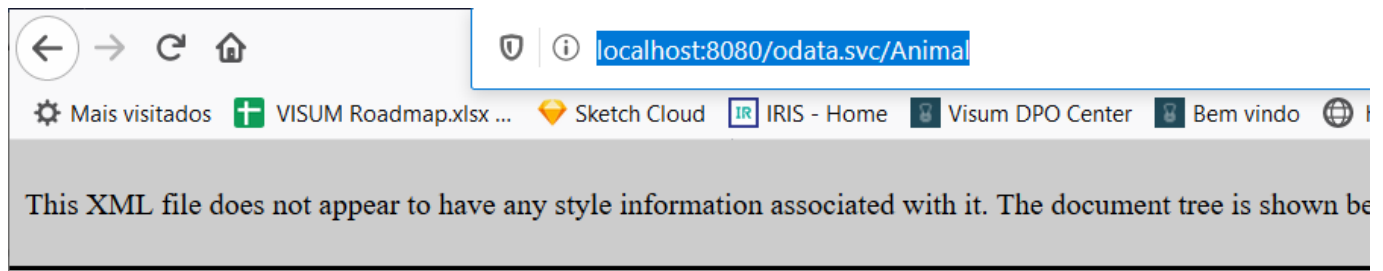
```

-<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. To navigate to a persistent class browse: <http://localhost:8080/odata.svc/<PersistentClass>> e.g.: <http://localhost:8080/odata.svc/Animal>

12. The OData server list Animal data, see:



This XML file does not appear to have any style information associated with it. The document tree is shown below.

```

<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. To see in JSON format browse: [http://localhost:8080/odata.svc/Animal?\\$format=application/json](http://localhost:8080/odata.svc/Animal?$format=application/json).
See:

```

@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: "write"
    Name: "bunny"
    
```

14. To see details about a row browse: [http://localhost:8080/odata.svc/Animal\(8\)?\\$format=application/json](http://localhost:8080/odata.svc/Animal(8)?$format=application/json)
15. To delete send a DELETE in your postman with [http://localhost:8080/odata.svc/Animal\(8\)](http://localhost:8080/odata.svc/Animal(8))
16. To insert send a POST in your postman with <http://localhost:8080/odata.svc/Animal> and a JSON body with property name and value pairs, like:

The screenshot shows a REST client interface. At the top, the method is set to POST and the URL is http://localhost:8080/odata.svc/Animal. The 'Body' tab is selected, showing a JSON payload:

```
{
  "Age": 2,
  "Color": "black",
  "Name": "fish"
}
```

 Below this, the 'Body' tab of the response is shown, displaying a pretty-printed JSON object:

```
{
  "@odata.context": "$metadata#Animal",
  "ID": null,
  "Age": 2,
  "Color": "black",
  "Name": "fish"
}
```

17. So you can do all CRUD operations with your persistent classes.

18. Many other features will be released in the future, if IRIS OData Server get the community adoption.

Thanks!

[#Contest](#) [#InterSystems IRIS](#) [#InterSystems IRIS for Health](#) [#Open Exchange](#)
[Check the related application on InterSystems Open Exchange](#)

Source URL: <https://community.intersystems.com/post/odata-and-intersystems-iris>