

IDRA REST SERVER API

REST Data Model Files and Libraries

 Home

REST Resources

This API supports a [Representational State Transfer \(REST\)](#) model for accessing a set of resources through a fixed set of operations. The following resources are accessible through the RESTful model:

- [CategoryController](#)
- [MaterialController](#)
- [SchedaController](#)
- [TypeObjectController](#)

The REST resources expose a data model that is supported by a set of client-side libraries that are made available on the [files and libraries](#) page.

There is also a [WADL document](#) describing the REST API.

Data Model

All endpoints act on a common set of data. The data can be represented with difference media (i.e. "MIME") types, depending on the endpoint that consumes and/or produces the data. The data can described by [XML Schema](#), which definitively describes the XML representation of the data, but is also useful for describing the other formats of the data, such as [JSON](#).

This document will describe the data using terms based on [XML Schema](#). Data can be grouped by namespace, with a schema document describing the elements and types of the namespace. Generally speaking, types define the structure of the data and elements are instances of a type. For example, elements are usually produced by (or consumed by) a REST endpoint, and the structure of each element is described by its type.

Namespace "ns0"

Namespace URI:	http://www.idra.info/rest
XSD:	idra.xsd

Data Elements

- [age](#)
- [artifact](#)
- [dimension](#)
- [inventory](#)
- [item](#)
- [location](#)
- [museum](#)
- [picture](#)

Data Types

- [age](#)
 - [artifact](#)
 - [dimension](#)
 - [inventory](#)
 - [itemBibliography](#)
 - [location](#)
 - [museum](#)
 - [picture](#)
-

Copyright © 2013 Andromeda

Generated by [Enunciate](#).

CATEGORY CONTROLLER

REST Data Model Files and Libraries

 Home > REST > CategoryController

CategoryController

The following resources are part of this group:

- [/category/all](#)

/category/all

Mount Point: [/category/all](#)

GET

The method will return the list of object's categories in the local storage.

Response Body

element:	(custom)
media types:	application/xml application/json

List of category

Copyright © 2013 2013 Andromeda

Generated by [Enunciate](#).

MATERIAL CONTROLLER

REST Data Model Files and Libraries

 Home > REST > MaterialController

MaterialController

The following resources are part of this group:

- [/material/all](#)

/material/all

Mount Point: [/material/all](#)

GET

The method will return the list of object's materials in the local storage.

Response Body

element:	(custom)
media types:	application/xml application/json

List of materials

Copyright © 2013 2013 Andromeda

Generated by [Enunciate](#).

SchedaController

The following resources are part of this group:

- [/artifact/all](#)
- [/artifact/{nctn}](#)

/artifact/all

Mount Point: [/artifact/all](#)

GET

The method will return the list of artifacts in the local storage.

Response Body

element:	(custom)
media types:	application/xml application/json

List of artifacts

/artifact/{nctn}

Mount Point: [/artifact/{nctn}](#)

GET

The method will return the single artifact in the local storage.

Parameters

name	description	type	default
nctn	number of the object	path	

Response Body

element:	artifact
media types:	application/xml application/json

Artifact single object

Copyright © 2013 2013 Andromeda

Generated by [Enunciate](#).

TYPEOBJECTCONTROLLER

REST Data Model Files and Libraries

 Home > REST > TypeObjectController

TypeObjectController

The following resources are part of this group:

- [/type/all](#)

/type/all

Mount Point: [/type/all](#)

GET

The method will return the list of object's typologies in the local storage.

Response Body

element:	(custom)
media types:	application/xml application/json

List of category

Copyright © 2013 2013 Andromeda

Generated by [Enunciate](#).

```

<?xml version="1.0" encoding="UTF-8" ?>
- <wadl:application xmlns:wadl="http://wadl.dev.java.net/2009/02" xmlns:xs="http://www.w3.org/2001/XMLSchema">
  <wadl:doc xmlns:enunciate="http://enunciate.codehaus.org/" enunciate:generatedBy="Enunciate-1.26.1" />
- <wadl:grammars>
  <wadl:include href="idra.xsd" />
</wadl:grammars>
- <wadl:resources base="http://api.idra.info">
  - <wadl:resource path="/artifact/all">
    - <wadl:method name="GET">
      - <wadl:doc>
        - <![CDATA[
          | The method will return the list of artifacts in the local
          | storage.
        ]]>
      </wadl:doc>
    - <wadl:response>
      - <wadl:doc>
        <![CDATA[ List of artifacts ]]>
      </wadl:doc>
      <wadl:representation mediaType="application/xml" />
      <wadl:representation mediaType="application/json" />
    </wadl:response>
  </wadl:method>
</wadl:resource>
- <wadl:resource path="/artifact/{nctn}">
  - <wadl:param name="nctn" style="template">
    - <wadl:doc>
      <![CDATA[ number of the object ]]>
    </wadl:doc>
  </wadl:param>
  - <wadl:method name="GET">
    - <wadl:doc>
      - <![CDATA[
        | The method will return the single artifact in the local
        | storage.
      ]]>
    </wadl:doc>
    <wadl:request />
  - <wadl:response>
    - <wadl:doc>
      <![CDATA[ Artifact single object ]]>
    </wadl:doc>
    <wadl:representation mediaType="application/xml" element="idra:artifact"
      xmlns:idra="http://www.idra.info/rest" />
    <wadl:representation mediaType="application/json" />
  </wadl:response>
  </wadl:method>
</wadl:resource>
- <wadl:resource path="/category/all">
  - <wadl:method name="GET">
    - <wadl:doc>
      - <![CDATA[
        | The method will return the list of object's categories in the local
        | storage.
      ]]>
    </wadl:doc>
  - <wadl:response>
    - <wadl:doc>
      <![CDATA[ List of category ]]>
    </wadl:doc>
    <wadl:representation mediaType="application/xml" />
    <wadl:representation mediaType="application/json" />
  </wadl:response>
  </wadl:method>
</wadl:resource>
- <wadl:resource path="/material/all">
  - <wadl:method name="GET">
    - <wadl:doc>
      - <![CDATA[
        | The method will return the list of object's materials in the local
        | storage.
      ]]>
    </wadl:doc>
  - <wadl:response>
    - <wadl:doc>
      <![CDATA[ List of materials ]]>
    </wadl:doc>
    <wadl:representation mediaType="application/xml" />
    <wadl:representation mediaType="application/json" />
  </wadl:response>
</wadl:method>

```

```
</wadl:resource>
- <wadl:resource path="/type/all">
- <wadl:method name="GET">
- <wadl:doc>
- <![CDATA[
| The method will return the list of object's typologies in the local
| storage.
]]>
</wadl:doc>
- <wadl:response>
- <wadl:doc>
<![CDATA[ List of category ]]>
</wadl:doc>
<wadl:representation mediaType="application/xml" />
<wadl:representation mediaType="application/json" />
</wadl:response>
</wadl:method>
</wadl:resource>
</wadl:resources>
</wadl:application>
```

DATA MODEL

REST Data Model Files and Libraries

 Home > Data Model

Data Model

All endpoints act on a common set of data. The data can be represented with difference media (i.e. "MIME") types, depending on the endpoint that consumes and/or produces the data. The data can described by [XML Schema](#), which definitively describes the XML representation of the data, but is also useful for describing the other formats of the data, such as [JSON](#).

This document will describe the data using terms based on [XML Schema](#). Data can be grouped by namespace, with a schema document describing the elements and types of the namespace. Generally speaking, types define the structure of the data and elements are instances of a type. For example, elements are usually produced by (or consumed by) a REST endpoint, and the structure of each element is described by its type.

Namespace "ns0"

Namespace URI:	http://www.idra.info/rest
XSD:	idra.xsd

Data Elements

- [age](#)
- [artifact](#)
- [dimension](#)
- [inventory](#)
- [item](#)
- [location](#)
- [museum](#)
- [picture](#)

Data Types

- [age](#)
- [artifact](#)
- [dimension](#)
- [inventory](#)
- [itemBibliography](#)
- [location](#)
- [museum](#)
- [picture](#)


```

<?xml version="1.0" encoding="UTF-8" ?>
- <xs:schema version="1.0" targetNamespace="http://www.idra.info/rest" xmlns:idra="http://www.idra.info/rest"
  xmlns:xs="http://www.w3.org/2001/XMLSchema">
  <xs:element name="age" type="idra:age" />
  <xs:element name="artifact" type="idra:artifact" />
  <xs:element name="dimension" type="idra:dimension" />
  <xs:element name="inventory" type="idra:inventory" />
  <xs:element name="item" type="idra:itemBibliography" />
  <xs:element name="location" type="idra:location" />
  <xs:element name="museum" type="idra:museum" />
  <xs:element name="picture" type="idra:picture" />
- <xs:complexType name="age">
  - <xs:sequence>
    <xs:element name="from" type="xs:int" minOccurs="0" />
    <xs:element name="to" type="xs:int" minOccurs="0" />
  </xs:sequence>
</xs:complexType>
- <xs:complexType name="artifact">
  - <xs:sequence>
    - <xs:element name="pictures" minOccurs="0" >
      - <xs:complexType>
        - <xs:sequence>
          <xs:element name="picture" type="idra:picture" minOccurs="0" maxOccurs="unbounded" />
        </xs:sequence>
      </xs:complexType>
    </xs:element>
    <xs:element name="category" type="xs:string" minOccurs="1" />
    <xs:element name="epoch" type="xs:string" minOccurs="0" />
    <xs:element name="extended-description" type="xs:string" minOccurs="0" />
  - <xs:element name="materials" minOccurs="0" >
    - <xs:complexType>
      - <xs:sequence>
        <xs:element name="material" type="xs:string" minOccurs="0" maxOccurs="unbounded" />
      </xs:sequence>
    </xs:complexType>
  </xs:element>
  <xs:element name="current-location" type="idra:location" minOccurs="0" />
- <xs:element name="inventories" minOccurs="0" >
  - <xs:complexType>
    - <xs:sequence>
      <xs:element name="inventory" type="idra:inventory" minOccurs="0" maxOccurs="unbounded" />
    </xs:sequence>
  </xs:complexType>
</xs:element>
<xs:element name="age" type="idra:age" minOccurs="0" />
<xs:element name="finding-location" type="idra:location" minOccurs="0" />
<xs:element name="nctn" type="xs:string" minOccurs="1" />
<xs:element name="description" type="xs:string" minOccurs="0" />
- <xs:element name="dimensions" minOccurs="0" >
  - <xs:complexType>
    - <xs:sequence>
      <xs:element name="dimension" type="idra:dimension" minOccurs="0" maxOccurs="unbounded" />
    </xs:sequence>
  </xs:complexType>
</xs:element>
- <xs:element name="bibliography" minOccurs="0" >
  - <xs:complexType>
    - <xs:sequence>
      <xs:element name="item" type="idra:itemBibliography" minOccurs="0" maxOccurs="unbounded" />
    </xs:sequence>
  </xs:complexType>
</xs:element>
<xs:element name="museum" type="idra:museum" minOccurs="0" />
</xs:sequence>
</xs:complexType>
- <xs:complexType name="dimension">
  - <xs:sequence>
    <xs:element name="thickness" type="xs:double" minOccurs="0" />
    <xs:element name="weight" type="xs:double" minOccurs="0" />
    <xs:element name="length" type="xs:double" minOccurs="0" />
    <xs:element name="unit" type="xs:string" minOccurs="0" />
    <xs:element name="width" type="xs:double" minOccurs="0" />
    <xs:element name="height" type="xs:double" minOccurs="0" />
  </xs:sequence>
</xs:complexType>
- <xs:complexType name="inventory">
  - <xs:sequence>
    <xs:element name="reference" type="xs:string" minOccurs="0" />
    <xs:element name="number" type="xs:int" minOccurs="0" />

```

```
</xs:sequence>
</xs:complexType>
- <xs:complexType name = "itemBibliography">
- <xs:sequence>
  <xs:element name = "book" type = "xs:string" minOccurs = "0" />
  <xs:element name = "author" type = "xs:string" minOccurs = "0" />
</xs:sequence>
</xs:complexType>
- <xs:complexType name = "location">
- <xs:sequence>
  <xs:element name = "other" type = "xs:string" minOccurs = "0" />
  <xs:element name = "latitude" type = "xs:double" minOccurs = "0" />
  <xs:element name = "region" type = "xs:string" minOccurs = "1" />
  <xs:element name = "country" type = "xs:string" minOccurs = "1" />
  <xs:element name = "city" type = "xs:string" minOccurs = "1" />
  <xs:element name = "province" type = "xs:string" minOccurs = "1" />
  <xs:element name = "longitude" type = "xs:double" minOccurs = "0" />
</xs:sequence>
</xs:complexType>
- <xs:complexType name = "museum">
- <xs:sequence>
  <xs:element name = "name" type = "xs:string" minOccurs = "0" />
  <xs:element name = "code" type = "xs:string" minOccurs = "0" />
</xs:sequence>
</xs:complexType>
- <xs:complexType name = "picture">
- <xs:sequence>
  <xs:element name = "number" type = "xs:string" minOccurs = "0" />
  <xs:element name = "year" type = "xs:int" minOccurs = "0" />
  <xs:element name = "url" type = "xs:string" minOccurs = "0" />
</xs:sequence>
</xs:complexType>
</xs:schema>
```

age element

Type:	age
Namespace:	http://www.idra.info/rest
XML Schema:	idra.xsd

Example XML

```
<?xml version="1.0" encoding="UTF-8"?><age xmlns="http://www.idra.info/rest"> <from  
xmlns="">... </from> <to xmlns="">... </to></age>
```

Example JSON

```
{ "from" : ..., "to" : ...}
```

ARTIFACT

REST Data Model Files and Libraries

[Home](#) > [Data Model](#) > ns0 > artifact element

artifact element

Type:	artifact
Namespace:	http://www.idra.info/rest
XML Schema:	idra.xsd

Example XML

```
<?xml version="1.0" encoding="UTF-8"?><artifact xmlns="http://www.idra.info/rest">
<pictures xmlns="">  <picture>  <number>...</number>  <year>...</year>
<url>...</url>  </picture>  <picture>  <!--...-->  </picture>  <!--...more "picture"
elements...-->  </pictures>  <category xmlns="">...</category>  <epoch
xmlns="">...</epoch>  <extended-description xmlns="">...</extended-description>
<materials xmlns="">  <material>...</material>  <material>...</material>  <!--...more
"material" elements...-->  </materials>  <current-location xmlns="">  <other>...</other>
<latitude>...</latitude>  <region>...</region>  <country>...</country>  <city>...</city>
<province>...</province>  <longitude>...</longitude>  </current-location>  <inventories
xmlns="">  <inventory>  <reference>...</reference>  <number>...</number>
</inventory>  <inventory>  <!--...-->  </inventory>  <!--...more "inventory"
elements...-->  </inventories>  <age xmlns="">  <from>...</from>  <to>...</to>  </age>
<finding-location xmlns="">  <other>...</other>  <latitude>...</latitude>
<region>...</region>  <country>...</country>  <city>...</city>  <province>...</province>
<longitude>...</longitude>  </finding-location>  <nctn xmlns="">...</nctn>  <description
xmlns="">...</description>  <dimensions xmlns="">  <dimension>
```

Example JSON

```
{ "pictures" : [ { "number" : "...", "year" : ..., "url" : "..." }, ... ], "category" : "...", "epoch" :
...", "extended-description" : "...", "materials" : [ "...", ... ], "current-location" : { "other" : "...",
"latitude" : ..., "region" : "...", "country" : "...", "city" : "...", "province" : "...", "longitude" :
... }, "inventories" : [ { "reference" : "...", "number" : ... }, ... ], "age" : { "from" : ..., "to" :
... }, "finding-location" : { "other" : "...", "latitude" : ..., "region" : "...", "country" : "...",
"city" : "...", "province" : "...", "longitude" : ... }, "nctn" : "...", "description" : "...",
"dimensions" : [ { "thickness" : ..., "weight" : ..., "length" : ..., "unit" : "...", "width" : ...,
"height" : ... }, ... ], "bibliography" : [ { "book" : "...", "author" : "..." }, ... ], "museum" : {
"name" : "...", "code" : "..." } }
```

dimension element

Type:	dimension
Namespace:	http://www.idra.info/rest
XML Schema:	idra.xsd

Example XML

```
<?xml version="1.0" encoding="UTF-8"?><dimension xmlns="http://www.idra.info/rest">
<thickness xmlns="">...</thickness> <weight xmlns="">...</weight> <lenght
xmlns="">...</lenght> <unit xmlns="">...</unit> <width xmlns="">...</width> <height
xmlns="">...</height></dimension>
```

Example JSON

```
{ "thickness" : ..., "weight" : ..., "lenght" : ..., "unit" : "...", "width" : ..., "height" : ...}
```

inventory element

Type:	inventory
Namespace:	http://www.idra.info/rest
XML Schema:	idra.xsd

Example XML

```
<?xml version="1.0" encoding="UTF-8"?><inventory xmlns="http://www.idra.info/rest">
<reference xmlns="">...</reference> <number xmlns="">...</number></inventory>
```

Example JSON

```
{ "reference" : "...", "number" : ...}
```

item element

Type:	itemBibliography
Namespace:	http://www.idra.info/rest
XML Schema:	idra.xsd

Example XML

```
<?xml version="1.0" encoding="UTF-8"?><item xmlns="http://www.idra.info/rest"> <book  
xmlns="">... </book> <author xmlns="">... </author></item>
```

Example JSON

```
{ "book" : "...", "author" : "..."}
```

LOCATION

[REST](#) [Data Model](#) [Files and Libraries](#)

 Home > Data Model > ns0 > **location element**

location element

Type:	location
Namespace:	http://www.idra.info/rest
XML Schema:	idra.xsd

Example XML

```
<?xml version="1.0" encoding="UTF-8"?><location xmlns="http://www.idra.info/rest"> <other  
xmlns="">...</other> <latitude xmlns="">...</latitude> <region xmlns="">...</region>  
<country xmlns="">...</country> <city xmlns="">...</city> <province  
xmlns="">...</province> <longitude xmlns="">...</longitude></location>
```

Example JSON

```
{ "other" : "...", "latitude" : ..., "region" : "...", "country" : "...", "city" : "...", "province" : "...",  
"longitude" : ...}
```

Copyright © 2013 2013 Andromeda

Generated by [Enunciate](#).

museum element

Type:	museum
Namespace:	http://www.idra.info/rest
XML Schema:	idra.xsd

Example XML

```
<?xml version="1.0" encoding="UTF-8"?><museum xmlns="http://www.idra.info/rest"> <name  
xmlns="">... </name> <code xmlns="">... </code></museum>
```

Example JSON

```
{ "name" : "...", "code" : "..."}
```

picture element

Type:	picture
Namespace:	http://www.idra.info/rest
XML Schema:	idra.xsd

Example XML

```
<?xml version="1.0" encoding="UTF-8"?><picture xmlns="http://www.idra.info/rest">
<number xmlns="">...</number>  <year xmlns="">...</year>  <url
xmlns="">...</url></picture>
```

Example JSON

```
{ "number" : "...", "year" : ..., "url" : "..."}
```

[Home](#) >[Data Model](#) >

ns0 >

[age](#)

age

Namespace:	http://www.idra.info/rest
XML Schema:	idra.xsd

XML

Elements

name (type)	min/max	description
	occurs	
from (int)	0/1	
to (int)	0/1	

JSON

property	type	description
from	from (int)	
to	to (int)	

ARTIFACT

REST Data Model Files and Libraries

 Home > Data Model > ns0 > artifact

artifact

Namespace:	http://www.idra.info/rest
XML Schema:	idra.xsd

XML

Elements

name (type)	min/max occurs	description
pictures/picture (picture)	0/unbounded	
category (string)	1/1	
epoch (string)	0/1	
extended-description (string)	0/1	
materials/material (string)	0/unbounded	
current-location (location)	0/1	
inventories/inventory (inventory)	0/unbounded	
age (age)	0/1	
finding-location (location)	0/1	
nctn (string)	1/1	
description (string)	0/1	
dimensions/dimension (dimension)	0/unbounded	
bibliography/item (itemBibliography)	0/unbounded	
museum (museum)	0/1	

JSON

property	type	description
pictures	array of pictures/picture (picture)	
category	category (string)	
epoch	epoch (string)	
extended-description	extended-description (string)	
materials	array of materials/material (string)	
current-location	current-location (location)	

inventories	array of inventories/inventory (inventory)
age	age (age)
finding-location	finding-location (location)
nctn	nctn (string)
description	description (string)
dimensions	array of dimensions/dimension (dimension)
bibliography	array of bibliography/item (itemBibliography)
museum	museum (museum)

Copyright © 2013 2013 Andromeda

Generated by [Enunciate](#).

DIMENSION

[REST](#) [Data Model](#) [Files and Libraries](#)

 Home > Data Model > ns0 > dimension

dimension

Namespace:	http://www.idra.info/rest
XML Schema:	idra.xsd

XML

Elements

name (type)	min/max	description
	occurs	
thickness (double)	0/1	
weight (double)	0/1	
length (double)	0/1	
unit (string)	0/1	
width (double)	0/1	
height (double)	0/1	

JSON

property	type	description
thickness	thickness (double)	
weight	weight (double)	
length	length (double)	
unit	unit (string)	
width	width (double)	
height	height (double)	

INVENTORY

[REST](#) [Data Model](#) [Files and Libraries](#)

 Home > Data Model > ns0 > **inventory**

inventory

Namespace:	http://www.idra.info/rest
XML Schema:	idra.xsd

XML

Elements

name (type)	min/max occurs	description
reference (string)	0/1	
number (int)	0/1	

JSON

property	type	description
reference	reference (string)	
number	number (int)	

Copyright © 2013 2013 Andromeda

Generated by [Enunciate](#).

ITEMBIBLIOGRAPHY

REST Data Model Files and Libraries

 Home > Data Model > ns0 > itemBibliography

itemBibliography

Namespace: <http://www.idra.info/rest>

XML Schema: [idra.xsd](#)

XML

Elements

name (type)	min/max	description
	occurs	
book (string)	0/1	
author (string)	0/1	

JSON

property	type	description
book	book (string)	
author	author (string)	

Copyright © 2013 2013 Andromeda

Generated by [Enunciate](#).

LOCATION

[REST](#) [Data Model](#) [Files and Libraries](#)

 Home > Data Model > ns0 > **location**

location

Namespace:	http://www.idra.info/rest
XML Schema:	idra.xsd

XML

Elements

name (type)	min/max occurs	description
other (string)	0/1	
latitude (double)	0/1	
region (string)	1/1	
country (string)	1/1	
city (string)	1/1	
province (string)	1/1	
longitude (double)	0/1	

JSON

property	type	description
other	other (string)	
latitude	latitude (double)	
region	region (string)	
country	country (string)	
city	city (string)	
province	province (string)	
longitude	longitude (double)	

MUSEUM

[REST](#) [Data Model](#) [Files and Libraries](#)

 Home > Data Model > ns0 > museum

museum

Namespace:	http://www.idra.info/rest
XML Schema:	idra.xsd

XML

Elements

name (type)	min/max	description
name	occurs	
name (string)	0/1	
code (string)	0/1	

JSON

property	type	description
name	name (string)	
code	code (string)	

Copyright © 2013 2013 Andromeda

Generated by [Enunciate](#).

PICTURE

REST Data Model Files and Libraries

 Home > Data Model > ns0 > picture

picture

Namespace:	http://www.idra.info/rest
XML Schema:	idra.xsd

XML

Elements

name (type)	min/max occurs	description
number (string)	0/1	
year (int)	0/1	
url (string)	0/1	

JSON

property	type	description
number	number (string)	
year	year (int)	
url	url (string)	

Copyright © 2013 2013 Andromeda

Generated by [Enunciate](#).

FILES AND LIBRARIES

REST Data Model Files and Libraries

 Home > Files and Libraries

Files and Libraries

The following files and libraries are available:

- [License](#)
- [C Client Library](#)
- [.NET Client Library](#)
- [Java Client Library](#)
- [Java JSON Client Library](#)
- [Objective C Client Library](#)

License

Created 20-giu-2013 22.05.28

Automatically added file

Files

name	size
LICENSE.txt	11,00bytes

C Client Library

Created 20-giu-2013 23.17.10

Introduction

The C module generates the source code for the ANSI-C-compatible data structures and (de)serialization functions that can be used in conjunction with [libxml2](#) to (de)serialize the REST resources as they are represented as XML data.

The generated C source code depends on the [XML Reader API](#) and the [XML Writer API](#) as well as the `<time.h>`, `<string.h>`, and `<stdlib.h>` C standard libraries.

REST XML Example

```
#include <idra-rest.c> //...xmlTextReaderPtr reader = ...; //set up the reader to the
url.idra_rest_idra_artifact *response_element = ...;response_element =
xml_read_idra_rest_idra_artifact(reader); //handle the response as needed...//free the
idra_rest_idra_artifactfree_idra_rest_idra_artifact(response_element);
```

Files

name	size	description
idra-rest.c	176,47K	
enunciate-common.c	39,70K	Common code needed for all projects.

.NET Client Library

Created 20-giu-2013 23.17.12

Introduction

The .NET client-side library defines the classes that can be (de)serialized to/from XML. This is useful for accessing the REST endpoints that are published by this application.

REST Example

```
//read a resource from a REST urlUri uri = new Uri(...);XmlSerializer s = new XmlSerializer(
typeof( Artifact )); //Create the request objectWebRequest req =
WebRequest.Create(uri);WebResponse resp = req.GetResponse();Stream stream =
resp.GetResponseStream();TextReader r = new StreamReader( stream );Artifact order = (Artifact)
s.Deserialize( r );//handle the result as needed...
```

This bundle contains C# source code.

Files

name	size
idra-rest-dotnet.zip	2,46K

Java Client Library

Created 20-giu-2013 23.17.12

Introduction

The Java client-side library is used to access the Web service API for this application.

The JAX-WS client-side library is used to provide the set of Java objects that can be serialized to/from XML using [JAXB](#). This is useful for accessing the REST endpoints that are published by this application.

REST Example (Raw JAXB)

```
java.net.URL url = new java.net.URL(baseUrl + "/artifact/{nctn}");JAXBContext context =  
JAXBContext.newInstance(Artifact.class);java.netURLConnection connection =  
url.openConnection();connection.connect();Unmarshaller unmarshaller =  
context.createUnmarshaller();Artifact result = (Artifact) unmarshaller.unmarshal(  
connection.getInputStream());//handle the result as needed...
```

REST Example (Jersey client)

```
com.sun.jersey.api.client.Client client = com.sun.jersey.api.client.Client.create();Artifact result =  
client.resource(baseUrl + "/artifact/{nctn}").get(Artifact.class); //handle the result as needed...
```

Files

name	size	description
idra-rest-client.jar	10,38K	The binaries for the Java client library.
idra-rest-client-sources.jar	6,59K	The sources for the Java client library.

Java JSON Client Library

Created 20-giu-2013 23.17.12

Introduction

The Java client-side library is used to provide the set of Java objects that can be serialized to/from JSON using [Jackson](#). This is useful for accessing the JSON REST endpoints that are published by this application.

REST Example (Raw Jackson)

```
java.net.URL url = new java.net.URL(baseUrl + "/artifact/{nctn}");ObjectMapper mapper = new  
ObjectMapper();java.netURLConnection connection =  
url.openConnection();connection.connect();Artifact result = (Artifact) mapper.readValue(  
connection.getInputStream(), Artifact.class); //handle the result as needed...
```

Files

name	size	description
idra-rest-json-client.jar	8,48K	The binaries for the Java JSON client library.
idra-rest-json-client-sources.jar	6,18K	The sources for the Java JSON client library.

Objective C Client Library

Created 20-giu-2013 23.17.10

Introduction

The Objective C module generates the source code for the Objective C classes and (de)serialization functions that can be used in conjunction with [libxml2](#) to (de)serialize the REST resources as they are represented as XML data.

The generated Objective C source code depends on the [XML Reader API](#) and the [XML Writer API](#) as well as the base OpenStep foundation classes.

REST XML Example

```
#import <idra-rest.h> //...IDRA_RESTIDRAArtifact *responseElement;NSData *responseData;
//data holding the XML from the response.NSURL *baseURL = ...; //the base url including the host
and subpath.NSURL *url = [NSURL URLWithString: @"/artifact/{nctn}" relativeToURL:
baseURL];NSMutableURLRequest *request = [[NSMutableURLRequest alloc]
initWithURL:url];NSURLResponse *response = nil;NSError *error = NULL;[request setHTTPMethod:
@"GET"]; //this example uses a synchronous request, //but you'll probably want to use an
asynchronous callresponseData = [NSURLConnection sendSynchronousRequest:request
returningResponse:&response error:&error];IDRA_RESTIDRAArtifact *responseElement =
[IDRA_RESTIDRAArtifact readFromXML: responseData];[responseElement retain]; //handle the
response as needed...
```

Files

name	size	description
idra-rest.h	8,84K	
idra-rest.m	137,38K	
enunciate-common.h	12,81K	Common header needed for all projects.
enunciate-common.m	42,61K	Common implementation code needed for all projects.