



Type: Web Service API

Doc. Name.: XTEST

Review: 4.9.1

Date: 19 giugno, 2015

All information, data, ideas, layouts, drawings, schemes and all their combinations thereof are proprietary information of THRON S.p.A. and may be protected by international copyright laws and other intellectual property rights.

Total or partial reproduction, as well as any other use of the information herein presented, including layouts, drawings, schemes is not allowed without written consent of THRON S.p.A.

Table of Contents

How to compose the http request message	4
xml representation of MAddressBook.....	5
json representation of MAddressBook.....	6
JAVA® Client snippet	6
Flash® Client snippet	7
Date and Time format	9
URL encoding in UTF-8	9
json data representation.....	9
Package: NVP	10
Package: TEST	11
Package: SERVICES	12
Package: MODEL	12
Package: MSG	13
MESSO <>Enumeration>>	14
MTestEnum <>Enumeration>>	15
MTestResponse <>Enumeration>>	16
Package: REQUEST	17
MTestServicelistReq <>Class>>	18
MTestServicepostReq <>Class>>	19
MTestServiceputReq <>Class>>	21
MTestServicetestPcontent2Req <>Class>>	23
MTestServicetestPcontentReq <>Class>>	24
Package: RESPONSE	25
MResponseTest <>Class>>	26
MResponseTestF <>Class>>	27
MResponseTestList <>Class>>	28
MResponseTestP <>Class>>	29
MResponseTestPcontent <>Class>>	31
MTestAClass <>Class>>	32
MTestClassBase <>Class>>	33
MTestClass <>Class>>	34
Package: REST	37
JTestService <>Interface>>	38

Web Service REST API

This document contains the webservice technical specifications of the current component. The SDK docs cover all services that are available for public use. THRON will provide support and backward compatibility analysis only for documented services. Any undocumented service usage will not be supported and may be changed in all its parts without notice. The services are clustered by package and have "J" as prefix (J<service name>). All the elements with "M" as prefix represent the model classes used by the webservice as request or response on the messages (M<class name>). The xml or json structure for the request /response messages fully reflect the structure of the model classes defined in this document.

How to compose the http request message

This section shows an example how to create an http request encoded in json or xml. Every single webservice reported in this document specifies the type of accepted message (GET / POST), and the services marked as <post-json> accept body message encoded in json or xml.

HEADER param on HTTP Request message:

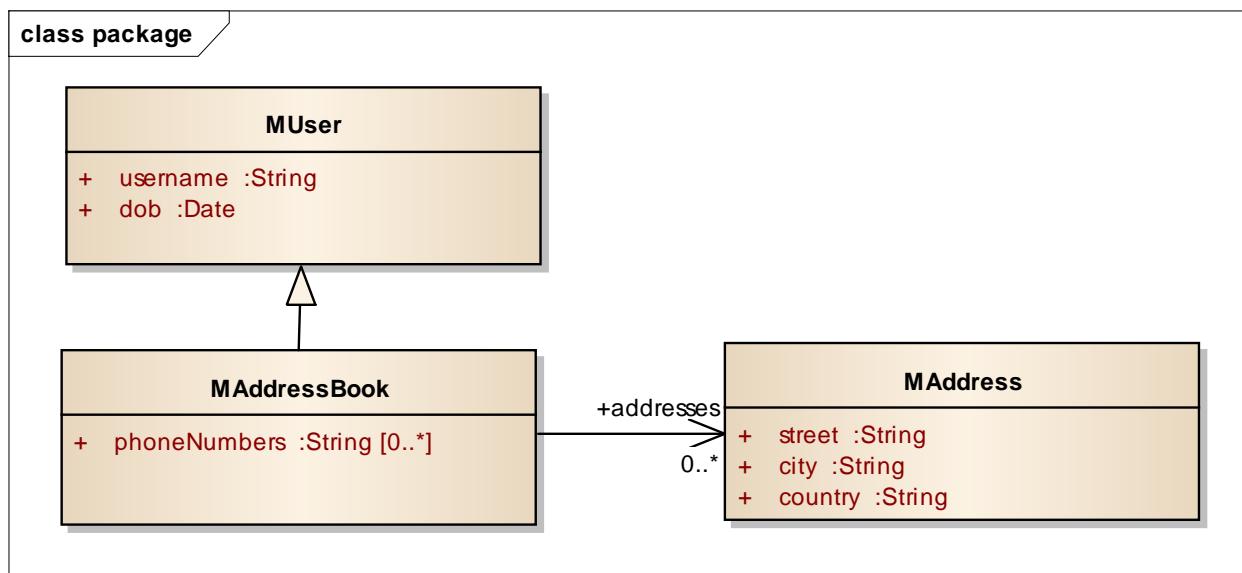
- Accept -> application/json or
- Accept -> application/xml

Operations/Services:

Method Name	Details
findByProperties()	Return Type: <u>MResponse</u>
«get»	<p>Method Notes:</p> <p>here some description about the specific method "findByProperties" and its behavior .</p>
Param Name	Param Details
tokenId	Type: <u>String</u> Tags: @HeaderParam("X-TOKENID") Notes: <i>the tokenId parameter</i>
Param Name	Param Details
id	Type: <u>String</u> Tags: @QueryParam("id") Notes: <i>the content id</i>

Example

Given the following class diagram, it is possible to represent in xml or json format following the definition of the classes.



xml representation of MAddressBook

Follows the xml structure of an object belonging to Muser class.

```

<MAddressBook>
  <username>user1</username>
  <dob>1980-02-01</dob>
  <phoneNumbers></phoneNumbers>
  <phoneNumbers></phoneNumbers>
  <phoneNumbers></phoneNumbers>
  <phoneNumbers></phoneNumbers>
  <addresses>
    <street></street>
    <city></city>
    <country></country>
  </addresses>
  <addresses>
    <street></street>
    <city></city>
    <country></country>
  </addresses>
  <addresses>
    <street></street>
    <city></city>
    <country></country>
  </addresses>
</MAddressBook>
  
```

json representation of MAddressBook

Follows the json structure of an object belonging to Muser class.

```
{"username":"",
"dob"",
"phoneNumbers":["","",""],
"addresses":[{"street":"",
"city":"",
"country":""},
{},
{},
{}]
}
```

JAVA® Client snippet

In this section we report an example of JAVA code used to call THRON RESTful webservice using the provided Java library.

Using the THRON® java libraries, the integration with the Platform is automatic, because Jersey serialize and deserialize automatically the http body request/response in the right java class of THRON.

In this example we have defined a Client Class that calls the JMediaContent.findbyproperties service of Platform.

Dependencies:

```
package sample;

import com.sun.jersey.api.client.Client;
import com.sun.jersey.api.client.ClientResponse;
import com.sun.jersey.api.client.WebResource;
import it.newvision.nvp.xadmin.services.model.mediacontent.MResponseFindContents;
import it.newvision.nvp.xadmin.services.model.request.MMEDIAContentfindContentsByPropertiesReq;

public class JMediaContentClientTest {
    // the resource endpoint: example http://<clientId>-view.4me.it/api/xadmin/resources/
    private String resourceEndpoint = "";

    JMediaContentClientTest(String endpoint) {
        this.resourceEndpoint = endpoint;
    }

    public MResponseFindContents findContentsByProperties(String tokenId, MMEDIAContentfindContentsByPropertiesReq param) {
        // create a new jersey client
        Client client = Client.create();
        try {
            WebResource webResource = client.resource(this.resourceEndpoint);

            // this is a POST web service as defined in the API documentation
            // jersey automatically serialize and deserialiaze (using jakson library) the json/xml request and response
            // to the corresponding java classes
        }
    }
}
```

```
MResponseFindContents response = webResource.path("mediacontent/findContentsByProperties")
    .accept(javax.ws.rs.core.MediaType.APPLICATION_XML)
    .type(javax.ws.rs.core.MediaType.APPLICATION_XML)
    .header("X-TOKENID", tokenId)
    .post(MResponseFindContents.class, param);
return response;
} catch (com.sun.jersey.api.client.UniformInterfaceException e) {
    ClientResponse response = e.getResponse();
    if (response.getStatus() >= 418) {
        return response.getEntity(MResponseFindContents.class);
    } else {
        throw e;
    }
}
```

Flash® Client snippet

In this section we report an example of FLASH code used to create a client class used to call THRON RESTful webservice.

Using the THRON® AS3 libraries the integration with the Platform is really easy, because the THRON AS3 libraries provides a way to serialize and deserialize the xml body request/response in the right AS3 class and deserialize automatically the http body request/response in the right java class of THRON.

In this example we have defined a Client Class that calls the JMediaContent.findByProperties service of Platform.

```
package {
import flash.events.Event;
import flash.events.IOErrorEvent;
import flash.net.URLLoader;
import flash.net.URLLoaderDataFormat;
import flash.net.URLRequest;
import flash.net.URLRequestHeader;
import flash.net.URLRequestMethod;
import it.newvision.nvp.xadmin.services.model.mediacontent.MMediaContentCriteria;
import it.newvision.nvp.xadmin.services.model.mediacontent.MResponseFindContents;
import it.newvision.nvp.xadmin.services.model.request.MMediaContentfindContentsByPropertiesReq;

public class Sample {

    public function Sample(tokenId:String, req:MMediaContentfindContentsByPropertiesReq) : void {
        var endpoint:String = "http://" + clientId + "-view.4me.it/api/xadmin/resources/mediacontent/findContentsByProperties";
        var urlLoader:URLLoader = new URLLoader();
        var request:URLRequest = new URLRequest(endpoint);
        request.data = req.toXml();
        request.method = URLRequestMethod.POST;
        request.contentType = "application/xml";
        request.requestHeaders = new Array();
        request.requestHeaders.push(new URLRequestHeader("Accept", "application/xml"));
        request.requestHeaders.push(new URLRequestHeader("X-TOKENID", tokenId));
        this.urlLoader.dataFormat = URLLoaderDataFormat.TEXT;
        this.urlLoader.addEventListener(Event.COMPLETE, this.onComplete);
        this.urlLoader.addEventListener(IOErrorEvent.IO_ERROR, this.onError);
        this.urlLoader.load(request);
    }
}
```

```
}

public function onComplete(result:String) : void {
    trace("onComplete");
    var response:MResponseFindContents =
        MResponseFindContents.getMResponseFindContents(new XML(result));
}
public function onError(result:String) : void {
    trace("onError: " + result);
}
}
```

Date and Time format

Class attributes or service parameters using Date (and time) Data Type are represented using the ISO 8601 standard (yyyy-MM-dd'T'HH:mm:ss.SSS) in Universal Time Coordinated (UTC)

Examples (XML):

```
<dateFrom>2010-01-15T10:02:04.000</dateFrom>
<dateFrom>2010-05-15</dateFrom>
<dateFrom>2010-05-15T06:00</dateFrom>
```

Examples (json):

```
"dateFrom":"2010-01-15T10:02:04.000"
"dateFrom":2010-05-15"
"dateFrom":2010-05-15T06:05"
```

URL encoding in UTF-8

For a correct representation of the values expressed in query or path params, it is recommended to encode the whole URI in UTF-8 format.

json data representation

This table explains some json data format that could be returned from webservice.

Data Types	Examples
Array with only one element	<pre>"array":"my element" "array":["myelement"] "array":[...] "array":{...}</pre>
Integer/long dataType	<pre>"intValue":3 "intValue":3</pre>
Boolean dataType	<pre>"boolValue":true "boolValue":true</pre>
String dataType	<pre>"stringValue":"my element"</pre>
Date dataType	<pre>"dateValue":"2010-01-15T10:02:04.000Z"</pre>
Object dataType	<pre>"objValue":{ "attribute1":"", "attribute2":"", ... }</pre>
Array of objects	<pre>"intValue":3 "array": [...,{...},...]</pre>

Package: NVP

Parent Package: newvision

Figure: 1

Package: TEST

Parent Package: nvp

Figure: 2

Package: SERVICES

Parent Package: test

Package: MODEL

Parent Package: services

Package: MSG

Parent Package: model

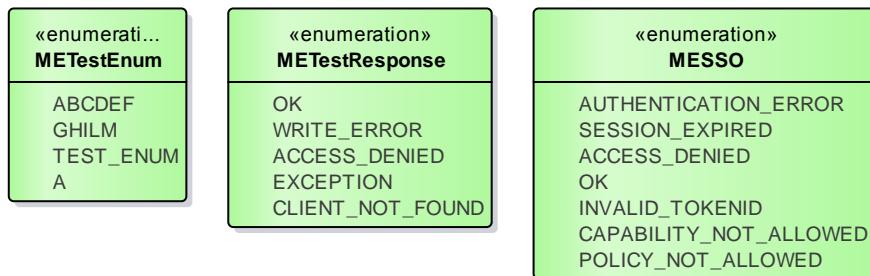


Figure: 5

MESSO <<Enumeration>>

Parent Class:

Stereotype: «enumeration»

Annotations:
@XmlType(name="MESSO")
@XmlEnum

Notes:

Attributes:

Attribute Name	Definition	Multiplicity	Visibility
AUTHENTICATION_ER	Type: ROR	Multiplicity:	Visibility: Public
	Notes:		
SESSION_EXPIRED	Type: Notes:	Multiplicity:	Visibility: Public
ACCESS_DENIED	Type: Notes:	Multiplicity:	Visibility: Public
OK	Type: Notes:	Multiplicity:	Visibility: Public
INVALID_TOKENID	Type: Notes:	Multiplicity:	Visibility: Public
CAPABILITY_NOT_ALL	Type: OWED	Multiplicity:	Visibility: Public
	Notes:		
POLICY_NOT_ALLOW	Type: ED	Multiplicity:	Visibility: Public
	Notes:		

METestEnum <<Enumeration>>**Parent Class:****Stereotype:** «enumeration»**Annotations:**
@XmlType(name="METestEnum")
@XmlEnum**Notes:****Attributes:**

Attribute Name	Definition		
	Type:	Multiplicity:	Visibility: Public
ABCDEF			
	Notes:		
GHILM			
	Notes:		
TEST_ENUM			
	Notes:		
A			
	Notes:		

METestResponse <<Enumeration>>

Parent Class:

Stereotype: «enumeration»

Annotations:
@XmlType(name="METestResponse")
@XmlEnum

Notes:

Attributes:

Attribute Name	Definition		
	Type:	Multiplicity:	Visibility: Public
OK			
	Notes:		
WRITE_ERROR	Type:	Multiplicity:	Visibility: Public
	Notes:		
ACCESS_DENIED	Type:	Multiplicity:	Visibility: Public
	Notes:		
EXCEPTION	Type:	Multiplicity:	Visibility: Public
	Notes:		
CLIENT_NOT_FOUND	Type:	Multiplicity:	Visibility: Public
	Notes:		

Package: REQUEST

Parent Package: model

MTestServicelistReq <<Class>>

Parent Class:

Stereotype:

Annotations: @XmlRootElement(name="MTestServicelistReq")
@XmlType(name="MTestServicelistReq")

Notes: Request message for service JTestService.list

Attributes:

Attribute Name	Definition	Multiplicity:	Visibility:
numberOfResults	Type: Integer Notes: Optional. default value is 50 Define the number of items returned by the service.		Public

MTestServicepostReq <<Class>>**Parent Class:****Stereotype:****Annotations:**
@XmlElement(name="MTestServicepostReq")
@XmlType(name="MTestServicepostReq")**Notes:** Request message for service JTestService.post**Attributes:**

Attribute Name	Definition	Multiplicity:	Visibility:
clientId	Type: String		Visibility: Public
	Notes:		
fakeResponse	Type: Boolean		Visibility: Public
	Notes: Optional.false, return the response filled with the parameters values		
pstring	Type: String		Visibility: Public
	Notes: Optional.		
pdate	Type: Date		Visibility: Public
	Notes: Optional.		
penum	Type: METestEnum		Visibility: Public
	Notes: Optional.		
plong	Type: Long		Visibility: Public
	Notes: Optional.		
pparams	Type: MTestClass		Visibility: Public
	Notes: Optional.		
pint	Type: Integer		Visibility: Public

	Notes: Optional.		
pbool	Type: Boolean	Multiplicity:	Visibility: Public
	Notes: Optional.		
pdouble	Type: Double	Multiplicity:	Visibility: Public
	Notes: Optional.		
pfloat	Type: Float	Multiplicity:	Visibility: Public
	Notes: Optional.		

MTestServiceputReq <<Class>>**Parent Class:****Stereotype:****Annotations:**
@XmlElement(name="MTestServiceputReq")
@XmlType(name="MTestServiceputReq")**Notes:** Request message for service JTestService.put**Attributes:**

Attribute Name	Definition	Multiplicity:	Visibility:
fakeResponse	Type: Boolean	Multiplicity:	Visibility: Public
Notes: false, return the response filled with the parameters values			
pstring	Type: String	Multiplicity:	Visibility: Public
Notes: Optional.			
pdate	Type: Date	Multiplicity:	Visibility: Public
Notes: Optional.			
penum	Type: METestEnum	Multiplicity:	Visibility: Public
Notes: Optional.			
plong	Type: Long	Multiplicity:	Visibility: Public
Notes: Optional.			
pparams	Type: MTestClass	Multiplicity:	Visibility: Public
Notes: Optional.			
pint	Type: Integer	Multiplicity:	Visibility: Public
Notes: Optional.			

	Type: Boolean	Multiplicity:	Visibility: Public
pbool			Notes: Optional.
	Type: Double	Multiplicity:	Visibility: Public
pdouble			Notes: Optional.
	Type: Float	Multiplicity:	Visibility: Public
pfloat			Notes: Optional.

MTestServicetestPcontent2Req <<Class>>**Parent Class:****Stereotype:****Annotations:**
@XmlElement(name="MTestServicetestPcontent2Req")
@XmlType(name="MTestServicetestPcontent2Req")**Notes:** Request message for service JTestService.testPcontent2**Attributes:**

Attribute Name	Definition		
clientId	Type: String	Multiplicity:	Visibility: Public
Notes:			
pcontentId	Type: String	Multiplicity:	Visibility: Public
Notes:			

MTestServicetestPcontentReq <<Class>>**Parent Class:****Stereotype:****Annotations:**
@XmlElement(name="MTestServicetestPcontentReq")
@XmlType(name="MTestServicetestPcontentReq")**Notes:** Request message for service JTestService.testPcontent**Attributes:**

Attribute Name	Definition		
clientId	Type: String	Multiplicity:	Visibility: Public
Notes:			
pContentId	Type: String	Multiplicity:	Visibility: Public
Notes:			

Package: RESPONSE

Parent Package: model

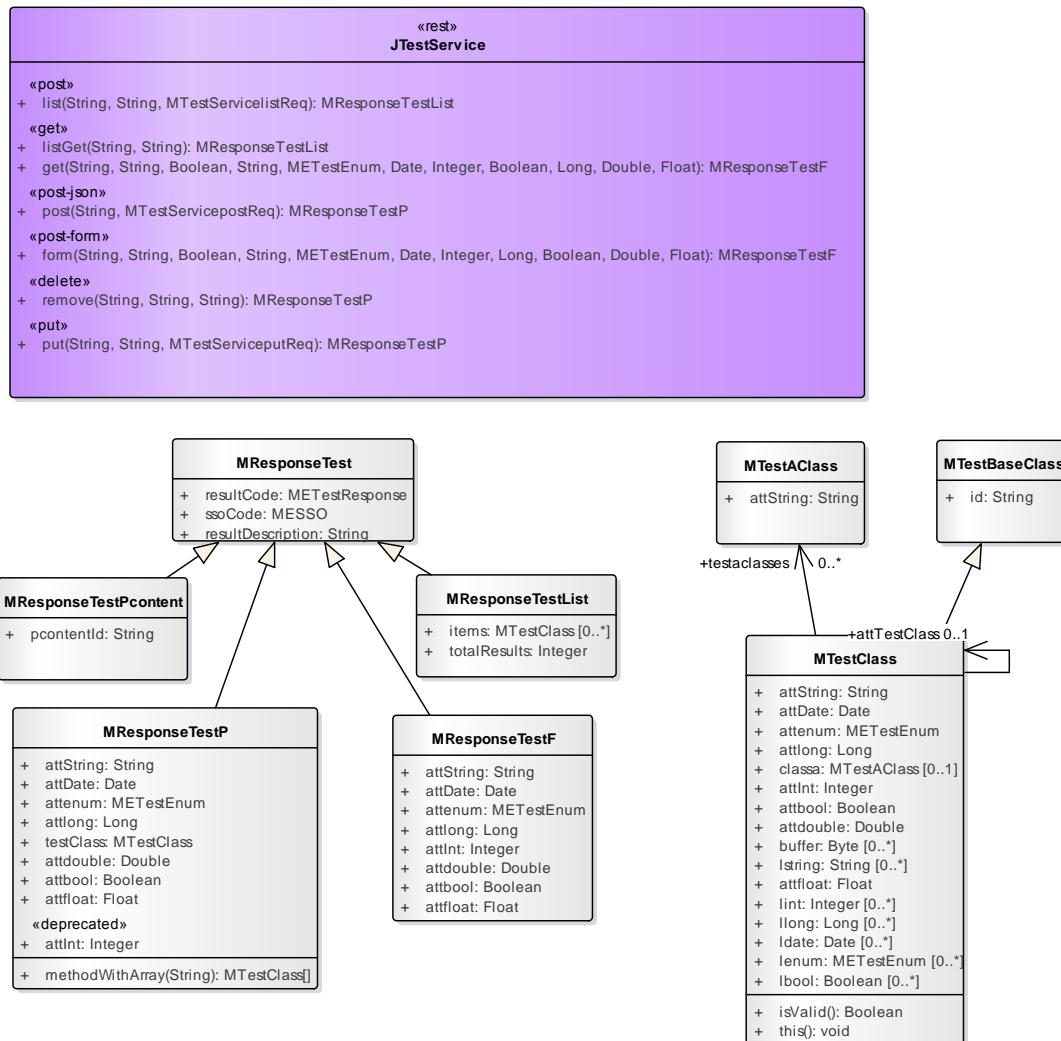


Figure: 7

MResponseTest <<Class>>

Parent Class:

Stereotype:

Annotations:
@XmlElement(name="MResponseTest")
@XmlType(name="MResponseTest")

Notes:

Attributes:

Attribute Name	Definition	Multiplicity	Visibility
resultCode	Type: METestResponse	Multiplicity:	Visibility: Public
Notes:			
ssoCode	Type: MESSO	Multiplicity:	Visibility: Public
Notes:			
resultDescription	Type: String	Multiplicity:	Visibility: Public
Notes:			

MResponseTestF <<Class>>**Parent Class:** MResponseTest**Stereotype:****Annotations:**
@XmlElement(name="MResponseTestF")
@XmlType(name="MResponseTestF")**Notes:****Attributes:**

Attribute Name	Definition	Multiplicity:	Visibility:
attString	Type: String		Visibility: Public
Notes:			
attDate	Type: Date	Multiplicity:	Visibility: Public
Notes:			
attenum	Type: METestEnum	Multiplicity:	Visibility: Public
Notes:			
attlong	Type: Long	Multiplicity:	Visibility: Public
Notes:			
attnum	Type: Integer	Multiplicity:	Visibility: Public
Notes:			
attdouble	Type: Double	Multiplicity:	Visibility: Public
Notes:			
attbool	Type: Boolean	Multiplicity:	Visibility: Public
Notes:			
attfloat	Type: Float	Multiplicity:	Visibility: Public
Notes:			

MResponseTestList <<Class>>

Parent Class: MResponseTest

Stereotype:

Annotations:
@XmlElement(name="MResponseTestList")
@XmlType(name="MResponseTestList")

Notes:

Attributes:

Attribute Name	Definition	Multiplicity	Visibility
items	Type: MTestClass	Multiplicity: [0..*]	Visibility: Public
Notes:			
totalResults	Type: Integer	Multiplicity:	Visibility: Public
Notes:			

MResponseTestP <<Class>>**Parent Class:** MResponseTest**Stereotype:****Annotations:**
@XmlElement(name="MResponseTestP")
@XmlType(name="MResponseTestP")**Notes:****Attributes:**

Attribute Name	Definition	Multiplicity	Visibility
attString	Type: String	Multiplicity:	Visibility: Public
Notes:			
attDate	Type: Date	Multiplicity:	Visibility: Public
Notes:			
attenum	Type: MTestEnum	Multiplicity:	Visibility: Public
Notes:			
attlong	Type: Long	Multiplicity:	Visibility: Public
Notes:			
attnInt	Type: Integer	Multiplicity:	Visibility: Public
Notes:			
testClass	Type: MTestClass	Multiplicity:	Visibility: Public
Notes:			
attdouble	Type: Double	Multiplicity:	Visibility: Public
Notes:			
attbool	Type: Boolean	Multiplicity:	Visibility: Public
Notes:			
attfloat	Type: Float	Multiplicity:	Visibility: Public
Notes:			

Operations/Services:

Method Name	Details	
	Return Type:	REST method:
	Method Notes:	
methodWithArray()		
	Param Name	Param Details
	param1	Type: String Annotations: Notes:

MResponseTestPcontent <<Class>>

Parent Class: MResponseTest

Stereotype:

Annotations:
@XmlElement(name="MResponseTestPcontent")
@XmlType(name="MResponseTestPcontent")

Notes:

Attributes:

Attribute Name	Definition	Multiplicity	Visibility
pcontentId	Type: String		Public
Notes:			

MTestAClass <<Class>>

Parent Class:

Stereotype:

Annotations:
@XmlElement(name="MTestAClass")
@XmlType(name="MTestAClass")

Notes:

Attributes:

Attribute Name	Definition	Multiplicity:	Visibility:
attString	Type: String		Public
Notes:			

MTestBaseClass <<Class>>

Parent Class:

Stereotype:

Annotations:
@XmlElement(name="MTestBaseClass")
@XmlType(name="MTestBaseClass")

Notes:

Attributes:

Attribute Name	Definition	Multiplicity:	Visibility:
id	Type: String		Public
Notes:			

MTestClass <<Class>>**Parent Class:** MTestClass**Stereotype:****Annotations:**
@XmlElement(name="MTestClass")
@XmlType(name="MTestClass")**Notes:****Linked Attributes:**

Linked Attribute Name	Definitions	Multiplicity	Visibility
testaclasses	Type: MTestClass	Multiplicity: [0..*]	Visibility: Public
Notes:			
attTestClass	Type: MTestClass	Multiplicity: [0..1]	Visibility: Public
Notes: attribute note definition			

Attributes:

Attribute Name	Definition	Multiplicity	Visibility
attString	Type: String		Visibility: Public
Notes:			
attDate	Type: Date		Visibility: Public
Notes:			
attenum	Type: MTestEnum		Visibility: Public
Notes:			
attlong	Type: Long		Visibility: Public
Notes:			
classa	Type: MTestClass	Multiplicity: [0..1]	Visibility: Public
Notes:			

attInt	Type: Integer	Multiplicity:	Visibility: Public
	Notes:		
attbool	Type: Boolean	Multiplicity:	Visibility: Public
	Notes:		
attdouble	Type: Double	Multiplicity:	Visibility: Public
	Notes:		
buffer	Type: Byte	Multiplicity: [0..*]	Visibility: Public
	Notes:		
Istring	Type: String	Multiplicity: [0..*]	Visibility: Public
	Notes:		
attfloat	Type: Float	Multiplicity:	Visibility: Public
	Notes:		
lint	Type: Integer	Multiplicity: [0..*]	Visibility: Public
	Notes:		
llong	Type: Long	Multiplicity: [0..*]	Visibility: Public
	Notes:		
ldate	Type: Date	Multiplicity: [0..*]	Visibility: Public
	Notes:		
lenum	Type: METestEnum	Multiplicity: [0..*]	Visibility: Public
	Notes:		
lbool	Type: Boolean	Multiplicity: [0..*]	Visibility: Public
	Notes:		

Operations/Services:

Method Name	Details	
isValid()	Return Type: Boolean	REST method:
	Method Notes:	

Method Name	Details	
this()	Return Type: <u>void</u> Method Notes: default constructor	REST method:

Package: REST

Parent Package: services

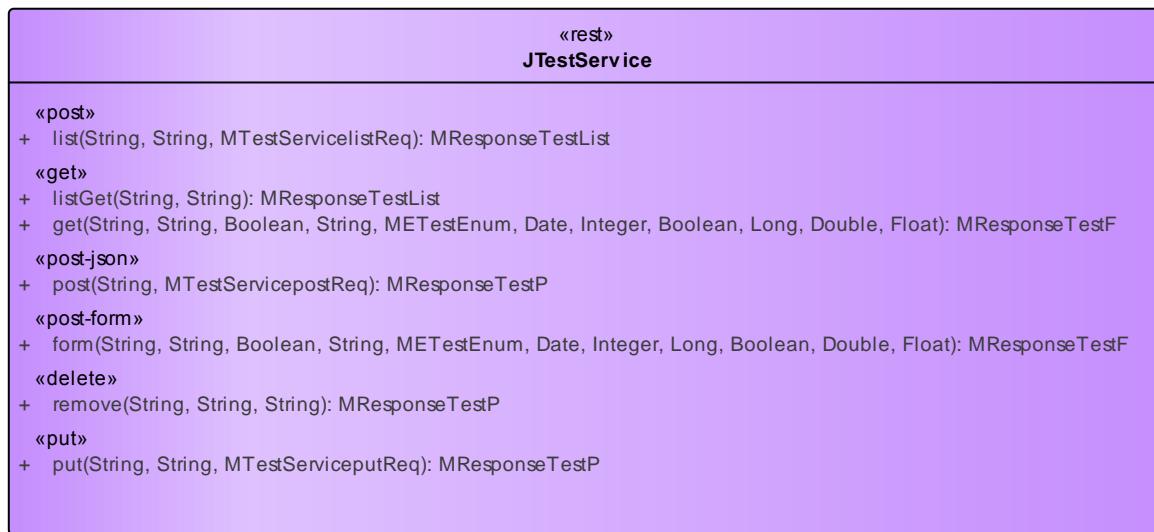


Figure: 8

JTestService <<Interface>>

Parent Class:

Stereotype: «rest»

Annotations: @Path("/testservice")

Notes: This is a test webservice user to test the different type of request and objects used in the platform.

With the fakeResponse parameter to false, the test service respond with the same parameters provided in the call.

Web Service Endpoints:

- WADL REST service (latest version): http://clientId-view.4me.it/api/xtest/resources/application.wadl

Operations/Services:

Method Name	Details				
list()	Return Type: MResponseTestList REST method: «post» Method Notes: return a list of items				
	<table border="1"> <thead> <tr> <th>Param Name</th><th>Param Details</th></tr> </thead> <tbody> <tr> <td>tokenId</td><td>Type: String Annotations: @HeaderParam("X-TOKENID") ##SWG##@ApiParam(name = "X-TOKENID", value = "session token", required=false) Notes:</td></tr> </tbody> </table>	Param Name	Param Details	tokenId	Type: String Annotations: @HeaderParam("X-TOKENID") ##SWG##@ApiParam(name = "X-TOKENID", value = "session token", required=false) Notes:
Param Name	Param Details				
tokenId	Type: String Annotations: @HeaderParam("X-TOKENID") ##SWG##@ApiParam(name = "X-TOKENID", value = "session token", required=false) Notes:				
	<table border="1"> <thead> <tr> <th>Param Name</th><th>Param Details</th></tr> </thead> <tbody> <tr> <td>clientId</td><td>Type: String Annotations: @PathParam("clientId") ##SWG##@ApiParam(value = "") Notes:</td></tr> </tbody> </table>	Param Name	Param Details	clientId	Type: String Annotations: @PathParam("clientId") ##SWG##@ApiParam(value = "") Notes:
Param Name	Param Details				
clientId	Type: String Annotations: @PathParam("clientId") ##SWG##@ApiParam(value = "") Notes:				
	<table border="1"> <thead> <tr> <th>Param Name</th><th>Param Details</th></tr> </thead> </table>	Param Name	Param Details		
Param Name	Param Details				

Method Name	Details	
	param	Type: MTestServicelistReq Annotations: Notes:
listGet()	Return Type: MResponseTestList	REST method: «get» Method Notes: return a list of items
	Param Name	Param Details
	tokenId	Type: String Annotations: @HeaderParam("X-TOKENID") ##SWG##@ApiParam(name = "X-TOKENID", value = "session token", required=false) Notes:
	Param Name	Param Details
	clientId	Type: String Annotations: @PathParam("clientId") ##SWG##@ApiParam(value = "") Notes:
post()	Return Type: MResponseTestP	REST method: «post-json» Method Notes:
	Param Name	Param Details
	tokenId	Type: String Annotations: @HeaderParam("X-TOKENID") ##SWG##@ApiParam(name = "X-TOKENID", value = "session token", required=false) Notes:
	Param Name	Param Details

Method Name	Details	
	param	Type: MTestServicepostReq Annotations: Notes:
get()	Return Type: MResponseTestF	REST method: «get»
	Method Notes:	
	Param Name	Param Details
	tokenId	Type: String Annotations: @HeaderParam("X-TOKENID") ##SWG##@ApiParam(name = "X-TOKENID", value = "session token", required=false)
		Notes:
	Param Name	Param Details
	clientId	Type: String Annotations: @PathParam("clientId") ##SWG##@ApiParam(value = "")
		Notes:
	Param Name	Param Details
	fakeResponse	Type: Boolean Annotations: @QueryParam("fakeResponse") ##SWG##@ApiParam(value = "Optional.")
		Notes: Optional.
	Param Name	Param Details
	pstring	Type: String Annotations: ##SWG##@ApiParam(value = "Optional.") @QueryParam("pstring")
		Notes: Optional.
	Param Name	Param Details

Method Name	Details
	<p>penum</p> <p>Type: METestEnum</p> <p>Annotations:</p> <pre>@QueryParam("penum") //SWG#@ApiParam(value = """Optional""")</pre> <p>Notes: Optional.</p>
Param Name	Param Details
pdate	<p>Type: Date</p> <p>Annotations:</p> <pre>@QueryParam("pdate") //SWG#@ApiParam(value = """Optional.""")</pre> <p>Notes: Optional.</p>
Param Name	Param Details
pint	<p>Type: Integer</p> <p>Annotations:</p> <pre>//SWG#@ApiParam(value = """Optional."""") @QueryParam("pint")</pre> <p>Notes: Optional.</p>
Param Name	Param Details
pbool	<p>Type: Boolean</p> <p>Annotations:</p> <pre>//SWG#@ApiParam(value = """Optional."""") @QueryParam("pbool")</pre> <p>Notes: Optional.</p>
Param Name	Param Details
plong	<p>Type: Long</p> <p>Annotations:</p> <pre>@QueryParam("plong") //SWG#@ApiParam(value = """Optional."")</pre> <p>Notes: Optional.</p>
Param Name	Param Details
pdouble	<p>Type: Double</p> <p>Annotations:</p> <pre>//SWG#@ApiParam(value = """Optional."""") @QueryParam("pdouble")</pre> <p>Notes: Optional.</p>

Method Name	Details	
	Param Name	Param Details
	pfloat	<p>Type: Float</p> <p>Annotations:</p> <pre>##SWG#@ApiParam(value = """Optional.""") @QueryParam("pfloat")</pre> <p>Notes: Optional.</p>
form()	Return Type: MResponseTestF	REST method: «post-form»
	Method Notes:	
	Param Name	Param Details
	tokenId	<p>Type: String</p> <p>Annotations:</p> <pre>@HeaderParam("X-TOKENID") ##SWG#@ApiParam(name = "X-TOKENID", value = "session token", required=false)</pre> <p>Notes:</p>
	Param Name	Param Details
	clientId	<p>Type: String</p> <p>Annotations:</p> <pre>@PathParam("clientId") ##SWG#@ApiParam(value = "")</pre> <p>Notes:</p>
	Param Name	Param Details
	fakeResponse	<p>Type: Boolean</p> <p>Annotations:</p> <pre>@FormParam("fakeResponse") ##SWG#@ApiParam(value = """Optional.""")</pre> <p>Notes: Optional.</p>
	Param Name	Param Details
	pstring	<p>Type: String</p> <p>Annotations:</p> <pre>@FormParam("pstring") ##SWG#@ApiParam(value = """Optional.""")</pre> <p>Notes: Optional.</p>
	Param Name	Param Details

Method Name	Details
	<p>penum</p> <p>Type: METestEnum</p> <p>Annotations:</p> <pre>@FormParam("penum") ##SWG##@ApiParam(value = """Optional.""") </pre> <p>Notes: Optional.</p>
Param Name	Param Details
pdate	<p>Type: Date</p> <p>Annotations:</p> <pre>##SWG##@ApiParam(value = """Optional.""") @FormParam("pdate") </pre> <p>Notes: Optional.</p>
Param Name	Param Details
pint	<p>Type: Integer</p> <p>Annotations:</p> <pre>@FormParam("pint") ##SWG##@ApiParam(value = """Optional.""") </pre> <p>Notes: Optional.</p>
Param Name	Param Details
plong	<p>Type: Long</p> <p>Annotations:</p> <pre>##SWG##@ApiParam(value = """Optional.""") @FormParam("plong") </pre> <p>Notes: Optional.</p>
Param Name	Param Details
pbool	<p>Type: Boolean</p> <p>Annotations:</p> <pre>##SWG##@ApiParam(value = """Optional.""") @FormParam("pbool") </pre> <p>Notes: Optional.</p>
Param Name	Param Details
pdouble	<p>Type: Double</p> <p>Annotations:</p> <pre>@FormParam("pdouble") ##SWG##@ApiParam(value = """Optional.""") </pre> <p>Notes: Optional.</p>

Method Name	Details	
	Param Name	Param Details
	pfloat	<p>Type: Float</p> <p>Annotations:</p> <pre>##SWG#@ApiParam(value = """Optional.""") @FormParam("pfloat")</pre> <p>Notes: Optional.</p>
remove()	Return Type: MResponseTestP	REST method: «delete»
	Method Notes:	test delete service
	Param Name	Param Details
	tokenId	<p>Type: String</p> <p>Annotations:</p> <pre>@HeaderParam("X-TOKENID") ##SWG#@ApiParam(name = "X-TOKENID", value = "session token", required=false)</pre> <p>Notes:</p>
	Param Name	Param Details
	clientId	<p>Type: String</p> <p>Annotations:</p> <pre>@PathParam("clientId") ##SWG#@ApiParam(value = "*****")</pre> <p>Notes:</p>
	Param Name	Param Details
	pstring	<p>Type: String</p> <p>Annotations:</p> <pre>@PathParam("pstring") ##SWG#@ApiParam(value = """Optional.""")</pre> <p>Notes: Optional.</p>
put()	Return Type: MResponseTestP	REST method: «put»
	Method Notes:	test put service

Method Name	Details	
	Param Name	Param Details
	tokenId	<p>Type: String</p> <p>Annotations:</p> <pre>@HeaderParam("X-TOKENID") ##SWG##@ApiParam(name = "X-TOKENID", value = "session token", required=false)</pre>
		<p>Notes:</p>
Param Name	Param Details	
	clientId	<p>Type: String</p> <p>Annotations:</p> <pre>@PathParam("clientId") ##SWG##@ApiParam(value = "")</pre>
		<p>Notes:</p>
Param Name	Param Details	
	param	<p>Type: MTestServiceputReq</p> <p>Annotations:</p>
		<p>Notes:</p>

