

Creating the API

1. In WebDesign, create a new page. The **Activity Group** field and **Page Name** field are not important. **Do not** set table for the page.
2. Add a grid to the page bound to a table. Add joins, custom filtering, and so on, if applicable.

Note

Each grid row corresponds to one item in the output.

3. Drag the columns into that grid that you want exposed as the properties of each item. Use **Column Name** to control the name of each property. Any spaces or dots are removed from the name to increase data portability.

Naming the API

1. Add a **LabelCalc** control and set its **VARNAME** property to be **META.APINAME**.
2. Set the **Default** property to the name of the API to be the URI path that will invoke the API.

Using Input Parameters

1. Add a **LabelCalc** control and set its **VARNAME** property to introduce a parameter. You can pass values to those cells when invoking the API via the query string.
2. Use the parameter cell in your expressions to filter or control behavior.

Nested Collections

1. Add another grid to contain the nested data.
2. To the parent grid, add a new Calc Column and set its **Column Name** property to specify name of collection.
3. Set its **VARNAME** to be **META.JOIN.X**, where *X* is the **GridId** of the child grid.

Set **Default** property to specify key columns in the parent grid and child grid. All rows from the child grid with keys that match the parent grid's keys will be nested under the parent row. Example:
PARENT.F1 = CHILD.F1 AND PARENT.F2 = CHILD.F2

Making the API Savable

1. Add a button with **Text** value: &Save
2. Specify save logic in **CMD** or **ServerCmd** properties.
3. Add a cell with **VARNAME** of **META.ID.X**, where *X* is the **GridId**, and a **Default** value of a comma-separated list of field names if using the PUT method.

Example

If your primary key is in G1 grid in columns F1 and F2, specify **Default** for META.ID.G1 to be: G1.F1, G1.F2

Custom Templating

Add a **META.TEMPLATE** cell whose **Default** property specifies how to transform the response. Only XML templates are currently supported.

- To loop over a grid, use the template:
<# GridId #> anything <#/ GridId #>
- To emit a value from row in grid, use the tag inside the template: <% FieldName %>

Calling the API and Shaping Response

Assume you named the API "myapi". It is available at:
[https://\(site\)/mriweb/mriapiservices/api.asp?\\$api=myapi](https://(site)/mriweb/mriapiservices/api.asp?$api=myapi)

Metadata is at: [api.asp?\\$api=myapi&\\$metadata](https://(site)/mriweb/mriapiservices/api.asp?$api=myapi&$metadata)

Pass parameters:

[api.asp?\\$api=myapi&PROPID=23455&BLDGID=123](https://(site)/mriweb/mriapiservices/api.asp?$api=myapi&PROPID=23455&BLDGID=123)

Desired Shape	HTTP Headers: Accept (read) Content-Type (save)	\$format= (trumps headers)
OData AtomPub	application/atom+xml	atom
OData JSON	application/json	json
Plain XML	application/xml	xml
CSV	text/csv	csv
CSV, no headers	text/csv;header=absent	N/A

HTTP Verbs

GET	Read data. When you visit the API in your web browser, the browser issues a GET request.
POST	Add to collection.
PUT or MERGE	Create new record if does not exist, update if exist. Uses values in META.ID.X columns to merge.

Authentication

Client ID	Database Name	Web Services User	Partner Key	Web Services User's Password
-----------	---------------	-------------------	-------------	------------------------------

P123456/MYDATABASE/WEBUSER/ZYXWVUTSRQ:p455w0rD

Base-64 Encoding

Authorization: Basic UDEyMzQ1Ni9NWURBVEFCQVNFL1dFQ1VTRVlWl1YV1ZVVFNSUTpwNDU1dzByRA==

Sample Read Request

```
GET https://example.com/mriapiservices/api.asp?$api=Vendors&city=Cleveland&State=OH HTTP/1.1
Host: example.com
Accept: application/xml
Authorization: Basic UDEyMzQ1Ni9NWURBVEFCQVNFL1dFQ1VTRVlWl1YV1ZVVFNSUTpwNDU1dzByRA==
```

```
HTTP/1.1 200 OK
Content-Type: application/xml
Date: Mon, 04 May 2015 20:48:14 GMT
Cache-Control: no-store,no-cache,private
Connection: close
```

```
<?xml version="1.0" encoding="utf-8"?>
<vendors>
  <entry>
    <VendorID>V1</VendorID>
    <VendorName>ACME, Inc.</VendorName>
    <City>Cleveland</City>
    <State>OH</State>
    <PostalCode>44123</PostalCode>
  </entry>
</vendors>
```

Sample POST Request

```
POST https://example.com/mriapiservices/api.asp?$api=Vendors HTTP/1.1
Host: example.com
Content-Type: application/xml
Accept: application/xml
Authorization: UDEyMzQ1Ni9NWURBVEFCQVNFL1dFQ1VTRVlWl1YV1ZVVFNSUTpwNDU1dzByRA==
```

```
<?xml version="1.0" encoding="utf-8"?>
<vendors>
  <entry>
    <VendorName>ACME, Inc.</VendorName>
    <City>Cleveland</City>
    <State>OH</State>
    <PostalCode>44123</PostalCode>
  </entry>
  <entry>
    <VendorName>Aperture Laboratories</VendorName>
    <City>Las Vegas</City>
    <State>NV</State>
    <PostalCode>89108</PostalCode>
  </entry>
</vendors>
```

```
HTTP/1.1 200 OK
Content-Type: application/xml
Date: Mon, 04 May 2015 20:48:14 GMT
Cache-Control: no-store,no-cache,private
Connection: close
```

```
<?xml version="1.0" encoding="utf-8"?>
<vendors>
  <entry>
    <VendorID>V1</VendorID>
    <VendorName>ACME, Inc.</VendorName>
    <City>Cleveland</City>
    <State>OH</State>
    <PostalCode>44123</PostalCode>
  </entry>
  <entry>
    <Error>
      <Message>Vendor already exists.</Message>
    </Error>
    <VendorID></VendorID>
    <VendorName>Aperture Laboratories</VendorName>
    <City>Las Vegas</City>
    <State>NV</State>
    <PostalCode>89108</PostalCode>
  </entry>
</vendors>
```

Troubleshooting Common Errors

When an API fails to execute, one of the errors listed below may be returned. The error response body often includes a description of the problem in the response body. If no description is given, use the table below to find potential solutions.

Error	Possible Solutions
400 Bad Request	- Make sure the shape of the data is indicated in the Content-Type header - Make sure you're sending valid data (XML, JSON), according to your Content-Type header. - Make sure the API can be called to save data.
401 Unauthorized	- Make sure the user name in the request contains the appropriate client ID, database name, user name, and partner key, separated by forward slashes. - In the Authorization header, make sure the type is Basic followed by the username:password. - If the partner key is limited to certain clients, make sure client is authorized. - If the partner key is limited to certain APIs, make sure the API being called authorized, or that the partner key is a DEVELOPER key.
405 Method Not Allowed	- Make sure one of the following supported HTTP methods is being used: GET, POST, PUT, MERGE - Make sure the API allows for the method you're using. - If blocked from issuing a PUT or MERGE request by the web server or proxy, use method override functionality and route through POST, specifying the actual method in the override header. <pre>POST https://[MRIWebDomain]/MRIAPIServices/api/InterestingAPI HTTP/1.1 ... X-HTTP-Method-Override: PUT</pre>
406 Not Acceptable	In the Accept or Content-Type headers, make sure a valid format of atom, JSON, or XML was specified. See "Shaping Response" section above.
500 Internal Server Error	- If the partner key is not a DEVELOPER key, receiving this error could mean the API does not match what is in the manifest. - Make sure the API is not defined in multiple activity groups.
503 Could not find more information with that token.	Waited too long before attempting to access next page. Re-issue original request.
503 Trust provider could not authorize the request.	Contact support or your system administrator. Ask them to check the Status of the manifest on the MRI Approved API and Partner Manifest Administration page