

MLA Statistics Database API Methodology

Overview

MLA has recently invested in creating an Application Programming Interface (API) that allow for the automatic transfer of data housed in MLA’s online statistics database, with the view to provide a higher level of service to levy payers and industry stakeholders. This will enable them to reduce or eliminate some of the manual data entry tasks associated with keeping their own data stores up to date. The API functionality must be used in accordance to MLA’s terms and conditions.

This document describes the Methodology of the Application Programming Interface (API) on the MLA statistics database (<http://statistics.mla.com.au>) as well as providing detailed information about the type and data format of the properties on the JSON objects that are returned by the API Methods.

Details

The API is a RESTful Web API that returns a JSON object. It consists of two Methods: A helper Method that will list the available reports, their GUID and any Parameters they take. The Second Method will take the ReportGUID and any report specific parameters in a standard URI query string format. If all the information provided is correct, it will return a JSON object that will have a string property containing an XML formatted copy of the report.

Helper Method (GetReportsList)

This Method will return a list of available reports and give details about what parameters they take. This will help the developer understand what is required for each report.

Method Url	http://statistics.mla.com.au/ReportApi/GetReportList
Parameters	None
HTTP Request Type	GET
Return Type	JSON

Object definitions

[ReturnObject](#) definition

Property Name	Data Type	Description
ResponseDate	String	A string representation of the Date and Time of execution
ResponseHeader	String	A string containing any Method specific information
ResponseStatus	String	Indicates if the request executed OK or had errors.
ResponseDisclaimer	String	Legal disclaimer regarding use of the API
ResponseError	String	If there was an error a message will be in here
ReturnValue	List<ReportDetail>	A list of ReportDetail objects that describe a report

ReportDetail definition

Property Name	Data Type	Description
Name	String	The name of the report
ReportGuid	String	A GUID identifying the report. Used for the RunReport Method
Parameters	List<Parameter>	A list of Parameter objects that describe a report parameter

Parameter definition

Property Name	Data Type	Description
ParameterName	String	The name of the Parameter.
Description	String	The friendly name or the user-prompt for this parameter
IsRequired	Bool	Advises whether the parameter is required or optional
DataType	String	Describes the datatype expected for this parameter
DataFormat	String	Data formatting requirements. Eg DD/MM/YYYY
DefaultValue	String	If optional, the default value or rule will be shown here

Report Execution (RunReport)

This Method takes in the ReportGUID and any parameters detailed for that report in the output of the GetReportList Method mentioned above. The parameters are passed in via standard URI query string format. i.e. ?ReoprtGuid=cf1fea4c-ec96-42c7-9cfa-f34f9a10d415&ParamName=SomeValue

Method Url	http://statistics.mla.com.au/ReportApi/RunReport
Required Parameter	ReportGUID
HTTP Request Type	GET
Return Type	JSON

Object definitions

ReturnObject definition

Property Name	Data Type	Description
ResponseDate	String	A string representation of the Date and Time of execution
ResponseHeader	String	A string containing any Method specific information
ResponseStatus	String	Indicates if request executed OK or had errors.
ResponseDisclaimer	String	Legal disclaimer regarding use of the API
ResponseError	String	If there was an error a message will be in here
ReturnValue	String	An XML document outputted to string, representing the Report

Upon successful execution of a report a developer should be able to programmatically convert the string back into an XML object and iterate through it to retrieve the values they are looking for.

Example Case

EYCI and ESTLI – Daily

Scenario: Find the report definition for the EYCI and ESTLI – Daily report and construct a URI to execute that report, retrieving data from Monday the 13th of March 2017 till Friday the 18th.

1. Look up the report.
 - a. First we contact the Helper function to find the details of the report.
<http://statistics.mla.com.au/ReportApi/GetReportList>

2. Parse the results



- a. We can see from the return JSON that:
 - i. The ReportGuid = 70587516-e17a-4065-a8aa-e3fe4c512159
 - ii. It accepts FromDate and ToDate parameters
3. Construct a RunReport URI
 - a. Base URI = <http://statistics.mla.com.au/ReportApi/RunReport>
 - b. Query string = `?ReportGuid=70587516-e17a-4065-a8aa-e3fe4c512159&FromDate=13%2F03%2F2017&ToDate=18%2F03%2F2017`
 - c. Full URI = <http://statistics.mla.com.au/ReportApi/RunReport?ReportGuid=70587516-e17a-4065-a8aa-e3fe4c512159&FromDate=13%2F03%2F2017&ToDate=18%2F03%2F2017>
 - d. Parse the results:

```
{
  "ReturnValue": "\u003c?xml version=\"1.0\" encoding=\"utf-8\"?\u003e\u003cReport xsi:schemaLocation=\"\" Name=\"Australia -
```

EYCI and ESTLI -

Daily\" xmlns:xsi=\"http://www.w3.org/2001/XMLSchema-
instance\" xmlns=\"Australia_x0020_-
_x0020_EYCI_x0020_and_x0020_ESTLI_x0020_-
_x0020_Daily\" \u003e\u003cTablix4\u003e\u003cCalendarYear_Collection\u003e\u003cCalendarYear_CalendarYear=\"2017\" \u003e\u003cCalendarDate_Collection\u003e\u003cCalendarDate_CalendarDate=\"2017-03-13T00:00:00\" \u003e\u003cTextbox79\u003e\u003cSourceName2_SourceName2=\"MLA\u0027s NLRs\" \u003e\u003cAttributeName3_Collection\u003e\u003cAttributeName3_AttributeName3=\"ESTLI\" ConvertedData=\"612\" / \u003e\u003cAttributeName3_AttributeName3=\"EYCI\" ConvertedData=\"614\" / \u003e\u003c/AttributeName3_Collection\u003e\u003c/SourceName2\u003e\u003c/Textbox79\u003e\u003c/CalendarDate\u003e\u003c/CalendarDate_CalendarDate=\"2017-03-14T00:00:00\" \u003e\u003cTextbox79\u003e\u003cSourceName2_SourceName2=\"MLA\u0027s NLRs\" \u003e\u003cAttributeName3_Collection\u003e\u003cAttributeName3_AttributeName3=\"ESTLI\" ConvertedData=\"616\" / \u003e\u003cAttributeName3_AttributeName3=\"EYCI\" ConvertedData=\"609.25\" / \u003e\u003c/AttributeName3_Collection\u003e\u003c/SourceName2\u003e\u003e\u003c/Textbox79\u003e\u003c/CalendarDate\u003e\u003c/CalendarDate_CalendarDate=\"2017-03-15T00:00:00\" \u003e\u003cTextbox79\u003e\u003cSourceName2_SourceName2=\"MLA\u0027s NLRs\" \u003e\u003cAttributeName3_Collection\u003e\u003cAttributeName3_AttributeName3=\"ESTLI\" ConvertedData=\"621\" / \u003e\u003cAttributeName3_AttributeName3=\"EYCI\" ConvertedData=\"611.25\" / \u003e\u003c/AttributeName3_Collection\u003e\u003c/SourceName2\u003e\u003e\u003c/Textbox79\u003e\u003c/CalendarDate\u003e\u003c/CalendarDate_CalendarDate=\"2017-03-16T00:00:00\" \u003e\u003cTextbox79\u003e\u003cSourceName2_SourceName2=\"MLA\u0027s NLRs\" \u003e\u003cAttributeName3_Collection\u003e\u003cAttributeName3_AttributeName3=\"ESTLI\" ConvertedData=\"622\" / \u003e\u003cAttributeName3_AttributeName3=\"EYCI\" ConvertedData=\"611\" / \u003e\u003c/AttributeName3_Collection\u003e\u003c/SourceName2\u003e\u003e\u003c/Textbox79\u003e\u003c/CalendarDate\u003e\u003c/CalendarDate_Collection\u003e\u003c/CalendarYear\u003e\u003c/CalendarYear_Collection\u003e\u003c/Tablix4\u003e\u003c/Report\u003e\",

"ResponseDate":"22/03/2017 7:46:00 AM",

"ResponseHeader":"Run Report:70587516-e17a-4065-a8aa-e3fe4c512159",

"ResponseStatus":"OK",

"ResponseDisclaimer":"All use of MLA publications, reports and information is subject to MLA's Market Report and Information Terms of Use. Please read our Terms of Use (<http://statistics.mla.com.au/content/documents/MLA-Market-Report-and-Information-Terms-of-use-June-2015.pdf>) carefully and ensure you are familiar with its content.",

```
    "ResponseError":null
  }
}
```

- e. From the above response, we can see that the "ResponseStatus" is "OK" and that there is no "ResponseError", which means we should have a valid ReturnValue. From there you can Parse the ReturnValue property into an XML Document object. If you are using .NET Framework you could, for example, use a System.Xml.Linq.XDocument object and use XDocument.Parse(ReturnValue) to load the return string into an XDocument object.

Please note that the Disclaimer is subject to change.

Troubleshooting

jQuery.ajax() cross site errors

If you are using a jQuery.ajax() function to call the api and you get a pre-flight response check error, remove the contentType setting from the call.

Eg.

```
$.ajax({
  type: 'GET',
  url: 'http://statistics.mla.com.au/ReportApi/GetReportList',
  contentType: 'application/json',
  success: function (data) {
    // do something with data object
  },
  error: function (request, status, error) {
    if (error) {
      console.log(error);
    }
  }
});
```

Will get a cross site error, whereas:

```
$.ajax({
  type: 'GET',
  url: 'http://statistics.mla.com.au/ReportApi/GetReportList',
  success: function (data) {
    // do something with data object
  },
  error: function (request, status, error) {
    if (error) {
      console.log(error);
    }
  }
});
```

Will succeed.