RESTful Model Access (RMA)

RESTful MODEL ACCESS (RMA)

RMA RESTful Model Access (RMA) is an HTTP service designed along RESTful principles to allow access to data in the Allen Institute API. Data model queries support JSON, XML and CSV formats. They can include join, filter, sort, page and eager loading of associations. RMA can also pipe results to perform compound queries and connect to data services. RMA services support JSON and XML formats. The API utility, RMA Query Builder, can be used to compose RMA queries.

RESTful Resources

The path to the RMA service is /api/v2/data. Other models may be substituted for Organism, such as Gene, Chromosome, or Structure. Model names are always capitalized and singular. The number following the model is a resource id. Valid result formats include JSON (.json), XML (.xml) and CSV (.csv)

Prototype

```
http://api.brain-map.org/api/v2/data/[Model]/[Model.id].[json|xml|csv]
```

Examples

Use a browser or other HTTP client to access:

```
http://api.brain-map.org/api/v2/data/Organism/1.xml
http://api.brain-map.org/api/v2/data/Gene/15.xml
http://api.brain-map.org/api/v2/data/Chromosome/12.json
http://api.brain-map.org/api/v2/data/Structure/4005.xml
```

Use the keyword query rather than an id to search across all resources in a model.

```
http://api.brain-map.org/api/v2/data/Organism/query.xml
```

Use the keyword describe to retrieve information about the available associations for a model. .json format is also available.

```
http://api.brain-map.org/api/v2/data/Gene/describe.xml
```

Use the keyword enumerate to retrieve information about all available models and associations. .json format is also available.

```
http://api.brain-map.org/api/v2/data/enumerate.xml
```

Use the keyword query without a model or id if the needed information is specified elsewhere in the url. The part after the question mark is explained throughout this document.

```
http://api.brain-map.org/api/v2/data/query.xml?include=model::Gene[id$eq15]
```

Response

JSON, XML and CSV formats are supported. For example:

```
    <Response success="true" id="0" start_row="0" num_rows="1" total_rows="1">
        <organisms>
            <organism>
                <id>1</id>
                <name>Homo Sapiens</name>
                <ncbitaxonomyid>9606</ncbitaxonomyid>
            </organism>
        </organisms>
    </Response>
```

Include

Resources in the models are associated with other resources. Display this additional information using the include= parameter:

Example

Retrieve details for the Chromosome with ID=12 and include its associated organism:

```
http://api.brain-map.org/api/v2/data/Chromosome/12.xml?include=organism
```

Response

The associated information will be nested in the response:

```
    <Response success="true" id="0" start_row="0" num_rows="1" total_rows="1">
        <chromosomes>
            <chromosome>
                <id>15</id>
                <name>21</name>
                <organism-id>1</organism-id>
                <organism>
                    <id>1</id>
                    <name>Homo Sapiens</name>
                    <ncbitaxonomyid>9606</ncbitaxonomyid>
                </organism>
            </chromosome>
        </chromosomes>
    </Response>
```

The include= parameter is discussed in more detail in the RMA Path Syntax section.

Criteria

Use the criteria parameter to refine the query.

Example

The following criteria will find chromosome resources from the organism with NCBI taxonomy id 9606 (which happens to be Homo Sapiens).

```
http://api.brain-map.org/api/v2/data/Chromosome/query.xml?criteria=organism[ncbitaxonomyid$eq9606]
```

The criteria= parameter is discussed in more detail in the RMA Path Syntax section.

Only, Except And Tabular

Use the only= or except= parameters to restrict the attributes returned in the response. By default the results include all literal attributes of the resource. These options can be applied to model stages, but not service or pipe stages.

Example

```
http://api.brain-map.org/api/v2/data/Organism/1.xml?only=name
```

Response

```
<Response success="true" id="0" start_row="0" num_rows="1" total_rows="1">
    <organisms>
        <organism>
            <name>Homo Sapiens</name>
        </organism>
    </organisms>
</Response>
```

Use tabular= to restrict the attributes and also return them in a tabular format. Use 'as' to alias an attribute name or 'distinct' to eliminate duplicate rows. This option is recommended for accessing data in applications where sorting and paging are used in a tabular, spreadsheet-like display. This option can be applied to model stages, but not service or pipe stages. The csv format is required to be tabular and it defaults to displaying all attributes of the model. If associated models are referenced in the tabular= option, they should be present in a criteria= option as well. The tabular= option masks include= for serialization purposes.

Example

```
http://api.brain-map.org/api/v2/data/Gene/15.xml?criteria=probes&tabular=distinct%20genes.name%20as%20gene_name,
probes.name%20as%20probe_name
```

Response

```
<Response success="true" id="0" start_row="0" num_rows="15" total_rows="15">
    <hash>
        <gene-name>4-aminobutyrate aminotransferase</gene-name>
        <probe-name>RP_100125_04_D02</probe-name>
    </hash>
    .....
</Response>
```

Sorting And Paging

Limit the number of results by using the start_row=, num_rows= and order= URL parameters. Use num_rows=all to retrieve all records (not recommended for large queries). The response body contains information about the progress of paging. Use count=false if you do not want paging information returned in the response.

Prototype

```
http://api.brain-map.org/api/v2/data/[Model]/query.xml?num_rows=[#]&start_row=[#]&order=[...]
```

Example

Request the first Product by specifying start_row=0:

```
http://api.brain-map.org/api/v2/data/Product/query.xml?start_row=0&num_rows=1
```

Request 10 Products beginning with the 20th row, and order by the Products' names:

```
http://api.brain-map.org/api/v2/data/Product/query.xml?num_rows=10&start_row=20&order=products.name
```

Parameters

num_rows	Integer (optional)	Requested page size. The default is 50.
start_row	Integer (optional)	First row number requested. This is zero-based.
order	String (optional)	The value of order= is an attribute name which may be qualified with a lowercase plural table name (i.e. organisms.name). Use a comma-separated list in the order= parameter to sort on multiple attributes. Sort in reverse order by adding +desc to the order= parameter.

Response

num_rows	Number of rows returned.
start_row	First row number requested. This is zero-based.
total_rows	The total number of rows that met the search criteria.

ActiveRecord Query and Serialization

RMA uses the ActiveRecord Query Interface.

Parameters

criteria	Associations are treated like an AR Query joins() argument and can be thought of as a series of SQL inner joins.
include	Associations are treated like an AR Query includes() argument and are used to drive eager loading of the associated resources. They can be thought of as a series of SQL outer joins.
start_row	Generates an offset() AR Query argument and is similar to an SQL OFFSET clause.
num_rows	Generates a limit() AR Query argument and is similar to an SQL ORDER BY clause.
order	Generates an order() AR Query argument and is similar to an SQL ORDER BY clause.
debug	Includes debugging information in the response such as equivalent ActiveRecord and SQL queries when set to 'true'.

The filter clauses in both the criteria= parameter and the includes= parameter are treated as an AR Query where() argument. They can be thought of as an SQL where clause combined with the boolean 'and' operation.

Models are similar to SQL tables, attributes correspond to literal SQL columns and associations can be thought of as SQL foreign keys.