Modification date: June 14, 2012

REST API Documentation for apm.pgia.ru/api service v1

by Juri Katkalov

This documentation contains description of apm.pgia.ru/api service with examples. The documentation is also available online from http://apm.pgia.ru/docs/api in pdf format. For any questions regarding using API service, please contact Juri Katkalov (work@katkalov.com).

Table of contents

API overview

apm.pgia.ru/api service provides access to auroral precipitation model (APM) via REST API. REST API allows to obtain data in different formats (xml, json, text) from common range of applications support web content requesting. See Using apm.pgia.ru/api section for examples.

Current version of apm.pgia.ru/api service provides access to 3 models for APM.

`1. boundaries`	Positions of poleward and equatorword boundaries of diffuse auroral zone (DAZ), auroral oval precipitation (AOP) and soft diffuse precipitation (SDP).
`2. energies`	Average electron precipitation energies (*keV*) in DAZ, AOP and SDP.
`3. fluxes`	Average electron precipitation fluxes (*erg/cm^{2}s*) in DAZ, AOP and SDP.

See also auroral precipitation model decription to get more details about the model.

Boundaries & Regions IDs

Boundaries IDs

`daz_eq`	equatorward boundary of DAZ
`daz_pol`	poleward boundary of DAZ
`aop_eq`	equatorward boundary of AOP
`aop_pol`	poleward boundary of AOP
`sdp_pol`	poleward boundary of SDP
`b4s`	is the latitude of the beginning of special structures in AOP precipitation. The b4s boundary divides the nighttime AOP region into equatorward and poleward parts.

Regions IDs

`daz_eq/daz_pol`	DAZ region, the region of diffuse precipitation equstorward of the auroral oval
`aop_eq/aop_pol`	AOP region, the region of structured precipitation
`aop_pol/sdp_pol`	SDP region, the region of soft diffuse precipitation poleward of AOP

On the nightside:

`aop_eq/b4s`	AOP$_{eq}$, the equatorward part of AOP
`b4s/aop_pol`	AOP$_{pol}$, the poleward part of AOP

See auroral precipitation model description for more details about boundaries and regions.

Approximations

By default, model calculates values in 8 points for each boundary (one point for each 3-hour MLT interval). We use `curvefit` function from *IDL* library to calculate approximate values for each boundary (500 points) and linear interpolation to calculate values for energies and fluxes model in approximated points.

This API allows to obtain both approximated and non- approximated values for each model. To get approximated values, set `approximate` parameter to either `true`, `t` or `1`, in API request. See also Parameters section for api/boundaries, api/energies and api/fluxes for more details.

Token Keys

Token keys use for access to API service for authorized users and applications. You must provide your token key as parameter in requests to API. See Parameters section for api/boundaries, api/energies and api/fluxes for more details.

Note 1: In the current documentation we use `18faea489820` token key only as example for requests. This token key can not be used in real-world applications.

Note 2: Different token keys can provide access to different models.

Note 3: Current version of REST API does not provide automatically generation of token key for new users. Please contact Juri Katkalov (work@katkalov.com) to request token key for access to API.

Response formats

Model outputs can be obtained in one of three supported formats.

xml	Mime-type: `application/xml`
json	Mime-type: `application/json`
text	Mime-type: `text/plain`

To obtain data in specific format, add `.`*`format`* extension to api requests. If `.`*`format`* extension is not set, the output will be sent in default format (see default response format for the model). It is also possible to request output in specific format by setting HTTP Accept-Header.

Request error handling

APi server sends special formated message on request errors . Error response will be send in the same format as a request. See Response Formats for details.

Note: Some applications, such do not display error message when API server return an error code. In this case, use `force_http_ok` parameter in requests to display error message.

XML Response Format

<error> node

@childrens

`text`	Text of error message **Example Value:** `Cannot authenticate: invalid token`

`code`	Internal error code value **Example Value:** `45`
`http_status_code`	A status code, returned by server. See HTTP status codes section for specific model for details. **Note:** if `force_http_ok parameter` is set either to `true`, `t` or `1`, the status code will be always set to `200`. **Example Value:** `200`

<request> node

Contains a request url, which led to the error

JSON Response Format

The structure of JSON Error object is the same as in XML response

TEXT Response Format

Contains four-lines response with the same structure as in XML response

1 line	Text of error message. See text child of error node in XML response for details.
2 line	An Internal error code value. See code child of error node in XML response for details.
3 line	A status code, returned by server. See http_status_code of error node in XML response for details.
4 line	Request url led to error. See request node in XML response for details.

GET api/boundaries

Return a list of boundaries with arrays of boundaries positions as pairs of mlt/lat coordinates.

Request URL Pattern

`http://apm.pgia.ru/api/boundaries.`*`format`*

HTTP Methods

GET

Requires Authentication?

yes

Response Formats

xml (default)

text

json

HTTP status codes

200	OK
400	Bad request. The request cannot be fulfilled due to bad syntax
403	Forbidden. The request was a legal request, but it is required to authenticate request
404	Not found. The requested resource could not be found
500	Internal server error.

Parameters

token mandotary	See Token Keys for details **Example Value:** `18faea489820`
al mandotary	This parameter set value of AL index in nanotesles. Value must be an integer. See auroral precipitation model decription for recommended values for AL. **Example Value:** `-100`

dst mandotary	This parameter sets value of DST index in nanotesles. Value must be an integer. See auroral precipitation model decription for recommended values for DST. **Example Value:** `-10`
approximate optional	When set to either `true`, `t` or `1`, the model outputs will contain approximated values **Example Value:** `true`
force_http_ok optional	When set to either `true`, `t` or `1`, the api server will return a 200 HTTP status code on an error (default return specific status code). This option is recommended to use for compability with applications which can not recognize http error status codes. See HTTP status codes for details. **Example Value:** `true`

XML Response Format Entities

Contains a list of six `boundary` nodes with array of coordinates as pairs of mlt/lat values.

<boundary> node

@attributes

`id`	A boundary id. See Boundaries & Regions IDs for details. **Example Value:** `daz_eq`

@childrens

`description`	Name and description of boundary **Example Value:** `DAZ equatorward boundary`
`coordinates_array`	A list of coordinates of boundary position as pairs of mlt/lat. See <coordinates_array> node for details.

<coordinates_array> node

@attributes

`approximated`	If is set to `true`, then coordinates_array node will contain 500 items. When set to false, `coordinates_array` will contain 8 items of mlt/lat pairs. **Example Value:** `true`

@childrens

`array_item`	A pair of mlt/lat coordinates. See <array_item> node for details.

<array_item> node

@childrens

`mlt`	The value of MLT coordinate **Example Value:** `1.50`
`lat`	The value of latitude coordinate **Example Value:** `63.21`

JSON Response Format Entities

Contains an array of objects, associated with boundaries

Boundary object

@fields

`id`	A boundary id in text format. See Boundaries & Regions IDs for details. **Example Value:** `daz_eq`
`description`	Name and description of boundary in text format. **Example Value:** `DAZ equatorward boundary`
`coordinates`	An object, see coordinates object description.

Coordinates object

@fields

`approximated`	If is set to `true`, then `data_array` field will contain 500 items. When set to `false`, the field will contain 8 items of mlt/lat pairs **Example Value:** `true`
`data_array`	An array of 8-by-2 or 500-by-2 items, containing pairs of mlt/lat values **Example Value:** `[[1.5,63.21],...]`

TEXT Response Format Entities

Contains 8 or 500 lines divided into 7 columns.

Columns description

1 column	MLT coordinate value
2 column	A latitude coordinates of `daz_eq` boundary
3 column	A latitude coordinates of `daz_pol` boundary
4 column	A latitude coordinates of `aop_eq` boundary
5 column	A latitude coordinates of `aop_pol` boundary
6 column	A latitude coordinates of `sdp_pol` boundary
7 column	A latitude coordinates of `b4s` boundary

Examples

request:

http://apm.pgia.ru/api/boundaries.xml?token=18faea489820&al=-100&dst=-10

response:

```
<hash>
<boundary id="daz_eq">
   <description>DAZ equatorward boundary</description>
   <coordinates_array approximated="false">
      <array_item>
         <mlt>1.50</mlt>
         <lat>63.21</lat>
      </array_item>
      <array_item>
         <mlt>4.50</mlt>
         <lat>62.88</lat>
      </array_item>
      <array_item>
         <mlt>7.50</mlt>
         <lat>64.25</lat>
      </array_item>
      <array_item>
         <mlt>10.50</mlt>
         <lat>66.34</lat>
      </array_item>
      <array_item>
         <mlt>13.50</mlt>
         <lat>68.56</lat>
      </array_item>
      <array_item>
         <mlt>16.50</mlt>
         <lat>68.47</lat>
      </array_item>
      <array_item>
         <mlt>19.50</mlt>
         <lat>67.18</lat>
      </array_item>
      <array_item>
         <mlt>22.50</mlt>
         <lat>63.45</lat>
      </array_item>
   </coordinates_array>
</boundary>
...
</hash>
```

request:

http://apm.pgia.ru/api/boundaries.text?token=18faea489820&al=-100&dst=-10

response:

```
1.50  63.21 65.86 65.86 72.87 73.70 69.30
4.50  62.88 66.20 66.20 75.20 76.33 71.07
7.50  64.25 72.00 70.90 76.53 77.19 72.00
10.50 66.34 75.10 73.16 77.13 79.05 75.10
13.50 68.56 76.12 72.79 78.21 79.15 76.12
16.50 68.47 72.53 71.51 76.10 77.35 72.53
19.50 67.18 69.15 69.15 73.51 74.67 70.31
22.50 63.45 65.65 65.65 71.10 71.88 67.53
```

GET api/energies

Return a list of zones with arrays of energy values.

Request URL Pattern

`http://apm.pgia.ru/api/energies.`*`format`*

HTTP Methods

GET

Requires Authentication?

yes

Response Formats

xml (default)
text
json

HTTP status codes

200	OK
400	Bad request. The request cannot be fulfilled due to bad syntax
403	Forbidden. The request was a legal request, but it is required to authenticate request
404	Not found. The requested resource could not be found
500	Internal server error.

Parameters

token mandotary	See Token Keys for details **Example Value:** `18faea489820`
al mandotary	This parameter sets value of AL index in nanotesles. Value must be an integer. See auroral precipitation model decription for recommended values for AL. **Example Value:** `-100`

dst mandotary	This parameter set value of DST index in nanotesles. Value must be an integer. See auroral precipitation model decription for recommended values for DST. **Example Value:** `-10`
approximate optional	When set to either `true`, `t` or `1`, the model outputs will contain approximated values **Example Value:** `true`
force_http_ok optional	When set to either `true`, `t` or `1`, the api server will return a 200 HTTP status code on an error (default return specific status code). This option is recommended to use for compability with applications which can not recognize http error status codes. See HTTP status codes for details. **Example Value:** `true`

XML Response Format Entities

Contains a list of four `region` nodes with array of energies.

<region> node

@attributes

`equatorward_boundary`	A equatorward boundary of the region. See Boundaries & Regions IDs for details. **Example Value:** `daz_eq`
`poleward_boundary`	A poleward boundary of the region. See Boundaries & Regions IDs for details. **Example Value:** `daz_pol`

@childrens

`description`	Name and description of the region **Example Value:** `DAZ region`
`energies_array`	A list items as pairs of mlt/energy values

<energies_array> node

@attributes

`approximated`	If is set to `true`, then `energies_array` node will contain 500 items. When set to `false`, `energies_array` will contain 8 items of mlt/energies pairs. **Example Value:** `true`

@childrens

`array_item`	A pair of mlt/energy values. See <array_item> node for details.

<array_item> node

@childrens

`mlt`	The value of MLT coordinate **Example Value:** `1.50`
`energy`	The value of energy **Example Value:** `3.10`

JSON Response Format Entities

Contains an array of objects, associated with regions

Region object

@fields

`equatorward_boundary`	A equatorward boundary of the region. See Boundaries & Regions IDs for details. **Example Value:** `daz_eq`
`poleward_boundary`	A poleward boundary of the region. See Boundaries & Regions IDs for details. **Example Value:** `daz_pol`

`description`	Name and description of the region **Example Value:** `DAZ region`
`energies`	An object, see Energies object description.

Energies object

@fields

`approximated`	If is set to `true`, then `data_array` field will contain 500 items. When set to `false`, the field will contain 8 items of mlt/energies pairs **Example Value:** `true`
`data_array`	An array of 8-by-2 or 500-by-2 items, containing pairs of mlt/energies values **Example Value:** `[[1.5,3.1],...]`

TEXT Response Format Entities

Contains 8 or 500 lines divided into 5 columns.

Columns description

1 column	MLT coordinate value
2 column	An energy values for `daz_eq/daz_pol` region
3 column	An energy values for `aop_eq/b4s` region
4 column	An energy values for `b4s/aop_pol` region
5 column	An energy values for `aop_pol/sdp_pol` region

Examples

request:

```
http://apm.pgia.ru/api/energies.xml?token=18faea489820&al=-100&dst=-10
```

response:

```
<hash>
<region equatorward_boundary="daz_eq" poleward_boundary="daz_pol">
   <description>DAZ region</description>
   <energies_array approximated="false">
      <array_item>
         <mlt>1.50</mlt>
         <energy>3.10</energy>
      </array_item>
      <array_item>
         <mlt>4.50</mlt>
         <energy>4.15</energy>
      </array_item>
      <array_item>
         <mlt>7.50</mlt>
         <energy>4.26</energy>
      </array_item>
      <array_item>
         <mlt>10.50</mlt>
         <energy>5.16</energy>
      </array_item>
      <array_item>
        <mlt>13.50</mlt>
        <energy>4.15</energy>
      </array_item>
      <array_item>
        <mlt>16.50</mlt>
        <energy>2.42</energy>
      </array_item>
      <array_item>
         <mlt>19.50</mlt>
         <energy>1.37</energy>
      </array_item>
      <array_item>
        <mlt>22.50</mlt>
        <energy>2.07</energy>
      </array_item>
    </energies_array>
</region>
...
</hash>
```

request:

```
http://apm.pgia.ru/api/energies.text?token=18faea489820&al=-100&dst=-10
```

response:

```
1.50  3.10 2.83 1.34 0.57
4.50  4.15 2.86 0.73 0.55
7.50  4.26 1.16 1.16 0.31
10.50 5.16 0.87 0.87 0.27
13.50 4.15 0.57 0.57 0.24
16.50 2.42 0.88 0.88 0.31
19.50 1.37 1.58 1.31 0.41
22.50 2.07 2.83 1.96 0.64
```

GET api/fluxes

Return a list of zones with arrays of flux values.

Request URL Pattern

```
http://apm.pgia.ru/api/fluxes.format
```

HTTP Methods

GET

Requires Authentication?

yes

Response Formats

xml (default)
text
json

HTTP status codes

200	OK
400	Bad request. The request cannot be fulfilled due to bad syntax
403	Forbidden. The request was a legal request, but it is required to authenticate request
404	Not found. The requested resource could not be found
500	Internal server error.

Parameters

token mandotary	See Token Keys for details **Example Value:** `18faea489820`
al mandotary	This parameter sets value of AL index in nanotesles. Value must be an integer. See auroral precipitation model decription for recommended values for AL. **Example Value:** `-100`

dst mandotary	This parameter sets value of DST index in nanotesles. Value must be an integer. See auroral precipitation model decription for recommended values for DST. **Example Value:** `-10`
approximate optional	When set to either `true`, `t` or `1`, the model outputs will contain approximated values **Example Value:** `true`
force_http_ok optional	When set to either `true`, `t` or `1`, the api server will return a 200 HTTP status code on an error (default return specific status code). This option is recommended to use for compability with applications which can not recognize http error status codes. See HTTP status codes for details. **Example Value:** `true`

XML Response Format Entities

Contains a list of four `region` nodes with array of fluxes.

<region> node

@attributes

`equatorward_boundary`	A equatorward boundary of the region. See Boundaries & Regions IDs for details. **Example Value:** `daz_eq`
`poleward_boundary`	A poleward boundary of the region. See Boundaries & Regions IDs for details. **Example Value:** `daz_pol`

@childrens

`description`	Name and description of the region **Example Value:** `DAZ region`
`fluxes_array`	A list items as pairs of mlt/flux values

<fluxes_array> node

@attributes

`approximated`

If is set to `true`, then `fluxes_array` node will contain 500 items. When set to `false`, `fluxes_array` will contain 8 items of mlt/fluxes pairs.

Example Value: `true`

@childrens

`array_item`

A pair of mlt/flux values. See <array_item> node for details.

<array_item> node

@childrens

`mlt`

The value of MLT coordinate

Example Value: `1.50`

`flux`

The value of a flux

Example Value: `1.28`

JSON Response Format Entities

Contains an array of objects, associated with regions

Region object

@fields

`equatorward_boundary`

A equatorward boundary of the region. See Boundaries & Regions IDs for details.

Example Value: `daz_eq`

`poleward_boundary`

A poleward boundary of the region. See Boundaries & Regions IDs for details.

Example Value: `daz_pol`

`description`

Name and description of the region

Example Value: `DAZ region`

`fluxes`	An object, see Fluxes object description.

Fluxes object

@fields

`approximated`	If is set to `true`, then `data_array` field will contain 500 items. When set to `false`, the field will contain 8 items of mlt/flux pairs **Example Value:** `true`
`data_array`	An array of 8-by-2 or 500-by-2 items, containing pairs of mlt/flux values **Example Value:** `[[1.5,1.28],...]`

TEXT Response Format Entities

Contains 8 or 500 lines divided into 5 columns.

Columns description

1 column	MLT coordinate value
2 column	Flux values for `daz_eq/daz_pol` region
3 column	Flux values for `aop_eq/b4s` region
4 column	Flux values for `b4s/aop_pol` region
5 column	Flux values for `aop_pol/sdp_pol` region

Examples

request:

```
http://apm.pgia.ru/api/fluxes.xml?token=18faea489820&al=-100&dst=-10
```

response:

```
<hash>
<region equatorward_boundary="daz_eq" poleward_boundary="daz_pol">
   <description>DAZ region</description>
   <fluxes_array approximated="false">
      <array_item>
         <mlt>1.50</mlt>
         <flux>1.28</flux>
      </array_item>
      <array_item>
         <mlt>4.50</mlt>
         <flux>1.91</flux>
      </array_item>
      <array_item>
         <mlt>7.50</mlt>
         <flux>1.09</flux>
      </array_item>
      <array_item>
         <mlt>10.50</mlt>
         <flux>0.66</flux>
      </array_item>
      <array_item>
         <mlt>13.50</mlt>
         <flux>0.16</flux>
      </array_item>
      <array_item>
         <mlt>16.50</mlt>
         <flux>0.15</flux>
      </array_item>
      <array_item>
         <mlt>19.50</mlt>
         <flux>0.34</flux>
      </array_item>
      <array_item>
         <mlt>22.50</mlt>
         <flux>0.69</flux>
      </array_item>
</fluxes_array>
</region>
...
</hash>
```

request:

```
http://apm.pgia.ru/api/fluxes.text?token=18faea489820&al=-100&dst=-10
```

response:

```
1.50 1.28  1.77 2.42 0.22
4.50 1.91  1.75 0.74 0.11
7.50 1.09  1.40 1.40 0.54
10.50 0.66 1.01 1.01 0.56
13.50 0.16 0.84 0.84 0.49
16.50 0.15 0.98 0.98 0.47
19.50 0.34 0.59 1.93 0.20
22.50 0.69 1.79 4.13 0.28
```

Using api.pgia.ru/api

cURL

The most easy way to get model output as a file is using cURL tool. cUrl is available on most platforms and allows reading and saving web content to a file.

```
curl 'http://apm.pgia.ru/api/boundaries.text?token=18faea489820&al=-100&dst=-10'
> boundaries.dat
```

Output from boundaries model will be save to `boundaries.dat` file in text format.

Web browser

Just insert request to address bar of a web browser to get model output in the browser window. To save model output as a file, use 'Save as' option from the browser menu or use special utilites such as *cURL* or *wget* to save content to the file.

MATLAB/Octave

To use apm.pgia.ru/api service from *MATLAB* or *Octave*, use `urlread` function from MATLAB/Octave library, which allows to send request and read response from remote server.

You can also use our MATLAB/Octave interface for apm.pgia.ru/api service available from http://apm.pgia.ru/api/external/matlab_octave_interface.zip

Examples

using `urlread` function:

```
% response variable will contain boundaries model output in text format
% You need to parse response to get data in matrix format

response = urlread('http://apm.pgia.ru/api/boundaries.text?
token=18faea489820&al=-100&dst=-10')

response =

1.50 63.21 65.86 65.86 72.87 73.70 69.30
4.50 62.88 66.20 66.20 75.20 76.33 71.07
7.50 64.25 72.00 70.90 76.53 77.19 72.00
10.50 66.34 75.10 73.16 77.13 79.05 75.10
13.50 68.56 76.12 72.79 78.21 79.15 76.12
16.50 68.47 72.53 71.51 76.10 77.35 72.53
19.50 67.18 69.15 69.15 73.51 74.67 70.31
22.50 63.45 65.65 65.65 71.10 71.88 67.53
```

using MATLAB/Octave interface:

```
setapmtoken('18faea489820');
[mlt,coordinates] = apm('boundaries',-100,-10,'approximate',0)

mlt =

    1.5000
    4.5000
    7.5000
   10.5000
   13.5000
   16.5000
   19.5000
   22.5000


coordinates =

   63.2100   65.8600   65.8600   72.8700   73.7000   69.3000
   62.8800   66.2000   66.2000   75.2000   76.3300   71.0700
   64.2500   72.0000   70.9000   76.5300   77.1900   72.0000
   66.3400   75.1000   73.1600   77.1300   79.0500   75.1000
   68.5600   76.1200   72.7900   78.2100   79.1500   76.1200
   68.4700   72.5300   71.5100   76.1000   77.3500   72.5300
   67.1800   69.1500   69.1500   73.5100   74.6700   70.3100
   63.4500   65.6500   65.6500   71.1000   71.8800   67.5300
```