



Weather Company Data | Data Visualization - Weather Imagery | Common Usage Guide

Domain Portfolio: Weather Imagery | Domain: Common | API Name: Data Visualization - Common Usage Guide

Standard HTTP Cache-Control headers are used to define caching length. The TTL value is provided in the HTTP Header as an absolute time value using the “Expires” parameter.  
Example: “Expires: Fri, 12 Jul 2013 12:00:00 GMT”.

Overview

The ‘Tiler’, and ‘Featurizer’ products support data visualization and analytics.

- Tiler provides gridded raster data, typically in tiles of 256x256 pixels at various levels of detail; a client-side SDK can use this data to create weather image tiles
- Featurizer provides geometric vector data, either a line or a polygon, indicating where meteorological values cross a particular threshold; this data can facilitate statistical analysis

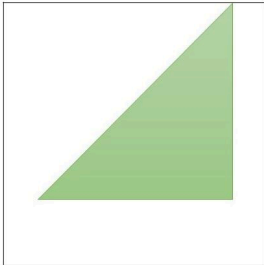
Using the Tiler and Featurizer products require a multi-step workflow to retrieve the necessary data for the specific product data request. Steps 2a, and 2b are dependent on which type you are requesting (i.e. Tiler Data or Featurizer Tile. Both step 2a, and step 2b require the ‘t’ parameter values as input into the ‘t’ parameter for the subsequent request (v2/tiler/data, and v2/featurizer/tile).

- **Step 1:** Get Tiler Info - Provides current dimensions ‘t’ and ‘rt’ parameter values on one or more products.
- **Step 2a:** Get Tiler Data - Provides tiles of meteorological data from one or more products.
- **Step 2b:** Get Featurizer Tile - Provides a polygon or line indicating when a product’s data has crossed a given threshold, in web-mercator projection.

Conceptual Visualization  
Gridded Data from Tiler

|     |     |     |     |     |     |     |     |
|-----|-----|-----|-----|-----|-----|-----|-----|
| 0.0 | 0.0 | 0.0 | 0.5 | 1.0 | 1.5 | 2.0 | 1.5 |
| 0.0 | 0.0 | 0.5 | 1.0 | 1.5 | 2.0 | 2.0 | 1.5 |
| 0.0 | 0.5 | 1.0 | 1.5 | 2.0 | 2.5 | 2.0 | 1.5 |
| 0.5 | 1.0 | 1.5 | 2.0 | 2.5 | 2.5 | 2.0 | 1.5 |
| 1.0 | 1.5 | 2.0 | 2.5 | 2.5 | 2.5 | 2.0 | 1.5 |
| 1.5 | 2.0 | 2.0 | 2.0 | 2.0 | 2.0 | 2.0 | 1.5 |
| 1.5 | 1.5 | 1.5 | 1.5 | 1.5 | 1.5 | 1.5 | 1.5 |
| 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 |

Conceptual Visualization  
Vector Data from Featurizer



Area with values ≥ 2.0

URL Construction

Step 1: Get Tiler Info

Tiler Info: **Required Parameters:** products, apiKey=yourApiKey || **Optional Parameters:** meta=true  
https://api.weather.com/v2/tiler/info?products=<productNumber>:<variableID>&apiKey=yourApiKeytiler

The [v2/tiler/info?] request response provides the ‘t’ parameter value required as input for the subsequent [v2/tiler/data?] request as well as a subsequent [v2/featurizer/tile?] request.  
If the product is an ‘Observation/Current Condition’ type then it will return a ‘t’ parameter value; if the product is a ‘Forecast’ type then it will return both a ‘t’ parameter value and ‘rt’ parameter value.

- Note: some exceptions may apply to the use of the ‘t’ parameter value and ‘rt’ parameter values; please see product specific details for all product specific required and optional parameters.

The meta=true additional information helps to provide available lod, x, and y parameters to use in subsequent requests.

https://api.weather.com/v2/tiler/info?products=17:VAR00198FROM25501heightaboveground&meta=true&apiKey=yourApiKey

Step 2a: Get Tiler Data

Tiler Data - Observations: **Required Parameters:** products, t, lod, x, y, apiKey=yourApiKey  
https://api.weather.com/v2/tiler/data?products=<productNumber>:<variableID>&t=<t>&lod=<lod>&x=<x>&y=<y>&apiKey=yourApiKey

|   |
|---|
| https://api.weather.com/v2/tiler/data?products=17:VAR00198FROM25501heightaboveground&t=1474300200000&lod=1&x=0&y=0&apiKey=yourApiKey  |
| <b>Tiler Data - Forecast:</b> <b>Required Parameters:</b> products, rt, t, lod, x, y, apiKey=yourApiKey<br>https://api.weather.com/v2/tiler/data?products=<productNumber>:<variableID>&rt=<rt>&t=<t>&lod=<lod>&x=<x>&y=<y>&apiKey=yourApiKey  |
| https://api.weather.com/v2/tiler/data?products=49:Temperatureheightaboveground&rt=1474375500000&t=1474400700000&lod=1&x=0&y=0&apiKey=yourApiKey   |
| Step 2b: Get Featurizer Tile  |
| <b>Featurizer Tile - Observations:</b> <b>Required Parameters:</b> product, t, lod, x, y, apiKey=yourApiKey    <b>Optional Parameters:</b> threshold<br>https://api.weather.com/v2/featurizer/tile?product=<productNumber>:<variableID>&t=<t>&lod=<lod>&x=<x>&y=<y>&apiKey=yourApiKey         |
| https://api.weather.com/v2/featurizer/tile?product=17:VAR00198FROM25501heightaboveground&t=1474300200000&lod=4&x=2&y=4&threshold=280&apiKey=yourApiKey  |
| <b>Featurizer Feature (Native Resolution) - Observations:</b> <b>Required Parameters:</b> product, t, apiKey=yourApiKey    <b>Optional Parameters:</b> threshold<br>https://api.weather.com/v2/featurizer/feature?product=<productNumber>:<variableID>&t=<t>&apiKey=yourApiKey                |
| https://api.weather.com/v2/featurizer/feature?product=17:VAR00198FROM25501heightaboveground&t=1474300200000&threshold=280&apiKey=yourApiKey   |
| <b>Featurizer Tile - Forecast:</b> <b>Required Parameters:</b> product, rt, t, lod, x, y, apiKey=yourApiKey    <b>Optional Parameters:</b> threshold<br>https://api.weather.com/v2/featurizer/tile?product=<productNumber>:<variableID>&rt=<rt>&t=<t>&lod=<lod>&x=<x>&y=<y>&apiKey=yourApiKey |
| https://api.weather.com/v2/featurizer/tile?product=49:Temperatureheightaboveground&rt=1474375500000&t=1474400700000&lod=1&x=0&y=0&threshold=280&apiKey=yourApiKey   |
| <b>Featurizer Feature (Native Resolution) - Forecast:</b> <b>Required Parameters:</b> product, rt, t, apiKey=yourApiKey    <b>Optional Parameters:</b> threshold<br>https://api.weather.com/v2/featurizer/feature?product=<productNumber>:<variableID>&rt=<rt>&t=<t>&apiKey=yourApiKey        |
| https://api.weather.com/v2/featurizer/feature?product=49:Temperatureheightaboveground&rt=1474375500000&t=1474400700000&threshold=280&apiKey=yourApiKey  |

Using Tiler Info & Parameters

The Tiler Info API is used to retrieve current details about one or more products. By calling the v2/tiler/info API with the optional parameter ‘meta=true’, the details about each requested product’s native projection and valid Web Mercator tiles are delivered in the response.

- **NOTE:** All API requests for this endpoint are required to include the following header, prompting the system to compress the response: “Accept-Encoding: gzip”.
- The ‘products’ parameter can accept up to fifteen (15) comma-delimited values:
  - Example: products=1:Precipitationtypesurface,6:VAR01515FROM25011surface,19,20
  - Sample: https://api.weather.com/v2/tiler/info?products=1:Precipitationtypesurface,6:VAR01515FROM25011surface,19,20&apiKey=yourApiKey
- To retrieve the details on exactly one type of data from a product number, include the product number and the variable ID separated by a colon:
  - Example: products=303:tdprob
  - Sample: https://api.weather.com/v2/tiler/info?products=303:tdprob&apiKey=yourApiKey

- To retrieve the details on all data associated with a product number, include this number while omitting any variable IDs:
  - Example: products=303
  - Sample: <https://api.weather.com/v2/tiler/info?products=303&apiKey=yourApiKey>

### Using Tiler Data & Parameters

The Tiler Data API retrieves tiles of meteorological data from one or more products.

- **NOTE:** All API requests for this endpoint are required to include the following header, prompting the system to compress the response: “Accept-Encoding: gzip”.
- products: the ‘products’ parameter for this endpoint must include both the product number and the variable ID, separated by a colon:
  - Example: products=300:PrecipIntensity
  - Sample: <https://api.weather.com/v2/tiler/data?products=300:PrecipIntensity&t=1451606400000&lod=4&x=4&y=6&apiKey=yourApiKey>
- dimension: the dimensions vary by API product, and valid dimensions can be found by calling /tiler/info
- lod, x, y: the combination of these tile parameters must correspond to a tile within the product’s tileset, found by calling /tiler/info?meta=true
- For the product’s native resolution, set lod to -1, and for the Web Mercator projection, use any valid lod value where lod ≥ 0

### Using Tiler Byte & Parameters

The Tiler Byte API retrieves a byteset object that contains information regarding the values contained in each tile at all levels of detail for a single time value. Data includes minimum value and maximum value for the tile and whether there are empty cells (NaN values) in the tile.

- **NOTE:** All API requests for this endpoint are required to include the following header, prompting the system to compress the response: “Accept-Encoding: gzip”.
- products: the ‘products’ parameter for this endpoint must include both the product number and the variable ID, separated by a colon:
  - Example: products=300:PrecipIntensity
  - Sample: <https://api.weather.com/v2/tiler/byte?products=300:PrecipIntensity&t=1451606400000&raw=false&apiKey=yourApiKey>
- dimension: the dimensions vary by API product, and valid dimensions can be found by calling /tiler/info
- If raw is true, then the native layer byteset is returned. If raw is not given or is false, then the Web Mercator byteset for the pyramid is returned.

The entire pyramid (all LODs) of the tiles is returned concatenated in ascending raster-order (i.e. info for lod=0,x=0,y=0 (0,0,0) followed by lod=1,x=0,y=0 (1,0,0), (1,0,1), (1,1,0), (1,1,1), (2,0,0) and so on. Each tile is described by a record of up to 3 floating point values. If the first 4 bytes make up a floating point value of not-a-number, then the record is only 4 bytes long and the tile has no data. Calling Tiler API for such tiles should result in a 400 response. If the first 4 bytes do not make up a not-a-number value, then the record is 9 bytes long and should be decoded as follows.

1. The first 4 bytes make up the floating point value for the minimum value in the tile.
2. The second 4 bytes make up a floating point value that is the maximum value in the tile.
3. The last byte is a boolean that describes if any cells in the tile are empty, that is, the cell value is not-a-number.

### Using Tiler Coverage & Parameters

The Tiler Coverage API retrieves a bitset object that contains a single bit for each tile at all levels of detail for a single time value. This bit flag denotes whether the given tile exists at that LOD and x, y.

- **NOTE:** All API requests for this endpoint are required to include the following header, prompting the system to compress the response: “Accept-Encoding: gzip”.
- products: the ‘products’ parameter for this endpoint must include both the product number and the variable ID, separated by a colon:
  - Example: products=300:PrecipIntensity
  - Sample: <https://api.weather.com/v2/tiler/coverage?products=300:PrecipIntensity&t=1451606400000&apiKey=yourApiKey>
- dimension: the dimensions vary by API product, and valid dimensions can be found by calling /tiler/info

- If *raw* is true, then the native layer bitset is returned. If raw is not given or is false, then the Web Mercator bitset for the pyramid is returned. **Note** that coverage data is typically not produced for the native layer. The API allows such requests, but a response will often be an error.

The entire pyramid (all LODs) of the tiles is returned concatenated in ascending raster-order (i.e. info for lod=0,x=0,y=0 (0,0,0) followed by lod=1,x=0,y=0 (1,0,0), (1,0,1), (1,1,0), (1,1,1), (2,0,0) and so on. Each tile is described by a single bit packed in reverse order into bytes.

Using Tiler Extras Data & Parameters

The Tiler Extras Data API returns the same data as the /tiler/data call discussed above except that it allows the user to choose to receive data only over the oceans or only over land. This API may also be used to provide interpolated tile data at LODs otherwise not supported for a given product. See “Geographic Coverage” for more information on LODs available.

- **NOTE:** All API requests for this endpoint are required to include the following header, prompting the system to compress the response: “Accept-Encoding: gzip”.
- products: the ‘products’ parameter for this endpoint must include both the product number and the variable ID, separated by a colon:
  - Example: products=300:PrecipIntensity
  - Sample: <https://api.weather.com/v2/tiler/extras/data?products=300:PrecipIntensity&stencil=true&interpolation=bicubic&t=1451606400000&lod=4&x=4&y=6&apiKey=yourApiKey>
- dimension: the dimensions vary by API product, and valid dimensions can be found by calling /tiler/info
- lod, x, y: the combination of these tile parameters must correspond to a tile within the product’s tileset, found by calling /tiler/info?meta=true
- Optional Parameters: stencil, interpolation
- This API should only be used for the Web Mercator projection, i.e. lod value ≥ 0

Using Featurizer Tile & Parameters

The Featurizer Tile API retrieves a polygon or line tracing a particular threshold for a product’s data, for a single tile in Web Mercator projection.

- **NOTE:** All API requests for the first endpoint are required to include the following header, prompting the system to compress the response: “Accept-Encoding: gzip”.
- product: the ‘product’ parameter must include both the product number and the variable ID, separated by a colon:
  - Example: product=300:PrecipIntensity
  - Sample: <https://api.weather.com/v2/featurizer/tile?product=300:PrecipIntensity&t=1451606400000&lod=4&x=4&y=6&apiKey=yourApiKey>
- <dimensions>: dimensions vary by API product, and valid dimensions can be found by calling /tiler/info
- lod, x, y: the combination of these tile parameters must correspond to a tile within the product’s tileset, found by calling /tiler/info?meta=true
- The lod parameter must indicate the Web Mercator projection, where lod ≥ 0
  - To request the product’s native resolution, use the featurizer/feature endpoints
- Optional Parameters: operation, threshold, replacement, geometryType, crs, bbox

The API returns a geographic feature, namely a polygon or line as defined by the geometryType parameter. The API examines each pixel in the requested tile against the operation and threshold values, and the resulting feature corresponds to the area where the comparison evaluates to “true.”

The type of the response depends on the specific endpoint that is called, as documented below.

| <b>REQUIRED: All API responses must be compressed.</b> The kmz, XZipped shp, and mvt file formats are compressed by definition; API requests for JSON objects, GeoJSON objects, and arrays of primitive data types <b>must</b> include the following HTTP header: “Accept-Encoding: gzip”. |          |                            |
|--|----------|----------------------------|
| Endpoint   | Response | MIME Content-Type          |
| /tile  | GeoJSON  | application/json (gzipped) |

|           |              |                                  |
|-----------|--------------|----------------------------------|
| /tile:kmz | .kmz         | application/vnd.google-earth.kmz |
| /tile:shp | XZipped .shp | application/x-zipped-shp         |
| /tile:mvt | .mvt         | application/x-vector-tile        |

Using Featurizer Feature & Parameters

The Featurizer Feature API retrieves a polygon or line tracing a particular threshold for a product's data, for all data in the product's native projection.

- products: for this endpoint, this parameter must include both the product number and the variable ID, separated by a colon:
  - Example: products=300:PrecipIntensity
  - Sample: https://api.weather.com/v2/featurizer/feature?product=300:PrecipIntensity&t=1451606400000&apiKey=yourApiKey
- <dimensions>: dimensions vary by API product, and valid dimensions can be found by calling /tiler/info
- Optional Parameters: operation, threshold, replacement, geometryType, crs, bbox

The API returns a geographic feature, namely a polygon or line as defined by the geometryType parameter. The API examines each pixel in the product's native projection against the operation and threshold values, and the resulting feature corresponds to the area where the comparison evaluates to “true.”

The type of the response depends on the specific endpoint that is called, as documented below.

| <b>REQUIRED: All API responses must be compressed.</b> The .kmz, XZipped.shp, and .mvt file formats are compressed by definition; API requests for JSON objects, GeoJSON objects, and arrays of primitive data types <b>must</b> include the following HTTP header: “Accept-Encoding: gzip”. |              |                                  |
|--|--------------|----------------------------------|
| Endpoint   | Response     | MIME Content-Type                |
| /feature   | GeoJSON      | application/json (gzipped)       |
| /feature:kmz   | .kmz         | application/vnd.google-earth.kmz |
| /feature:shp   | XZipped .shp | application/x-zipped-shp         |

Using Featurizer Isoband & Parameters

The Featurizer Isoband API retrieves polygons containing all data between a lower and upper threshold, for all data in the product's native projection.

- **NOTE:** All API requests for the first endpoint are required to include the following header, prompting the system to compress the response: “Accept-Encoding: gzip”.
- product: the ‘product’ parameter must include both the product number and the variable ID, separated by a colon:
  - Example: product=6:VAR01515FROM25011surface
  - Sample: https://api.weather.com/v2/featurizer/isoband?product=6:VAR01515FROM25011surface&threshold=35.0:40.0&t=1451606400000&apiKey=yourApiKey
- <dimensions>: dimensions vary by API product, and valid dimensions can be found by calling /tiler/info
- *threshold*: this should be two floating point values for lower and upper bounds, separated by a colon, e.g. threshold=35.0:40.0
- Optional Parameters: replacement, crs, bbox, format
- Unsupported parameters: lod, x, y

Query Parameter Elements & Definitions

| Parameter Name                                    | Description   | Is Required | Sample                             |
|---|---|-------------|------------------------------------|
| apiKey  | Client's API Key  | Required    | apiKey= <a href="#">yourApiKey</a> |
| lod   | <p>Level of Detail (LOD), also known as the zoom level. The metadata found by calling /tiler/info provides more details about the valid ranges of lod, x, and y for a particular product.</p> <ul style="list-style-type: none"><li>The product's native LOD can be invoked with lod=-1.</li><li>A global image tile can be invoked for the lowest LOD with lod=0.</li></ul> <p>The single image tile at lod=0 encompasses the entire globe in Web Mercator projection, and each subsequent LOD doubles:</p> <ul style="list-style-type: none"><li>Zoom (level of detail)</li><li>Range of possible x values</li><li>Range of possible y values</li></ul> | Required    | lod=4                              |
| rt  | "Run time" Timestamp, in milliseconds - Required parameter for 'Forecast' type products.  | Required    | rt=1451649600000                   |
| t   | Timestamp, in milliseconds  | Required    | t=1451649600000                    |
| x   | Horizontal tile position  | Required    | x=4                                |
| y   | Vertical tile position  | Required    | y=6                                |
| Tiler APIs - Parameters Unique To Tiler           |   |             |                                    |
| Parameter Name                                    | Description   | Is Required | Sample                             |
| products  | Products (plural) used in the tiler APIs. Specific API products, where each is typically referenced with a colon-separated Product Number and Variable ID   | Required    | products=303:tdprob                |
| meta  | <p>Request to display metadata, for /tiler/info</p> <p>The metadata found by calling /tiler/info provides more details about the valid ranges of lod, x, and y for a particular product. For example, a geographically limited product may further restrict the valid ranges for x and y to include only those tiles that cover the product's region.</p>   | Optional    | meta=true                          |
| Featurizer APIs - Parameters Unique To Featurizer |   |             |                                    |
| Parameter Name                                    | Description   | Is Required | Sample                             |
| product   | Product (singular) Specific API product requested in the featurizer APIs  | Required    | product=303:tdprob                 |
| bbox  | Comma-delimited string of decimal pair of x-y coordinates, restricting the area subject to analysis by Featurizer.  | Optional    | bbox=-129.4,21.37,-64.56,50.35     |

|              |   |          |   |
|--------------|---|----------|---|
|              | <p>These coordinates set the <b>lower-left (southwest)</b> and <b>upper-right (northeast)</b> corners of the rectangular bounding box that Featurizer will examine, and its values must be (<b>long,lat,long,lat</b>), with the lower-left corner defined first.</p> <p><b>NOTE</b> that the order of the coordinates is (x,y) -- longitude, latitude -- which is consistent with the GeoJSON convention.</p>   |          |   |
| crs          | <p>Type of coordinate reference system used to define the projection of the GeoJSON geometry object returned by Featurizer</p> <p><b>Default Value:</b> EPSG:4326</p> <p><b>Accepted Values:</b> Any EPSG code, such as EPSG:4326 or EPSG:3857</p> <p>The default crs value of EPSG:4326 corresponds to WGS 84, a standard established in 1984 as part of the the World Geodetic System developed by the US Department of Defense.</p> <p>Information on EPSG codes can be found at <a href="http://spatialreference.org/">http://spatialreference.org/</a>.</p>  | Optional | crs=EPSG:4326   |
| geometryType | <p>Type of GeoJSON object to be returned by Featurizer.</p> <p><b>Default Value:</b> linestring</p> <p><b>Accepted Values:</b> linestring, polygon</p> <p>The GeoJSON polygon object is guaranteed to be a closed object, but the linestring object may be open. If the user is in doubt about which geometryType to use, we recommend explicitly using the polygon type, <b>geometryType=polygon</b>.</p> <p>Information on the GeoJSON format can be found at <a href="http://geojson.org">http://geojson.org</a></p>   | Optional | geometryType=polygon  |
| operation    | <p>Mathematical operator used with threshold to determine which data points are included within the Featurizer line or polygon.</p> <p><b>Default Value:</b> greaterThan</p> <p><b>Accepted Values:</b> lessThan, lessThanOrEqualTo, greaterThan, greaterThanOrEqualTo</p> <p>For example, if data records current temperatures in degrees Fahrenheit, the user can retrieve a linestring or polygon containing freezing temperatures by submitting an operation value of “lessThanOrEqualTo” and a threshold value of 32.0.</p>  | Optional | operation=lessThanOrEqualTo                                       |
| replacement  | <p>Value to be used with Featurizer, when a pixel has no data.</p> <p><b>Default Value:</b> 0.0</p> <p>Carefully chosen with the operation and threshold parameters in mind, the replacement parameter can cause missing data to be included within or excluded from the polygon or linestring returned by Featurizer.</p> <p>For example, if data records current temperatures in degrees Fahrenheit, and if the user retrieves a polygon or linestring containing freezing temperatures, the user can exclude missing data by setting a replacement value of 40.0 or any other value above the freezing mark.</p> | Optional | replacement=40.0  |
| threshold    | <p>Value used with operation to determine which data points are included within the Featurizer linestring or polygon.</p> <p><b>Default Value:</b> 25.0</p> <p>Float value used with the operation value to determine which data points are included within the polygon or linestring returned by Featurizer. The threshold’s units are determined by the product being evaluated.</p>  | Optional | <p>threshold=32.0</p> <p>For Isoband:<br/>threshold=25.0:32.0</p> |



|                                     |   |          |                       |
|-------------------------------------|---|----------|-----------------------|
|                                     | For the /featurizer/isoband call, this parameter should be two floating point values, separated by a colon, to determine the lower and upper bounds of the values to include in the generated polygon(s), respectively.   |          |                       |
| stencil<br>(extras/data only)       | <b>Accepted Values:</b> true, false, or compliment.<br><b>Default Value:</b> false<br>If true then the oceans will be removed. If false, no stenciling is done (same as /tiler/data). If compliment, then the land will be removed.   | Optional | stencil=true          |
| interpolation<br>(extras/data only) | Which interpolation algorithm to use when providing data at higher LODs than available in a product's /tiler/info data.<br><b>Default Value:</b> bicubic<br><b>Accepted Values:</b><br><br>1. bicubic - Custom bicubic interpolation method.<br>2. typed_bicubic - A proprietary use of bicubic interpolation to choose a category.<br>3. bilinear - Custom bilinear interpolation.<br>4. nearest - Custom nearest neighbor interpolation.<br>5. influence - An experimental interpolation method.<br>6. bicubic2 - GeoTrellis bicubic implementation.<br>7. bilinear2 - GeoTrellis bilinear implementation.<br>8. nearest2 - GeoTrellis nearest neighbor implementation.<br>9. mode2 - GeoTrellis modal resampling.<br>10. ptypebicubic - A proprietary use of bicubic interpolation to choose a category. | Optional | interpolation=bicubic |
| format<br>(isoband only)            | Return a response in this format. This may be <i>json</i> for GeoJSON, <i>kml</i> for KML, <i>kmz</i> for a zipped KML, <i>shp</i> for a Shapefile or <i>mvt</i> for a Mapbox Vector Tile.<br><b>Default Value:</b> json  | Optional | format=shp            |

Examples Using Select Parameters for Featurizer

For sample request: <https://api.weather.com/v2/featurizer/tile?product=34:VAR0191200FROM25501heightaboveground&t=1451606400000&lod=4&x=4&y=6&operation=greaterThanOrEqual&threshold=6.0&replacement=3.0&geometryType=polygon&apiKey=yourApiKey>

| Goal for Geometry Object Returned by Featurizer  | Value for <b>operation</b> | Value for <b>threshold</b> | Value for <b>replacement</b> |
|--|----------------------------|----------------------------|------------------------------|
| Product data: temperature in degrees Fahrenheit  |                            |                            |                              |
| Goal area with temperatures strictly below freezing, <i>excluding</i> missing data   | lessThan                   | 32.0                       | 32.0 or higher (ex. 40.0)    |
| Goal area with triple-digit temperatures, <i>including</i> missing data  | greaterThanOrEqual         | 100.0                      | 100.0 or higher (ex.101.0)   |
| Product data: UV index measuring ultraviolet (UV) exposure, where 0-2 is Low, 3-5 is Moderate, 6-7 is High, 8-10 is Very High, and 11 is Extreme |                            |                            |                              |



|   |                    |     |                        |
|---|--------------------|-----|------------------------|
| Goal area with UV index of “High” or greater, <i>excluding</i> missing data                               | greaterThan        | 5.0 | 5.0 or lower (ex. 3.0) |
| <b>Alternate approach:</b><br>Goal area with UV index of “High” or greater, <i>excluding</i> missing data | greaterThanOrEqual | 6.0 | 5.0 or lower (ex. 3.0) |

Data Elements & Definitions

Note: The table below does not necessarily represent the sort order of the API response.

| Field Name                | Description  | Type       | Range                 | Sample  |
|---------------------------|--|------------|-----------------------|---|
| Tiler Info: v2/tiler/info |  |            |                       |   |
| <productNumber>           | Object containing a child element for one or more variable ID's associated with the product number   | object     |                       | 303: { tdprob: { }, mhrprob: { }, ... }   |
| <variableID>              | Object containing details on the data that can be retrieved from a given product   | object     |                       | tdprob: { dimensions: [ ], meta: { } }  |
| dimensions                | Array of unnamed JSON objects, one for each set of current dimensions for the product number and variable ID, typically timestamps. The dimensions include the 't' and 'rt' parameter values for the products requested in the 'tiler/info' request. | [ object ] |                       | dimensions: [ {rt: [ ], t: [ ]}, {rt: [ ], t: [ ]}, ... ]   |
| t                         | Array of timestamps, in milliseconds, where each is generally the start time for the period covered by the product's data  | [epoch]    |                       | rt: ["1471824000000"]   |
| rt                        | Array of "run time" timestamp, in milliseconds, where each is generally when a forecast was generated  | [epoch]    |                       | t: ["1473120000000", "1473033600000", ... ]   |
| meta                      | Object containing metadata for the given product number and variable ID  | object     |                       | meta: { data-href, data-aggregate-href, ... }   |
| data-href                 | URL for the /tiler/data API call   | string     | Request API URL       | data-href:<br>"https://api.weather.com/v2/tiler/data?products=303:tdprob&rt={rt}&t={t}&lod={lod}&x={x}&y={y}&apiKey={apiKey}"                     |
| data-aggregate-href       | URL for the /tiler/data-aggregate API call   | string     | Request API URL       | data-aggregate-href:<br>"https://api.weather.com/v2/tiler/data-aggregate?products=303:tdprob&rt={rt}&t={t}&lod={lod}&x={x}&y={y}&apiKey={apiKey}" |
| cover-href                | URL for the /tiler/coverage API call   | string     | Request API URL       | cover-href:<br>"https://api.weather.com/v2/tiler/coverage?products=303:tdprob&rt={rt}&t={t}&apiKey={apiKey}"                                      |
| label                     | Variable ID  | string     | Any valid variable ID | label: "tdprob"   |
| missingValue              | Placeholder, equivalent to NaN (Not a Number) and useful for working with tiler/coverage   | string     |                       | missingValue: "v=-34  v=-35"  |
| attributes                | Object containing various metadata values, typically from the ingest source  | object     |                       | attributes: {Grib2_Level_Type, Grib2_Parameter, .... }  |

|             |  |          |  |  |
|-------------|--|----------|--|--|
| dimensions  | Dimensions for the data associated with the product number and variable ID   | [string] |  | dimensions: ["rt", "t", "geox", "geoy"]          |
| dimension   | Details for the entries listed in the dimensions array   | object   |  | dimension: { rt: {}, t: {}, geox: {}, geoy: {} } |
| <dimension> | Details for a single dimension   | object   |  | rt: {id, primary, type, unit}                    |
| id          | Dimension ID: t for the timestamp of the observation or forecast, rt for the forecast run time, geox for the x-axis location, geoy for the y-axis location, elevation for the geographic height (elsewhere known as the z-axis location), ensemble for the runtime iteration | string   | t, rt, geox, geoy, elevation, ensemble | id: "rt"   |
| primary     | Indicator for the outermost dimension  | boolean  | true, false                            | primary: true                                    |
| type        | Dimension's data type  | string   |  | type: "long"                                     |
| unit        | Dimension's unit of measurement  | string   |  | unit: "time in epoch"                            |
| tilesets    | List of tilesets for the data associated with the product number and variable ID   | [string] | Native, Web Mercator                   | tilesets: ["Native", "Web Mercator"]             |
| tileset     | Details for the entries listed in the tilesets array   | object   |  | tileset: { Native: {}, Web Mercator: {} }        |
| <tileset>   | Details for a single tileset   | object   |  | Native: { name, crs, ... }                       |
| name        | Tileset ID, matching an element in the tilesets array and tileset object   | string   | Native, Web Mercator                   | name: "Web Mercator"                             |
| crs         | Coordinate Reference System, such as “EPSG:3857” for Web Mercator  | string   | Any EPSG code or Well Known Text (WKT) | crs: "EPSG:4326"                                 |
| envelope    | Bounding box defining the region covered by the tileset  | object   |  | envelope: { minX, minY, maxX, maxY}              |
| minX        | Longitude value for the envelope bounding box: degrees or some other measure, depending on the crs value   | decimal  |  | minX: "-100.0"                                   |
| minY        | Latitude value for the envelope bounding box: degrees or some other measure, depending on the crs value  | decimal  |  | minY: "0.0"                                      |
| maxX        | Longitude value for the envelope bounding box: degrees or some other measure, depending on the crs value   | decimal  |  | maxX: "-5.0"                                     |
| maxY        | Latitude value for the envelope bounding box: degrees or some other measure, depending on the crs value  | decimal  |  | maxY: "50.0"                                     |
| pixel       | Object defining the dimensions, in pixels, for the data covered by the tileset   | object   |  | pixel: { w: 8192, h: 4096 }                      |
| tile        | Object defining the dimensions, in pixels, for a single data tile produced by the tile set   | object   |  | tile: { w: 256, h: 256 }                         |
| w           | Width for the pixel object or tile object  | integer  | >0                                     | w: 8192  |
| h           | Height for the pixel object or tile object   | integer  | >0                                     | h: 4096  |

|                 |  |            |         |  |
|-----------------|--|------------|---------|--|
| tiles           | Array of unnamed JSON objects defining the valid set of data tiles at each lod level of detail, for each tileset   | [ object ] |         | tiles: [ {x, y, width, height, lod}, ... ] |
| x               | Zero-based horizontal location of the first tile at a particular lod value, starting at the far left (west)  | integer    | >=0     | x: 0                                       |
| y               | Zero-based vertical location of the first tile at a particular lod value, starting at the top (north)  | integer    | >=0     | y: 1                                       |
| width           | Number of tiles across the horizontal length of the data set at a particular lod value   | integer    | >0      | width: 2                                   |
| height          | Number of tiles down the vertical length of the data set at a particular lod value   | integer    | >0      | height: 1                                  |
| lod             | <p>Level of detail, where -1 indicates the Native Resolution, and 0 and above indicates Web Mercator, where each subsequent LOD value doubles a) the zoom, b) the range of possible x values, and c) the range of possible y values. These ranges are defined below.</p> <p>For any lod value z, where <math>z \geq 0</math>, <math>0 \leq x \leq (2^z)-1</math>, and <math>0 \leq y \leq (2^z)-1</math></p> | integer    | -1 to 9 | lod: 2                                     |
| ttl             | Time to live; the minimum length of time that the data will be available in the system, from its publication to its expected roll-off in seconds   | integer    | >0      | ttl: 7200                                  |
| updateFrequency | Frequency in which new data will be provided in seconds  | integer    | >0      | updateFrequency: 900                       |

Appendix: Tiler Info meta=true

By calling the v2/tiler/info API with the optional parameter ‘meta=true’, the details about each requested products native projection and valid Web Mercator tiles are delivered in the response.

| Example  | Section              | Notes   |
|--|----------------------|---|
| <pre>tilesets:[   "Native",   "Web Mercator" ], tileset:{   Native:{     name: "Native",     crs:"EPSG:4326",     envelope:{       minX:"-39.3695068359375",       minY:"0.005500791594386101",       maxX:"50.6195068359375",       maxY:"89.99449920654297"     },     pixel:{       w:8192,</pre> | tilesets             | This section has two tilesets, one for each projection, “Native” and “Web Mercator.”  |
|  | tileset.Native       | <p>The native projection for this particular incoming data was given in the World Geodetic System developed by the US Department of Defense (officially identified as <b>EPSG:4326</b>, as indicated in the <b>crs</b> field).</p> <p>The data was limited to an <b>envelope</b> in the Northern Hemisphere, encompassing the prime meridian:</p> <ul style="list-style-type: none"><li>Longitude, west to east, approximate: 39.4°W to 50.6°E</li><li>Latitude, south to north, approximate: 0° to 90.0°N</li></ul> <p>(More details about the approximate coverage of Global Radar Tile 2 can be found in <b>Appendix B.</b>)</p> |
|  | tileset.Native.pixel | <p>The original data was given in a grid of pixels, 8192 pixels on each side, and each pixel contains a single point of data.</p> <p>Not all native-resolution grids will be squares, with the same height and width.</p>   |

|   |                             |  |
|---|-----------------------------|--|
| <pre>h:8192 }, tile:{   w:256,   h:256 }, tiles:[   {     x:0,     y:0,     width:32,     height:32,     lod:-1   } ] }, Web Mercator:{   name:"Web Mercator",   crs:"EPSG:3857",   tile:{     w:256,     h:256   },   tiles:[     {       x:0,       y:0,       width:1,       height:1,       lod:0     },     {       x:0,       y:0,       width:2,       height:1,       lod:1     },     {       x:1,       y:0,       width:2,       height:2,       lod:2     }   ] }</pre> | <b>tileset.Native.tile</b>  | <p>The original data was split into tiles, each tile containing the standard 256x256 pixels.</p>   |
|   | <b>tileset.Native.tiles</b> | <p>The original data was divided into tiles of 256x256 pixels, and these tiles were mapped onto a grid, 32 tiles on each side. The grid is assigned a level of detail indicating that the grid is in the native resolution, <b>lod=-1</b>.</p> <ul style="list-style-type: none"><li>width: there are 32 tiles, each tile is 256 pixels wide, and 32 x 256 = 8192, matching the data's original width</li><li>height: there are 32 tiles, each tile is 256 pixels tall, and 32 x 256 = 8192, matching the data's original height</li></ul> <p>As with the original grid, the native-resolution tiles will not always form a square, with the same height and width.</p> <p>The original data will not always fit exactly into a whole number of 256x256 pixels, so there can be tiles with partial data.</p> <p>For example, consider this separate set of <b>Native</b> details from the following API call.</p> <ul style="list-style-type: none"><li><a href="https://api.weather.com/v2/tiler/info?products=125&amp;meta=true&amp;apiKey=yourApiKey">https://api.weather.com/v2/tiler/info?products=125&amp;meta=true&amp;apiKey=yourApiKey</a></li></ul> <pre>pixel:{   w:3000,   h:2008 }, tile:{   w:256,   h:256 }, tiles:[   {     x:0,     y:0,     width:12,     height:8,     lod:-1   } ]</pre> <p>The original grid of 3000x2008 pixels is split in tiles of 256x256 pixels, for a grid of 12x8 tiles.</p> <ul style="list-style-type: none"><li>width: there are 12 tiles, each tile is 256 pixels wide, and 12 x 256 = 3072, slightly more than the data's original width</li><li>height: there are 8 tiles, each tile is 256 pixels tall, and 8 x 256 = 2048, slightly more than the data's original height</li></ul> |
|   | <b>tileset.Web Mercator</b> | <p>Taking the standard approach for all products provided through Tiler and Featurizer, the system converts the incoming data from its native projection to the Web Mercator projection (officially identified as <b>EPSG:3857</b>, as indicated in the <b>crs</b> field).</p> <p>Even when the data is limited to a smaller area -- as is the case here -- the tile scheme for this projection encompasses almost the</p>   |

| <pre>}, {   x:3,   y:0,   width:3,   height:4,   lod:3 }, {   x:6,   y:0,   width:5,   height:8,   lod:4 }, {   x:12,   y:0,   width:9,   height:16,   lod:5 }, {   x:25,   y:0,   width:16,   height:32,   lod:6 }, {   x:50,   y:0,   width:32,   height:64,   lod:7 } ]</pre> |  | <p>entire globe, limited only by the fact that Web Mercator projects both poles at infinity.</p> <ul style="list-style-type: none"><li>• Longitude, west to east: 180° to 180°</li><li>• Latitude, south to north, approximate: 85.1°S to 85.1°N</li></ul> <p>Because the Web Mercator tiles are consistent, the <b>envelope</b> information is neither needed nor explicitly provided.</p> <p>The <b>pixel</b> information is also omitted for the Web Mercator projection, but it can be calculated for each <b>lod</b> level of detail.</p>   |   |  |  |   |   |   |   |   |   |        |        |   |   |        |        |        |   |        |        |        |   |         |         |        |   |         |          |         |   |         |          |         |   |          |          |
|--|--|--|---|--|--|---|---|---|---|---|---|--------|--------|---|---|--------|--------|--------|---|--------|--------|--------|---|---------|---------|--------|---|---------|----------|---------|---|---------|----------|---------|---|----------|----------|
|  | <a href="#">tileset.Web Mercator.tile</a>  | For each <b>lod</b> value, the data is split into tiles, each tile containing the standard 256x256 pixels.   |   |  |  |   |   |   |   |   |   |        |        |   |   |        |        |        |   |        |        |        |   |         |         |        |   |         |          |         |   |         |          |         |   |          |          |
|  | <a href="#">tileset.Web Mercator.tiles</a>   | <p>The system converts the data from its native projection into the Web Mercator projection at one or more <b>lod</b> value, and the set of valid tiles is defined for each level of detail.</p> <p>For each <b>lod</b> value, the set of valid tiles will contain the <b>envelope</b> of data from the native projection. A valid tile may contain partial data -- and a tile may not always contain data for every timestamp -- but all data will be limited to valid tiles.</p> <p>For each level of detail, the possible x and y coordinates range between 0 and 2<sup>z</sup>-1, where z = <b>lod</b>.</p> <ul style="list-style-type: none"><li>• Along the x-axis, the first valid tile is found at <b>x</b>, and the since the <b>width</b> includes this tile, the last tile is at <b>x + <i>width</i> - 1</b>.</li><li>• Along the y-axis, the first valid tile is found at <b>y</b>, and the since the <b>height</b> includes this tile, the last tile is at <b>y + <i>height</i> - 1</b>.</li></ul> <p>The following table summarizes this particular product’s valid tile locations at each <b>lod</b> value for the Web Mercator projection.</p> <table><tr><th><b>lod</b></th><th>Maximum range of <b>x</b> and <b>y</b> values<br/><i>0 ≤ x,y ≤ 2<sup>z</sup>-1, where z = lod</i></th><th>Valid <b>x</b> Values for This Product<br/><i>From x to x+width-1</i></th><th>Valid <b>y</b> Values for This Product<br/><i>From y to y+height-1</i></th></tr><tr><td>0</td><td>0</td><td>0</td><td>0</td></tr><tr><td>1</td><td>0 to 1</td><td>0 to 1</td><td>0</td></tr><tr><td>2</td><td>0 to 3</td><td>1 to 2</td><td>0 to 1</td></tr><tr><td>3</td><td>0 to 7</td><td>3 to 5</td><td>0 to 3</td></tr><tr><td>4</td><td>0 to 15</td><td>6 to 10</td><td>0 to 7</td></tr><tr><td>5</td><td>0 to 31</td><td>12 to 20</td><td>0 to 15</td></tr><tr><td>6</td><td>0 to 63</td><td>25 to 40</td><td>0 to 31</td></tr><tr><td>7</td><td>0 to 127</td><td>50 to 81</td><td>0 to 63</td></tr></table> | <b>lod</b>  | Maximum range of <b>x</b> and <b>y</b> values<br><i>0 ≤ x,y ≤ 2<sup>z</sup>-1, where z = lod</i> | Valid <b>x</b> Values for This Product<br><i>From x to x+width-1</i> | Valid <b>y</b> Values for This Product<br><i>From y to y+height-1</i> | 0 | 0 | 0 | 0 | 1 | 0 to 1 | 0 to 1 | 0 | 2 | 0 to 3 | 1 to 2 | 0 to 1 | 3 | 0 to 7 | 3 to 5 | 0 to 3 | 4 | 0 to 15 | 6 to 10 | 0 to 7 | 5 | 0 to 31 | 12 to 20 | 0 to 15 | 6 | 0 to 63 | 25 to 40 | 0 to 31 | 7 | 0 to 127 | 50 to 81 |
| <b>lod</b>   | Maximum range of <b>x</b> and <b>y</b> values<br><i>0 ≤ x,y ≤ 2<sup>z</sup>-1, where z = lod</i> | Valid <b>x</b> Values for This Product<br><i>From x to x+width-1</i>   | Valid <b>y</b> Values for This Product<br><i>From y to y+height-1</i> |  |  |   |   |   |   |   |   |        |        |   |   |        |        |        |   |        |        |        |   |         |         |        |   |         |          |         |   |         |          |         |   |          |          |
| 0  | 0  | 0  | 0   |  |  |   |   |   |   |   |   |        |        |   |   |        |        |        |   |        |        |        |   |         |         |        |   |         |          |         |   |         |          |         |   |          |          |
| 1  | 0 to 1   | 0 to 1   | 0   |  |  |   |   |   |   |   |   |        |        |   |   |        |        |        |   |        |        |        |   |         |         |        |   |         |          |         |   |         |          |         |   |          |          |
| 2  | 0 to 3   | 1 to 2   | 0 to 1  |  |  |   |   |   |   |   |   |        |        |   |   |        |        |        |   |        |        |        |   |         |         |        |   |         |          |         |   |         |          |         |   |          |          |
| 3  | 0 to 7   | 3 to 5   | 0 to 3  |  |  |   |   |   |   |   |   |        |        |   |   |        |        |        |   |        |        |        |   |         |         |        |   |         |          |         |   |         |          |         |   |          |          |
| 4  | 0 to 15  | 6 to 10  | 0 to 7  |  |  |   |   |   |   |   |   |        |        |   |   |        |        |        |   |        |        |        |   |         |         |        |   |         |          |         |   |         |          |         |   |          |          |
| 5  | 0 to 31  | 12 to 20   | 0 to 15   |  |  |   |   |   |   |   |   |        |        |   |   |        |        |        |   |        |        |        |   |         |         |        |   |         |          |         |   |         |          |         |   |          |          |
| 6  | 0 to 63  | 25 to 40   | 0 to 31   |  |  |   |   |   |   |   |   |        |        |   |   |        |        |        |   |        |        |        |   |         |         |        |   |         |          |         |   |         |          |         |   |          |          |
| 7  | 0 to 127   | 50 to 81   | 0 to 63   |  |  |   |   |   |   |   |   |        |        |   |   |        |        |        |   |        |        |        |   |         |         |        |   |         |          |         |   |         |          |         |   |          |          |

|  |  |  |
|--|--|--|
|  |  |  |
|--|--|--|

### Appendix: Geographical Coverage

Data Tiles - Native Resolution, and Web Mercator Projection:

Each product used by Tiler and Featurizer uses data that was originally received in a native projection and then converted into the Web Mercator projection. Users can typically retrieve data and in either projection. There are more details on the lod parameter and the related parameters x and y in the section on API String Parameters, each projection and each level of detail value is summarized in the table below.

- The **/tiler/data** endpoint, the query string parameters **lod**, **x**, and **y** are used to retrieve individual tiles in both projections.
- The **/featurizer/tile** endpoints, the query string parameters **lod**, **x**, and **y** are used to retrieve geographic features for individual tiles only in the **Web Mercator projection**.
- The **/featurizer/feature** set of endpoints produce geographic features for *all* data in the **native projection**, without partitioning the data into tiles.

| lod<br>Level of Detail  | Tile Size<br>(pixels) | Length (tiles)<br>= 2 <sup>z</sup> , where z = lod and z ≥ 0 | Width (tiles)<br>= 2 <sup>z</sup> , where z = lod and z ≥ 0 | Total Number of Tiles<br>= Length x Width | Total Number of Pixels<br>= Total Number of Tiles x Tile Size | Approximate Ground Resolution<br>at the Equator |
|---|-----------------------|--|---|---|---|---|
| Native projection: amount of data and resolution varies with each product                         |                       |  |   |   |   |   |
| -1  | 256 x 256             | varies   | varies  | varies                                    | varies  | varies  |
| Web Mercator projection: number of lod values depends on the resolution for the Native projection |                       |  |   |   |   |   |
| 0   | 256 x 256             | 1  | 1   | 1   | 65,536  | 160 km / pixel                                  |
| 1   | 256 x 256             | 2  | 2   | 4   | 262,144   | 80 km / pixel                                   |
| 2   | 256 x 256             | 4  | 4   | 16  | 1,048,576   | 40 km / pixel                                   |
| 3   | 256 x 256             | 8  | 8   | 64  | 4,194,304   | 20 km / pixel                                   |
| 4   | 256 x 256             | 16   | 16  | 256                                       | 16,777,216  | 10 km / pixel                                   |
| 5   | 256 x 256             | 32   | 32  | 1,024                                     | 67,108,864  | 5 km / pixel                                    |
| 6   | 256 x 256             | 64   | 64  | 4,096                                     | 268,435,456   | 2 km / pixel                                    |
| 7   | 256 x 256             | 128  | 128   | 16,384                                    | 1,073,741,824   | 1 km / pixel                                    |
| 8   | 256 x 256             | 256  | 256   | 65,536                                    | 4,294,967,296   | 0.5 km / pixel                                  |
| 9   | 256 x 256             | 512  | 512   | 262,144                                   | 17,179,869,184  | 0.25 km / pixel                                 |

Appendix: Global Radar Tiles

For some sets of radar products, data is partitioned into Global Radar Tiles, where eight adjacent tiles of roughly equal size provide combined coverage for the entire globe. Two of these tiles span the International Date Line, so the data for each of these tiles is partitioned into two products to eliminate any possible ambiguity about the data’s time and date. The following table summarizes the approximate coverage for the Global Radar Tiles, including the split tiles that span the International Date Line. Ranges run west to east and north to south.

| TILE 4 WEST   | TILE 1   | TILE 2   | TILE 3  | TILE 4 EAST  |
|---|--|--|---|--|
| <i>Alaska, North Pacific Ocean</i><br>180° to 130°W<br>90°N to 0°       | <i>North America, Caribbean</i><br>130°W to 40°W<br>90°N to 0° | <i>Europe, Middle East, North Africa</i><br>40°W to 50°E<br>90°N to 0° | <i>India, China, Central Asia</i><br>50°E to 140°E<br>90°N to 0°      | <i>E. Russia, E. Japan, N. Pacific Ocean</i><br>140°E to 180°<br>90°N to 0°  |
| TILE 8 WEST   | TILE 5   | TILE 6   | TILE 7  | TILE 8 EAST  |
| <i>Cook Islands, South Pacific Ocean</i><br>180° to 130°W<br>0° to 90°S | <i>South America</i><br>130°W to 40°W<br>0° to 90°S            | <i>Southern Africa</i><br>40°W to 50°E<br>0° to 90°S                   | <i>Western Australia, Indian Ocean</i><br>50°E to 140°E<br>0° to 90°S | <i>Eastern Australia, South Pacific Ocean</i><br>140°E to 180°<br>0° to 90°S |

Appendix: Global Satellite Tiles

For some sets of satellite products, data is provided in Global Satellite Tiles, where ten adjacent tiles of various sizes provide coverage for the entire globe between 67°N to 67°S. The following table summarizes the approximate coverage for the Global Satellite Tiles, including the split tiles that span the International Date Line. Ranges run west to east and north to south.

| TILE 1  | TILE 2   | TILE 3  | TILE 4   | TILE 5  |
|---|--|---|--|---|
| <i>North America West (Alaska)</i><br>180° to 125°W<br>67°N to 13°N | <i>North America East (CONUS)</i><br>125°W to 39°W<br>67°N to 13°N | <i>Europe</i><br>39°W to 60°E<br>67°N to 13°N | <i>Asia West</i><br>60°E to 100°E<br>67°N to 0°                          | <i>Asia East</i><br>100°E to 180°<br>67°N to 0°                       |
| TILE 6  | TILE 7   | TILE 8  | TILE 9   | TILE 10   |
| <i>Southern Pacific</i><br>180° to 120°W<br>13°N to 66°S            | <i>South America</i><br>120°W to 20°W<br>13°N to 66°S              | <i>Africa</i><br>20°W to 60°E<br>13°N to 66°S | <i>Indian Ocean (“Western Australia”)</i><br>60°E to 100°E<br>0° to 66°S | <i>Australia (“Eastern Australia”)</i><br>100°E to 180°<br>0° to 66°S |

Appendix: Aggregation

This endpoint returns **up to four aggregated tiles**. Subsequent tiles are requested by appending comma separated arguments to query arguments. The number of comma separated arguments to each query argument must be equal or the request is invalid. The aggregate tile response is composed of one flag float value optionally followed by a 256x256 float length tile. If the flag float value is 0x00000000 then no tile exists for the combination of comma separated arguments given, and therefore no tile data follows the flag. If the value of negative infinity is given, then a tile follows the flag.

- For example, `/tiler/data-aggregate?products={pn1:vid1},{pn2:vid2},{pn:vid3},{pn4:vid4}&t={t1},{t2},{t3},{t4}&lod={lod1},{lod2},{lod3},{lod4}&x={x1},{x2},{x3},{x4}&y={y1},{y2},{y3},{y4}`



- Here, **pn** is the product number, **vid** is the variable ID, and **t** is the “t” (time) dimension; **lod**, **x**, and **y** are the tile coordinates
- Because the order is mandatory, the user should only aggregate products that use the same dimensions.
  - The user could aggregate observation products that all use just the **t** dimension.
  - The user could aggregate forecast products that all use **t** and **rt** dimensions.
  - The user should **not** aggregate a combination of the two types of products.

Examples

Adjacent tiles for the same product:

- <https://api.weather.com/v2/tiler/data-aggregate?products=300:PrecipIntensity,300:PrecipIntensity&t=1451606400000,1451606400000&lod=4,4&x=3,4&y=6,6&apiKey=yourApiKey>

Identical tiles for different products:

- <https://api.weather.com/v2/tiler/data-aggregate?products=77:GOESNA1kmVSIRTOAV2,78:GOESNA4kmIR108TOAV2&t=1451606400000,1451606400000&lod=4,4&x=4,4&y=6,6&apiKey=yourApiKey>

Response

The API returns a binary file with the MIME Content-Type “application/octet-stream”, compressed in response to the following **mandatory** header in the request: “Accept-Encoding: gzip”.

- The response is a gzipped float array, where there are 4 bytes per float in BIG\_ENDIAN order.
- The results within the response will be ordered to match the comma-delimited list in the request’s parameters.

For each requested tile, the API returns one of the following.

| Condition                                  | Response  | Notes  |
|--|---|--|
| The requested tile is available            | Boundary marker, followed by data<br><br>1. “Success” marker: a 4-byte float with the value of <i>-Infinity</i><br>2. Data for the requested tile | The value of <i>-Infinity</i> is equivalent to the value returned by <i>Float.intBitsToFloat(0xff800000)</i> .<br><br>Data is sequenced in rows, left to right (west to east), beginning with the top (northmost) row.<br><br>Since each tile is defined to contain 256 x 256 pixels, and each pixel corresponds to a 4-byte float, the data for each tile will contain 262,144 bytes. |
| The requested tile is <b>NOT</b> available | Boundary marker, without data<br><br>1. “Failure” marker: a 4-byte float of 0x00000000<br><br>No other data is returned for the requested tile    | If this response is for the last requested tile, the data stream will end with this “Failure” boundary marker.<br><br>Otherwise, the next 4-byte float will be the “Success” or “Failure” boundary marker for the next requested tile.   |

Examples

- Two tiles requested, both tiles are successfully returned: **524,296 bytes in the response**
  - **Tile 1:** 4-byte float of *-Infinity* followed by 262,144 bytes of data
  - **Tile 2:** 4-byte float of *-Infinity* followed by 262,144 bytes of data
- Two tiles requested, first tile is missing: **262,152 bytes in the response**
  - **Tile 1:** 4-byte float with the value 0x00000000
  - **Tile 2:** 4-byte float of *-Infinity* followed by 262,144 bytes of data
- Two tiles requested, both tiles are missing: **8 bytes in the response**
  - **Tile 1:** 4-byte float with the value 0x00000000
  - **Tile 2:** 4-byte float with the value 0x00000000

Document Revision History

| Revision | Date        | Notes   |
|----------|-------------|---|
| 1.0      | Dec 7, 2018 | Initial versioned document; addition of Document Revision History   |
| 2.0      | Jun 7, 2022 | Added information for /tiler/byte, /tiler/coverage, /tiler/extras/data, and /featurizer/isoband endpoints |