Web Services

Application Programming Interface (API) Specification

Version 1.3

Jan 3, 2019
# Table of Contents

1. Terrain Sources .................................................................................................................. 4  
   1.1. GET Method .............................................................................................................. 4  
   1.2. POST Method .......................................................................................................... 5  
   1.3. Success Responses ...................................................................................................... 5  
   1.4. Error Responses ....................................................................................................... 8  

2. Terrain Values ................................................................................................................... 9  
   2.1. GET Method .............................................................................................................. 9  
   2.2. POST Method ........................................................................................................... 10  
   2.3. Success Responses .................................................................................................... 10  
   2.4. Error Responses ....................................................................................................... 11  
   2.5. Asynchronous Processing ........................................................................................ 12  
   2.6. Asynchronous Status .............................................................................................. 12  
   2.7. Asynchronous Result .............................................................................................. 14  

3. Terrain Profile .................................................................................................................. 16  
   3.1. GET Method .............................................................................................................. 16  
   3.2. POST Method .......................................................................................................... 17  
   3.3. Success Responses .................................................................................................... 18  
   3.4. Error Responses ....................................................................................................... 20  
   3.5. Asynchronous Processing ......................................................................................... 21  
      3.5.1 Asynchronous Status ........................................................................................... 21  
      3.5.2 Asynchronous Result .......................................................................................... 22  

4. Terrain Viewshed ................................................................................................................. 26  
   4.1. GET Method .............................................................................................................. 27  
   4.2. POST Method .......................................................................................................... 28  
   4.3. Success Responses .................................................................................................... 29  
   4.4. Error Responses ....................................................................................................... 30  
   4.5. Viewshed Result – GET Method ........................................................................... 30  
   4.6. Viewshed Result – POST Method .......................................................................... 31  
   4.7. Viewshed Result - Success Responses .................................................................. 32  
   4.8. Viewshed Result - Error Responses ...................................................................... 32  
   4.9. Asynchronous Processing ......................................................................................... 33  
      4.9.1 Asynchronous Status ........................................................................................... 33  
      4.9.2 Asynchronous Result .......................................................................................... 35
5. Link Profile ............................................................................................................................................... 37
  5.1. GET Method ..................................................................................................................................... 38
  5.2. POST Method ................................................................................................................................... 39
  5.3. Success Responses ............................................................................................................................ 41
  5.4. Error Responses ................................................................................................................................. 44
  5.5. Link Profile Result – GET Method ....................................................................................................... 44
  5.6. Linkpro Result – POST Method .......................................................................................................... 46
  5.7. Linkpro Result - Success Responses .................................................................................................. 47
  5.8. Linkpro Result - Error Responses ....................................................................................................... 50
  5.9. Asynchronous Processing .................................................................................................................... 51
    5.9.1 Asynchronous Status ...................................................................................................................... 51
    5.9.2 Asynchronous Result ...................................................................................................................... 53
1. Terrain Sources

Provides a list data sources available to the Terrain Services. Each data source is given a unique identifier which is used as in the endpoint URL in the other Terrain Services calls.

<table>
<thead>
<tr>
<th>Endpoint URL</th>
<th><a href="https://api.intermap.com/api/raster">https://api.intermap.com/api/raster</a></th>
</tr>
</thead>
<tbody>
<tr>
<td>Authentication</td>
<td>Basic</td>
</tr>
<tr>
<td>Methods</td>
<td>GET, POST</td>
</tr>
</tbody>
</table>

1.1. GET Method

<table>
<thead>
<tr>
<th>URL Parameters</th>
<th>Name</th>
<th>Required</th>
<th>Description</th>
<th>Example</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>easting</td>
<td>No</td>
<td>Northing coordinates of a point on the earth’s surface. When used, only data sources containing the given point are returned, otherwise, all data sources are returned. If the crs parameter is specified, the coordinate units are determined by the crs. If no crs parameter is given, then the coordinates are in WGS84 Geographic decimal degrees (EPSG:4326).</td>
<td>easting=10 northing=50</td>
</tr>
<tr>
<td></td>
<td>northing</td>
<td>No</td>
<td>Coordinate reference system (CRS) in format EPSG:identifier where identifier is the EPSG code for the given CRS.</td>
<td>crs=EPSG:4326</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Header Parameters</th>
<th>Content</th>
</tr>
</thead>
</table>

<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Header Parameters</td>
<td>Content</td>
</tr>
</tbody>
</table>
1.2. POST Method

<table>
<thead>
<tr>
<th>URL Parameters</th>
<th>Content-Type: application/json</th>
</tr>
</thead>
<tbody>
<tr>
<td>Header Parameters</td>
<td>Content-Type: application/json</td>
</tr>
<tr>
<td>Content</td>
<td>If the content is empty, all data sources are returned. Request content is in GeoJSON format and contains a FeatureCollection. Each Feature within the FeatureCollection contains Point geometry. Only the data sources that contain at least one of the points are returned. The FeatureCollection may also contain a crs which is used to identify the Coordinate Reference System of the point coordinates. If no crs is given then WGS84 Geographic decimal degrees (EPSG:4326) is used.</td>
</tr>
</tbody>
</table>

Example Request

<table>
<thead>
<tr>
<th>URL</th>
<th><a href="https://api.intermap.com/api/raster">https://api.intermap.com/api/raster</a></th>
</tr>
</thead>
<tbody>
<tr>
<td>Header Parameters</td>
<td>Content-Type: application/json</td>
</tr>
<tr>
<td>Content</td>
<td>{&quot;type&quot;: &quot;FeatureCollection&quot;, &quot;crs&quot;: { &quot;type&quot;: &quot;name&quot;, &quot;properties&quot;: { &quot;name&quot;: &quot;EPSG:4326&quot; } }, &quot;features&quot;: [ {&quot;type&quot;: &quot;Feature&quot;, &quot;geometry&quot;: {&quot;type&quot;: &quot;Point&quot;, &quot;coordinates&quot;: [-105,40]}}, {&quot;type&quot;: &quot;Feature&quot;, &quot;geometry&quot;: {&quot;type&quot;: &quot;Point&quot;, &quot;coordinates&quot;: [10,50]}} ] }</td>
</tr>
</tbody>
</table>

1.3. Success Responses

<table>
<thead>
<tr>
<th>Code</th>
<th>200</th>
</tr>
</thead>
<tbody>
<tr>
<td>Description</td>
<td>Success – Response contains terrain sources content</td>
</tr>
<tr>
<td>Content Type</td>
<td>application/json</td>
</tr>
<tr>
<td>Content Description</td>
<td>The contents are returned as a GeoJSON FeatureCollection where each Feature in the FeatureCollection contains an “id” element which is the ID to use in other Terrain Services calls. Each Feature also contains a polygon geometry defining the bounds of the data source and some other properties.</td>
</tr>
<tr>
<td>Content Example</td>
<td>{ &quot;type&quot;: &quot;FeatureCollection&quot;, &quot;features&quot;: [ { &quot;type&quot;: &quot;Feature&quot;, &quot;id&quot;: &quot;NEXTMap-Europe_DTM&quot;, } ] }</td>
</tr>
</tbody>
</table>
"geometry": {
  "type": "Polygon",
  "coordinates": [
    [
      [-10.751979166666665, 35.37068750000004],
      [-10.751979166666665, 60.88535416675661],
      [19.008020833333536, 60.88535416675661],
      [19.008020833333536, 35.37068750000004],
      [-10.751979166666665, 35.37068750000004]
    ]
  ],
  "properties": {
    "name": "NEXTMap-Europe_DTM",
    "remarks": null,
    "summary": "DTM - NEXTMap®Europe",
    "version": "3.3"
  }
},
{
  "type": "Feature",
  "id": "NEXTMap-Europe_DSM",
  "geometry": {
    "type": "Polygon",
    "coordinates": [
      [-10.751979166666665, 35.37068750000004],
      [-10.751979166666665, 60.88535416675661],
      [19.008020833333536, 60.88535416675661],
      [19.008020833333536, 35.37068750000004]
    ]
  }
}
```json
[
  {
    "type": "Feature",
    "id": "NEXTMap-Europe_DSM",
    "geometry": {
      "type": "Polygon",
      "coordinates": [
        [-10.751979166666665, 35.370687500000004]
      ]
    },
    "properties": {
      "name": "NEXTMap-Europe_DSM",
      "remarks": null,
      "summary": "DSM - NEXTMap®Europe",
      "version": "3.3"
    }
  },
  {
    "type": "Feature",
    "id": "NEXTMap-World30_DEM",
    "geometry": {
      "type": "Polygon",
      "coordinates": [
        [-179.9999, -89.9999],
        [-179.9999, 84.0534722221286],
        [179.9999, 84.0534722221286],
        [179.9999, -89.9999],
        [-179.9999, -89.9999]
      ]
    },
    "properties": {
      "name": "NEXTMap-World30_DEM",
      "remarks": null,
      "summary": "DEM30 - NEXTMap®World30"
    }
  }
]
## 1.4. Error Responses

<table>
<thead>
<tr>
<th>Code</th>
<th>Description</th>
<th>Content Type</th>
<th>Content Example</th>
</tr>
</thead>
</table>
| 401   | Unauthorized: Access is denied due to invalid credentials.                  | text/xml     | ```xml
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
<meta http-equiv="Content-Type" content="text/html; charset=iso-8859-1"/>
<title>401 Unauthorized</title>
<style type="text/css">body{margin:0;font-size:.7em;font-family:Verdana, Arial, Helvetica, sans-serif;background:#EEEEEE;} fieldset{padding:0 15px 15px;} h1{font-size:2.4em;margin:0;color:#FFF;} h2{font-size:1.7em;margin:0;color:#CC0000;} h3{font-size:1.2em;margin:10px 0 0 0;color:#000000;} #header{width:96%;margin:0 0 0 0;padding:6px 10px 6px 10px;} #content{margin:0 0 0 2%;position:relative;} .content-container{background:#FFF;width:96%;margin-top:8px;padding:10px;position:relative;} --> </style>
<body>
<header>
<h1>Server Error</h1>
<fieldset>
<h2>401 Unauthorized: Access is denied due to invalid credentials.</h2>
<h3>You do not have permission to view this directory or page using the credentials that you supplied.</h3>
</fieldset></header>
</body>
</html>``` |

| Code  | Model is not valid. This may be due to specifying the parameters incorrectly. | application/xml | ```xml
<string>Model is not valid. This error has been assigned an Incident ID of '66480-API2-TEST03-170605-193846'. Provide this Incident ID to customer support when you need assistance.</string>``` |
2. Terrain Values

Intermap's Values allows the user to quickly get the value (From any raster data set) at any geographic coordinate (assuming it is in the data set). This is very useful for analyzing data at specific locations. Intermap's values tool allows a list (batch) of points to be specified for running batch analysis. Examples include: getting the elevation height, risk information, flood depth, slope, etc.

Gets the values (elevations) of a set of points on a terrain data source.

**Endpoint URL**
https://api.intermap.com/api/raster/<Source ID>/values
Where `<Source ID>` is a data source ID returned from a Terrain Sources call

**Authentication**
Basic

**Methods**
GET, POST

2.1. GET Method

**URL Parameters**

<table>
<thead>
<tr>
<th>Name</th>
<th>Required</th>
<th>Description</th>
<th>Example</th>
</tr>
</thead>
<tbody>
<tr>
<td>easting</td>
<td>Yes</td>
<td>Easting and northing coordinates of a point for which to get the value for.</td>
<td>easting=10</td>
</tr>
<tr>
<td>northing</td>
<td></td>
<td>If the crs parameter is specified, the coordinate units are determined by the</td>
<td>northing=50</td>
</tr>
<tr>
<td>crs</td>
<td>No</td>
<td>Coordinate reference system (CRS) in format EPSG:identifier where identifier</td>
<td>crs=EPSG:4326</td>
</tr>
<tr>
<td></td>
<td></td>
<td>is the EPSG code for the given CRS.</td>
<td></td>
</tr>
</tbody>
</table>

**Header Parameters**

**Content**

**Example Request**

<table>
<thead>
<tr>
<th>URL</th>
<th>Example Request</th>
</tr>
</thead>
</table>

**Header Parameters**
2.2. POST Method

**URL Parameters**

**Header Parameters**

Content-Type: application/json

**Content**

Request content is in GeoJSON format and contains a FeatureCollection. Each Feature within the FeatureCollection contains Point geometry. The values (elevations) for all given points is returned. The FeatureCollection may also contain a crs which is used to identify the Coordinate Reference System of the point coordinates. If no crs is given then WGS84 Geographic decimal degrees (EPSG:4326) is used.

**Example Request**

<table>
<thead>
<tr>
<th>URL</th>
<th><a href="https://api.intermap.com/api/raster/NEXTMap-USA_DSM/values">https://api.intermap.com/api/raster/NEXTMap-USA_DSM/values</a></th>
</tr>
</thead>
<tbody>
<tr>
<td>Header Parameters</td>
<td>Content-Type: application/json</td>
</tr>
<tr>
<td>Content</td>
<td></td>
</tr>
</tbody>
</table>

```json
{"type": "FeatureCollection",
 "crs": {
  "type": "name",
  "properties": {
    "name": "EPSG:4326"
  }
},
"features": [
  {"type": "Feature", "geometry": {
    "type": "Point",
    "coordinates": [-105,40]
  }},
  {"type": "Feature", "geometry": {
    "type": "Point",
    "coordinates": [10,50]
  }}
]
```
2.4. Error Responses

<table>
<thead>
<tr>
<th>Code</th>
<th>Description</th>
<th>Header Parameters</th>
<th>Content Type</th>
<th>Content Description</th>
<th>Content Example</th>
</tr>
</thead>
<tbody>
<tr>
<td>401</td>
<td>401 UNAUTHORIZED</td>
<td>Location</td>
<td>text/xml</td>
<td>Error when the authentication fails possibly due to incorrect login</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Error when the authentication fails possibly due to incorrect login</td>
<td></td>
<td></td>
<td></td>
<td>&lt;html xmlns=&quot;http://www.w3.org/1999/xhtml&quot;&gt; &lt;head&gt; &lt;meta http-equiv=&quot;Content-Type&quot; content=&quot;text/html; charset=iso-8859-1&quot;/&gt; &lt;title&gt;401 - Unauthorized: Access is denied due to invalid credentials.&lt;/title&gt; &lt;/head&gt; &lt;body&gt;...content-</td>
</tr>
</tbody>
</table>
2.5. Asynchronous Processing

When a task will take a long time, it will be sent off for asynchronous processing rather than making the caller wait for a response. In this case, a response with a code of 202 is returned. This response contains a Location header parameter which is a URL for obtaining the status and when complete, the result.

2.6. Asynchronous Status

The asynchronous status is a GET method that allows the user to check the status of the asynchronous job, and when complete, it gives the URL for the result. It uses the same basic authentication as the other methods.

<table>
<thead>
<tr>
<th>URL Parameters</th>
<th>None</th>
</tr>
</thead>
<tbody>
<tr>
<td>Header Parameters</td>
<td>None</td>
</tr>
<tr>
<td>Content</td>
<td>None</td>
</tr>
<tr>
<td>Example Request</td>
<td></td>
</tr>
<tr>
<td>URL</td>
<td><a href="https://api-test.intermap.com/tasks/02237c3e-4f0e-e711-80c3-000d3a604bff/status">https://api-test.intermap.com/tasks/02237c3e-4f0e-e711-80c3-000d3a604bff/status</a></td>
</tr>
<tr>
<td>Header Parameters</td>
<td></td>
</tr>
<tr>
<td>Content</td>
<td></td>
</tr>
<tr>
<td>Success Responses</td>
<td></td>
</tr>
<tr>
<td>Code</td>
<td>200</td>
</tr>
<tr>
<td>Description</td>
<td>Success – Status check is successful, but the job is still processing</td>
</tr>
<tr>
<td>Content Type</td>
<td>application/json</td>
</tr>
</tbody>
</table>
### Content Description
The contents are simple JSON with a “location” element echoing back the status URL and a “status” element that gives the current status of the processing.

<table>
<thead>
<tr>
<th>Content Example</th>
</tr>
</thead>
</table>
| {
  "location": "https://api-test.intermap.com/tasks/02237c3e-4f0e-e711-80c3-000d3a604bff/status",
  "status": "processing"
} |

<table>
<thead>
<tr>
<th>Code</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>303</td>
<td>Redirect – Status check is successful and the job is complete</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Header Parameters</th>
<th>Content Type</th>
<th>Content Description</th>
<th>Content Example</th>
</tr>
</thead>
<tbody>
<tr>
<td>Location</td>
<td>No Content</td>
<td>There is no content</td>
<td></td>
</tr>
</tbody>
</table>

### Failure Responses

<table>
<thead>
<tr>
<th>Code</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>401</td>
<td>UNAUTHORIZED: Access is denied due to invalid credentials. You do not have permission to view this directory or page using the credentials that you supplied.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Code</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>400</td>
<td>Model is not valid. This may be due to specifying the parameters or content incorrectly.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Content Type</th>
<th>Content Example</th>
</tr>
</thead>
<tbody>
<tr>
<td>text/xml</td>
<td>&lt;string&gt;Model is not valid. This error has been assigned an Incident ID of '66480-API2-TEST03-170605-193846'. Provide this Incident ID to customer support when you need assistance.&lt;/string&gt;</td>
</tr>
</tbody>
</table>

Copyright © 2019 by Intermap Technologies. All rights reserved
Document Version: 2017-05-15
### 2.7. Asynchronous Result

The asynchronous result is a GET method that simply gets back the result from an asynchronous processing job with the URL given in the asynchronous status call.

<table>
<thead>
<tr>
<th>URL Parameters</th>
<th>None</th>
</tr>
</thead>
<tbody>
<tr>
<td>Header Parameters</td>
<td>None</td>
</tr>
<tr>
<td>Content</td>
<td>None</td>
</tr>
</tbody>
</table>

**Example Request**

<table>
<thead>
<tr>
<th>URL</th>
<th><a href="https://api.intermap.com/tasks/02237c3e-4f0e-e711-80c3-000d3a604bff">https://api.intermap.com/tasks/02237c3e-4f0e-e711-80c3-000d3a604bff</a></th>
</tr>
</thead>
<tbody>
<tr>
<td>Header Parameters</td>
<td></td>
</tr>
<tr>
<td>Content</td>
<td></td>
</tr>
</tbody>
</table>

**Success Responses**

<table>
<thead>
<tr>
<th>Code</th>
<th>200</th>
</tr>
</thead>
<tbody>
<tr>
<td>Description</td>
<td>Success – Result returned</td>
</tr>
<tr>
<td>Content Type</td>
<td>application/json</td>
</tr>
</tbody>
</table>

**Content Description**

The contents are the same as if they were returned without an asynchronous call. They contain a GeoJSON FeatureCollection where each Feature in the FeatureCollection contains a point geometry matching the requested points and a “value” property containing the value from the data source at that point.

**Content Example**

```json
{
    "type": "FeatureCollection",
    "features": [
        {
            "type": "Feature",
            "geometry": {
                "type": "Point",
                "coordinates": [-105, 40]
            },
            "properties": {
                "value": 1589.68359
            }
        },
        {
            "type": "Feature",
            "geometry": {
                "type": "Point",
                "coordinates": [1589.68359, 40]
            },
            "properties": {
                "value": 1589.68359
            }
        }
    ]
}
```
<table>
<thead>
<tr>
<th>Code</th>
<th>401 UNAUTHORIZED</th>
</tr>
</thead>
<tbody>
<tr>
<td>Description</td>
<td>Error when the authentication fails possibly due to incorrect login</td>
</tr>
<tr>
<td>Content Type</td>
<td>text/xml</td>
</tr>
<tr>
<td>Content Example</td>
<td><code>&lt;!DOCTYPE html PUBLIC &quot;-//W3C//DTD XHTML 1.0 Strict//EN&quot; &quot;http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd&quot;&gt; &lt;html xmlns=&quot;http://www.w3.org/1999/xhtml&quot;&gt; &lt;head&gt; &lt;meta http-equiv=&quot;Content-Type&quot; content=&quot;text/html; charset=iso-8859-1&quot;/&gt; &lt;title&gt;401 - Unauthorized: Access is denied due to invalid credentials.&lt;/title&gt; &lt;style&gt; body{margin:0;font-size:7em;font-family:Verdana, Arial, Helvetica, sans-serif;background:#EEEEEE;} h1{font-size:2.4em;margin:0;color:#FFF;} h2{font-size:1.7em;margin:0;color:#CC0000;} h3{font-size:1.2em;margin:10px 0 0 0;color:#000000;} #header{width:96%;margin:0 0 0 0;padding:6px 2% 6px 2%;font-family:&quot;trebuchet MS&quot;, Verdana, sans-serif;color:#FFF; background-color:#555555;} #content{margin:0 0 2%;position:relative;} .content-container{background:#FFF;width:96%;margin-top:8px;padding:10px;position:relative;} --&gt;&lt;/style&gt; &lt;/head&gt; &lt;body&gt; &lt;div id=&quot;header&quot;&gt; &lt;h1&gt;Server Error&lt;/h1&gt; &lt;/div&gt; &lt;div id=&quot;content&quot;&gt; &lt;div class=&quot;content-container&quot;&gt; &lt;fieldset&gt; &lt;h2&gt;401 - Unauthorized: Access is denied due to invalid credentials.&lt;/h2&gt; &lt;h3&gt;You do not have permission to view this directory or page using the credentials that you supplied.&lt;/h3&gt; &lt;/fieldset&gt; &lt;/div&gt; &lt;/div&gt; &lt;/html&gt;</code></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Code</th>
<th>400</th>
</tr>
</thead>
<tbody>
<tr>
<td>Description</td>
<td>Model is not valid. This may be due to specifying the parameters or content incorrectly.</td>
</tr>
<tr>
<td>Content Type</td>
<td>application/xml</td>
</tr>
<tr>
<td>Content Example</td>
<td><code>&lt;string&gt;Model is not valid. This error has been assigned an Incident ID of '66480-API2-TEST03-170605-193846'. Provide this Incident ID to customer support when you need assistance.&lt;/string&gt;</code></td>
</tr>
</tbody>
</table>
3. Terrain Profile

A terrain profile is a set of points along a straight line from one location to another. It is called a profile, since it is typically used on elevation to create an elevation profile or cross section. Intermap’s terrain profile allows the user to specify the number of steps (samples) between the start and end points required. Profiles are often plotted as shown in the example below.

---

**Endpoint URL**

https://api.intermap.com/api/raster/<Source ID>/profile

Where <Source ID> is a data source ID returned from a Terrain Sources call

**Authentication**

Basic

**Methods**

GET, POST

---

3.1. GET Method

<table>
<thead>
<tr>
<th>URL Parameters</th>
<th>Name</th>
<th>Required</th>
<th>Description</th>
<th>Example</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>eastingStart</td>
<td>Yes</td>
<td>Easting and northing coordinates of the start point for the profile. If the crs parameter is specified, the coordinate units are determined by the crs. If no crs parameter is given, then</td>
<td>eastingStart=-105</td>
</tr>
<tr>
<td></td>
<td>northingStart</td>
<td></td>
<td></td>
<td>northingStart=40</td>
</tr>
</tbody>
</table>
### Header Parameters

### Content

### Example Request

#### URL

<table>
<thead>
<tr>
<th>Request URL</th>
</tr>
</thead>
<tbody>
<tr>
<td><a href="https://api.intermap.com/api/raster/NEXTMap-USA_DSM/profile?eastingStart=-105&amp;northingStart=40&amp;eastingEnd=-105.01&amp;northingEnd=40.01&amp;step=500&amp;crs=EPSG:4326">https://api.intermap.com/api/raster/NEXTMap-USA_DSM/profile?eastingStart=-105&amp;northingStart=40&amp;eastingEnd=-105.01&amp;northingEnd=40.01&amp;step=500&amp;crs=EPSG:4326</a></td>
</tr>
</tbody>
</table>

#### Header Parameters

<table>
<thead>
<tr>
<th>Content</th>
</tr>
</thead>
</table>

### 3.2. POST Method

#### URL

<table>
<thead>
<tr>
<th>Parameters</th>
</tr>
</thead>
</table>

| the coordinates are in WGS84 Geographic decimal degrees (EPSG:4326). |
| eastingEnd northingEnd Yes Easting and northing coordinates of the end point for the profile. If the crs parameter is specified, the coordinate units are determined by the crs. If no crs parameter is given, then the coordinates are in WGS84 Geographic decimal degrees (EPSG:4326). |
| eastingEnd=-105.01 northingEnd=40.01 |
| step Yes The sampling step size along the profile between the start and end points. The value is in meters. |
| step=100 |
| crs No Coordinate reference system (CRS) for the input coordinates in format EPSG:identifier where identifier is the EPSG code for the given CRS. Default if not specified is EPSG:4326. |
| crs=EPSG:4326 |
### Header

**Parameters**

Content-Type: application/json

### Content

Request content is in GeoJSON format and contains a FeatureCollection. There are two Features within the FeatureCollection which specify the start and end point of the profile. There is also a “step” property in the FeatureCollection that identifies the sampling step size in meters along the profile between the start and end points. The FeatureCollection may also contain a crs which is used to identify the Coordinate Reference System of the point coordinates. If no crs is given then WGS84 Geographic decimal degrees (EPSG:4326) is used.

### Example Request

<table>
<thead>
<tr>
<th>URL</th>
<th><a href="https://api.intermap.com/api/raster/NEXTMap-USA_DSM/profile">https://api.intermap.com/api/raster/NEXTMap-USA_DSM/profile</a></th>
</tr>
</thead>
<tbody>
<tr>
<td>Header Parameters</td>
<td>Content-Type: application/json</td>
</tr>
<tr>
<td>Content</td>
<td></td>
</tr>
</tbody>
</table>

```json
{
  "type": "FeatureCollection",
  "crs": {
    "type": "name",
    "properties": {
      "name": "EPSG:4326"
    }
  },
  "features": [
    {
      "type": "Feature",
      "geometry": {
        "type": "Point",
        "coordinates": [-105, 40]
      }
    }
    ,
    {
      "type": "Feature",
      "geometry": {
        "type": "Point",
        "coordinates": [-105.01, 40.01]
      }
    }
  ],
  "properties": {
    "step": 500
  }
}
```

### 3.3. Success Responses

<table>
<thead>
<tr>
<th>Code</th>
<th>200</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Description</strong></td>
<td>Success – Response contains requested profile</td>
</tr>
<tr>
<td><strong>Content Type</strong></td>
<td>application/json</td>
</tr>
<tr>
<td><strong>Content Description</strong></td>
<td>The contents are returned as a GeoJSON FeatureCollection where each Feature in the FeatureCollection is a point along the profile which has the point’s geometry along with a</td>
</tr>
</tbody>
</table>
“distance” property which gives the distance in meters from the previous point in the profile and a “value” property which is the value (elevation) of the terrain data source at that point.

Content Example

```json
{
    "type": "FeatureCollection",
    "features": [
        {
            "type": "Feature",
            "geometry": {
                "type": "Point",
                "coordinates": [-105, 40]
            },
            "properties": {
                "distance": 0,
                "value": 1589.68359
            }
        },
        {
            "type": "Feature",
            "geometry": {
                "type": "Point",
                "coordinates": [-105.00356929493735, 40.003569770806926]
            },
            "properties": {
                "distance": 500,
                "value": 1601.67566
            }
        },
        {
            "type": "Feature",
            "geometry": {
                "type": "Point",
                "coordinates": [-105.00713896168041, 40.00713938952522]
            },
            "properties": {
                "distance": 500,
                "value": 1610.09192
            }
        },
        {
            "type": "Feature",
            "geometry": {
                "type": "Point",
                "coordinates": [-105.01, 40.01]
            }
        }
    ]
}
```
3.4. Error Responses

<table>
<thead>
<tr>
<th>Code</th>
<th>Description</th>
<th>Content Type</th>
<th>Content Example</th>
</tr>
</thead>
<tbody>
<tr>
<td>401</td>
<td>Unauthorized: Access is denied due to invalid credentials.</td>
<td>text/html</td>
<td>&lt;html xmlns=&quot;http://www.w3.org/1999/xhtml&quot;&gt;...&lt;/html&gt;</td>
</tr>
<tr>
<td>400</td>
<td>Model is not valid. This may be due to specifying the parameters or content incorrectly.</td>
<td>application/xml</td>
<td>&lt;string&gt;Model is not valid. This error has been assigned an Incident ID of '66480-API2-TEST03-170605-193846'. Provide this Incident ID to customer support when you need assistance.&lt;/string&gt;</td>
</tr>
</tbody>
</table>
3.5. Asynchronous Processing

When a task will take a long time, it will be sent off for asynchronous processing rather than making the caller wait for a response. In this case, a response with a code of 202 is returned. This response contains a Location header parameter which is a URL for obtaining the status and when complete, the result.

The asynchronous status is a GET method that allows the user to check the status of the asynchronous job, and when complete, it gives the URL for the result. It uses the same basic authentication as the other methods.

<table>
<thead>
<tr>
<th>URL Parameters</th>
<th>None</th>
</tr>
</thead>
<tbody>
<tr>
<td>Header Parameters</td>
<td>None</td>
</tr>
<tr>
<td>Content</td>
<td>None</td>
</tr>
</tbody>
</table>

**Example Request**

<table>
<thead>
<tr>
<th>URL</th>
<th><a href="https://api-test.intermap.com/tasks/02237c3e-4f0e-e711-80c3-000d3a604bff/status">https://api-test.intermap.com/tasks/02237c3e-4f0e-e711-80c3-000d3a604bff/status</a></th>
</tr>
</thead>
<tbody>
<tr>
<td>Header Parameters</td>
<td></td>
</tr>
<tr>
<td>Content</td>
<td></td>
</tr>
</tbody>
</table>

**Success Responses**

<table>
<thead>
<tr>
<th>Code</th>
<th>200</th>
</tr>
</thead>
<tbody>
<tr>
<td>Description</td>
<td>Success – Status check is successful, but the job is still processing</td>
</tr>
<tr>
<td>Content Type</td>
<td>application/json</td>
</tr>
<tr>
<td>Content Description</td>
<td>The contents are simple JSON with a “location” element echoing back the status URL and a “status” element that gives the current status of the processing.</td>
</tr>
<tr>
<td>Content Example</td>
<td>{</td>
</tr>
</tbody>
</table>
| | "location": "https://api-test.intermap.com/tasks/02237c3e-4f0e-e711-80c3-000d3a604bff/status",
| | "status": "processing" |
| | } |
| Code | 303 |
| Description | Redirect – Status check is successful and the job is complete |
| Header Parameters | Location – URL that contains the result of the processing |
## Failure Responses

<table>
<thead>
<tr>
<th>Code</th>
<th>Description</th>
<th>Content Type</th>
<th>Content Example</th>
</tr>
</thead>
<tbody>
<tr>
<td>401</td>
<td>UNAUTHORIZED: Access is denied due to invalid credentials.</td>
<td>text/html</td>
<td><code>&lt;html xmlns=&quot;http://www.w3.org/1999/xhtml&quot;&gt;</code></td>
</tr>
<tr>
<td>400</td>
<td>Model is not valid. This may be due to specifying the parameters or content incorrectly.</td>
<td>application/xml</td>
<td><code>&lt;string&gt;Model is not valid. This error has been assigned an Incident ID of ’66480-API2-TEST03-170605-193846’. Provide this Incident ID to customer support when you need assistance.&lt;/string&gt;</code></td>
</tr>
</tbody>
</table>

The asynchronous result is a GET method that simply gets back the result from an asynchronous processing job with the URL given in the asynchronous status call.

<table>
<thead>
<tr>
<th>URL Parameters</th>
<th>None</th>
</tr>
</thead>
</table>
### Header

<table>
<thead>
<tr>
<th>Parameters</th>
<th>None</th>
</tr>
</thead>
</table>

### Content

<table>
<thead>
<tr>
<th>Parameters</th>
<th>None</th>
</tr>
</thead>
</table>

### Example Request

<table>
<thead>
<tr>
<th>URL</th>
<th><a href="https://api.intermap.com/tasks/02237c3e-4f0e-e711-80c3-000d3a604bff">https://api.intermap.com/tasks/02237c3e-4f0e-e711-80c3-000d3a604bff</a></th>
</tr>
</thead>
<tbody>
<tr>
<td>Header</td>
<td>None</td>
</tr>
<tr>
<td>Content</td>
<td>None</td>
</tr>
</tbody>
</table>

### Success Responses

<table>
<thead>
<tr>
<th>Code</th>
<th>200</th>
</tr>
</thead>
<tbody>
<tr>
<td>Description</td>
<td>Success – Result returned</td>
</tr>
<tr>
<td>Content Type</td>
<td>application/json</td>
</tr>
</tbody>
</table>

**Content Description**

The contents are the same as if they were returned without an asynchronous call. They contain a GeoJSON FeatureCollection where each Feature in the FeatureCollection is a point along the profile which has the point’s geometry along with a “distance” property which gives the distance in meters from the previous point in the profile and a “value” property which is the value (elevation) of the terrain data source at that point.

**Content Example**

```json
{
    "type": "FeatureCollection",
    "features": [
        {
            "type": "Feature",
            "geometry": {
                "type": "Point",
                "coordinates": [-105, 40]
            },
            "properties": {
                "distance": 0,
                "value": 1589.68359
            }
        },
        {
            "type": "Feature",
            "geometry": {
                "type": "Point",
                "coordinates": [-105.0035692949375, 40.003569770806926]
            },
            "properties": {
                "distance": 500,
                "value": 1589.68359
            }
        }
    ]
} 
```
### Failure Responses

<table>
<thead>
<tr>
<th>Code</th>
<th>Description</th>
<th>Content Type</th>
<th>Content Example</th>
</tr>
</thead>
</table>
| 401        | Unauthorized: Error when the authentication fails possibly due to incorrect login | text/xml     | ```xml
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
<meta http-equiv="Content-Type" content="text/html; charset=iso-8859-1"/>
<title>401 Unauthorized: Access is denied due to invalid credentials.</title>

<body>
<h1>Server Error</h1>
<fieldset>
<h2>401 Unauthorized: Access is denied due to invalid credentials.</h2>
<h3>You do not have permission to view this directory or page using the credentials that you supplied.</h3>
</fieldset>
</body>
</html>
``` |
<table>
<thead>
<tr>
<th>Code</th>
<th>400</th>
</tr>
</thead>
<tbody>
<tr>
<td>Description</td>
<td>Model is not valid. This may be due to specifying the parameters or content incorrectly.</td>
</tr>
<tr>
<td>Content Type</td>
<td>application/xml</td>
</tr>
<tr>
<td>Content Example</td>
<td><code>&lt;string&gt;Model is not valid. This error has been assigned an Incident ID of '66480-API2-TEST03-170605-193846'. Provide this Incident ID to customer support when you need assistance.&lt;/string&gt;</code></td>
</tr>
</tbody>
</table>
4. Terrain Viewshed

A viewshed is an area that is visible from a specific location. Viewshed analyses are a common function of most geographic information system (GIS) software. The analysis uses the elevation value of each cell of the digital elevation model (DEM) to determine visibility to or from a particular cell. The location of this particular cell varies depending on the needs of the analysis. A viewshed is created from a DEM by using an algorithm that estimates the difference of elevation from one cell (the viewpoint cell) to the next (the target cell). To determine the visibility of a target cell, each cell between the viewpoint cell and target cell is examined for line of sight. Where cells of higher value are between the viewpoint and target cells the line of sight is blocked. If the line of sight is blocked then the target cell is determined to not be part of the viewshed.

Intermap's viewshed can be configured with a variety of variables. For example, if the analysis is to determine the location of a radio tower, the height of the tower (viewpoint cell) could be added to the elevation of that location (cell value). If no height is given, then the viewshed analysis uses the cell value of the DEM in which the tower is located. Similarly, the receiver cell (Target cell) can have a height associate with it. The viewshed analysis can also have a limited viewing angle: The first value is the lowest possible azimuth angle and the second value is the highest possible azimuth angle. Intermap's viewshed also allows the radius to be adjusted to limit the range (and run time) of the calculation.

Gets a viewshed from a given location.

<table>
<thead>
<tr>
<th>Endpoint URL</th>
<th><a href="https://api.intermap.com/api/raster/">https://api.intermap.com/api/raster/</a>&lt;Source ID&gt;/viewshed</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Where &lt;Source ID&gt; is a data source ID returned from a Terrain Sources call</td>
</tr>
<tr>
<td>Authentication</td>
<td>Basic</td>
</tr>
<tr>
<td>Methods</td>
<td>GET, POST</td>
</tr>
</tbody>
</table>
### 4.1. GET Method

#### URL Parameters

<table>
<thead>
<tr>
<th>Name</th>
<th>Required</th>
<th>Description</th>
<th>Example</th>
</tr>
</thead>
<tbody>
<tr>
<td>easting</td>
<td>Yes</td>
<td>Easting and northing coordinates of the center point (source) of the viewshed. If the crs parameter is specified, the coordinate units are determined by the crs. If no crs parameter is given, then the coordinates are in WGS84 Geographic decimal degrees (EPSG:4326).</td>
<td>easting=-105</td>
</tr>
<tr>
<td>northing</td>
<td></td>
<td></td>
<td>northing=40</td>
</tr>
<tr>
<td>radius</td>
<td>Yes</td>
<td>Radius of the viewshed from the center point in meters</td>
<td>radius=500</td>
</tr>
<tr>
<td>center</td>
<td>No</td>
<td>The height of the center point in meters above the terrain (default = 0)</td>
<td>centerHeight=0.5</td>
</tr>
<tr>
<td>Height</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>fieldHeight</td>
<td>No</td>
<td>The height of values in the field above the terrain. (default = 0)</td>
<td>fieldHeight=0.3</td>
</tr>
<tr>
<td>azimuth</td>
<td>No</td>
<td>The starting azimuth in degrees of the resulting viewshed field. (default = 0)</td>
<td>azimuthStart=10</td>
</tr>
<tr>
<td>azimuthStart</td>
<td>No</td>
<td></td>
<td></td>
</tr>
<tr>
<td>azimuthEnd</td>
<td>No</td>
<td>The ending azimuth in degrees of the resulting viewshed field. (default = 360)</td>
<td>azimuthEnd=350</td>
</tr>
<tr>
<td>crs</td>
<td>No</td>
<td>Coordinate reference system (CRS) for the input coordinates in format EPSG:identifier where identifier is the EPSG code for the given CRS.</td>
<td>crs=EPSG:4326</td>
</tr>
</tbody>
</table>

#### Header Parameters

#### Content

#### Example Request

**URL**

```
https://api.intermap.com/api/raster/NEXTMap-USA_DSM/viewshed?easting=-105&northing=40&radius=100&centerHeight=0.5&fieldHeight=0.3&azimuthStart=10&azimuthEnd=350&crs=EPSG:4326
```
### 4.2. POST Method

<table>
<thead>
<tr>
<th>URL Parameters</th>
</tr>
</thead>
</table>

<table>
<thead>
<tr>
<th>Header Parameters</th>
</tr>
</thead>
</table>

<table>
<thead>
<tr>
<th>Content</th>
</tr>
</thead>
</table>
| Request content is in GeoJSON format and contains a Point Feature whose geometry specifies the coordinate of the center point for the viewshed. There are also the following properties in the Feature:
- radius – radius of the viewshed from the center point in meters (required)
- centerHeight – height of the center point in meters above the terrain (optional : default = 0)
- fieldHeight – height of the values in the field above the terrain (optional : default = 0)
- azimuthStart – starting azimuth in degrees of the resulting viewshed field (optional : default = 0)
- azimuthEnd – ending azimuth in degrees of the resulting viewshed field (optional : default = 360)

The Feature may also contain a crs which is used to identify the Coordinate Reference System of the point coordinates. If no crs is given then WGS84 Geographic decimal degrees (EPSG:4326) is used.

<table>
<thead>
<tr>
<th>Example Request</th>
</tr>
</thead>
<tbody>
<tr>
<td>URL</td>
</tr>
<tr>
<td>Header Parameters</td>
</tr>
</tbody>
</table>
| Content | ```
{  "type": "Feature",  "crs": {  "type": "name",  "properties": {  "name": "EPSG:4326" }  },  "geometry": {  "type": "Point",  "coordinates": [-105, 40] }  },  "properties": {  "azimuthStart": 10,  "azimuthEnd": 350,  "centerHeight": 0.5,  "fieldHeight": 0.3,  "radius": 500  }
``` |
### 4.3. Success Responses

<table>
<thead>
<tr>
<th>Code</th>
<th>Description</th>
<th>Content Type</th>
<th>Content Description</th>
</tr>
</thead>
</table>
| 200  | Success – Response contains results | application/json | The contents are returned as a GeoJSON Polygon Feature with the geometry defining the bounding box of the result. It also contains the following properties:  
- visibility – The visible area of the viewshed in percent  
- visibleArea – The visible area of the viewshed in square meters  
- invisibleArea – The invisible area of the viewshed in square meters  
- resultUrl – Url which is used in a follow-up Viewshed Result query |

**Content Example**

```json
{
    "type": "Feature",
    "geometry": {
        "type": "Polygon",
        "coordinates": [
            [
                [-105.00585416666672, 39.99547916666667],
                [-105.00585416666672, 40.00452083333333],
                [-104.99414583333332, 40.00452083333333],
                [-104.99414583333332, 39.99547916666667],
                [-105.00585416666672, 39.99547916666667]
            ]
        ],
        "properties": {
            "invisibleArea": 889631.41,
            "resultUrl": "https://api-test.intermap.com/api/raster/NEXTMap-USA_DSM/viewshed/0174a6df720c4dfd99265e7954504b0d/result",
            "visibility": 10.76,
            "visibleArea": 107264.94
        }
    }
}
```
Code | 202
---|---
Description | Accepted – Processing will take a long time so an asynchronous job has been created

| Header Parameters | Location | Contains a URL which is used in a GET call to get the status and result URL for the job (see Asynchronous Processing) |
---|---|---|

<table>
<thead>
<tr>
<th>Content Type</th>
<th>application/json</th>
</tr>
</thead>
</table>

| Content Description | The contents simply contain the word null because the desired URL is in the header rather than the contents |
---|---|
| Content Example | null |

### 4.4. Error Responses

| Code | 401 UNAUTHORIZED |
---|---|
| Description | Error when the authentication fails possibly due to incorrect login |
| Content Type | text/xml |

| Content Example | <!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd"> <html xmlns="http://www.w3.org/1999/xhtml"> <head> <meta http-equiv="Content-Type" content="text/html; charset=iso-8859-1"/> <title>401 - Unauthorized: Access is denied due to invalid credentials.</title> <style> body{margin:0;font-size:.7em;font-family:Verdana, Arial, Helvetica, sans-serif;background:#EEE;} fieldset{padding:0 15px 10px 15px;} h1{font-size:2.4em;margin:0;color:#FFF;} h2{font-size:1.7em;margin:0;color:#CC0000;} h3{font-size:1.2em;margin:10px 0 0 0;color:#000000;} #header{width:96%;margin:0 0 0 0;padding:6px 2% 6px 2%;font-family:"trebuchet MS", Verdana, sans-serif;color:#FFF; background-color:#555555;} #content{margin:0 0 0 2%;position:relative;} .content-container{background:#FFF;width:96%;margin-top:8px;padding:10px;position:relative;} </style> </head> <body> <div id="header"><h1>Server Error</h1></div> <div id="content"> <div class="content-container"> <fieldset> <h2>401 - Unauthorized: Access is denied due to invalid credentials.</h2> <h3>You do not have permission to view this directory or page using the credentials that you supplied.</h3> </fieldset> </div> </div> </body> </html> |

| Code | 400 |
---|---|
| Description | Model is not valid. This may be due to specifying the parameters or content incorrectly. |
| Content Type | application/xml |

| Content Example | <string>Model is not valid. This error has been assigned an Incident ID of '66480-API2-TEST03-170605-193846'. Provide this Incident ID to customer support when you need assistance.</string> |

### 4.5. Viewshed Result – GET Method

Endpoint URL is extracted from the resultUrl property in the result from the Viewshed GET or POST method. It uses the same basic authentication.
### Header Parameters

| Accept: application/vnd.google.earth.kmz or Accept: image/png or Accept: image/jpeg |

### Content

### Example Request

| URL | https://api-test.intermap.com/api/raster/NEXTMap-USA_DSM/viewshed/0174a6df-720c-4dfd-9926-5e7954504b0d/result |
|-----------------------------------------------|
| Header Parameters | Accept: image/png |
| Content | |

### 4.6. Viewshed Result – POST Method

<table>
<thead>
<tr>
<th>URL Parameters</th>
</tr>
</thead>
<tbody>
<tr>
<td>Content-Type: application/json</td>
</tr>
<tr>
<td>Header Parameters</td>
</tr>
</tbody>
</table>
| Content | The content is used to specify how the resulting viewshed appears. The content may be empty to use the default visualization options. The content is simple JSON with the following elements:  
- colorVisible – color of the visible area (default = Lime)  
- colorInvisible – color of the invisible area (default = Red)  
- colorNoData – color of areas with no data (null areas) (default = rgb(0,0,128,1))  
- colorShadow – color of areas defined by depthShadowLimit (default = pink)  
- depthShadowLimit – if negative, areas which are not visible, but higher that this value below terrain are displayed in colorShadow  

Color formatting can use standard color names (e.g. red), rgb colors with transparency (e.g. rgb(0, 255, 0, 0.5)) or without transparency (e.g. rgb(0, 255, 0)), or they can use rgb hex format with 8, 6, 3, or 4 digits (e.g. #00ff00cc) |
Example Request

<table>
<thead>
<tr>
<th>URL</th>
<th><a href="https://api-test.intermap.com/api/raster/NEXTMap-USA_DSM/viewshed/0174a6df-720c-4dfd-9926-5e7954504b0d/result">https://api-test.intermap.com/api/raster/NEXTMap-USA_DSM/viewshed/0174a6df-720c-4dfd-9926-5e7954504b0d/result</a></th>
</tr>
</thead>
<tbody>
<tr>
<td>Header Parameters Content</td>
<td>Accept: image/png</td>
</tr>
<tr>
<td>Content</td>
<td></td>
</tr>
</tbody>
</table>

```
{
"colorVisible": "blue",
"colorInvisible": "yellow",
"colorNoData": "red",
"colorShadow": "green",
"depthShadowLimit": -5
}
```

4.7. Viewshed Result - Success Responses

<table>
<thead>
<tr>
<th>Code</th>
<th>200</th>
</tr>
</thead>
<tbody>
<tr>
<td>Description</td>
<td>Success – Response contains results</td>
</tr>
<tr>
<td>Content Type</td>
<td>Depends on Accept header parameter in request</td>
</tr>
<tr>
<td>Content Description</td>
<td>The contents are returned as imagery in the requested format</td>
</tr>
</tbody>
</table>

4.8. Viewshed Result - Error Responses

<table>
<thead>
<tr>
<th>Code</th>
<th>401 UNAUTHORIZED</th>
</tr>
</thead>
<tbody>
<tr>
<td>Description</td>
<td>Error when the authentication fails possibly due to incorrect login</td>
</tr>
<tr>
<td>Content Type</td>
<td>text/xml</td>
</tr>
</tbody>
</table>
| Content Example | <!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd"><html xmlns="http://www.w3.org/1999/xhtml"><head><meta http-equiv="Content-Type" content="text/html; charset=iso-8859-1"/>
<title>401 Unauthorized: Access is denied due to invalid credentials.</title>
<style type="text/css">
-->
</style></head>
<body>
<div id="header"><h1>Server Error</h1></div>
<div id="content">
<div class="content-container">
<fieldset>
<h2>401 Unauthorized: Access is denied due to invalid credentials.</h2>
<h3>You do not</h3>
</fieldset>
</div>
</div>
```
have permission to view this directory or page using the credentials that you supplied.

4.9. Asynchronous Processing

When a task will take a long time, it will be sent off for asynchronous processing rather than making the caller wait for a response. In this case, a response with a code of 202 is returned. This response contains a Location header parameter which is a URL for obtaining the status and when complete, the result.

The asynchronous status is a GET method that allows the user to check the status of the asynchronous job, and when complete, it gives the URL for the result. It uses the same basic authentication as the other methods.

<table>
<thead>
<tr>
<th>URL Parameters</th>
<th>None</th>
</tr>
</thead>
<tbody>
<tr>
<td>Header Parameters</td>
<td>None</td>
</tr>
<tr>
<td>Content</td>
<td>None</td>
</tr>
<tr>
<td>Example Request</td>
<td></td>
</tr>
<tr>
<td>URL</td>
<td><code>https://api-test.intermap.com/tasks/02237c3e-4f0e-e711-80c3-000d3a604bff/status</code></td>
</tr>
<tr>
<td>Header Parameters</td>
<td></td>
</tr>
<tr>
<td>Content</td>
<td></td>
</tr>
<tr>
<td>Success Responses</td>
<td></td>
</tr>
<tr>
<td>Code</td>
<td>200</td>
</tr>
<tr>
<td>Description</td>
<td>Success – Status check is successful, but the job is still processing</td>
</tr>
<tr>
<td>Content Type</td>
<td>application/json</td>
</tr>
<tr>
<td>Content Description</td>
<td>The contents are simple JSON with a “location” element echoing back the status URL and a “status” element that gives the current status of the processing.</td>
</tr>
</tbody>
</table>
Content Example

```json
{
"location": "https://api-test.intermap.com/tasks/02237c3e-4f0e-e711-80c3-000d3a604bff/status",
"status": "processing"
}
```

**Code** 303  
**Description** Redirect – Status check is successful and the job is complete  
**Header Parameters**  
**Location** – URL that contains the result of the processing  
**Content Type** No Content  
**Content Description** There is no content  
**Content Example**

---

### Failure Responses

**Code** 401 UNAUTHORIZED  
**Description** Error when the authentication fails possibly due to incorrect login  
**Content Type** text/xml  
**Content Example**

```xml
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd"> <html xmlns="http://www.w3.org/1999/xhtml"> <head> <meta http-equiv="Content-Type" content="text/html; charset=iso-8859-1"/> <title>401 - Unauthorized: Access is denied due to invalid credentials.</title> <style type="text/css"> <!--body{margin:0;font-size:.7em;font-family:Verdana, Arial, Helvetica, sans-serif;background:#EEEEEE;}fieldset{padding:0 15px 10px 15px;} h1{font-size:2.4em;margin:0;color:#FFF;} h2{font-size:1.7em;margin:0;color:#CC0000;} h3{font-size:1.2em;margin:10px 0 0 0;color:#000000;}#header{width:96%;margin:0 0 0 0;padding:6px 2% 6px 2%;font-family:"trebuchet MS", Verdana, sans-serif;color:#FFF; background-color:#555555;}#content{margin:0 0 0 2%;position:relative;} .content-container{background:#FFF;width:96%;margin-top:8px;padding:10px;position:relative;} --> </style> </head> <body> <div id="header"> <div id="content"> <div class="content-container"> <fieldset> <h2>401 - Unauthorized: Access is denied due to invalid credentials.</h2> <h3>You do not have permission to view this directory or page using the credentials that you supplied.</h3> </fieldset> </div> </div> </div> </body> </html>
```

**Code** 400  
**Description** Model is not valid. This may be due to specifying the parameters or content incorrectly.  
**Content Type** application/xml  
**Content Example**

```xml
<string>Model is not valid. This error has been assigned an Incident ID of '66480-API2-TEST03-170605-193846'. Provide this Incident ID to customer support when you need assistance.</string>
```
The asynchronous result is a GET method that simply gets back the result from an asynchronous processing job with the URL given in the asynchronous status call.

<table>
<thead>
<tr>
<th>URL Parameters</th>
<th>None</th>
</tr>
</thead>
<tbody>
<tr>
<td>Header Parameters</td>
<td>None</td>
</tr>
<tr>
<td>Content</td>
<td>None</td>
</tr>
</tbody>
</table>

**Example Request**

<table>
<thead>
<tr>
<th>URL</th>
<th><a href="https://api.intermap.com/tasks/02237c3e-4f0e-e711-80c3-000d3a604bff">https://api.intermap.com/tasks/02237c3e-4f0e-e711-80c3-000d3a604bff</a></th>
</tr>
</thead>
<tbody>
<tr>
<td>Header Parameters</td>
<td></td>
</tr>
<tr>
<td>Content</td>
<td></td>
</tr>
</tbody>
</table>

**Success Responses**

<table>
<thead>
<tr>
<th>Code</th>
<th>200</th>
</tr>
</thead>
<tbody>
<tr>
<td>Description</td>
<td>Success – Result returned</td>
</tr>
<tr>
<td>Content Type</td>
<td>application/json</td>
</tr>
<tr>
<td>Content Description</td>
<td>The contents are the same as if they were returned without an asynchronous call. The contents are returned as a GeoJSON Polygon Feature with the geometry defining the bounding box of the result. It also contains the following properties:</td>
</tr>
<tr>
<td>Visibility</td>
<td>The visible area of the viewshed in percent</td>
</tr>
<tr>
<td>Visible Area</td>
<td>The visible area of the viewshed in square meters</td>
</tr>
<tr>
<td>Invisible Area</td>
<td>The invisible area of the viewshed in square meters</td>
</tr>
<tr>
<td>Result URL</td>
<td>Url which is used in a follow-up Viewshed Result query</td>
</tr>
</tbody>
</table>

| Content Example | { |
| | "type": "Feature", |
| | "geometry": [ |
| | "type": "Polygon", |
| | "coordinates": [ |
| | [ |
| | -105.00585416666672, 39.99547916666667 |
| | 39.99547916666667 |
| |], |
| | [ |
| | -105.00585416666672, 40.00452083333333 |
| | 40.004520833333333 ] |} |
Failure Responses

<table>
<thead>
<tr>
<th>Code</th>
<th>Description</th>
<th>Content Type</th>
<th>Content Example</th>
</tr>
</thead>
<tbody>
<tr>
<td>401</td>
<td>Unauthorized: Access is denied due to invalid credentials.</td>
<td>text/xml</td>
<td><code>&lt;html xmlns=&quot;http://www.w3.org/1999/xhtml&quot;&gt;</code></td>
</tr>
<tr>
<td>400</td>
<td>Model is not valid. This may be due to specifying the parameters or content incorrectly.</td>
<td>application/xml</td>
<td><code>&lt;html&gt;</code></td>
</tr>
</tbody>
</table>
5. Link Profile

A link profile is a set of points along a straight line from one location to another that combines a terrain profile with basic Fresnel zone information. The terrain profile component samples the elevation along the line. The Fresnel zone is based on an input frequency and k-factor and shows the zone in which obstructions are likely to contribute to interference in a signal transmission. Additionally, a link profile user can specify a corridor width to draw multiple parallel elevation profiles to assess the line-of-sight within a transmission corridor. Along-track step size between the start and end points can also be specified. Link profiles are often plotted as shown in the example below.

![Link Profile Diagram](image)

Gets a link profile between two points on a terrain data source.

<table>
<thead>
<tr>
<th>Endpoint URL</th>
<th><a href="https://api.intermap.com/linkpro/profile">https://api.intermap.com/linkpro/profile</a></th>
</tr>
</thead>
<tbody>
<tr>
<td>Authentication</td>
<td>Basic</td>
</tr>
<tr>
<td>Methods</td>
<td>GET, POST</td>
</tr>
</tbody>
</table>
## 5.1. GET Method

<table>
<thead>
<tr>
<th>Name</th>
<th>Required</th>
<th>Description</th>
<th>Example</th>
</tr>
</thead>
<tbody>
<tr>
<td>startEasting</td>
<td>Yes</td>
<td>Easting and northing coordinates of the start point for the profile.</td>
<td>startEasting=-122.3096</td>
</tr>
<tr>
<td>startNorthing</td>
<td></td>
<td>If the crs parameter is specified, the coordinate units are determined</td>
<td>startNorthing=37.9020</td>
</tr>
<tr>
<td></td>
<td></td>
<td>by the crs. If no crs parameter is given, then the coordinates are in WGS84</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Geographic decimal degrees (EPSG:4326).</td>
<td></td>
</tr>
<tr>
<td>endEasting</td>
<td></td>
<td>Easting and northing coordinates of the end point for the profile.</td>
<td>endEasting=-122.3093</td>
</tr>
<tr>
<td>endNorthing</td>
<td></td>
<td>If the crs parameter is specified, the coordinate units are determined</td>
<td>endNorthing=37.8954</td>
</tr>
<tr>
<td></td>
<td></td>
<td>by the crs. If no crs parameter is given, then the coordinates are in WGS84</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Geographic decimal degrees (EPSG:4326).</td>
<td></td>
</tr>
<tr>
<td>startAntennaHeight</td>
<td>Yes</td>
<td>Antenna height for each of the start and end points above ground [meters]</td>
<td>startAntennaHeight=5</td>
</tr>
<tr>
<td>endAntennaHeight</td>
<td></td>
<td></td>
<td>endAntennaHeight=1</td>
</tr>
<tr>
<td>corridorWidth</td>
<td>Yes</td>
<td>Width of Corridor [meters]</td>
<td>corridorWidth=4</td>
</tr>
<tr>
<td>corridorStep</td>
<td>Yes</td>
<td>Step size between parallel profiles in corridor calculation [meter].</td>
<td>corridorStep=1</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Must be an integer divisor of corridorWidth.</td>
<td></td>
</tr>
<tr>
<td>profileStep</td>
<td>Yes</td>
<td>Sampling step size along the profile between the start and end points</td>
<td>profileStep=5</td>
</tr>
<tr>
<td></td>
<td></td>
<td>[meters].</td>
<td></td>
</tr>
<tr>
<td>frequency</td>
<td>Yes</td>
<td>radio link frequency [MHz]</td>
<td>frequency=2400</td>
</tr>
<tr>
<td>kFactor</td>
<td>No</td>
<td>K-Factor used in LOS calculation.</td>
<td>kfactor=1.333</td>
</tr>
<tr>
<td></td>
<td></td>
<td>If not specified, default is 1.333.</td>
<td></td>
</tr>
<tr>
<td>dataSource</td>
<td>Yes</td>
<td>An elevation data source used for terrain profiles calculation. Values</td>
<td>dataSource=Sample-</td>
</tr>
<tr>
<td></td>
<td></td>
<td>for this parameter are any of the data source IDs returned from a Terrain</td>
<td>NEXTMapOne-Elevation-DSM</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Sources call. This should typically be a DSM layer.</td>
<td></td>
</tr>
<tr>
<td>dataSourceAntennaHeight</td>
<td>No</td>
<td>An elevation data source used for antenna height calculation. Values</td>
<td>dataSourceAntennaHeight=</td>
</tr>
<tr>
<td></td>
<td></td>
<td>for this parameter are any of the data source IDs returned from a Terrain</td>
<td>Sample-NEXTMapOne-Elevation-</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Sources call. This allows for</td>
<td>DTM</td>
</tr>
</tbody>
</table>
an above-ground antenna height as opposed to an above-obstruction antenna height if a DTM data source is specified here. If it is not specified, dataSource is used.

<table>
<thead>
<tr>
<th>crs</th>
<th>No</th>
</tr>
</thead>
<tbody>
<tr>
<td>Coordinate reference system (CRS) for the input coordinates in format EPSG:identifier where identifier is the EPSG code for the given CRS. Default if not specified is EPSG:4326.</td>
<td>crs=EPSG:4326</td>
</tr>
</tbody>
</table>

### Header Parameters

### Content

### Example Request

<table>
<thead>
<tr>
<th>URL</th>
</tr>
</thead>
<tbody>
<tr>
<td><a href="https://api.intermap.com/linkpro/profile?startEasting=-122.3096&amp;startNorthing=37.902&amp;endEasting=-122.3093&amp;endNorthing=37.8954&amp;corridorStep=1&amp;startAntennaHeight=5&amp;endAntennaHeight=1&amp;corridorWidth=4&amp;profileStep=5&amp;frequency=2400&amp;dataSource=Sample-NEXTMapOne-Elevation-DSM&amp;kFactor=1.333">https://api.intermap.com/linkpro/profile?startEasting=-122.3096&amp;startNorthing=37.902&amp;endEasting=-122.3093&amp;endNorthing=37.8954&amp;corridorStep=1&amp;startAntennaHeight=5&amp;endAntennaHeight=1&amp;corridorWidth=4&amp;profileStep=5&amp;frequency=2400&amp;dataSource=Sample-NEXTMapOne-Elevation-DSM&amp;kFactor=1.333</a></td>
</tr>
</tbody>
</table>

### 5.2. POST Method

<table>
<thead>
<tr>
<th>URL Parameters</th>
</tr>
</thead>
<tbody>
<tr>
<td>Content-Type: application/json</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Header Parameters</th>
</tr>
</thead>
<tbody>
<tr>
<td>Accept: application/json</td>
</tr>
<tr>
<td>X-Task-Sync-Timeout: 120 (value is timeout in seconds)</td>
</tr>
</tbody>
</table>
Content

Request content is in GeoJSON format and contains two Point Features in a FeatureCollection whose geometry specify the coordinates of the start and end points for the link profile. Each point must also contain an antennaHeight property. There are also the following properties in the Feature Collection:

- corridorWidth – width of the corridor in meters (required)
- corridorStep – step size between parallel profiles in corridor calculation in meters. Note that this must be an integer divisor of corridorWidth (required)
- profileStep – sampling step size along the profile between the start and end points in meters (required)
- frequency – radio link frequency in MHz (required)
- kFactor – K-factor used in LOS calculation (optional: default = 1.333)
- dataSource – elevation data source used for terrain profiles calculation. Values for this parameter are any of the data source IDs returned from a Terrain Sources call. This should typically be a DSM layer. (required)
- dataSourceAntennaHeight – elevation data source used for antenna height calculation. Values for this parameter are any of the data source IDs returned from a Terrain Sources call. This allows for an above-ground antenna height as opposed to an above-obstruction antenna height if a DTM source is specified here. (optional: default = dataSource)

The Feature Collection may also contain a crs which is used to identify the Coordinate Reference System of the point coordinates. If no crs is given, WGS84 Geographic decimal degrees (EPSG:4326) is used.

Example Request

<table>
<thead>
<tr>
<th>URL</th>
<th><a href="https://api.intermap.com/linkpro/profile">https://api.intermap.com/linkpro/profile</a></th>
</tr>
</thead>
<tbody>
<tr>
<td>Header Parameters</td>
<td>Content-Type: application/json</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Content</th>
</tr>
</thead>
<tbody>
<tr>
<td>{</td>
</tr>
</tbody>
</table>
|   "type": "FeatureCollection",
|   "crs": {     |
|     "type": "name",
|     "properties": { |
|       "name": "EPSG:4326"
|     }           |
|   }             |
|   "features": [ |
|     {         |
|       "type": "Feature",
|       "geometry": { |
|         "type": "Point",
|         "coordinates": [-122.3096, 37.902]
|       }         |
|       "properties": { |
|         "antennaHeight": 5
|       }           |
|     },          |
|     {         |
|       "type": "Feature",
|       "geometry": { |
|     }         |
| }               |
5.3. Success Responses

<table>
<thead>
<tr>
<th>Code</th>
<th>200</th>
</tr>
</thead>
<tbody>
<tr>
<td>Description</td>
<td>Success – Response contains results</td>
</tr>
<tr>
<td>Content Type</td>
<td>application/json</td>
</tr>
</tbody>
</table>
| Content Description | The contents are returned as a json Feature Collection with multiple point features where the geometry defines the coordinates of the point in the profile. There will be features for each point in the parallel lines of the corridorWidth, labelled with IDs of L*, R*, or 0* depending on if the profile is on the left (L) or right (R) of the center profile (0*). If there are more than 3 parallel profiles, they will be labeled moving outwards from the center with L1, L2, etc. and R1, R2, etc. All point IDs will then be followed with a numeric ID, starting at 0 and increasing by 1 from the start of the profile. For example, the center profile will be labeled 0.0, 0.1, 0.2, etc. and the first left profile will be labelled L1.0, L1.1, L1.2, etc. All points will contain properties distance and value, where distance is the distance in meters since the last point (ex. 0.0 to 0.1) and value is the sampled terrain height at that point in meters. Additionally, the center profile will contain the following properties:  
  - fresnelZone – The height in meters of the first Fresnel zone at that point  
  - lineOfSight – The height in meters of the direct line of sight vector at that point  
  - maxValue – The maximum terrain height in meters from a cross-section of the parallel profiles at that point  
  - minValue – The minimum terrain height in meters from a cross-section of the parallel profiles at that point The overall FeatureCollection also contains the following properties:  
  - endAntenaElevation – The height in meters of the end point of the profile  
  - startAntenaElevation – The height in meters of the start point of the profile  
  - length – the overall length of the profile in meters  
  - resultUrl – Url which is used in a follow-up Linkpro Result query |
Content Example

```json
{
   "type": "FeatureCollection",
   "features": [ 
   {
      "type": "Feature",
      "id": "L2.0",
      "geometry": {
         "type": "Point",
         "coordinates": [
            -122.30962272565768,
            37.901999389489326
         ]
      },
      "properties": {
         "distance": 0,
         "value": 29.60384
      }
   },
   {
      "type": "Feature",
      "id": "0.0",
      "geometry": {
         "type": "Point",
         "coordinates": [
            -122.3096,
            37.902
         ]
      },
      "properties": {
         "distance": 0,
         "fresnelZone": 34.603840000000005,
         "lineOfSight": 34.603840000000005,
         "maxValue": 29.60384,
         "minValue": 29.60384,
         "value": 29.60384
      }
   },
   {
      "type": "Feature",
      "id": "0.1",
      "geometry": {
         "type": "Point",
         "coordinates": [
            -122.30959795354305,
            37.901955020082362
         ]
      },
      "properties": {
         "distance": 5,
         "fresnelZone": 33.705601743977176,
         "lineOfSight": 34.493470340967178,
         "maxValue": 29.60384,
         "minValue": 29.60384,
         "value": 29.60384
      }
   }
   ]
}
```

```json
{
    "type": "Feature",
    "id": "R2.146",
    "geometry": {
        "type": "Point",
        "coordinates": [
            -122.30927852003252,
            37.895428088024111
        ]
    },
    "properties": {
        "distance": 5,
        "value": 16.97756
    }
},
{
    "type": "Feature",
    "id": "R2.147",
    "geometry": {
        "type": "Point",
        "coordinates": [
            -122.30927727636852,
            37.8954006866731
        ]
    },
    "properties": {
        "distance": 3.0391387143232813,
        "value": 16.97756
    }
}
```

<table>
<thead>
<tr>
<th>Code</th>
<th>202</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Description</strong></td>
<td>Accepted – Processing will take a long time so an asynchronous job has been created</td>
</tr>
<tr>
<td><strong>Header Parameters</strong></td>
<td><strong>Location</strong> – Contains a URL which is used in a GET call to get the status and result URL for the job (see <a href="#">Asynchronous Processing</a>)</td>
</tr>
<tr>
<td><strong>Content Type</strong></td>
<td>application/json</td>
</tr>
<tr>
<td><strong>Content Description</strong></td>
<td>The contents simply contain the word null because the desired URL is in the header rather than the contents</td>
</tr>
<tr>
<td><strong>Content Example</strong></td>
<td>null</td>
</tr>
</tbody>
</table>
5.4. Error Responses

<table>
<thead>
<tr>
<th>Code</th>
<th>Description</th>
<th>Content Type</th>
<th>Content Example</th>
</tr>
</thead>
<tbody>
<tr>
<td>401</td>
<td>UNAUTHORIZED Error when the authentication fails possibly due to incorrect login</td>
<td>text/xml</td>
<td><code>&lt;!DOCTYPE html PUBLIC &quot;-//W3C//DTD XHTML 1.0 Strict//EN&quot; &quot;http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd&quot;&gt; &lt;html xmlns=&quot;http://www.w3.org/1999/xhtml&quot;&gt; &lt;head&gt; &lt;meta http-equiv=&quot;Content-Type&quot; content=&quot;text/html; charset=iso-8859-1&quot;/&gt; &lt;title&gt;401 - Unauthorized: Access is denied due to invalid credentials.&lt;/title&gt; &lt;style type=&quot;text/css&quot;&gt; body{margin:0;font-size:7em;font-family:Verdana, Arial, Helvetica, sans-serif;background:#EEEEEE;} fieldset{padding:0 15px 15px;}; h1{font-size:2.4em;margin:0;color:#FFF;} h2{font-size:1.7em;margin:0;color:#CC0000;} h3{font-size:1.2em;margin:10px 0 0 0;color:#000000;} #header{width:96%;margin:0 0 0 0;padding:6px 2% 6px 2%;font-family:&quot;trebuchet MS&quot;, Verdana, sans-serif;color:#FFF;background-color:#555555;} #content{margin:0 0 0 2%;position:relative;} .content-container{background:#FFF;width:96%;margin-top:8px;padding:10px;position:relative;} --&lt;/style&gt; &lt;/head&gt; &lt;body&gt; &lt;div id=&quot;header&quot;&gt;&lt;h1&gt;Server Error&lt;/h1&gt;&lt;/div&gt; &lt;div id=&quot;content&quot;&gt;&lt;div class=&quot;content-container&quot;&gt;&lt;fieldset&gt;&lt;h2&gt;401 - Unauthorized: Access is denied due to invalid credentials.&lt;/h2&gt;&lt;h3&gt;You do not have permission to view this directory or page using the credentials that you supplied.&lt;/h3&gt;&lt;/fieldset&gt;&lt;/div&gt;&lt;/div&gt;&lt;/html&gt;</code></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Code</th>
<th>Description</th>
<th>Content Type</th>
<th>Content Example</th>
</tr>
</thead>
<tbody>
<tr>
<td>400</td>
<td>Model is not valid. This may be due to specifying the parameters or content incorrectly.</td>
<td>application/xml</td>
<td><code>&lt;string&gt;Model is not valid. This error has been assigned an Incident ID of '66480-API2-TEST03-170605-193846'. Provide this Incident ID to customer support when you need assistance.&lt;/string&gt;</code></td>
</tr>
</tbody>
</table>

5.5. Link Profile Result – GET Method

Endpoint URL is extracted from the resultUrl property in the result from the Link Profile GET or POST method. It uses the same basic authentication.

**URL Parameters**

The parameters are used to specify how the resulting link profile appears. All parameters are optional. Note that the color elements are applicable only for Accept: application/vnd.google-earth.kml+xml format.

<table>
<thead>
<tr>
<th>Name</th>
<th>Required</th>
<th>Description</th>
<th>Example</th>
</tr>
</thead>
<tbody>
<tr>
<td>lineOfSight</td>
<td>No</td>
<td>TRUE/FALSE setting controlling if line of sight geometry is returned (default = TRUE)</td>
<td>lineOfSight=TRUE</td>
</tr>
<tr>
<td>fresnelZone</td>
<td>No</td>
<td>TRUE/FALSE setting controlling if line of sight geometry is returned (default = TRUE)</td>
<td>fresnelZone=TRUE</td>
</tr>
<tr>
<td>minProfile</td>
<td>No</td>
<td>TRUE/FALSE setting controlling if line of sight geometry is returned (default = TRUE)</td>
<td>minProfile=TRUE</td>
</tr>
<tr>
<td>Parameter</td>
<td>Required</td>
<td>Description</td>
<td>Value</td>
</tr>
<tr>
<td>-----------------------</td>
<td>----------</td>
<td>-----------------------------------------------------------------------------</td>
<td>--------------------</td>
</tr>
<tr>
<td>maxProfile</td>
<td>No</td>
<td>TRUE/FALSE setting controlling if line of sight geometry is returned</td>
<td>maxProfile=TRUE</td>
</tr>
<tr>
<td>corridorProfile</td>
<td>No</td>
<td>TRUE/FALSE setting controlling if line of sight geometry is returned</td>
<td>corridorProfile=TRUE</td>
</tr>
<tr>
<td>lineOfSightColor</td>
<td>No</td>
<td>Color of line of sight (default = Yellow)</td>
<td>lineOfSightColor=Yellow</td>
</tr>
<tr>
<td>fresnelZoneColor</td>
<td>No</td>
<td>Color of Fresnel zone (default = Cyan)</td>
<td>fresnelZoneColor=Cyan</td>
</tr>
<tr>
<td>minProfileColor</td>
<td>No</td>
<td>Color of minimum terrain profile (default = Green)</td>
<td>minProfileColor=Green</td>
</tr>
<tr>
<td>maxProfileColor</td>
<td>No</td>
<td>Color of maximum terrain profile (default = Red)</td>
<td>maxProfileColor=Red</td>
</tr>
<tr>
<td>corridorProfilesColor</td>
<td>No</td>
<td>Color of parallel corridor terrain profiles (default = Blue)</td>
<td>corridorProfilesColor=Blue</td>
</tr>
</tbody>
</table>

Color formatting can use standard color names (e.g. red), rgb colors with transparency (e.g. rgb(0, 255, 0, 0.5)) or without transparency (e.g. rgb(0, 255, 0)), or they can use rgb hex format with 8, 6, 3, or 4 digits (e.g. #00ff00cc)

**Header Parameters**

- Accept: application/vnd.google-earth.kml+xml or application/json

**Content**

**Example Request**

- **URL**: https://api.intermap.com/linkpro/profile/e808a194-5a2b-4d55-b4cb-d6320dd22585/result
  - **Header Parameters**: Accept: application/json
  - **Content**: 

---

Copyright © 2019 by Intermap Technologies. All rights reserved
Document Version: 2017-05-15
5.6. Linkpro Result – POST Method

<table>
<thead>
<tr>
<th>URL Parameters</th>
<th>Accept: application/vnd.google-earth.kml+xml or application/json</th>
</tr>
</thead>
<tbody>
<tr>
<td>Header Parameters</td>
<td>Accept: application/json</td>
</tr>
</tbody>
</table>
| Content | The content is used to specify how the resulting link profile appears. The content may be empty to use the default visualization options. Note that the color elements are applicable only for Accept: application/vnd.google-earth.kml+xml format. The content is simple JSON with the following elements:
    - lineOfSight – controls if the line of sight geometry is returned (default = TRUE)
    - fresnelZone – controls if the Fresnel zone geometry is returned (default = TRUE)
    - minProfile – controls if the minimum terrain profile is returned (default = TRUE)
    - maxProfile – controls if the maximum terrain profile is returned (default = TRUE)
    - corridorProfile – controls if the corridor terrain profiles are returned (default = TRUE)
    - lineOfSightColor – color of line of sight (default is Yellow)
    - fresnelZoneColor – color of Fresnel zone profile (default is Cyan)
    - minProfileColor – color of the minimum terrain profile (default is Green)
    - maxProfileColor – color of the maximum terrain profile (default is Red)
    - corridorProfilesColor – color of parallel corridor profiles (default is Blue)

Color formatting can use standard color names (e.g. red), rgb colors with transparency (e.g. rgb(0, 255, 0, 0.5)) or without transparency (e.g. rgb(0, 255, 0)), or they can use rgb hex format with 8, 6, 3, or 4 digits (e.g. #00ff00cc)

<table>
<thead>
<tr>
<th>Example Request</th>
<th>URL</th>
<th><a href="https://api.intermap.com/linkpro/profile/e808a194-5a2b-4d55-b4cb-d6320dd22585/result">https://api.intermap.com/linkpro/profile/e808a194-5a2b-4d55-b4cb-d6320dd22585/result</a></th>
</tr>
</thead>
<tbody>
<tr>
<td>Header Parameters</td>
<td>Accept: application/json</td>
<td></td>
</tr>
</tbody>
</table>
| Content | {  
  "lineOfSight" : "true|false",
  "fresnelZone" : "true|false",
  "minProfile" : "true|false",
  "maxProfile" : "true|false",
  "corridorProfiles" : "true|false",
  "lineOfSightColor" : "yellow",
  "fresnelZoneColor" : "cyan",
  "minProfileColor" : "green",
  "maxProfileColor" : "red",
  "corridorProfilesColor" : "blue"
} |
### 5.7. Linkpro Result - Success Responses

<table>
<thead>
<tr>
<th>Code</th>
<th>Description</th>
<th>Content Type</th>
<th>Content Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>200</td>
<td>Success – Response contains results</td>
<td>Depends on Accept header parameter in request</td>
<td>The contents are in the same format as the Link Profile method, but only contains the components that were selected in the request (e.g. only minimum and maximum of center profile). For JSON, the color definitions are ignored.</td>
</tr>
</tbody>
</table>

**Content Example**

```json
{
  "type": "FeatureCollection",
  "features": [
    {
      "type": "Feature",
      "id": "0.0",
      "geometry": {
        "type": "Point",
        "coordinates": [-122.3096, 37.902]
      },
      "properties": {
        "distance": 0,
        "maxValue": 29.60384,
        "minValue": 29.60384
      }
    },
    {
      "type": "Feature",
      "id": "0.1",
      "geometry": {
        "type": "Point",
        "coordinates": [-122.30959795354305, 37.901955020082362]
      },
      "properties": {
        "distance": 5,
        "maxValue": 29.60384,
        "minValue": 29.60384
      }
    },
    ..................,
    {
      "type": "Feature",
      "id": "0.146",
      "geometry": {
        "type": "Point",
        "coordinates": [-122.30927852003252, 37.895428088024111]
      },
      "properties": {
        "distance": 10,
        "maxValue": 29.60384,
        "minValue": 29.60384
      }
    }
  ]
}
```
KML example:

```xml
<?xml version="1.0" encoding="UTF-8"?>
<kml xmlns="http://www.opengis.net/kml/2.2" xmlns:gx="http://www.google.com/kml/ext/2.2" xmlns:kml="http://www.opengis.net/kml/2.2" xmlns:atom="http://www.w3.org/2005/Atom">
<Document>
<open>1</open>
<name />  
<Style id="profile_sub">
<LineStyle>
<color>ffff0000</color>
<width>2</width>
</LineStyle>
</Style>
<Style id="profile_max">
<LineStyle>
<color>ff0000ff</color>
<width>2</width>
</LineStyle>
</Style>
<Style id="profile_min">
<LineStyle>
<color>ff008000</color>
<width>2</width>
</LineStyle>
</Style>
</Document>
</kml>
```
<LineStyle>
</LineStyle>
<LineStyle id="line_visibility">

<LineStyle>
<color>ff00ffff</color>
<width>2</width>
</LineStyle>
</LineStyle>
<LineStyle id="fresnel_visibility">

<LineStyle>
<color>ffffffff</color>
<width>2</width>
</LineStyle>
</LineStyle>
<LineStyle id="profile_avg">

<LineStyle>
<color>ff00ffff</color>
<width>4</width>
</LineStyle>
</LineStyle>
<LineStyle id="profile_line-of-sight">

<LineStyle>
<color>ffffffff</color>
<width>4</width>
</LineStyle>
</LineStyle>
<LineStyle id="profile_landuse">

<LineStyle>
<color>00000000</color>
<width>4</width>
</LineStyle>
</LineStyle>
<LineStyle id="profile_datasource">

<LineStyle>
<color>00000000</color>
<width>4</width>
</LineStyle>
</LineStyle>
<LineStyle id="profile_quality">

<LineStyle>
<color>00000000</color>
<width>4</width>
</LineStyle>
</LineStyle>
<Style id="point_A">
<IconStyle>

<Icon>
<href>http://maps.google.com/mapfiles/kml/paddle/A.png</href>
</Icon>
</IconStyle>
</Style>
<Style id="point_B">
<IconStyle>

<Icon>
<href>http://maps.google.com/mapfiles/kml/paddle/B.png</href>
</Icon>
</IconStyle>
</Style>
5.8. Linkpro Result - Error Responses

<table>
<thead>
<tr>
<th>Code</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>401 UNAUTHORIZED</td>
<td>Error when the authentication fails possibly due to incorrect login</td>
</tr>
</tbody>
</table>
5.9. Asynchronous Processing

When a task will take a long time, it will be sent off for asynchronous processing rather than making the caller wait for a response. In this case, a response with a code of 202 is returned. This response contains a Location header parameter which is a URL for obtaining the status and when complete, the result.

The asynchronous status is a GET method that allows the user to check the status of the asynchronous job, and when complete, it gives the URL for the result. It uses the same basic authentication as the other methods.

<table>
<thead>
<tr>
<th>URL Parameters</th>
<th>None</th>
</tr>
</thead>
<tbody>
<tr>
<td>Header Parameters</td>
<td>None</td>
</tr>
<tr>
<td>Content</td>
<td>None</td>
</tr>
<tr>
<td>Example Request</td>
<td><strong>URL</strong></td>
</tr>
<tr>
<td>Success Responses</td>
<td>Code</td>
</tr>
<tr>
<td></td>
<td>Description</td>
</tr>
<tr>
<td></td>
<td>Content Type</td>
</tr>
</tbody>
</table>
### Content Description

The contents are simple JSON with a “location” element echoing back the status URL and a “status” element that gives the current status of the processing.

#### Content Example

```json
{
  "location": "https://api.intermap.com/linkpro/tasks/02237c3e-4f0e-e711-80c3-000d3a604bff/status",
  "status": "processing"
}
```

### Code 303

#### Description
Redirect – Status check is successful and the job is complete

#### Header Parameters

- **Location** – URL that contains the result of the processing

#### Content Type

- No Content

#### Content Description

There is no content

#### Content Example


### Failure Responses

#### Code 401 UNAUTHORIZED

#### Description
Error when the authentication fails possibly due to incorrect login

#### Content Type
text/xml

#### Content Example

```xml
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
  <meta http-equiv="Content-Type" content="text/html; charset=iso-8859-1"/>
  <title>401 Unauthorized: Access is denied due to invalid credentials.</title>
  <style type="text/css">
    body{margin:0;font-size:.7em;font-family:Verdana, Arial, Helvetica, sans-serif;background:#EEEEEE;}
    fieldset{padding:0 15px 10px 15px;} h1{font-size:2.4em;margin:0;color:#FFF;} h2{font-size:1.7em;margin:0;color:#CCCC00;} h3{font-size:1.2em;margin:10px 0 0 0;color:#000000;} #header{width:96%;margin:0 0 0 0;padding:8px;position:relative;}.content-container{background:#FFF;width:96%;margin-top:8px;padding:10px;position:relative;} --></style>

<body>
  <h1>Server Error</h1>
  <div>401 - Unauthorized: Access is denied due to invalid credentials.</div>
  <div>You do not have permission to view this directory or page using the credentials that you supplied.</div>
</body>
</html>
```

#### Code 400

#### Description
Model is not valid. This may be due to specifying the parameters or content incorrectly.

#### Content Type

- application/xml

#### Content Example

```xml
<string>Model is not valid. This error has been assigned an Incident ID of '66480-API2-TEST03-170605-193846'. Provide this Incident ID to customer support when you need assistance.</string>
```
The asynchronous result is a GET method that simply gets back the result from an asynchronous processing job with the URL given in the asynchronous status call.

<table>
<thead>
<tr>
<th>URL Parameters</th>
<th>None</th>
</tr>
</thead>
<tbody>
<tr>
<td>Header Parameters</td>
<td>None</td>
</tr>
<tr>
<td>Content</td>
<td>None</td>
</tr>
</tbody>
</table>

### Example Request

| URL | https://api.intermap.com/linkpro/tasks/02237c3e-4f0e-e711-80c3-000d3a604bff |
| Header Parameters | |
| Content | |

### Success Responses

| Code | 200 |
| Description | Success – Result returned |
| Content Type | application/json |
| Content Description | The contents are returned as a json Feature Collection with multiple point features where the geometry defines the coordinates of the point in the profile. There will be features for each point in the parallel lines of the corridorWidth, labelled with IDs of L*, R*, or 0* depending on if the profile is on the left (L) or right (R) of the center profile (0*). If there are more than 3 parallel profiles, they will be labeled moving outwards from the center with L1, L2, etc. and R1, R2, etc. All point IDs will then be followed with a numeric ID, starting at 0 and increasing by 1 from the start of the profile. For example, the center profile will be labeled 0.0, 0.1, 0.2, etc. and the first left profile will be labelled L1.0, L1.1, L1.2, etc. All points will contain properties distance and value, where distance is the distance in meters since the last point (ex. 0.0 to 0.1) and value is the sampled terrain height at that point in meters. Additionally, the center profile will contain the following properties: |
| | • fresnelZone – The height in meters of the first Fresnel zone at that point |
• `lineOfSight` – The height in meters of the direct line of sight vector at that point
• `maxValue` – The maximum terrain height in meters from a cross-section of the parallel profiles at that point
• `minValue` – The minimum terrain height in meters from a cross-section of the parallel profiles at that point

The overall FeatureCollection also contains the following properties:
• `endAntenaElevation` – The height in meters of the end point of the profile
• `startAntenaElevation` – The height in meters of the start point of the profile
• `length` – the overall length of the profile in meters
• `resultUrl` – Url which is used in a follow-up Linkpro Result query

---

**Content Example**

```json
{
  "type": "FeatureCollection",
  "features": [
    {
      "type": "Feature",
      "id": "L2.0",
      "geometry": {
        "type": "Point",
        "coordinates": [-122.30962272565768, 37.901999389489326]
      },
      "properties": {
        "distance": 0,
        "value": 29.60384
      }
    },
    {
      "type": "Feature",
      "id": "0.0",
      "geometry": {
        "type": "Point",
        "coordinates": [-122.3096, 37.902]
      },
      "properties": {
        "distance": 0,
        "fresnelZone": 34.603840000000005,
        "lineOfSight": 34.603840000000005,
        "maxValue": 29.6038,
        "minValue": 29.6038,
        "value": 29.60384
      }
    }
  ]
}
```
"type": "Feature",
"id": "R2.146",
"geometry": {
  "type": "Point",
  "coordinates": [
    -122.30927852003252,
    37.895428088024111
  ]
},
"properties": {
  "distance": 5,
  "value": 16.97756
}
},

{"type": "Feature",
"id": "R2.147",
"geometry": {
  "type": "Point",
  "coordinates": [
    -122.30927727636852,
    37.8954006866731
  ]
},
"properties": {
  "distance": 3.0391387143232813,
  "value": 16.97756
}
},

{"type": "Feature",
"id": "R2.148",
"geometry": {
  "type": "Point",
  "coordinates": [
    -122.30927535354305,
    37.901955020082362
  ]
},
"properties": {
  "distance": 5,
  "lineOfSight": 34.493470340967178,
  "maxValue": 29.60384,
  "minValue": 29.60384,
  "value": 29.60384
}
},

..............................
### Failure Responses

<table>
<thead>
<tr>
<th>Code</th>
<th>Description</th>
<th>Content Type</th>
<th>Content Example</th>
</tr>
</thead>
<tbody>
<tr>
<td>401</td>
<td>UNAUTHORIZED Error when the authentication fails possibly due to incorrect login</td>
<td>text/xml</td>
<td><code>&lt;string&gt;Error when the authentication fails possibly due to incorrect login&lt;/string&gt;</code></td>
</tr>
<tr>
<td>400</td>
<td>Model is not valid. This may be due to specifying the parameters or content incorrectly.</td>
<td>application/xml</td>
<td><code>&lt;string&gt;Model is not valid. This error has been assigned an Incident ID of '66480-API2-TEST03-170605-193846'. Provide this Incident ID to customer support when you need assistance.&lt;/string&gt;</code></td>
</tr>
</tbody>
</table>