

Your Seasonal Forecast API guidelines

Summary:

This document explains how to access Your Seasonal Forecast data in Senaps once you have been given access. We use the Boorowa location available as part of the trial product as an example. This guide shows how to use the Sensor Data API in Senaps to extract results.

Prerequisites:

- You must have Senaps credentials which have access to the Your Seasonal Forecast trial sites data within Senaps.
- An understanding of how Your Seasonal Forecast data is structured. Please see the [Your Seasonal Forecast User Guide](#) document for details.



Please refer to the shop Terms and Conditions for specific terms relating to accessing Your Seasonal Forecast data.

Introduction

The your Seasonal Forecast prediction outputs are supplied in a few arrays of data, and the confidence information is available as JSON metadata. Senaps provides API's for accessing these arrays of data (referred to as vectors in Senaps) and confidence metadata. These instructions describe how to access those vectors and metadata. A prior understanding of the contents of those vectors is assumed. Please see the [Your Seasonal Forecast User Guide](#) document for details on the product and how to interpret the contents of the data vectors.

Senaps Object Names:

Location: https://senaps.io/dashboard/#/app/location/detail/cscf-r.csiro-trial_user.boorowa

Datastream: https://senaps.io/dashboard/#/app/stream/detail/cscf-r.csiro-trial_user.boorowa.categorised_forecast

Datastream: https://senaps.io/dashboard/#/app/stream/detail/cscf-r.csiro-trial_user.boorowa.monthly_rain_to_date

Equivalent locations and streams exist for Toowoomba and Narrogin (https://senaps.io/dashboard/#/app/location/all?id=cscf-r.csiro-trial_user, and https://senaps.io/dashboard/#/app/stream/all?id=cscf-r.csiro-trial_user)

The confidence information is in the metadata of the location object, the rainfall data and predictions are accessed through the Datastreams

Accessing Categorical Forecast Data - Example

As an example, we can use the interactive API Doc's to accessing data for the Boorowa trial site.

The API operation is documented here: <https://senaps.io/api-docs/#/>. We will use the default drop down selection of "Sensor Data API" to access the forecast data; enter your credentials (username + password or apikey) and select '**explore**', and then click '**operations**' to see the list of API endpoints.

To see details of the Boorowa Forecast stream scroll down and expand the GET /streams/{id} section by clicking on that 'GET' button. You can now enter the parameters to get the details of the data stream.

Paste "cscf-r.csiro-trial_user.boorowa.categorised_forecast" without the quotes into the ID field and hit the "Try it out" button.

GET

/streams/{id}

Get details about a stream.

Response Class (Status 200)

Model

Model Schema

```
{
  "id": "string",
  "_links": {
    "self": {
      "href": "string"
    },
    "observations": {
      "href": "string"
    }
  }
},
"
```

Response Content Type

application/json

Parameters

Parameter	Value	Description	Parameter Type	Data Type
id	r.csiro-trial_user.boorowa.categorised_forecast		path	string
recursive		Return full details of embedded resources	query	boolean

Response Messages

HTTP Status Code	Reason	Response Model	Headers
401	The stream was not found, or you don't have authorisation to read this stream	<div><div>Model</div><div>Model Schema</div></div> <div><pre>{ "_embedded": { "user": { "id": "string", "hidden": false, "_links": { "self": { "href": "string" } } }, "_embedded": { "roles": {</pre></div>	

 |

Try it out!

If your authorisation is set up correctly you should see a response similar to this:

Try it out!

[Hide Response](#)

Curl

```
curl -X GET --header "Accept: application/json" --header "Authorization: Basic am9obi5tY2N1bGxvY2hAY3Npcm8uYXU6V2ViUGEkJHcwcmQh" "
```

Request URL

```
https://senaps.io/api/sensor/v2/streams/cscf-r.csiro-trial_user.boorowa.categorised_forecast
```

Response Body

```
{
  "_links": {
    "self": {
      "href": "https://senaps.io/api/sensor/v2/streams/cscf-r.csiro-trial_user.boorowa.categorised_forecast"
    },
    "observations": [
      {
        "href": "https://senaps.io/api/sensor/v2/observations?streamid=cscf-r.csiro-trial_user.boorowa.categorised_forecast"
      }
    ]
  },
  "id": "cscf-r.csiro-trial_user.boorowa.categorised_forecast",
  "reportingPeriod": "P1M",
  "resultsSummary": {
    "count": 3,
    "first": {
      "t": "2020-05-15T00:00:00.000Z",
      "v": {
        "v": [
          1
        ]
      }
    }
  }
}
```

Response Code

```
200
```

Response Headers

```
{
  "accept-ranges": "bytes",
  "access-control-allow-origin": "*",
  "allow": "OPTIONS, GET, POST, PUT",
  "connection": "keep-alive",
  "content-length": "3056",
  "content-security-policy-report-only": "default-src 'self' data: 'unsafe-inline' 'unsafe-eval' https://jira.csiro.au https://www",
  "content-type": "application/json; charset=UTF-8",
  "date": "Tue, 19 May 2020 07:26:07 GMT",
  "expect-ct": "max-age=0, enforce",
  "feature-policy": "accelerometer 'none'; ambient-light-sensor 'none'; autoplay 'none'; camera 'none'; encrypted-media 'none'; fu",
  "referrer-policy": "strict-origin-when-cross-origin",
  "vary": "Accept-Charset, Accept-Encoding, Accept-Language, Accept",
  "via": "kong/0.15.0",
  "www-authenticate": "Key realm=\"kong\"",
  "x-content-type-options": "nosniff",
  "x-frame-options": "SAMEORIGIN",
  "x-kong-proxy-latency": "0",
  "x-kong-upstream-latency": "29"
}
```

The full Response body is similar to:

```
{
  "_links": {
    "self": {
      "href": "https://senaps.io/api/sensor/v2/streams/cscf-r.csiro-trial_user.boorowa.categorised_forecast"
    },
    "observations": [
      {
        "href": "https://senaps.io/api/sensor/v2/observations?streamid=cscf-r.csiro-trial_user.boorowa.categorised_forecast"
      }
    ]
  }
}
```

```

    }
  ]
},
"id": "cscf-r.csiro-trial_user.boorowa.categorised_forecast",
"reportingPeriod": "P1M",
"resultsSummary": {
  "count": 3,
  "first": {
    "t": "2020-05-15T00:00:00.000Z",
    "v": {
      "v": [
        1,
        4.4,
        29.8,
        50.5,
        99.9,
        174.7,
        0,
        0.303,
        0.697,
        3,
        8.696,
        37.3664,
        71.8932,
        109.106
      ]
    }
  },
  "last": {
    "t": "2020-07-15T00:00:00.000Z",
    "v": {
      "v": [
        32.3,
        69.5,
        113.9,
        163.6,
        258.6,
        303.3,
        0,
        0.0606,
        0.9394,
        3,
        120.196,
        186.2804,
        228.58,
        296.72
      ]
    }
  }
},
"resulttype": "vectorvalue",
"samplePeriod": "P3M",
"usermetadata": {
  "version": [
    1
  ],
  "mappings": [
    {
      "i": 0,
      "id": "obs.p0",
      "label": "Observed Rainfall at 0th percentile",
      "group": "observed",
      "percentile": 0,
      "categories": null
    },
    {
      "i": 1,
      "id": "obs.p05",
      "label": "Observed Rainfall at 5th percentile",
      "group": "observed",
      "percentile": 0.05,

```

```

    "categories": null
  },
  {
    "i": 2,
    "id": "obs.p33",
    "label": "Observed Rainfall at 33rd percentile",
    "group": "observed",
    "percentile": 0.33,
    "categories": null
  },
  {
    "i": 3,
    "id": "obs.p66",
    "label": "Observed Rainfall at 66th percentile",
    "group": "observed",
    "percentile": 0.66,
    "categories": null
  },
  {
    "i": 4,
    "id": "obs.p95",
    "label": "Observed Rainfall at 95th percentile",
    "group": "observed",
    "percentile": 0.95,
    "categories": null
  },
  {
    "i": 5,
    "id": "obs.p100",
    "label": "Observed Rainfall at 100th percentile",
    "group": "observed",
    "percentile": 1,
    "categories": null
  },
  {
    "i": 6,
    "id": "frcst.p.t1",
    "label": "Proportion of ensembles in tercile 1",
    "group": "proportion_ensemble",
    "categories": null
  },
  {
    "i": 7,
    "id": "frcst.p.t2",
    "label": "Proportion of ensembles in tercile 2",
    "group": "proportion_ensemble",
    "categories": null
  },
  {
    "i": 8,
    "id": "frcst.p.t3",
    "label": "Proportion of ensembles in tercile 3",
    "group": "proportion_ensemble",
    "categories": null
  },
  {
    "i": 9,
    "id": "frcst_cat",
    "label": "Forecast Category",
    "group": "forecast_category",
    "categories": [
      {
        "value": 0,
        "label": "Inconclusive"
      },
      {
        "value": 1,
        "label": "Tercile 1"
      },
      {
        "value": 2,

```

```

        "label": "Tercile 2"
      },
      {
        "value": 3,
        "label": "Tercile 3"
      }
    ]
  },
  {
    "i": 10,
    "id": "frcst.p05",
    "label": "Forecast Rainfall at 5th percentile",
    "group": "forecast",
    "percentile": 0.05,
    "categories": null
  },
  {
    "i": 11,
    "id": "frcst.p33",
    "label": "Forecast Rainfall at 33rd percentile",
    "group": "forecast",
    "percentile": 0.33,
    "categories": null
  },
  {
    "i": 12,
    "id": "frcst.p66",
    "label": "Forecast Rainfall at 66th percentile",
    "group": "forecast",
    "percentile": 0.66,
    "categories": null
  },
  {
    "i": 13,
    "id": "frcst.p95",
    "label": "Forecast Rainfall at 95th percentile",
    "group": "forecast",
    "percentile": 0.95,
    "categories": null
  }
]
},
"_embedded": {
  "metadata": [
    {
      "length": 14,
      "type": ".VectorStreamMetaData"
    }
  ],
  "organisation": [
    {
      "_links": {
        "self": {
          "href": "https://senaps.io/api/sensor/v2/organisations/csiro"
        }
      },
      "id": "csiro"
    }
  ],
  "groups": [
    {
      "_links": {
        "self": {
          "href": "https://senaps.io/api/sensor/v2/groups/cscf-r.csiro-trial_user"
        }
      },
      "id": "cscf-r.csiro-trial_user"
    }
  ],
  "location": [
    {

```

```

    "_links": {
      "self": {
        "href": "https://senaps.io/api/sensor/v2/locations/cscf-r.csiro-trial_user.boorowa"
      }
    },
    "id": "cscf-r.csiro-trial_user.boorowa"
  }
]
}

```

This response contains the metadata described in the User Guide as well as the first and most recent observations. The observations are the arrays of numbers - see the [Your Seasonal Forecast User Guide](#) to interpret the individual values.

Note with this forecast product only the most recent forecast is stored and it will contain 3 sets of values (predictions for 1, 2 and 3 months out). Note that the GET /Streams/{id} call above only returns the first and last predictions. To get all three forecasts you need to use the GET /observations API call. As before enter "cscf-r.csiro-trial_user.boorowa.categorised_forecast" without the quotes into the streamid and click the Try it out! button.

GET

/observations

Get a collection of observations.

Implementation Notes

Get a collection of observations

Parameters

Parameter	Value	Description	Parameter Type	Data Type
streamid	<div>r.csiro-trial_user.boorowa.categorised_forecast</div>	Stream identifier or a comma separated list of stream identifiers	query	string
start	<div></div>	Start date (url encoded iso8601 format)	query	string
end	<div></div>	End date (url encoded iso8601 format)	query	string
time	<div></div>	Timestamp of a specific result. (url encoded iso8601 format)	query	string
si	<div></div>	Is the start parameter treated as an inclusive boundary	query	boolean
ei	<div></div>	Is the end parameter treated as an inclusive boundary	query	boolean
bounds	<div></div>	Boundary filter for a geolocation stream, or any stream located by a geolocation stream. The boundary is provided as a POLYGON in WTK format.	query	string
media	<div></div>	Format of response. Valid formats are csv, json, geojson (for geolocationvalue streams)	query	string
limit	<div></div>	Limit the number of results. The limit is 1000 by default.	query	double
sort	<div></div>	Sort the results. By default results are returned in ascending order.	query	string

Response Messages

HTTP Status Code	Reason	Response Model	Headers
200	Results returned		
401	The client is not authorised to view the results of the stream. The client must have a role of suitable scope with the readresultspermission.		

Try it out!

Hide Response

The result will be similar to below where you can see it contains values for future months, in this case the forecast was produced on May 15 so results (predictions) exist for the periods mid May to mid June (timestamped 15 May), mid June to mid July (timestamped 15 June) and mid July to mid August (timestamped 15 July).

```
{
  "_links": {
    "self": {
      "href": "https://senaps.io/api/sensor/v2/observations"
    }
  },
  "_embedded": {
```



```
"stream": {
  "_links": {
    "self": {
      "href": "https://senaps.io/api/sensor/v2/streams?id=cscf-r.csiro-trial_user.boorowa.
categorised_forecast",
      "id": "cscf-r.csiro-trial_user.boorowa.categorised_forecast"
    }
  }
},
"results": [
  {
    "t": "2020-05-15T00:00:00.000Z",
    "v": {
      "v": [
        1,
        4.4,
        29.8,
        50.5,
        99.9,
        174.7,
        0,
        0.303,
        0.697,
        3,
        8.696,
        37.3664,
        71.8932,
        109.106
      ]
    }
  },
  {
    "t": "2020-06-15T00:00:00.000Z",
    "v": {
      "v": [
        21,
        45.1,
        63.2,
        103,
        183.3,
        217.9,
        0.0303,
        0,
        0.9697,
        3,
        65.648,
        110.8092,
        145.2588,
        179.714
      ]
    }
  },
  {
    "t": "2020-07-15T00:00:00.000Z",
    "v": {
      "v": [
        32.3,
        69.5,
        113.9,
        163.6,
        258.6,
        303.3,
        0,
        0.0606,
        0.9394,
        3,
        120.196,
        186.2804,
        228.58,
        296.72
      ]
    }
  }
]
```

```

    ]
  }
},
"count": 3
}

```

Accessing Categorical Forecast Data

Accessing the rain to date data uses the same process as above, but with the stream name set to 'cscf-r.csiro-trial_user.boorowa.monthly_rain_to_date'.

Accessing Confidence Data - Example

The confidence level/skill information is attached as metadata to the location object in Senaps. It can be obtained by making a GET /locations/{id} API Call. The location name for the Boorowa site is 'cscf-r.csiro-trial_user.boorowa'

GET
/locations/{id}
Get details about a location.

Parameters

Parameter	Value	Description	Parameter Type	Data Type
id	cscf-r.csiro-trial_user.boorowa	Location id	path	string
recursive	<input type="checkbox"/>	Return full details of embedded resources	query	boolean

Response Messages

HTTP Status Code	Reason	Response Model	Headers
200	Successful response		

Try it out!

after clicking 'Try it out!' the body of the return value contains the metadata as indicated below.

```

{
  "_links": {
    "self": {
      "href": "https://senaps.io/api/sensor/v2/locations/cscf-r.csiro-trial_user.boorowa"
    }
  },
  "description": "cscf-r.csiro-trial_user_station",
  "geojson": {
    "type": "Point",
    "coordinates": [
      148.716,
      -34.433
    ]
  },
  "id": "cscf-r.csiro-trial_user.boorowa",
  "usermetadata": {
    "skill_data": {
      "score_by_month": {
        "lead_time": {
          "month1": [
            2,
            2,
            2,
            2,
            2,
            1,
            2,
            3,
            2,
            2,
            1,
            1
          ]
        }
      }
    }
  }
}

```

```

],
"month2": [
  2,
  2,
  2,
  2,
  1,
  2,
  2,
  2,
  1,
  2,
  1,
  2
],
"month3": [
  1,
  2,
  2,
  2,
  1,
  1,
  2,
  2,
  2,
  3,
  1,
  1
]
}
},
"score_by_month_legend": [
  {
    "value": 1,
    "label": "Low"
  },
  {
    "value": 2,
    "label": "Medium"
  },
  {
    "value": 3,
    "label": "High"
  }
],
"correct_by_month": {
  "lead_time": {
    "month1": [
      52.1739,
      43.4783,
      13.0435,
      26.087,
      34.7826,
      26.087,
      26.087,
      39.1304,
      34.7826,
      56.5217,
      26.087,
      30.4348
    ],
    "month2": [
      43.4783,
      21.7391,
      21.7391,
      21.7391,
      30.4348,
      34.7826,
      30.4348,
      39.1304,
      30.4348,

```

```

        34.7826,
        17.3913,
        34.7826
    ],
    "month3": [
        17.3913,
        43.4783,
        26.087,
        21.7391,
        39.1304,
        21.7391,
        39.1304,
        30.4348,
        21.7391,
        34.7826,
        26.087,
        8.6957
    ]
}
},
"correct_by_month_legend": [
    [
        "value: percentage"
    ]
],
"incorrect_by_month": {
    "lead_time": {
        "month1": [
            13.0435,
            4.3478,
            4.3478,
            8.6957,
            17.3913,
            17.3913,
            4.3478,
            0,
            13.0435,
            4.3478,
            17.3913,
            21.7391
        ],
        "month2": [
            13.0435,
            8.6957,
            8.6957,
            13.0435,
            17.3913,
            8.6957,
            4.3478,
            4.3478,
            13.0435,
            13.0435,
            13.0435,
            8.6957
        ],
        "month3": [
            13.0435,
            4.3478,
            13.0435,
            13.0435,
            17.3913,
            17.3913,
            13.0435,
            8.6957,
            13.0435,
            0,
            21.7391,
            13.0435
        ]
    }
}
},

```

```

    "incorrect_by_month_legend": [
      [
        "value: percentage"
      ]
    ],
    "skill_version": [
      "Skill_Score_v03"
    ],
    "skill_notes1": [
      "Skill scores defined using proportion of hindcasts incorrect and correct minus inconclusive hindcasts.
      If incorrect > 1 in 4.5 years; skill score = low. If incorrect < 1 in 4.5 years and > 1 in 20 years; skill
      score = med. If incorrect < 1 in 20 years; skill score = high. If incorrect score > 1 in 4.5 years & correct >
      1 in 1.5 years, skill score = med."
    ],
    "skill_datasource": {
      "hindcasts": [
        "Hindcasts: ACCESS-S1 v03. Calibrated to 5km x 5km using quantile-quantile matching approach."
      ],
      "obs": [
        "SILO grid 5km x 5km"
      ],
      "skill_table": [
        "0001_boorowa_skill_summary.csv"
      ]
    },
    "skill_updated_on": [
      "2020-05-15 17:26:31"
    ]
  },
  "_embedded": {
    "streams": [
      {
        "_links": {
          "self": {
            "href": "https://senaps.io/api/sensor/v2/streams/cscf-r.csiro-trial_user.boorowa.
categorised_forecast"
          }
        },
        "id": "cscf-r.csiro-trial_user.boorowa.categorised_forecast"
      },
      {
        "_links": {
          "self": {
            "href": "https://senaps.io/api/sensor/v2/streams/cscf-r.csiro-trial_user.boorowa.
monthly_rain_to_date"
          }
        },
        "id": "cscf-r.csiro-trial_user.boorowa.monthly_rain_to_date"
      }
    ],
    "organisation": [
      {
        "_links": {
          "self": {
            "href": "https://senaps.io/api/sensor/v2/organisations/csiro"
          }
        },
        "id": "csiro"
      }
    ],
    "groups": [
      {
        "_links": {
          "self": {
            "href": "https://senaps.io/api/sensor/v2/groups/cscf-r.csiro-trial_user"
          }
        },
        "id": "cscf-r.csiro-trial_user"
      }
    ]
  }
}

```

```
}  
}
```

Directly using the API endpoint:

(Most of this info can be deduced from the API Docs examples above.)

- a. The URL for the request is <https://senaps.io/api/sensor/v2/>
- b. The request method is *GET*
- c. Every request needs authenticating. Which can either be done either with a URL parameter API key, or a HTTP header API key or a HTTP Basic auth username and password. eg '&apikey=.....'
- d. Example parameters can be seen in the api-docs and the above examples.

The equivalent calls to above are:

```
https://senaps.io/api/sensor/v2/observations?streamid=cscf-r.csiro-trial_user.boorowa.categorised_forecast
```

```
https://senaps.io/api/sensor/v2/locations/cscf-r.csiro-trial_user.boorowa
```

The 'Try it out!' functionality in the API Docs also provides examples of making those same API calls using the common command line tool curl.