

北農研気象関連 API について

API① 先行降雨指数 API

GetAntPrecIndexResult

URL: <https://api.wagri2.net/naro-harc/others/met1/v1/GetAntPrecIndexResult>

API② 雪踏み API

GetYukifumiReuslt

URL: <https://api.wagri2.net/naro-harc/others/met1/v1/GetYukifumiReuslt>

API① 先行降雨指数 API

1. リクエスト仕様

URL： <https://api.wagri2.net/naro-harc/others/met1/v1/GetAntPrecIndexResult>

メソッド： POST

ヘッダ： 以下の表の通り

要素	指定値
X-Authorization	WAGRI アクセストークン

ボディ： JSON 形式で与える。

パラメータ名	値の説明	型	補足
lat	緯度	number	必須 (JGD2000)
lon	経度	number	必須 (JGD2000)
start_date	データ出力開始日	ISO8601(日付のみ)	必須
end_date	データ出力最終日	ISO8601(日付のみ)	必須
output_type	出力内容の選択	string	デフォルトは 0

2. 正常レスポンス

status: HTTP ステータスコード 200

result: output_type の指定（無指定の場合のデフォルト値=0）に応じて、下表の内容のデータを返す。

output_type の値	レスポンスデータの中身	レスポンスデータの型
0	指定期間の先行降雨指数を出力	float (少数 1 位)
1	指定期間の先行降雨指数と日付を出力	float (少数 1 桁)、ISO8601 (日付)
2	指定期間の先行降雨指数と日付、計算に使用した入力気象データ（気温、降水量、日射量）を出力	float (少数 1 桁)、ISO8601 (日付)
3	指定期間の先行降雨指数と中間変数（蒸発量）、日付を出力	float (少数 1 桁)、ISO8601 (日付)

※ 先行降雨指数を計算する python プログラムは、発注者側で用意する。計算は、データ出力開始日の 3 ヶ月前から、データ出力最終日まで実行する。

3. エラーレスポンス

status: HTTP ステータスコードを返す。(下表：200（正常）除く）

ステータスコード	名称	値の説明
200	OK	正常動作
400	Bad Request	リクエストエラー（変数指定、指定変数のエラー）
401	Unauthorized	認証エラー
403	Forbidden	許可されていない
404	Not Found	URL にリソースが存在しない
408	Request Timeout	リクエストが時間内に完了しなかった
500	Internal Server Error	サーバー処理エラー

4. API の実行例

実際に API を実行して得られる出力 JSON について、入力 JSON と共に以下に示す。

- 入力

```
{
  "lat":42.8458,
  "lon":143.02539,
  "start_date":"2024-03-16",
  "end_date":"2024-03-17",
  "output_type":"2"
}
```
 - 出力

```
{
  'antprec_index': [3.1, 8.4],
  'date': ['2024-03-16T00:00:00', '2024-03-17T00:00:00'],
  'tmpa': [-1.4, -2.1],
  'prec': [0.0, 6.5],
  'srad': [17.6, 9.5]
}
```
-

API② 雪踏み API

1. リクエスト仕様

URL： <https://api.wagri2.net/naro-harc/others/met1/v1/GetYukifumiReuslt>

メソッド： **POST**

ヘッダ： 以下の表の通り

要素	指定値
X-Authorization	Access Token

ボディ（パラメータ）： 以下の表の通り（JSON 形式で与える）

パラメータ名	値の説明	型	補足
lat	緯度	number	必須（JGD2000）
lon	経度	number	必須（JGD2000）
start_date	データ出力開始日	ISO8601(日付のみ)	必須
end_date	データ出力最終日	ISO8601(日付のみ)	必須
work_date	雪踏み日	ISO8601(日付のみ)	複数設定可
output_type	出力内容の選択	string	デフォルトは 4

2. 正常レスポンス

status: HTTP ステータスコード 200

result: output_type の指定（無指定の場合のデフォルト値=0）に応じて、下表の内容のデータを返す。

output_type の値	レスポンスデータの中身	レスポンスデータの型
0	日付（指定期間）のみ出力	ISO8601（日付）
1	日付（指定期間）と 2cm 深地温を出力	ISO8601（日付）、float（少数 1 桁）
2	日付（指定期間）と 15cm 深地温を出力	ISO8601（日付）、float（少数 1 桁）
3	日付(指定期間)と 2cm 深地温、15cm 深地温を出力	ISO8601（日付）、float（少数 1 桁）
4	日付（指定期間）と地温以外の要素（積雪深、土壌凍結深、気温）を出力	ISO8601（日付）、float（少数 1 桁）
5	日付（指定期間）と 2cm 深地温と、地温以外の要素を出力	ISO8601（日付）、float（少数 1 桁）

6	日付（指定期間）と 15cm 深地温と、 地温以外の要素を出力	ISO8601（日付）、float（少 数 1 桁）
7	日付（指定期間）と 2cm 深地温、 1 5 cm 地温、地温以外の要素を出力	ISO8601（日付）、float（少 数 1 桁）

※ 雪踏みを考慮した地温を計算する python プログラムは、発注者側で用意する。計算は、計算対象とする冬の 10 月 1 日から、データ出力最終日まで実施する。

3. エラーレスポンス

status: HTTP ステータスコードを返す。（下表：200（正常）除く）

ステータスコード	名称	値の説明
200	OK	正常動作
400	Bad Request	リクエストエラー（変数指定、 指定変数のエラー）
401	Unauthorized	認証エラー
403	Forbidden	許可されていない
404	Not Found	URL にリソースが存在しない
408	Request Timeout	リクエストが時間内に完了し なかった
500	Internal Server Error	サーバー処理エラー

4. API の実行例

実際に API を実行して得られる出力 JSON について、入力 JSON と共に以下に示す。

- 入力


```
{
  "lat":42.8458,
  "lon":143.02539,
  "start_date":"2022-01-01",
  "end_date":"2022-01-05",
  "work_date":["2022-01-03"],
  "output_type":"7"
}
```

- 出力

```
{  
  'date': ['2022-01-01T00:00:00', '2022-01-02T00:00:00', '2022-01-  
03T00:00:00', '2022-01-04T00:00:00', '2022-01-05T00:00:00'],  
  'tmps_2cm': [4.1, 4.3, 2.7, 1.4, 1.0],  
  'tmps_15cm': [9.1, 8.3, 7.6, 6.9, 6.2],  
  'snow_depth': [7.0, 15.0, 2.9, 2.9, 2.9],  
  'frozen_depth': [-0.1, 0.0, -0.1, -1.0, -1.0],  
  'tmpa': [-11.9, -9.2, -7.1, -7.0, -4.5]  
}
```