syngenta

INSIGHTS ENGINE API SERVICE DOCUMENTATION

Contents

1 Introduction

The Insights Engine (IE) is a unique model serving and deployment platform that offers a selection of scientifically verified agronomic models, encapsulated as APIs, and ready for testing and integration in product features.

Models are classified in below categories:

1) **Crop Monitoring Models**
   Those models aim at characterizing field environment or simulating crop physiology mechanisms by taking into account crop interactions with the environment. They return timely fundamental crop status information and predictions. As a general rule, they answer the question “What is the status of my field or crop at this point in time?”

   Within Crop Monitoring are the Crop Phenology models which aim at forecasting crop growth stages. Crop stage timing is a core insight to monitor crop status and schedule farm operations and crop stage predictions are often used as a module for more advanced agronomy models.

2) **Pest Monitoring Models**
   Pest monitoring models are designed to assess and track pest reproduction cycles and population dynamics at regional or field level. These models aim to provide valuable information for pest management strategies, helping agronomists and farmers to optimize pest mitigation strategies. They respond to “How vulnerable is my crop to pest risks?”.

3) **Crop Management Models**
   Crop Management Models intend to optimize farm operations by providing advanced prescriptions for a wide range of tactical decisions and precision agriculture techniques. Thanks to remote sensing imagery inputs, some models support in-field variability assessment and management. They support enquiries such as “What are the best parameters (i.e. timing, product/seed selection, application/seed parameters.. etc) to execute this task?” or “Where in my field should I do this ?”

2 Access permission

This section is created provide guideline about how to access insights engine models using crowise base authentication

Perquisite

You must have Cropwise Base account. In case you don't please use below steps to create one:

Select the environment URL from below table in which you want to create the account:

Environment	Endpoint
DEV	https://dev.accounts.cropwise.com
STAGE	https://staging.accounts.cropwise.com
PROD	https://accounts.cropwise.com

Follow the steps as detailed here: How-to create a Cropwise Base service account for your new Protector service

Cropwise Base Auth Token Boilerplate

Below we have provided a boilerplate code to get a Cropwise Base Access Token using your Cropwise Base credentials

2.1.1 Python Code

```
import requests
from base64 import b64encode

IE_STAGING_CLIENT_ID = '<client_id>'
USERNAME = '<username>'
PASSWORD = '<password>'
URL = 'https://api.staging.base.cropwise.com/oauth/token'


# Authorization token: we need to base 64 encode it
# and then decode it to acsii as python 3 stores it as a byte string
def basic_auth(username, password):
    token = b64encode(f"{username}:{password}".encode('utf-8')).decode("ascii")
    return f'Basic {token}'


headers = {
    'Authorization': basic_auth(IE_STAGING_CLIENT_ID, ''),
    'Content-Type': 'application/x-www-form-urlencoded'
```

```
}

payload = {
    'grant_type': 'password',
    'username': USERNAME,
    'password': PASSWORD
}

r = requests.post(URL, data=payload, headers=headers)
print("Access Token:", r.json()['access_token'])
```

2.1.2 Curl

```
curl --location 'https://api.staging.base.cropwise.com/oauth/token' \
--header 'Authorization: Basic <Encoded Client ID>' \
--header 'Content-Type: application/x-www-form-urlencoded' \
--data-urlencode 'grant_type=password' \
--data-urlencode 'username=<username>' \
--data-urlencode 'password=<password>'
```

2.1.3 Insights Engine RestAPI Boilerplate

- List of Insights Engine environments

Environment	Endpoint
DEV	https://dev.api.insights.cropwise.com
STAGE	https://qa.api.insights.cropwise.com
PROD	https://api.insights.cropwise.com

Below we have provided boilerplate code to connect with Insights Engine predictions endpoint

2.1.4 Python Code

```
import requests
import json

IE_URL = '<insights engine env url>'

url = f"{IE_URL}/v2.0/predictions"

payload = json.dumps({
  "request_version": "v1.0",
  "fields": [
    {
      "id": "cwp:base::field:id:<field_id>",
      "models": [
        {
          "name": "<model name>",
          "version": "v1.0"
        }
      ]
```

```
    }
  ]
})
headers = {
  'accept': 'application/json',
  'Content-Type': 'application/json',
  'Authorization': 'Bearer <Cropwise Base Token>'
}

response = requests.request("POST", url, headers=headers, data=payload)

print(response.text)
```

2.1.5 Curl

```
curl --location '<url>/v2.0/predictions' \
--header 'accept: application/json' \
--header 'Content-Type: application/json' \
--header 'Authorization: Bearer <Cropwise Base Token>' \
--data '{
    "request_version": "v1.0",
    "fields": [
        {
            "id": "cwp:base::field:id:<field_id>",
            "models": [
                {
                    "name": "<model name>",
                    "version": "v1.0"
                }
            ]
        }
    ]
}'
```

3 Computational Agronomy models

3.1 API documentation

Swagger link：https://docs.api.insights.cropwise.com/#tag/Computational-Agronomy
Credential required: username: **insights-engine** / password: **e.7@NU)gW-xt{W**

3.2 Model catalogue

Model/API Name	Version	Status	PROD release date	Description	Category	Type
Crop Phenology	2.0	PROD	Aug-23	Get current year crop stage timings and forecast	Crop monitoring	Mechanistic
Relative Yield	1.0	PROD	Feb-24	Relative yield forecast for multiple crops	Crop monitoring	Mechanistic
Avizio	1.0	PROD	Mar-23	Get diverse disease risk forecast features for Cereals	Pest monitoring	Mechanistic
BYDV	2.0	PROD	Sep-23	Forecasts the occurrence of the secondary and tertiary spreads of cereal aphids based on degree days.	Pest monitoring	Rule-based
Corn Grain Dry-Down	1.0	PROD	Oct-23	Predicts grain moisture after physiological maturity	Crop monitoring	Mechanistic

3.3 Crop Management Models

3.3.1 DSSAT Pheno

The DSSAT Pheno models predict growth stages for various crops and regions, using field-centric data such as field location and crop management practices, coupled with long-term local weather data, and soil profile characteristics. The goal is to provide growers with a timeline of growth stages before the growing season (using historical weather data) and during the season (using the current year's weather data). The predictions do not attempt to fully replace scouting for growth stages at fields.

It provides growth stage prediction for Maize, Soybean, Wheat, Barley and Sunflower in multiple countries. Model predictions are based on gridded weather data from CE-hub, soils profile characteristics from ISRIC SoilGrids, and cultivar calibrations performed across a wide range of environment condition using Syngenta trials.

API Specification

Endpoint: https://api.insights.cropwise.com/dssat/v2.0/api/predictions

Configuration

Data	Description
API Endpoint	• DEV: https://dev.api.insights.cropwise.com/v2.0/predictions • QA: https://qa.api.insights.cropwise.com/v2.0/predictions • PROD: https://api.insights.cropwise.com/v2.0/predictions
Content-Type	`application/JSON`
Authorization	Bearer Token Cropwise
Method	`POST`

Request Schema

Parameter	Type	Required	Default	Constraints / Possible Value	Description
request_version	String	Yes	`v1.0`	`v1.0`	version of standardize response structure/template
fields	Array	Yes			An array of fields to be simulated

Parameter	Type	Required	Default
id	String	No	
models	Array	Yes	
models.name	String	Yes	`DSSAT`
models.version	String	Yes	`v2.0`
query	Object	No	
query.transformers	Array of strings	No	`["cwp:ins:transformer::post:growth_stage_prediction:v1"]`
location	GeoJSON Object	Yes	
location.type	String	Yes	

location.geometry	Object	Yes	
location.geometry.coordinates	Array<double>	Yes	
location.properties.distance_to_grid.value	int	No	50
location.properties.distance_to_grid.unit		No	km
crop	String	Yes	
crop_variety	Object	Yes	
crop_variety.name	String	No	
crop_variety.relative_maturity	String	No	
planting	Object	Yes	
planting.date	String	Yes	
planting.depth.value	Int	No	
planting.depth.unit	String	No	cm
planting.density.value	Int	No	
planting.density.unit	String	No	"plants/m2"
planting.row_spacing.value	Int	No	
planting.row_spacing.unit	String	No	"cm"
planting.field_water_capacity.value	Int	No	100
planting.field_water_capacity.unit	String	No	"percentage"
water_supply	Object	No	
water_supply.is_irrigated	Boolean	No	false
time_period	Object	No	
time_period.forecast	Object	No	
time_period.forecast.start_date	String with a datetime format in UTC	No	"planting date (i.e. start of crop cycle)"
time_period.forecast.end_date	String with a datetime format in UTC	No	"date of last growth stage (i.e. end date of crop cycle)"

time_period.historical	String with a datetime format in UTC	No	
time_period.historical.start_date	String with a datetime format in UTC	No	`"current date - 20 years"`
time_period.historical.end_date	String with a datetime format in UTC	No	

Response Schema

Response parameter	Type	Required	Description
response_version	string	Yes	
results	array	Yes	
results.id	string	Yes	id from request
results.location	object	Yes	GeoJSON Geometry.
results.location.type	string	Yes	A GeoJSON object with the type "Feature" is a feature object.(Feature)
results.location.geometry	object	Yes	The geographic objects of API use the GeoJSON (RFC 7946) format.
results.location.geometry.type	string	Yes	A GeoJSON geometry object of any type other than "GeometryCollection" must have a member with the name "coordinates". The value of the coordinates member is always an array. The structure for the elements in this array is determined by the type of geometry. (type)
results.location.geometry.coordinate	array [long,lat]	Yes	Valid Longitude and Latitude values should be specified. It will automatically define country.
results.location.properties	object	optional	
results.location.properties.grid_id	string	optional	A temporary input to associate to a field location (to help model output validation)

results.location.properties.country_code	string	optional	two-letter or three-letter code that is used to represent a specific country or geographical region.
results.location.properties.landscape	array	optional	A landscape is the visible features of an area of land
results.location.properties.landscape.woodland	object	optional	Percentage of woodland area.
results.location.properties.landscape.woodland.value	float	optional	Value should be greater than or equal to 0 but less than or equal to 100.
results.location.properties.landscape.woodland.unit	string	optional	Unit of the value above. Default "percentage"
results.location.properties.landscape.grassland	object	optional	Percentage of grassland area.
results.location.properties.landscape.grassland.value	float	optional	Value should be greater than or equal to 0 but less than or equal to 100.
results.location.properties.landscape.grassland.unit	string	optional	Unit of the value above. Default "percentage"
results.location.properties.distance_to_grid	object	optional	This is the distance radius to get the data points available for the specified user geo co-ordinates.
fields.location.properties.distance_to_grid.value	integer	optional	distance to grid value
fields.location.properties.distance_to_grid.unit	string	optional	unit
results.metadata	object	Yes	
results.metadata.type	string	Yes	Type of the model
results.metadata.models	array	Yes	Executed model
results.metadata.models.name	string	Yes	Name of executed model
results.metadata.models.version	string	Yes	Version of executed model
results.metadata.models.models	array	optional	submodels
results.metadata.models.models.name	string		name of submodel
results.metadata.models.models.version	string		version of submodel
results.metadata.packages	array	optional	Model specific
results.metadata.packages.name	string		name of the package
results.metadata.packages.version	string		version of package
results.metadata.result_time	integer	Yes	epoch time
results.metadata.attributes	object		Attributes of metadata, must be in key value format.

results.page	object		Pagination is a process whereby a huge data set is distributed into several sub-category pages.
results.page.offset	string		specify the desired starting point in the model response data set.
results.page.count	string		Number of items on the current page.
results.predictions	array	Yes	predicted results
results.predictions.feature_category	string	Yes	Related model outputs , should be grouped into feature_categories relevant to specific use cases. Example: Daily disease risk, Historical statistics, Absolute predictions, Relative predictions
results.predictions.prediction_window	object	optional	This object should return attributes related to the time period and time aggregation over which a prediction is made
results.predictions.prediction_window.type	string		model specific. can be forecast, historical_trend, etc.
results.predictions.prediction_window.frequency	string		Can be "daily", "monthly", "yearly", "season", "custom", etc. Note : One date in the UTC date format can be passed if the frequency is "daily".
results.predictions.prediction_window.start	string with date format		start date with UTC format. Full-date notation as defined by RFC 3339, section 5.6, link
results.predictions.prediction_window.end	string with date format		end date with UTC format. Full-date notation as defined by RFC 3339, section 5.6, link
results.predictions.prediction_window.date	string with date format	No	Date of the prediction for predictions for particular hour or day. end date with UTC format. Full-date notation as defined by RFC 3339, section 5.6, link
results.predictions.features	array	Yes	Feature objects gather values and attributes for the model outputs returned by the API
results.predictions.features.type	string	Yes	Type is a descriptive label of the feature object. geoJSON - only for imaginary models

results.predictions.features.value	string/ integer	Yes	Value refers to the specific data point or content of information that is associated with a particular feature.
results.predictions.features.unit	string	Yes	When necessary, the unit of the value should be specified
results.predictions.features.attributes	object	optional	Feature specific characteristics. Must be in key value format.
results.predictions.features.model	array		A model object may be required in the case of multi-model systems to specify the type attributes of the model used to generate a specific feature value.
results.predictions.features.model.name	string	Yes	name of model
results.predictions.features.model.version	string	Yes	version of model
results.predictions.features.model.type	string	Yes	type
results.predictions.features.model.algorithm	array[math]	Yes	Neural, Average, Polynomial , Linear algorithms
results.predictions.attributes	object		High level characteristics referring to the whole model predictions object. Must be in key value format.
response_version	string	Yes	
results	array	Yes	
results.id	string	Yes	id from request - SIRET_ID
results.fields	array	Yes	List of fields for the SIRET_ID
results.fields.id	string	Yes	unique id of each field **Syntax** - "cwp:application::field:mainEntity:subEntity:value"
results.fields.location	object	Yes	GeoJSON Geometry. (similar to response structure in V1.0)
results.fields.metadata	object	Yes	similar to response structure in V1.0
results.predictions	array	Yes	List of model outputs categorized by feature_category (similar to response structure in V1.0)

3.3.2 Corn grain Dry Down v1.0

This document serves as a comprehensive guide to the Dry Down Model API, which is available within the Insights-Engine platform. It provides detailed information about the request and response structures, along with an overview of the API's functionality .

Request Schema

Parameter	Type	Required	Default value (if non required)	Valid range and Constraints	Business Description
`fields.location.type`		yes		Only GeoJson Feature is supported	A GeoJSON object with the type `"Feature"` is a feature object. See: GeoJSON Feature Object
`fields.location.geometry.type`		yes		Only GeoJson Point is supported	A GeoJSON geometry object of any type other than `"GeometryCollection"` must have a member with the name `"coordinates"`. The value of the coordinates member is always an array. The structure for the elements in this array is determined by the type of geometry. See: GeoJSON Geometry Object
`fields.location.geometry.coordinates`		yes		Valid countries: USA	Geographical coordinates provided as a sequence in which first place is for longitude and second place is for latitude. The country will be automatically identified based on coordinates.
`fields.crop`		yes	CORN	Valid crops: CORN	Crop name

fields.observations.category		yes	crop_stages	For Dry Down model, only crop_stage is currently accepted.	Type of field or crop observations
fields.observations.values		yes			Values associated with observed or forecasted crop stages
fields.observations.values.scale			ritchie_scale	For Dry Down model, only Ritchie scale is currently accepted.	Crop growth stage scale used to report crop stage observation.
fields.observations.values.stage_name			R6	For Dry Down model, only R6 stage is currently accepted.	Name of the crop stage according to used crop stage.
fields.observations.values.date		yes		For Corn USA: date must be between July 1st and October 31st and need to occur less than 6 month prior to current date.	Date when start of crop stage is observed or forecasted. For Dry Down model, a forecast or observed date for R6 must be provided.

`fields.crop_variety.drying_coefficient_k`			0.0336	0.0001 to 0.9999	*k* is a proportionally drying coefficient used to generate the dry-down curve and that may be calibrated at crop variety level. Currently the drying coefficient (k) default value refers to calibration on trials referring to even years in the US. The coefficient that minimized the sum of the square of the residuals between the observed and predicted moisture content at harvest was selected. Reference values: • k for US: 0.0336 • k from literature: 0.0622 Advanced explanations about k: The drying coefficient (k) is a coefficient that influences the behavior of the dry-down curve. This reflects mainly region, crop, and hybrid characteristics. So ideally the k value should be calibrated for each situation with a good number of observations for each condition. As of now, the model was tested for corn in the USA and k value is calibrated over a large region including variability in environment and

					management decisions. In future use cases, the k coefficient could be adjusted based on ground-truth observations feedbacks from users and handled by editing the API input value. The k coefficient affects the dry down pattern is the following way: • if k increases, it will infer that the grain moisture dries faster and the optimal harvest time will reached sooner. • if k decreases, the grain drying will be slower and the time to harvest will be later
grain_moisture_at_harvest			14.5	Minimum allowed value = 14 Maximum allowed value = 35	Percentage value of grain moisture to be used to predict optimal_harvest_time

Response Schema

Parameter	Type	Required	Valid range	Example	Business Description
response_version	string	Yes		v1.0	version number of response structure/template

results.location.type	string	Yes	Feature	Feature	The geographic objects of API use the GeoJSON (RFC 7946) format. The regions and fields can be represented as GeoJSON Features with a specific property schema. example: Feature
results.location.geometry.type	string	Yes	Point	Point	Only GeoJson Point type is supported
results.location.geometry.coordinates	array [long,lat]	Yes	Longitude in (-180, 180) Latitude in (-90, 90)	[19.5058, 47.161]	In geometry, coordinates should be in sequence. First place is for longitude and second place is for latitude. Example : [-0.97024, 53.5282]
results.metadata.type	string	Yes	mathematical	mathematical	Type of model operating
results.metadata.models.name	string	Yes	dry-down	dry-down	model name
results.metadata.models.version	string	Yes	v1.0	v1.0	model version
results.metadata.packages.name	string	Yes	dry-down	dry-down	package name
results.metadata.packages.version	string	Yes	0.6.3	0.6.3	package version
results.predictions.feature_category **dry_down_curve**	string	Yes	dry_down_curve		**dry_down_curve**: the features in this category will return timestamped values of forecasted mean grain moisture. Those values represent the grain dry-down curve. Predictions are returned daily from R6 date until 6 months later.
results.predictions.features.type **dry_down_curve**	string	Yes	grain_moisture:mean		**grain_moisture:mean:** this represent the grain moisture for the corn crop after R6
results.predictions.features.value **dry_down_curve**	string	Yes	36 - 0	31.165	Percentage value of grain moisture

results.predictions.features.type **dry_down_curve**	string	Yes	From R6 date until 6 months later	date	**date:** day on which grain_moisture:mean value is predicted
results.predictions.feature_category **optimal_harvest_time**	string	Yes	optimal_harvest_time		**optimal_harvest_time**: returns the first date that "`grain_moisture:mean`" was below the requested "`grain_moisture_at_harvest`". These value represent the day when farmer should plan for harvest.
results.predictions.features.type **optimal_harvest_time**	string	Yes	"`optimal_harvest_time`".date	date	
results.predictions.features.value **optimal_harvest_time**	string	Yes	"`optimal_harvest_time`".value	`2023-07-24`	This is the day when grain_moisture:mean got equal or below requested "`grain_moisture_at_harvest`"

3.3.3 Corn Growth stage Model (GDD model)

Introduction

The rate that corn grows and develops changes during the season. Under normal growing conditions, the rate of plant development is largely dependent on temperature. Environmental factors, such as water and nutrient deficiencies, can alter the relationship between plant growth and temperature. Many management decisions consider the stage of growth and development of the crop. For example, some pesticide products are labeled for use only at certain stages because of the potential for crop damage or other undesirable effects. Fertilizers applied at the right time can provide a greater crop response; however, if fertilizer is applied at the wrong growth stage, benefits can be reduced or negative responses can occur. Water stress at certain stages is more critical than at other stages. Management efficiency can be improved by matching the crop's need to the treatment. Understanding how a corn plant grows and develops is important for maximizing efficiency.

Reqest Schema

Parameter	Type	Required	Constraints	Business Description
geometry.type	string	Yes	Equal to “Point”	
geometry.coordinates	array [long,lat]	Yes	Longitude in (-180, 180) Latitude in (-90, 90) Supported countries: United States of America	
gdd_flowering	integer	Yes	gdd_flowering in [1000-1600]	
gdd_black_layer	integer	Yes	gdd_black_layer in [1700-2800]	
planting_date	string<date>	Yes	Month should not be any of [01, 02, 08, 09, 10, 11, 12] Planting Date should be any in 25 years ago from today to 10 years in future from today	
last_observed_stage.stage	string	Optional	Supported Stages ['VE', 'V1', 'V2', 'V3', 'V4', 'V5', 'V6', 'V7', 'V8', 'V9', 'V10', 'V11', 'V12', 'V13', 'V14', 'V15', 'V16', 'V17', 'V18', 'V19', 'V20', 'V21', 'R1 Silk - Pollination', 'R2 Blister', 'R3 Milk', 'R4 Dough', 'R5 Dent', 'R6 Black Layer']	
last_observed_stage.date	string<date>	Optional	Last Observed Stage Date should be greater than Planting Date Last Observed Stage Date should lie in same year as of Planting Date	

Response Schema

Parameter	Type	Constrain	Business Description
geometry.type	string	Only GeoJson Point type	Equal to “Point”
geometry.coordinates	array [long,lat]		Supported countries: United States of America
results.metadata.type	string		Type of model operating `"mathematical"` `"algorithm"` `"rule"` `"mechanistic"`

results.metadata.model	string		Category to which model falls into.
results.metadata.version	string		model version
results.metadata.calibration.standard_deviation.value	integer		Model calibration factor depends on the type of model
results.metadata.calibration.standard_deviation.unit	string		Model calibration unit
results.metadata.crop	array	`corn`	supported crops by the model
results.metadata.result_time	integer		API response time in epoch seconds
results.predictions.type	string		Type for the predictions
results.predictions.date	string<date>		Date for the predictions
results.predictions.gdd_cumulative_sum	float		GDD cumulative sum for the predictions
results.predictions.vegetative_growth_stage	string		Vegetative Growth Stage for the predictions

Errors

Code	Internal Code	Target	Type	Message
400	INS_IN_VAL_0001	dynamic ex:- geometry.coordinates	required	dynamic ex:- required field coordinates is missing
400	INS_IN_VAL_0002	dynamic ex:- planting_date	invalid	dynamic ex:- invalid date format
500	INS_EV_SER_0001	external.api.call	server_error	Weather data not available
500	INS_EV_SER_0001	db.connectivity	server_error	Error while connecting to DocumentDB
500	INS_EV_SER_0004	type	invalid	url not found
400	INS_GDD_VAL_0001	geometry.coordinates	invalid	Invalid country ['US']: CA
400	INS_GDD_VAL_0002	geometry.coordinates	invalid	Invalid state ['North Dakota', 'South Dakota', 'Minnesota', 'Wisconsin', 'Michigan', 'Ohio', 'Indiana', 'Illinois', 'Iowa', 'Nebraska', 'Kansas', 'Missouri']: California
400	INS_GDD_VAL_0003	gdd_flowering	invalid	Invalid gdd_flowering (1000, 1600): 500
400	INS_GDD_VAL_0004	gdd_black_layer	invalid	Invalid gdd_black_layer (1700, 2800): 800

400	INS_GDD_VAL_0005	last_observed_stage.stage	invalid	Invalid last_observed_stage.stage ['VE', 'V1', 'V2', 'V3', 'V4', 'V5', 'V6', 'V7', 'V8', 'V9', 'V10', 'V11', 'V12', 'V13', 'V14', 'V15', 'V16', 'V17', 'V18', 'V19', 'V20', 'V21', 'R1 Silk - Pollination', 'R2 Blister', 'R3 Milk', 'R4 Dough', 'R5 Dent', 'R6 Black Layer']: V23
400	INS_GDD_VAL_0006	last_observed_stage.date	invalid	last_observed_stage.date should be greater than planting_date
400	INS_GDD_VAL_0007	last_observed_stage.date	invalid	last_observed_stage.date should lie in same year as of planting_date
400	INS_GDD_VAL_0008	planting_date	invalid	Invalid planting_date[min 1997-04-11]: 1970-03-02
400	INS_GDD_VAL_0009	planting_date	invalid	Invalid planting_date[max 2032-04-11]: 2035-03-02
400	INS_GDD_VAL_0010	planting_date	invalid	Invalid planting_date.month [3, 4, 5, 6, 7]: 1
400	INS_GDD_VAL_0011	request.body	invalid	Unable to calculate percentile 10 for the given input data
400	INS_GDD_VAL_0012	request.body	invalid	Unable to execute model with provided request data

3.4 Pest Monitoring Models

3.4.1 BYDV

Introduction

BYDV stands for Barley Yellow Dwarf Virus. It is a Luteovirus which is transmitted to winter cereals at early growth stages (BBCH 11-31) in autumn by feeding of cereal aphids: Sitobion avenae (MACSAV, grain aphid), Rhopalosiphum padi (RHOPPA, bird cherry oat aphid), Metopolophium dirhodum (METODR, rose grain aphid). The plant infection occurs around 30 minutes after aphid feeding. BYDV infection can cause up to 30% (winter wheat) and 80% (winter barley) yield loss.

Request Schema

Parameter	Type	Required	Constraints	Description
geometry.type		Yes		
geometry.coordinates		Yes		
start_plant_emergence_date	date	Yes	It cannot be in the future.	Date when first crops seedlings are observed out of ground. Starts the DD sum. Specific to Field (i.e. coordinates). I
last_aphid_insecticide_application_date	date	optional	Validity period: must be after first_crop_emergence_date	If an aphid pesticide was applied. The DD sum will be reset to 0 and on hold for 7 days. Specific to Field (i.e. coordinates)
planting_date	date	optional	Validity period: must be prior to first_crop_emergence_date	Placeholder to be consumed for future model improvement. Specific to Field (i.e. coordinates)
t_base	float	optional Default value: 3°C (UK reference)		DD formula parameter. Can be edited if model is calibrated for another region or crop.
early_warning_dd_threshold	integer	optional Default value: 145 DD (UK reference)		DD alert parameter. Can be edited if model is calibrated for another region or crop.
risk_warning_dd_threshold	integer	optional Default value: 170 DD (UK reference)		DD alert parameter. Can be edited if model is calibrated for another region or crop.

Response Schema

Parameter	Type	Constraints	Description
geometry.type	string		
geometry. coordinates	Array [long,lat]		
early_warning.date	date		First day when Early_warning_DD_threshold is reached
early_warning.type	string	Valid values: FORECASTED | PROJECTED	Forecasted: when DD calculated upon forecasted temperature (short term) Projected: when DD calculated upon forecasted temperature and historical local projection (long term)
risk_warning.date	date		First day when risk_warning_DD_threshold is reached
risk_warning.type	string	Valid values: FORECASTED | PROJECTED	Forecasted: when DD calculated upon forecasted temperature (short term) Projected: when DD calculated upon forecasted temperature and historical local projection (long term)

Errors

Code	Internal Code	Target	Type	Message
400	INS_IN_VAL_0001	geometry	invalid	Required field geometry is missing
400	INS_IN_VAL_0002	planting date	invalid	Invalid date format for planting date. Should be of format YYYY-MM-DD
400	INS_IN_VAL_0002	start_plant_emergence_date	invalid	Invalid emergence date. Provided date cannot be in the future
400	INS_IN_VAL_0002	start_plant_emergence_date	invalid	Invalid date format for emergence date. Should be of format YYYY-MM-DD
400	INS_IN_VAL_0002	Applications date	invalid	Invalid date format for Applications date. Should be of format YYYY-MM-DD
400	INS_IN_VAL_0002	Applications date	invalid	Invalid application date. Provided date cannot be in the future.
400	INS_IN_VAL_0002	geometry	invalid	Invalid Value for location cordinates. Should be of format [long, lat]. Where (-180 <= long <= 180) and (-90 <= lat <= 90)

400	INS_IN_VAL_0002	planting date	invalid	Planting date should be less than emergence date
400	INS_IN_VAL_0004	start_plant_emergence_date	invalid	Required start_plant_emergence_date missing
404	INS_EV_SER_0004	type	invalid	url not found
500	INS_EV_SER_0001	db.connectivity	server_error	Weather API down

3.4.2 Avizio Model

Introduction

The Avizio model forecast various infestation and damage patterns for the major cereal foliar diseases. It provide disease and organ specific insights, both before planting and during the season, according to field characteristics, cropping practices and weather. The model also returns crop stage predictions.

Quantifying potential crop damages and alerting on disease risks can help growers make profitable and sustainable crop protection decisions. However, the model does not attempt to replace field scouting and in-field disease observations.

Request Schema

Parameter	Type	Required	Constraints / Possible Value	Description
request_version	String	Yes	v1.0	A version of the request made.
fields	Array<Objects>	Yes		
fields.models	Array<Objects>	Yes		
fields.models.name	String	Yes		Name of the model to be used.
fields.models.version	String	Yes	v1.0	Version of the model to be used.
fields.models.version.business_rule_country_code	String	No	Only FR is accepted right now	It is the country code required to run the business rules inside the model.
fields.location	GeoJSON Object	Yes	A valid GeoJSON object with type and geometry	It is the location for which the Avizio model needs to return a prediction.

`fields.location.type`	String	Yes	Should be `"Feature"`	
`fields.location.geometry`	Object	Yes	Should contain type and coordinates	The geometry for a specific location.
`fields.location.geometry.coordinates`	Array<double>	Yes	max items: 2 min items: 2	First place is for longitude and second place is for latitude. Currently supported countries: • FR: France • UK: United Kingdom • DE: Germany • BE: Belgium • NL: Netherlands • LU: Luxembourg • CH: Switzerland • IT: Italy • PL: Poland • UA: Ukraine • RU: Russia • HU: Hungary • ES: Spain
`fields.location.properties.grid_id`	Integer	No		The grid id maps to a location. When provided the `geometry.coordinates` won't be considered.
`fields.crop`	String	Yes	Enum: `"WINTER_WHEAT"` `"WINTER_BARLEY"` `"SPRING_BARLEY"` `"SOFT_WHEAT"` `"HARD_WHEAT"` `"WHEAT"` `"BARLEY"`	Crop for which disease prediction is requested.

`fields.crop_variety.name`	String	Yes		Crop variety name as referenced to in official country catalogue. No validation applied currently from API Supported varieties currently managed as a dependency table specific to the model. If name is recognized in dependency catalogue, then specific variety traits will be used unless provided. If name is not provided or not recognized in dependency table, then traits that are not provided will be default values. Synonyms: cultivar, hybrid, commercial name.
`fields.crop_variety.traits`	Array<Object>	No		Inputs for variety agronomical ratings. If not provided and if variety name is not recognized in dependency table, an average value will be used. Variety ratings have a significant impact on model reliability, hence it is strongly recommended to provide all those inputs. Cereal ratings must be provided as per Arvalis rating scales. https://choix-des-varietes.arvalis-infos.fr/bletendre/savoirplus#sp_etape3

`fields.crop_variety.traits.category`	String	Yes if `fields.crop_variety.traits` provided	Enum:
			`"agronomic"`
			`"genetic"`
				Trait can genetic traits or agronomic traits.

`fields.crop_variety.traits.type`	String	Yes if `fields.crop_variety.traits` provided		Currently supported: • disease_resistance: 1- variety is very sensitive to disease / 9- variety is highly resistant to disease • flowering_earlyness: 4- very late flowering / 9- very early flowering • elongation_earlyness: refer to BBCH 30, 0- very late elongation / 6- very early elongation • vernalization: 1- induction requires long exposure to cold temperature / 9- no temperature requirements for induction • height: 1- very short compared to standard / 9- very tall compared to standard • eyespot_gene
`fields.crop_variety.traits.disease_code`	String	No		Example: 'SEPTTR' EPPO code for disease. Optional in v1.0. Avizio will support avizio_code over disease_code for v1.0.
`fields.crop_variety.traits.value`				Value for each trait category.

fields.crop_variety.traits.attributes.avizio_code	String	Yes if fields.crop_variety.traits provided and fields.crop_variety.traits.type = "disease_resistance"		Example: 'SEP' It is disease code against which trait value is provided for a particular crop variety. Avizio will support avizio_code over disease_code for v1.0.
fields.planting	Object	Yes		Details related to planting. Synonyms: sowing, drilling.
fields.planting.date	String<Datetime>	Yes	Full-date notation as defined by RFC 3339, section 5.6 for example, 2021-09-03T00:00:00Z	Date when crop seeds were introduced in the field. For Cereals in EAME region: planting date must be within September 1st and December 31st.
fields.water_supply	Object	No		Details about water supply.
fields.water_supply.irrigation	Object	Yes if fields.water_supply is provided. otherwise No		Log of irrigation dates and water volume supplied.
fields.water_supply.irrigation.date	String<Datetime>	Yes if fields.water_supply.irrigation is provided.	Full-date notation as defined by RFC 3339, section 5.6 for example, 2021-09-03T00:00:00Z	Date of irrigation.

`fields.water_supply.irrigation.volume`	Object	Yes if `fields.water_supply.irrigation` is provided.		Details about the water volume used for irrigation.
`fields.water_supply.irrigation.volume.value`	Float	Yes if `fields.water_supply.irrigation` is provided.		Value of volume of water.
`fields.water_supply.irrigation.volume.unit`	String	No	Only mm (millimeters) is supported.	Unit.
`fields.soil`	Object	Yes		Specific soil attributes and soil practices for requested field.
`fields.soil.practice`	String	Yes	Enum: `"TILL"` `"NO-TILL"`	Soil tillage practices carried out on the field prior to planting Synonyms: plowing, ploughing Currently supported: • TILL: if any type of tillage practice has been carried-out on the field, even if shallow. • NO-TILL: if no tillage at all and direct seeding.

`fields.soil.texture`	String	Yes	Enum: `"SANDY_CLAY" "CLAY" "SILTY_CLAY" "CLAY_LOAM" "SILTY_CLAY_LOAM" "SANDY_LOAM" "SANDY_CLAY_LOAM" "LOAM" "SILT_LOAM" "SILT" "LOAMY_SAND" "SAND"`	Soil texture category as per FAO/USDA soil taxonomy based on silt, clay and sand proportions.
`fields.soil.nitrogen`	Object	No		Amount of nitrogen fertilizer amended to the field prior to BBCH 30.
`fields.soil.nitrogen.value`	Float	Yes if `fields.soil.nitrogen` is provided otherwise No		
`fields.soil.nitrogen.unit`	String	No	Only `"kg/ha"`	Only kg/ha is supported as of now (also referred to as 'N units').
`fields.soil.calcareous`	Boolean	No		Should be true if CaCO3 > 10% or 100g/Kg.
`fields.soil.organic`	Boolean	No		Should be TRUE if Organic matter content > 3% or 30 g/Kg.
`fields.soil.observations`	Object	Yes		Object with multiple observations regarding: • yield_potential (REQUIRED) • crop_history (REQUIRED) • disease_occurrence • bbch_stages

`fields.soil.observations.[yield_potential].category`	String	Yes	one of the object in `observations` should have category as `"yield_potential"`	Observations regarding yield_potential category.
`fields.soil.observations.category=yield_potential.yield_potential`	Object	Yes		Estimated yield potential as per grower experience and field history.
`fields.soil.observations.category=yield_potential.yield_potential.value`	Float	Yes	Valid range: 0 - 200	
`fields.soil.observations.category=yield_potential.yield_potential.unit`	String	Yes	Only `q/ha` supported	
`fields.soil.observations.[crop_history].category`	String	Yes	one of the object in `observations` should have category as `"crop_history"`	Observations regarding crop_history category.
`fields.soil.observations.category=crop_history.values`	Object	Yes		Log of crops grown on requested field during previous seasons. Model requires data for 2 previous years. More than that is accepted but would not be considered for calculations.
`fields.soil.observations.category=crop_history.values.harvest_date`	String <Datetime>	Yes	Full-date notation as defined by RFC 3339, section 5.6 for example, 2021-09-03T00:00:00Z	Date when crop is harvested.

fields.soil.observations.category=crop_history.values.crops	Array<String>	Yes	Valid values for crops for harvest_date.year = planting.date - 1 (N1) and for harvest_date.year = planting.date - 2 (N2) can be found in catalogue	Example: 'CARROTS' List of crops harvested in the given year. For model we only need 1 crop for a given year. And we need 2 years of harvest data with one crop in each year. Valid values for N1 and N2.
fields.soil.observations.[bbch_stages].category	String	No	category = "bbch_stages"	Observations regarding crop_history category.
fields.soil.observations.category=bbch_stages.values	Object	Yes if category = "bbch_stages" is provided otherwise No		Ground truth crop stage observations as per field scouting. Providing this input will trigger model execution on actual crop status and increase reliability of outputs.
fields.soil.observations.category=bbch_stages.values.date	String<Datetime>	Yes if category = "bbch_stages" is provided otherwise No	Full-date notation as defined by RFC 3339, section 5.6 for example, 2021-09-03T00:00:00Z	Date of observed BBCH stage.
fields.soil.observations.category=bbch_stages.values.bbch	Float	Yes if category = "bbch_stages" is provided otherwise No		Observed BBCH stage of the crop.
fields.soil.observations.[disease_occurrence].category	String	No		Observations regarding disease_occurrence category.

fields.soil.observations.category=disease_occurrence.values	Object	Yes if category = "disease_occurrence" is provided otherwise No		Disease infestations observed in the field during previous or current season. If not provided, it will be calculated by the inoculum function Providing this input will improve reliability of outputs.
fields.soil.observations.category=disease_occurrence.values.season	String	Yes if category = "disease_occurrence" is provided otherwise No	Enum: "PREVIOUS" "CURRENT"	The year = “CURRENT“ or “PREVIOUS“ when the observation was made.
fields.soil.observations.category=disease_occurrence.values.crop	String	Yes if category = "disease_occurrence" is provided otherwise No	Enum: "WINTER_WHEAT" "WINTER_BARLEY" "SPRING_BARLEY" "SOFT_WHEAT" "HARD_WHEAT" "WHEAT" "BARLEY"	Crop on which the observation was done. Currently WINTER_WHEAT and WINTER_BARLEY are processed by model. If any other crop is given it will be accepted but would not be processed.
fields.soil.observations.category=disease_occurrence.values.disease_code	String	No		Example: 'SEPTTR' EPPO code for disease. Avizio will support avizio_code over disease_code for v1.0.
fields.soil.observations.category=disease_occurrence.values.attributes.avizio_code	String	Yes if category=disease_occurrence is provided otherwise No		Example: 'SEP' Avizio will support avizio_code over disease_code for v1.0.

`fields.soil.observations.category=disease_occurrence.values.value`	Float	Yes if `category=disease_occurrence` is provided otherwise No		Value for the observed disease.
`fields.applications`	Object	No		Log of crop protection treatments applied in the field during the season. If provided, only 3 applications will be considered. More are accepted but would not be used. If less than 3 applications are given model will assign remaining as null. Note: Currently entry for commercial formulated product, will evolve in future versions to active ingredient input.
`fields.applications.sequence`	Integer	Yes `fields.applications` is provided otherwise No		The time order in which application are done. First application means it is done before application 2 and 3.
`fields.applications.date`	String<Datetime>	Yes `fields.applications` is provided otherwise No	Full-date notation as defined by RFC 3339, section 5.6 for example, 2021-09-03T00:00:00Z	Date of treatment application.

`fields.applications.products`	Array<Object>	Yes `fields.applications` is provided otherwise No		List of products applied. If application is provided then maximum 3 product objects inside one application object is considered by the model currently.
`fields.applications.products.name`	String	Yes `fields.applications` is provided otherwise No		Example: AXIMA Commercial name of product applied. Must be recognized in Avizio SQL catalogue.
`fields.applications.products.code`	String	No		Currently not relevant. Placeholder for future version.
`fields.applications.products.rate_of_use`	Object	Yes `fields.applications` is provided otherwise No		The amount of product used.
`fields.applications.products.rate_of_use.value`	Float	Yes `fields.applications` is provided otherwise No		
`fields.applications.products.rate_of_use.unit`	String	No	Enum: `"L/ha"` `"kg/ha"`	L/ha (liter/hectare)and kg/ha currently supported.
`fields.applications.products.diseases`	Array<Object>	Yes `fields.applications` is provided otherwise No		The diseases for which the applied product is effective.

fields.applications.products.name	String	Yes fields.applications is provided otherwise No	Enum: "ANTI_PIETIN" "ANTI_OIDIUM" "ANTI_FUSARIOSE"	Product selectivity: • ANTI_PIETIN: foot rot • ANTI_OIDIUM: powdery mildew • ANTI_FUSARIOSE: fusarium
fields.applications.products.is_effective	Boolean	Yes fields.applications is provided otherwise No		true means product is effective on disease and false means it is not effective.

Response Schema

Parameter	Type	Constraint	Description
results._.location.type	String	Literal['Feature']	
results._.location.geometry	Object	GeoJSON	Geometry information in GeoJSON format
result._.location.geometry.type	String	Literal['Point']	GeoJSON *type* attribute. Only *Point* type is supported.
result._.location.geometry.coordinates	Array <Float>	Pair[Longitude, Latitude]	Array with two values, first is for longitude and second is for latitude.

result._.location.properties.grid_id	Integer		An id that is mapped to a specific field. Overrides geometry for the location information.
result._.metadata.type	String	Literal['mathematical']	Describes the type of the model.
result._.metadata.models._.name	String	Literal['avizio']	Name of the models served by API.
result._.metadata.models._.version	String		Version of the model.
result._.metadata.result_time	Integer		Epoch Time of computation of result.
result._.predictions	Array		Array of predictions object. Each individual object describes individual predictions.

`result._.predictions._.feature_category`	String	*List of features categories, features and other associated information like units and attributes can be found here.*	It acts as an identifier for category in which related features are grouped.
`result._.predictions._.features`	Array		List of features associated with a specific category.
`result._.predictions._.features._.type`	String	*List of features categories, features and other associated information like units and attributes can be found here.*	Type of associated *feature*
`result._.predictions._.features._.value`	Number	*List of features categories, features and other associated information like units and attributes can be found here.*	Value of the associated *feature*
`result._.predictions._.features._.unit`	String	*List of features categories, features and other associated information like units and attributes can be found here.*	Unit of the value for the associated *feature*
`result._.predictions._.features._.attributes`	Object	`{Key: Value}`	Optional attributes for the associated feature. This key-value pairs that provide additional relevant information.

result._.predictions._.features._.attributes.disease_code	String	For understanding attributes and feature association refer to *this*.	Unique Code of Disease associated with the feature.
result._.predictions._.features._.attributes.plant_compartment	String	For understanding attributes and feature association refer to *this*.	Unique Code of Plant Compartment associated with the feature.
result._.predictions._.features._.attributes.number_of_days_after_planting	Integer	For understanding attributes and feature association refer to *this*.	Number of days after planting the mentioned stage of the crop is observed. It is associated with the *avg_bbch_by_plot* of *crop_stage* category.

`result._.predictions._.features._.attributes.predicted_date`	String	For understanding attributes and feature association refer to *this*.	Predicted date at which the stage of a crop is observed. It is associated with the *avg_bbch_by_plot* of *crop_stage* category.
`result._.predictions._.prediction_window`	Object		Object providing information for the period associated with the respective prediction.
`result._.predictions._.prediction_window.type`	String	`Literal['forecast']`	Type of prediction window. For Avizio, It is *forecast*.

result._.predictions._.prediction_window.frequency	String	Literal['Daily','Weekly','Monthly','Yearly','Seasonl','Custom']	Frequency of Prediction of the associated feature. For Avizio, it is *Daily* for the associated features.
result._.predictions._.prediction_window.date	String <datetime>	Format: 'yyyy-mm-ddThh:mm:ssZ'	Date of predicted feature.

Errors

Code	Internal Code	Target	Type	Message
400	INS_IN_VAL_0001	request body at char: request_version	required	Required Field request_version is missing.
400	INS_IN_VAL_0002	request body at char: fields.0.models.0.metadata.business_rule_country_code	invalid	value is not a valid enumeration member; permitted: 'FR'
500	INS_EV_SER_0001	plot_yf	server_error, model_error	\"None of [Index(['date', 'temperature_c_2_m_above_gnd_min'
500	INS_OUT_AVIZIO_0001	results.0.location.properties.grid_id	invalid	field required
500	INS_OUT_AVIZIO_0002	results.0.location	required	field required

3.4.3 Bird cherry-oat Aphid ML weekly traps models

Surveys are crucial for monitoring insect activity, crop pest levels, and are widely used in pest management programs. In autumn fields crops just emerging as wheat, barley are very attractive for insects foraging. Aphids (*Rhopalosiphum padi*) prevail mainly on cereals and cause damage due to their food bites. At the end of vegetative cycle, are even more harmful they returns to winter cereal sowing to which they can transmit certain viral diseases as BYDV (Barley Yellow Dwarf Virus).

Request Schema

Parameter	Type	Required	Default	Constraints	Description
geometry.type	string	Yes		= Point	
geometry.coordinates	array [float]	Yes		In [FR, UK]	Valid Longitude and Latitude values should be specified. It will automatically define country. Just France and United Kingdom are supported by this service.
forecast_period.start_date	string<date>	No	Today	Max 7 days from today; Must be within 1 September and 31 December	
forecast_period.end_date	string<date>	No	Today + 6	Max 7 days from today; Must be within 1 September and 31 December	

landscape.woodland.value	float	No	Closest value found in LANDSCAPE_COLLECTION_NAME	0<= X <= 100	LANDSCAPE_COLLECTION_NAME is a static table, which can be queried by coordinate. Percentage of the 5KM grid which are covered by Woodland.
landscape.grassland.value	float	No	Closest value found in LANDSCAPE_COLLECTION_NAME	0<= X <= 100	LANDSCAPE_COLLECTION_NAME is a static table, which can be queried by coordinate. Percentage of the 5KM grid which are covered by Grassland.
trapped_aphids	integer	No		X >= 0	Number of trapped aphids in the last 7 days before the forecast. As shown in the section “Agroservice Reliability”, inputting a number of trapped aphids improves the models' accuracy.

Multipoint Predictions for an interval (NOT RECOMMENDED FOR BULK TRANSACTIONS)

Parameter	Type	Required	Default	Constraints	Description
geometry.type	string	Yes		= `MultiPoint`	
geometry.coordinates	array [float]	Yes		In [FR, UK]	Valid Longitude and Latitude values should be specified. It will automatically define country. Just France and United Kingdom are supported by this service.
forecast_period.start_date	string<date>	No	Today	Max 7 days from today; Must be within 1 September and 31 December	Planting date (syn. sowing date, drilling date)

forecast_period.end_date	string<date>	No	Today + 6	Max 7 days from today; Must be within 1 September and 31 December	

Response Schema

Model Output	Type	Value constraints	Description
GLMmodel_regression	integer	0 - inf	Linear regression model - Return the estimated number of aphid(s)/week of a yellow trap (FR) or suction traps (UK) that would be detected on the requested date
GLMmodel_poly_regression	Integer	0 - inf	Polynomial regression model - Return the estimated number of aphid(s)/week of a yellow trap (FR) or suction traps (UK) that would be detected on the requested date
MLPmodel_regression	integer	0 - inf	Neural regression model - Return the estimated number of aphid(s)/week of a yellow trap (FR) or suction traps (UK) that would be detected on the requested date
ALLmodel_regression	float	0 - inf	Average of the previous 3 regression models - Return the estimated number of aphid(s)/week of a yellow trap (FR) or suction traps (UK) that would be detected on the requested date
MLPmodel_classifier	float	FR: 0 = [0-10] or 1 = [10-100] UK: 0 = [0-247] or 1 = [247-inf]	Classified linear regression model
GLMmodel_regression_binarize	float	FR: 0 = [0-10] or 1 = [10-100] UK: 0 = [0-247] or 1 = [247-inf]	Classified linear regression model
GLMmodel_regression_poly_binarize	float	FR: 0 = [0-10] or 1 = [10-100] UK: 0 = [0-247] or 1 = [247-inf]	Classified polynomial regression model

MLPmodel_regression_binarize	float	FR: 0 = [0-10] or 1 = [10-100] UK: 0 = [0-247] or 1 = [247-inf]	Probability. Classified Neural regression model
ALLmodel_regression_binarize	float	FR: 0 = [0-10] or 1 = [10-100] UK: 0 = [0-247] or 1 = [247-inf]	Probability. Average of the previous 4 classification models

Errors

Internal Code	Target	Type	Message	Description
INS_IN_VAL_0001	any	required	required field missing	
INS_IN_VAL_0002	any	invalid	invalid input field	
INS_APHID_VAL_0001	geometry.coordinates	invalid	Country CA is not allowed	
INS_APHID_VAL_0002	forecast_period	invalid	Allowed forecast period is between 1st september and 31st december. Please specify start_date and end_date in forecast_period.	
INS_APHID_VAL_0003	forecast_period	invalid	Forecast period should be less than equal to 7 days	
INS_APHID_VAL_0004	landscape	invalid	The filter on the landscape table returns empty	
INS_APHID_VAL_0005	landscape.woodland.value	invalid	The woodland parameter should be in the range of values from 0 to 100	
INS_APHID_VAL_0006	landscape.grassland.value	invalid	The grassland parameter should be in the range of values from 0 to 100	
INS_APHID_VAL_0007	forecast_period	invalid	Invalid forecast_period: end_date < start_date	
INS_APHID_VAL_0008	trapped_aphids	invalid	Invalid trapped_aphids value, must be >= 0	
INS_APHID_VAL_0009	forecast_period.start_date	invalid	start_date cannot be later than 7 days from today	
INS_APHID_VAL_0010	forecast_period.end_date	invalid	end_date cannot be later than 7 days from today	

INS_EV_SER_0001	external.api.call	server_error	Unable to connect to DocumentDB Weather data not available	

3.5 Crop Management Models

3.5.1 Sunflower Herbicide

The Herbicide Sunflower Logic is a decision tree created with the expert knowledge from local and regional Technical leads. The decision tree contains different Herbicide strategies to control weeds in sunflower fields. Each control strategy is made out of different Herbicides and applications timings, and can have different levels of control for the key weeds impacting Sunflower fields. Since usually more than 1 weed species is present in a field, the logic contains a simple formula to calculate the theoretically level of control that each strategy can offer for different weed combinations, aka “weed basket”. That calculations is then used to select the best possible strategy for the given weed basket.

Note: Please contact Insights Engine team to Grab Insights Engine token to access the Sunflower Model

Email: Chetan.kanda@syngenta.com
Email: yogesh.limaye@syngenta.com

Request Schema

Parameter	Type	Required	Constraints	Example	Business Description
request_version	string	Yes		v2.0	version number of standardize request structure/template
fields.models.name	string	Yes		sunflower_herbicide_strategies	name of the model to be executed
fields.models.version	string	Yes		v1.0	version of the model to be executed

fields.location.type	string	Yes	Feature	Feature	The geographic objects of API use the GeoJSON (RFC 7946) format. The regions and fields can be represented as GeoJSON Features with a specific property schema. example: Feature
fields.location.geometry.type	string	Yes	Point	Point	Only GeoJSON Point type is supported
fields.location.geometry.coordinates	array [long,lat]	Yes	Longitude in (-180, 180) Latitude in (-90, 90)	[19.5058, 47.161]	In geometry, coordinates should be in sequence. First place is for longitude and second place is for latitude. Example : [-0.97024, 53.5282] Currently supported countries: • HU: Hungary
fields.crop	string	Yes	SUNFLOWER	SUNFLOWER	For this model currently supported crop: • Sunflower

fields.observations.category	string	Yes	pest_infestation	pest_infestation	observations category example- pest_infestation Every "pest", which can be a weed, insect or fungus has an EPPO Code This model just supports Weeds as a Pest.
fields.observations.value.eppo_code	string	Yes		"ABUTH", "AGRRE", "AMAAL", "AMACL"	EPPO codes are computer codes developed for plants, pests (including pathogens) which are standard in agriculture and plant protection. Every "pest", which can be a weed, insect or fungus has an EPPO Code This model just supports Weeds as a Pest. https://gd.eppo.int/
fields.observations.value.control_difficulty	integer	Yes	Only in between 1,2,3	1	Difficulty in controlling each weed. We just accept 1,2,3 for this field. 1 = Easy 2 = Difficult 3 = Very Difficult

Response Schema

Parameter	Type	Required	Constraints	Example	Business Description
response_version	string	Yes		v2.0	version number of response structure/template
results.location.type	string	Yes	Feature	Feature	The geographic objects of API use the GeoJSON (RFC 7946) format. The regions and fields can be represented as GeoJSON Features with a specific property schema. example: Feature
results.location.geometry.type	string	Yes	Point	Point	Only GeoJson Point type is supported

results.location.geometry.coordinates	array [long,lat]	Yes	Longitude in (-180, 180) Latitude in (-90, 90)	[19.5058, 47.161]	In geometry, coordinates should be in sequence. First place is for longitude and second place is for latitude. Example : [-0.97024, 53.5282] Currently supported countries: • HU: Hungary
results.metadata.type	string	Yes	rule	rule	Type of model operating
results.metadata.models.name	string	Yes	sunflower_herbicide_strategies	sunflower_herbicide_strategies	model name
results.metadata.models.version	string	Yes	v1.0	v1.0	model version
results.metadata.result_time	integer	Yes	epoch time	1690806450	API response time in epoch seconds
results.predictions.feature_category	string	Yes	weed_control_strategy	weed_control_strategy	It denotes category of the features. predictions of weed control strategy

results.predictions.features.type	string	Yes		strategy_1	Internal sequential number, unique to each herbicide strategy. **strategy_{value} - this value is in between 1 to 56 for Hungary. Example - strategy_1**
results.predictions.features.value	string	Yes		1	value of the Strategy_ID. Value is in between 1 to 56 for Hungary.
results.predictions.features.attributes.efficacy	float	Yes		96	Theoretical control provided by the strategy for the given weed basket. This calculation is based on expert knowledge.
results.predictions.features.attributes.herbicide_tolerance_traits	array	Yes	[Sulfonylurea-tolerant, CLEARFIELD, CLEARFIELD PLUS, CONVENTIONAL, A.I.R.]	["CLEARFIELD PLUS","AIR"]	Shows with which seed technology (traits) the herbicide strategies are compatible.

results.predictions.features.attributes.cost_level	integer	Yes	Only in between 1,2,3	1	Reference to the cost of the products contained in the strategy. 1 - Low cost 2 - Average cost 3 - High cost
results.predictions.features.attributes.number_of_applications	integer	Yes		3	Total number of application timings contained in each herbicide strategy. For herbicides in Sunflower it can vary from 2 to 4 timings.
results.predictions.features.attributes.herbicide_strategies	array	Yes			The strategies contains different herbicide product combinations which can be applied at 4 different timings.
results.predictions.features.attributes.herbicide_strategies.timing	string	Yes	"PRE EMERGENCE (BBCH <09)", "EARLY POST EMERGENCE (BBCH 12-14)", "POST (BBCH 14-16)", "LATE POST (> BBCH 16)"	"PRE EMERGENCE (BBCH <09)", "EARLY POST EMERGENCE (BBCH 12-14)", "POST (BBCH 14-16)", "LATE POST (> BBCH 16)"	Timing in which the application of the product or products combination must be carried out.

results.predictions.features.attributes.herbicide_strategies.products	array	Yes			List of products or active ingredients with their rates to be applied in each timing within a strategy.
results.predictions.features.attributes.herbicide_strategies.products.name	string	Yes		Dual Gold 960 EC	name of the products or active ingredients
results.predictions.features.attributes.herbicide_strategies.products.rate	float	Yes		1.5	Rate or dosage in which the product or product combination must be applied
results.predictions.features.attributes.products.unit	string	Yes		l/ha	Unit in which the rate is described.

Error Codes

Code	Internal Code	Target	Type	Message
400	INS_IN_VAL_0001	field	missing	Required {field} missing
400	INS_IN_VAL_0001	request_data	missing	Invalid observations entry. It must have category and values
400	INS_IN_VAL_0002	location	invalid	Required field geometry.type should be equal to Point
400	INS_IN_VAL_0002	location	invalid	Required field geometry.coordinate with a dimension equal to 2
400	INS_IN_VAL_0002	location	invalid	Invalid longitude range (-180,180): ' + str(v[0])
400	INS_IN_VAL_0002	location	invalid	Invalid latitude range (-90,90): ' + str(v[1])
400	INS_IN_VAL_0002	location	invalid	Current location not supported by this service
400	INS_IN_VAL_0002	crop	invalid	Invalid crop value. Must be SUNFLOWER

400	INS_IN_VAL_0002	request_data	invalid	observations field. It must be a non-empty list
400	INS_IN_VAL_0002	request_data	invalid	Invalid request version
400	INS_IN_VAL_0002	request_data	invalid	Weed not available in the database: {eppo_code}
400	INS_IN_VAL_0002	request_data	invalid	Invalid control_difficulty value. It must be 1, 2, or 3
400	INS_IN_VAL_0002	request_data	invalid	Invalid model name
400	INS_IN_VAL_0002	request_data	invalid	Invalid model version
404	INS_EV_SER_0004	type	invalid	url not found
500	INS_EV_SER_0001	db.connectivity	server_error	documentdb error