Service Template Specification

STS id field\_data

Version 0.3.6

Status PUBLISHED

Author(s) M. Caballero (AgriCircle AG)

**The "field\_data" Service Template**

Abstract

This document describes the specifications for field\_data services whose purpose is to store and make accessible a number of shareworthy intrinsic attributes and properties of a farmer's fields to authorized ATLAS network consumers.

Table of Contents

[1 Introduction 3](#_Toc104899428)

[2 Terminology 3](#_Toc104899429)

[3 Pre-requisites 3](#_Toc104899430)

[4 field\_data Use Scenarios 3](#_Toc104899431)

[4.1 Field Synchronisation 3](#_Toc104899432)

[4.2 Soil Fertilisation 4](#_Toc104899433)

[5 Service Template Functions 4](#_Toc104899434)

[5.1 Basic Functions 5](#_Toc104899435)

[5.1.1 List Fields 5](#_Toc104899436)

[5.1.2 Get Field Information 6](#_Toc104899437)

[5.1.3 List Groups 6](#_Toc104899438)

[5.2 Crop Information 7](#_Toc104899439)

[5.2.1 Get Crop History 7](#_Toc104899440)

[5.2.2 Get Crop Details 7](#_Toc104899441)

[5.3 Driving Path 8](#_Toc104899442)

[5.3.1 Get Driving Path 8](#_Toc104899443)

[5.4 Applications Results Functions 8](#_Toc104899444)

[5.4.1 Get Application Results 8](#_Toc104899445)

[5.4.2 Add Application Result 9](#_Toc104899446)

[5.5 Subscription Management Functions 9](#_Toc104899447)

[5.5.1 Subscribe 9](#_Toc104899448)

[5.5.2 Refresh a Subscription 9](#_Toc104899449)

[5.5.3 Unsubscribe 9](#_Toc104899450)

[6 Data Formats 10](#_Toc104899451)

[6.1 Application Results File Formats 10](#_Toc104899452)

[6.1.1 Fertilization Application Type 12](#_Toc104899453)

[6.1.2 Crop Protection Application Type 13](#_Toc104899454)

[6.1.3 Irrigation Application Type 14](#_Toc104899455)

[6.1.4 Harvesting Application Type 16](#_Toc104899456)

[6.1.5 Sowing Application Type 17](#_Toc104899457)

[6.1.6 Other Application Type 18](#_Toc104899458)

[7 Access and Authentication 19](#_Toc104899459)

[8 Dynamic Behaviour 19](#_Toc104899460)

[8.1 Subscriptions 19](#_Toc104899461)

[8.2 Subscription Notification 21](#_Toc104899462)

[ANNEX 1 – Crop Reference Data 22](#_Toc104899463)

# Introduction

Field data is information that is relevant to a large number of services and platforms in the ATLAS network. Services implementing the field\_data service template provide a reference storage (source of truth) from which other services and platforms may retrieve data or synchronize their data with.

The field\_data service template is designed with the idea that a single service instance is configured in all the services used by a farmer that require field information. Having a common field\_data service for all farmer's services means that field information MAY be exchanged via a "field reference" (URI) rather than with the actual data and that shareable information can be managed at a single location and be available for all consumers.

# Terminology

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "NOT RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in [RFC 2119](https://datatracker.ietf.org/doc/html/rfc2119) and indicate requirement levels for compliant implementations.

The notation "[xxx]" (xxx in square brackets) is equivalent to "array of xxx".

When used alone, the term "field\_data" refers to "field\_data service template". Instead, "field\_data service" is equivalent to "a service implementing the field\_data service template".

# Pre-requisites

A thorough understanding in the following is required for both service consumers or service implementors:

* GeoJSON specifications (<https://geojson.org/>)
* GeoPackage specifications (https://www.geopackage.org/)

# field\_data Use Scenarios

The following sections describe some aspects of the use of field\_data. The examples were chosen to illustrate the basic operation of applications using field\_data, not to limit what field\_data may be used for.

## Field Synchronisation

Farmers typically use several software platforms. It is often the case that a few of these platforms require information of field boundaries for their operations.

In ATLAS, field\_data services provide a single source of truth for field information but most existing software require a local copy of the data.

The field\_data subscriptions enable the establishment of a simple data mirroring mechanism so that all participating farmer's systems may update their local mirror as soon as changes occur in a field\_data service.

## Soil Fertilisation

As a fertilisation operation is performed on a field, the actual amounts of nutrients that have been applied (retrieved from the equipment) are updated on the field\_data service.

When another soil fertilisation operation is planned at a later time, a field\_data service may be queried for nutrients applied from prior operations in order to adjust the new prescription.

# Service Template Functions

This section provides a very high-level summary of the field\_data functions:

Basic Functions

List Fields

Get Field Information

List Groups

Crop Information

Get Crop History

Get Crop Details

Driving Path

Get Driving Path

Applications Results Functions

Get Application Result

Add Application Result

Subscription Management Functions

Subscribe

Refresh a Subscription

Unsubscribe

Note that there are no operations to update, create, delete fields. These operations are deliberately excluded from the field\_data template.

Implementations of field\_data are responsible for providing end-user configuration and management tools in their proprietary user interface, such as import from a shapefile, boundary editing in the browser, … Any change generated by these tools must generate an appropriate notification to subscribers (see 5.5 Subscription Management Functions).

Fields SHALL NOT be deleted under any condition (by the implementation's management application) so that references held to this field by external consumers remain valid over time. When a field is no longer relevant for new operations, it's archived property will be set to "true". Important note: an archived field can never be modified, including changing its archived state.

Within this section, operations are summarised with simple tables:

+--------------------------------------------+

| <logical operation name> |

+-------------+------------------------------+

| Inputs | <URL parameters or |

| | request body attributes |

+-------------+------------------------------+

| Outputs | <response body attributes> |

+-------------+------------------------------+

Only the most meaningful parameters are discussed in this document. Please refer to the OpenAPI specifications for full details.

field\_data services are not required to handle intense traffic from a single client, such as the one that may result from being directly invoked on user interface interactions in an FMIS, for instance. Implementors MAY generate a 429 TOO MANY REQUESTS error response if the rate of calls exceed some pre-defined quota. Clients requiring more intense field information are advised to implement a mirroring mechanism using the subscription functions, as outlined in 4.1 Field Synchronisation.

## Basic Functions

These operations return basic field information about a specific field or all fields in an account.

The basic field information includes the id, name and GeoJSON polygon boundary of a field. For detailed specifications, refer to the field\_data OpenAPI specifications.

### List Fields

This function returns the list of all fields (base information only) in an account

+--------------------------------------------+

| list\_fields |

+-------------+------------------------------+

| Inputs | optional group id |

+-------------+------------------------------+

| Outputs | [urn, field info] |

+-------------+------------------------------+

Archived fields SHALL NOT be returned by list\_fields.

### Get Field Information

This function returns the base information of a specific field.

+--------------------------------------------+

| get\_field\_info |

+-------------+------------------------------+

| Inputs | optional group id |

+-------------+------------------------------+

| Outputs | id, urn, name, boundaries |

+-------------+------------------------------+

Services MUST return information even when the field is archived.

A field urn MUST have the following format:

urn:<service\_id>:<field\_id>

where <service\_id> is the ATLAS id of the field\_data service on which the query was made. Field URNs are often passed as parameters to other types of services. These services must validate that the <service\_id> in the URN matches a field\_data service to which they are paired.

Boundaries are defined as a GeoJSON polygon feature where the "holes" represent areas with driving restrictions

(water, trees, fenced area, …). The coordinates MAY include elevation information relevant to drivability.

### List Groups

This function returns the list of groups defined in an account. Fields may be arbitrarily grouped for convenience in a field\_data service (i.e., annual crops, orchards, vineyards …) in order to optimize the queries for relevant fields. The group definition and configuration, if any, should be provided by the vendor's proprietary configuration and management tools.

+--------------------------------------------+

| list\_groups |

+-------------+------------------------------+

| Inputs | none |

+-------------+------------------------------+

| Outputs | [id, name, description] |

+-------------+------------------------------+

A specific implementation may elect not to provide groups in which case they must simply return an empty array.

## Crop Information

### Get Crop History

Returns information on crops over time in the field.

+--------------------------------------------+

| get\_crop\_history |

+-------------+------------------------------+

| Inputs | field id |

+-------------+------------------------------+

| Outputs | [crop details] |

+-------------+------------------------------+

Services MUST return the history in chronological order (sorted by ascending dates)

Crop history information MUST be immutable except for the most current item which may be edited.

### Get Crop Details

+--------------------------------------------+

| get\_crop\_details |

+-------------+------------------------------+

| Inputs | field id, date |

+-------------+------------------------------+

| Outputs | date, crop id, layout, |

| | season\_start\_date |

+-------------+------------------------------+

Services MUST return the crop information corresponding to the most recent information whose date is smaller or equal to the requested date.

If no crop details are available, services MUST return 204 NO CONTENT.

If date is not specified, the current (latest) crop details are returned.

If date is specified it MUST NOT be in the future.

If no date match is found (date is older that oldest crop details), services MUST return 204 NO CONTENT.

The crop layout represents the "seedable" areas of the field, such as vineyard rows, individual trees, etc. The coordinates MAY include elevation information relevant to the type of crop.

Crop layout, if present, MUST be fully contained within the field boundaries.

season\_start\_date is the date at which the latest crop season starts. When available, it MUST NOT be older than 9 months from the current date. Otherwise, it MUST be omitted (even if an outdated value is available in the service).

## Driving Path

### Get Driving Path

Only "AB Line" are supported at the moment. By convention, the AB line must be aligned with the edge of a field boundary; it will be dynamically adjusted by the consumer to take the width of the attached implement into account.

+--------------------------------------------+

| get\_driving\_path |

+-------------+------------------------------+

| Inputs | field id |

+-------------+------------------------------+

| Outputs | start point coordinates, |

| | end point coordinates |

+-------------+------------------------------+

If not driving path is defined or not supported, services MUST return 204 NO CONTENT error.

## Applications Results Functions

Application results are operations that are performed on a field, yielding results in the form of application maps, that are necessary for reporting or as inputs for other services. At this time, field\_data defines specific application results formats for "fertilization", "crop\_protection", "irrigation", "harvesting" and "sowing". A general-purpose type "other" is available for types of applications that have not yet been formalized (see 6.1 Application Results File Formats).

The results of an application are always in GeoPackage format but with conventions that are specific to each application type.

### Get Application Results

Returns the information on specific nutrient amounts that have been applied to a field

+--------------------------------------------+

| get\_application\_results |

+-------------+------------------------------+

| Inputs | field id, application\_type, |

| | from\_date |

+-------------+------------------------------+

| Outputs | [date, application metadata, |

| | gpkg download url] |

+-------------+------------------------------+

### Add Application Result

Stores application data for resulting from a specific application type performed on a field. The applications are immutable and cannot be updated or deleted.

+--------------------------------------------+

| add\_application\_result |

+-------------+------------------------------+

| Inputs | field id, gpkg |

+-------------+------------------------------+

| Outputs | none |

+-------------+------------------------------+

A field\_data service MUST validate that gpkg (application result GeoPackage) is in valid format (see 6.1 Application Results File Formats.

## Subscription Management Functions

### Subscribe

Registers a notification URL that will be invoked by the field service whenever information about a field is changed. Subscriptions may automatically expire after some time and must be explicitly refreshed by their creator to remain in the system.

+--------------------------------------------+

| subscribe |

+-------------+------------------------------+

| Inputs | Notification URL |

+-------------+------------------------------+

| Outputs | Subscription id, |

| | expiration date |

+-------------+------------------------------+

### Refresh a Subscription

Refresh a subscription to prevent it from being automatically cancelled.

+--------------------------------------------+

| refresh\_subscription |

+-------------+------------------------------+

| Inputs | Subscription id |

+-------------+------------------------------+

| Outputs | expiration date |

+-------------+------------------------------+

### Unsubscribe

Refresh a subscription to prevent it from being automatically cancelled.

+--------------------------------------------+

| unsubscribe |

+-------------+------------------------------+

| Inputs | Subscription id |

+-------------+------------------------------+

| Outputs | none |

+-------------+------------------------------+

# Data Formats

This section focuses on the description of binary (file) data formats. Please refer to the field\_data OpenAPI specifications for details on all other payload and parameter descriptions.

## Application Results File Formats

Application result data MUST be in GeoPackage (<https://www.geopackage.org/>) format version 1.2 or newer.

+=========================================================+

| TABLE gpkg\_contents |

+=====+=============+=====================================+

| ROW | COLUMN | VALUE |

+-----+-------------+-------------------------------------+

| | table\_name | "atlas" |

| +-------------+-------------------------------------+

| | data\_type | "attributes" |

+-----+-------------+-------------------------------------+

| | table\_name | "application\_info" |

| +-------------+-------------------------------------+

| | data\_type | "attributes" |

+-----+-------------+-------------------------------------+

All Atlas GeoPackage files MUST contain an attributes table named "atlas" with the following structure:

+============================+

| TABLE atlas |

+================+===========+

| COLUMN | TYPE |

+----------------+-----------+

| id | INTEGER |

| type | TEXT |

| participant | TEXT |

| format\_version | TEXT |

+----------------+-----------+

and a single row:

+=========================================================+

| TABLE atlas |

+=====+================+==================================+

| ROW | COLUMN | VALUE |

+-----+----------------+----------------------------------+

| | type | "application\_result" |

| +----------------+----------------------------------+

| | participant | "<atlas participant id>" |

| 1 +----------------+----------------------------------+

| | application | "…" |

| +----------------+----------------------------------+

| | format\_version | "MAJOR.MINOR" |

+-----+----------------+----------------------------------+

Services SHALL validate that the "atlas.participant" field matches the information attached to the authentication context in which the file is uploaded.

If the GeoPackage was generated and uploaded by an Atlas service, then "atlas.application" SHOULD be "<service name>-<service version>", otherwise it MAY contain "<application name>-<application version>". In either case, no validation will be performed.

The "atlas.format\_version" field MUST be the version of the field\_data template that is targeted by the client uploading the file. GeoPackage files of a given type are guaranteed to be compatible (no breaking changes) for a same MAJOR version of "atlas.format\_version".

field\_data services MUST validate that the "atlas.type" column is equal to "application\_result".

By general convention, all application result GeoPackage files MUST contain an attributes table named "application\_info" with the following structure:

+============================+

| TABLE application\_info |

+================+===========+

| COLUMN | TYPE |

+----------------+-----------+

| id | INTEGER |

| type | TEXT |

| date | DATETIME |

| info | TEXT |

+----------------+-----------+

and a single row:

+=========================================================+

| TABLE application\_info |

+=====+=============+=====================================+

| ROW | COLUMN | VALUE |

+-----+-------------+-------------------------------------+

| | type | "fertilization|crop\_protection| |

| | | irrigation|harvesting|sowing|other" |

| 1 +-------------+-------------------------------------+

| | date | "yyyy-MM-ddTHH:mm:ssZ" |

| +-------------+-------------------------------------+

| | info | "<type-specific info (JSON)>" |

+-----+-------------+-------------------------------------+

Additional application type-specific conventions, within the GeoPackage standard, are defined below. Service implementations MUST perform basic validity checks on submitted GeoPackage files to make sure they comply with the requirements of this section.

### Fertilization Application Type

In addition to the base tables, a fertilization application result GeoPackage MUST contain a product data table.

+=========================================================+

| TABLE gpkg\_contents |

+=====+=============+=====================================+

| ROW | COLUMN | VALUE |

+-----+-------------+-------------------------------------+

| … |

+-----+-------------+-------------------------------------+

| | table\_name | "product" |

| +-------------+-------------------------------------+

| | data\_type | "2d-gridded-coverage|features" |

+-----+-------------+-------------------------------------+

If the data\_type of the "product" table is "features", then the columns must be as follows:

+=========================================================+

| TABLE product |

+=====+=================+=================================+

| ROW | COLUMN | VALUE |

+-----+-----------------+---------------------------------+

| | id | "INTEGER" |

| +-----------------+---------------------------------+

| | geometry | "GeoPackage Geometry" |

| +-----------------+---------------------------------+

| | amount | "amount in product units (REAL)"|

+-----+-----------------+---------------------------------+

Instead, if the data\_type of the product table is "2d-gridded-coverage" then it must be defined as follows:

+=========================================================+

| TABLE product |

+=====+=================+=================================+

| ROW | COLUMN | VALUE |

+-----+-----------------+---------------------------------+

| | id | "INTEGER" |

| +-----------------+---------------------------------+

| | zoom\_level | "INTEGER" |

| +-----------------+---------------------------------+

| | tile\_column | "INTEGER" |

| +-----------------+---------------------------------+

| | tile\_row | "INTEGER" |

| +-----------------+---------------------------------+

| | tile\_data | "BLOB (TIFF)" |

+-----+-----------------+---------------------------------+

Where tile\_data is a tiff blob holding Float32 values as per GeoPackage specifications.

In either case, the product values are in the units specified in "application\_info.info.product.unit".

Fertilization "application\_info.info" is a JSON object serialized as a string, such as:

+-----------------------------------------------+

| { |

| "product": { |

| "name": "optional name", |

| "unit": "kg|l", |

| "nutrients": { |

| "N": <amount\_in\_grams\_per\_unit>, |

| "P2O5": <amount\_in\_grams\_per\_unit>, |

| "K2O": <amount\_in\_grams\_per\_unit>, |

| … |

| } |

| }, |

| "amount": <amount\_in\_product\_unit> |

| } |

+-----------------------------------------------+

The nutrient identifier (e.g., P2O5) are defined in the fertilisation\_advisor service template. The "amount" attribute is the total amount of product that was applied on the entire field; it must match the sum of all values in the "product" table.

### Crop Protection Application Type

In addition to the base tables, a crop protection application result GeoPackage MUST contain a product data table.

+=========================================================+

| TABLE gpkg\_contents |

+=====+=============+=====================================+

| ROW | COLUMN | VALUE |

+-----+-------------+-------------------------------------+

| … |

+-----+-------------+-------------------------------------+

| | table\_name | "product" |

| +-------------+-------------------------------------+

| | data\_type | "2d-gridded-coverage|features" |

+-----+-------------+-------------------------------------+

If the data\_type of the "product" table is "features", then the columns must be as follows:

+=========================================================+

| TABLE product |

+=====+=================+=================================+

| ROW | COLUMN | VALUE |

+-----+-----------------+---------------------------------+

| | id | "INTEGER" |

| +-----------------+---------------------------------+

| | geometry | "GeoPackage Geometry" |

| +-----------------+---------------------------------+

| | amount | "amount in product units (REAL)"|

+-----+-----------------+---------------------------------+

Instead, if the data\_type of the product table is "2d-gridded-coverage" then it must be defined as follows:

+=========================================================+

| TABLE product |

+=====+=================+=================================+

| ROW | COLUMN | VALUE |

+-----+-----------------+---------------------------------+

| | id | "INTEGER" |

| +-----------------+---------------------------------+

| | zoom\_level | "INTEGER" |

| +-----------------+---------------------------------+

| | tile\_column | "INTEGER" |

| +-----------------+---------------------------------+

| | tile\_row | "INTEGER" |

| +-----------------+---------------------------------+

| | tile\_data | "BLOB (TIFF)" |

+-----+-----------------+---------------------------------+

Where tile\_data is a tiff blob holding Float32 values as per GeoPackage specifications.

In either case, the product values are in the units specified in "application\_info.info.product.unit".

Crop protection "application\_info.info" is a JSON object serialized as a string, such as:

+-----------------------------------------------+

| { |

| "product": { |

| "name": "optional name", |

| "unit": "kg|l", |

| "substances": { |

| "80-56-8": |

| <amount\_in\_grams\_per\_unit>, |

| … |

| } |

| }, |

| "amount": <amount\_in\_product\_unit> |

| } |

+-----------------------------------------------+

The substance identifier (e.g., 80-56-8) are defined in the crop\_protection\_advisor service template. The "amount" attribute is the total amount of product that was applied on the entire field; it must match the sum of all values in the "product" table.

### Irrigation Application Type

In addition to the base tables, an irrigation application result GeoPackage MUST contain a water data table.

+=========================================================+

| TABLE gpkg\_contents |

+=====+=============+=====================================+

| ROW | COLUMN | VALUE |

+-----+-------------+-------------------------------------+

| … |

+-----+-------------+-------------------------------------+

| | table\_name | "water" |

| +-------------+-------------------------------------+

| | data\_type | "2d-gridded-coverage|features" |

+-----+-------------+-------------------------------------+

If the data\_type of the "water" table is "features", then the columns must be as follows:

+=========================================================+

| TABLE water |

+=====+=================+=================================+

| ROW | COLUMN | VALUE |

+-----+-----------------+---------------------------------+

| | id | "INTEGER" |

| +-----------------+---------------------------------+

| | geometry | "GeoPackage Geometry" |

| +-----------------+---------------------------------+

| | amount | "amount in product units (REAL)"|

+-----+-----------------+---------------------------------+

Instead, if the data\_type of the water table is "2d-gridded-coverage" then it must be defined as follows:

+=========================================================+

| TABLE water |

+=====+=================+=================================+

| ROW | COLUMN | VALUE |

+-----+-----------------+---------------------------------+

| | id | "INTEGER" |

| +-----------------+---------------------------------+

| | zoom\_level | "INTEGER" |

| +-----------------+---------------------------------+

| | tile\_column | "INTEGER" |

| +-----------------+---------------------------------+

| | tile\_row | "INTEGER" |

| +-----------------+---------------------------------+

| | tile\_data | "BLOB (TIFF)" |

+-----+-----------------+---------------------------------+

Where tile\_data is a tiff blob holding Float32 values as per GeoPackage specifications.

In either case, the water values are in m3.

Irrigation "application\_info.info" is a JSON object serialized as a string, such as:

+-----------------------------------------------+

| { |

| "amount": <amount\_in\_m3> |

| } |

+-----------------------------------------------+

The "amount" attribute is the total amount of water that was applied on the entire field; it must match the sum of all values in the "water" table.

### Harvesting Application Type

In addition to the base tables, a harvesting application result GeoPackage MUST contain a yield table.

+=========================================================+

| TABLE gpkg\_contents |

+=====+=============+=====================================+

| ROW | COLUMN | VALUE |

+-----+-------------+-------------------------------------+

| … |

+-----+-------------+-------------------------------------+

| | table\_name | "yield" |

| +-------------+-------------------------------------+

| | data\_type | "2d-gridded-coverage|features" |

+-----+-------------+-------------------------------------+

If the data\_type of the "yield" table is "features", then the columns must be as follows:

+=========================================================+

| TABLE yield |

+=====+=================+=================================+

| ROW | COLUMN | VALUE |

+-----+-----------------+---------------------------------+

| | id | "INTEGER" |

| +-----------------+---------------------------------+

| | geometry | "GeoPackage Geometry" |

| +-----------------+---------------------------------+

| | amount | "amount in crop units (REAL)" |

+-----+-----------------+---------------------------------+

Instead, if the data\_type of the yield table is "2d-gridded-coverage" then it must be defined as follows:

+=========================================================+

| TABLE yield |

+=====+=================+=================================+

| ROW | COLUMN | VALUE |

+-----+-----------------+---------------------------------+

| | id | "INTEGER" |

| +-----------------+---------------------------------+

| | zoom\_level | "INTEGER" |

| +-----------------+---------------------------------+

| | tile\_column | "INTEGER" |

| +-----------------+---------------------------------+

| | tile\_row | "INTEGER" |

| +-----------------+---------------------------------+

| | tile\_data | "BLOB (TIFF)" |

+-----+-----------------+---------------------------------+

Where tile\_data is a tiff blob holding Float32 values as per GeoPackage specifications.

In either case, the product values are in the units specified in "application\_info.info.crop.unit".

Harvesting "application\_info.info" is a JSON object serialized as a string, such as:

+-----------------------------------------------+

| { |

| "crop": { |

| "crop\_id": "<crop\_id>", |

| "unit": "kg|count" |

| }, |

| "amount": <amount> |

| } |

+-----------------------------------------------+

The "crop\_id" attribute is defined as per ANNEX 1 – Crop Reference Data. The "amount" attribute is the total amount of crops (in the specified units) harvested in the entire field; it must match the sum of all values in the "yield" table.

### Sowing Application Type

In addition to the base tables, a sowing application result GeoPackage MUST contain a "seed\_or\_plant" table.

+=========================================================+

| TABLE gpkg\_contents |

+=====+=============+=====================================+

| ROW | COLUMN | VALUE |

+-----+-------------+-------------------------------------+

| … |

+-----+-------------+-------------------------------------+

| | table\_name | "seed" |

| +-------------+-------------------------------------+

| | data\_type | "2d-gridded-coverage|features" |

+-----+-------------+-------------------------------------+

If the data\_type of the "product" table is "features", then the columns must be as follows:

+=========================================================+

| TABLE seed |

+=====+=================+=================================+

| ROW | COLUMN | VALUE |

+-----+-----------------+---------------------------------+

| | id | "INTEGER" |

| +-----------------+---------------------------------+

| | geometry | "GeoPackage Geometry" |

| +-----------------+---------------------------------+

| | amount | "amount in seed units (REAL)" |

+-----+-----------------+---------------------------------+

Instead, if the data\_type of the product table is "2d-gridded-coverage" then it must be defined as follows:

+=========================================================+

| TABLE seed |

+=====+=================+=================================+

| ROW | COLUMN | VALUE |

+-----+-----------------+---------------------------------+

| | id | "INTEGER" |

| +-----------------+---------------------------------+

| | zoom\_level | "INTEGER" |

| +-----------------+---------------------------------+

| | tile\_column | "INTEGER" |

| +-----------------+---------------------------------+

| | tile\_row | "INTEGER" |

| +-----------------+---------------------------------+

| | tile\_data | "BLOB (TIFF)" |

+-----+-----------------+---------------------------------+

Where tile\_data is a tiff blob holding Float32 values as per GeoPackage specifications.

In either case, the seed values are in the units specified in "application\_info.info.seed.unit".

Sowing "application\_info.info" is a JSON object serialized as a string, such as:

+-----------------------------------------------+

| { |

| "seed": { |

| "crop\_id": "<crop\_id>", |

| "variety": "<optional variety name>", |

| "batch\_no": "<optional batch number>", |

| "unit": "kg|count", |

| "weight\_of\_1000": "<in kg>" |

| }, |

| "amount": <amount> |

| } |

+-----------------------------------------------+

The "crop\_id" attribute is defined as per ANNEX 1 – Crop Reference Data. The "weight\_of\_1000" attribute is the weight in kg for 1000 seeds; it can be used to map units between kg and count. The "amount" attribute is the total amount of seed (in the specified units) sowed in the entire field; it must match the sum of all values in the "seed" table.

### Other Application Type

The "other" application type is meant to provide a temporary solution for application types that have not yet been standardized.

In addition to the base tables, an "other" application result GeoPackage MAY contain an arbitrary number of additional data and attribute tables.

"Other" "application\_info.info" is an arbitrary JSON object serialized as a string with no defined convention. It is expected that the consumers and producers of these files will agree on private interpretation of the "application\_info.info" to carry out their operations.

field\_data services MUST store "other" application result GeoPackage files with only minimal generic validation.

# Access and Authentication

Farmers MUST have an account setup on a field\_data service in order to authenticate and access API endpoints. The service implementor is responsible for the creation of accounts; it is not covered in the service template specifications.

Unless specifically documented in the OpenAPI specifications, all operation must include credentials in form of Bearer authentication (also called token authentication). Clients can obtain an access token on behalf of their user from the service's authorization server (see ATLAS service pairing).

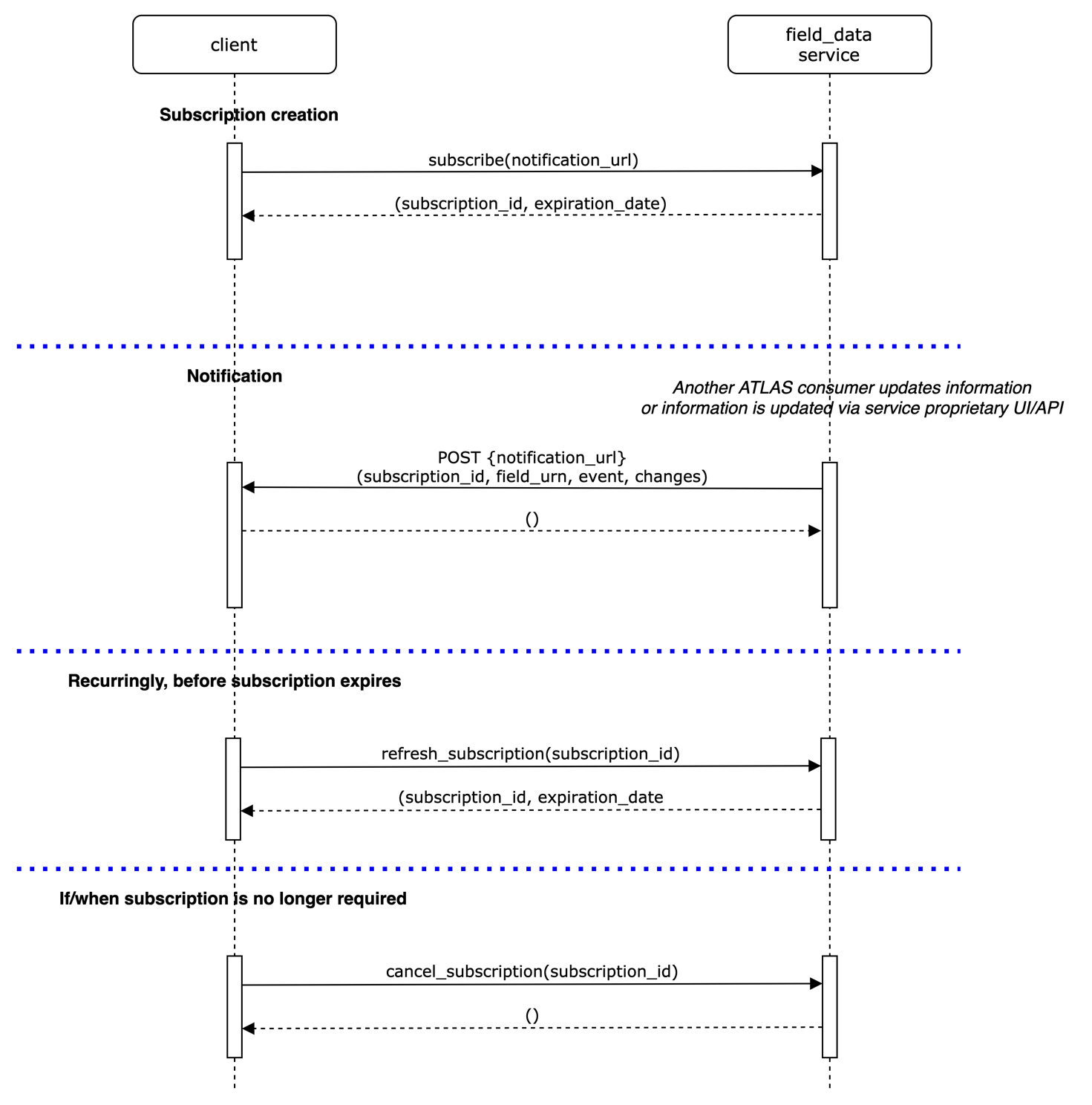
Some of the information held by field\_data services are considered to be sensitive from a GDPR perspective. The service's authorization server MUST request the client's end-user consent at service pairing time in order to deliver an access token.

# Dynamic Behaviour

The purpose of the diagrams in this section is to illustrate communication patterns, more complex than plain request/response API calls, that involve several interactions and/or asynchronous behaviour. Even though a sequence diagram representation is used, the diagrams are by no means to be interpreted as UML Sequence Diagrams. Specifically, in the spirit of focusing on functional behaviour and readability, error handling is deliberately not covered in the diagram.

## Subscriptions

The service template mandates a "notification event" pattern. In this pattern, only the information that an event (creation, update, deletion) has occurred will be sent (see Subscription Notification), and in case of updates, the kind of information that changed. It is up to the subscriber to decide whether the event is relevant and whether to perform a data query upon reception of the notification or not.



A notification is considered successful if the client returns a http result code 2XX.

Errors may occur during notification delivery. Depending on the type of error, services must react in different ways:

i) Network error - the connection to the client's host (from notification URL) cannot be established. The service MUST retry a certain number of times. The number of retries and possible backoff strategy is left at the discretion of the service implementer. If the

ii) Server errors (5XX result code) – these errors are potentially transient. The same strategy as for Network errors SHOULD be applied.

iii) Client errors (4XX result code) – typically when the notification URL is invalid, or the authentication is invalid/expired. 4XX errors should never be sent for transient client-side conditions and therefore services SHOULD NOT attempt retries.

Upon an excessive number of errors, services MAY automatically cancel a subscription. In that case, they SHOULD perform a "soft" cancellation where, instead of deleting the subscription record, a subscription can be reactivated upon reception of a refresh\_subscription call.

## Subscription Notification

The notification URL supplied by the subscriber MUST be invoked with an HTTP POST command.

When a notification authorisation parameter was supplied by the subscriber, the service MUST include a bearer token authentication in the form of an HTTP header:

Authorization: bearer <notification authorization>

A notification payload describing the type of event MUST be included:

{

"subscription\_id": "<my subscription id>",

"date": "<ISO 8601 UTC date/time>",

"field\_urn": "<target field URN>",

"event ": "<CREATED|UPDATED|DELETED>",

"changes": ["attributes"]

}

In case of an UPDATE event, services SHOULD include the name of the attributes that were changed in the form of attribute JSONPaths. This allows subscribers to filter relevant notifications based on the field data that is relevant to their purpose. In the absence of changed attributes information, the subscriber must assume that all attributes have changed.

Notifications are generated for the following "attributes":

* base
* crop (for any change in crop details or crop\_history)
* driving\_path (for any change in driving path)
* applications (after application results are added)

Implementations of field\_data are responsible for providing end-user configuration and management tools in their proprietary user interface, such as import from a shapefile, boundary editing in the browser, … Changes generated by these tools must generate an appropriate notification where relevant.

# ANNEX 1 – Crop Reference Data

Crop IDs in ATLAS are taken from the FAO's LIST OF CROPS, BOTANICAL NAME AND CODE NUMBER (see <http://www.fao.org/economic/the-statistics-division-ess/world-census-of-agriculture/programme-for-the-world-census-of-agriculture-2000/appendix-3-alphabetical-list-of-crops-botanical-name-and-code-number/en/>).

Only the crop id scheme is fundamentally relevant as a common reference. Software applications are free to provide their preferred names and translations.

|  |  |  |
| --- | --- | --- |
| CROP | BOTANICAL NAME | CROP ID |
| Abaca (Manila hemp) | Musa textilis | 2.2.4.1 |
| Alfalfa for fodder | Medicago sativa | 1.7.1.1 |
| Alfalfa for seed | Medicago sativa | 1.8.2 |
| Almond | Prunus dulcis | 2.1.7.1 |
| Anise seeds | Pimpinella anisum | 1.1.4.3.4 |
| Apple | Malus sylvestris | 2.1.2.1 |
| Apricot | Prunus armeniaca | 2.1.3.1 |
| Areca (betel nut) | Areca catechu | 2.2.7.6 |
| Arracha | Arracacia xanthorhiza | 1.2.9 |
| Arrowroot | Maranta arundinacea | 1.2.5 |
| Artichoke | Cynara scolymus | 1.5.1.0.1 |
| Artichoke, Jerusalem | Helianthus tuberosus | 1.7.5.0.2 |
| Asparagus | Asparagus officinalis | 1.5.1.0.2 |
| Avocado | Persea americona | 2.1.6.0.1 |
| Bajra (millet) | Pennisetum americanum | 1.1.5.1.2 |
| Bambara groundnuts | Vigna subterranea | 1.3.9 |
| Banana | Musa paradisiaca | 2.1.6.0.2 |
| Barley | Hordeum vulgare | 1.1.4 |
| Bean, dry, edible | Phaseolus vulgaris | 1.3.1 |
| Bean, harvested green | Phaseolus vigna spp. | 1.5.4.1 |
| Beet, fodder (mangel) | Beta vulgaris | 1.7.5.0.5 |
| Beet, red | Beta vulgaris | 1.5.3.5 |
| Beet, sugar | Beta vulgaris | 1.4.1.2 |
| Beet, sugar for fodder | Beta vulgaris | 1.7.5.0.6 |
| Bergamot | Citrus bergamia | 2.1.1.6 |
| Betel nut | Areca catechu | 2.2.7.6 |
| Blackberry | Rubus spp. | 2.1.5.7 |
| Black pepper | Piper nigrum | 2.2.3.1 |
| Black wattle | Acacia mearnsii | 2.2.5.2 |
| Blueberry | Vaccinium spp. | 2.1.5.1 |
| Brazil nut | Bertholletia excelsa | 2.1.7.6 |
| Breadfruit | Artocarpus altilis | 2.1.6.0.3 |
| Broad bean, dry | Vicia faba | 1.3.2 |
| Broad bean, harvested green | Vicia faba | 1.5.4.3 |
| Broccoli and cauliflower | Brassica oleracea botrytis | 1.5.5.1 |
| Broom millet | Sorghum bicolor | 1.4.5.4 |
| Broom sorghum | Sorghum bicolor | 1.4.5.4 |
| Brussels sprouts | Brassica oleracea gemmifera | 1.5.1.0.3 |
| Buckwheat | Fagopyrum esculentum | 1.1.9 |
| Cabbage (red, white, Savoy) | Brassica oleracea capitata | 1.5.1.0.4 |
| Cabbage, Chinese | Brassica rapa | 1.5.1.0.7 |
| Cabbage, for fodder | Brassica spp. | 1.7.5.0.1 |
| Cacao (cocoa) | Theobroma cacao | 2.2.1.2 |
| Cantaloupe and other melons | Cucumis melo | 1.5.2.0.1 |
| Caraway seeds | Carum carvi | 1.4.3.4 |
| Cardamom | Elettaria cardamomum | 2.2.3.7 |
| Cardoon | Cynara cardunculus | 1.5.1.1.5 |
| Carob | Ceratonia siliqua | 2.1.6.1.5 |
| Carrot, edible | Daucus carota | 1.5.3.1 |
| Carrot, for fodder | Daucus carota | 1.7.5.1.0 |
| Cashew nuts | Anacardium occidentale | 2.1.7.2 |
| Cassava (manioc) | Manihot esculenta | 1.2.3 |
| Castor bean | Ricinus communis | 1.4.2.1.0 |
| Cauliflower and broccoli | Brassica oleracea botrytis | 1.5.5.1 |
| Celeriac | Apium graveolens | 1.5.3.8 |
| Celery | Apium graveolens | 1.5.3.8 |
| Chayote | Sechium edule | 1.5.2.1.2 |
| Cherry (including sour) | Prunus spp. | 2.1.3.2 |
| Chesnut | Castanea sativa | 2.1.7.4 |
| Chickpea (gram pea) | Cicer arietinum | 1.3.4 |
| Chicory | Cichorium intybus | 1.4.5.2 |
| Chicory for greens | Cichorium intybus | 1.5.1.0.6 |
| Chilly, dry | Capsicum spp. | 1.4.3.1 |
| Chilly, fresh | Capsicum spp. | 1.5.2.0.2 |
| Chinese cabbage | Brassica rapa | 1.5.1.0.7 |
| Cinnamon | Cinnamomum verum | 2.2.3.2 |
| Citronella | Cymbopogon nardus | 2.2.7.1 |
| Citron | Citrus medica | 2.1.1.6 |
| Clementine | Citrus veticulata | 2.1.1.2 |
| Clover for fodder | Trifolium spp. | 1.7.1.2 |
| Clover for seed | Trifolium spp. | 1.8.3 |
| Clove | Syzygium aromaticum | 2.2.3.3 |
| Cocoa (cacao) | Theobroma cacao | 2.2.1.2 |
| Coconut | Cocos nucifera | 2.2.2.1 |
| Cocoyam | Colocasia spp.; Xanthosoma spp. | 1.2.8 |
| Coffee | Coffee spp. | 2.2.1.1 |
| Cola | Cola spp. | 2.2.1.6 |
| Colza (rapeseed) | Brassica napus | 1.4.2.0.5 |
| Corn (maize) | Zea mays | 1.1.3 |
| Corn (maize), for silage | Zea mays | 1.7.3.1 |
| Corn (sweet) | Zea mays | 1.5.5.2 |
| Corn salad | Valerianella locusta | 1.5.1.1.5 |
| Cotton | Gossypium spp. | 1.4.4.1 |
| Cottonseed | Gossypium spp. | 1.4.2.0.8 |
| Cowpea | Vigna unguiculata | 1.3.6 |
| Cowpea, harvested green | Vigna unguiculata | 1.5.4.3 |
| Cranberry | Vaccinium spp. | 2.1.5.2 |
| Cress | Lepidium sativum | 1.5.1.0.8 |
| Cucumber | Cucumis sativus | 1.5.2.0.3 |
| Currants | Ribes spp. | 2.1.5.3 |
| Custard apple | Annona reticulata | 2.1.6.0.4 |
| Dasheen | Colocasia esculenta | 1.2.8 |
| Dates | Phoenix dactylifera | 2.1.6.0.5 |
| Drumstick tree | Moringa oleifera | 2.2.3.8 |
| Dry, bean, edible | Phaseolus spp. | 1.3.1 |
| Dry broad bean | Vicia faba | 1.3.2 |
| Dry chilli | Capsicum spp. | 1.4.3.1 |
| Dry garlic | Allium sativum | 1.2.7 |
| Dry onion | Allium cepa | 1.2.6 |
| Dry pea, edible | Pisum sativum | 1.3.3 |
| Dry pepper | Capsicum spp. | 1.4.3.1 |
| Durum wheat | Triticum durum | 1.1.1.1.1 |
| Durum wheat | Triticum durum | 1.1.1.2.1 |
| Durra (sorghum) | Sorghum bicolor | 1.1.5.2 |
| Earth pea (bambara groundnut) | Vigna subterranea | 1.3.9 |
| Edo (eddoe) | Xanthosoma spp.; Colocasia spp. | 1.2.8 |
| Eggplant | Solanum melongena | 1.5.2.0.4 |
| Endive | Cichorium endivia | 1.5.1.0.9 |
| Esparto grass | Lygeum spartum | 1.4.5.4 |
| Fennel | Foeniculum vulgare | 1.5.1.1.5 |
| Fenugreek | Trigonella foenumgraecum | 1.3.9 |
| Fig | Ficus carica | 2.1.6.0.6 |
| Filbert (hazelnut) | Corylus avellana | 2.1.7.5 |
| Fique | Furcraea macrophylla | 2.2.4.6 |
| Flax for fibre | Linum usitatissimum | 1.4.4.2 |
| Flax for oil seed (linseed) | Linum usitatissimum | 1.4.2.0.3 |
| Fodder beet (mangel) | Beta vulgaris | 1.7.5.0.5 |
| Formio (New Zealand flax) | Phormium tenax | 2.2.4.4 |
| Garlic, dry | Allium sativum | 1.2.7 |
| Garlic, green | Allium sativum | 1.5.3.2 |
| Geranium | Pelargonium spp.; Geranium spp. | 1.4.3.2 |
| Ginger | Zingiber officinalis | 2.2.3.4 |
| Gooseberry | Ribes spp. | 2.1.5.4 |
| Gourd | Langenaria spp; Cucurbita spp. | 1.5.2.0.5 |
| Gram pea (chickpea) | Cicer arietinum | 1.3.4 |
| Grapefruit | Citrus paradisi | 2.1.1.4 |
| Grape | Vitis vinifera | 2.1.4 |
| Grapes for raisins | Vitis vinifera | 2.1.4.3 |
| Grapes for table use | Vitis vinifera | 2.1.4.2 |
| Grapes for wine | Vitis vinifera | 2.1.4.1 |
| Grass esparto | Lygeum spartum | 1.4.5.4 |
| Grass, orchard | Dactylis glomerata | 1.7.2.1 |
| Grass, Sudan | Sorghum bicolor | 1.7.2.1 |
| Green garlic | Allium sativum | 1.5.3.2 |
| Green onion | Allium cepa | 1.5.3.3 |
| Groundnut (peanut) | Arachis hypogaea | 1.4.2.0.1 |
| Guava | Psidium guajava | 2.1.6.0.7 |
| Guinea corn (sorghum) | Sorghum bicolor | 1.1.5.2 |
| Hazelnut (filbert) | Corylus avellana | 2.1.7.5 |
| Hemp fibre | Cannabis sativa | 1.4.4.3 |
| Hemp, Manila (abaca) | Musa textilis | 2.2.4.1 |
| Hempseed | Cannabis sativa | 1.4.2.1.1 |
| Hemp, sunn | Crotalaria juncea | 1.4.4.5 |
| Henequen | Agave fourcroydes | 2.2.4.2 |
| Henna | Lawsonia inermis | 1.4.5.4 |
| Hop | Humulus lupulus | 2.2.1.4 |
| Horse bean | Vicia faba | 1.3.2 |
| Horseradish | Armoracia rusticana | 1.5.3.8 |
| Hybrid maize | Zea mays | 1.1.3.1 |
| Indigo | Indigofera tinctoria | 1.4.5.4 |
| Jasmine | Jasminum spp. | 2.2.6.2 |
| Jerusalem artichoke | Helianthus tuberosus | 1.7.5.0.2 |
| Jute | Corchorus spp. | 1.4.4.4 |
| Jowar (sorghum) | Sorghum bicolor | 1.1.5.2 |
| Kale | Brassica oleracea acephala | 1.5.1.1.0 |
| Kapok | Ceiba pentandra | 2.2.4.3 |
| Kenaf | Hibiscus cannabinus | 1.4.4.5 |
| Kohlrabi | Brassica oleracea gongylodes | 1.5.3.6 |
| Lavender | Lavandula spp. | 1.4.3.2 |
| Leek | Allium ampeloprasum | 1.5.1.1.1 |
| Lemon | Citrus limon | 2.1.1.3 |
| Lemon grass | Cymbopogon citratus | 2.2.7.7 |
| Lentil | Lens culinaris | 1.3.5 |
| Lespedeza | Lespendeza spp. | 1.7.1.3 |
| Lettuce | Lactuca sativa | 1.5.1.1.2 |
| Liquorice | Glycyrrhiza glabra | 1.4.3.3 |
| Lime, sour | Citrus aurantifolia | 2.1.1.5 |
| Lime, sweet | Citrus limetta | 2.1.1.6 |
| Linseed (flax for oil seed) | Linum usitatissimum | 1.4.2.0.3 |
| Litchi | Litchi chinensis | 2.1.6.1.5 |
| Loquat | Eriobotrya japonica | 2.1.2.5 |
| Lupine | Lupinus spp. | 1.3.9 |
| Macadamia (Queensland nut) | Macadamia spp. | 2.1.7.6 |
| Mace and nutmeg | Myristica fragrans | 2.2.3.5 |
| Maguey | Agave atrovirens | 2.2.4.6 |
| Maize (corn) | Zea mays | 1.1.3 |
| Maize (hybrid) | Zea mays | 1.1.3.1 |
| Maize, ordinary | Zea mays | 1.1.3.2 |
| Maize (corn) for silage | Zea mays | 1.7.3.1 |
| Mandarin and tangerine | Citrus reticulata | 2.1.1.2 |
| Mangel (fodder beet) | Beta vulgaris | 1.7.5.0.5 |
| Mango | Mangifera indica | 2.1.6.0.8 |
| Manila hemp (abaca) | Musa textilis | 2.2.4.1 |
| Manioc (cassava) | Manihot esculenta | 1.2.3 |
| Maslin (mixed cereals) | Mixture of Triticum spp.; Secale cereale | 1.1.8 |
| Medlar | Mespilus germanica | 2.1.2.4 |
| Melon and cantaloupe | Cucumis melo | 1.5.2.0.1 |
| Millet, proso | Penicum miliaceum | 1.1.5.1.1 |
| Millet, bajra | Pennisetum americanum | 1.1.5.1.2 |
| Millet broom | Sorghum bicolor | 1.4.5.4 |
| Millet, bulrush | Pennisetum americanum | 1.1.5.1.1 |
| Millet, finger | Eleusine coracana | 1.1.5.1.1 |
| Millet, Italian | Setaria italica | 1.1.5.1.2 |
| Millet, Japanese | Echinochloa crusgalli | 1.1.5.1.2 |
| Millet, pearl | Pennisetum americanum | 1.1.5.1.1 |
| Millet and sorghum | Pennisetum americanum; Eleusine coracana; Sorghum bicolor | 1.1.5 |
| Mint | Mentha spp. | 1.4.3.2 |
| Mulberry for fruit | Morus spp. | 2.1.6.1.5 |
| Mulberry for silkworms | Morus alba | 2.2.7.2 |
| Mushrooms | Agaricus spp.; Pleurotus spp. Volvariella | 1.6.2.2 |
| Mustard | Brassica nigra; Sinapis alba | 1.4.2.0.6 |
| Nectarine | Prunus persica | 2.1.3.5 |
| New Zealand flax (formio) | Phormium tenax | 2.2.4.4 |
| Niger seed | Guizotia abyssinica | 1.4.2.1.1 |
| Nutmeg and mace | Myristica fragrans | 2.2.3.5 |
| Oats, for fodder | Avena spp. | 1.7.3.2 |
| Oats, for grain | Avena spp. | 1.1.6 |
| Oil palm | Elaeis guineensis | 2.2.2.2 |
| Okra | Abelmoschus esculentus | 1.5.2.0.6 |
| Olive | Olea europea | 2.2.2.3 |
| Onion, dry | Allium cepa | 1.2.6 |
| Onion, green | Allium cepa | 1.5.3.3 |
| Onion seed | Allium cepa | 1.8.5 |
| Opium | Papaver somniferum | 1.4.3.3 |
| Orange | Citrus sinensis | 2.1.1.1 |
| Orange, bitter | Citrus aurantium | 2.1.1.6 |
| Orchard grass | Dactylis glomerata | 1.7.2.1 |
| Ornamental plants | &nbsp; | 1.6.1.2 |
| Palm, oil | Elaeis guineensis | 2.2.2.2 |
| Palm, sago | Metroxylon spp. | 2.2.7.3 |
| Palmyra palm | Borassus flabellifer | 2.2.7.4 |
| Papaya (pawpaw) | Carica papaya | 2.1.6.0.9 |
| Parsnip | Pastinaca sativa | 1.5.3.8 |
| Peach | Prunus persica | 2.1.3.3 |
| Peanut (groundnut) | Arachis hypogaea | 1.4.2.0.1 |
| Pear | Pyrus communis | 2.1.2.2 |
| Pea, edible dry | Pisum sativum | 1.3.3 |
| Pea, harvested green | Pisum sativum | 1.5.4.2 |
| Pecan nut | Carya illinoensis | 2.1.7.6 |
| Pepper, black | Piper nigrum | 2.2.3.1 |
| Pepper, dry | Capsicum spp. | 1.4.3.1 |
| Pepper, sweet | Capsicum annuum | 1.5.2.0.9 |
| Persimmon | Diospyros kaki; Diospyros virginiana | 2.1.6.1.0 |
| Pigeon pea | Cajanus cajan | 1.3.7 |
| Pimento | Capsicum annuum | 1.4.3.1 |
| Pineapple | Ananas comosus | 2.1.6.1.1 |
| Pistachio nut | Pistacia spp. | 2.1.7.6 |
| Plantain | Musa paradisiaca | 2.1.6.1.2 |
| Plum and prune | Prunus domestica | 2.1.3.4 |
| Pomegranate | Punica granatum | 2.1.6.1.3 |
| Pomelo | Citrus grandis | 2.1.1.4 |
| Poppy seed | Papaver somniferum | 1.4.2.1.1 |
| Potato | Solamum tuberosum | 1.2.1 |
| Potato, sweet | Ipomoea batatas | 1.2.2 |
| Prune and plum | Prunus domestica | 2.1.3.4 |
| Pumpkin, edible | Cucurbita spp. | 1.5.2.0.7 |
| Pumpkin, for fodder | Cucurbita spp. | 1.7.5.0.3 |
| Pyrethum | Chrysanthemum cineraraiefolium | 1.4.5.3 |
| Quebracho | Aspidosperma spp. | 2.2.5.3 |
| Queensland nut (Macadamia) | Macadamia spp. | 2.1.7.6 |
| Quince | Cydonia oblonga | 2.1.2.3 |
| Quinine | Cinchona spp. | 2.2.7.5 |
| Quinoa | Chenopodium quinoa | 1.1.9 |
| Radish | Raphanus sativus (including Cochlearia armoracia) | 1.5.3.4 |
| Ramie and rhea | Boehmeria nivea | 2.2.4.5 |
| Rapeseed (colza) | Brassica napus | 1.4.2.0.5 |
| Raspberry | Rubus spp. | 2.1.5.5 |
| Red beet | Beta vulgaris | 1.5.3.5 |
| Red cabbage | Brassica oleracea capitata | 1.5.1.0.4 |
| Redtop | Agrostis spp. | 1.7.2.1 |
| Rhea and ramie | Boehmeria nivea | 2.2.4.5 |
| Rhubarb | Rheum spp. | 1.5.1.1.3 |
| Rice | Oryza sativa; Oryza glaberrima | 1.1.2 |
| Rose | Rose spp. | 2.2.6.1 |
| Rubber | Hevea brasiliensis | 2.2.5.1 |
| Rutabaga or swede | Brassica napus | 1.5.3.6 |
| Rye | Secale cereale | 1.1.7 |
| Ryegrass seed | Lolium spp. | 1.8.4 |
| Safflower seed | Carthamus tinctorius | 1.4.2.0.9 |
| Sago palm | Metroxylon spp. | 2.2.7.3 |
| Sainfoin | Onobrychis viciifolia | 1.7.1.3 |
| Salad, corn | Valerianella locusta | 1.5.1.1.2 |
| Salsify | Tragopogon porrifolius | 1.5.3.8 |
| Sapodilla | Achras sapota | 2.1.6.1.4 |
| Satsuma | Citrus reticulata | 2.1.1.2 |
| Savoy cabbage | Brassica ileracea capitata | 1.5.1.0.4 |
| Scorzonera | Scorzonera hispanica | 1.5.3.8 |
| Sesame | Sesamum indicum | 1.4.2.0.7 |
| Shea butter | Butyrospermum paradoxum | 2.2.2.4 |
| Sisal | Agave sislana | 2.2.4.2 |
| Sorghum | Sorghum bicolor | 1.1.5.2 |
| Sorghum, broom | Sorghum bicolor | 1.4.5.4 |
| Sorghum, durra | Sorghum bicolor | 1.1.5.2 |
| Sorghum, Guinea corn | Sorghum bicolor | 1.1.5.2 |
| Sorghum, jowar | Sorghum bicolor | 1.1.5.2 |
| Sorghum, sweet | Sorghum bicolor | 1.4.1.3 |
| Sour lime | Citrus aurantifolia | 2.1.1.5 |
| Soybean | Glycine max | 1.4.2.0.2 |
| Soybean hay | Glycine max | 1.7.1.3 |
| Spelt | Triticum spelta | 1.1.9 |
| Spinach | Spinacia oleracea | 1.5.1.1.4 |
| Spring barley | Hordeum vulgare | 1.1.4.2 |
| Spring rye | Secale cereale | 1.1.7.2 |
| Spring wheat | Triticum aestivum | 1.1.1.2 |
| Spring wheat, hard | Triticum aestivum | 1.1.1.2.1 |
| Spring wheat, soft | Triticum aestivum | 1.1.1.2.2 |
| Sprouts, Brussels | Brassica oleracea gemmifera | 1.5.1.0.3 |
| Squash | Cucurbita spp. | 1.5.2.0.8 |
| Strawberry | Fragaria spp. | 2.1.5.6 |
| Sudan grass | Sorghum bicolor Sudanense | 1.7.2.1 |
| Sugar beet | Beta vulgaris | 1.4.1.2 |
| Sugar beet for fodder | Beta vulgaris | 1.7.5.0.6 |
| Sugar beet for seed | Beta vulgaris | 1.8.1 |
| Sugarcane for fodder | Saccharum officinarum | 1.7.5.0.7 |
| Sugarcane for sugar | Saccharum officinarum | 1.4.1.1 |
| Sugarcane for thatching | Saccharum officinarum | 1.4.5.4 |
| Sunflower for fodder | Helianthus annuus | 1.7.5.0.4 |
| Sunflower for oil seed | Helianthus annuus | 1.4.2.0.4 |
| Sunhemp | Crotalaria juncea | 1.4.4.5 |
| Swede (rutabaga) | Brassica napus | 1.5.3.6 |
| Swede for fodder | Brassica napus | 1.7.5.0.8 |
| Sweet corn | Zea mays | 1.5.5.2 |
| Sweet lime | Citrus limetta | 2.1.1.6 |
| Sweet pepper | Capsicum annuum | 1.5.2.0.9 |
| Sweet potato | Iopmoea batatas | 1.2.2 |
| Sweet sorghum | Sorghum bicolor | 1.4.1.3 |
| Tangerine and mandarin | Citrus reticulata | 2.1.1.2 |
| Tannia | Xanthosoma sagittifolium | 1.2.8 |
| Tapioca (cassava) | Manihot esculenta | 1.2.3 |
| Taro | Colocasia esculenta | 1.2.8 |
| Tea | Camellia sinensis | 2.2.1.3 |
| Teff | Eragrostis abyssinica | 1.1.9 |
| Timothy | Phleum pratense | 1.7.2.1 |
| Tobacco | Nicotiana tabacum | 1.4.5.1 |
| Tomato | Lycopersicum esculentum | 1.5.2.1.0 |
| Trefoil | Lotus spp. | 1.8.6 |
| Tung tree | Aleurites spp. | 2.2.2.4 |
| Turnip, edible | Brassica rapa | 1.5.3.7 |
| Turnip, for fodder | Brassica rapa | 1.7.5.0.9 |
| Urena (Congo jute) | Urena lobata | 1.4.4.5 |
| Vanilla | Vanilla planifolia | 2.2.3.6 |
| Vetch for grain | Vicia sativa | 1.3.8 |
| Walnut | Juglans spp. | 2.1.7.3 |
| Watermelon | Citrullus lanatus | 1.5.2.1.1 |
| Wattle, black | Acacia mearnsii | 2.2.5.2 |
| Wheat | Triticum spp. | 1.1.1 |
| White cabbage | Brassica oleracea capitata | 1.5.1.0.4 |
| Winter barley | Hordeum spp. | 1.1.4.1 |
| Winter rye | Secale cereale | 1.1.7.1 |
| Winter wheat | Triticum aestivum | 1.1.1.1 |
| Winter wheat, hard | Triticum aestivum | 1.1.1.1.1 |
| Winter wheat, soft | Triticum aestivum | 1.1.1.1.2 |
| Yams | Dioscorea spp. | 1.2.4 |
| Yerba mate | Ilex paraguariensis | 2.2.1.5 |