

---

# **SANSA Online Catalogue Documentation**

***Release 2.0***

**Tim Sutton**

March 25, 2011



# CONTENTS

<b>1</b>	<b>Catalogue Models Documentation</b>	<b>3</b>
1.1	Modules: . . . . .	3
<b>2</b>	<b>Catalogue Views Documentation</b>	<b>13</b>
2.1	Modules: . . . . .	13
<b>3</b>	<b>Indices and tables</b>	<b>19</b>
	<b>Python Module Index</b>	<b>21</b>
	<b>Index</b>	<b>23</b>



Contents:



# CATALOGUE MODELS DOCUMENTATION

This page contains the Catalogue Models documentation.

## 1.1 Modules:

### 1.1.1 Dictionaries Documentation

This page contains the Dictionaries Module documentation.

#### The dictionaries Module

**class** dictionaries.**AcquisitionMode** (\*args, \*\*kwargs)

Bases: django.db.models.base.Model

Acquisition mode. @note: mode examples:

J = Multispectral 10m P/M = Panchromatic 10m A/B = Panchromatic 5m T = Panchromatic 2.5m X  
= Multispectral 20m JT = Pansharpened 2.5m Multispectral

**exception DoesNotExist**

Bases: django.core.exceptions.ObjectDoesNotExist

**exception** AcquisitionMode.**MultipleObjectsReturned**

Bases: django.core.exceptions.MultipleObjectsReturned

AcquisitionMode.**sensor\_type**

**class** dictionaries.**CreatingSoftware** (\*args, \*\*kwargs)

Bases: django.db.models.base.Model

CreatingSoftware(id, name, version)

**exception DoesNotExist**

Bases: django.core.exceptions.ObjectDoesNotExist

**exception** CreatingSoftware.**MultipleObjectsReturned**

Bases: django.core.exceptions.MultipleObjectsReturned

**class** dictionaries.**Institution** (\*args, \*\*kwargs)

Bases: django.db.models.base.Model

```

Institution(id, name, address1, address2, address3, post_code)

exception DoesNotExist
    Bases: django.core.exceptions.ObjectDoesNotExist

exception Institution.MultipleObjectsReturned
    Bases: django.core.exceptions.MultipleObjectsReturned

class dictionaries.License (*args, **kwargs)
    Bases: django.db.models.base.Model

    Licenses for Products

    exception DoesNotExist
        Bases: django.core.exceptions.ObjectDoesNotExist

    exception License.MultipleObjectsReturned
        Bases: django.core.exceptions.MultipleObjectsReturned

    License.get_type_display (*moreargs, **morekwargs)

class dictionaries.Mission (*args, **kwargs)
    Bases: django.db.models.base.Model

    Satellite or Mission

    exception DoesNotExist
        Bases: django.core.exceptions.ObjectDoesNotExist

    exception Mission.MultipleObjectsReturned
        Bases: django.core.exceptions.MultipleObjectsReturned

    Mission.mission_group

    Mission.missionsensor_set

class dictionaries.MissionGroup (*args, **kwargs)
    Bases: django.db.models.base.Model

    Mission Group

    exception DoesNotExist
        Bases: django.core.exceptions.ObjectDoesNotExist

    exception MissionGroup.MultipleObjectsReturned
        Bases: django.core.exceptions.MultipleObjectsReturned

    MissionGroup.mission_set

class dictionaries.MissionSensor (*args, **kwargs)
    Bases: django.db.models.base.Model

    MissionSensor on the mission or satellite - e.g. HRV ABP: TODO: get rid of has_data (or auto-update with soft
    trigger ?)

    exception DoesNotExist
        Bases: django.core.exceptions.ObjectDoesNotExist

    exception MissionSensor.MultipleObjectsReturned
        Bases: django.core.exceptions.MultipleObjectsReturned

    MissionSensor.mission

    MissionSensor.sensortype_set

```



```
class dictionaries.Place (*args, **kwargs)
    Bases: django.db.models.base.Model

    A collection on named places based largely on geonames (which all get a place type of Nearest named place)

    exception DoesNotExist
        Bases: django.core.exceptions.ObjectDoesNotExist

    exception Place.MultipleObjectsReturned
        Bases: django.core.exceptions.MultipleObjectsReturned

    Place.geometry

    Place.place_type

class dictionaries.PlaceType (*args, **kwargs)
    Bases: django.db.models.base.Model

    A dictionary to define place types e.g. Global, Continent, Region, Country, Province, City etc.

    exception DoesNotExist
        Bases: django.core.exceptions.ObjectDoesNotExist

    exception PlaceType.MultipleObjectsReturned
        Bases: django.core.exceptions.MultipleObjectsReturned

    PlaceType.place_set

class dictionaries.ProcessingLevel (*args, **kwargs)
    Bases: django.db.models.base.Model

    Processing level e.g. L3Aa

    class Admin

    exception ProcessingLevel.DoesNotExist
        Bases: django.core.exceptions.ObjectDoesNotExist

    exception ProcessingLevel.MultipleObjectsReturned
        Bases: django.core.exceptions.MultipleObjectsReturned

class dictionaries.Projection (*args, **kwargs)
    Bases: django.db.models.base.Model

    Projection(id, epsg_code, name)

    class Admin

    exception Projection.DoesNotExist
        Bases: django.core.exceptions.ObjectDoesNotExist

    exception Projection.MultipleObjectsReturned
        Bases: django.core.exceptions.MultipleObjectsReturned

class dictionaries.Quality (*args, **kwargs)
    Bases: django.db.models.base.Model

    Quality(id, name)

    exception DoesNotExist
        Bases: django.core.exceptions.ObjectDoesNotExist

    exception Quality.MultipleObjectsReturned
        Bases: django.core.exceptions.MultipleObjectsReturned
```

```

class dictionaries.SensorType(*args, **kwargs)
    Bases: django.db.models.base.Model

    Sensor type / camera number e.g. CAM1

    exception DoesNotExist
        Bases: django.core.exceptions.ObjectDoesNotExist

    exception SensorType.MultipleObjectsReturned
        Bases: django.core.exceptions.MultipleObjectsReturned

    SensorType.acquisitionmode_set

    SensorType.mission_sensor

class dictionaries.Topic(*args, **kwargs)
    Bases: django.db.models.base.Model

    A dictionary to define geospatial dataset topics e.g. LANDUSE, ROADS etc.

    exception DoesNotExist
        Bases: django.core.exceptions.ObjectDoesNotExist

    exception Topic.MultipleObjectsReturned
        Bases: django.core.exceptions.MultipleObjectsReturned

class dictionaries.Unit(*args, **kwargs)
    Bases: django.db.models.base.Model

    A dictionary to define unit types e.g. m, km etc.

    exception DoesNotExist
        Bases: django.core.exceptions.ObjectDoesNotExist

    exception Unit.MultipleObjectsReturned
        Bases: django.core.exceptions.MultipleObjectsReturned

```

## 1.1.2 Legacy Documentation

This page contains the Legacy Module documentation.

### The Legacy Module

```

class legacy.Cbers(*args, **kwargs)
    Bases: django.db.models.base.Model

    Cbers(gid, id, source, sceneid, k, j, adate, ayear, amonth, aday, ahour, amin, asec, utmzo, utmalt, sazia, seleva,
    the_geom)

    exception DoesNotExist
        Bases: django.core.exceptions.ObjectDoesNotExist

    exception Cbers.MultipleObjectsReturned
        Bases: django.core.exceptions.MultipleObjectsReturned

    Cbers.the_geom

class legacy.Sacc(*args, **kwargs)
    Bases: django.db.models.base.Model

    Sacc(gid, id, source, sceneid, wrsp, wrsr, adate, utmzo, utmalt, the_geom)

```

**exception DoesNotExist**

Bases: django.core.exceptions.ObjectDoesNotExist

**exception Sacc.MultipleObjectsReturned**

Bases: django.core.exceptions.MultipleObjectsReturned

Sacc.the\_geom

**class** legacy.Spot5(\*args, \*\*kwargs)

Bases: django.db.models.base.Model

Spot5(gid, a2l, sc\_num, seg\_num, satel, ang\_inc, ang\_acq, date\_acq, month\_acq, time\_acq, cloud\_quot, cloud\_per, snow\_quot, lat\_cen, lon\_cen, lat\_up\_l, lon\_up\_l, lat\_up\_r, lon\_up\_r, lat\_lo\_l, lon\_lo\_l, lat\_lo\_r, lon\_lo\_r, resol, mode, type, url\_ql, the\_geom)

**exception DoesNotExist**

Bases: django.core.exceptions.ObjectDoesNotExist

**exception Spot5.MultipleObjectsReturned**

Bases: django.core.exceptions.MultipleObjectsReturned

Spot5.the\_geom

**class** legacy.Sumb(\*args, \*\*kwargs)

Bases: django.db.models.base.Model

Sumb(gid, id, source, sceneid, k, j, adate, utmzo, utmalt, the\_geom)

**exception DoesNotExist**

Bases: django.core.exceptions.ObjectDoesNotExist

**exception Sumb.MultipleObjectsReturned**

Bases: django.core.exceptions.MultipleObjectsReturned

Sumb.the\_geom

### 1.1.3 Orders Documentation

This page contains the Orders Module documentation.

#### The orders Module

**class** orders.Datum(\*args, \*\*kwargs)

Bases: django.db.models.base.Model

Datum(id, name)

**class Admin**

**exception Datum.DoesNotExist**

Bases: django.core.exceptions.ObjectDoesNotExist

**exception Datum.MultipleObjectsReturned**

Bases: django.core.exceptions.MultipleObjectsReturned

Datum.deliverydetail\_set

**class** orders.DeliveryDetail(\*args, \*\*kwargs)

Bases: django.db.models.base.Model

DeliveryDetail(id, user\_id, processing\_level\_id, projection\_id, datum\_id, resampling\_method\_id, file\_format\_id, geometry)

```
exception DoesNotExist
    Bases: django.core.exceptions.ObjectDoesNotExist

exception DeliveryDetail.MultipleObjectsReturned
    Bases: django.core.exceptions.MultipleObjectsReturned

DeliveryDetail.datum
DeliveryDetail.file_format
DeliveryDetail.geometry
DeliveryDetail.order_set
DeliveryDetail.processing_level
DeliveryDetail.projection
DeliveryDetail.resampling_method
DeliveryDetail.user

class orders.DeliveryMethod (*args, **kwargs)
    Bases: django.db.models.base.Model

    DeliveryMethod(id, name)

class Admin

exception DeliveryMethod.DoesNotExist
    Bases: django.core.exceptions.ObjectDoesNotExist

exception DeliveryMethod.MultipleObjectsReturned
    Bases: django.core.exceptions.MultipleObjectsReturned

DeliveryMethod.order_set

class orders.FileFormat (*args, **kwargs)
    Bases: django.db.models.base.Model

    FileFormat(id, name)

class Admin

exception FileFormat.DoesNotExist
    Bases: django.core.exceptions.ObjectDoesNotExist

exception FileFormat.MultipleObjectsReturned
    Bases: django.core.exceptions.MultipleObjectsReturned

FileFormat.deliverydetail_set

class orders.Order (*args, **kwargs)
    Bases: django.db.models.base.Model

    Order(id, user_id, notes, order_status_id, delivery_method_id, delivery_detail_id, order_date)

class Admin

exception Order.DoesNotExist
    Bases: django.core.exceptions.ObjectDoesNotExist

exception Order.MultipleObjectsReturned
    Bases: django.core.exceptions.MultipleObjectsReturned

Order.delivery_detail
Order.delivery_method
```

```

Order.get_next_by_order_date(*moreargs, **morekwargs)
Order.get_previous_by_order_date(*moreargs, **morekwargs)
Order.order_status
Order.orderstatushistory_set
Order.taskingrequest
Order.user

class orders.OrderStatus(*args, **kwargs)
    Bases: django.db.models.base.Model
    OrderStatus(id, name)

    class Admin

    exception OrderStatus.DoesNotExist
        Bases: django.core.exceptions.ObjectDoesNotExist

    exception OrderStatus.MultipleObjectsReturned
        Bases: django.core.exceptions.MultipleObjectsReturned

    OrderStatus.new_order_status
    OrderStatus.old_order_status
    OrderStatus.order_set

class orders.OrderStatusHistory(*args, **kwargs)
    Bases: django.db.models.base.Model
    Used to maintain provenance of all status changes that happen to an order

    class Admin

    exception OrderStatusHistory.DoesNotExist
        Bases: django.core.exceptions.ObjectDoesNotExist

    exception OrderStatusHistory.MultipleObjectsReturned
        Bases: django.core.exceptions.MultipleObjectsReturned

    OrderStatusHistory.get_next_by_order_change_date(*moreargs, **morekwargs)
    OrderStatusHistory.get_previous_by_order_change_date(*moreargs, **morekwargs)

    OrderStatusHistory.new_order_status
    OrderStatusHistory.old_order_status
    OrderStatusHistory.order
    OrderStatusHistory.user

class orders.ResamplingMethod(*args, **kwargs)
    Bases: django.db.models.base.Model
    ResamplingMethod(id, name)

    class Admin

    exception ResamplingMethod.DoesNotExist
        Bases: django.core.exceptions.ObjectDoesNotExist

    exception ResamplingMethod.MultipleObjectsReturned
        Bases: django.core.exceptions.MultipleObjectsReturned

```

```
ResamplingMethod.deliverydetail_set
```

```
class orders.TaskingRequest(*args, **kwargs)
    Bases: orders.Order
```

A tasking request inherits from the order model and adds three fields: geometry, target date and sensor. The tasking request is used by end users to queue up acquisition requests for a given sensor.

```
class Admin
```

```
exception TaskingRequest.DoesNotExist
    Bases: orders.DoesNotExist
```

```
exception TaskingRequest.MultipleObjectsReturned
    Bases: orders.MultipleObjectsReturned
```

```
TaskingRequest.geometry
```

```
TaskingRequest.get_next_by_target_date(*moreargs, **morekwargs)
```

```
TaskingRequest.get_previous_by_target_date(*moreargs, **morekwargs)
```

```
TaskingRequest.mission_sensor
```

```
TaskingRequest.order_ptr
```

## 1.1.4 Others Documentation

This page contains the Others Module documentation.

### The others Module

## 1.1.5 Products Documentation

This page contains the Products Module documentation.

### The products Module

```
products.runconcrete(func)
```

This decorator calls the method in the concrete subclass and raise an exception if the method is found only in the base GenericProduct abstract class

## 1.1.6 Search Documentation

This page contains the Search Module documentation.

### The search Module

## 1.1.7 Signals Documentation

This page contains the Signals Module documentation.

## The `signals` Module

`signals.setGenericProductDate` (*sender, instance, \*\*kw*)

Sets the `product_date` based on acquisition date, if both `start_date` and `end_date` are set, then calculate the avg, uses the `start_date` if `end_date` is not set

`signals.setGeometricResolution` (*sender, instance, \*\*kw*)

Sets default `geometric_resolution_x` and `geometric_resolution_y` from `AcquisitionMode.geometric_resolution`

Sets `geometric_resolution` as the average value from `geometric_resolution_x` and `geometric_resolution_y`





# CATALOGUE VIEWS DOCUMENTATION

This page contains the Catalogue Views documentation.

## 2.1 Modules:

### 2.1.1 Geoiputils Documentation

This page contains the Geoiputils Module documentation.

#### The `geoiputils` Module

**class** `geoiputils.GeoIpUtils`

A class for resolving lat/long from and IP address

**getMyIp** (*request*)

Fetch an IP when request.META returns localhost

**getMyLatLong** (*request*)

### 2.1.2 Helpers Documentation

This page contains the Helpers Module documentation.

#### The `helpers` Module

### 2.1.3 Orders Documentation

This page contains the Orders Module documentation.

#### The `orders` Module

**class** `orders.Datum` (*\*args, \*\*kwargs*)

Bases: `django.db.models.base.Model`

`Datum(id, name)`

**class** `Admin`

```
exception Datum.DoesNotExist
    Bases: django.core.exceptions.ObjectDoesNotExist

exception Datum.MultipleObjectsReturned
    Bases: django.core.exceptions.MultipleObjectsReturned

Datum.deliverydetail_set

class orders.DeliveryDetail (*args, **kwargs)
    Bases: django.db.models.base.Model

    DeliveryDetail(id, user_id, processing_level_id, projection_id, datum_id, resampling_method_id,
    file_format_id, geometry)

exception DoesNotExist
    Bases: django.core.exceptions.ObjectDoesNotExist

exception DeliveryDetail.MultipleObjectsReturned
    Bases: django.core.exceptions.MultipleObjectsReturned

DeliveryDetail.datum
DeliveryDetail.file_format
DeliveryDetail.geometry
DeliveryDetail.order_set
DeliveryDetail.processing_level
DeliveryDetail.projection
DeliveryDetail.resampling_method
DeliveryDetail.searchrecord_set
DeliveryDetail.user

class orders.DeliveryMethod (*args, **kwargs)
    Bases: django.db.models.base.Model

    DeliveryMethod(id, name)

class Admin

exception DeliveryMethod.DoesNotExist
    Bases: django.core.exceptions.ObjectDoesNotExist

exception DeliveryMethod.MultipleObjectsReturned
    Bases: django.core.exceptions.MultipleObjectsReturned

DeliveryMethod.order_set

class orders.FileFormat (*args, **kwargs)
    Bases: django.db.models.base.Model

    FileFormat(id, name)

class Admin

exception FileFormat.DoesNotExist
    Bases: django.core.exceptions.ObjectDoesNotExist

exception FileFormat.MultipleObjectsReturned
    Bases: django.core.exceptions.MultipleObjectsReturned

FileFormat.deliverydetail_set
```

```
class orders.Order (*args, **kwargs)
    Bases: django.db.models.base.Model

    Order(id, user_id, notes, order_status_id, delivery_method_id, delivery_detail_id, order_date)

    class Admin

    exception Order.DoesNotExist
        Bases: django.core.exceptions.ObjectDoesNotExist

    exception Order.MultipleObjectsReturned
        Bases: django.core.exceptions.MultipleObjectsReturned

    Order.delivery_detail
    Order.delivery_method
    Order.get_next_by_order_date (*moreargs, **morekwargs)
    Order.get_previous_by_order_date (*moreargs, **morekwargs)
    Order.order_status
    Order.orderstatushistory_set
    Order.searchrecord_set
    Order.taskingrequest
    Order.user

class orders.OrderStatus (*args, **kwargs)
    Bases: django.db.models.base.Model

    OrderStatus(id, name)

    class Admin

    exception OrderStatus.DoesNotExist
        Bases: django.core.exceptions.ObjectDoesNotExist

    exception OrderStatus.MultipleObjectsReturned
        Bases: django.core.exceptions.MultipleObjectsReturned

    OrderStatus.new_order_status
    OrderStatus.old_order_status
    OrderStatus.order_set

class orders.OrderStatusHistory (*args, **kwargs)
    Bases: django.db.models.base.Model

    Used to maintain provenance of all status changes that happen to an order

    class Admin

    exception OrderStatusHistory.DoesNotExist
        Bases: django.core.exceptions.ObjectDoesNotExist

    exception OrderStatusHistory.MultipleObjectsReturned
        Bases: django.core.exceptions.MultipleObjectsReturned

    OrderStatusHistory.get_next_by_order_change_date (*moreargs, **morekwargs)
    OrderStatusHistory.get_previous_by_order_change_date (*moreargs, **morekwargs)

    OrderStatusHistory.new_order_status
```

```
OrderStatusHistory.old_order_status
OrderStatusHistory.order
OrderStatusHistory.user
class orders.ResamplingMethod (*args, **kwargs)
    Bases: django.db.models.base.Model
    ResamplingMethod(id, name)
    class Admin
    exception ResamplingMethod.DoesNotExist
        Bases: django.core.exceptions.ObjectDoesNotExist
    exception ResamplingMethod.MultipleObjectsReturned
        Bases: django.core.exceptions.MultipleObjectsReturned
    ResamplingMethod.deliverydetail_set
class orders.TaskingRequest (*args, **kwargs)
    Bases: orders.Order

A tasking request inherits from the order model and adds three fields: geometry, target date and sensor. The
tasking request is used by end users to queue up acquisition requests for a given sensor.

    class Admin
    exception TaskingRequest.DoesNotExist
        Bases: orders.DoesNotExist
    exception TaskingRequest.MultipleObjectsReturned
        Bases: orders.MultipleObjectsReturned
    TaskingRequest.geometry
    TaskingRequest.get_next_by_target_date (*moreargs, **morekwargs)
    TaskingRequest.get_previous_by_target_date (*moreargs, **morekwargs)
    TaskingRequest.mission_sensor
    TaskingRequest.order_ptr
```

## 2.1.4 Others Documentation

This page contains the Others Module documentation.

### The others Module

## 2.1.5 Search Documentation

This page contains the Search Module documentation.

### The search Module

## 2.1.6 Searcher Documentation

This page contains the Searcher Module documentation.

## The searcher Module

**class** `searcher.Searcher` (*theRequest, theGuid*)

Class members - variables declared here act like static class members in C++ i.e. if you change them from an object, all objects will receive that change.

**clearTemplateData** ()

Clear the searcher data

**logResults** ()

**meanCloudString** ()

**rowPathAsString** ()

**search** (*thePaginateFlag=True*)

Show a map of a single search for scenes

**templateData** ()

Return data from the searcher suitable for passing along to the map.html template

## 2.1.7 Shopping\_Cart Documentation

This page contains the Shopping\_Cart Module documentation.

### The shopping\_cart Module

## 2.1.8 Tasking Documentation

This page contains the Tasking Module documentation.

### The tasking Module



# INDICES AND TABLES

- *genindex*
- *modindex*
- *search*





# PYTHON MODULE INDEX

## d

dictionaries, [3](#)

## g

geoiputils, [13](#)

## l

legacy, [6](#)

## o

orders, [13](#)

others, [16](#)

## p

products, [10](#)

## s

search, [16](#)

searcher, [17](#)

signals, [11](#)



# INDEX

## A

AcquisitionMode (class in dictionaries), 3  
AcquisitionMode.DoesNotExist, 3  
AcquisitionMode.MultipleObjectsReturned, 3  
acquisitionmode\_set (dictionaries.SensorType attribute), 6

## C

Cbers (class in legacy), 6  
Cbers.DoesNotExist, 6  
Cbers.MultipleObjectsReturned, 6  
clearTemplateData() (searcher.Searcher method), 17  
CreatingSoftware (class in dictionaries), 3  
CreatingSoftware.DoesNotExist, 3  
CreatingSoftware.MultipleObjectsReturned, 3

## D

Datum (class in orders), 7, 13  
datum (orders.DeliveryDetail attribute), 8, 14  
Datum.Admin (class in orders), 7, 13  
Datum.DoesNotExist, 7, 13  
Datum.MultipleObjectsReturned, 7, 14  
delivery\_detail (orders.Order attribute), 8, 15  
delivery\_method (orders.Order attribute), 8, 15  
DeliveryDetail (class in orders), 7, 14  
DeliveryDetail.DoesNotExist, 7, 14  
DeliveryDetail.MultipleObjectsReturned, 8, 14  
deliverydetail\_set (orders.Datum attribute), 7, 14  
deliverydetail\_set (orders.FileFormat attribute), 8, 14  
deliverydetail\_set (orders.ResamplingMethod attribute), 10, 16  
DeliveryMethod (class in orders), 8, 14  
DeliveryMethod.Admin (class in orders), 8, 14  
DeliveryMethod.DoesNotExist, 8, 14  
DeliveryMethod.MultipleObjectsReturned, 8, 14  
dictionaries (module), 3

## F

file\_format (orders.DeliveryDetail attribute), 8, 14  
FileFormat (class in orders), 8, 14  
FileFormat.Admin (class in orders), 8, 14

FileFormat.DoesNotExist, 8, 14  
FileFormat.MultipleObjectsReturned, 8, 14

## G

GeoIpUtils (class in geoiputils), 13  
geoiputils (module), 13  
geometry (dictionaries.Place attribute), 5  
geometry (orders.DeliveryDetail attribute), 8, 14  
geometry (orders.TaskingRequest attribute), 10, 16  
get\_next\_by\_order\_change\_date() (orders.OrderStatusHistory method), 9, 15  
get\_next\_by\_order\_date() (orders.Order method), 8, 15  
get\_next\_by\_target\_date() (orders.TaskingRequest method), 10, 16  
get\_previous\_by\_order\_change\_date() (orders.OrderStatusHistory method), 9, 15  
get\_previous\_by\_order\_date() (orders.Order method), 9, 15  
get\_previous\_by\_target\_date() (orders.TaskingRequest method), 10, 16  
get\_type\_display() (dictionaries.License method), 4  
getMyIp() (geoiputils.GeoIpUtils method), 13  
getMyLatLong() (geoiputils.GeoIpUtils method), 13

## I

Institution (class in dictionaries), 3  
Institution.DoesNotExist, 4  
Institution.MultipleObjectsReturned, 4

## L

legacy (module), 6  
License (class in dictionaries), 4  
License.DoesNotExist, 4  
License.MultipleObjectsReturned, 4  
logResults() (searcher.Searcher method), 17

## M

meanCloudString() (searcher.Searcher method), 17  
Mission (class in dictionaries), 4  
mission (dictionaries.MissionSensor attribute), 4  
Mission.DoesNotExist, 4

Mission.MultipleObjectsReturned, 4  
 mission\_group (dictionaries.Mission attribute), 4  
 mission\_sensor (dictionaries.SensorType attribute), 6  
 mission\_sensor (orders.TaskingRequest attribute), 10, 16  
 mission\_set (dictionaries.MissionGroup attribute), 4  
 MissionGroup (class in dictionaries), 4  
 MissionGroup.DoesNotExist, 4  
 MissionGroup.MultipleObjectsReturned, 4  
 MissionSensor (class in dictionaries), 4  
 MissionSensor.DoesNotExist, 4  
 MissionSensor.MultipleObjectsReturned, 4  
 missionsensor\_set (dictionaries.Mission attribute), 4

## N

new\_order\_status (orders.OrderStatus attribute), 9, 15  
 new\_order\_status (orders.OrderStatusHistory attribute), 9, 15

## O

old\_order\_status (orders.OrderStatus attribute), 9, 15  
 old\_order\_status (orders.OrderStatusHistory attribute), 9, 16  
 Order (class in orders), 8, 14  
 order (orders.OrderStatusHistory attribute), 9, 16  
 Order.Admin (class in orders), 8, 15  
 Order.DoesNotExist, 8, 15  
 Order.MultipleObjectsReturned, 8, 15  
 order\_ptr (orders.TaskingRequest attribute), 10, 16  
 order\_set (orders.DeliveryDetail attribute), 8, 14  
 order\_set (orders.DeliveryMethod attribute), 8, 14  
 order\_set (orders.OrderStatus attribute), 9, 15  
 order\_status (orders.Order attribute), 9, 15  
 orders (module), 7, 13  
 OrderStatus (class in orders), 9, 15  
 OrderStatus.Admin (class in orders), 9, 15  
 OrderStatus.DoesNotExist, 9, 15  
 OrderStatus.MultipleObjectsReturned, 9, 15  
 OrderStatusHistory (class in orders), 9, 15  
 OrderStatusHistory.Admin (class in orders), 9, 15  
 OrderStatusHistory.DoesNotExist, 9, 15  
 OrderStatusHistory.MultipleObjectsReturned, 9, 15  
 orderstatushistory\_set (orders.Order attribute), 9, 15  
 others (module), 10, 16

## P

Place (class in dictionaries), 4  
 Place.DoesNotExist, 5  
 Place.MultipleObjectsReturned, 5  
 place\_set (dictionaries.PlaceType attribute), 5  
 place\_type (dictionaries.Place attribute), 5  
 PlaceType (class in dictionaries), 5  
 PlaceType.DoesNotExist, 5  
 PlaceType.MultipleObjectsReturned, 5  
 processing\_level (orders.DeliveryDetail attribute), 8, 14

ProcessingLevel (class in dictionaries), 5  
 ProcessingLevel.Admin (class in dictionaries), 5  
 ProcessingLevel.DoesNotExist, 5  
 ProcessingLevel.MultipleObjectsReturned, 5  
 products (module), 10  
 Projection (class in dictionaries), 5  
 projection (orders.DeliveryDetail attribute), 8, 14  
 Projection.Admin (class in dictionaries), 5  
 Projection.DoesNotExist, 5  
 Projection.MultipleObjectsReturned, 5

## Q

Quality (class in dictionaries), 5  
 Quality.DoesNotExist, 5  
 Quality.MultipleObjectsReturned, 5

## R

resampling\_method (orders.DeliveryDetail attribute), 8, 14  
 ResamplingMethod (class in orders), 9, 16  
 ResamplingMethod.Admin (class in orders), 9, 16  
 ResamplingMethod.DoesNotExist, 9, 16  
 ResamplingMethod.MultipleObjectsReturned, 9, 16  
 rowPathAsString() (searcher.Searcher method), 17  
 runconcrete() (in module products), 10

## S

Sacc (class in legacy), 6  
 Sacc.DoesNotExist, 6  
 Sacc.MultipleObjectsReturned, 7  
 search (module), 10, 16  
 search() (searcher.Searcher method), 17  
 Searcher (class in searcher), 17  
 searcher (module), 17  
 searchrecord\_set (orders.DeliveryDetail attribute), 14  
 searchrecord\_set (orders.Order attribute), 15  
 sensor\_type (dictionaries.AcquisitionMode attribute), 3  
 SensorType (class in dictionaries), 5  
 SensorType.DoesNotExist, 6  
 SensorType.MultipleObjectsReturned, 6  
 sensortype\_set (dictionaries.MissionSensor attribute), 4  
 setGenericProductDate() (in module signals), 11  
 setGeometricResolution() (in module signals), 11  
 signals (module), 11  
 Spot5 (class in legacy), 7  
 Spot5.DoesNotExist, 7  
 Spot5.MultipleObjectsReturned, 7  
 Sumb (class in legacy), 7  
 Sumb.DoesNotExist, 7  
 Sumb.MultipleObjectsReturned, 7

## T

TaskingRequest (class in orders), 10, 16

taskingrequest (orders.Order attribute), 9, 15  
TaskingRequest.Admin (class in orders), 10, 16  
TaskingRequest.DoesNotExist, 10, 16  
TaskingRequest.MultipleObjectsReturned, 10, 16  
templateData() (searcher.Searcher method), 17  
the\_geom (legacy.Cbers attribute), 6  
the\_geom (legacy.Sacc attribute), 7  
the\_geom (legacy.Spot5 attribute), 7  
the\_geom (legacy.Sumb attribute), 7  
Topic (class in dictionaries), 6  
Topic.DoesNotExist, 6  
Topic.MultipleObjectsReturned, 6

## U

Unit (class in dictionaries), 6  
Unit.DoesNotExist, 6  
Unit.MultipleObjectsReturned, 6  
user (orders.DeliveryDetail attribute), 8, 14  
user (orders.Order attribute), 9, 15  
user (orders.OrderStatusHistory attribute), 9, 16