nrewebservices Documentation

Release 0.1.5-dev

George Goldberg

May 14, 2018

Contents

Contents:

Introduction

1.1 Overview

NREWebService is a Python 2/3 compatible library to ease working with the various Web Service APIs offered by National Rail Enquiries for access to rail industry data. The web services this library provides access to are:

- The Live Departure Boards Web Service (*ldbws*)
- The Live Departure Boards Web Service - Staff Version (*staff*)
- The Historical Service Performance Web Service (*hsp*)

1.2 Accessing the APIs

These APIs are provided by National Rail Enquiries under varying license terms. In order to access them you will need the relevant API keys (which are different for each web service). For more information about the available APIs and how to get access to them, please see the National Rail Enquiries Developers Page.

1.3 Getting Help

If you run into difficulties using this library or the National Rail Enquiries APIs, you can get help from the Open Rail Data Google Talk Group. If you want to learn more about the National Rail Enquiries data feeds, as well as other UK rail open data feeds, you should visit the Open Rail Data Wiki.

This is not an official API client library, and is not affiliated with National Rail Enquiries or the Association of Train Operating Companies in any way. Please do not contact them for help with this library.

1.4 Bugs and Feature Requests

If you find a bug in this library, or would like to request a new feature, please use the GitHub Issue Tracker.

Live Departure Boards

2.1 Overview

The ldbws module provides an API for querying the Darwin OpenLDBWS web service. This web service allows for querying all the basic information provided via station departure boards, and is suitable for building applications which replicate some or all of this functionality. For more involved projects, especially those where you wish to combine web service data with data from the Darwin Push Port, the *LDBWS Staff Version* web service may be more appropriate due to the extra fields it provides on response objects.

2.2 Official Documentation

The official documentation can be found here. Whilst a useful resource for the more experienced developer, experience has shown this page to be somewhat lacking, and even entirely incorrect at times. This library provides detailed documentation of the individual API calls and the response object properties, derived from the official documentation, but augmented with additional explanations and corrections where necessary.

This library also provides a number of convenience functions which perform commonly needed transformations on the raw data returned by the web service to make life easier for developers.

2.3 Getting an Access Token

In order to access the web service, you will need to sign up for an access token from National Rail Enquiries here. To do this you are required to sign the NRE Open Data license. You should note in particular the usage caps and charges for high usage. As a convenience, methods which generatecalls to the web service that count towards usage caps or charges are clearly indicated in this calls to the web service that count towards usage caps or charges are clearly indicated in this documentation. However the accuracy of this information cannot be guaranteed, so if you are concerned about usage caps or charges you should make your own investigations to check.

2.4 Web Service Sessions

Interaction with the LDBWS API takes place through an instance of the `Session` object. On instantiating the Session, the WSDL for the web service is retrieved and parsed. This requires a series of files to be retrieved from the LDBWS server, and so will take some time to return (typically this takes around 1-2 seconds from an internet connected server). The WSDL is then parsed and cached for the life of the Session object ensuring subsequent API requests can take place immediately.

Once the `Session` object has been instantiated, API requests are made by means of calling the available methods on this object.

2.5 Times

Many of the objects returned by the LDBWS API contain attributes which represent times. However, in the raw API these are treated as strings, because there are some values they can contain which are not valid times. In general, times are provided in the format hh:mm. However, sometimes they may be followed by an asterisk indicating that the Darwin system is unsure as to their reliability. They may also be replaced in some cases by strings which can include the following "Delayed", "On Time", "Cancelled" and "No report". The values of these fields are sutiable to be displayed directly as provided on departure boards (in fact, that is what the on-platform ones at stations do). However, if you are planning to parse them as times, you should take these alternative values into consideration.

2.6 Usage Example

See the file *examples/ldbws.py* in this repository for a simple usage example.

2.7 API Reference

For details of all the methods available on the *Session* object, as well as the response objects and their properties, please see the *LDBWS API Reference*.

LDBWS Staff Version

Warning: The LDBWS Staff Version API is not supported in this version of NREWebServices. Support for it will be added in a future version.

Historical Service Performance

> **Warning:** The Historical Service Performance API is not supported in this version of NREWebServices. Support for it will be added in a future version.

Public API Reference

5.1 nrewebservices.ldbws

5.1.1 Session

class nrewebservices.ldbws.**Session**(*wsdl=None*, *api_key=None*, *timeout=5*)
This class provides the interface to the LDBWS web service session.

Note: There are some (unknown) internal rules on the LDBWS server which limit the number of services returned in a response, sometimes to less than the number requested by the *time_window* and/or *rows* parameters to a request. Unfortunately there is nothing that can be done about this, so you just have to work with it.

__init__(*wsdl=None*, *api_key=None*, *timeout=5*)
You should normally instantiate this class only once per application, as it fetches and parses the WSDL from the server on instantiation, normally taking a few seconds to complete.

Parameters

- **wsdl** (*str*) – the URL of the web service WSDL. Be sure to pass the ?ver=2016-02-16 on the end of the URL to get the version this library currently supports. If this parameter is not provided, the code expects to find an environment variable called **NRE_LDBWS_WSDL** containing it instead.

- **api_key** (*str*) – your LDBWS API key. If this is not provided, the code expects to find an environment variable called **NRE_LDBWS_API_KEY** containing it instead.

- **timeout** (*int*) – the number of seconds the after which the underlying SOAP client should timeout unfinished requests.

Raises

- ValueError – if neither of the *wsdl* parameter or the **NRE_LDBWS_WSDL** environment variable are provided.

- `ValueError` – if neither of the *api_key* parameter or the **NRE_LDBWS_API_KEY** environment variable are provided.

get_fastest_departures(*crs*, *destinations*, *time_offset=None*, *time_window=None*)

Get the fastest public departures (within the supplied time window and offset) from the station indicated by *crs* to the stations indicated by *destinations*. The difference between this method and *get_next_departures* is that for each destination, the train which arrives first at the destination out of the next departures from this station is returned, rather than the one which departs from this station first.

Parameters

- **crs** (*str*) – the CRS code of the station for which this board is being fetched.

- **destinations** (*[str]*) – a list of CRS codes representing the stations for which the next departure from *crs* will be fetched. This parameter must contain at least 1, but no more than 25 station CRS codes.

- **time_offset** (*int, from -120 to 120*) – An offset in minutes against the current time which determines the starting point of the time window for which services are returned. If set to *None*, the value of 0 will be used.

- **time_window** (*int, from -120 to 120*) – How far into the future from the value passed as *time_offset* should services be fetched. If the value passed is negative, the time window starts before the value of *time_offset* and ends at *time_offset*. If *None* is passed, the default value is 120.

Returns a *NextDeparturesBoard* object containing the station details and the fastest departures to each of the requested destinations.

Return type *NextDeparturesBoard*

Raises `ValueError` – if *destinations* is not a list of between 1 and 25 values.

Note: Each time this his method is called, it makes **1** request to the LDBWS server.

---

get_fastest_departures_with_details(*crs*, *destinations*, *time_offset=None*, *time_window=None*)

Get the fastest public departures (within the supplied time window and offset) from the station indicated by *crs* to the stations indicated by *destinations*. The difference between this method and *get_next_departures_with_details* is that for each destination, the train which arrives first at the destination out of the next departures from this station is returned, rather than the one which departs from this station first.

This method is identical in arguments and result to *get_fastest_departures*, except that the returned result is of type *NextDeparturesBoardWithDetails*, which includes the calling points on the services, allowing access to them without an additional call to *get_service_details* for each service.

Parameters

- **crs** (*str*) – the CRS code of the station for which this board is being fetched.

- **destinations** (*[str]*) – a list of CRS codes representing the stations for which the next departure from *crs* will be fetched. This parameter must contain at least 1, but no more than 25 station CRS codes.

- **time_offset** (*int, from -120 to 120*) – An offset in minutes against the current time which determines the starting point of the time window for which services are returned. If set to *None*, the value of 0 will be used.

- **time_window** (*int, from -120 to 120*) – How far into the future from the value passed as *time_offset* should services be fetched. If the value passed is negative, the time

window starts before the value of *time_offset* and ends at *time_offset*. If *None* is passed, the default value is 120.

> **Returns** a *NextDeparturesBoardWithDetails* object containing the station details and the fastest departures to each of the requested destinations.
>
> **Return type** *[NextDeparturesBoardWithDetails](#)*
>
> **Raises** `ValueError` – if *destinations* is not a list of between 1 and 25 values.

Note: Each time this his method is called, it makes **1** request to the LDBWS server.

---

get_next_departures(*crs*, *destinations*, *time_offset=None*, *time_window=None*)
Get the next public departures (within the supplied time window and offset) from the station indicated by *crs* to the stations indicated by *destinations*.

> **Parameters**
>
> - **crs** (`str`) – the CRS code of the station for which this board is being fetched.
>
> - **destinations** (`[str]`) – a list of CRS codes representing the stations for which the next departure from *crs* will be fetched. This parameter must contain at least 1, but no more than 25 station CRS codes.
>
> - **time_offset** (`int, from -120 to 120`) – An offset in minutes against the current time which determines the starting point of the time window for which services are returned. If set to *None*, the value of 0 will be used.
>
> - **time_window** (`int, from -120 to 120`) – How far into the future from the value passed as *time_offset* should services be fetched. If the value passed is negative, the time window starts before the value of *time_offset* and ends at *time_offset*. If *None* is passed, the default value is 120.
>
> **Returns** a *NextDeparturesBoard* object containing the station details and the next departures to each of the requested destinations.
>
> **Return type** *[NextDeparturesBoard](#)*
>
> **Raises** `ValueError` – if *destinations* is not a list of between 1 and 25 values.

Note: Each time this his method is called, it makes **1** request to the LDBWS server.

---

get_next_departures_with_details(*crs*, *destinations*, *time_offset=None*, *time_window=None*)
Get the next public departures (within the supplied time window and offset) from the station indicated by *crs* to the stations indicated by *destinations*.

This method is identical in arguments and result to *get_next_departures*, except that the returned result is of type *NextDeparturesBoardWithDetails*, which includes the calling points on the services, allowing access to them without an additional call to *get_service_details* for each service.

> **Parameters**
>
> - **crs** (`str`) – the CRS code of the station for which this board is being fetched.
>
> - **destinations** (`[str]`) – a list of CRS codes representing the stations for which the next departure from *crs* will be fetched. This parameter must contain at least 1, but no more than 25 station CRS codes.

- **time_offset** (*int, from -120 to 120*) – An offset in minutes against the current time which determines the starting point of the time window for which services are returned. If set to *None*, the value of 0 will be used.

- **time_window** (*int, from -120 to 120*) – How far into the future from the value passed as *time_offset* should services be fetched. If the value passed is negative, the time window starts before the value of *time_offset* and ends at *time_offset*. If *None* is passed, the default value is 120.

> **Returns** a *NextDeparturesBoardWithDetails* object containing the station details and the next departures to each of the requested destinations.

> **Return type** *[NextDeparturesBoardWithDetails](#)*

> **Raises** `ValueError` – if *destinations* is not a list of between 1 and 25 values.

Note: Each time this his method is called, it makes **1** request to the LDBWS server.

---

get_service_details(*service_id*)
> Get the full details of a service from a board.

> > **Parameters service_id** (`str`) – the service_id of the relevant ServiceItem on the board.

> > **Returns** a *ServiceDetails* object containing the details of the requested service.

> > **Return type** *[ServiceDetails](#)*

Note: Each time this his method is called, it makes **1** request to the LDBWS server.

---

get_station_board(*crs, rows=10, include_departures=True, include_arrivals=False, from_filter_crs=None, to_filter_crs=None, time_offset=None, time_window=None*)
> Get a list of public services at a station as would populate a departure/arrival board.

> > **Parameters**

- **crs** (`str`) – the CRS code of the station for which this board is being fetched.

- **rows** (*int, from 1 to 150*) – the maximum number of services to include in the returned board.

- **include_departures** (`boolean`) – whether the returned services should include departures from this station. At least one of *include_departures* or *include_arrivals* must be set to true.

- **include_arrivals** (`boolean`) – whether the returned services should include arrivals at this station. At least one of *include_departures* or *include_arrivals* must be set to true.

- **from_filter_crs** (`str`) – the CRS code of a station at which all services returned must have called previously. Only one of *from_filter_crs* and *to_filter_crs* can be set for a given request.

- **to_filter_crs** (`str`) – the CRS code of a station at which all services returned must subsequently call. Only one of *from_filter_crs* and *to_filter_crs* can be set for a given request.

- **time_offset** (*int, from -120 to 120*) – An offset in minutes against the current time which determines the starting point of the time window for which services are returned. If set to *None*, the value of 0 will be used.

- **time_window**(*int, from -120 to 120*) – How far into the future from the value passed as *time_offset* should services be fetched. If the value passed is negative, the time window starts before the value of *time_offset* and ends at *time_offset*. If *None* is passed, the default value is 120.

Returns a *StationBoard* object containing the station details and the requested services.

Return type *StationBoard*

Raises ValueError – if neither include_departures or include_arrivals are set to True.

Note: Each time this his method is called, it makes **1** request to the LDBWS server.

---

get_station_board_with_details(*crs*, *rows=10*, *include_departures=True*, *include_arrivals=False*, *from_filter_crs=None*, *to_filter_crs=None*, *time_offset=None*, *time_window=None*)

Get a list of public services at a station as would populate a departure/arrival board. This method is identical in arguments and result to *get_station_board*, except that the returned result is of type *StationBoardWithDetails*, which includes the calling points on the services, allowing access to them without an additional call to *get_service_details* for each service.

Parameters

- **crs** (*str*) – the CRS code of the station for which this board is being fetched.

- **rows** (*int, from 1 to 150*) – the maximum number of services to include in the returned board.

- **include_departures** (*boolean*) – whether the returned services should include departures from this station. At least one of *include_departures* or *include_arrivals* must be set to true.

- **include_arrivals** (*boolean*) – whether the returned services should include arrivals at this station. At least one of *include_departures* or *include_arrivals* must be set to true.

- **from_filter_crs** (*str*) – the CRS code of a station at which all services returned must have called previously. Only one of *from_filter_crs* and *to_filter_crs* can be set for a given request.

- **to_filter_crs** (*str*) – the CRS code of a station at which all services returned must subsequently call. Only one of *from_filter_crs* and *to_filter_crs* can be set for a given request.

- **time_offset** (*int, from -120 to 120*) – An offset in minutes against the current time which determines the starting point of the time window for which services are returned. If set to *None*, the value of 0 will be used.

- **time_window** (*int, from -120 to 120*) – How far into the future from the value passed as *time_offset* should services be fetched. If the value passed is negative, the time window starts before the value of *time_offset* and ends at *time_offset*. If *None* is passed, the default value is 120.

Returns a *StationBoardWithDetails* object containing the station details and the requested services, along with their calling points.

Return type *StationBoardWithDetails*

Raises ValueError – if neither include_departures or include_arrivals are set to True.

Note: Each time this his method is called, it makes **1** request to the LDBWS server.

5.1.2 BoardBase

class nrewebservices.ldbws.responses.**BoardBase**(*soap_response*)

Bases: nrewebservices.common.soap.SoapResponseObject

This class acts as the base class containing the common attributes for the various classes that encapsulate API responses in the form of a list of services at a station. You do not normally need to instantiate this class directly.

generated_at
: *datetime* – the time at which the board was generated on the LDBWS server.

location_name
: *str* – the name of the location the board is for.

crs
: *str* – the CRS code of the location the board is for.

filter_location_name
: *str* – if a filter was provided in the request, the name of the location at which the board services were filtered.

filter_crs
: *str* – if a filter was provided in the request, the CRS code of the location at which the board services were filtered.

filter_type
: *str* – if a filter was provided in the request, this can contain either "from", indicating that the filtered services must have previously called at the filter location or "to", indicating that the filtered services must subsequently call at the filter location.

platforms_available
: *boolean* – if true, this indicates that platform information is available at this station and can be provided in the user interface. If false, this means that platform information is not currently available at this station and should be suppressed in the user interface.

services_available
: *boolean* – if true, this indicates that services are currently available in the returned board. If false, this means that services will not be returned in the provided board. An example of when this might be set false is when a station has been closed due to an incident, but trains are still passing through non-stopping. Normally the *nrcc_messages* will contain a message explaining the reason for this when it occurs.

nrcc_messages
: *list[str]* – a list of textual messages that should be displayed with the services on the board. These typically contain service information at times when there are problems or disruption. They may sometimes contain HTML style <A> an <P> tags.

5.1.3 StationBoard

class nrewebservices.ldbws.responses.**StationBoard**(*soap_response*, **args*, ***kwargs*)

Bases: *nrewebservices.ldbws.responses.BoardBase*

This class represents the arrivals/departures board of a station, provided in response to a *get_station_board* request. You do not normally need to instantiate this class directly.

---

train_services
> *list[ServiceItem]* – the list of train services that appear on the requested board. This list is provided in the order in which it should be displayed.

bus_services
> *list[ServiceItem]* – the list of bus services that appear on the requested board. This list is provided in the order in which it should be displayed.

ferry_services
> *list[ServiceItem]* – the list of ferry services that appear on the requested board. This list is provided in the order in which it should be displayed.

Note: If you are showing a single combined list of train, bus and ferry services, the sort order that is typically used is to sort the services based on scheduled time (arrival/departure as appropriate) with the lowest first. Be careful when showing services across midnight.

5.1.4 StationBoardWithDetails

class nrewebservices.ldbws.responses.**StationBoardWithDetails**(*soap_response*, **args*, ***kwargs*)

> Bases: *nrewebservices.ldbws.responses.BoardBase*

This class represents the arrivals/departures board of a station, provided in response to a *get_station_board_with_details* request. The difference from *StationBoard* is that this lists of services are of type ServiceItemWithCallingPoints, which includes the service calling points, which would otherwise have to be requested with individual calls to *get_service_details*. You do not normally need to instantiate this class directly.

train_services
> *list[ServiceItemWithCallingPoints]* – the list of train services that appear on the requested board. This list is provided in the order in which it should be displayed.

bus_services
> *list[ServiceItemWithCallingPoints]* – the list of bus services that appear on the requested board. This list is provided in the order in which it should be displayed.

ferry_services
> *list[ServiceItemWithCallingPoints]* – the list of ferry services that appear on the requested board. This list is provided in the order in which it should be displayed.

Note: If you are showing a single combined list of train, bus and ferry services, the sort order that is typically used is to sort the services based on scheduled time (arrival/departure as appropriate) with the lowest first. Be careful when showing services across midnight.

5.1.5 NextDeparturesBoard

class nrewebservices.ldbws.responses.**NextDeparturesBoard**(*soap_response*, **args*, ***kwargs*)

> Bases: *nrewebservices.ldbws.responses.BoardBase*

This class represents the board containing the next departures to the requested destinations. You do not normally need to instantiate this class directly.

---

next_departures
> *list[NextDeparturesItem]* – a list of objects containing one of the requested locations and the next services that depart to that location.

5.1.6 NextDeparturesBoardWithDetails

class nrewebservices.ldbws.responses.**NextDeparturesBoardWithDetails**(*soap_response*,
                                                                                                    *args,
                                                                                                    **kwargs)

    Bases: *[nrewebservices.ldbws.responses.BoardBase](#)*

This class represents the board containing the next departures to the requested destinations. The difference from *NextDeparturesBoard* is that the services returned within the *NextDepartureItem's of this board are of type 'NexteDepartureItemWithCallingPoints*, which includes the service calling points, which otherwise would have to be requested with individual calls to *get_service_details*. You do not normally need to instantiate this class directly.

next_departures
> *list[NextDeparturesItemWithCallingPoints]* – a list of objects containing one of the requested locations and the next services that depart to that location.

5.1.7 ServiceItemBase

class nrewebservices.ldbws.responses.**ServiceItemBase**(*soap_response*)
    Bases: nrewebservices.common.soap.SoapResponseObject

This class acts as the base class containing the common attributes for the various classes that encapsulate individual services as shown on the various boards. You do not normally need to instantiate this class directly.

origins
> *list[ServiceLocation]* – a list of the origins of this service. Services may have multiple origins when they are formed of two separate trains that have been joined en-route. This attribute is only populated for boards where arrivals are included.

destinations
> *list[ServiceLocation]* – a list of the destinations of this service. Services may have multiple destinations when they split at a subsequent station into two trains to different destinations. This attribute is only populated for boards where departures are included.

current_origins
> *list[ServiceLocation]* – a list of the currently valid origins of the service. This attribute is only populated when the scheduled origin(s) have been cancelled and the board requsted includes arrivals.

current_destinations
> *list[ServiceLocation]* – a list of the currently valid destinations of the service. This attribute is only populated when the scheduled destination(s) have been cancelled and the board requested includes departures.

sta
> *str* – the scheduled arrival time of the service at this location. The value of the field is as outlined in the *[LDBWS Times Section](#)*. This field is only populated when the board requested includes arrivals and there is an arrival event scheduled at this location.

eta
> *str* – the estimated arrival time of the service at this location. The value of the field is as outlined in the *[LDBWS Times Section](#)*. This field is only populated when the board requested includes arrivals and there is an arrival event scheduled at this location.

std
> *str* – the scheduled departure time of the service at this location. The value of the field is as outlined in the *LDBWS Times Section*. This field is only populated when the board requested includes departures and there is a departure event scheduled at this location.

etd
> *str* – the estimated departure time of the service at this location. The value of the field is as outlined in the *LDBWS Times Section*. This field is only populated when the board requested includes departures and there is a departure event scheduled at this location.

platform
> *str* – the platform that this service will use at this location. This will only be present where it is available from station CIS systems and where *platforms_available* is set to True on the parent board.

operator
> *str* – the name of the Train Operating Company that operates this service.

operator_code
> *str* – the two-letter code identifying the Train Operating Company that operates this service. Please see the Open Rail Data Wiki Toc Codes Page for the full list.

circular_route
> *boolean* – when True, this service is operating on a circular route and will call at this station again later on its journey. This should be clearly communicated in user interfaces to ensure users can effectively identify the right train to take from the different options that exist in this case.

cancelled
> *boolean* – when True, indicates that this service has been cancelled at this location.

filter_location_cancelled
> *boolean* – when True, indicates that the service has had its stop at the requested filter location cancelled and now will not stop there.

service_type
> *str* – the type of transport this service consists of. Can be *train*, *bus* or *ferry*.

length
> *str* – the train length (in number of units). If this is set to 0 then the length of the train is unknown.

detach_front
> *boolean* – if True then the service detaches units from the front at this location.

reverse_formation
> *boolean* – if True then the service is operating in reverse formation (i.e. the order of the train carriages is the reverse of what it normally is).

cancel_reason
> *str* – if this service is canelled, the reason for this cancellation.

delay_reason
> *str* – if this service is delayed, the reason for this delay.

service_id
> *str* – the unique identifier of this service relative to the station of the board at which this service exists. This ID can be used to lookup the service details with *get_service_details*. However, this ID is not useful for using with any other API, and will expire and stop working within a few hours of the service being deactivated.

adhoc_alerts
> *str* – a list of adhoc alerts to show for this service at this location.

rsid
> *str* – the Retail Service ID of the service, if known by the Darwin system.

origin
> *str* – the origin (or origins) of this service as a single string which is suitable for display directly to users.

destination
> *str* – the destination (or destinations) of this service as a single string which is suitable for display directly to users.

5.1.8 ServiceItem

class nrewebservices.ldbws.responses.**ServiceItem**(*soap_response*, **args*, ***kwargs*)
> Bases: *nrewebservices.ldbws.responses.ServiceItemBase*

This class represents a single service that appears on an arrival/departure board. You do not normally need to instantiate this class directly.

5.1.9 ServiceItemWithCallingPoints

class nrewebservices.ldbws.responses.**ServiceItemWithCallingPoints**(*soap_response*,
> **args*,
> ***kwargs*)

> Bases: *nrewebservices.ldbws.responses.ServiceItemBase*

This class represents a single service that appears on an arrival/departure board where details were requested. The main difference from a regular *ServiceItem* is that it is prepopulated with the previous and subsequent calling points of the service without requiring an additional request to *get_service_details* to retrieve these. You do not normally need to instantiate this class directly.

previous_calling_points
> *[CallingPointList]* – a list of CallingPointList objects on this service before the location at which the board was requested. If there is more than one CallingPointList, the first list represents the through service, and the subsequent lists represent associated services, with the location of the join being the fina station on each of the subsequent lists. This is property is only populated when requesting an arrivals or arrivals/departures board.

subsequent_calling_points
> *[CallingPoint]* – a list of CallingPointList objects on this service before the location at which the board was requested. If there is more than one CallingPointList, the first list represents the through service, and the subsequent lists represent associated services, with the location of the join being the final station on each of the subsequent lists. This property is only populated when requesting a departures or arrivals/departures board.

5.1.10 ServiceLocation

class nrewebservices.ldbws.responses.**ServiceLocation**(*soap_response*, **args*,
> ***kwargs*)
> Bases: nrewebservices.common.soap.SoapResponseObject

This class represents a single location (station) at which a service either begins or ends. You do not normally need to instantiate this class directly.

location_name
> *str* – the name of the location.

---

crs
> *str* – the CRS code of the location. A CRS code of ??? indicates that the location has no CRS code that is known to the Darwin system.

via
> *str* – when the service is taking an ambiguous route, a string of text describing the station or stations it travels via in order to disambiguate. This is typically displayed after the destination on departure boards. It is only present for destinations.

future_change_to
> *str* – if the service is to change to a different mode in future, that mode is indicated here (either bus/train/ferry).

association_is_cancelled
> *boolean* – True if the associated service has been cancelled that took this service to this origin or destination, meaning this location will no longer be reached by the running service.

5.1.11 CallingPoint

class nrewebservices.ldbws.responses.**CallingPoint**(*soap_response*, **args*, ***kwargs*)
> Bases: nrewebservices.common.soap.SoapResponseObject

This class represents a single calling point along a service (i.e. a the details of a stop made at a station by the service). You do not normally need to instantiate this class directly.

location_name
> *str* – the name of the station where the stop occurs.

crs
> *str* – the CRS code of the station at which the stop occurs. A CRS code of ??? indicates that the location has no CRS code that is known to the Darwin system.

st
> *str* – the scheduled time at which the stop occurs. If this is a previous calling point, the value will be the departure time. If it is a subsequent calling point, then the value will be an arrival time. The value of this field is as outlined in the section on *LDBWS times*.

et
> *str* – the estimated time at which the service will make this stop. If this is a previous calling point, the value will be the departure time. If it is a subsequent calling point then the value will be the arrival time. Only one of *et* and *at* is ever populated at a time for a given calling point. The value of this field is as outlined in the section on *LDBWS times*.

at
> *str* – the actual time at which the service made this stop. If this is a previous calling point, the value will be the departure time. If it is a subsequent calling point then the value will be the arrival time. Only one of *et* and *at* is ever populated at a time for a given calling point. The value of this field is as outlined in the section on *LDBWS times*.

cancelled
> *boolean* – when True, indicates that this service is cancelled at this location.

length
> *str* – the train length (in number of units). If this is set to 0 then the length of the train is unknown.

detach_front
> *boolean* – if True then the service detaches units from the front at this location.

adhoc_alerts
> *str* – a list of adhoc alerts to show for this service at this location.

5.1.12 CallingPointList

class nrewebservices.ldbws.responses.**CallingPointList**(*soap_response*)
> Bases: nrewebservices.common.soap.SoapResponseObject

> This class represents a list of calling points along a service. You do not normally need to instantiate this class directly.

> **calling_points**
>> *[CallingPoint]* – the list of calling points this CallingPointList represents.

> **service_type**
>> *str* – the type of service these calling points are provided by. Can be either 'train', 'bus' or 'ferry'.

> **change_required**
>> *boolean* – if True, passengers must change vehicles to access these calling points.

> **association_is_cancelled**
>> *boolean* – if True, the association between the service providing these calling points and the through service has been cancelled, meaning these calling points can no longer be reached on this service.

> **Note:** If a CallingPointList is treated as a regular python list, the calling_points property is implicitly accessed.

5.1.13 NextDeparturesItemBase

class nrewebservices.ldbws.responses.**NextDeparturesItemBase**(*soap_response*)
> Bases: nrewebservices.common.soap.SoapResponseObject

> This class acts as the base clas for entries on the next departures boards, containing the common attributes therof. You do not normally need to instantiate this class directly.

> **crs**
>> *str* – the CRS code of the location that this departure item contains the next departures to.

5.1.14 NextDeparturesItem

class nrewebservices.ldbws.responses.**NextDeparturesItem**(*soap_response*, *args*, *kwargs*)
> Bases: *nrewebservices.ldbws.responses.NextDeparturesItemBase*

> This class represents an entry on a NextDeparturesBoard, containing the next departures to a single location. You do not normally need to instantiate this class directly.

> **service**
>> *ServiceItem* – the next service to depart to the requested location.

5.1.15 NextDeparturesItemWithCallingPoints

class nrewebservices.ldbws.responses.**NextDeparturesItemWithCallingPoints**(*soap_response*, *args*, *kwargs*)
> Bases: *nrewebservices.ldbws.responses.NextDeparturesItemBase*

This class represents an entry on a NextDeparturesBoardWithDetails, containing the next departures to a single location. The difference between this and *NextDeparturesItem* is that the services are of type *ServiceItemWithCallingPoints*, which includes the service calling points which otherwise would have to be requested with individual calls to *get_service_details*. You do not normally need to instantiate this class directly.

service
> *ServiceItemWithCallingPoints* – the next service to depart to the requested location.

5.1.16 ServiceDetails

class nrewebservices.ldbws.responses.**ServiceDetails**(*soap_response*, **args*, ***kwargs*)
> Bases: nrewebservices.common.soap.SoapResponseObject

This class represents the details of a service from the point of view of the station at which that service was requested. You do not normally need to instantiate this class directly.

generated_at
> *datetime* – the time at which the service details were generated on the LDBWS server.

rsid
> *str* – the Retail Service ID of the service, if known by the Darwin system.

service_type
> *str* – The type of service ("bus/ferry/train"). Note that real-time information such as estimated and actual times and cancellations is only available for train services.

location_name
> *str* – the name of the location at which this service was requested. This is determined by the board that the service ID was originally retrieved from. All other fields in this object are from the point of view of inspecting the service at this location.

crs
> *str* – the CRS code of the location at which this service was requested. See the *location_name* attribute for more details.

operator
> *str* – the name of the Train Operating Company that operates this service.

operator_code
> *str* – the two-letter code identifying the Train Operating Company that operates this service. Please see the Open Rail Data Wiki Toc Codes Page for the full list.

cancelled
> *boolean* – when True, indicates that this service has been cancelled at this location.

cancel_reason
> *str* – if this service is canelled, the reason for this cancellation.

delay_reason
> *str* – if this service is delayed, the reason for this delay.

overdue_message
> *str* – if a report on an expected movement of this service has not been received on time, this attribute will contain a plain-English explanation of the report which was expected.

length
> *str* – the train length (in number of units). If this is set to 0 then the length of the train is unknown.

detach_front
> *boolean* – if True then the service detaches units from the front at this location.

reverse_formation
> *boolean* – if True then the service is operating in reverse formation (i.e. the order of the train carriages is the reverse of what it normally is).

platform
> *str* – the platform that this service will use at this location. This will only be present where it is available from station CIS systems and where *platforms_available* is set to True on the board from which this service was requested.

sta
> *str* – the scheduled arrival time of the service at this location. The value of the field is as outlined in the *LDBWS Times Section*. This field is only populated when the board requested includes arrivals and there is an arrival event scheduled at this location.

eta
> *str* – the estimated arrival time of the service at this location. The value of the field is as outlined in the *LDBWS Times Section*. This field is only populated when there is an arrival event scheduled at this location. Only one of *eta* and *ata* will be present at a time for a given service.

ata
> *str* – the actual arrival time of the service at this location. The value of the field is as outlined in the *LDBWS Times Section*. This field is only populated when there is an arrival event scheduled at this location. Only one of *eta* and *ata* will be present at a time for a given service.

std
> *str* – the scheduled departure time of the service at this location. The value of the field is as outlined in the *LDBWS Times Section*. This field is only populated when the board requested includes departures and there is a departure event scheduled at this location.

etd
> *str* – the estimated departure time of the service at this location. The value of the field is as outlined in the *LDBWS Times Section*. This field is only populated when there is a departure event scheduled at this location. Only one of *etd* and *atd* will be present at a time for a given service.

atd
> *str* – the actual departure time of the service at this location. The value of the field is as outlined in the *LDBWS Times Section*. This field is only populated when there is a departure event scheduled at this location. Only one of *etd* and *atd* will be present at a time for a given service.

adhoc_alerts
> *str* – a list of adhoc alerts to show for this service at this location.

previous_calling_points
> *[CallingPointList]* – a list of CallingPointList objects on this service before the location at which the board was requested. If there is more than one CallingPointList, the first list represents the through service, and the subsequent lists represent associated services, with the location of the join being the final station on each of the subsequent lists.

subsequent_calling_points
> *[CallingPointList]* – a list of CallingPointList objects on this service after the location at which the board was requested. If there is more than one CallingPointList, the first list represents the through service, and the subsequent lists represent associated services, with the location of the join being the first station on each of the subsequent lists.

5.2 nrewebservice.staff

> **Warning:** The LDBWS Staff Version API is not supported in this version of NREWebServices. Support for it will be added in a future version.

5.3 nrewebservices.hsp

> **Warning:** The Historical Service Performance API is not supported in this version of NREWebServices. Support for it will be added in a future version.

CHAPTER 6

Indices

- genindex

- modindex

Python Module Index

n

Index

Symbols

__init__() (nrewebservices.ldbws.Session method), [11](#)

A

adhoc_alerts (nrewebservices.ldbws.responses.CallingPoint attribute), [21](#)

adhoc_alerts (nrewebservices.ldbws.responses.ServiceDetails attribute), [24](#)

adhoc_alerts (nrewebservices.ldbws.responses.ServiceItemBase attribute), [19](#)

association_is_cancelled (nrewebservices.ldbws.responses.CallingPointList attribute), [22](#)

association_is_cancelled (nrewebservices.ldbws.responses.ServiceLocation attribute), [21](#)

at (nrewebservices.ldbws.responses.CallingPoint attribute), [21](#)

ata (nrewebservices.ldbws.responses.ServiceDetails attribute), [24](#)

atd (nrewebservices.ldbws.responses.ServiceDetails attribute), [24](#)

B

BoardBase (class in nrewebservices.ldbws.responses), [16](#)

bus_services (nrewebservices.ldbws.responses.StationBoard attribute), [17](#)

bus_services (nrewebservices.ldbws.responses.StationBoardWithDetails attribute), [17](#)

C

calling_points (nrewebservices.ldbws.responses.CallingPointList attribute), [22](#)

CallingPoint (class in nrewebservices.ldbws.responses), [21](#)

CallingPointList (class in nrewebservices.ldbws.responses), [22](#)

cancel_reason (nrewebservices.ldbws.responses.ServiceDetails attribute), [23](#)

cancel_reason (nrewebservices.ldbws.responses.ServiceItemBase attribute), [19](#)

cancelled (nrewebservices.ldbws.responses.CallingPoint attribute), [21](#)

cancelled (nrewebservices.ldbws.responses.ServiceDetails attribute), [23](#)

cancelled (nrewebservices.ldbws.responses.ServiceItemBase attribute), [19](#)

change_required (nrewebservices.ldbws.responses.CallingPointList attribute), [22](#)

circular_route (nrewebservices.ldbws.responses.ServiceItemBase attribute), [19](#)

crs (nrewebservices.ldbws.responses.BoardBase attribute), [16](#)

crs (nrewebservices.ldbws.responses.CallingPoint attribute), [21](#)

crs (nrewebservices.ldbws.responses.NextDeparturesItemBase attribute), [22](#)

crs (nrewebservices.ldbws.responses.ServiceDetails attribute), [23](#)

crs (nrewebservices.ldbws.responses.ServiceLocation attribute), [20](#)

current_destinations (nrewebservices.ldbws.responses.ServiceItemBase attribute), [18](#)

current_origins (nrewebservices.ldbws.responses.ServiceItemBase attribute), [18](#)