

# Package: trainR (via r-universe)

August 23, 2024

**Title** An Interface to the National Rail Enquiries Systems

**Version** 0.0.1.9000

**Description** The goal of 'trainR' is to provide a simple interface to the National Rail Enquiries (NRE) systems. There are few data feeds available, the simplest of them is Darwin, which provides real-time arrival and departure predictions, platform numbers, delay estimates, schedule changes and cancellations. Other data feeds provide historical data, Historic Service Performance (HSP), and much more. 'trainR' simplifies the data retrieval, so that the users can focus on their analyses. For more details visit <<https://www.nationalrail.co.uk/46391.aspx>>.

**License** MIT + file LICENSE

**Encoding** UTF-8

**LazyData** true

**Roxygen** list(markdown = TRUE)

**RoxygenNote** 7.1.1

**Depends** R (>= 2.10)

**URL** <https://github.com/villegar/trainR/>,  
<https://villegar.github.io/trainR/>

**BugReports** <https://github.com/villegar/trainR/issues/>

**Language** en-GB

**Imports** RCurl, crayon, dplyr, glue, lubridate, magrittr, purrr,  
stringr, tibble, tidyr, usethis, xml2

**Suggests** knitr, openxlsx

**Repository** <https://villegar.r-universe.dev>

**RemoteUrl** <https://github.com/villegar/trainr>

**RemoteRef** HEAD

**RemoteSha** 2e01d35913abc51d9b562717619534b334c11f39

## Contents

GetArrBoardRequest . . . . .	2
GetArrBoardWithDetailsRequest . . . . .	4
GetArrDepBoardWithDetailsRequest . . . . .	6
GetDepBoardRequest . . . . .	8
GetDepBoardWithDetailsRequest . . . . .	10
GetServiceDetailsRequest . . . . .	12
get_crs . . . . .	13
get_token . . . . .	14
set_token . . . . .	15
station_codes . . . . .	15
store . . . . .	16
<b>Index</b>	<b>18</b>

---

GetArrBoardRequest	<i>Get all public arrivals</i>
--------------------	--------------------------------

---

## Description

Get all public arrivals for the supplied CRS code within a defined time window.

## Usage

```
GetArrBoardRequest(
  crs,
  filterCrs = NA,
  filterType = "from",
  numRows = 150,
  timeOffset = 0,
  timeWindow = 120,
  token = get_token(),
  url = "https://lite.realtime.nationalrail.co.uk/OpenLDBWS/ldb11.asmx",
  verbose = FALSE
)
```

## Arguments

crs	(string, 3 characters, alphabetic): The CRS code of the location for which the request is being made.
filterCrs	(string, 3 characters, alphabetic): The CRS code of either an origin or destination location to filter in. Optional.
filterType	(string, either "from" or "to"): The type of filter to apply. Filters services to include only those originating or terminating at the filterCrs location. Defaults to "to".
numRows	(integer, between 0 and 150 exclusive): The number of services to return in the resulting station board.

timeOffset	(integer, between -120 and 120 exclusive): An offset in minutes against the current time to provide the station board for. Defaults to 0.
timeWindow	(integer, between -120 and 120 exclusive): How far into the future in minutes, relative to timeOffset, to return services for. Defaults to 120.
token	Token to access the data feed. The token can be obtained at <a href="http://realtime.nationalrail.co.uk/OpenLDBWSRegistration/">http://realtime.nationalrail.co.uk/OpenLDBWSRegistration/</a> .
url	Data feed source URL.
verbose	Boolean flag to indicate whether or not to show status messages.

## Value

Tibble with arrival records. Each column is described below:

**generatedAt** The time at which the station board was generated.

**locationName** The name of the location that the station board is for.

**crs** The CRS code of the location that the station board is for.

**filterLocationName** If a filter was requested, the location name of the filter location.

**filtercrs** If a filter was requested, the CRS code of the filter location.

**filterType** If a filter was requested, the type of filter.

**nrccMessages** An optional list of textual messages that should be displayed with the station board. The message may include embedded and XML encoded HTML-like hyperlinks and paragraphs. The messages are typically used to display important disruption information that applies to the location that the station board was for. Any embedded <p> tags are used to force a new-line in the output. Embedded <a> tags allow links to external web pages that may provide more information. Output channels that do not support HTML should strip out the <a> tags and just leave the enclosed text.

**platformAvailable** An optional value that indicates if platform information is available. If this value is present with the value "true" then platform information will be returned in the service lists. If this value is not present, or has the value "false", then the platform "heading" should be suppressed in the user interface for this station board.

**areServicesAvailable** An optional value that indicates if services are currently available for this station board. If this value is present with the value "false" then no services will be returned in the service lists. This value may be set, for example, if access to a station has been closed to the public at short notice, even though the scheduled services are still running. It would be usual in such cases for one of the nrccMessages to describe why the list of services has been suppressed.

**trainServices** A list of ServiceItem object for each services of the relevant type that is to appear on the station board. May contain zero items, or may not be present at all.

**busServices** A list of ServiceItem object for each services of the relevant type that is to appear on the station board. May contain zero items, or may not be present at all.

**ferryServices** A list of ServiceItem object for each services of the relevant type that is to appear on the station board. May contain zero items, or may not be present at all.

**Source**

Documentation for the Live Departure Boards Web Service (LDBWS / OpenLDBWS): <http://lite.realtime.nationalrail.co.uk/openldbws/>

**See Also**

Other OpenLDBSVWS requests: [GetArrBoardWithDetailsRequest\(\)](#), [GetArrDepBoardWithDetailsRequest\(\)](#), [GetDepBoardRequest\(\)](#), [GetDepBoardWithDetailsRequest\(\)](#), [GetServiceDetailsRequest\(\)](#)

**Examples**

```
## Not run:
rdg_arr <- trainR::GetArrBoardRequest("RDG")
rdg_arr <- trainR::GetArrBoardRequest("RDG", filterCrs = "BRI")
trainR::print(rdg_arr)

## End(Not run)
```

---

GetArrBoardWithDetailsRequest  
*Get all public arrivals*

---

**Description**

Get all public arrivals for the supplied CRS code within a defined time window, including service details.

**Usage**

```
GetArrBoardWithDetailsRequest(
  crs,
  filterCrs = NA,
  filterType = "from",
  numRows = 150,
  timeOffset = 0,
  timeWindow = 120,
  token = get_token(),
  url = "https://lite.realtime.nationalrail.co.uk/OpenLDBWS/ldb11.aspx",
  verbose = FALSE
)
```

**Arguments**

crs	(string, 3 characters, alphabetic): The CRS code of the location for which the request is being made.
filterCrs	(string, 3 characters, alphabetic): The CRS code of either an origin or destination location to filter in. Optional.

<code>filterType</code>	(string, either "from" or "to"): The type of filter to apply. Filters services to include only those originating or terminating at the <code>filterCrS</code> location. Defaults to "to".
<code>numRows</code>	(integer, between 0 and 150 exclusive): The number of services to return in the resulting station board.
<code>timeOffset</code>	(integer, between -120 and 120 exclusive): An offset in minutes against the current time to provide the station board for. Defaults to 0.
<code>timeWindow</code>	(integer, between -120 and 120 exclusive): How far into the future in minutes, relative to <code>timeOffset</code> , to return services for. Defaults to 120.
<code>token</code>	Token to access the data feed. The token can be obtained at <a href="http://realtime.nationalrail.co.uk/OpenLDBWSRegistration/">http://realtime.nationalrail.co.uk/OpenLDBWSRegistration/</a> .
<code>url</code>	Data feed source URL.
<code>verbose</code>	Boolean flag to indicate whether or not to show status messages.

### Value

Tibble with arrival records. Each column is described below:

**generatedAt** The time at which the station board was generated.

**locationName** The name of the location that the station board is for.

**crs** The CRS code of the location that the station board is for.

**filterLocationName** If a filter was requested, the location name of the filter location.

**filtercrs** If a filter was requested, the CRS code of the filter location.

**filterType** If a filter was requested, the type of filter.

**nrccMessages** An optional list of textual messages that should be displayed with the station board. The message may include embedded and XML encoded HTML-like hyperlinks and paragraphs. The messages are typically used to display important disruption information that applies to the location that the station board was for. Any embedded `<p>` tags are used to force a new-line in the output. Embedded `<a>` tags allow links to external web pages that may provide more information. Output channels that do not support HTML should strip out the `<a>` tags and just leave the enclosed text.

**platformAvailable** An optional value that indicates if platform information is available. If this value is present with the value "true" then platform information will be returned in the service lists. If this value is not present, or has the value "false", then the platform "heading" should be suppressed in the user interface for this station board.

**areServicesAvailable** An optional value that indicates if services are currently available for this station board. If this value is present with the value "false" then no services will be returned in the service lists. This value may be set, for example, if access to a station has been closed to the public at short notice, even though the scheduled services are still running. It would be usual in such cases for one of the `nrccMessages` to describe why the list of services has been suppressed.

**trainServices** A list of `ServiceItem` object for each services of the relevant type that is to appear on the station board. May contain zero items, or may not be present at all.

**busServices** A list of `ServiceItem` object for each services of the relevant type that is to appear on the station board. May contain zero items, or may not be present at all.

**ferryServices** A list of ServiceItem object for each services of the relevant type that is to appear on the station board. May contain zero items, or may not be present at all.

### Source

Documentation for the Live Departure Boards Web Service (LDBWS / OpenLDBWS): <http://lite.realtime.nationalrail.co.uk/openldbws/>

### See Also

Other OpenLDBSVWS requests: [GetArrBoardRequest\(\)](#), [GetArrDepBoardWithDetailsRequest\(\)](#), [GetDepBoardRequest\(\)](#), [GetDepBoardWithDetailsRequest\(\)](#), [GetServiceDetailsRequest\(\)](#)

### Examples

```
## Not run:
rdg_arr <- trainR::GetArrBoardWithDetailsRequest("RDG")
rdg_arr <- trainR::GetArrBoardWithDetailsRequest("RDG", filterCrs = "BRI")
trainR::print(rdg_arr)

## End(Not run)
```

---

GetArrDepBoardWithDetailsRequest

*Get all public arrivals and departures*

---

### Description

Get all public arrivals and departures for the supplied CRS code within a defined time window, including service details.

### Usage

```
GetArrDepBoardWithDetailsRequest(
  crs,
  filterCrs = NA,
  filterType = "from",
  numRows = 150,
  timeOffset = 0,
  timeWindow = 120,
  token = get_token(),
  url = "https://lite.realtime.nationalrail.co.uk/OpenLDBWS/ldb11.aspx",
  verbose = FALSE
)
```

**Arguments**

<code>crs</code>	(string, 3 characters, alphabetic): The CRS code of the location for which the request is being made.
<code>filterCrs</code>	(string, 3 characters, alphabetic): The CRS code of either an origin or destination location to filter in. Optional.
<code>filterType</code>	(string, either "from" or "to"): The type of filter to apply. Filters services to include only those originating or terminating at the <code>filterCrs</code> location. Defaults to "to".
<code>numRows</code>	(integer, between 0 and 150 exclusive): The number of services to return in the resulting station board.
<code>timeOffset</code>	(integer, between -120 and 120 exclusive): An offset in minutes against the current time to provide the station board for. Defaults to 0.
<code>timeWindow</code>	(integer, between -120 and 120 exclusive): How far into the future in minutes, relative to <code>timeOffset</code> , to return services for. Defaults to 120.
<code>token</code>	Token to access the data feed. The token can be obtained at <a href="http://realtime.nationalrail.co.uk/OpenLDBWSRegistration/">http://realtime.nationalrail.co.uk/OpenLDBWSRegistration/</a> .
<code>url</code>	Data feed source URL.
<code>verbose</code>	Boolean flag to indicate whether or not to show status messages.

**Value**

Tibble with arrival and departure records. Each column is described below:

**generatedAt** The time at which the station board was generated.

**locationName** The name of the location that the station board is for.

**crs** The CRS code of the location that the station board is for.

**filterLocationName** If a filter was requested, the location name of the filter location.

**filtercrs** If a filter was requested, the CRS code of the filter location.

**filterType** If a filter was requested, the type of filter.

**nrrccMessages** An optional list of textual messages that should be displayed with the station board. The message may include embedded and XML encoded HTML-like hyperlinks and paragraphs. The messages are typically used to display important disruption information that applies to the location that the station board was for. Any embedded `<p>` tags are used to force a new-line in the output. Embedded `<a>` tags allow links to external web pages that may provide more information. Output channels that do not support HTML should strip out the `<a>` tags and just leave the enclosed text.

**platformAvailable** An optional value that indicates if platform information is available. If this value is present with the value "true" then platform information will be returned in the service lists. If this value is not present, or has the value "false", then the platform "heading" should be suppressed in the user interface for this station board.

**areServicesAvailable** An optional value that indicates if services are currently available for this station board. If this value is present with the value "false" then no services will be returned in the service lists. This value may be set, for example, if access to a station has been closed to the public at short notice, even though the scheduled services are still running. It would be

usual in such cases for one of the nrccMessages to describe why the list of services has been suppressed.

**trainServices** A list of ServiceItem object for each services of the relevant type that is to appear on the station board. May contain zero items, or may not be present at all.

**busServices** A list of ServiceItem object for each services of the relevant type that is to appear on the station board. May contain zero items, or may not be present at all.

**ferryServices** A list of ServiceItem object for each services of the relevant type that is to appear on the station board. May contain zero items, or may not be present at all.

### Source

Documentation for the Live Departure Boards Web Service (LDBWS / OpenLDBWS): <http://lite.realtime.nationalrail.co.uk/openldbws/>

### See Also

Other OpenLDBSVWS requests: [GetArrBoardRequest\(\)](#), [GetArrBoardWithDetailsRequest\(\)](#), [GetDepBoardRequest\(\)](#), [GetDepBoardWithDetailsRequest\(\)](#), [GetServiceDetailsRequest\(\)](#)

### Examples

```
## Not run:
rdg<- trainR::GetArrDepBoardWithDetailsRequest("RDG")
rdg <- trainR::GetArrDepBoardWithDetailsRequest("RDG", filterCrs = "BRI")
trainR::print(rdg)

## End(Not run)
```

---

GetDepBoardRequest	<i>Get all public departures</i>
--------------------	----------------------------------

---

### Description

Get all public departures for the supplied CRS code within a defined time window.

### Usage

```
GetDepBoardRequest(
  crs,
  filterCrs = NA,
  filterType = "to",
  numRows = 150,
  timeOffset = 0,
  timeWindow = 120,
  token = get_token(),
  url = "https://lite.realtime.nationalrail.co.uk/OpenLDBWS/ldb11.aspx",
  verbose = FALSE
)
```



**Arguments**

<code>crs</code>	(string, 3 characters, alphabetic): The CRS code of the location for which the request is being made.
<code>filterCrs</code>	(string, 3 characters, alphabetic): The CRS code of either an origin or destination location to filter in. Optional.
<code>filterType</code>	(string, either "from" or "to"): The type of filter to apply. Filters services to include only those originating or terminating at the <code>filterCrs</code> location. Defaults to "to".
<code>numRows</code>	(integer, between 0 and 150 exclusive): The number of services to return in the resulting station board.
<code>timeOffset</code>	(integer, between -120 and 120 exclusive): An offset in minutes against the current time to provide the station board for. Defaults to 0.
<code>timeWindow</code>	(integer, between -120 and 120 exclusive): How far into the future in minutes, relative to <code>timeOffset</code> , to return services for. Defaults to 120.
<code>token</code>	Token to access the data feed. The token can be obtained at <a href="http://realtime.nationalrail.co.uk/OpenLDBWSRegistration/">http://realtime.nationalrail.co.uk/OpenLDBWSRegistration/</a> .
<code>url</code>	Data feed source URL.
<code>verbose</code>	Boolean flag to indicate whether or not to show status messages.

**Value**

Tibble with departure records. Each column is described below:

**generatedAt** The time at which the station board was generated.

**locationName** The name of the location that the station board is for.

**crs** The CRS code of the location that the station board is for.

**filterLocationName** If a filter was requested, the location name of the filter location.

**filtercrs** If a filter was requested, the CRS code of the filter location.

**filterType** If a filter was requested, the type of filter.

**nrrccMessages** An optional list of textual messages that should be displayed with the station board. The message may include embedded and XML encoded HTML-like hyperlinks and paragraphs. The messages are typically used to display important disruption information that applies to the location that the station board was for. Any embedded `<p>` tags are used to force a new-line in the output. Embedded `<a>` tags allow links to external web pages that may provide more information. Output channels that do not support HTML should strip out the `<a>` tags and just leave the enclosed text.

**platformAvailable** An optional value that indicates if platform information is available. If this value is present with the value "true" then platform information will be returned in the service lists. If this value is not present, or has the value "false", then the platform "heading" should be suppressed in the user interface for this station board.

**areServicesAvailable** An optional value that indicates if services are currently available for this station board. If this value is present with the value "false" then no services will be returned in the service lists. This value may be set, for example, if access to a station has been closed to the public at short notice, even though the scheduled services are still running. It would be

usual in such cases for one of the nrccMessages to describe why the list of services has been suppressed.

**trainServices** A list of ServiceItem object for each services of the relevant type that is to appear on the station board. May contain zero items, or may not be present at all.

**busServices** A list of ServiceItem object for each services of the relevant type that is to appear on the station board. May contain zero items, or may not be present at all.

**ferryServices** A list of ServiceItem object for each services of the relevant type that is to appear on the station board. May contain zero items, or may not be present at all.

### Source

Documentation for the Live Departure Boards Web Service (LDBWS / OpenLDBWS): <http://lite.realtime.nationalrail.co.uk/openldbws/>

### See Also

Other OpenLDBSVWS requests: [GetArrBoardRequest\(\)](#), [GetArrBoardWithDetailsRequest\(\)](#), [GetArrDepBoardWithDetailsRequest\(\)](#), [GetDepBoardWithDetailsRequest\(\)](#), [GetServiceDetailsRequest\(\)](#)

### Examples

```
## Not run:
rdg_dep <- trainR::GetDepBoardRequest("RDG")
rdg_dep <- trainR::GetDepBoardRequest("RDG", filterCrs = "BRI")
trainR::print(rdg_dep)

## End(Not run)
```

---

GetDepBoardWithDetailsRequest  
*Get all public departures*

---

### Description

Get all public departures for the supplied CRS code within a defined time window, including service details.

### Usage

```
GetDepBoardWithDetailsRequest(
  crs,
  filterCrs = NA,
  filterType = "to",
  numRows = 150,
  timeOffset = 0,
  timeWindow = 120,
  token = get_token(),
```

```

    url = "https://lite.realtime.nationalrail.co.uk/OpenLDBWS/ldb11.aspx",
    verbose = FALSE
)

```

### Arguments

<code>crs</code>	(string, 3 characters, alphabetic): The CRS code of the location for which the request is being made.
<code>filterCrs</code>	(string, 3 characters, alphabetic): The CRS code of either an origin or destination location to filter in. Optional.
<code>filterType</code>	(string, either "from" or "to"): The type of filter to apply. Filters services to include only those originating or terminating at the <code>filterCrs</code> location. Defaults to "to".
<code>numRows</code>	(integer, between 0 and 150 exclusive): The number of services to return in the resulting station board.
<code>timeOffset</code>	(integer, between -120 and 120 exclusive): An offset in minutes against the current time to provide the station board for. Defaults to 0.
<code>timeWindow</code>	(integer, between -120 and 120 exclusive): How far into the future in minutes, relative to <code>timeOffset</code> , to return services for. Defaults to 120.
<code>token</code>	Token to access the data feed. The token can be obtained at <a href="http://realtime.nationalrail.co.uk/OpenLDBWSRegistration/">http://realtime.nationalrail.co.uk/OpenLDBWSRegistration/</a> .
<code>url</code>	Data feed source URL.
<code>verbose</code>	Boolean flag to indicate whether or not to show status messages.

### Value

Tibble with departure records. Each column is described below:

**generatedAt** The time at which the station board was generated.

**locationName** The name of the location that the station board is for.

**crs** The CRS code of the location that the station board is for.

**filterLocationName** If a filter was requested, the location name of the filter location.

**filtercrs** If a filter was requested, the CRS code of the filter location.

**filterType** If a filter was requested, the type of filter.

**nrccMessages** An optional list of textual messages that should be displayed with the station board. The message may include embedded and XML encoded HTML-like hyperlinks and paragraphs. The messages are typically used to display important disruption information that applies to the location that the station board was for. Any embedded `<p>` tags are used to force a new-line in the output. Embedded `<a>` tags allow links to external web pages that may provide more information. Output channels that do not support HTML should strip out the `<a>` tags and just leave the enclosed text.

**platformAvailable** An optional value that indicates if platform information is available. If this value is present with the value "true" then platform information will be returned in the service lists. If this value is not present, or has the value "false", then the platform "heading" should be suppressed in the user interface for this station board.

**areServicesAvailable** An optional value that indicates if services are currently available for this station board. If this value is present with the value "false" then no services will be returned in the service lists. This value may be set, for example, if access to a station has been closed to the public at short notice, even though the scheduled services are still running. It would be usual in such cases for one of the nrccMessages to describe why the list of services has been suppressed.

**trainServices** A list of ServiceItem object for each services of the relevant type that is to appear on the station board. May contain zero items, or may not be present at all.

**busServices** A list of ServiceItem object for each services of the relevant type that is to appear on the station board. May contain zero items, or may not be present at all.

**ferryServices** A list of ServiceItem object for each services of the relevant type that is to appear on the station board. May contain zero items, or may not be present at all.

### Source

Documentation for the Live Departure Boards Web Service (LDBWS / OpenLDBWS): <http://lite.realtime.nationalrail.co.uk/openldbws/>

### See Also

Other OpenLDBSVWS requests: [GetArrBoardRequest\(\)](#), [GetArrBoardWithDetailsRequest\(\)](#), [GetArrDepBoardWithDetailsRequest\(\)](#), [GetDepBoardRequest\(\)](#), [GetServiceDetailsRequest\(\)](#)

### Examples

```
## Not run:
rdg_dep <- trainR::GetDepBoardWithDetailsRequest("RDG")
rdg_dep <- trainR::GetDepBoardWithDetailsRequest("RDG", filterCrs = "BRI")
trainR::print(rdg_dep)

## End(Not run)
```

---

GetServiceDetailsRequest  
*Get service details*

---

### Description

Get the service details for a specific service identified by a station board. These details are supplied relative to the station board from which the serviceID field value was generated. Service details are only available while the service appears on the station board from which it was obtained. This is normally for two minutes after it is expected to have departed, or after a terminal arrival. If a request is made for a service that is no longer available then a NULL value is returned.

**Usage**

```
GetServiceDetailsRequest(
  serviceID,
  token = get_token(),
  url = "https://lite.realtime.nationalrail.co.uk/OpenLDBWS/ldb11.aspx",
  verbose = FALSE
)
```

**Arguments**

serviceID	(string): The LDBWS service ID of the service to request the details of. The service ID is obtained from a service listed in a StationBoard object returned from any other request.
token	Token to access the data feed. The token can be obtained at <a href="http://realtime.nationalrail.co.uk/OpenLDBWSRegistration/">http://realtime.nationalrail.co.uk/OpenLDBWSRegistration/</a> .
url	Data feed source URL.
verbose	Boolean flag to indicate whether or not to show status messages.

**Value**

Tibble with departure records.

**See Also**

Other OpenLDBSVWS requests: [GetArrBoardRequest\(\)](#), [GetArrBoardWithDetailsRequest\(\)](#), [GetArrDepBoardWithDetailsRequest\(\)](#), [GetDepBoardRequest\(\)](#), [GetDepBoardWithDetailsRequest\(\)](#)

---

get_crs	<i>Get CRS code</i>
---------	---------------------

---

**Description**

Get CRS code from a string with the station name.

**Usage**

```
get_crs(station, ...)
```

**Arguments**

station	String with the station name, partial names are allowed.
...	Arguments passed on to <a href="#">base::grep</a>
	ignore.case if FALSE, the pattern matching is <i>case sensitive</i> and if TRUE, case is ignored during matching.
	perl logical. Should Perl-compatible regexps be used?

`value` if FALSE, a vector containing the (integer) indices of the matches determined by `grep` is returned, and if TRUE, a vector containing the matching elements themselves is returned.

`fixed` logical. If TRUE, `pattern` is a string to be matched as is. Overrides all conflicting arguments.

`useBytes` logical. If TRUE the matching is done byte-by-byte rather than character-by-character. See 'Details'.

`invert` logical. If TRUE return indices or values for elements that do *not* match.

### Value

Vector of strings with the matched CRS codes, or NULL if no match is found.

### Examples

```
trainR::get_crs("London")
trainR::get_crs("Manchester")
trainR::get_crs("manchester", ignore.case = TRUE)
```

---

`get_token`

*Get user's token*

---

### Description

Get user's token to access the National Rail Enquiries (NRE) data feeds.

### Usage

```
get_token(ENV = "NRE")
```

### Arguments

`ENV` String with environment variable containing the token to access the NRE data feeds (default = "NRE").

### Value

String with token.

---

set_token	<i>Configure user's token</i>
-----------	-------------------------------

---

**Description**

Configure user's token to access the National Rail Enquiries (NRE) data feeds.

**Usage**

```
set_token()
```

**Details**

To obtain an access token, you must complete the registration form found at <http://realtime.nationalrail.co.uk/OpenLDBWSRegistration/>.

**Value**

Nothing, helper function to set up environment variable.

---

station_codes	<i>National Rail Enquiries (NRE) Station Codes dataset</i>
---------------	--

---

**Description**

A dataset containing information of 2579 British National Rail Enquiries (NRE) station 3 letter alpha (CRS) codes.

**Usage**

```
data(station_codes)
```

**Format**

A data frame with 2579 rows and 2 variables:

**name** Station name.

**crs** Station code.

**Author(s)**

National Rail Enquiries (NRE)

**Source**

[https://www.nationalrail.co.uk/stations\\_destinations/48541.aspx](https://www.nationalrail.co.uk/stations_destinations/48541.aspx)

---

store	<i>Store object to disk</i>
-------	-----------------------------

---

## Description

### Experimental

## Usage

```
store(x, ...)
```

```
## S3 method for class 'StationBoard'
```

```
store(x, ..., file)
```

```
## S3 method for class 'busServices'
```

```
store(x, ..., wb)
```

```
## S3 method for class 'ferryServices'
```

```
store(x, ..., wb)
```

```
## S3 method for class 'trainServices'
```

```
store(x, ..., wb)
```

```
## S3 method for class 'CallingPoints'
```

```
store(x, ..., wb, serviceID)
```

```
## S3 method for class 'previousCallingPoints'
```

```
store(x, ..., wb, serviceID)
```

```
## S3 method for class 'subsequentCallingPoints'
```

```
store(x, ..., wb, serviceID)
```

## Arguments

x	Data.
...	Optional parameters.
file	String with filename and path.
wb	Workbook object, created with <a href="#">openxlsx::createWorkbook</a> .
serviceID	(string): The LDBWS service ID of the service to request the details of. The service ID is obtained from a service listed in a <code>StationBoard</code> object returned from any other request.

## Details

The goal of this generic is to facilitate storing data retrieved from arrival and departure boards. Each request has multiple parts that should be stored individually (e.g. Excel Workbook or individual CSV files).



**Value**

Nothing, call for its side effect.

**Examples**

```
## Not run:
`%>%` <- magrittr::`%>%`
pad <- trainR::GetArrBoardWithDetailsRequest("PAD")
pad %>%
  trainR::store(file = "arrivals-PAD.xlsx")

# Delete test file
unlink("arrivals-PAD.xlsx")

## End(Not run)
```

# Index

## \* **OpenLDBSVWS requests**

- GetArrBoardRequest, [2](#)
- GetArrBoardWithDetailsRequest, [4](#)
- GetArrDepBoardWithDetailsRequest,  
[6](#)
- GetDepBoardRequest, [8](#)
- GetDepBoardWithDetailsRequest, [10](#)
- GetServiceDetailsRequest, [12](#)

## \* **datasets**

- station\_codes, [15](#)

base::grep, [13](#)

get\_crs, [13](#)

get\_token, [14](#)

GetArrBoardRequest, [2](#), [6](#), [8](#), [10](#), [12](#), [13](#)

GetArrBoardWithDetailsRequest, [4](#), [4](#), [8](#),  
[10](#), [12](#), [13](#)

GetArrDepBoardWithDetailsRequest, [4](#), [6](#),  
[6](#), [10](#), [12](#), [13](#)

GetDepBoardRequest, [4](#), [6](#), [8](#), [8](#), [12](#), [13](#)

GetDepBoardWithDetailsRequest, [4](#), [6](#), [8](#),  
[10](#), [10](#), [13](#)

GetServiceDetailsRequest, [4](#), [6](#), [8](#), [10](#), [12](#),  
[12](#)

openxlsx::createWorkbook, [16](#)

set\_token, [15](#)

station\_codes, [15](#)

store, [16](#)