

Fares Feed Data - Fare Lookup

Linus Norton

Contents

Query Structure	3
Lingua Franca	4
Query Locations	6
Fare Querying	7
Flow Fares	7
Railcard Application	7
Passenger Set	8
Railcard restrictions	8
Railcard geography	8
Applying the railcard	8
Non Derivable Fares	9
Merging the Results	11
Season Tickets	12
7 day seasons	12
>= 1 month seasons	12
First class seasons	13
Fares Query Process	14

Fare Lookup

The high level process of calculating a fare is to work out the stations involved in the query, then check for relevant flow fares and non-derivable fares. There are several pieces of business logic that need to be applied along the way in order to correctly apply discounts, offer fares on the correct date and select the correct fare to return where multiple are available.

This document covers how to use the data in the fares feed rather than the specifics of the data structure, as that is defined in [RSPS5045](#).

There are additional types of fares that are not covered by this document, namely: carnets, packages, rangers and rovers.

Query Structure

The minimum set of inputs required for a fare query are:

- origin (CRS code or NLC code)
- destination (CRS code or NLC code)
- date
- passengers
 - passenger type (Adult | Child)
 - railcard (Railcard code, optional)

Lingua Franca

There are many overlapping terms and concepts in the rail industry. For the purpose of this document it can be assumed the following words or phrases have the meaning defined here.

Fare

.A conceptual price for a ticket comprised of an origin, destination, route code, status code, restriction code and ticket type code.

Ticket

An instance of a fare that has concrete dates as opposed to a date period or duration. Once a ticket has been created it is a passenger's right to travel.

Coupon

A physical representation of a ticket (the thing you hold). In the case of CCST tickets, one ticket may product two coupons. One coupon for the outward portion and one for the return portion.

Ticket Type

Each fare has a ticket type, denoted by a three character ticket type code. A ticket type code will in turn specify a two digit validity code that contains the validity period information.

The ticket type code will specify contain a number of properties common to any fare with the ticket type code, such as min/max passengers, whether or not a reservation is required and the quote date.

The ticket type code is sometimes referred to as the FTOT (fares type of ticket) as opposed to the LTOT (LENNON type of ticket) or capri code.

Location

A location may be a physical station or a logical location used for fare setting. Locations may be used to define origins or destinations that only exist for fare setting purposes (i.e PlusBus stations), group stations which contain several physical stations as members or a cluster may contain several other locations for the purpose of mapping fares. Identified by a 4 character NLC code.

Physical station

A station that a journey may stop at or pass through. Identified by a three character CRS code.

Flow Fare

A fare sourced from the flow and fare records in the fares feed. A flow fare is always returned as the adult price (status 000).

Non Derivable Fare

Non-derivables are fares that cannot be derived from a flow. This means that the normal application of railcards does not apply and the non-derivable fares contain a specific price for certain railcards or even no railcard.

Status

Every passenger has a status code which may depend on their type (adult = 000, child = 001) or railcard that they possess. A status code may entitle a discount that can be applied to the base fare.

Query Locations

There are roughly 20,000 locations in the feed and in order to minimize the amount of data in the flow files stations and group stations are sometimes organised into station clusters for the purpose of a fare lookup..

When looking up a fare the first step is to create a list of origins and destinations that might be relevant to the query. The relevant stations are a set of NLCs comprised of:

- input NLC
- any clusters that input NLC belongs to
- any group stations the input NLC belongs to
- any cluster stations the group stations belong to

During a fares query it is assumed that the London Terminals Mapping does not apply and that fares between invalid combinations of station and London terminal do not exist. It is still possible that this will result in fares being returned that will not work on any journey.

For example, a fare between Tonbridge and London Kings Cross does not exist but London Kings Cross is a member of London Terminals where a fare does exist. Validating the whether the fare can be used on a journey will fail on every journey because London Kings Cross is not a London Terminal for Tonbridge according to the London Terminals Mapping.

Relevant data:

- LOC (location, group location, group members, clusters)

Fare Querying

Fares are split into two groups; flow fares and non-derivable fares (NDFs). Flow fares are considered the norm (sometimes referred to as the base fare) but there a greater number of NDFs. Since the introduction of PMS all NDFs are now defined in the non-derivable fare override (NFO) file.

When looking up either type of fare the target date of the query (the input date) must be between the start and end date of the record and the date the query is performed (the current date) must be on or after the quote date of the record.

Flow Fares

To look up a flow fare, return any fare records that have a flow between any of the origin and destination NLCs, or the NLCs of group stations and station clusters the origin and destination belong to.

Some flows have the reversible flag set, in which case the fare is the same in the other direction and the origin and destination can be swapped.

If you are using a flow that has been found by reversing the origin and destination you should switch the origin and destination of the flow record so that it matches the input query.

If the flow origin and destination is a cluster it should be mapped back to the cluster member NLC of the origin or destination.

Relevant data:

- FFL (fare, flow)

Railcard Application

Railcards can be applied to a fare during the fare look up or once the railcards validity on a journey has been determined. The advantage of applying it before checking validity is that it reduces the amount of checking required for non-derivable fares that would override the flow fare (covered later).

Before applying the railcard there are a few checks that need to be performed:

Passenger Set

Railcards have a number of min and max fields (quantity of adults, children or total passengers). The passengers in the passenger set must be within the limits of the railcard. The data in the feed is largely incorrect but can be found online.

There are some types of railcard that the passenger does not need to physically hold and can be applied if they are valid for the passenger set. One example of this would be the GS3 or Group Save railcard that entitles a discount to any groups of three or more adults.

Railcard restrictions

Railcard restrictions may be a total ban on the use of the railcard on particular ticket codes, origins or routes, or can apply additional restrictions in the form of a restriction code.

If there is a restriction for the railcard it will either disallow its use or apply a minimum fare override depending on the restriction time min fare flag. The restrictions data is covered in more depth in the Fares Feed Data - Fare Validation document.

Multiple railcard restrictions can apply to a fare. One may have a restriction code and one may ban its use for a specific ticket type or at a specific location.

Railcard geography

There are geographical rules that apply to some railcards (such as the Network Gold railcard). If the railcard "is restricted by area" flag is set there should be corresponding railcard locations inside the locations file. The origin and destination of the fare must be within the railcard area in order for it to be applied.

Applying the railcard

Assuming the railcard can be applied, the ticket types discount category can be matched to a status discount record for the status code(s) of the railcard. Each status discount has a discount indicator, which when combined with the status record controls the logic used to discount the fare. This logic is described section 4.17.3 of [RSPS5045](#).

Some discount statuses have a discount indicator of X or N which means they cannot be applied.

Once you have applied the discount the correct rounding rules should be applied. There is data in the fares feed for rounding rules but this should be ignored as it has been superseded by the logic described in Section 6 of RSPS7001.

The public railcard covers normal adult and child fares (status 000 and 001) and can be applied like any other fare.

If the flow has a non standard discount indicator then you cannot apply railcard discounts using the method above. Either the flow will be marked as not discountable or there will be an origin and destination that should be used to return another flow discount to be used as an "add on" to the original fare. Non standard discounts do not apply to status "000" (standard adult fares).

Relevant data:

- RLC (railcard)
- RCM (railcard minimum fare)
- DIS (status, status discount)
- TTY (ticket type)

Non Derivable Fares

To look up an NDF, match the origin and destination and any railcards that are relevant to your query in the non-derivable fare override file.

Non derivable fares may be overridden other non-derivable fares later in the file (this works in a similar manner to the schedules in the timetable feed). The matching criteria for fares is origin, destination, route code, ticket type, status (or railcard), If relying on the ordering records in the file is inconvenient sorting them by start date should have the same effect.

Suppression records are used to remove other non-derivable fares for part or all of their duration. Since the introduction of PMS there are not any records with the suppress marker set.

Records with either an adult price or child price of 999999 indicate that the fare is not retailable and any flow fares that share the same matching criteria (defined above) should also be removed.

NDF records with a composite record of N can be ignored as they are also in the flow file rendering them useless.

It's possible railcard restrictions also need to be applied to NDFs but as the fares have been added manually this should not be the case.

Note that unlike flow fares there is no fare setter field for non-derivable fares and the cross London indicator is a boolean Y/N rather than an integer field. A N is taken to be equivalent 0 (not valid) and Y is equivalent to 1 (valid via underground only).

Relevant data:

- NFO (non derivable fare override)

Merging the Results

As noted above, non-derivable fares take precedence over flow fares. When creating the combined set of results, any flow fare with an origin, destination, route code, ticket code and railcard that matches a non-derivable should be overridden by the non-derivable fare.

It is important not to discard any non-derivable fare records with an adult price or child price of 999999 until after the flow fares have been merged in as they may override (suppress) some flow fares.

After the result set has been created the calendar restriction records should be checked to remove any fares that have a matching ticket type code and route code (if specified in the calendar restriction).

Some ticket types are classified as Advance tickets and must be purchased in advance of travel. The TAP file contains the advance ticket booking horizon the must be respected for each ticket type.

If the restriction flag of record is 0 the ticket type code and restriction code of the fare must match, if it is 2 only the ticket type code needs to match.

There are some rules that don't currently need to be implemented as there is no data for them. For instance the `AP_DATA` field currently always 2 indicated a specific number of days in advance are required.

Relevant data:

- RST (ticket calendar restrictions)
- TAP (advance tickets)

Season Tickets

To look up a season ticket perform the normal fare query process with a filter that only returns fares where the ticket type code has a tkt_type of N (as opposed to S or R).

All season ticket ticket type codes have a validity type of 00 (USE SEASON) which means that it must be set based on the input query.

Railcards typically do not apply to season tickets but there are exceptions. Status discounts can be applied like any other fare and there are non derivable fare overrides to set specific values for child/railcard fares.

7 day seasons

7 day fares can be returned without further processing. There is no data to identify which ticket type codes are 7 day ticket types but they typically start with a 7, e.g. 7DS, 7DF, 7TS, 7TF.

>= 1 month seasons

The fares feed only contains pricing for 7 day season tickets, all other durations must be calculated by taking the PSS/PSF fare (or 7DS/7DF if the PSS/PSF does not exist) and applying a formula*:

base fare * 3.84 per month
+ base fare * 0.64 for every 5 extra days
+ base fare * 0.13 for every extra day over a multiple of 5

Annual tickets are capped at 10 months and 13 days.

The fare should then rounded in one of two ways:

- Adults fares: up to the nearest 10p
- Children fares: down to the nearest penny then up to the nearest 5p

If a status code discount going to be applied the rounding should take place at the very end (after the discount and after the formula have been applied).

* Note this formula came from a [forum post](#) in 1999 the actual process is to use a lookup table defined in Knowledgebase but the method above is easier to apply programmatically.

First class seasons

Where a fare for a first class season does not exist but a standard class season does exist the fare multiplier can be used to calculate the price of the first class ticket. It appears that all flows now have a first class fare and this method is not used any more.

Relevant data:

- TTY (ticket types)
- FFL (flow fares)
- NDO (non-derivable fares)

Fares Query Process

