Transperth Documentation

Release v0.0.2

Dominic May

June 19, 2014

Contents

CHAPTER 1

Quickstart

This is an API for Transperth's Journey Planner, though extensions are likely.

The basic workflow is as such;

```
import os
import sys
sys.path.insert(
    0,
    os.path.join(os.path.dirname(__file__), '..', '..')
)


# create the Location object you wish to resolve;
from transperth.jp.location import (
    Location,
    ResolvedLocation,
    determine_location
)

from_location = Location.from_location('Curtin University, Perth')
to_location = Location.from_location('Arena Joondalup')


# then we resolve it into something that the transperth api will accept
locations = determine_location(from_location, to_location)

# determine_location will return a dictionary like so;
{
    '<DIRECTION>': [
        ResolvedLocation('<NAME>', '<CODE>'),
        # etc
    ]
}

# it would be reasonable to assume the first result is correct,
# or to let the end user choose from a list
from_location = locations['from'][0]
to_location = locations['to'][0]

# once we have these, we can grab the routes
from transperth.jp.routes import determine_routes

routes = determine_routes(from_location, to_location)
```

```
# take your pick of the routes
route = routes[0]

# and use 'em how you like
from transperth.smart_rider.trips import timedelta_repr
print(timedelta_repr(route['meta']['duration']))
```

CHAPTER 2

exceptions Module

exception transperth.exceptions.**BadStationError**
Bases: transperth.exceptions.TransperthException

Thrown when the library encounters a bad station name

exception transperth.exceptions.**IncompleteTrip**
Bases: transperth.exceptions.TransperthException

Thrown when a trip is missing its conclusion

exception transperth.exceptions.**InvalidDirection**
Bases: transperth.exceptions.TransperthException

Thrown when the provided direction is neither to nor from

exception transperth.exceptions.**InvalidStep**
Bases: transperth.exceptions.TransperthException

Thrown when a step is not one of 'bus', 'walk', or 'train'

exception transperth.exceptions.**InvalidStopNumber**
Bases: transperth.exceptions.TransperthException

Thrown when the provided stop number is not a five digit number

exception transperth.exceptions.**LoginFailed**
Bases: transperth.exceptions.NotLoggedIn

Thrown when a login attempt fails

exception transperth.exceptions.**NoFareData**
Bases: transperth.exceptions.TransperthException

Thrown when transperth does not provide data from which we can assertain the fare for a route

exception transperth.exceptions.**NotLoggedIn**
Bases: transperth.exceptions.TransperthException

Raised when the library detects that the session has expired

exception transperth.exceptions.**TransperthException**
Bases: builtins.Exception

Base exception for all exceptions thrown by this library

CHAPTER 3

`livetimes` Module

transperth.livetimes.**times_for_station**(*station_name*)

Given a station name (from TRAIN_STATIONS_SET) return the associated incoming train timings

CHAPTER 4

jp Package

4.1 fares Module

4.2 location Module

4.3 route_parser Module

4.4 routes Module

4.5 utils Module

CHAPTER 5

smart_rider Package

5.1 smart_rider Module

5.2 post_back Module

5.3 trips Module

Parses actions into separate trips

class transperth.smart_rider.trips.**TripTracer**(*actions*)

Bases: builtins.object

consume_trip(*self*) → list

Consumes a single trip, bar the first step

determine_path(*self*, *trip:list*) → list

Iterates through the steps in the trip, return a list of locations that were visited, in order, during the trip, along with the service used to travel there

determine_price(*self*, *trip*)

Determines the total price for the given trip

generate_meta(*self*, *trip:list*) → dict

Computes the following;

- from location (for the trip)
- to location (for the trip)
- trip duration
- wait time between steps. this is zero on single step trips
- total price for the trip

Returns metadata for a trip

grab_step(*self*) → dict

Returns a dictionary representing the step

trace(*self*)

Consumes the provided actions, yielding trips consisting of stepss

transperth.smart_rider.trips.**determine_breadth**(*iterable*)
Determines the breadth of the addable contents of the iterable

transperth.smart_rider.trips.**determine_trips**(*actions:list*) → list
Analysis's the given actions, and groups them into trips with appropriate metadata

Returns a list of trips

transperth.smart_rider.trips.**determine_wait_time**(*trip:list*) → datetime.timedelta

transperth.smart_rider.trips.**timedelta_repr**(*td:datetime.timedelta*) → str

Returns a human readable representation of the provided timedelta object

CHAPTER 6

Indices and tables

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

t