BusBuddy

Generated by Doxygen 1.8.3.1

Fri Apr 19 2013 23:05:55

Contents

1 Namespace Index 1

1.1 Packages . . . 1

2 Hierarchical Index 1

2.1 Class Hierarchy . . . 1

3 Class Index 3

3.1 Class List . . . 3

4 Namespace Documentation 5

4.1 Package alert . . . 5

4.1.1 Detailed Description . . . 5

4.2 Package tracking . . . 5

4.2.1 Detailed Description . . . 6

4.3 Package transit . . . 6

4.3.1 Detailed Description . . . 7

4.4 Package user . . . 7

4.4.1 Detailed Description . . . 7

5 Class Documentation 7

5.1 AbstractFeedParserTemplate Class Reference . . . 7

5.1.1 Detailed Description . . . 8

5.1.2 Member Function Documentation . . . 9

5.2 ITrackingService.AlertType Enum Reference . . . 10

5.3 ATransitVehicleFactory Class Reference . . . 10

5.4 BusVehicle Class Reference . . . 10

5.5 CommercialTracking Class Reference . . . 11

5.5.1 Detailed Description . . . 11

5.5.2 Constructor & Destructor Documentation . . . 12

5.6 CommercialTracking.CommercialTrackingHolder Class Reference . . . 12

5.6.1 Detailed Description . . . 12

5.7 Detour Class Reference . . . 12

5.7.1 Detailed Description . . . 12

5.7.2 Member Data Documentation . . . 13

5.8 Fare Class Reference . . . 13

5.8.1 Detailed Description . . . 13

5.8.2 Member Function Documentation . . . 13

5.9 GoogleTransitServiceAdapter Class Reference . . . 13

5.9.1 Detailed Description . . . 14

5.9.2 Constructor & Destructor Documentation . . . 14

5.9.3 Member Function Documentation 14

5.10 GoogleTransitServiceAPI Interface Reference 15

5.10.1 Detailed Description 15

5.11 GPSLocationObserver Interface Reference 15

5.11.1 Detailed Description 15

5.11.2 Member Function Documentation 15

5.12 GPSLocationTracking Interface Reference 16

5.12.1 Detailed Description 16

5.12.2 Member Function Documentation 16

5.13 GPSPusher Class Reference 17

5.13.1 Detailed Description 17

5.13.2 Constructor & Destructor Documentation 18

5.14 GPSPusher.GPSPusherHolder Class Reference 18

5.14.1 Detailed Description 18

5.15 GPSVehicleTracker Class Reference 18

5.15.1 Detailed Description 18

5.15.2 Constructor & Destructor Documentation 19

5.16 GTFSFeedParser Class Reference 19

5.16.1 Detailed Description 19

5.16.2 Member Function Documentation 19

5.17 IAlertService Interface Reference 19

5.18 InvalidRouteParseException Class Reference 19

5.18.1 Detailed Description 20

5.18.2 Constructor & Destructor Documentation 20

5.18.3 Member Data Documentation 20

5.19 ITrackingService Interface Reference 21

5.20 JSONResponseRenderer Class Reference 21

5.21 Location Class Reference 21

5.21.1 Detailed Description 22

5.21.2 Constructor & Destructor Documentation 22

5.22 ResponseRenderer Interface Reference 22

5.23 Route Class Reference 22

5.23.1 Detailed Description 23

5.23.2 Member Data Documentation 23

5.24 RouteDisruptionAlert Class Reference 23

5.24.1 Detailed Description 23

5.25 RouteRepository Interface Reference 23

5.25.1 Detailed Description 24

5.25.2 Member Function Documentation 24

5.26 RouteSpecification Class Reference 25

5.26.1 Detailed Description 25
5.26.2 Member Function Documentation 26
5.27 Specification< T > Interface Reference 26
5.27.1 Detailed Description 26
5.27.2 Member Function Documentation 26
5.28 Stop Class Reference 27
5.28.1 Detailed Description 28
5.28.2 Member Function Documentation 28
5.29 TrackingAlert Class Reference 28
5.30 TrackingAlertObserver Class Reference 29
5.31 TrackingDelayAlert Class Reference 29
5.32 TrackingLocationAlert Class Reference 29
5.32.1 Constructor & Destructor Documentation 30
5.33 TrackingService Class Reference 30
5.34 TrackingServiceController Class Reference 30
5.34.1 Member Function Documentation 31
5.35 TransitFeed Interface Reference 31
5.35.1 Detailed Description 31
5.36 TransitInfo Class Reference 31
5.36.1 Detailed Description 31
5.37 TransitProvider Class Reference 31
5.37.1 Detailed Description 32
5.37.2 Member Data Documentation 32
5.38 TransitService Interface Reference 32
5.38.1 Detailed Description 33
5.38.2 Member Function Documentation 33
5.39 TransitVehicle Class Reference 33
5.39.1 Detailed Description 34
5.40 TransitVehicleFactory Class Reference 34
5.40.1 Member Function Documentation 34
5.41 Trip Class Reference 35
5.41.1 Detailed Description 35
5.42 TripService Interface Reference 35
5.42.1 Detailed Description 35
5.42.2 Member Function Documentation 35
5.43 User Class Reference 36
5.44 UserLoginService Interface Reference 36
5.45 UserManagementService Interface Reference 36
5.46 VehicleObject Class Reference 36
5.47 vehicleRepository Class Reference 36

5.47.1 Detailed Description 36

Index 36

1 Namespace Index

1.1 Packages

Here are the packages with brief descriptions (if available):

alert **The Alert Module**	5
tracking **The Tracking Module**	5
transit **The Transit Module is an interface to get Route/Fare/Detour information from a TransitProvider**	6
user **The User Module**	7

2 Hierarchical Index

2.1 Class Hierarchy

This inheritance list is sorted roughly, but not completely, alphabetically:

AbstractFeedParserTemplate	7
GTFSFeedParser	19
ITrackingService.AlertType	10
ATransitVehicleFactory	10
TransitVehicleFactory	34
CommercialTracking.CommercialTrackingHolder	12
Detour	12
Exception	
InvalidRouteParseException	19
Fare	13
GoogleTransitServiceAPI	15
GPSLocationObserver	15
GPSVehicleTracker	18
GPSLocationTracking	16
CommercialTracking	11

- GPSPusher 17
- GPSPusher.GPSPusherHolder 18
- IAlertService 19
- ITrackingService 21
  - TrackingService 30
- Location 21
- ResponseRenderer 22
  - JSONResponseRenderer 21
- Route 22
- RouteDisruptionAlert 23
- RouteRepository 23
- Specification< T > 26
- Stop 27
- TrackingAlert 28
- TrackingAlertObserver 29
  - TrackingDelayAlert 29
  - TrackingLocationAlert 29
- TrackingServiceController 30
- TransitFeed 31
- TransitInfo 31
- TransitProvider 31
- TransitService 32
  - GoogleTransitServiceAdapter 13
- TransitVehicle 33
  - BusVehicle 10
- Trip 35
- TripService 35
- User 36
- UserLoginService 36
- UserManagementService 36
- VehicleObject 36

vehicleRepository 36
Specification

RouteSpecification 25

3 Class Index

3.1 Class List

Here are the classes, structs, unions and interfaces with brief descriptions:

AbstractFeedParserTemplate
A Template Method pattern to allow for the import of data from different TransitProviders in potentially different formats 7

ITrackingService.AlertType 10

ATransitVehicleFactory 10

BusVehicle 10

CommercialTracking
Implements Subject Location Tracking for retrieving GPS location updates from outside commercial tracking services 11

CommercialTracking.CommercialTrackingHolder
Commercial Tracking Holder is loaded on the first execution of CommercialTracking.getInstance() or the first access to CommercialTracking.INSTANCE, not before (lazy instantiation) 12

Detour
A disruption in service due to an unexpected event 12

Fare
An immutable Value Object representing the cost, or 'fare,' required to ride a TransitVehicle on a particular Route 13

GoogleTransitServiceAdapter
An Adapter Class to allow a {} service to appear as a TransitService 13

GoogleTransitServiceAPI
A client to Google's `Maps API` 15

GPSLocationObserver
Observer Pattern - Observer interface for GPS location tracking 15

GPSLocationTracking
GPSLocationTracking - interface Subject of the Observer Pattern Defines methods for an observer GPS Device to register and receive updates on vehicle location 16

GPSPusher
Implements Subject Location Tracking for retrieving GPS location updates from outside commercial tracking services 17

GPSPusher.GPSPusherHolder
GPS Pusher Holder is loaded on the first execution of GPSPusher.getInstance() or the first access to GPSPusher.INSTANCE, not before (lazy instantiation) 18

GPSVehicleTracker
Implementation of the Observer, update the transit vehicle GPS location 18

GTFSFeedParser
A AbstractFeedParserTemplate implementation designed to parse GTFS format ZIP files into Routes 19

IAlertService 19

InvalidRouteParseException
An InvalidRouteParseException indicates an invalid batch of parsed Routes has been been detected 19

ITrackingService 21

JSONResponseRenderer 21

Location
An immutable Value Object representing a physical point on the geographic coordinate system 21

ResponseRenderer 22

Route
A Route is a TransitVehicle path of travel, or a "Line," as referred to by a TransitProvider 22

RouteDisruptionAlert
TODO: 23

RouteRepository
A Repository Pattern supporting lifecycle operations of Routes, such as Read, Save, Delete, and Query functionality 23

RouteSpecification
A Specification Pattern class for validating a Route 25

Specification< T >
A Generic Specification to be used for chaining business validation rules together 26

Stop
A point on a Route in which a TransitVehicle will stop to pick up and drop off passengers 27

TrackingAlert 28

TrackingAlertObserver 29

TrackingDelayAlert 29

TrackingLocationAlert 29

TrackingService 30

TrackingServiceController 30

TransitFeed
TODO: 31

TransitInfo
TODO: 31

TransitProvider
A TransitProvider is a description of a company or organization that is the producer of public transportation services 31

TransitService
The TransitService is an interface to get Route/Fare/Detour information from a TransitProvider 32

TransitVehicle
Abstract transit vehicle class contains the common data for all types of vehicles and the Subject GPS Tracking and the GPS observer to receive GPS location updates 33

TransitVehicleFactory 34

Trip
A Trip is considered an ordered collection of Routes going from a starting point to an ending point 35

TripService
A Service to calculate a collection of Routes, or a Trip, allowing for a continuous transit path from a start Location to an end Location 35

User 36

UserLoginService 36

UserManagementService 36

VehicleObject 36

vehicleRepository
Repository for information on vehicles registered on a route 36

4 Namespace Documentation

4.1 Package alert

The Alert Module.

Classes

- interface IAlertService

4.1.1 Detailed Description

The Alert Module.

4.2 Package tracking

The Tracking Module.

Classes

- class ATransitVehicleFactory
- class BusVehicle
- class CommercialTracking

  Implements Subject Location Tracking for retrieving GPS location updates from outside commercial tracking services.
- interface GPSLocationObserver

  Observer Pattern - Observer interface for GPS location tracking.
- interface GPSLocationTracking

  GPSLocationTracking - interface Subject of the Observer Pattern Defines methods for an observer GPS Device to register and receive updates on vehicle location.

- class GPSPusher

  Implements Subject Location Tracking for retrieving GPS location updates from outside commercial tracking services.
- class GPSVehicleTracker

  Implementation of the Observer, update the transit vehicle GPS location.
- interface ITrackingService
- class TrackingAlert
- class TrackingAlertObserver
- class TrackingDelayAlert
- class TrackingLocationAlert
- class TrackingService
- class TrackingServiceController
- class TransitVehicle

  Abstract transit vehicle class contains the common data for all types of vehicles and the Subject GPS Tracking and the GPS observer to receive GPS location updates.
- class TransitVehicleFactory
- class VehicleObject
- class vehicleRepository

  Repository for information on vehicles registered on a route.

4.2.1 Detailed Description

The Tracking Module.

4.3 Package transit

The Transit Module is an interface to get Route/Fare/Detour information from a TransitProvider.

Classes

- class AbstractFeedParserTemplate

  A Template Method pattern to allow for the import of data from different TransitProviders in potentially different formats.
- class Detour

  A disruption in service due to an unexpected event.
- class Fare

  An immutable Value Object representing the cost, or 'fare,' required to ride a TransitVehicle on a particular Route.
- class GoogleTransitServiceAdapter

  An Adapter Class to allow a {} service to appear as a TransitService.
- interface GoogleTransitServiceAPI

  A client to Google's `Maps API`.
- class GTFSFeedParser

  A AbstractFeedParserTemplate implementation designed to parse `GTFS` *format ZIP files into Routes.*
- class InvalidRouteParseException

  An InvalidRouteParseException indicates an invalid batch of parsed Routes has been been detected.
- class JSONResponseRenderer
- class Location

  An immutable Value Object representing a physical point on the geographic coordinate system.
- interface ResponseRenderer
- class Route

  A Route is a TransitVehicle path of travel, or a "Line," as referred to by a TransitProvider.
- class RouteDisruptionAlert

  TODO:

- interface RouteRepository

    A Repository Pattern supporting lifecycle operations of Routes, such as Read, Save, Delete, and Query functionality.
- class RouteSpecification

    A Specification Pattern class for validating a Route.
- class Stop

    A point on a Route in which a TransitVehicle will stop to pick up and drop off passengers.
- interface TransitFeed

    TODO:
- class TransitInfo

    TODO:
- class TransitProvider

    A TransitProvider is a description of a company or organization that is the producer of public transportation services.
- interface TransitService

    The TransitService is an interface to get Route/Fare/Detour information from a TransitProvider.
- class Trip

    A Trip is considered an ordered collection of Routes going from a starting point to an ending point.
- interface TripService

    A Service to calculate a collection of Routes, or a Trip, allowing for a continuous transit path from a start Location to an end Location.

4.3.1 Detailed Description

The Transit Module is an interface to get Route/Fare/Detour information from a TransitProvider. The main module interface, the TransitService, provides a consistent interface for the application logic to query for this information.

From a design perspective, there are two main tasks performed by the Transit Module: Consuming Transit Information from a TransitProvider, and Providing Transit Information to Users.

4.4 Package user

The User Module.

Classes

- class User
- interface UserLoginService
- interface UserManagementService

4.4.1 Detailed Description

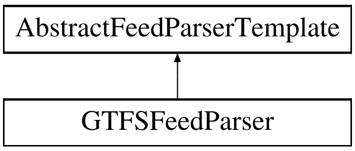
The User Module.

5 Class Documentation

5.1 AbstractFeedParserTemplate Class Reference

A Template Method pattern to allow for the import of data from different TransitProviders in potentially different formats.

Inheritance diagram for AbstractFeedParserTemplate:



Public Member Functions

- RouteRepository **getRouteRepository** ()
- void **setRouteRepository** (RouteRepository routeRepository)
- Specification< Route > **getRouteSpecification** ()
- void **setRouteSpecification** (Specification< Route > routeSpecification)

Protected Member Functions

- void start (URL location) throws InvalidRouteParseException

  The start method initiates the process and calls the appropriate methods in the appropriate order.
- InputStream loadFeed (URL location)

  Converts the resource URL into an InputStream for further processing.
- abstract Set< Route > parseFeed (InputStream feed)

  Parses the feed InputStream into a Set of Routes.
- boolean validate (Route route)

  Allow subclasses to validate Routes as they are parsed.
- void saveRoutes (Set< Route > routes)

  Save the Routes to the RouteRepository.

Private Attributes

- RouteRepository routeRepository

  The RouteRepository dependency allows for the persistence of the parsed Routes.
- Specification< Route > routeSpecification

  This Specification allows subclasses to validate Routes as they are parsed.

5.1.1 Detailed Description

A Template Method pattern to allow for the import of data from different TransitProviders in potentially different formats.

The algorithm sequence is as follows:

1. A URL of a resource location is passed into the start(URL) method. This method initiates the parsing/transformation process.
2. The start(URL) method calls the method loadFeed(URL) to establish the InputStream.
3. The InputStream returned by loadFeed(URL) is passed into the abstract parseFeed(InputStream) method. Subclasses will implement this as necessary to produce the resulting Routes
4. The newly created Routes are saved to the RouteRepository via the saveRoutes(Set<Route>) method.

5.1.2 Member Function Documentation

5.1.2.1 InputStream loadFeed (URL *location*) [protected]

Converts the resource URL into an InputStream for further processing.

Precondition

location exists and has been validated.

Parameters

location	The resource location

Returns

The resulting InputStream

5.1.2.2 abstract Set<Route> parseFeed (InputStream *feed*) [protected], [pure virtual]

Parses the feed InputStream into a Set of Routes.

Subclasses will implement this abstract method with the appropriate parsing logic for the particular input format.

Parameters

feed	The resource InputStream

Returns

The resulting Set of Routes

Implemented in GTFSFeedParser.

5.1.2.3 void saveRoutes (Set< Route > *routes*) [protected]

Save the Routes to the RouteRepository.

Precondition

routes may be an empty Set, but must not be null.

Parameters

routes	The Set of Routes to persist.

5.1.2.4 void start (URL *location*) throws InvalidRouteParseException [protected]

The start method initiates the process and calls the appropriate methods in the appropriate order.

Exceptions

InvalidRouteParseException	if any of the parsed Routes fail to validate via the given routeSpecification.

Parameters

location	The input data resource location. This may be a local file or a remote resource.

5.1.2.5 boolean validate (Route *route*) [protected]

Allow subclasses to validate Routes as they are parsed.

Subclasses are encouraged to use this method

Parameters

route	the route

Returns

true, if successful

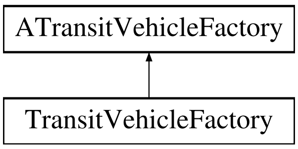
5.2 ITrackingService.AlertType Enum Reference

Public Attributes

- **LOCATION**
- **DELAY**

5.3 ATransitVehicleFactory Class Reference

Inheritance diagram for ATransitVehicleFactory:

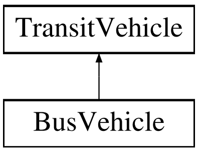


Public Member Functions

- abstract TransitVehicle **createTransitVehicle** (URL url, int vehicle)

5.4 BusVehicle Class Reference

Inheritance diagram for BusVehicle:



Public Member Functions

- void registerTrackingAlert (TrackingAlertObserver ao)

  Transit Vehicle is also the subject for tracking user subscribed alerts.
- void **unregisterTrackingAlert** (TrackingAlertObserver ao)
- void **triggerAlert** ()

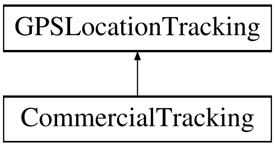
Private Attributes

- ArrayList< TrackingAlertObserver > **alertList**

5.5 CommercialTracking Class Reference

Implements Subject Location Tracking for retrieving GPS location updates from outside commercial tracking services.

Inheritance diagram for CommercialTracking:



Classes

- class CommercialTrackingHolder

  Commercial Tracking Holder is loaded on the first execution of CommercialTracking.getInstance() or the first access to CommercialTracking.INSTANCE, not before (lazy instantiation).

Public Member Functions

- void registerGPSDevice (GPSLocationObserver gpsObs)

  Register a GPS Device to the list to poll for updates.
- void unregisterGPSDevice (GPSLocationObserver gpsObs)

  Remove a GPS device from the list currently being polled for updates.
- void pollGPSDevice ()

  Continuously poll the registered GPS devices for location updates.

Static Public Member Functions

- static CommercialTracking **getInstance** ()

Private Member Functions

- CommercialTracking ()

  Only need one Commercial Tracking Service running to track by polling all registered GPS devices.

Private Attributes

- ArrayList< GPSLocationObserver > gpsObserver

  Array list of GPS devices registered for updates.

5.5.1 Detailed Description

Implements Subject Location Tracking for retrieving GPS location updates from outside commercial tracking services.

5.5.2 Constructor & Destructor Documentation

5.5.2.1 CommercialTracking ( ) [private]

Only need one Commercial Tracking Service running to track by polling all registered GPS devices.

Constructor, creates ArrayList<GPSLocationObserver> to hold registered observers.

5.6 CommercialTracking.CommercialTrackingHolder Class Reference

Commercial Tracking Holder is loaded on the first execution of CommercialTracking.getInstance() or the first access to CommercialTracking.INSTANCE, not before (lazy instantiation).

Static Public Attributes

- static final CommercialTracking **INSTANCE** = new CommercialTracking()

5.6.1 Detailed Description

Commercial Tracking Holder is loaded on the first execution of CommercialTracking.getInstance() or the first access to CommercialTracking.INSTANCE, not before (lazy instantiation).

5.7 Detour Class Reference

A disruption in service due to an unexpected event.

Public Member Functions

- String **getCause** ()
- void **setCause** (String cause)
- int **getEstimatedDelay** ()
- void **setEstimatedDelay** (int estimatedDelay)
- Set< Stop > **getAffectedStops** ()
- void **setAffectedStops** (Set< Stop > affectedStops)

Private Attributes

- String cause

  A text-based description of the cause of the Detour, intended to be displayed to customers.
- int estimatedDelay

  The estimated time (in minutes) that each of the Stops in the affectedStops will be delayed.
- Set< Stop > affectedStops

  All Stops that are subject to the noted estimatedDelay.

5.7.1 Detailed Description

A disruption in service due to an unexpected event.

A Detour may not affect all Stops in a Route, as a Detour may only alter portions of the Route. Any affected Stop will be listed in the affectedStops attribute.

5.7.2 Member Data Documentation

5.7.2.1 String cause [private]

A text-based description of the cause of the Detour, intended to be displayed to customers.

If null or blank, the cause is considered Unspecified or Unknown.

5.8 Fare Class Reference

An immutable Value Object representing the cost, or 'fare,' required to ride a TransitVehicle on a particular Route.

Public Member Functions

- BigDecimal **getRegularFare** ()
- void setRegularFare (BigDecimal regularFare)
- BigDecimal **getDiscountedFare** ()
- void setDiscountedFare (BigDecimal discountedFare)

Private Attributes

- BigDecimal regularFare

  The normally applied fare.
- BigDecimal discountedFare

  A discounted fare for children, elderly, or other adjustment criteria as supplied by the TransitProvider.

5.8.1 Detailed Description

An immutable Value Object representing the cost, or 'fare,' required to ride a TransitVehicle on a particular Route.

5.8.2 Member Function Documentation

5.8.2.1 void setDiscountedFare (BigDecimal *discountedFare*)

Precondition

{regularFare} >= 0

5.8.2.2 void setRegularFare (BigDecimal *regularFare*)

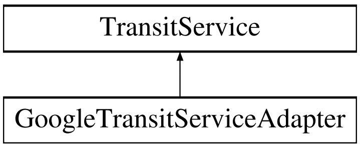
Precondition

regularFare >= 0

5.9 GoogleTransitServiceAdapter Class Reference

An Adapter Class to allow a {} service to appear as a TransitService.

Inheritance diagram for GoogleTransitServiceAdapter:



Public Member Functions

- GoogleTransitServiceAdapter (GoogleTransitServiceAPI googleTransitServiceAPI)

  Instantiates a new GoogleTransitServiceAdapter with a {} to delegate calls to.
- Route getRoute (String routeId)

  Gets the route.
- Set< Route > getRoutes (Location pickup, int distance)

  Gets the routes.
- TransitInfo **getTransitInfo** ()
- URL **getServiceURL** ()
- GoogleTransitServiceAPI **getGoogleTransitServiceAPI** ()
- void **setGoogleTransitServiceAPI** (GoogleTransitServiceAPI googleTransitServiceAPI)

Private Attributes

- GoogleTransitServiceAPI googleTransitServiceAPI

  The {} to adapt as a TransitService.

5.9.1 Detailed Description

An Adapter Class to allow a {} service to appear as a TransitService.

5.9.2 Constructor & Destructor Documentation

5.9.2.1 GoogleTransitServiceAdapter (GoogleTransitServiceAPI *googleTransitServiceAPI*)

Instantiates a new GoogleTransitServiceAdapter with a {} to delegate calls to.

Parameters

googleTransitServiceAPI	the google transit service api

5.9.3 Member Function Documentation

5.9.3.1 Route getRoute (String *routeId*)

Gets the route.

Precondition

routeId is not null or blank.

Postcondition

The Route is returned if the **routeId** is found, else null.

Parameters

routeId	the route id

Returns

the route

Implements TransitService.

5.9.3.2 Set< Route > getRoutes (Location *pickup*, int *distance*)

Gets the routes.

Parameters

pickup	the pickup
distance	the distance

Returns

the routes

Implements TransitService.

5.10 GoogleTransitServiceAPI Interface Reference

A client to Google's `Maps API`.

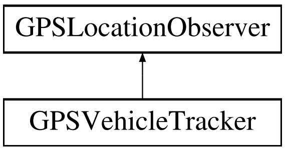
5.10.1 Detailed Description

A client to Google's `Maps API`.

5.11 GPSLocationObserver Interface Reference

Observer Pattern - Observer interface for GPS location tracking.

Inheritance diagram for GPSLocationObserver:



Public Member Functions

- void gpsUpdate (int gpsID, double latitude, double longitude)

  Observer Pattern update method to update transit vehicle GPS location.

5.11.1 Detailed Description

Observer Pattern - Observer interface for GPS location tracking.

5.11.2 Member Function Documentation

5.11.2.1 void gpsUpdate (int *gpsID*, double *latitude*, double *longitude*)

Observer Pattern update method to update transit vehicle GPS location.

Parameters

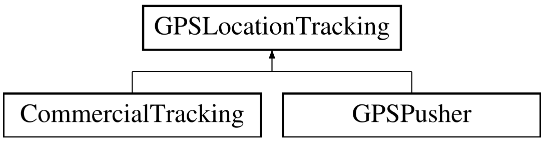
gpsID	- integer device ID from the GPS unit being tracked
latitude	- double new latitude from GPS device
longitude	- double new longitude from GPS device

Implemented in GPSVehicleTracker.

5.12 GPSLocationTracking Interface Reference

GPSLocationTracking - interface Subject of the Observer Pattern Defines methods for an observer GPS Device to register and receive updates on vehicle location.

Inheritance diagram for GPSLocationTracking:



Public Member Functions

- void registerGPSDevice (GPSLocationObserver gpsObs)

  registerGPSDevice - register a GPS device with the Location Tracking Service
- void unregisterGPSDevice (GPSLocationObserver gpsObs)

  unregisterGPSDevice - remove a vehicle from list.
- void pollGPSDevice ()

  pollGPSDevice - continuously poll registered GPS Devices for location updates

5.12.1 Detailed Description

GPSLocationTracking - interface Subject of the Observer Pattern Defines methods for an observer GPS Device to register and receive updates on vehicle location.

5.12.2 Member Function Documentation

5.12.2.1 void registerGPSDevice (GPSLocationObserver *gpsObs*)

registerGPSDevice - register a GPS device with the Location Tracking Service

Parameters

GPSLocationObserver	- Vehicle location to notify when new vehicle GPS location is received

Implemented in CommercialTracking, and GPSPusher.

5.12.2.2 void unregisterGPSDevice (GPSLocationObserver *gpsObs*)

unregisterGPSDevice - remove a vehicle from list.

Stop updating vehicle location.

Parameters

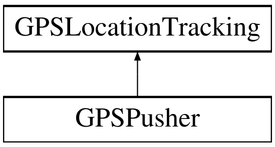
gpsObs	GPSLocationObserver - vehicle to remove from notification list

Implemented in CommercialTracking, and GPSPusher.

5.13 GPSPusher Class Reference

Implements Subject Location Tracking for retrieving GPS location updates from outside commercial tracking services.

Inheritance diagram for GPSPusher:



Classes

- class GPSPusherHolder

  GPS Pusher Holder is loaded on the first execution of GPSPusher.getInstance() or the first access to GPSPusher.INSTANCE, not before (lazy instantiation).

Public Member Functions

- void registerGPSDevice (GPSLocationObserver gpsObs)

  Register a GPS Device to the list to poll for updates.
- void unregisterGPSDevice (GPSLocationObserver gpsObs)

  Remove a GPS device from the list currently being polled for updates.
- void pollGPSDevice ()

  Continuously poll the registered GPS devices for location updates.

Static Public Member Functions

- static GPSPusher **getInstance** ()

Private Member Functions

- GPSPusher ()

  Only need one GPS Pusher Service running to track by polling all registered GPS devices.

Private Attributes

- ArrayList< GPSLocationObserver > gpsObserver

  Array list of GPS devices registered for updates.

5.13.1 Detailed Description

Implements Subject Location Tracking for retrieving GPS location updates from outside commercial tracking services.

5.13.2 Constructor & Destructor Documentation

5.13.2.1 GPSPusher ( ) [private]

Only need one GPS Pusher Service running to track by polling all registered GPS devices.

Constructor, creates ArrayList<GPSLocationObserver> to hold registered observers.

5.14 GPSPusher.GPSPusherHolder Class Reference

GPS Pusher Holder is loaded on the first execution of GPSPusher.getInstance() or the first access to GPSPusher.-INSTANCE, not before (lazy instantiation).

Static Public Attributes

- static final GPSPusher **INSTANCE** = new GPSPusher()

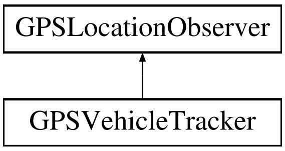
5.14.1 Detailed Description

GPS Pusher Holder is loaded on the first execution of GPSPusher.getInstance() or the first access to GPSPusher.-INSTANCE, not before (lazy instantiation).

5.15 GPSVehicleTracker Class Reference

Implementation of the Observer, update the transit vehicle GPS location.

Inheritance diagram for GPSVehicleTracker:



Public Member Functions

- GPSVehicleTracker (GPSLocationTracking gpsDevice)

  Register the Transit Vehicle GPS device with GPS location tracking.
- void gpsUpdate (int gpsID, double latitude, double longitude)

  Notify method to get the new GPS coordinates from GPS location tracking.

Private Attributes

- GPSLocationTracking **gpsDevice**
- int **gpsID**
- double **latitude**
- double **longitude**

5.15.1 Detailed Description

Implementation of the Observer, update the transit vehicle GPS location.

5.15.2 Constructor & Destructor Documentation

5.15.2.1 GPSVehicleTracker (GPSLocationTracking *gpsDevice*)

Register the Transit Vehicle GPS device with GPS location tracking.

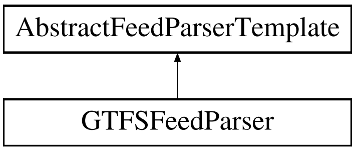
Parameters

gpsDevice	- GPSLocationTracking Subject being observed

5.16 GTFSFeedParser Class Reference

A AbstractFeedParserTemplate implementation designed to parse `GTFS` format ZIP files into Routes.

Inheritance diagram for GTFSFeedParser:



Protected Member Functions

- Set< Route > parseFeed (InputStream feed)

  Parse the `GTFS` format ZIP files into Routes.

Additional Inherited Members

5.16.1 Detailed Description

A AbstractFeedParserTemplate implementation designed to parse `GTFS` format ZIP files into Routes.

5.16.2 Member Function Documentation

5.16.2.1 Set<Route> parseFeed (InputStream *feed*) `[protected], [virtual]`

Parse the `GTFS` format ZIP files into Routes.

See Also

AbstractFeedParserTemplate::parseFeed(InputStream)

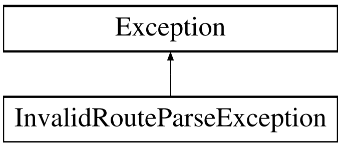
Implements AbstractFeedParserTemplate.

5.17 IAlertService Interface Reference

5.18 InvalidRouteParseException Class Reference

An InvalidRouteParseException indicates an invalid batch of parsed Routes has been been detected.

Inheritance diagram for InvalidRouteParseException:



Public Member Functions

- Set< Route > **getRouteBatch** ()
- void **setRouteBatch** (Set< Route > routeBatch)

Protected Member Functions

- InvalidRouteParseException (Set< Route > routeBatch)

  Instantiates a new invalid route parse exception.

Private Attributes

- Set< Route > routeBatch

  The failed Route batch.

Static Private Attributes

- static final long serialVersionUID = -4399874766965916500L

  The Constant serialVersionUID.

5.18.1 Detailed Description

An InvalidRouteParseException indicates an invalid batch of parsed Routes has been been detected.

Note that one ore more of the referenced Routes are invalid, but not necessarily all of them are invalid.

5.18.2 Constructor & Destructor Documentation

5.18.2.1 InvalidRouteParseException (Set< Route > *routeBatch*) [protected]

Instantiates a new invalid route parse exception.

Parameters

routeBatch	the route batch

5.18.3 Member Data Documentation

5.18.3.1 Set<Route> routeBatch [private]

The failed Route batch.

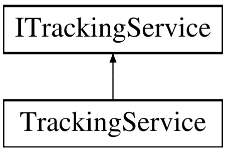
Handlers of this Exception may choose to re-validate, fix, and/or retry the operation with an adjusted batch.

5.18.3.2 final long serialVersionUID = -4399874766965916500L [static], [private]

The Constant serialVersionUID.

5.19 ITrackingService Interface Reference

Inheritance diagram for ITrackingService:



Classes

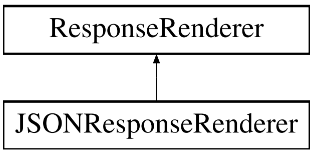
- enum AlertType

Public Member Functions

- void **AddTransitVehicle** ()
- void **getTransitVehicleLocation** ()
- void **getTransitVehicleAlerts** ()
- void **createTrackingAlert** (AlertType type)

5.20 JSONResponseRenderer Class Reference

Inheritance diagram for JSONResponseRenderer:



Public Member Functions

- void **render** (InputStream inputStream, OutputStream response)

Protected Member Functions

- InputStream **toJson** (InputStream inputStream)

5.21 Location Class Reference

An immutable Value Object representing a physical point on the geographic coordinate system.

Public Member Functions

- Location (double latitude, double longitude)

  Instantiates a new immutable Location with the given latitude and longitude.
- double **getLatitude** ()
- double **getLongitude** ()

Private Attributes

- double latitude

  The latitude of the point.
- double longitude

  The longitude of the point.

5.21.1 Detailed Description

An immutable Value Object representing a physical point on the geographic coordinate system.

5.21.2 Constructor & Destructor Documentation

5.21.2.1 Location (double *latitude,* double *longitude*)

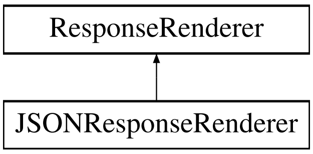
Instantiates a new immutable Location with the given latitude and longitude.

Parameters

latitude	The point latitude
longitude	The point longitude

5.22 ResponseRenderer Interface Reference

Inheritance diagram for ResponseRenderer:



Public Member Functions

- void **render** (InputStream inputStream, OutputStream response)

5.23 Route Class Reference

A Route is a TransitVehicle path of travel, or a "Line," as referred to by a TransitProvider.

Public Member Functions

- List< Stop > **getStops** ()
- void **setStops** (List< Stop > stops)
- String **getRouteId** ()
- void **setRouteId** (String routeId)
- String **getRouteName** ()
- void **setRouteName** (String routeName)

Private Attributes

- String routeId

A unique identifier for this Route.

- String routeName

    Text to display in maps and other literature to denote this Route.

- List< Stop > stops

    And ordered list of Stops to be visited in this Route.

5.23.1 Detailed Description

A Route is a TransitVehicle path of travel, or a "Line," as referred to by a TransitProvider.

A Route can be thought of as an ordered list of Stops.

Note that Routes may add/remove stops, change Stop times, or be disrupted by Detours, while still remaining the same Route.

5.23.2 Member Data Documentation

5.23.2.1 String routeName [private]

Text to display in maps and other literature to denote this Route.

Uniqueness is not enforced, but this name should provide enough context to allow users to distinguish this Route.

5.23.2.2 List<Stop> stops [private]

And ordered list of Stops to be visited in this Route.

Stops must be visited in order unless there is a disruption in service, in which case clients can expect a RouteDistruptionAlert and/or an Detour.

5.24 RouteDisruptionAlert Class Reference

TODO:

5.24.1 Detailed Description

TODO:

5.25 RouteRepository Interface Reference

A Repository Pattern supporting lifecycle operations of Routes, such as Read, Save, Delete, and Query functionality.

Public Member Functions

- void delete (String routeId)

    *Deletes the Route corresponding to the given **routeId**.*

- void save (Route route)

    Saves the Route to the Repository.

- void save (Set< Route > routes)

    Saves all of the Routes to the Repository.

- Route read (String routeId)

    Read a single Route from the Repository by its identifier.

- Collection< Route > getAll ()

    Retrieves all available Routes in the Repository.

5.25.1 Detailed Description

A Repository Pattern supporting lifecycle operations of Routes, such as Read, Save, Delete, and Query functionality.

5.25.2 Member Function Documentation

5.25.2.1 void delete (String *routeId*)

Deletes the Route corresponding to the given **routeId**.

Precondition

A Route with the given **routeId** exists in the Repository.

Postcondition

A Route with the given **routeId** is removed from the Repository and is no longer available for retrieval.

Parameters

routeId	

5.25.2.2 Collection<**Route**> getAll ( )

Retrieves all available Routes in the Repository.

Returns

All available Routes.

5.25.2.3 **Route** read (String *routeId*)

Read a single Route from the Repository by its identifier.

If no Route is found with the requested **routeId**, a null value is returned.

Parameters

routeId	The identifier of the requested Route

Returns

The requested Route

5.25.2.4 void save (**Route** *route*)

Saves the Route to the Repository.

Precondition

The Route has been validated with all appropriate business rules.

See Also

RouteSpecification

Postcondition

The Route is available for retrieval by id and also by appropriate Queries.

Parameters

route	The Route to save.

5.25.2.5 void save (Set< **Route** > *routes*)

Saves all of the Routes to the Repository.

Precondition

The Routes have been validated with all appropriate business rules.

See Also

RouteSpecification

Postcondition

The Routes are available for retrieval by id and also by appropriate Queries.

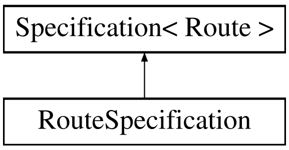
Parameters

route	The Route to save.

5.26 RouteSpecification Class Reference

A Specification Pattern class for validating a Route.

Inheritance diagram for RouteSpecification:



Public Member Functions

- boolean isSatisfiedBy (Route candidate)

  Validates the given Route with the appropriate validation rules of this Specification.
- Specification< Route > **or** (Specification< Route > specification)
- Specification< Route > **and** (Specification< Route > specification)
- Specification< Route > **not** (Specification< Route > specification)

5.26.1 Detailed Description

A Specification Pattern class for validating a Route.

Validation can happen in this class, or business rules can be combined using the and(Specification), or #not(-Specification) methods and separate Specifications. A Route

This Specification is necessary because Routes aren't necessarily created by a controlled Factory, but as a result of parsing of input files via subclasses of the AbstractFeedParserTemplate. Because creation of Routes isn't centralized, it is necessary to centralize the validation rules so that subclasses of AbstractFeedParserTemplate have access to it.

This Specification must validate the following conditions:

1. Each Route has an non-null routeId
2. Each Route has an non-blank routeName
3. Each Route has one or more Stops

5.26.2 Member Function Documentation

5.26.2.1 boolean isSatisfiedBy (Route *candidate*)

Validates the given Route with the appropriate validation rules of this Specification.

Parameters

candidate	The Route to be validated

Returns

True if a valid Route, False if not

5.27 Specification< T > Interface Reference

A Generic Specification to be used for chaining business validation rules together.

Public Member Functions

- boolean isSatisfiedBy (T candidate)

  Checks if the given candidate satisfies the specification.
- Specification< T > or (Specification< T > specification)

  Returns a Specification representing the 'OR' boolean operation of the Specifications.
- Specification< T > and (Specification< T > specification)

  Returns a Specification representing the 'AND' boolean operation of the Specifications.
- Specification< T > not (Specification< T > specification)

  Returns a Specification representing the 'NOT' boolean operation of the Specifications.

5.27.1 Detailed Description

A Generic Specification to be used for chaining business validation rules together.

Parameters

<T>	The candidate Type accepted by the Specification.

5.27.2 Member Function Documentation

5.27.2.1 Specification<T> and (Specification< T > *specification*)

Returns a Specification representing the 'AND' boolean operation of the Specifications.

Parameters

specification	The Specification to apply the 'AND' operation to.

Returns

The 'AND' Specification

5.27.2.2 boolean isSatisfiedBy (T *candidate*)

Checks if the given candidate satisfies the specification.

Parameters

candidate	The candidate

Returns

true, if is satisfied by the candidate

5.27.2.3 Specification$<$T$>$ not (Specification$<$ T $>$ *specification*)

Returns a Specification representing the 'NOT' boolean operation of the Specifications.

Parameters

specification	The Specification to apply the 'NOT' operation to.

Returns

The 'NOT' Specification

5.27.2.4 Specification$<$T$>$ or (Specification$<$ T $>$ *specification*)

Returns a Specification representing the 'OR' boolean operation of the Specifications.

Parameters

specification	The Specification to apply the 'OR' operation to.

Returns

The 'OR' Specification

5.28 Stop Class Reference

A point on a Route in which a TransitVehicle will stop to pick up and drop off passengers.

Public Member Functions

- Set$<$ Date $>$ getStopTimes (Date begin, Date end)

  Reports the expected times in which a TransitVehicle will be at the given Stop for a given time period.
- Location **getLocation** ()
- void **setLocation** (Location location)

Private Attributes

- Location location

    The physical location of the Stop.

5.28.1 Detailed Description

A point on a Route in which a TransitVehicle will stop to pick up and drop off passengers.

A Stop also is responsible for providing a set of the times in which the TransitVehicle will be at the Stop.

A Stop is identified within the context of a single Route. This means that two Routes may share the same physical Stop location, but maintain different schedules.

5.28.2 Member Function Documentation

5.28.2.1 Set$<$Date$>$ getStopTimes (Date *begin,* Date *end*)

Reports the expected times in which a TransitVehicle will be at the given Stop for a given time period.

Precondition

begin $<$ **end**.

Parameters

begin	The start of the reporting time period. All Stop Times returned will be on (or after) this time. If null, assume to be the current time.
end	The end of the reporting time period. All Stop Times returned will before this time.

Returns

Stop Times associated with this Stop that satisfy the begin and end criteria.

5.29 TrackingAlert Class Reference

Public Member Functions

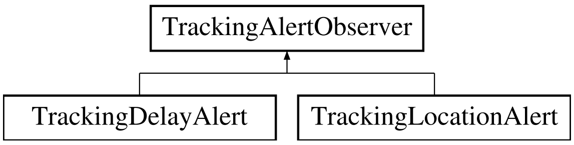
- String **getUserContactInfo** ()
- void **setUserContactInfo** (String userContactInfo)
- double **getNotifyVehicleLatitude** ()
- void **setNotifyVehicleLatitude** (double notifyVehicleLatitude)
- double **getNotifyVehicleLongitude** ()
- void **setNotifyVehicleLongitude** (double notifyVehicleLongitude)

Private Attributes

- String **userContactInfo**
- double **notifyVehicleLatitude**
- double **notifyVehicleLongitude**
- Timestamp **alertTime**

5.30 TrackingAlertObserver Class Reference

Inheritance diagram for TrackingAlertObserver:



Public Member Functions

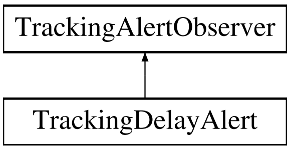
- abstract void **updateAlert** ()
- String **toString** ()
- double **getNotifyLocation** ()
- void **setNotifyLocation** (double notifyLocation)
- double **getNotifyLatitude** ()
- void **setNotifyLatitude** (double notifyLatitude)
- Timestamp **getNotifyTime** ()
- void **setNotifyTime** (Timestamp notifyTime)

Private Attributes

- String **sendAlertTo**
- double **notifyLocation**
- double **notifyLatitude**
- Timestamp **notifyTime**

5.31 TrackingDelayAlert Class Reference

Inheritance diagram for TrackingDelayAlert:

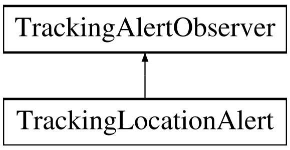


Public Member Functions

- void **updateAlert** ()

5.32 TrackingLocationAlert Class Reference

Inheritance diagram for TrackingLocationAlert:



Public Member Functions

- TrackingLocationAlert (TransitVehicle vehicle, TrackingAlert userAlertInfo)

  Tracking Location Alert constructor.
- void updateAlert ()

  Vehicle is in vicinity where user registered to be notified, call Alert Service.

5.32.1 Constructor & Destructor Documentation

5.32.1.1 TrackingLocationAlert (TransitVehicle *vehicle,* TrackingAlert *userAlertInfo*)

Tracking Location Alert constructor.

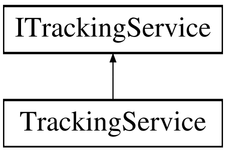
Associates user information with vehicle to monitor.

Parameters

vehicle	TransitVehicle Vehicle to add alert monitoring
alert	- TrackingAlert User contact details and vehicle location indicating when user wants to receive alert.

5.33 TrackingService Class Reference

Inheritance diagram for TrackingService:



Public Member Functions

- void **AddTransitVehicle** ()
- void **getTransitVehicleLocation** ()
- void **getTransitVehicleAlerts** ()
- void **createTrackingAlert** (AlertType type)

5.34 TrackingServiceController Class Reference

Public Member Functions

- void **registerVehicleOnRoute** (char type, String url, int vehicleNum)
- void **addTransitVehicle** (TransitVehicle tv)
- void **addUserTrackingAlert** ()

Static Public Member Functions

- static void main (String[] args)

Private Attributes

- ArrayList< TransitVehicle > **vehicleList** = null

5.34.1 Member Function Documentation

5.34.1.1 static void main (String[] *args*) [static]

Parameters

args	

5.35 TransitFeed Interface Reference

TODO:

Public Member Functions

- Route **getRoute** (String routeId)
- Set< Route > **getRoutes** (Location pickup, Location dropoff, int distance)

5.35.1 Detailed Description

TODO:

5.36 TransitInfo Class Reference

TODO:

Public Member Functions

- String **getTransitAuthorityName** ()
- void **setTransitAuthorityName** (String transitAuthorityName)
- URL **getWebsite** ()
- void **setWebsite** (URL website)
- byte[] **getLogo** ()
- void **setLogo** (byte[] logo)

Private Attributes

- String **transitAuthorityName**
- URL **website**
- byte[] **logo**

5.36.1 Detailed Description

TODO:

5.37 TransitProvider Class Reference

A TransitProvider is a description of a company or organization that is the producer of public transportation services.

Public Member Functions

- String **getProviderId** ()
- void **setProviderId** (String providerId)
- String **getName** ()
- void **setName** (String name)

Private Attributes

- String providerId

  A unique identifier that globally identifies this TransitProvider.
- String name

  A text description of the TransitProvider.

5.37.1 Detailed Description

A TransitProvider is a description of a company or organization that is the producer of public transportation services.

5.37.2 Member Data Documentation

5.37.2.1 String name [private]

A text description of the TransitProvider.

This is the text that will be displayed on guides, Route maps, and advertisements.

5.37.2.2 String providerId [private]

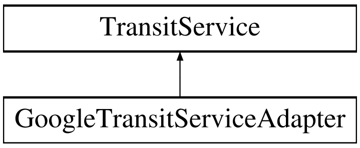
A unique identifier that globally identifies this TransitProvider.

The actual identifier type is unspecified and left to implementations. It may be the same as the name, if that is sufficient to provide uniqueness.

5.38 TransitService Interface Reference

The TransitService is an interface to get Route/Fare/Detour information from a TransitProvider.

Inheritance diagram for TransitService:



Public Member Functions

- Route getRoute (String routeId)

  Gets the route.
- Set< Route > getRoutes (Location pickup, int distance)

  Gets the routes.
- TransitInfo **getTransitInfo** ()
- URL **getServiceURL** ()

5.38.1 Detailed Description

The TransitService is an interface to get Route/Fare/Detour information from a TransitProvider.

This service will provide a consistent interface for the application logic to query to get this information.

5.38.2 Member Function Documentation

5.38.2.1 Route getRoute (String *routeId*)

Gets the route.

Precondition

routeId is not null or blank.

Postcondition

The Route is returned if the **routeId** is found, else null.

Parameters

routeId	the route id

Returns

the route

Implemented in GoogleTransitServiceAdapter.

5.38.2.2 Set<Route> getRoutes (Location *pickup*, int *distance*)

Gets the routes.

Parameters

pickup	the pickup
distance	the distance

Returns

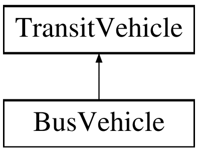
the routes

Implemented in GoogleTransitServiceAdapter.

5.39 TransitVehicle Class Reference

Abstract transit vehicle class contains the common data for all types of vehicles and the Subject GPS Tracking and the GPS observer to receive GPS location updates.

Inheritance diagram for TransitVehicle:



Public Member Functions

- abstract void registerTrackingAlert (TrackingAlertObserver ao)

  Transit Vehicle is also the subject for tracking user subscribed alerts.
- abstract void **unregisterTrackingAlert** (TrackingAlertObserver ao)
- abstract void **triggerAlert** ()
- String toString ()

  Provide a generic method to output Transit Vehicle information.

Private Attributes

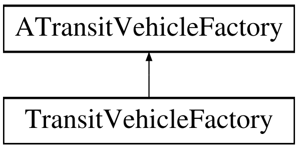
- URL **url**
- int **vehicleID**
- int **routeID**
- double **vehicleLatitude**
- double **vehicleLongitude**
- int **gpsDeviceID**
- GPSLocationObserver **gpsObserver**
- GPSLocationTracking **gpsTracker**

5.39.1 Detailed Description

Abstract transit vehicle class contains the common data for all types of vehicles and the Subject GPS Tracking and the GPS observer to receive GPS location updates.

5.40 TransitVehicleFactory Class Reference

Inheritance diagram for TransitVehicleFactory:



Public Member Functions

- TransitVehicle **createTransitVehicle** (URL url, int vehicle)

Protected Member Functions

- int getVehicleGPSDeviceID (URL url)

  Retrieve the GPS Device ID from repository of vehicles registered for route identified by type and URL.

5.40.1 Member Function Documentation

5.40.1.1 int getVehicleGPSDeviceID (URL *url*) [protected]

Retrieve the GPS Device ID from repository of vehicles registered for route identified by type and URL.

Parameters

url	- URL identifying the transit company

Returns

integer GPS Device ID

5.41 Trip Class Reference

A Trip is considered an ordered collection of Routes going from a starting point to an ending point.

Public Member Functions

- Collection< Route > **getRoutes** ()
- void **setRoutes** (Collection< Route > routes)

Private Attributes

- Collection< Route > routes

  The ordered collection of Routes that when combined make a navigable Trip.

5.41.1 Detailed Description

A Trip is considered an ordered collection of Routes going from a starting point to an ending point.

A Trip can be thought of as a composition of Routes, and the TripService is the service that composes them.

5.42 TripService Interface Reference

A Service to calculate a collection of Routes, or a Trip, allowing for a continuous transit path from a start Location to an end Location.

Public Member Functions

- Trip calculateTrip (Location start, Location end)

  *Calculate an optimal Trip given a **start** Location and an **end** Location.*

5.42.1 Detailed Description

A Service to calculate a collection of Routes, or a Trip, allowing for a continuous transit path from a start Location to an end Location.

5.42.2 Member Function Documentation

5.42.2.1 Trip calculateTrip (Location *start*, Location *end*)

Calculate an optimal Trip given a **start** Location and an **end** Location.

Parameters

start	The requested start Location of the Trip.
end	The requested end Location of the Trip.

Returns

The calculated Trip

5.43 User Class Reference

Private Attributes

- int **userId**
- String **userName**

5.44 UserLoginService Interface Reference

5.45 UserManagementService Interface Reference

5.46 VehicleObject Class Reference

Public Member Functions

- int **getGpsDeviceID** ()
- void **setGpsDeviceID** (int gpsDeviceID)
- int **getGpsDeviceType** ()
- void **setGpsDeviceType** (int gpsDeviceType)
- int **getVehicleType** ()
- void **setVehicleType** (int vehicleType)
- String **getTransitCoURL** ()
- void **setTransitCoURL** (String transitCoURL)
- int **getCurrentRoute** ()
- void **setCurrentRoute** (int currentRoute)

Private Attributes

- int **gpsDeviceID**
- int **gpsDeviceType**
- int **vehicleType**
- String **transitCoURL**
- int **currentRoute**

5.47 vehicleRepository Class Reference

Repository for information on vehicles registered on a route.

Public Member Functions

- void **addVehicle** (VehicleObject vehicle)
- void **removeVehicle** (int gpsDeviceID)
- VehicleObject **findVehicle** (int gpsDeviceID)
- ArrayList< VehicleObject > **findVehiclesByRoute** (URL transitCoURL, int routeID)

5.47.1 Detailed Description

Repository for information on vehicles registered on a route.

Index

ATransitVehicleFactory, 10
AbstractFeedParserTemplate, 7
alert, 5
and
    common::Specification< T >, 26

BusVehicle, 10

calculateTrip
    transit::TripService, 35
cause
    transit::Detour, 12
CommercialTracking, 10
    tracking::CommercialTracking, 11
CommercialTracking.CommercialTrackingHolder, 11
common::Specification< T >
    and, 26
    isSatisfiedBy, 26
    not, 27
    or, 27

delete
    transit::RouteRepository, 23
Detour, 12

Fare, 12

GPSLocationObserver, 15
GPSLocationTracking, 15
GPSPusher, 16
    tracking::GPSPusher, 17
GPSPusher.GPSPusherHolder, 17
GPSVehicleTracker, 18
    tracking::GPSVehicleTracker, 18
GTFSFeedParser, 18
getAll
    transit::RouteRepository, 24
getRoute
    transit::GoogleTransitServiceAdapter, 14
    transit::TransitService, 32
getRoutes
    transit::GoogleTransitServiceAdapter, 14
    transit::TransitService, 33
getStopTimes
    transit::Stop, 28
getVehicleGPSDeviceID
    tracking::TransitVehicleFactory, 34
GoogleTransitServiceAPI, 15
GoogleTransitServiceAdapter, 13
    transit::GoogleTransitServiceAdapter, 14
gpsUpdate
    tracking::GPSLocationObserver, 15

IAlertService, 19
ITrackingService, 20
ITrackingService.AlertType, 10
InvalidRouteParseException, 19
    transit::InvalidRouteParseException, 20
isSatisfiedBy
    common::Specification< T >, 26
    transit::RouteSpecification, 25

JSONResponseRenderer, 21

loadFeed
    transit::AbstractFeedParserTemplate, 8
Location, 21
    transit::Location, 22

main
    tracking::TrackingServiceController, 30

name
    transit::TransitProvider, 32
not
    common::Specification< T >, 27

or
    common::Specification< T >, 27

parseFeed
    transit::AbstractFeedParserTemplate, 9
    transit::GTFSFeedParser, 19
providerId
    transit::TransitProvider, 32

read
    transit::RouteRepository, 24
registerGPSDevice
    tracking::GPSLocationTracking, 16
ResponseRenderer, 22
Route, 22
routeBatch
    transit::InvalidRouteParseException, 20
RouteDisruptionAlert, 23
routeName
    transit::Route, 23
RouteRepository, 23
RouteSpecification, 25

save
    transit::RouteRepository, 24
saveRoutes
    transit::AbstractFeedParserTemplate, 9
serialVersionUID
    transit::InvalidRouteParseException, 20
setDiscountedFare
    transit::Fare, 13
setRegularFare
    transit::Fare, 13
Specification< T >, 26
start
    transit::AbstractFeedParserTemplate, 9
Stop, 27

stops
  transit::Route, 23

tracking, 5
tracking::CommercialTracking
  CommercialTracking, 11
tracking::GPSLocationObserver
  gpsUpdate, 15
tracking::GPSLocationTracking
  registerGPSDevice, 16
  unregisterGPSDevice, 16
tracking::GPSPusher
  GPSPusher, 17
tracking::GPSVehicleTracker
  GPSVehicleTracker, 18
tracking::TrackingLocationAlert
  TrackingLocationAlert, 29
tracking::TrackingServiceController
  main, 30
tracking::TransitVehicleFactory
  getVehicleGPSDeviceID, 34
TrackingAlert, 28
TrackingAlertObserver, 28
TrackingDelayAlert, 29
TrackingLocationAlert, 29
  tracking::TrackingLocationAlert, 29
TrackingService, 30
TrackingServiceController, 30
transit, 6
transit::AbstractFeedParserTemplate
  loadFeed, 8
  parseFeed, 9
  saveRoutes, 9
  start, 9
  validate, 9
transit::Detour
  cause, 12
transit::Fare
  setDiscountedFare, 13
  setRegularFare, 13
transit::GTFSFeedParser
  parseFeed, 19
transit::GoogleTransitServiceAdapter
  getRoute, 14
  getRoutes, 14
  GoogleTransitServiceAdapter, 14
transit::InvalidRouteParseException
  InvalidRouteParseException, 20
  routeBatch, 20
  serialVersionUID, 20
transit::Location
  Location, 22
transit::Route
  routeName, 23
  stops, 23
transit::RouteRepository
  delete, 23
  getAll, 24
  read, 24
  save, 24
transit::RouteSpecification
  isSatisfiedBy, 25
transit::Stop
  getStopTimes, 28
transit::TransitProvider
  name, 32
  providerId, 32
transit::TransitService
  getRoute, 32
  getRoutes, 33
transit::TripService
  calculateTrip, 35
TransitFeed, 30
TransitInfo, 31
TransitProvider, 31
TransitService, 32
TransitVehicle, 33
TransitVehicleFactory, 34
Trip, 34
TripService, 35

unregisterGPSDevice
  tracking::GPSLocationTracking, 16
User, 35
user, 7
UserLoginService, 35
UserManagementService, 36

validate
  transit::AbstractFeedParserTemplate, 9
VehicleObject, 36
vehicleRepository, 36