

# RouteFinder

Generated by Doxygen 1.8.8

Fri Nov 14 2014 17:58:24



# Contents

|          |  |          |
|----------|--|----------|
| <b>1</b> | <b>Hierarchical Index</b>                        | <b>1</b> |
| 1.1      | Class Hierarchy . . . . .                        | 1        |
| <b>2</b> | <b>Class Index</b>                               | <b>3</b> |
| 2.1      | Class List . . . . .                             | 3        |
| <b>3</b> | <b>Class Documentation</b>                       | <b>5</b> |
| 3.1      | DataBase Class Reference . . . . .               | 5        |
| 3.1.1    | Detailed Description . . . . .                   | 5        |
| 3.1.2    | Member Enumeration Documentation . . . . .       | 5        |
| 3.1.2.1  | LoadMethod . . . . .                             | 5        |
| 3.1.3    | Constructor & Destructor Documentation . . . . . | 6        |
| 3.1.3.1  | DataBase . . . . .                               | 6        |
| 3.1.4    | Member Function Documentation . . . . .          | 7        |
| 3.1.4.1  | isValid . . . . .                                | 7        |
| 3.1.4.2  | printTimeTable . . . . .                         | 7        |
| 3.1.5    | Member Data Documentation . . . . .              | 7        |
| 3.1.5.1  | routes . . . . .                                 | 7        |
| 3.1.5.2  | services . . . . .                               | 7        |
| 3.1.5.3  | stops . . . . .                                  | 7        |
| 3.1.5.4  | stopTimes . . . . .                              | 7        |
| 3.1.5.5  | trips . . . . .                                  | 7        |
| 3.2      | DataReader Class Reference . . . . .             | 7        |
| 3.2.1    | Detailed Description . . . . .                   | 8        |
| 3.2.2    | Member Function Documentation . . . . .          | 8        |
| 3.2.2.1  | readRoutes . . . . .                             | 8        |
| 3.2.2.2  | readServices . . . . .                           | 8        |
| 3.2.2.3  | readStops . . . . .                              | 8        |
| 3.2.2.4  | readStopTimes . . . . .                          | 9        |
| 3.2.2.5  | readTrips . . . . .                              | 10       |
| 3.3      | Edge Class Reference . . . . .                   | 10       |
| 3.3.1    | Detailed Description . . . . .                   | 10       |

|          |  |    |
|----------|--|----|
| 3.3.2    | Constructor & Destructor Documentation     | 11 |
| 3.3.2.1  | Edge                                       | 11 |
| 3.3.3    | Member Function Documentation              | 11 |
| 3.3.3.1  | getEndNode                                 | 11 |
| 3.3.3.2  | getID                                      | 11 |
| 3.3.3.3  | getStartNode                               | 11 |
| 3.3.3.4  | getType                                    | 11 |
| 3.3.3.5  | getWeight                                  | 11 |
| 3.3.3.6  | operator"!="                               | 11 |
| 3.3.3.7  | operator<                                  | 12 |
| 3.3.3.8  | operator=                                  | 12 |
| 3.3.3.9  | operator==                                 | 12 |
| 3.3.3.10 | setType                                    | 12 |
| 3.3.3.11 | setWeight                                  | 12 |
| 3.3.4    | Friends And Related Function Documentation | 13 |
| 3.3.4.1  | operator<<                                 | 13 |
| 3.4      | GTFSReader Class Reference                 | 13 |
| 3.4.1    | Detailed Description                       | 13 |
| 3.4.2    | Member Function Documentation              | 13 |
| 3.4.2.1  | getRoutes                                  | 13 |
| 3.4.2.2  | getServices                                | 14 |
| 3.4.2.3  | getStops                                   | 14 |
| 3.4.2.4  | getStopTimes                               | 14 |
| 3.4.2.5  | getTrips                                   | 14 |
| 3.4.2.6  | readGTFS                                   | 14 |
| 3.4.3    | Friends And Related Function Documentation | 14 |
| 3.4.3.1  | operator<<                                 | 14 |
| 3.5      | Network Class Reference                    | 14 |
| 3.5.1    | Detailed Description                       | 15 |
| 3.5.2    | Constructor & Destructor Documentation     | 15 |
| 3.5.2.1  | Network                                    | 15 |
| 3.5.2.2  | Network                                    | 15 |
| 3.5.2.3  | ~Network                                   | 15 |
| 3.5.2.4  | Network                                    | 15 |
| 3.5.3    | Member Function Documentation              | 15 |
| 3.5.3.1  | findRouteBetween                           | 15 |
| 3.5.3.2  | getAllNodes                                | 16 |
| 3.5.3.3  | loadFromFile                               | 16 |
| 3.5.3.4  | setSover                                   | 16 |
| 3.5.4    | Friends And Related Function Documentation | 16 |

|          |  |    |
|----------|--|----|
| 3.5.4.1  | operator<<                                 | 16 |
| 3.6      | Node Class Reference                       | 17 |
| 3.6.1    | Detailed Description                       | 17 |
| 3.6.2    | Constructor & Destructor Documentation     | 17 |
| 3.6.2.1  | Node                                       | 17 |
| 3.6.3    | Member Function Documentation              | 17 |
| 3.6.3.1  | getID                                      | 17 |
| 3.6.3.2  | getLatitude                                | 17 |
| 3.6.3.3  | getLongitude                               | 18 |
| 3.6.3.4  | getName                                    | 18 |
| 3.6.3.5  | operator!=                                 | 18 |
| 3.6.3.6  | operator<                                  | 18 |
| 3.6.3.7  | operator=                                  | 18 |
| 3.6.3.8  | operator==                                 | 18 |
| 3.6.4    | Friends And Related Function Documentation | 19 |
| 3.6.4.1  | operator<<                                 | 19 |
| 3.7      | Route Class Reference                      | 19 |
| 3.7.1    | Detailed Description                       | 20 |
| 3.7.2    | Constructor & Destructor Documentation     | 20 |
| 3.7.2.1  | Route                                      | 20 |
| 3.7.3    | Member Function Documentation              | 20 |
| 3.7.3.1  | addEdge                                    | 20 |
| 3.7.3.2  | begin                                      | 20 |
| 3.7.3.3  | end  | 20 |
| 3.7.3.4  | getEndNode                                 | 20 |
| 3.7.3.5  | getLength                                  | 20 |
| 3.7.3.6  | getStartNode                               | 20 |
| 3.7.3.7  | getWeight                                  | 21 |
| 3.7.3.8  | isConnectionBetween                        | 21 |
| 3.7.3.9  | isEdgeIn                                   | 21 |
| 3.7.3.10 | isNodeIn                                   | 21 |
| 3.7.3.11 | switchEdge                                 | 21 |
| 3.7.3.12 | switchRoute                                | 21 |
| 3.7.3.13 | validate                                   | 22 |
| 3.7.4    | Friends And Related Function Documentation | 22 |
| 3.7.4.1  | operator<<                                 | 22 |
| 3.8      | RouteData Class Reference                  | 22 |
| 3.8.1    | Detailed Description                       | 23 |
| 3.8.2    | Constructor & Destructor Documentation     | 23 |
| 3.8.2.1  | RouteData                                  | 23 |

|          |  |    |
|----------|--|----|
| 3.8.2.2  | RouteData                                  | 23 |
| 3.8.2.3  | RouteData                                  | 23 |
| 3.8.2.4  | ~RouteData                                 | 23 |
| 3.8.3    | Member Function Documentation              | 23 |
| 3.8.3.1  | getId                                      | 23 |
| 3.8.3.2  | getName                                    | 23 |
| 3.8.3.3  | operator=                                  | 23 |
| 3.8.3.4  | operator==                                 | 24 |
| 3.8.4    | Friends And Related Function Documentation | 24 |
| 3.8.4.1  | operator<<                                 | 24 |
| 3.9      | ServiceData Class Reference                | 24 |
| 3.9.1    | Detailed Description                       | 25 |
| 3.9.2    | Constructor & Destructor Documentation     | 25 |
| 3.9.2.1  | ServiceData                                | 25 |
| 3.9.2.2  | ServiceData                                | 25 |
| 3.9.2.3  | ServiceData                                | 25 |
| 3.9.2.4  | ~ServiceData                               | 25 |
| 3.9.3    | Member Function Documentation              | 25 |
| 3.9.3.1  | getId                                      | 25 |
| 3.9.3.2  | getName                                    | 25 |
| 3.9.3.3  | operator=                                  | 25 |
| 3.9.3.4  | operator==                                 | 26 |
| 3.9.4    | Friends And Related Function Documentation | 26 |
| 3.9.4.1  | operator<<                                 | 26 |
| 3.10     | SimAnnealingAlg Class Reference            | 26 |
| 3.10.1   | Detailed Description                       | 27 |
| 3.10.2   | Member Function Documentation              | 27 |
| 3.10.2.1 | solve                                      | 27 |
| 3.11     | Solver Class Reference                     | 27 |
| 3.11.1   | Detailed Description                       | 27 |
| 3.11.2   | Member Function Documentation              | 27 |
| 3.11.2.1 | solve                                      | 27 |
| 3.12     | StopData Class Reference                   | 28 |
| 3.12.1   | Detailed Description                       | 28 |
| 3.12.2   | Constructor & Destructor Documentation     | 28 |
| 3.12.2.1 | StopData                                   | 28 |
| 3.12.2.2 | StopData                                   | 28 |
| 3.12.2.3 | StopData                                   | 29 |
| 3.12.2.4 | ~StopData                                  | 29 |
| 3.12.3   | Member Function Documentation              | 29 |

|          |  |    |
|----------|--|----|
| 3.12.3.1 | getId                                      | 29 |
| 3.12.3.2 | getLat                                     | 29 |
| 3.12.3.3 | getLng                                     | 29 |
| 3.12.3.4 | getName                                    | 29 |
| 3.12.3.5 | operator=                                  | 29 |
| 3.12.3.6 | operator==                                 | 30 |
| 3.12.4   | Friends And Related Function Documentation | 30 |
| 3.12.4.1 | operator<<                                 | 30 |
| 3.13     | StopTimeData Class Reference               | 30 |
| 3.13.1   | Detailed Description                       | 31 |
| 3.13.2   | Constructor & Destructor Documentation     | 31 |
| 3.13.2.1 | StopTimeData                               | 31 |
| 3.13.2.2 | StopTimeData                               | 31 |
| 3.13.2.3 | StopTimeData                               | 31 |
| 3.13.2.4 | ~StopTimeData                              | 31 |
| 3.13.3   | Member Function Documentation              | 31 |
| 3.13.3.1 | getId                                      | 31 |
| 3.13.3.2 | getName                                    | 32 |
| 3.13.3.3 | getServiceId                               | 32 |
| 3.13.3.4 | getStopId                                  | 32 |
| 3.13.3.5 | getStopTime                                | 32 |
| 3.13.3.6 | getTripld                                  | 32 |
| 3.13.3.7 | operator=                                  | 32 |
| 3.13.3.8 | operator==                                 | 32 |
| 3.13.4   | Friends And Related Function Documentation | 33 |
| 3.13.4.1 | operator<<                                 | 33 |
| 3.14     | Time Class Reference                       | 34 |
| 3.14.1   | Detailed Description                       | 34 |
| 3.14.2   | Constructor & Destructor Documentation     | 34 |
| 3.14.2.1 | Time                                       | 34 |
| 3.14.2.2 | Time                                       | 34 |
| 3.14.2.3 | Time                                       | 35 |
| 3.14.2.4 | Time                                       | 35 |
| 3.14.2.5 | ~Time                                      | 35 |
| 3.14.3   | Member Function Documentation              | 35 |
| 3.14.3.1 | operator int                               | 35 |
| 3.14.3.2 | operator"!="                               | 35 |
| 3.14.3.3 | operator+                                  | 35 |
| 3.14.3.4 | operator-                                  | 36 |
| 3.14.3.5 | operator<                                  | 37 |

|              |  |           |
|--------------|--|-----------|
| 3.14.3.6     | operator=                                  | 37        |
| 3.14.3.7     | operator==                                 | 37        |
| 3.14.3.8     | operator>                                  | 37        |
| 3.14.4       | Friends And Related Function Documentation | 37        |
| 3.14.4.1     | operator<<                                 | 37        |
| 3.15         | TripData Class Reference                   | 38        |
| 3.15.1       | Detailed Description                       | 38        |
| 3.15.2       | Constructor & Destructor Documentation     | 38        |
| 3.15.2.1     | TripData                                   | 38        |
| 3.15.2.2     | TripData                                   | 38        |
| 3.15.2.3     | TripData                                   | 39        |
| 3.15.2.4     | ~TripData                                  | 39        |
| 3.15.3       | Member Function Documentation              | 39        |
| 3.15.3.1     | getId                                      | 39        |
| 3.15.3.2     | getName                                    | 39        |
| 3.15.3.3     | getRouteId                                 | 39        |
| 3.15.3.4     | getStopSec                                 | 39        |
| 3.15.3.5     | operator=                                  | 39        |
| 3.15.3.6     | operator==                                 | 40        |
| 3.15.4       | Friends And Related Function Documentation | 40        |
| 3.15.4.1     | operator<<                                 | 40        |
| <b>Index</b> |  | <b>41</b> |



# Chapter 1

## Hierarchical Index

### 1.1 Class Hierarchy

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

|                           |    |
|---------------------------|----|
| DataBase . . . . .        | 5  |
| DataReader . . . . .      | 7  |
| Edge . . . . .            | 10 |
| GTFSReader . . . . .      | 13 |
| Network . . . . .         | 14 |
| Node . . . . .            | 17 |
| Route . . . . .           | 19 |
| RouteData . . . . .       | 22 |
| ServiceData . . . . .     | 24 |
| Solver . . . . .          | 27 |
| SimAnnealingAlg . . . . . | 26 |
| StopData . . . . .        | 28 |
| StopTimeData . . . . .    | 30 |
| Time . . . . .            | 34 |
| TripData . . . . .        | 38 |



## Chapter 2

# Class Index

### 2.1 Class List

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

|                                 |    |
|---------------------------------|----|
| <a href="#">DataBase</a>        | 5  |
| <a href="#">DataReader</a>      | 7  |
| <a href="#">Edge</a>            | 10 |
| <a href="#">GTFSReader</a>      | 13 |
| <a href="#">Network</a>         | 14 |
| <a href="#">Node</a>            | 17 |
| <a href="#">Route</a>           | 19 |
| <a href="#">RouteData</a>       | 22 |
| <a href="#">ServiceData</a>     | 24 |
| <a href="#">SimAnnealingAlg</a> | 26 |
| <a href="#">Solver</a>          | 27 |
| <a href="#">StopData</a>        | 28 |
| <a href="#">StopTimeData</a>    | 30 |
| <a href="#">Time</a>            | 34 |
| <a href="#">TripData</a>        | 38 |



## Chapter 3

# Class Documentation

### 3.1 DataBase Class Reference

```
#include <DataBase.h>
```

#### Public Types

- enum [LoadMethod](#) { **JSON** = 0, **GTFS** = 1, **MULTJSON** = 2 }

#### Public Member Functions

- [DataBase](#) ([DataBase::LoadMethod](#) method, std::string path)
- void [printTimeTable](#) ()
- bool [isValid](#) ()

#### Public Attributes

- std::vector< [RouteData](#) > [routes](#)
- std::vector< [TripData](#) > [trips](#)
- std::vector< [StopData](#) > [stops](#)
- std::vector< [StopTimeData](#) > [stopTimes](#)
- std::vector< [ServiceData](#) > [services](#)

#### 3.1.1 Detailed Description

Database class, used to loading from files and then being converted into [Network](#) object. Provides input from gtfs or json formats.

#### 3.1.2 Member Enumeration Documentation

##### 3.1.2.1 enum [DataBase::LoadMethod](#)

Enum defining loading method being used.

### 3.1.3 Constructor & Destructor Documentation

#### 3.1.3.1 DataBase::DataBase ( DataBase::LoadMethod *method*, std::string *path* )

Constructor.

## Parameters

|               |   |
|---------------|---|
| <i>method</i> | Defines load method being used.   |
| <i>path</i>   | Provides path to files being loaded. If one file is being used it needs to get path to file, otherwise - to directory containing files. |

## 3.1.4 Member Function Documentation

## 3.1.4.1 bool DataBase::isValid ( )

Method checking validity of loaded database.

## Returns

true if all vectors got populated with data, false otherwise.

## 3.1.4.2 void DataBase::printTimeTable ( )

Dunno...

## 3.1.5 Member Data Documentation

## 3.1.5.1 std::vector&lt;RouteData&gt; DataBase::routes

std::vector object containing loaded [RouteData](#).

## 3.1.5.2 std::vector&lt;ServiceData&gt; DataBase::services

std::vector object containing loaded [ServiceData](#).

## 3.1.5.3 std::vector&lt;StopData&gt; DataBase::stops

std::vector object containing loaded [StopData](#).

## 3.1.5.4 std::vector&lt;StopTimeData&gt; DataBase::stopTimes

std::vector object containing loaded [StopTimeData](#).

## 3.1.5.5 std::vector&lt;TripData&gt; DataBase::trips

std::vector object containing loaded [TripData](#).

The documentation for this class was generated from the following files:

- /home/vka/Workspace/RouteFinder/src/db/DataBase.h
- /home/vka/Workspace/RouteFinder/src/db/DataBase.cpp

## 3.2 DataReader Class Reference

```
#include <DataReader.h>
```

## Static Public Member Functions

- static `std::vector< RouteData > readRoutes` (`std::string filename`, `bool oneFile=true`)
- static `std::vector< StopData > readStops` (`std::string filename`, `bool oneFile=true`)
- static `std::vector< TripData > readTrips` (`std::string filename`, `bool oneFile=true`)
- static `std::vector< StopTimeData > readStopTimes` (`std::string filename`, `bool oneFile=true`)
- static `std::vector< ServiceData > readServices` (`std::string filename`, `bool oneFile=true`)

### 3.2.1 Detailed Description

[DataReader](#) reads data from one or multiple json files. If `oneFile` is set, functions assume that whole database is contained in given file, otherwise, assume given file contains only necessary data and so `std::vector sth = root;` (no `root['sth']` is needed.)

### 3.2.2 Member Function Documentation

#### 3.2.2.1 `std::vector< RouteData > DataReader::readRoutes ( std::string filename, bool oneFile = true ) [static]`

Function loading routes from given file.

##### Parameters

|                 |   |
|-----------------|---|
| <i>filename</i> | path to file.                                     |
| <i>oneFile</i>  | Defines if all data is being loaded from oneFile. |

##### Returns

vector containing [RouteData](#).

#### 3.2.2.2 `std::vector< ServiceData > DataReader::readServices ( std::string filename, bool oneFile = true ) [static]`

Function loading services from given file.

##### Parameters

|                 |   |
|-----------------|---|
| <i>filename</i> | path to file.                                     |
| <i>oneFile</i>  | Defines if all data is being loaded from oneFile. |

##### Returns

vector containing [ServiceData](#).

#### 3.2.2.3 `std::vector< StopData > DataReader::readStops ( std::string filename, bool oneFile = true ) [static]`

Function loading stops from given file.

##### Parameters

|                 |   |
|-----------------|---|
| <i>filename</i> | path to file.                                     |
| <i>oneFile</i>  | Defines if all data is being loaded from oneFile. |

##### Returns

vector containing [StopData](#).



3.2.2.4 `std::vector< StopTimeData > DataReader::readStopTimes ( std::string filename, bool oneFile = true )`  
[static]

Function loading stop times from given file.

**Parameters**

|                 |   |
|-----------------|---|
| <i>filename</i> | path to file.                                     |
| <i>oneFile</i>  | Defines if all data is being loaded from oneFile. |

**Returns**

vector containing [StopTimeData](#).

**3.2.2.5** `std::vector< TripData > DataReader::readTrips ( std::string filename, bool oneFile = true ) [static]`

Function loading trips from given file.

**Parameters**

|                 |   |
|-----------------|---|
| <i>filename</i> | path to file.                                     |
| <i>oneFile</i>  | Defines if all data is being loaded from oneFile. |

**Returns**

vector containing [TripData](#).

The documentation for this class was generated from the following files:

- /home/vka/Workspace/RouteFinder/src/db/lib/DataReader.h
- /home/vka/Workspace/RouteFinder/src/db/lib/DataReader.cpp

## 3.3 Edge Class Reference

```
#include <Edge.h>
```

**Public Member Functions**

- [Edge](#) (unsigned int id, [Node](#) \*start, [Node](#) \*end, const double weight, const TransportType type)
- unsigned int [getID](#) () const
- double [getWeight](#) () const
- const [Node](#) \* [getStartNode](#) () const
- const [Node](#) \* [getEndNode](#) () const
- TransportType [getType](#) () const
- void [setWeight](#) (double w)
- void [setType](#) (TransportType t)
- bool [operator==](#) (const [Edge](#) &e) const
- bool [operator!=](#) (const [Edge](#) &e) const
- bool [operator<](#) (const [Edge](#) &e) const
- [Edge](#) & [operator=](#) (const [Edge](#) &e)

**Friends**

- std::ostream & [operator<<](#) (std::ostream &s, const [Edge](#) &e)

### 3.3.1 Detailed Description

Object used for storage of one connections. Includes info about start and end positions, type of connection and weight of it.

### 3.3.2 Constructor & Destructor Documentation

#### 3.3.2.1 `Edge::Edge ( unsigned int id, Node * start, Node * end, const double weight, const TransportType type )`

Constructor of [Edge](#) object.

Parameters

|               |  |
|---------------|--|
| <i>id</i>     | Identificator of object.                                       |
| <i>start</i>  | Pointer to <a href="#">Node</a> assigned as starting position. |
| <i>end</i>    | Pointer to <a href="#">Node</a> assigned as ending position.   |
| <i>weight</i> | Given weight.  |
| <i>type</i>   | Enumerated value describing type of route.                     |

### 3.3.3 Member Function Documentation

#### 3.3.3.1 `const Node * Edge::getEndNode ( ) const`

Returns

Returns pointer to ending [Node](#).

#### 3.3.3.2 `unsigned int Edge::getID ( ) const`

Returns

Returns id of itself.

#### 3.3.3.3 `const Node * Edge::getStartNode ( ) const`

Returns

Returns pointer to starting [Node](#).

#### 3.3.3.4 `TransportType Edge::getType ( ) const`

Returns

Returns TransportType enum value describing type.

#### 3.3.3.5 `double Edge::getWeight ( ) const`

Returns

Returns weight of itself.

#### 3.3.3.6 `bool Edge::operator!= ( const Edge & e ) const`

Compares itself id with given edge id.

## Parameters

|          |                              |
|----------|------------------------------|
| <i>e</i> | Given <a href="#">Edge</a> . |
|----------|------------------------------|

## Returns

True if ids are not equal, false otherwise.

**3.3.3.7 bool Edge::operator< ( const Edge & e ) const**

Compares itself id with given edge id. Method used in [Network](#) class.

## Parameters

|          |                              |
|----------|------------------------------|
| <i>e</i> | Given <a href="#">Edge</a> . |
|----------|------------------------------|

## Returns

True if this->id is smaller than e.id, false otherwise.

**3.3.3.8 Edge & Edge::operator= ( const Edge & e )**

Assign operator.

## Parameters

|          |                              |
|----------|------------------------------|
| <i>e</i> | Given <a href="#">Edge</a> . |
|----------|------------------------------|

## Returns

Reference to itself.

**3.3.3.9 bool Edge::operator== ( const Edge & e ) const**

Compares itself id with given edge id.

## Parameters

|          |                              |
|----------|------------------------------|
| <i>e</i> | Given <a href="#">Edge</a> . |
|----------|------------------------------|

## Returns

True if ids are equal, false otherwise.

**3.3.3.10 void Edge::setType ( TransportType t )**

Sets type to given.

## Parameters

|          |   |
|----------|---|
| <i>t</i> | Given type value. Should be not equal to UNKNOWN. |
|----------|---|

**3.3.3.11 void Edge::setWeight ( double w )**

Sets weight to given.

## Parameters

|          |   |
|----------|---|
| <i>w</i> | Given weight value. Should be greater than 0. |
|----------|---|

## 3.3.4 Friends And Related Function Documentation

3.3.4.1 `std::ostream& operator<< ( std::ostream & s, const Edge & e )` [*friend*]

Operator used for console debug purposes.

## Parameters

|          |   |
|----------|---|
| <i>s</i> | Stream which is used for output.                  |
| <i>e</i> | <a href="#">Edge</a> on which operator is called. |

## Returns

Given stream.

The documentation for this class was generated from the following files:

- /home/vka/Workspace/RouteFinder/src/graph/Edge.h
- /home/vka/Workspace/RouteFinder/src/graph/Edge.cpp

## 3.4 GTFSReader Class Reference

```
#include <GTFSReader.h>
```

## Public Member Functions

- void [readGTFS](#) (std::string filename)
- std::vector< [RouteData](#) > [getRoutes](#) ()
- std::vector< [StopData](#) > [getStops](#) ()
- std::vector< [TripData](#) > [getTrips](#) ()
- std::vector< [StopTimeData](#) > [getStopTimes](#) ()
- std::vector< [ServiceData](#) > [getServices](#) ()

## Friends

- std::ostream & [operator<<](#) (std::ostream &stream, const [GTFSReader](#) &reader)

## 3.4.1 Detailed Description

[GTFSReader](#) reads data from GTFS format zip archive. It is compatible with [DataReader](#) class.

## 3.4.2 Member Function Documentation

3.4.2.1 `std::vector< RouteData > GTFSReader::getRoutes ( )`

## Returns

vector containing [RouteData](#).

#### 3.4.2.2 `std::vector< ServiceData > GTFSReader::getServices ( )`

##### Returns

vector containing [ServiceData](#).

#### 3.4.2.3 `std::vector< StopData > GTFSReader::getStops ( )`

##### Returns

vector containing [StopData](#).

#### 3.4.2.4 `std::vector< StopTimeData > GTFSReader::getStopTimes ( )`

##### Returns

vector containing [StopTimeData](#).

#### 3.4.2.5 `std::vector< TripData > GTFSReader::getTrips ( )`

##### Returns

vector containing [TripData](#).

#### 3.4.2.6 `void GTFSReader::readGTFS ( std::string filename )`

Unpacks gtfs archive, creates network, deletes created in progress files.

##### Parameters

|                 |                    |
|-----------------|--------------------|
| <i>filename</i> | Path to gtfs file. |
|-----------------|--------------------|

### 3.4.3 Friends And Related Function Documentation

#### 3.4.3.1 `std::ostream& operator<< ( std::ostream & stream, const GTFSReader & reader )` [friend]

Helper output function.

##### Parameters

|               |                  |
|---------------|------------------|
| <i>stream</i> | Stream           |
| <i>reader</i> | Outputed object. |

##### Returns

Reference to given stream.

The documentation for this class was generated from the following files:

- /home/vka/Workspace/RouteFinder/src/db/lib/GTFSReader.h
- /home/vka/Workspace/RouteFinder/src/db/lib/GTFSReader.cpp

## 3.5 Network Class Reference

```
#include <Network.h>
```

## Public Member Functions

- [Network](#) ()
- [Network](#) (std::string f)
- [~Network](#) ()
- [Network](#) (DataBase &dataB)
- void [loadFromFile](#) (std::string f)
- void [setSover](#) ([Solver](#) \*s)
- [Route](#) \* [findRouteBetween](#) (const [Node](#) \*start, const [Node](#) \*end, const unsigned int maxSwitches)
- std::list< [Node](#) \* > [getAllNodes](#) ()

## Friends

- std::ostream & [operator<<](#) (std::ostream &s, const [Network](#) &n)

### 3.5.1 Detailed Description

main class, contains information about nodes and edges between them. Should be created from file containing data in GTFS or other format. //todo loadFromFile method should load "db/db.ext" file and save it to inner variables.

### 3.5.2 Constructor & Destructor Documentation

#### 3.5.2.1 [Network::Network](#) ( )

[Network](#) object constructor. If this constructor is called, [loadFromFile](#) method need to be called after.

#### 3.5.2.2 [Network::Network](#) ( std::string f )

[Network](#) object constructor in which [Network::loadFromFile\(\)](#) method is being called.

##### Parameters

|   |   |
|---|---|
| f | Name of file from which database is loaded. |
|---|---|

#### 3.5.2.3 [Network::~~Network](#) ( )

Destructs all objects in [Network](#) and itself.

#### 3.5.2.4 [Network::Network](#) ( DataBase & dataB )

Creates [Network](#) from database

### 3.5.3 Member Function Documentation

#### 3.5.3.1 [Route](#) \* [Network::findRouteBetween](#) ( const [Node](#) \* start, const [Node](#) \* end, const unsigned int maxSwitches )

Searches for [Route](#) beetween two given points.

**Parameters**

|                    |  |
|--------------------|--|
| <i>start</i>       | Start <a href="#">Node</a> .                               |
| <i>end</i>         | End <a href="#">Node</a> .                                 |
| <i>maxSwitches</i> | Defines number of maximum transfers permitted during trip. |

**Returns**

Pointer to [Route](#) between given nodes, NULL if no route can be found.

**3.5.3.2 `std::list< Node * > Network::getAllNodes ( )`**

This function is necessary for GUI. //todo

**Returns**

Returns some kind of stl container in which all nodes are stored in alphabetical order.

**3.5.3.3 `void Network::loadFromFile ( std::string f )`**

Load database entries from given file.

**Parameters**

|          |   |
|----------|---|
| <i>f</i> | Filename from which database is being loaded. |
|----------|---|

**3.5.3.4 `void Network::setSover ( Solver * s )`**

Set solved used in [Network::findRouteBetween\(\)](#) method.

**Parameters**

|          |   |
|----------|---|
| <i>s</i> | Pointer to <a href="#">Solver</a> being used. |
|----------|---|

**3.5.4 Friends And Related Function Documentation****3.5.4.1 `std::ostream& operator<< ( std::ostream & s, const Network & n )` [[friend](#)]**

Used of debug in console purposes.

**Parameters**

|          |   |
|----------|---|
| <i>s</i> | Stream used for output.                             |
| <i>n</i> | Reference to <a href="#">Network</a> being printed. |

**Returns**

Given stream.

The documentation for this class was generated from the following files:

- /home/vka/Workspace/RouteFinder/src/graph/Network.h
- /home/vka/Workspace/RouteFinder/src/graph/Network.cpp



## 3.6 Node Class Reference

```
#include <Node.h>
```

### Public Member Functions

- [Node](#) (unsigned int id, std::string name, double lon, double lat)
- double [getLongitude](#) () const
- double [getLatitude](#) () const
- unsigned int [getID](#) () const
- std::string [getName](#) () const
- bool [operator==](#) (const [Node](#) &n) const
- bool [operator!=](#) (const [Node](#) &n) const
- bool [operator<](#) (const [Node](#) &n) const
- [Node](#) & [operator=](#) (const [Node](#) &n)

### Friends

- std::ostream & [operator<<](#) (std::ostream &s, const [Node](#) &n)

### 3.6.1 Detailed Description

primary element. Contains info about position and name of itself.

### 3.6.2 Constructor & Destructor Documentation

#### 3.6.2.1 [Node::Node](#) ( unsigned int *id*, std::string *name*, double *lon*, double *lat* )

Constructs new [Node](#) object.

##### Parameters

|             |                      |
|-------------|----------------------|
| <i>id</i>   | Id of an node.       |
| <i>name</i> | Name of node (stop). |
| <i>lon</i>  | Longitude coord.     |
| <i>lat</i>  | Latitude coord.      |

### 3.6.3 Member Function Documentation

#### 3.6.3.1 unsigned int [Node::getID](#) ( ) const

##### Returns

Returns id of itself.

#### 3.6.3.2 double [Node::getLatitude](#) ( ) const

##### Returns

Returns latitude as double.

### 3.6.3.3 double Node::getLongitude ( ) const

#### Returns

Returns longitude as double.

### 3.6.3.4 std::string Node::getName ( ) const

#### Returns

Returns string containing name.

### 3.6.3.5 bool Node::operator!= ( const Node & n ) const

Operator compares id of this and given node.

#### Parameters

|          |   |
|----------|---|
| <i>n</i> | <a href="#">Node</a> which is being compared. |
|----------|---|

#### Returns

False if ids are equal, true otherwise.

### 3.6.3.6 bool Node::operator< ( const Node & n ) const

Operator used in sets in [Network](#) class. Compares ids.

#### Parameters

|          |   |
|----------|---|
| <i>n</i> | <a href="#">Node</a> which is being compared. |
|----------|---|

#### Returns

True if this->id is smaller than n.id, false otherwise.

### 3.6.3.7 Node & Node::operator= ( const Node & n )

Copies params from given [Node](#) to itself.

#### Parameters

|          |   |
|----------|---|
| <i>n</i> | <a href="#">Node</a> which is being copied. |
|----------|---|

#### Returns

Reference to itself.

### 3.6.3.8 bool Node::operator== ( const Node & n ) const

Operator compares id of this and given node.

## Parameters

|          |   |
|----------|---|
| <i>n</i> | <a href="#">Node</a> which is being compared. |
|----------|---|

## Returns

True if ids are equal, false otherwise.

## 3.6.4 Friends And Related Function Documentation

3.6.4.1 `std::ostream& operator<< ( std::ostream & s, const Node & n )` [[friend](#)]

Operator used for console debug purposes.

## Parameters

|          |   |
|----------|---|
| <i>s</i> | Stream which is used for output.                  |
| <i>n</i> | <a href="#">Node</a> on which operator is called. |

## Returns

Given stream.

The documentation for this class was generated from the following files:

- `/home/vka/Workspace/RouteFinder/src/graph/Node.h`
- `/home/vka/Workspace/RouteFinder/src/graph/Node.cpp`

## 3.7 Route Class Reference

```
#include <Route.h>
```

## Public Member Functions

- [Route](#) ()
- unsigned int [getLength](#) () const
- double [getWeight](#) () const
- bool [validate](#) () const
- bool [addEdge](#) (const [Edge](#) \*e)
- bool [switchEdge](#) (const [Edge](#) \*e)
- bool [switchRoute](#) ([Route](#) &r)
- const [Node](#) \* [getStartNode](#) () const
- const [Node](#) \* [getEndNode](#) () const
- bool [isNodeIn](#) (const [Node](#) \*n) const
- bool [isEdgeIn](#) (const [Edge](#) \*e) const
- bool [isConnectionBetween](#) (const [Node](#) \*start, const [Node](#) \*end) const
- std::list< const [Edge](#) \* >  
::const\_iterator [begin](#) ()
- std::list< const [Edge](#) \* >  
::const\_iterator [end](#) ()

## Friends

- std::ostream & [operator<<](#) (std::ostream &s, [Route](#) &r)

### 3.7.1 Detailed Description

contains information about route between two points Should be used in solver class.

### 3.7.2 Constructor & Destructor Documentation

#### 3.7.2.1 Route::Route ( )

Constructs object. No data is necessary.

### 3.7.3 Member Function Documentation

#### 3.7.3.1 bool Route::addEdge ( const Edge \* e )

Add one [Edge](#) to the end of [Route](#). Given edge must start in [Node](#), in which current route ends.

Parameters

|          |                        |
|----------|------------------------|
| <i>e</i> | Pointer to given edge. |
|----------|------------------------|

Returns

True if edge could be connected to route, false otherwise.

#### 3.7.3.2 std::list< const Edge \* >::const\_iterator Route::begin ( )

Returns

Returns iterator to the beginning of route.

#### 3.7.3.3 std::list< const Edge \* >::const\_iterator Route::end ( )

Returns

Returns iterator to point after last [Edge](#) in route.

#### 3.7.3.4 const Node \* Route::getEndNode ( ) const

Returns

pointer to [Node](#) on the end of [Route](#).

#### 3.7.3.5 unsigned int Route::getLength ( ) const

Returns

Returns length - number of [Edge](#) objects.

#### 3.7.3.6 const Node \* Route::getStartNode ( ) const

Returns

Returns pointer to [Node](#) on the beginning of [Route](#).

**3.7.3.7 double Route::getWeight ( ) const****Returns**

Returns sum of weights of [Edge](#) objects.

**3.7.3.8 bool Route::isConnectionBetween ( const Node \* start, const Node \* end ) const**

Checks for subroute between given nodes.

**Parameters**

|              |   |
|--------------|---|
| <i>start</i> | Pointer to <a href="#">Node</a> of start. |
| <i>end</i>   | Pointer to <a href="#">Node</a> of end.   |

**Returns**

True if there is subroute from start to end in route, false otherwise.

**3.7.3.9 bool Route::isEdgeln ( const Edge \* e ) const****Parameters**

|          |  |
|----------|--|
| <i>e</i> | Pointer to <a href="#">Edge</a> which is being searched for. |
|----------|--|

**Returns**

True if [Edge](#) is included in route, false otherwise.

**3.7.3.10 bool Route::isNodeIn ( const Node \* n ) const****Parameters**

|          |   |
|----------|---|
| <i>n</i> | Pointer to <a href="#">Node</a> which is being checked. |
|----------|---|

**Returns**

True if given node is currently in route. False otherwise.

**3.7.3.11 bool Route::switchEdge ( const Edge \* e )**

Switches given edge with one included in path,

**Parameters**

|          |  |
|----------|--|
| <i>e</i> | Pointer to <a href="#">Edge</a> . Its start <a href="#">Node</a> and end <a href="#">Node</a> must be same as start and end nodes of edge included in route. |
|----------|--|

**Returns**

True if switch was successful, false otherwise.

**3.7.3.12 bool Route::switchRoute ( Route & r )**

Switches part of [Route](#) with given route.

**Parameters**

|          |  |
|----------|--|
| <i>r</i> | Reference to subroute which needs to be inserted into object. It must to be correct (validate method is being called), and start and end of subroute must have corresponding values as start and end of some subroute inside current object. Length of switched subroutes do not need to be equal. |
|----------|--|

**Returns**

True if switch was successful, false otherwise.

**3.7.3.13 bool Route::validate ( ) const**

Checks if [Route](#) does not contain any loops and if all edges are connected. I.e. A->B and then B->C is ok, but A->B and C->D is wrong.

**Returns**

True if test is passed, false otherwise.

**3.7.4 Friends And Related Function Documentation****3.7.4.1 std::ostream& operator<< ( std::ostream & s, Route & r ) [friend]**

Used of debug in console purposes.

**Parameters**

|          |   |
|----------|---|
| <i>s</i> | Stream used for output.                           |
| <i>r</i> | Reference to <a href="#">Route</a> being printed. |

**Returns**

Given stream.

The documentation for this class was generated from the following files:

- /home/vka/Workspace/RouteFinder/src/graph/Route.h
- /home/vka/Workspace/RouteFinder/src/graph/Route.cpp

**3.8 RouteData Class Reference**

```
#include <RouteData.h>
```

**Public Member Functions**

- [RouteData](#) ()
- [RouteData](#) (const [RouteData](#) &src)
- [RouteData](#) (std::string name, unsigned int id)
- [~RouteData](#) ()
- [RouteData](#) & [operator=](#) (const [RouteData](#) &src)
- bool [operator==](#) (const [RouteData](#) &src)
- std::string [getName](#) () const
- unsigned int [getId](#) () const

## Friends

- `std::ostream & operator<< (std::ostream &output, const RouteData &src)`

### 3.8.1 Detailed Description

Class containing data about routes when database is being created. Then converted into [Route](#) object.

### 3.8.2 Constructor & Destructor Documentation

#### 3.8.2.1 RouteData::RouteData ( )

Default constructor.

#### 3.8.2.2 RouteData::RouteData ( const RouteData & src )

Copying constructor.

##### Parameters

|            |                      |
|------------|----------------------|
| <i>src</i> | Object being copied. |
|------------|----------------------|

#### 3.8.2.3 RouteData::RouteData ( std::string name, unsigned int id )

Constructor.

##### Parameters

|             |                |
|-------------|----------------|
| <i>name</i> | Name of route. |
| <i>id</i>   | id of route.   |

#### 3.8.2.4 RouteData::~RouteData ( )

Destructor.

### 3.8.3 Member Function Documentation

#### 3.8.3.1 unsigned int RouteData::getId ( ) const

##### Returns

return id value.

#### 3.8.3.2 std::string RouteData::getName ( ) const

##### Returns

Returns name value.

#### 3.8.3.3 RouteData & RouteData::operator= ( const RouteData & src )

Assignment operator.

## Parameters

|            |                                  |
|------------|----------------------------------|
| <i>src</i> | Reference object being assigned. |
|------------|----------------------------------|

## Returns

Reference to self.

### 3.8.3.4 bool RouteData::operator== ( const RouteData & *src* )

Comparison operator.

## Parameters

|            |                               |
|------------|-------------------------------|
| <i>src</i> | Reference to compared object. |
|------------|-------------------------------|

## Returns

True if name and is are equal.

## 3.8.4 Friends And Related Function Documentation

### 3.8.4.1 std::ostream& operator<< ( std::ostream & *output*, const RouteData & *src* ) [friend]

Helper output function.

## Parameters

|               |                  |
|---------------|------------------|
| <i>output</i> | Stream           |
| <i>src</i>    | Outputed object. |

## Returns

Reference to given stream.

The documentation for this class was generated from the following files:

- /home/vka/Workspace/RouteFinder/src/db/lib/RouteData.h
- /home/vka/Workspace/RouteFinder/src/db/lib/RouteData.cpp

## 3.9 ServiceData Class Reference

```
#include <ServiceData.h>
```

## Public Member Functions

- [ServiceData](#) ()
- [ServiceData](#) (const [ServiceData](#) &src)
- [ServiceData](#) (unsigned int id, std::string name)
- [~ServiceData](#) ()
- [ServiceData operator=](#) (const [ServiceData](#) src)
- bool [operator==](#) (const [ServiceData](#) src)
- unsigned int [getId](#) ()
- std::string [getName](#) ()



## Friends

- `std::ostream & operator<< (std::ostream &output, const ServiceData &src)`

### 3.9.1 Detailed Description

Class containing data about services when database is being created.

### 3.9.2 Constructor & Destructor Documentation

#### 3.9.2.1 ServiceData::ServiceData ( )

Default constructor.

#### 3.9.2.2 ServiceData::ServiceData ( const ServiceData & src )

Copying constructor.

##### Parameters

|            |                      |
|------------|----------------------|
| <i>src</i> | Object being copied. |
|------------|----------------------|

#### 3.9.2.3 ServiceData::ServiceData ( unsigned int *id*, std::string *name* )

Constructor.

##### Parameters

|             |                |
|-------------|----------------|
| <i>name</i> | Name of route. |
| <i>id</i>   | id of route.   |

#### 3.9.2.4 ServiceData::~ServiceData ( )

Destructor.

### 3.9.3 Member Function Documentation

#### 3.9.3.1 unsigned int ServiceData::getId ( )

##### Returns

Returns name value.

#### 3.9.3.2 std::string ServiceData::getName ( )

##### Returns

return id value.

#### 3.9.3.3 ServiceData ServiceData::operator= ( const ServiceData *src* )

Assignment operator.

**Parameters**

|            |                                  |
|------------|----------------------------------|
| <i>src</i> | Reference object being assigned. |
|------------|----------------------------------|

**Returns**

Reference to self.

**3.9.3.4 bool ServiceData::operator==( const ServiceData src )**

Comparison operator.

**Parameters**

|            |                               |
|------------|-------------------------------|
| <i>src</i> | Reference to compared object. |
|------------|-------------------------------|

**Returns**

True if name and is are equal.

**3.9.4 Friends And Related Function Documentation****3.9.4.1 std::ostream& operator<< ( std::ostream & output, const ServiceData & src ) [friend]**

Helper output function.

**Parameters**

|               |                  |
|---------------|------------------|
| <i>output</i> | Stream           |
| <i>src</i>    | Outputed object. |

**Returns**

Reference to given stream.

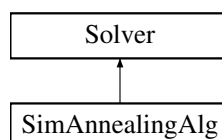
The documentation for this class was generated from the following files:

- /home/vka/Workspace/RouteFinder/src/db/lib/ServiceData.h
- /home/vka/Workspace/RouteFinder/src/db/lib/ServiceData.cpp

**3.10 SimAnnealingAlg Class Reference**

```
#include <SimAnnealingAlg.h>
```

Inheritance diagram for SimAnnealingAlg:

**Public Member Functions**

- virtual [Route](#) \* [solve](#) (const [Network](#) \*n)

### 3.10.1 Detailed Description

Simulated Annealing Algorithm used for finding routes. See doc folder for more information.

### 3.10.2 Member Function Documentation

#### 3.10.2.1 virtual Route\* SimAnnealingAlg::solve ( const Network \* n ) [virtual]

Method used in [Network](#) class for finding best connection between points.

##### Parameters

|          |  |
|----------|--|
| <i>n</i> | Pointer to <a href="#">Network</a> in which <a href="#">Route</a> is being searched for. |
|----------|--|

##### Returns

Pointer to found [Route](#), NULL if no route can be found.

Implements [Solver](#).

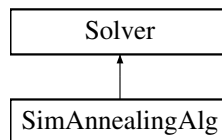
The documentation for this class was generated from the following file:

- /home/vka/Workspace/RouteFinder/src/algorithm/SimAnnealingAlg.h

## 3.11 Solver Class Reference

```
#include <Solver.h>
```

Inheritance diagram for Solver:



### Public Member Functions

- virtual [Route](#) \* [solve](#) (const [Network](#) \*n)=0

### 3.11.1 Detailed Description

wrapper class for solver algorithm. Those shall inherit from [Solver](#) class. [Solver](#) needs to implement solve method, which gets [Network](#) map as

### 3.11.2 Member Function Documentation

#### 3.11.2.1 virtual Route\* Solver::solve ( const Network \* n ) [pure virtual]

Method used in [Network](#) class for finding best connection between points. This method need to be implemented in any class inheriting from [Solver](#) class.

**Parameters**

|          |  |
|----------|--|
| <i>n</i> | Pointer to <a href="#">Network</a> in which <a href="#">Route</a> is being searched for. |
|----------|--|

**Returns**

Pointer to found [Route](#), NULL if no route can be found.

Implemented in [SimAnnealingAlg](#).

The documentation for this class was generated from the following file:

- /home/vka/Workspace/RouteFinder/src/algorithm/Solver.h

## 3.12 StopData Class Reference

```
#include <StopData.h>
```

**Public Member Functions**

- [StopData](#) ()
- [StopData](#) (const [StopData](#) &src)
- [StopData](#) (std::string name, unsigned int id, double lat, double lng)
- [~StopData](#) ()
- [StopData operator=](#) (const [StopData](#) src)
- bool [operator==](#) (const [StopData](#) src)
- std::string [getName](#) () const
- unsigned int [getId](#) () const
- double [getLat](#) () const
- double [getLng](#) () const

**Friends**

- std::ostream & [operator<<](#) (std::ostream &output, const [StopData](#) &src)

### 3.12.1 Detailed Description

Class containing data about routes when database is being created. Then converted into [Route](#) object.

### 3.12.2 Constructor & Destructor Documentation

#### 3.12.2.1 StopData::StopData ( )

Default constructor.

#### 3.12.2.2 StopData::StopData ( const StopData & src )

Copying constructor.

## Parameters

|            |                      |
|------------|----------------------|
| <i>src</i> | Object being copied. |
|------------|----------------------|

3.12.2.3 StopData::StopData ( std::string *name*, unsigned int *id*, double *lat*, double *lng* )

Constructor.

## Parameters

|                |                 |
|----------------|-----------------|
| <i>name</i>    | Name of route.  |
| <i>id</i>      | id of route.    |
| <i>lat,lng</i> | coords of stop. |

## 3.12.2.4 StopData::~~StopData ( )

Destructor.

## 3.12.3 Member Function Documentation

## 3.12.3.1 unsigned int StopData::getId ( ) const

## Returns

return id value.

## 3.12.3.2 double StopData::getLat ( ) const

## Returns

return latitude value.

## 3.12.3.3 double StopData::getLng ( ) const

## Returns

return longitude value.

## 3.12.3.4 std::string StopData::getName ( ) const

## Returns

Returns name value.

3.12.3.5 StopData StopData::operator= ( const StopData *src* )

Assignment operator.

**Parameters**

|            |                                  |
|------------|----------------------------------|
| <i>src</i> | Reference object being assigned. |
|------------|----------------------------------|

**Returns**

Reference to self.

**3.12.3.6 bool StopData::operator== ( const StopData src )**

Comparison operator.

**Parameters**

|            |                               |
|------------|-------------------------------|
| <i>src</i> | Reference to compared object. |
|------------|-------------------------------|

**Returns**

True if name and is are equal.

**3.12.4 Friends And Related Function Documentation****3.12.4.1 std::ostream& operator<< ( std::ostream & output, const StopData & src ) [friend]**

Helper output function.

**Parameters**

|               |                  |
|---------------|------------------|
| <i>output</i> | Stream           |
| <i>src</i>    | Outputed object. |

**Returns**

Reference to given stream.

The documentation for this class was generated from the following files:

- /home/vka/Workspace/RouteFinder/src/db/lib/StopData.h
- /home/vka/Workspace/RouteFinder/src/db/lib/StopData.cpp

**3.13 StopTimeData Class Reference**

```
#include <StopTimeData.h>
```

**Public Member Functions**

- [StopTimeData](#) ()
- [StopTimeData](#) (const [StopTimeData](#) &src)
- [StopTimeData](#) (unsigned int id, unsigned int stopID, unsigned int serviceID, unsigned int tripID, std::string name)
- [~StopTimeData](#) ()
- [StopTimeData operator=](#) (const [StopTimeData](#) src)
- bool [operator==](#) (const [StopTimeData](#) src)
- unsigned int [getId](#) () const
- unsigned int [getServiceId](#) () const

- unsigned int [getStopId](#) () const
- unsigned int [getTripId](#) () const
- Time [getStopTime](#) () const
- std::string [getName](#) () const

## Friends

- std::ostream & [operator<<](#) (std::ostream &output, const [StopTimeData](#) &src)

### 3.13.1 Detailed Description

Class containing data about stopTimes when database is being created.

### 3.13.2 Constructor & Destructor Documentation

#### 3.13.2.1 StopTimeData::StopTimeData ( )

Default constructor.

#### 3.13.2.2 StopTimeData::StopTimeData ( const StopTimeData & src )

Copying constructor.

Parameters

|            |                      |
|------------|----------------------|
| <i>src</i> | Object being copied. |
|------------|----------------------|

#### 3.13.2.3 StopTimeData::StopTimeData ( unsigned int *id*, unsigned int *stopID*, unsigned int *serviceID*, unsigned int *tripID*, std::string *name* )

Constructor.

Parameters

|                  |   |
|------------------|---|
| <i>name</i>      | Name of route.  |
| <i>id</i>        | id of route.  |
| <i>stopID</i>    | ID of <a href="#">StopData</a> object of which timetable info contains this object. |
| <i>serviceID</i> | id of <a href="#">ServiceData</a> object.   |
| <i>tripID</i>    | <a href="#">TripData</a> object being mentioned.                                    |

#### 3.13.2.4 StopTimeData::~StopTimeData ( )

Destructor.

### 3.13.3 Member Function Documentation

#### 3.13.3.1 unsigned int StopTimeData::getId ( ) const

Returns

Returns id value.

### 3.13.3.2 `std::string StopTimeData::getName ( ) const`

#### Returns

Returns name value.

### 3.13.3.3 `unsigned int StopTimeData::getServiceId ( ) const`

#### Returns

Returns [ServiceData](#) id value.

### 3.13.3.4 `unsigned int StopTimeData::getStopId ( ) const`

#### Returns

Returns [StopData](#) id value.

### 3.13.3.5 `Time StopTimeData::getStopTime ( ) const`

#### Returns

Returns [Time](#) value.

### 3.13.3.6 `unsigned int StopTimeData::getTripld ( ) const`

#### Returns

Returns [TripData](#) id value.

### 3.13.3.7 `StopTimeData StopTimeData::operator= ( const StopTimeData src )`

Assignment operator.

#### Parameters

|            |                                  |
|------------|----------------------------------|
| <i>src</i> | Reference object being assigned. |
|------------|----------------------------------|

#### Returns

Reference to self.

### 3.13.3.8 `bool StopTimeData::operator== ( const StopTimeData src )`

Comparation operator.

#### Parameters

|            |                               |
|------------|-------------------------------|
| <i>src</i> | Reference to compared object. |
|------------|-------------------------------|

#### Returns

True if name and is are equal.



### 3.13.4 Friends And Related Function Documentation

3.13.4.1 `std::ostream& operator<< ( std::ostream & output, const StopTimeData & src )` [friend]

Helper output function.

## Parameters

|               |                  |
|---------------|------------------|
| <i>output</i> | Stream           |
| <i>src</i>    | Outputed object. |

## Returns

Reference to given stream.

The documentation for this class was generated from the following files:

- /home/vka/Workspace/RouteFinder/src/db/lib/StopTimeData.h
- /home/vka/Workspace/RouteFinder/src/db/lib/StopTimeData.cpp

## 3.14 Time Class Reference

```
#include <Time.h>
```

### Public Member Functions

- [Time](#) ()
- [Time](#) (unsigned int hour, unsigned int minute)
- [Time](#) (unsigned int minutes)
- [Time](#) (const [Time](#) &src)
- [~Time](#) ()
- [Time operator=](#) ([Time](#) src)
- bool [operator==](#) (const [Time](#) src)
- bool [operator!=](#) (const [Time](#) src)
- bool [operator>](#) (const [Time](#) src)
- bool [operator<](#) (const [Time](#) src)
- [Time operator+](#) (const [Time](#) src)
- [Time operator-](#) (const [Time](#) src)
- [operator int](#) ()

### Friends

- std::ostream & [operator<<](#) (std::ostream &output, const [Time](#) src)

#### 3.14.1 Detailed Description

Class providing time functions used in timetable generation.

#### 3.14.2 Constructor & Destructor Documentation

##### 3.14.2.1 [Time::Time](#) ( )

Default constructor.

##### 3.14.2.2 [Time::Time](#) ( unsigned int *hour*, unsigned int *minute* )

Constructor.

## Parameters

|               |         |
|---------------|---------|
| <i>hour</i>   | Hour.   |
| <i>minute</i> | Minute. |

3.14.2.3 Time::Time ( unsigned int *minutes* )

Constructor creating object pointing to time minutes ahead of midnight.

## Parameters

|                |                        |
|----------------|------------------------|
| <i>minutes</i> | Minutes from midnight. |
|----------------|------------------------|

3.14.2.4 Time::Time ( const Time & *src* )

Copying constructor.

## Parameters

|            |                      |
|------------|----------------------|
| <i>src</i> | Object being copied. |
|------------|----------------------|

## 3.14.2.5 Time::~Time ( )

Destructor.

## 3.14.3 Member Function Documentation

## 3.14.3.1 Time::operator int ( )

Converts [Time](#) to int.

3.14.3.2 bool Time::operator!= ( const Time *src* )

Comparison operator.

## Parameters

|            |                               |
|------------|-------------------------------|
| <i>src</i> | Reference to compared object. |
|------------|-------------------------------|

## Returns

True if given hours are not equal.

3.14.3.3 Time Time::operator+ ( const Time *src* )

Addition operator. Creates object pointing to time created by adding two given hours.

## Parameters

|            |                                    |
|------------|------------------------------------|
| <i>src</i> | given <a href="#">Time</a> object. |
|------------|------------------------------------|

## Returns

[Time](#) object.

#### 3.14.3.4 Time Time::operator- ( const Time *src* )

Subtraction operator. Creates object pointing to time created by subtracting two given hours.

## Parameters

|            |                                    |
|------------|------------------------------------|
| <i>src</i> | given <a href="#">Time</a> object. |
|------------|------------------------------------|

## Returns

[Time](#) object.

**3.14.3.5** `bool Time::operator< ( const Time src )`

Greater than operator.

## Returns

True if given "this" time is sooner from midnight than src time.

**3.14.3.6** `Time Time::operator= ( Time src )`

Assignment operator.

## Parameters

|            |                                  |
|------------|----------------------------------|
| <i>src</i> | Reference object being assigned. |
|------------|----------------------------------|

## Returns

Reference to self.

**3.14.3.7** `bool Time::operator== ( const Time src )`

Comparison operator.

## Parameters

|            |                               |
|------------|-------------------------------|
| <i>src</i> | Reference to compared object. |
|------------|-------------------------------|

## Returns

True if given hours are equal.

**3.14.3.8** `bool Time::operator> ( const Time src )`

Greater than operator.

## Returns

True if given "this" time is later from midnight than src time.

**3.14.4 Friends And Related Function Documentation****3.14.4.1** `std::ostream& operator<< ( std::ostream & output, const Time src )` [*friend*]

Helper output function.

## Parameters

|               |                  |
|---------------|------------------|
| <i>output</i> | Stream           |
| <i>src</i>    | Outputed object. |

## Returns

Reference to given stream.

The documentation for this class was generated from the following files:

- /home/vka/Workspace/RouteFinder/src/db/lib/Time.h
- /home/vka/Workspace/RouteFinder/src/db/lib/Time.cpp

## 3.15 TripData Class Reference

```
#include <TripData.h>
```

### Public Member Functions

- [TripData](#) ()
- [TripData](#) (const [TripData](#) &src)
- [TripData](#) (unsigned int id, unsigned int routeld, std::string name, std::vector< int > stopSec)
- [~TripData](#) ()
- [TripData operator=](#) (const [TripData](#) src)
- bool [operator==](#) (const [TripData](#) src)
- unsigned int [getId](#) () const
- unsigned int [getRouteld](#) () const
- std::vector< int > [getStopSec](#) () const
- std::string [getName](#) () const

### Friends

- std::ostream & [operator<<](#) (std::ostream &output, const [TripData](#) &src)

### 3.15.1 Detailed Description

Class containing data about Trip when database is being created. Then converted into Trip object.

### 3.15.2 Constructor & Destructor Documentation

#### 3.15.2.1 [TripData::TripData](#) ( )

Default constructor.

#### 3.15.2.2 [TripData::TripData](#) ( const [TripData](#) & *src* )

Copying constructor.

## Parameters

|            |                      |
|------------|----------------------|
| <i>src</i> | Object being copied. |
|------------|----------------------|

3.15.2.3 TripData::TripData ( unsigned int *id*, unsigned int *routeId*, std::string *name*, std::vector< int > *stopSec* )

Constructor.

## Parameters

|                |                                     |
|----------------|-------------------------------------|
| <i>name</i>    | Name of route.                      |
| <i>id</i>      | id of route.                        |
| <i>routeId</i> | <a href="#">RouteData</a> id value. |
| <i>stopSec</i> | Vector containing stop sequence.    |

## 3.15.2.4 TripData::~~TripData ( )

Destructor.

## 3.15.3 Member Function Documentation

## 3.15.3.1 unsigned int TripData::getId ( ) const

## Returns

Returns id value.

## 3.15.3.2 std::string TripData::getName ( ) const

## Returns

Returns name value.

## 3.15.3.3 unsigned int TripData::getRouteId ( ) const

## Returns

Returns [RouteData](#) id value.

## 3.15.3.4 std::vector&lt; int &gt; TripData::getStopSec ( ) const

## Returns

Returns stop sequence in vector<int>.

3.15.3.5 TripData TripData::operator= ( const TripData *src* )

Assignment operator.

## Parameters

|            |                                  |
|------------|----------------------------------|
| <i>src</i> | Reference object being assigned. |
|------------|----------------------------------|

## Returns

Reference to self.

**3.15.3.6 bool TripData::operator== ( const TripData *src* )**

Comparison operator.

## Parameters

|            |                               |
|------------|-------------------------------|
| <i>src</i> | Reference to compared object. |
|------------|-------------------------------|

## Returns

True if name and is are equal.

**3.15.4 Friends And Related Function Documentation****3.15.4.1 std::ostream& operator<< ( std::ostream & *output*, const TripData & *src* ) [friend]**

Helper output function.

## Parameters

|               |                  |
|---------------|------------------|
| <i>output</i> | Stream           |
| <i>src</i>    | Outputed object. |

## Returns

Reference to given stream.

The documentation for this class was generated from the following files:

- /home/vka/Workspace/RouteFinder/src/db/lib/TripData.h
- /home/vka/Workspace/RouteFinder/src/db/lib/TripData.cpp



# Index

begin  
    Route, [20](#)

Edge, [10](#)  
    Edge, [11](#)  
    operator!=, [11](#)  
    operator<, [12](#)  
    operator<<, [13](#)  
    operator=, [12](#)  
    operator==, [12](#)

end  
    Route, [20](#)

Network, [14](#)  
    Network, [15](#)  
    operator<<, [16](#)

Node, [17](#)  
    Node, [17](#)  
    operator!=, [18](#)  
    operator<, [18](#)  
    operator<<, [19](#)  
    operator=, [18](#)  
    operator==, [18](#)

operator int  
    Time, [35](#)

operator!=  
    Edge, [11](#)  
    Node, [18](#)  
    Time, [35](#)

operator<  
    Edge, [12](#)  
    Node, [18](#)  
    Time, [37](#)

operator<<  
    Edge, [13](#)  
    Network, [16](#)  
    Node, [19](#)  
    Route, [22](#)  
    Time, [37](#)

operator>  
    Time, [37](#)

operator+  
    Time, [35](#)

operator-  
    Time, [35](#)

operator=  
    Edge, [12](#)  
    Node, [18](#)  
    Time, [37](#)

operator==  
    Edge, [12](#)  
    Node, [18](#)  
    Time, [37](#)

Route, [19](#)  
    begin, [20](#)  
    end, [20](#)  
    operator<<, [22](#)  
    Route, [20](#)  
    validate, [22](#)

solve  
    Solver, [27](#)

Solver, [27](#)  
    solve, [27](#)

Time, [34](#)  
    operator int, [35](#)  
    operator!=, [35](#)  
    operator<, [37](#)  
    operator<<, [37](#)  
    operator>, [37](#)  
    operator+, [35](#)  
    operator-, [35](#)  
    operator=, [37](#)  
    operator==, [37](#)  
    Time, [34](#), [35](#)

validate  
    Route, [22](#)