

API Documentation

API Documentation

January 21, 2018

Contents

Contents	1
1 Package src	2
1.1 Modules	2
1.2 Functions	3
1.3 Variables	3
2 Module src.__main__	4
2.1 Variables	4
3 Module src.b_exceptions	5
3.1 Variables	5
3.2 Class BadFormatException	5
3.2.1 Methods	5
3.2.2 Properties	5
3.3 Class BadValueException	6
3.3.1 Methods	6
3.3.2 Properties	6
3.4 Class BenzlimException	7
3.4.1 Methods	7
3.4.2 Properties	7
3.5 Class PriceNotFoundException	8
3.5.1 Methods	8
3.5.2 Properties	8
3.6 Class StationNotFoundException	9
3.6.1 Methods	9
3.6.2 Properties	9
3.7 Class TrainingDataMissingException	10
3.7.1 Methods	10
3.7.2 Properties	10
4 Module src.benchmark	11
4.1 Functions	11
4.2 Variables	11
5 Package src.compat	12
5.1 Modules	12
5.2 Variables	12

6	Module <code>src.compat.py2</code>	13
6.1	Functions	13
6.2	Variables	13
7	Module <code>src.compat.py3</code>	14
7.1	Functions	14
8	Module <code>src.config</code>	15
8.1	Variables	15
8.2	Class Configuration	15
8.2.1	Methods	15
8.2.2	Class Variables	15
9	Package <code>src.coverage</code>	16
9.1	Functions	16
9.2	Variables	16
10	Package <code>src.dao</code>	17
10.1	Modules	17
10.2	Class CSVDAO	17
10.2.1	Methods	17
10.2.2	Properties	18
10.3	Class StationDAO	19
10.3.1	Methods	19
10.3.2	Properties	20
10.3.3	Class Variables	20
10.4	Class DBManager	20
10.4.1	Methods	20
10.4.2	Properties	21
10.4.3	Class Variables	21
11	Module <code>src.dao.csv_</code>	23
11.1	Variables	23
11.2	Class CSVDAO	23
11.2.1	Methods	23
11.2.2	Properties	24
12	Module <code>src.dao.db</code>	25
12.1	Class DBManager	25
12.1.1	Methods	25
12.1.2	Properties	26
12.1.3	Class Variables	26
12.2	Class StationDAO	26
12.2.1	Methods	26
12.2.2	Properties	27
12.2.3	Class Variables	27
13	Module <code>src.exceptions_</code>	29
13.1	Variables	29
13.2	Class BenzlimException	29
13.2.1	Methods	29
13.2.2	Properties	30
13.3	Class StationNotFoundException	30

13.3.1	Methods	30
13.3.2	Properties	31
13.4	Class PriceNotFoundException	31
13.4.1	Methods	31
13.4.2	Properties	32
13.5	Class TrainingDataMissingException	32
13.5.1	Methods	32
13.5.2	Properties	33
13.6	Class BadFormatException	33
13.6.1	Methods	33
13.6.2	Properties	34
13.7	Class BadValueException	34
13.7.1	Methods	34
13.7.2	Properties	35
14	Package src.prediction	36
14.1	Modules	36
14.2	Functions	36
14.3	Variables	36
15	Module src.prediction.classification	37
15.1	Variables	37
15.2	Class CSClassifier	37
15.2.1	Methods	37
15.2.2	Properties	37
15.3	Class Classifier	38
15.3.1	Methods	38
15.3.2	Properties	39
16	Module src.prediction.db	40
16.1	Functions	40
16.2	Variables	40
16.3	Class DBManager	40
16.3.1	Methods	40
16.3.2	Properties	41
16.3.3	Class Variables	41
16.4	Class PriceDAO	42
16.4.1	Methods	42
16.4.2	Class Variables	42
16.5	Class StationDAO	42
16.5.1	Methods	42
16.5.2	Class Variables	43
17	Module src.prediction.predict	44
17.1	Functions	44
17.2	Variables	45
18	Package src.routing	46
18.1	Modules	46
18.2	Functions	46
18.3	Variables	46

19 Module <code>src.routing.graph</code>	47
19.1 Variables	47
19.2 Class Graph	47
19.2.1 Methods	47
20 Module <code>src.routing.node</code>	48
20.1 Variables	48
20.2 Class Node	48
20.2.1 Methods	48
20.2.2 Properties	48
21 Package <code>src.tests</code>	49
21.1 Functions	49
21.2 Variables	49
22 Module <code>src.train</code>	50
22.1 Variables	50
22.2 Class Trainer	50
22.2.1 Methods	50
22.2.2 Properties	50
23 Module <code>src.utils</code>	51
23.1 Functions	51
23.2 Variables	51

1 Package src

Benzlim

1.1 Modules

- **__main__**: (Section 2, p. 4)
- **b_exceptions**: (Section 3, p. 5)
- **benchmark**: benchmark.py - Benchmarking tool
(Section 4, p. 11)
- **compat**: compat - Compatibility packages for ython2 and python3
(Section 5, p. 12)
 - **py2**: py2 - Python2 compatibility module
(Section 6, p. 13)
 - **py3**: py3 - Python3 compatibility module
(Section 7, p. 14)
- **config**: config.py - access benzlim's instance configuration
(Section 8, p. 15)
- **coverage**: (Section ??, p. ??)
- **coverage'**: coverage - Coverage informations generation about benzlim
(Section 9, p. 16)
- **dao**: dao - Data Access Object packages for IO tasks
(Section 10, p. 17)
 - **csv_**: csv_.py - read/write/investigate csv related files
(Section 11, p. 23)
 - **db**: db.py - access station related informations
(Section 12, p. 25)
- **exceptions_**: exceptions_.py - benzlim exceptions
(Section 13, p. 29)
- **prediction**: prediction - The benzlim core prediction
(Section 14, p. 36)
 - **classification**: classification.py - gas stations classification tools
(Section 15, p. 37)
 - **db**: (Section 16, p. 40)
 - **predict**: predict.py - core prediction tools
(Section 17, p. 44)
- **routing**: routing - gas tank strategy manager
(Section 18, p. 46)
 - **graph**: graph.y - Tank strategy optimizer for graph based routes
(Section 19, p. 47)
 - **node**: node.py - Nodes for graph based representation of gas stations in a route
(Section 20, p. 48)
- **tests**: tests - Tests runner
(Section 21, p. 49)
- **train**: train.py - manage the whole training
(Section 22, p. 50)
- **utils**: utils.py - usefool tools
(Section 23, p. 51)

1.2 Functions

<code>main()</code>

1.3 Variables

Name	Description
<code>--package--</code>	Value: 'src'
<code>lvl</code>	Value: 0

2 Module src.__main__

2.1 Variables

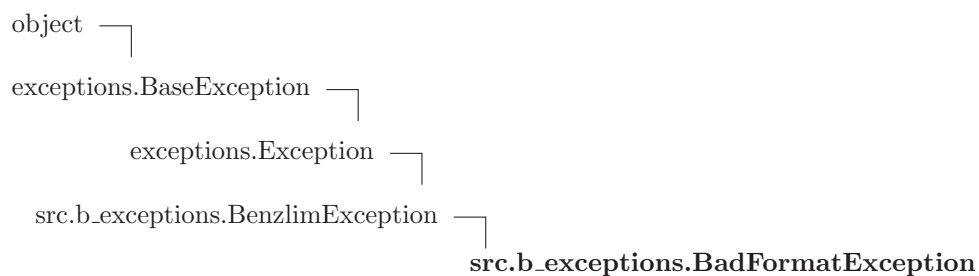
Name	Description
__package__	Value: None

3 Module *src.b_exceptions*

3.1 Variables

Name	Description
<code>--package--</code>	Value: None

3.2 Class *BadFormatException*



3.2.1 Methods

`--init--(self, *args, **kwargs)`
`x.--init--(...)` initializes x; see `help(type(x))` for signature
 Overrides: `object.--init--` `exitit`(inherited documentation)

Inherited from `exceptions.Exception`

`--new--()`

Inherited from `exceptions.BaseException`

`--delattr--()`, `--getattr--()`, `--getitem--()`, `--getslice--()`, `--reduce--()`, `--repr--()`,
`--setattr--()`, `--setstate--()`, `--str--()`, `--unicode--()`

Inherited from `object`

`--format--()`, `--hash--()`, `--reduce_ex--()`, `--sizeof--()`, `--subclasshook--()`

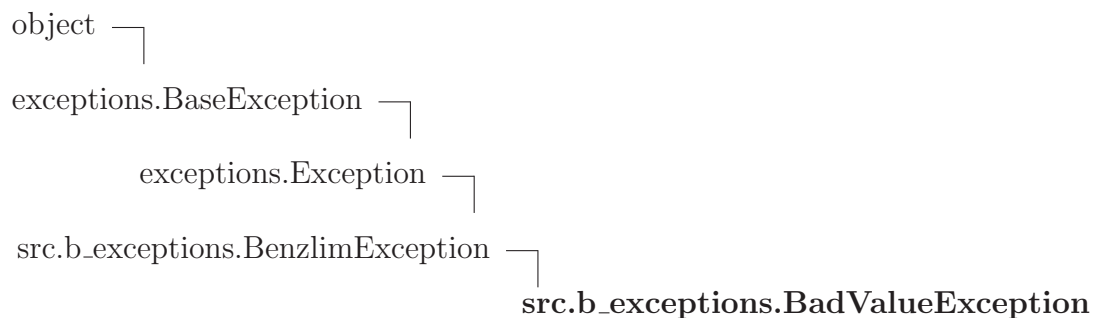
3.2.2 Properties

Name	Description
<i>Inherited from <code>exceptions.BaseException</code></i>	
args, message	
<i>Inherited from <code>object</code></i>	

continued on next page

Name	Description
<code>--class--</code>	

3.3 Class *BadValueException*



3.3.1 Methods

```

__init__(self, *args, **kwargs)
x.__init__(...) initializes x; see help(type(x)) for signature
Overrides: object.__init__ extit(inherited documentation)

```

Inherited from exceptions.Exception

```
__new__()
```

Inherited from exceptions.BaseException

```

__delattr__(), __getattr__(), __getitem__(), __getslice__(), __reduce__(), __repr__(),
__setattr__(), __setstate__(), __str__(), __unicode__()

```

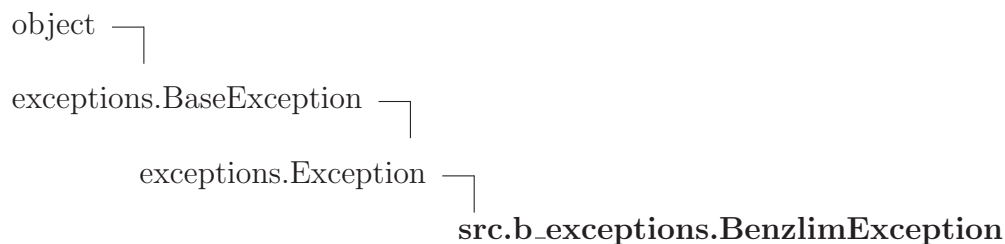
Inherited from object

```
__format__(), __hash__(), __reduce_ex__(), __sizeof__(), __subclasshook__()
```

3.3.2 Properties

Name	Description
<i>Inherited from exceptions.BaseException</i> args, message	
<i>Inherited from object</i> <code>--class--</code>	

3.4 Class *BenzlimException*



Known Subclasses: *src.b_exceptions.BadFormatException*, *src.b_exceptions.BadValueException*, *src.b_exceptions.PriceNotFoundException*, *src.b_exceptions.StationNotFoundException*, *src.b_exceptions.T*

3.4.1 Methods

`--init--`(*self*, **args*, ***kwargs*)

x.`--init--`(...) initializes *x*; see `help(type(x))` for signature

Overrides: *object*.`--init--` `extit`(inherited documentation)

Inherited from exceptions.Exception

`--new--`()

Inherited from exceptions.BaseException

`--delattr--`(), `--getattr--`(), `--getitem--`(), `--getslice--`(), `--reduce--`(), `--repr--`(),
`--setattr--`(), `--setstate--`(), `--str--`(), `--unicode--`()

Inherited from object

`--format--`(), `--hash--`(), `--reduce_ex--`(), `--sizeof--`(), `--subclasshook--`()

3.4.2 Properties

Name	Description
<i>Inherited from exceptions.BaseException</i>	
<i>args</i> , <i>message</i>	
<i>Inherited from object</i>	
<code>--class--</code>	

3.5 Class `PriceNotFoundException`



3.5.1 Methods

`--init--(self, *args, **kwargs)`
`x.__init__(...)` initializes `x`; see `help(type(x))` for signature
 Overrides: `object.__init__` `__init__`(inherited documentation)

Inherited from `exceptions.Exception`

`--new--()`

Inherited from `exceptions.BaseException`

`--delattr--()`, `--getattr__()`, `--getitem--()`, `--getslice--()`, `--reduce--()`, `--repr--()`,
`--setattr--()`, `--setstate--()`, `--str--()`, `--unicode--()`

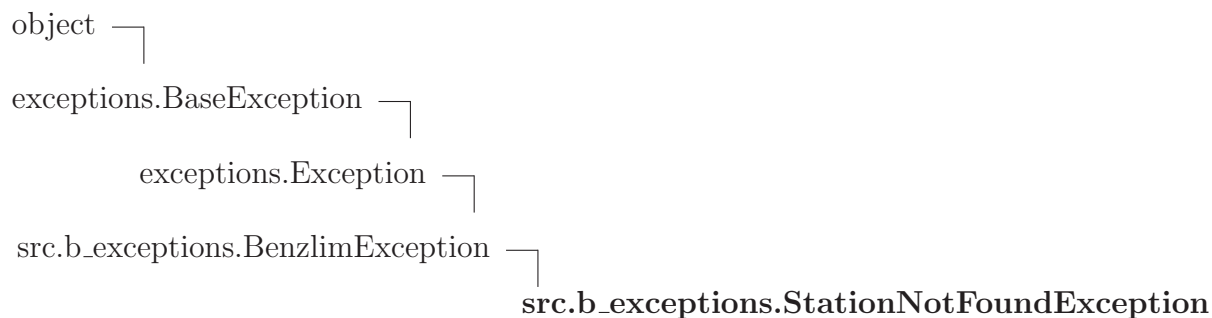
Inherited from `object`

`--format--()`, `--hash--()`, `--reduce_ex--()`, `--sizeof--()`, `--subclasshook--()`

3.5.2 Properties

Name	Description
<i>Inherited from <code>exceptions.BaseException</code></i>	
<code>args</code> , <code>message</code>	
<i>Inherited from <code>object</code></i>	
<code>--class--</code>	

3.6 Class `StationNotFoundException`



3.6.1 Methods

`--init--`(*self*, *args, **kwargs)
x.**`--init--`**(...) initializes *x*; see `help(type(x))` for signature
 Overrides: `object.__init__` `extit`(inherited documentation)

Inherited from `exceptions.Exception`

`--new--`()

Inherited from `exceptions.BaseException`

`--delattr--`(), `--getattr__`(), `--getitem--`(), `--getslice--`(), `--reduce--`(), `--repr--`(),
`--setattr--`(), `--setstate--`(), `--str--`(), `--unicode--`()

Inherited from `object`

`--format--`(), `--hash--`(), `--reduce_ex--`(), `--sizeof--`(), `--subclasshook--`()

3.6.2 Properties

Name	Description
<i>Inherited from <code>exceptions.BaseException</code></i>	
	args, message
<i>Inherited from <code>object</code></i>	
<code>--class--</code>	

3.7 Class *TrainingDataMissingException*



3.7.1 Methods

`--init--`(*self*, **args*, ***kwargs*)

x.`--init--`(...) initializes *x*; see `help(type(x))` for signature

Overrides: `object.--init--` `extit`(inherited documentation)

Inherited from exceptions.Exception

`--new--`()

Inherited from exceptions.BaseException

`--delattr--`(), `--getattr__`(), `--getitem--`(), `--getslice--`(), `--reduce--`(), `--repr--`(),
`--setattr--`(), `--setstate--`(), `--str--`(), `--unicode--`()

Inherited from object

`--format--`(), `--hash--`(), `--reduce_ex--`(), `--sizeof--`(), `--subclasshook--`()

3.7.2 Properties

Name	Description
<i>Inherited from exceptions.BaseException</i>	
args, message	
<i>Inherited from object</i>	
<code>--class--</code>	

4 Module `src.benchmark`

`benchmark.py` - Benchmarking tool

4.1 Functions

```
evaluate_prediction(station_id, ts, ground_ts, end_train_timestamp,
dir_prices, nb_predictions)
```

```
process_benchmark_prediction(args)
```

```
benchmark_with_prices(nb_stations, nb_predictions, dir_prices)
```

```
benchmark_without_prices(nb_stations, nb_predictions, dir_prices)
```

```
benchmark_predictions(nb_stations, nb_predictions, dir_prices)
```

```
benchmark_routing(nb_stations, nb_predictions, dir_prices)
```

```
process_benchmark(dir_prices, nb_stations=1, nb_predictions=5)
```

4.2 Variables

Name	Description
<code>__package__</code>	Value: <code>'src'</code>

5 Package `src.compat`

`compat` - Compatibility packages for `ython2` and `python3`

5.1 Modules

- **py2**: `py2` - Python2 compatibility module
(Section 6, p. 13)
- **py3**: `py3` - Python3 compatibility module
(Section 7, p. 14)

5.2 Variables

Name	Description
<code>__package__</code>	Value: <code>'src.compat'</code>

6 Module *src.compat.py2*

py2 - Python2 compatibility module

6.1 Functions

printf (*args, **kwargs)

str2unicode (value)

convert a str to unicode

6.2 Variables

Name	Description
<code>--package--</code>	Value: <code>'src.compat'</code>

7 Module `src.compat.py3`

py3 - Python3 compatibility module

7.1 Functions

printf (*args, **kwargs)

str2unicode (value)

convert a str to unicode

8 Module `src.config`

`config.py` - access `benzlim`'s instance configuration

8.1 Variables

Name	Description
<code>--package--</code>	Value: <code>'src'</code>

8.2 Class Configuration

Contains the configuration to run a `benzlim` instance

8.2.1 Methods

```
--init--(self, **kwargs)
```

```
get_instance(**kwargs)
```

```
config(**kwargs)
```

```
get_pool(self)
```

8.2.2 Class Variables

Name	Description
<code>RESOURCE_DIR</code>	Value: <code>'resources'</code>
<code>OUTPUT_DIR</code>	Value: <code>'out'</code>
<code>CLASSIFIER_FILENAM- E</code>	Value: <code>'classifier.pkl'</code>
<code>DATABASE_FILENAME</code>	Value: <code>'db.sqlite3'</code>
<code>TIME_BINS</code>	Value: <code>['00:00', '01:00', '02:00', '03:00', '04:00', '05:00', '0...</code>
<code>h</code>	Value: <code>23</code>

9 Package *src.coverage*

coverage - Coverage informations generation about benzlim

9.1 Functions

```
clean_benzlim()
```

```
clean_mp_coverages(empty_only=True)
```

```
coverage()
```

```
execute_coverage()
```

9.2 Variables

Name	Description
<code>--package--</code>	Value: <code>'src.coverage'</code>

10 Package *src.dao*

dao - Data Access Object packages for IO tasks

10.1 Modules

- **csv_:** *csv_.py* - read/write/investigate csv related files
(Section 11, p. 23)
- **db:** *db.py* - access station related informations
(Section 12, p. 25)

10.2 Class *CSVDAO*

object └─ **src.dao.csv_.CSVDAO**

10.2.1 Methods

get_station_filename (<i>cls, station_id, prices_dir=None</i>)

return the filename containing prices for the station <station_id>
--

is_prices_available (<i>cls, station_id</i>)

return True if prices are available for the given station else False
--

get_station_dataframe (<i>cls, station_id, dir_prices</i>)

return a DataFrame containing timestamps and prices of the station <station_id>
--

get_all_extended_stations_infos (<i>cls</i>)

return station informations: id: int => Station id name: str => Station name mark: str => Mark name street: str => streetname street-number: int => house number/ street number zipcode: int => zipcode town: str latitude: float longitude: float prices-available: bool => if prices are available begin_timestamp: str => the first price timestamp
--

get_all_stations_infos(*cls*)

return station informations: id: int => Station id name: str => Station name
 mark: str => Mark name street: str => streetname street-number: int =>
 house number/ street number zipcode: int => zipcode town: str latitude:
 float longitude: float

get_predict_params(*cls, filename*)

return [<end_timestamp>, <prediction_timestamp>, <station_id>]

get_route_params(*cls, filename*)

return <capacity>, [<timestamp>, <station_id>]

get_route_prices_params(*cls, filename*)

return [<timestamp>, <station_id>, <pred_price>]

get_route_as_predict_params(*cls, filename*)**get_predicted_prices(*cls, filename*)**

return [<end_timestamp>, <prediction_timestamp>, <station_id>,
 <pred_price>]

export_to_csv(*cls, filename, rows, header=None*)

rows in the file <filename> as csv

Inherited from object

`__delattr__()`, `__format__()`, `__getattr__()`, `__hash__()`, `__init__()`, `__new__()`, `__reduce__()`,
`__reduce_ex__()`, `__repr__()`, `__setattr__()`, `__sizeof__()`, `__str__()`, `__subclasshook__()`

10.2.2 Properties

Name	Description
<i>Inherited from object</i>	
<code>__class__</code>	

10.3 Class StationDAO

object —
src.dao.db.StationDAO

10.3.1 Methods

__init__ (<i>self</i>)
x.__init__(...) initializes x; see help(type(x)) for signature Overrides: object.__init__ extit(inherited documentation)
get_all (<i>cls</i>)
get (<i>cls</i> , <i>pk</i>)
get_all_before (<i>cls</i> , <i>timestamp</i>)
Return all stations with prices available before <timestamp>
get_latitude_longitude (<i>cls</i> , <i>pk</i>)
Return the latitude and longitude of the station with id <pk>
is_prices_available (<i>cls</i> , <i>pk</i>)
Return True if the station <pk> has prices else False
get_all_with_prices (<i>cls</i>)
Return all stations with prices available
get_all_without_prices (<i>cls</i>)
Return all stations without prices available
populate (<i>cls</i> , <i>data</i>)
Populate the corresponding table with items in data.

Inherited from object

__delattr__(), __format__(), __getattr__(), __hash__(), __new__(), __reduce__(), __reduce_ex__(),
__repr__(), __setattr__(), __sizeof__(), __str__(), __subclasshook__()

10.3.2 Properties

Name	Description
<i>Inherited from object</i> __class__	

10.3.3 Class Variables

Name	Description
table	Value: 'stations'
schema	Value: '\nCREATE TABLE IF NOT EXISTS stations(\n id INTEGER P...
indexes	Value: ('CREATE INDEX IF NOT EXISTS prices_index on stations(pri...
select_all_before_sql	Value: 'select * from stations where datetime(begin_timestamp \...
select_all_query_sql	Value: 'select * from stations'
select_query_sql	Value: 'select * from stations where id=?'
select_all_prices_available_sql	Value: 'select * from stations where prices_available'
select_all_prices_missing_sql	Value: 'select * from stations where not (prices_available)'
select_prices_is_available_sql	Value: 'select prices_available from stations where id=?'
select_latitude_longitude	Value: 'select latitude, longitude from stations where id=?'
insert_station_sql	Value: 'insert into stations (id, name, mark, street, street_num...

10.4 Class DBManager

```

object └─
        src.dao.db.DBManager

```

10.4.1 Methods

open (cls)

get_conn (<i>cls</i>)

close (<i>cls</i>)

init_db (<i>cls</i>)

Init the database if it doesn't exist yet.
--

force_init_db (<i>cls</i>)

Force the initialisation of the database and overwrite it.
--

execute (<i>cls</i> , <i>sql</i> , <i>data</i> =None)

executemany (<i>cls</i> , <i>sql</i> , <i>data</i> =None)

populate_db (<i>cls</i> , <i>data</i> , <i>sql_query</i> =None)

Populate the database with items in data.

set_auto_commit (<i>cls</i> , <i>value</i> =True)

enable/disable auto_commits to speed-up batch queries

Inherited from object

__delattr__(), __format__(), __getattr__(), __hash__(), __init__(), __new__(), __reduce__(),
__reduce_ex__(), __repr__(), __setattr__(), __sizeof__(), __str__(), __subclasshook__()

10.4.2 Properties

Name	Description
<i>Inherited from object</i>	
__class__	

10.4.3 Class Variables

Name	Description
sql_schemas	Value: ('\\nCREATE TABLE IF NOT EXISTS stations(\\n id INTEGER ...
sql_indexes	Value: ('CREATE INDEX IF NOT EXISTS prices_index on stations(pri...
conn	Value: None

continued on next page

Name	Description
table	Value: 'stations'
table_stations	Value: 'stations'
sql_insert_station_sql	Value: 'insert into stations (id, name, mark, street, street_num...
sql_get	Value: ''
sql_update	Value: ''
sql_delete	Value: ''
sql_save	Value: ''

11 Module src.dao.csv_

csv_.py - read/write/investigate csv related files

11.1 Variables

Name	Description
--package--	Value: 'src.dao'

11.2 Class CSVDAO

object └
src.dao.csv_.CSVDAO

11.2.1 Methods

get_station_filename(*cls, station_id, prices_dir=None*)

return the filename containing prices for the station <station_id>

is_prices_available(*cls, station_id*)

return True if prices are available for the given station else False

get_station_dataframe(*cls, station_id, dir_prices*)

return a DataFrame containing timestamps and prices of the station
<station_id>

get_all_extended_stations_infos(*cls*)

return station informations: id: int => Station id name: str => Station name
mark: str => Mark name street: str => streetname street-number: int =>
house number/ street number zipcode: int => zipcode town: str latitude:
float longitude: float prices_available: bool => if prices are available
begin_timestamp: str => the first price timestamp

get_all_stations_infos(*cls*)

return station informations: id: int => Station id name: str => Station name
 mark: str => Mark name street: str => streetname street-number: int =>
 house number/ street number zipcode: int => zipcode town: str latitude:
 float longitude: float

get_predict_params(*cls, filename*)

return [<end_timestamp>, <prediction_timestamp>, <station_id>]

get_route_params(*cls, filename*)

return <capacity>, [<timestamp>, <station_id>]

get_route_prices_params(*cls, filename*)

return [<timestamp>, <station_id>, <pred_price>]

get_route_as_predict_params(*cls, filename*)**get_predicted_prices(*cls, filename*)**

return [<end_timestamp>, <prediction_timestamp>, <station_id>,
 <pred_price>]

export_to_csv(*cls, filename, rows, header=None*)

rows in the file <filename> as csv

Inherited from object

`__delattr__()`, `__format__()`, `__getattr__()`, `__hash__()`, `__init__()`, `__new__()`, `__reduce__()`,
`__reduce_ex__()`, `__repr__()`, `__setattr__()`, `__sizeof__()`, `__str__()`, `__subclasshook__()`

11.2.2 Properties

Name	Description
<i>Inherited from object</i>	
<code>__class__</code>	

12 Module *src.dao.db*

db.py - access station related informations

12.1 Class *DBManager*



12.1.1 Methods

open (<i>cls</i>)

get_conn (<i>cls</i>)

close (<i>cls</i>)

init_db (<i>cls</i>)

Init the database if it doesn't exist yet.
--

force_init_db (<i>cls</i>)

Force the initialisation of the database and overwrite it.
--

execute (<i>cls</i> , <i>sql</i> , <i>data=None</i>)

executemany (<i>cls</i> , <i>sql</i> , <i>data=None</i>)

populate_db (<i>cls</i> , <i>data</i> , <i>sql_query=None</i>)

Populate the database with items in data.

set_auto_commit (<i>cls</i> , <i>value=True</i>)

enable/disable auto-commits to speed-up batch queries

Inherited from object

`--delattr--()`, `--format--()`, `--getattribute--()`, `--hash--()`, `--init--()`, `--new--()`, `--reduce--()`,
`--reduce_ex--()`, `--repr--()`, `--setattr--()`, `--sizeof--()`, `--str--()`, `--subclasshook--()`

12.1.2 Properties

Name	Description
<i>Inherited from object</i>	
<code>__class__</code>	

12.1.3 Class Variables

Name	Description
<code>sql_schemas</code>	Value: ('\\nCREATE TABLE IF NOT EXISTS stations\\n id INTEGER ...
<code>sql_indexes</code>	Value: ('CREATE INDEX IF NOT EXISTS prices_index on stations(pri...
<code>conn</code>	Value: None
<code>table</code>	Value: 'stations'
<code>table_stations</code>	Value: 'stations'
<code>sql_insert_station_sql</code>	Value: 'insert into stations (id, name, mark, street, street_num...
<code>sql_get</code>	Value: ''
<code>sql_update</code>	Value: ''
<code>sql_delete</code>	Value: ''
<code>sql_save</code>	Value: ''

12.2 Class StationDAO

object —
src.dao.db.StationDAO

12.2.1 Methods

`__init__(self)`

`x.__init__(...)` initializes x; see `help(type(x))` for signature

Overrides: `object.__init__` `extit`(inherited documentation)

`get_all(cls)`

`get(cls, pk)`

get_all_before (<i>cls, timestamp</i>)
Return all stations with prices available before <timestamp>
get_latitude_longitude (<i>cls, pk</i>)
Return the latitude and longitude of the station with id <pk>
is_prices_available (<i>cls, pk</i>)
Return True if the station <pk> has prices else False
get_all_with_prices (<i>cls</i>)
Return all stations with prices available
get_all_without_prices (<i>cls</i>)
Return all stations without prices available
populate (<i>cls, data</i>)
Populate the corresponding table with items in data.

Inherited from object

`--delattr--()`, `--format--()`, `--getattr--()`, `--hash--()`, `--new--()`, `--reduce--()`, `--reduce_ex--()`,
`--repr--()`, `--setattr--()`, `--sizeof--()`, `--str--()`, `--subclasshook--()`

12.2.2 Properties

Name	Description
<i>Inherited from object</i>	
<code>--class--</code>	

12.2.3 Class Variables

Name	Description
table	Value: 'stations'
schema	Value: '\nCREATE TABLE IF NOT EXISTS stations(\n id INTEGER P...
indexes	Value: ('CREATE INDEX IF NOT EXISTS prices_index on stations(pri...
select_all_before_sql	Value: 'select * from stations where datetime(begin.timestamp \...

continued on next page

Name	Description
select_all_query_sql	Value: 'select * from stations'
select_query_sql	Value: 'select * from stations where id=?'
select_all_prices_available_sql	Value: 'select * from stations where prices_available'
select_all_prices_missing_sql	Value: 'select * from stations where not (prices_available)'
select_prices_is_available_sql	Value: 'select prices_available from stations where id=?'
select_latitude_longitude	Value: 'select latitude, longitude from stations where id=?'
insert_station_sql	Value: 'insert into stations (id, name, mark, street, street_num...

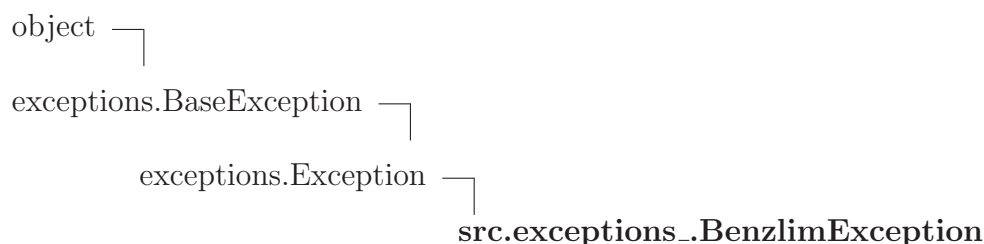
13 Module *src.exceptions_*

exceptions_.py - benzlim exceptions

13.1 Variables

Name	Description
<code>--package--</code>	Value: None

13.2 Class *BenzlimException*



Known Subclasses: *src.exceptions_.BadFormatException*, *src.exceptions_.BadValueException*, *src.exceptions_.PriceNotFoundException*, *src.exceptions_.StationNotFoundException*, *src.exceptions_.TrainNotFoundException*

13.2.1 Methods

```

__init__(self, message, *args)

x.__init__(...) initializes x; see help(type(x)) for signature
Overrides: object.__init__ extit(inherited documentation)

```

```

__str__(self)

str(x)
Overrides: object.__str__ extit(inherited documentation)

```

Inherited from exceptions.Exception

```
__new__()
```

Inherited from exceptions.BaseException

```

__delattr__(), __getattr__(), __getitem__(), __getslice__(), __reduce__(), __repr__(),
__setattr__(), __setstate__(), __unicode__()

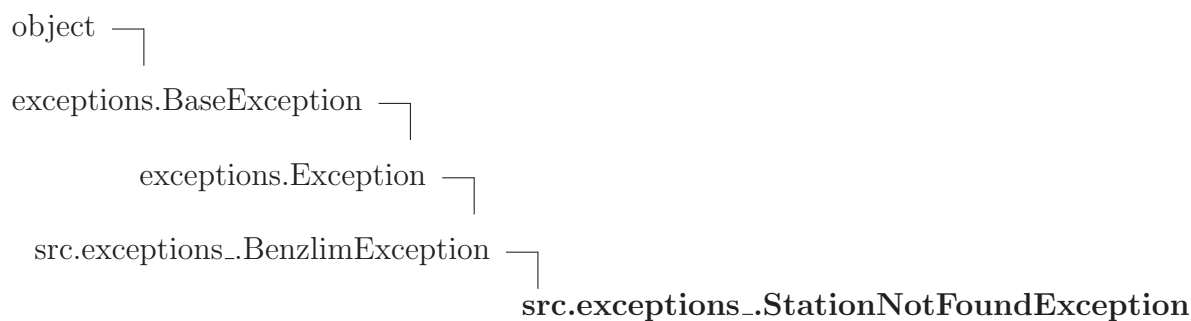
```


Inherited from object

`__format__()`, `__hash__()`, `__reduce_ex__()`, `__sizeof__()`, `__subclasshook__()`

13.2.2 Properties

Name	Description
<i>Inherited from exceptions.BaseException</i>	
<code>args</code> , <code>message</code>	
<i>Inherited from object</i>	
<code>__class__</code>	

13.3 Class StationNotFoundException**13.3.1 Methods**

<p><code>__init__(self, message, *args)</code></p> <p><code>x.__init__(...)</code> initializes <code>x</code>; see <code>help(type(x))</code> for signature</p> <p>Overrides: <code>object.__init__</code> <code>extit</code>(inherited documentation)</p>
--

Inherited from src.exceptions_.BenzlimException(Section 13.2)

`__str__()`

Inherited from exceptions.Exception

`__new__()`

Inherited from exceptions.BaseException

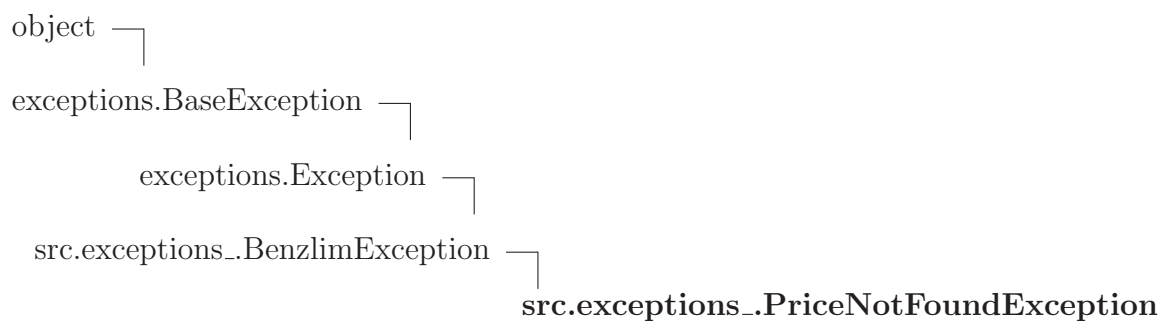
`__delattr__()`, `__getattr__()`, `__getitem__()`, `__getslice__()`, `__reduce__()`, `__repr__()`,
`__setattr__()`, `__setstate__()`, `__unicode__()`

Inherited from object

`__format__()`, `__hash__()`, `__reduce_ex__()`, `__sizeof__()`, `__subclasshook__()`

13.3.2 Properties

Name	Description
<i>Inherited from exceptions.BaseException</i>	
<code>args</code> , <code>message</code>	
<i>Inherited from object</i>	
<code>__class__</code>	

13.4 Class PriceNotFoundException**13.4.1 Methods**

<p><code>__init__(self, message, *args)</code></p> <p><code>x.__init__(...)</code> initializes <code>x</code>; see <code>help(type(x))</code> for signature</p> <p>Overrides: <code>object.__init__</code> extit(inherited documentation)</p>

Inherited from src.exceptions_.BenzlimException(Section 13.2)

`__str__()`

Inherited from exceptions.Exception

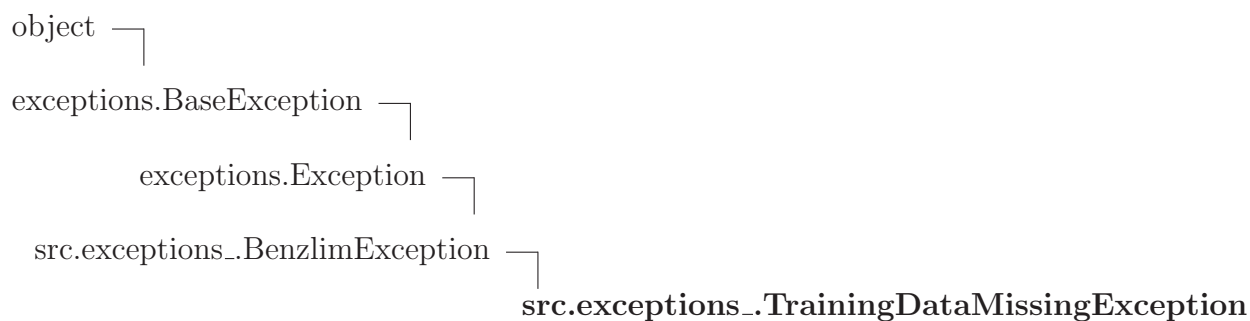
`__new__()`

Inherited from exceptions.BaseException

`__delattr__()`, `__getattr__()`, `__getitem__()`, `__getslice__()`, `__reduce__()`, `__repr__()`,
`__setattr__()`, `__setstate__()`, `__unicode__()`

Inherited from object`__format__()`, `__hash__()`, `__reduce_ex__()`, `__sizeof__()`, `__subclasshook__()`**13.4.2 Properties**

Name	Description
<i>Inherited from <code>exceptions.BaseException</code></i>	
<code>args</code> , <code>message</code>	
<i>Inherited from object</i>	
<code>__class__</code>	

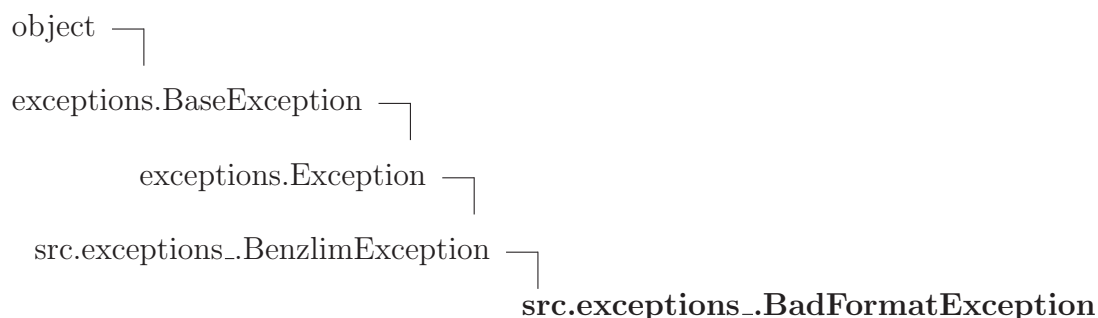
13.5 Class `TrainingDataMissingException`**13.5.1 Methods**

<code>__init__(self, message, *args)</code> <code>x.__init__(...)</code> initializes <code>x</code> ; see <code>help(type(x))</code> for signature Overrides: <code>object.__init__</code> extit(inherited documentation)

Inherited from `src.exceptions_.BenzlimException`(Section 13.2)`__str__()`***Inherited from `exceptions.Exception`***`__new__()`***Inherited from `exceptions.BaseException`***`__delattr__()`, `__getattr__()`, `__getitem__()`, `__getslice__()`, `__reduce__()`, `__repr__()`, `__setattr__()`, `__setstate__()`, `__unicode__()`

Inherited from object`__format__()`, `__hash__()`, `__reduce_ex__()`, `__sizeof__()`, `__subclasshook__()`**13.5.2 Properties**

Name	Description
<i>Inherited from <code>exceptions.BaseException</code></i>	
<code>args</code> , <code>message</code>	
<i>Inherited from object</i>	
<code>__class__</code>	

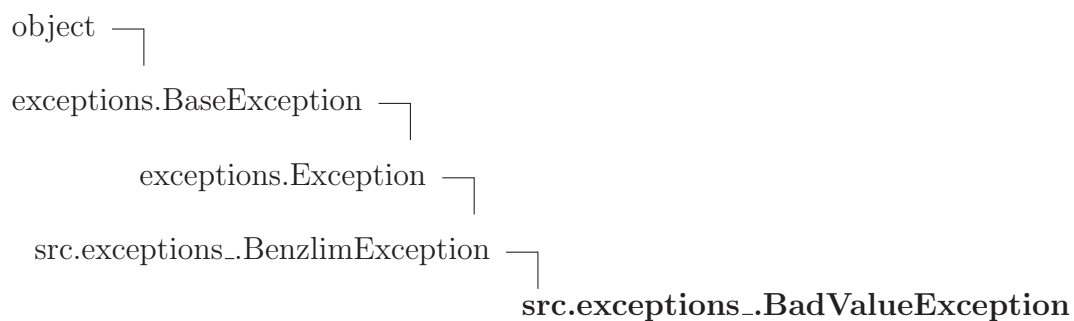
13.6 Class `BadFormatException`**13.6.1 Methods**

<code>__init__(self, message, *args)</code> <code>x.__init__(...)</code> initializes <code>x</code> ; see <code>help(type(x))</code> for signature Overrides: <code>object.__init__</code> <code>extit</code> (inherited documentation)

Inherited from `src.exceptions_.BenzlimException`(Section 13.2)`__str__()`***Inherited from `exceptions.Exception`***`__new__()`***Inherited from `exceptions.BaseException`***`__delattr__()`, `__getattr__()`, `__getitem__()`, `__getslice__()`, `__reduce__()`, `__repr__()`, `__setattr__()`, `__setstate__()`, `__unicode__()`

Inherited from object`__format__()`, `__hash__()`, `__reduce_ex__()`, `__sizeof__()`, `__subclasshook__()`**13.6.2 Properties**

Name	Description
<i>Inherited from <code>exceptions.BaseException</code></i>	<code>args</code> , <code>message</code>
<i>Inherited from object</i>	<code>__class__</code>

13.7 Class `BadValueException`**13.7.1 Methods**

<code>__init__(self, message, *args)</code> <code>x.__init__(...)</code> initializes <code>x</code> ; see <code>help(type(x))</code> for signature Overrides: <code>object.__init__</code> <code>extit</code> (inherited documentation)

Inherited from `src.exceptions_.BenzlimException`(Section 13.2)`__str__()`***Inherited from `exceptions.Exception`***`__new__()`***Inherited from `exceptions.BaseException`***`__delattr__()`, `__getattr__()`, `__getitem__()`, `__getslice__()`, `__reduce__()`, `__repr__()`, `__setattr__()`, `__setstate__()`, `__unicode__()`

Inherited from object`__format__()`, `__hash__()`, `__reduce_ex__()`, `__sizeof__()`, `__subclasshook__()`**13.7.2 Properties**

Name	Description
	<i>Inherited from <code>exceptions.BaseException</code></i>
	args, message
	<i>Inherited from object</i>
<code>__class__</code>	

14 Package `src.prediction`

`prediction` - The benzlim core prediction

14.1 Modules

- **classification:** `classification.py` - gas stations classification tools
(Section 15, p. 37)
- **db** (Section 16, p. 40)
- **predict:** `predict.py` - core prediction tools
(Section 17, p. 44)

14.2 Functions

```
process_task(args)
```

```
predict_prices_timestamps_x2_stations(timestamps_x2_stations, dir_prices,  
nb_workers=None)
```

```
return [<end_timestamp>, <timestamp>, <station_id>, <pred_price>],  
timestamps_x2_stations: list[<end_timestamp>, <timestamp>, <station_id>]  
dir_prices: directory path
```

```
process_predictions(filename, dir_prices, out_filename=None,  
nb_workers=None)
```

```
process_routing(filename, dir_prices, out_filename=None,  
gas_prices_file=None, nb_workers=None, auto_end_timestamp=True)
```

14.3 Variables

Name	Description
<code>--package--</code>	Value: <code>'src.prediction'</code>

15 Module src.prediction.classification

classification.py - gas stations classification tools

15.1 Variables

Name	Description
LATITUDE_MAX	Value: 90.0
LONGITUDE_MAX	Value: 180.0
HASH_MAX	Value: 982451653
NB_CHARS	Value: 45
__package__	Value: 'src.prediction'

15.2 Class CScClassifier

object └─
src.prediction.classification.CScClassifier

15.2.1 Methods

```
__init__(self, scoring_function=None, partition_index=0)
```

x.__init__(...) initializes x; see help(type(x)) for signature

Overrides: object.__init__ extit(inherited documentation)

```
fit(self, x_values, labels)
```

```
predict(self, x, try_skip_id=None)
```

predict class for features x based on the training data

Inherited from object

```
__delattr__(), __format__(), __getattr__(), __hash__(), __new__(), __reduce__(), __reduce_ex__(),  
__repr__(), __setattr__(), __sizeof__(), __str__(), __subclasshook__()
```

15.2.2 Properties

Name	Description
<i>Inherited from object</i>	
<code>--class--</code>	

15.3 Class Classifier



15.3.1 Methods

<code>--init--(self)</code>
<code>x.__init__(...)</code> initializes <code>x</code> ; see <code>help(type(x))</code> for signature
Overrides: <code>object.__init__</code> <code>exitit</code> (inherited documentation)

<code>get_category(cls, station_row)</code>
Return a category for the given station

<code>station_id2id(cls, station_id, end_train_timestamp=None, ignore_station=False)</code>
Return a usable station id

<code>station_row2id(cls, station_row, end_train_timestamp=None, ignore_station=False)</code>
Return a usable (with prices available) id

<code>get_station_features(cls, station_row)</code>
Return features for the given station

<code>get_prepared_data(cls, ext_stations=None)</code>
return features with corresponding classes

<code>train(cls, features=None, classes=None)</code>

<code>dump(cls, classifier, filename=None)</code>
--

<code>load(cls, filename=None, create_on_error=True)</code>

Inherited from object

`__delattr__()`, `__format__()`, `__getattr__()`, `__hash__()`, `__new__()`, `__reduce__()`, `__reduce_ex__()`,
`__repr__()`, `__setattr__()`, `__sizeof__()`, `__str__()`, `__subclasshook__()`

15.3.2 Properties

Name	Description
<i>Inherited from object</i>	
<code>__class__</code>	

16 Module src.prediction.db

16.1 Functions

icompare(*text1*, *text2*)

16.2 Variables

Name	Description
DB_PATH	Value: 'resources/db/db.sqlite3'
DB_SQL_INDEX_MARK	Value: 'CREATE INDEX IF NOT EXISTS mark_index on stations(mark C...
DB_SQL_INDEX_NAME	Value: 'CREATE INDEX IF NOT EXISTS word_index on stations(name C...
DB_SQL_INDEX_PLACE	Value: 'CREATE INDEX IF NOT EXISTS place_index on stations(place...
DB_SQL_INDEX_STATION	Value: 'CREATE INDEX IF NOT EXISTS station_index on prices(stati...
DB_SQL_SCHEMA_PRICES	Value: 'CREATE TABLE IF NOT EXISTS prices(\n id INTEGER PRIMA...
DB_SQL_SCHEMA_STATIONS	Value: '\nCREATE TABLE IF NOT EXISTS stations(\n id INTEGER P...
--package--	Value: 'src.prediction'

16.3 Class DBManager

object └─
 src.prediction.db.DBManager

16.3.1 Methods

close(*cls*)

execute(*cls*, *sql*, *data*=None)

executemany(*cls*, *sql*, *data*=None)

force_init_db (cls)

Force the initialisation of the database and overwrite it.
--

getConn (cls)

init_db (cls)

Init the database if it doesn't exist yet.
--

open (cls)

populate_db (cls, data, sql_query=None)
--

Populate the database with items in data.

set_auto_commit (cls, value=True)
--

Inherited from object

`__delattr__()`, `__format__()`, `__getattr__()`, `__hash__()`, `__init__()`, `__new__()`, `__reduce__()`,
`__reduce_ex__()`, `__repr__()`, `__setattr__()`, `__sizeof__()`, `__str__()`, `__subclasshook__()`

16.3.2 Properties

Name	Description
<i>Inherited from object</i>	
<code>__class__</code>	

16.3.3 Class Variables

Name	Description
<code>conn</code>	Value: None
<code>filename</code>	Value: 'resources/db/db.sqlite3'
<code>sql_delete</code>	Value: ''
<code>sql_get</code>	Value: ''
<code>sql_indexes</code>	Value: ('CREATE INDEX IF NOT EXISTS mark_index on stations(mark ...
<code>sql_insert_price</code>	Value: 'insert into Prices (id, station_id, timestamp, price) va...
<code>sql_insert_station</code>	Value: 'insert into Stations (id, name, mark, street, street_num...
<code>sql_save</code>	Value: ''

continued on next page

Name	Description
sql_schemas	Value: ('\\nCREATE TABLE IF NOT EXISTS stations(\\n id INTEGER ...
sql_update	Value: ''
table	Value: 'Stations'
table_prices	Value: 'Prices'
table_stations	Value: 'Stations'

16.4 Class PriceDAO

16.4.1 Methods

```
get(cls, pk=None)
```

```
getAll(cls)
```

16.4.2 Class Variables

Name	Description
select_all_query	Value: 'select * from prices'
select_query	Value: 'select * from prices where id=?'
table	Value: 'prices'

16.5 Class StationDAO

16.5.1 Methods

```
get(cls, pk)
```

```
getAll(cls)
```

```
get_all_with_prices(cls)
```

```
get_all_without_prices(cls)
```

```
is_prices_missing(cls, pk)
```

16.5.2 Class Variables

Name	Description
select_all_prices_available	Value: 'select * from stations where prices_available=1'
select_all_prices_missing	Value: 'select * from stations where prices_available=0'
select_all_query	Value: 'select * from stations'
select_prices_is_available	Value: 'select * from stations where id=?'
select_query	Value: 'select * from stations where id=?'
table	Value: 'stations'

17 Module `src.prediction.predict`

`predict.py` - core prediction tools

17.1 Functions

`get_time_range(timestamp)`

`get_freq_avg(ts, freq='10T', fill_method='pad', fill_method2=None)`

resample the timeserie with the new frequency `<freq>` using the fill methods for NaNs

`get_time(timestamp, field=None)`

return the corresponding value of the attribut corresponding to `<field>`
timestamp: `<pd.Timestamp>` field: `<str>` Y, M, W, D, H, T

`get_price_predictor(station_id, dir_prices, ts=None, time_begin=None, time_end=None, end_train_timestamp=None, poly_deg=2)`

Generate a price predictor for gas station `<station_id>` of the timeserie `<ts>`,
station_id: str, the id of the station ts: DataFrame|Serie, the price's timeserie
of as gas station time_begin: str, time_end: str, end_train_timestamp: str, the
last usable timestamp for learning, poly_deg: int, the degree of polynomial
approximation [1,2,3,4,5] return the callable `prediction(timestamp)`

if `station_id` is submitted, the predictor is cached resp. recovered from the
cache if available

If the difference between the predicted value and the average is bigger than
20% of the average, the predictor will return the average instead of the
predicted value

```
get_price_predictor2(station_id, dir_prices, ts=None, time_begin=None,
time_end=None, end_train_timestamp=None, poly_deg=2)
```

Generate a price predictor for gas station <station_id> of the timeserie <ts>, station_id: str, the id of the station ts: DataFrame|Series, the price's timeserie of as gas station end_train_timestamp: str, the last usable timestamp for learning, poly_deg: int, the degree of polynomial approximation return the predictor as a `numpy.poly1d`

if station_id is submitted, the predictor is cached resp. recovered from the cache

If the difference between the predicted value and the average is bigger than 20% of the average, the predictor will return the average instead of the predicted value

```
predict_price(station_id, timestamp, end_train_timestamp, dir_prices,
bench_ts=None)
```

17.2 Variables

Name	Description
MAX_MARGIN_COEF	Value: 0.2
CACHE_PREDICTORS	Value: {}
__package__	Value: 'src.prediction'

18 Package *src.routing*

routing - gas tank strategy manager

18.1 Modules

- **graph:** *graph.y* - Tank strategy optimizer for graph based routes
(Section 19, p. 47)
- **node:** *node.py* - Nodes for graph based representation of gas stations in a route
(Section 20, p. 48)

18.2 Functions

<code>generate_tank_infos(<i>capacity, timestamps_stations_prices</i>)</code>

18.3 Variables

Name	Description
<code>--package--</code>	Value: <code>'src.routing'</code>

19 Module `src.routing.graph`

`graph.y` - Tank strategy optimizer for graph based routes

19.1 Variables

Name	Description
<code>--package--</code>	Value: <code>'src.routing'</code>

19.2 Class Graph

19.2.1 Methods

<code>--init--(<i>self</i>, <i>capacity</i>)</code>
<code>gas_for_km(<i>self</i>, <i>km</i>)</code>
<code>km_for_gas(<i>self</i>, <i>gas</i>)</code>
<code>find_prevs(<i>self</i>)</code>
<code>find_nexts(<i>self</i>)</code>
<code>generate_refuel_infos(<i>self</i>)</code>

20 Module src.routing.node

node.py - Nodes for graph based representation of gas stations in a route

20.1 Variables

Name	Description
<code>--package--</code>	Value: 'src.routing'

20.2 Class Node

20.2.1 Methods

```
--init--(self, id_, lat, lon, price=0, timestamp='')
```

```
--lt--(self, other)
```

```
--le--(self, other)
```

```
--eq--(self, other)
```

```
--str--(self)
```

```
--repr--(self)
```

```
distance_to(self, other, g, use_tolerance=False)
```

```
price_for_gas(self, amount)
```

```
set_price(self, price)
```

20.2.2 Properties

Name	Description
key	

21 Package *src.tests*

tests - Tests runner

21.1 Functions

```
diff_prices(data1, data2)
```

```
get_route_files_prices()
```

```
get_predict_files_prices()
```

```
verify_route(route_filename, route_prices_filename, nb_runs=12)
```

```
test_predict()
```

```
test_route()
```

```
test()
```

21.2 Variables

Name	Description
--package--	Value: 'src.tests'

22 Module *src.train*

train.py - manage the whole training

22.1 Variables

Name	Description
<code>--package--</code>	Value: <code>'src'</code>

22.2 Class Trainer



22.2.1 Methods

<code>train(force_train=False)</code>

<code>autotrain()</code>

Inherited from object

`__delattr__()`, `__format__()`, `__getattr__()`, `__hash__()`, `__init__()`, `__new__()`, `__reduce__()`,
`__reduce_ex__()`, `__repr__()`, `__setattr__()`, `__sizeof__()`, `__str__()`, `__subclasshook__()`

22.2.2 Properties

Name	Description
<i>Inherited from object</i>	
<code>--class--</code>	

23 Module `src.utils`

`utils.py` - usefool tools

23.1 Functions

<code>diff_score</code> (<i>v1</i> , <i>v2</i>)
--

return the norm of the difference between both vectors
--

<code>str2latitude</code> (<i>value</i>)

convert a str to valid latitude

<code>str2longitude</code> (<i>value</i>)
--

convert a str to valid longitude

<code>str2mark</code> (<i>value</i>)

convert a str to unicode

<code>str2town</code> (<i>value</i>)

convert a str to unicode

<code>str2zipcode</code> (<i>value</i>)
--

convert a str to int

<code>create_file_dirs</code> (<i>filename</i>)
--

create all directories contained in the tree to filename
--

<code>create_dirs</code> (<i>path</i>)

create all directories leading to path (inclusive itself)

23.2 Variables

Name	Description
<code>ERROR_FILE_EXISTS</code>	Value: 17
<code>__package__</code>	Value: 'src'

Index

- src (*package*), 2–3
 - src.__main__ (*module*), 4
 - src.b_exceptions (*module*), 5–10
 - src.b_exceptions.BadFormatException (*class*), 5–6
 - src.b_exceptions.BadValueException (*class*), 6
 - src.b_exceptions.BenzlimException (*class*), 6–7
 - src.b_exceptions.PriceNotFoundException (*class*), 7–8
 - src.b_exceptions.StationNotFoundException (*class*), 8–9
 - src.b_exceptions.TrainingDataMissingException (*class*), 9–10
 - src.benchmark (*module*), 11
 - src.benchmark.benchmark_predictions (*function*), 11
 - src.benchmark.benchmark_routing (*function*), 11
 - src.benchmark.benchmark_with_prices (*function*), 11
 - src.benchmark.benchmark_without_prices (*function*), 11
 - src.benchmark.evaluate_prediction (*function*), 11
 - src.benchmark.process_benchmark (*function*), 11
 - src.benchmark.process_benchmark_prediction (*function*), 11
 - src.compat (*package*), 12
 - src.compat.py2 (*module*), 13
 - src.compat.py3 (*module*), 14
 - src.config (*module*), 15
 - src.config.Configuration (*class*), 15
 - src.coverage' (*package*), 16
 - src.coverage'.clean_benzlim (*function*), 16
 - src.coverage'.clean_mp_coverages (*function*), 16
 - src.coverage'.coverage (*function*), 16
 - src.coverage'.execute_coverage (*function*), 16
 - src.dao (*package*), 17–22
 - src.dao.csv_ (*module*), 23–24
 - src.dao.db (*module*), 25–28
 - src.exceptions_ (*module*), 29–35
 - src.exceptions_.BadFormatException (*class*), 33–34
 - src.exceptions_.BadValueException (*class*), 34–35
 - src.exceptions_.BenzlimException (*class*), 29–30
 - src.exceptions_.PriceNotFoundException (*class*), 31–32
 - src.exceptions_.StationNotFoundException (*class*), 30–31
 - src.exceptions_.TrainingDataMissingException (*class*), 32–33
 - src.main (*function*), 3
 - src.prediction (*package*), 36
 - src.prediction.classification (*module*), 37–39
 - src.prediction.db (*module*), 40–43
 - src.prediction.predict (*module*), 44–45
 - src.prediction.predict_prices_timestamps_x2_stations (*function*), 36
 - src.prediction.process_predictions (*function*), 36
 - src.prediction.process_routing (*function*), 36
 - src.prediction.process_task (*function*), 36
 - src.routing (*package*), 46
 - src.routing.generate_tank_infos (*function*), 46
 - src.routing.graph (*module*), 47
 - src.routing.node (*module*), 48
 - src.tests (*package*), 49
 - src.tests.diff_prices (*function*), 49
 - src.tests.get_predict_files_prices (*function*), 49
 - src.tests.get_route_files_prices (*function*), 49
 - src.tests.test (*function*), 49
 - src.tests.test_predict (*function*), 49

- src.tests.test_route (*function*), 49
- src.tests.verify_route (*function*), 49
- src.train (*module*), 50
 - src.train.Trainer (*class*), 50
- src.utils (*module*), 51
 - src.utils.create_dirs (*function*), 51
 - src.utils.create_file_dirs (*function*), 51
 - src.utils.diff_score (*function*), 51
 - src.utils.str2latitude (*function*), 51
 - src.utils.str2longitude (*function*), 51
 - src.utils.str2mark (*function*), 51
 - src.utils.str2town (*function*), 51
 - src.utils.str2zipcode (*function*), 51