

---

# **SRTrain**

***Release 0.0.0***

**ryanking13**

**Sep 07, 2023**



# CONTENTS

<b>1</b>	<b>Overview</b>	<b>1</b>
<b>2</b>	<b>Quickstart</b>	<b>3</b>
<b>3</b>	<b>Index</b>	<b>5</b>
	<b>Index</b>	<b>13</b>



## OVERVIEW

SRTrain is a python wrapper of SRT(Super Rapid Train: <https://etk.srail.kr/>).

With SRTrain, you **can**:

- Search SRT train schedules.
- Reserve SRT trains.
- Find your reservations/tickets information.
- Cancel reservations/tickets.

while you **can't**:

- Pay for a ticket.
- Search or reserve non-SRT trains (KTX, ITX, ... ) (use [korail2](#) instead)

This project was inspired from [korail2](#) of [carpedm20](#).



## QUICKSTART

```
pip install SRTtrain
```

```
>>> from SRT import SRT
>>> srt = SRT("010-1234-xxxx", YOUR_PASSWORD)

>>> dep = ''
>>> arr = ''
>>> date = '20190930'
>>> time = '144000'
>>> trains = srt.search_train(dep, arr, date, time)
>>> trains
# [[SRT] 09 30, ~(15:00~17:34)  ,  ,
# [SRT] 09 30, ~(15:30~18:06)  ,  ,
# [SRT] 09 30, ~(16:00~18:24)  ,  ,
# [SRT] 09 30, ~(16:25~18:45)  ,  , ...]

>>> reservation = srt.reserve(trains[1])
>>> reservation
# [SRT] 09 30, ~(15:30~18:06) 53700(1), 09 20 23:38
```





## 3.1 Installation

Python Support: SRTrain requires **Python version >= 3.6**

Install SRTrain with **pip**:

```
pip install SRTrain
```

## 3.2 SRT Reference

**class** `SRT.SRT(srt_id: str, srt_pw: str, auto_login: bool = True, verbose: bool = False)`

SRT

### Parameters

- `srt_id(str)` – SRT (,,)
- `srt_pw(str)` – SRT
- `auto_login(bool)` – [login\(\)](#)
- `verbose(bool)` –

```
>>> srt = SRT("1234567890", YOUR_PASSWORD) # with membership number
>>> srt = SRT("def6488@gmail.com", YOUR_PASSWORD) # with email
>>> srt = SRT("010-1234-xxxx", YOUR_PASSWORD) # with phone number
```

**login**(*srt\_id*: str | None = None, *srt\_pw*: str | None = None)

SRT .

, .

### Parameters

- `srt_id(str, optional)` – SRT
- `srt_pwd(str, optional)` – SRT

### Returns

### Return type

bool

**logout()** → bool

SRT .

**search\_train**(*dep: str, arr: str, date: str | None = None, time: str | None = None, time\_limit: str | None = None, available\_only: bool = True*) → list[SRT.train.SRTTrain]

SRT .

#### Parameters

- **dep** (str) –
- **arr** (str) –
- **date** (str, optional) – (yyyyMMdd) (default: )
- **time** (str, optional) – (hhmmss) (default: 0 0 0)
- **time\_limit** (str, optional) – (hhmmss)
- **available\_only** (bool, optional) – (default: True)

#### Returns

#### Return type

list[SRTTrain]

**reserve**(*train: SRTTrain, passengers: list[SRT.passenger.Passenger] | None = None, special\_seat: SeatType = SeatType.GENERAL\_FIRST, window\_seat: bool | None = None*) → SRTReservation

.

```
>>> trains = srt.search_train("", "", "210101", "000000")
>>> srt.reserve(trains[0])
```

#### Parameters

- **train** (SRTTrain) –
- **passengers** (list[Passenger], optional) – (default: 1)
- **special\_seat** (SeatType) – / (default: )
- **window\_seat** (bool, optional) –

#### Returns

#### Return type

SRTReservation

**reserve\_standby**(*train: SRTTrain, passengers: list[SRT.passenger.Passenger] | None = None, special\_seat: SeatType = SeatType.GENERAL\_FIRST, mblPhone: str | None = None*) → SRTReservation

.

```
>>> trains = srt.search_train("", "", "210101", "000000")
>>> srt.reserve_standby(trains[0])
```

#### Parameters

- **train** (SRTTrain) –

- **passengers** (list[Passenger], optional) – (default: 1)
- **special\_seat** (SeatType) – / (default: )
- **mblPhone** (str, optional) –

**Returns**

**Return type**

SRTReservation

**reserve\_standby\_option\_settings**(reservation: SRTReservation | int, isAgreeSMS: bool, isAgreeClassChange: bool, telNo: str | None = None) → bool

```
>>> trains = srt.search_train("", "", "210101", "000000")
>>> srt.reserve_standby(trains[0])
>>> srt.reserve_standby_option_settings("1234567890", True, True, "010-1234-xxxx
↪")
```

**Parameters**

- **reservation** (SRTReservation or int) –
- **isAgreeSMS** (bool) – SMS
- **isAgreeClassChange** (bool) –
- **telNo** (str, optional) –

**Returns**

**Return type**

bool

**get\_reservations**(paid\_only: bool = False) → list[SRT.reservation.SRTReservation]

**Parameters**

**paid\_only** (bool) –

**Returns**

**Return type**

list[SRTReservation]

**ticket\_info**(reservation: SRTReservation | int) → list[SRT.reservation.SRTTicket]

```
>>> reservations = srt.get_reservations()
>>> reservations
# [[SRT] 09 30, ~(15:30~18:06) 130700(3), 09 19 19:11]
>>> reservations[0].tickets
# [18 9C () / [52300(600 )],
# 18 10C () / [52300(600 )],
# 18 10D () 4~6 [26100(26800 )]]
```

**Parameters**

**reservation** (SRTReservation or int) –

**Returns**

list[SRTTicket]

**cancel**(*reservation: SRTReservation | int*) → bool

```
>>> reservation = srt.reserve(train)
>>> srt.cancel(reservation)
>>> reservations = srt.get_reservations()
>>> srt.cancel(reservations[0])
```

**Parameters**

**reservation** (SRTReservation or int) –

**Returns**

**Return type**

bool

## 3.3 Advanced Usages

SRT, .

```
>>> from SRT import SRT, SeatType
>>> srt = SRT('your-id', 'your-password')
>>> trains = srt.search_train(' ', ' ', '20221122', '000000')
```

### 3.3.1

**WARNING:**

Highly inspired by [@dotaitch](#).

) 2, 1

```
>>> from SRT.passenger import Adult, Child
>>> srt.reserve(trains[0], passengers=[Adult(), Adult(), Child()])
```

- Adult: /
- Child:
- Senior:
- Disability1To3: 1~3
- Disability4To6: 4~6

### 3.3.2 /

)

```
>>> from SRT import SeatType
>>> srt.reserve(self, trains[0], special_seat=SeatType.GENERAL_FIRST)
```

- `SeatType.GENERAL_FIRST` :
- `SeatType.GENERAL_ONLY` :
- `SeatType.SPECIAL_FIRST` :
- `SeatType.SPECIAL_ONLY` :

## 3.4 Contributing

You can contribute to SRTrain by:

- **Reporting bugs:** please report when bug occurred through [github issue](#).
- **Fixing bugs:** when you fixes a bug, please run tests before submitting a PR.
- **Enhancing documentations:** SRTrain uses Sphinx + Markdown for documentation.
- **Adding/Proposing new features**

### 3.4.1 How to run tests

```
export SRT_USERNAME=<YOUR_SRT_USERNAME>
export SRT_PASSWORD=<YOUR_SRT_PASSWORD>

# For Windows
# set SRT_USERNAME=<YOUR_SRT_USERNAME>
# set SRT_PASSWORD=<YOUR_SRT_PASSWORD>

pip install -r requirements/test.txt
pip install -r requirements.txt

black SRT
pytest SRT -v -x

# For full test
# Warning:      /
pytest SRT -v -x --full
```

### 3.4.2 How to build the documentation

```
pip install requirements/docs.txt
cd docs/
make html
```

## 3.5 Changelog

### 3.5.1 Unreleased

#### 3.5.2 v2.1.0 (2023/08/15)

- SRT . (#243)

#### 3.5.3 v2.0.8 (2023/08/02)

- `search_train()` . (#238)
- Public API static typing (#234)

#### 3.5.4 v2.0.7 (2023/04/05)

- `get_reservations()` `payed_only` (#228)

#### 3.5.5 v2.0.6 (2023/02/23)

- (#221)

#### 3.5.6 v2.0.5 (2022/11/24)

- SRT (#201)

#### 3.5.7 v2.0.4 (2022/10/31)

- SRT (#193)

#### 3.5.8 v2.0.3 (2022/06/11)

- `search_train()` `time_limit` parameter (#130)
- (#166)

### 3.5.9 v2.0.2 (2021/06/28)

- 
- PEP517(pyproject.toml)
- PyPI

### 3.5.10 v2.0.1 (2021/04/07)

- SRT
- srt-reserve

### 3.5.11 v2.0.0 (2021/01/22)

- SRT API URL
- , ,

### 3.5.12 v1.0.2 (2019/12/24)

- /
- CI CircleCI Github Actions

### 3.5.13 v1.0.0 (2019/09/13)

- SRT NEO API (#1)

### 3.5.14 v0.1.5 (2019/02/15)

- 

### 3.5.15 v0.1.3 (2019/02/15)

- search\_train()





## INDEX

### C

`cancel()` (*SRT.SRT method*), 8

### G

`get_reservations()` (*SRT.SRT method*), 7

### L

`login()` (*SRT.SRT method*), 5

`logout()` (*SRT.SRT method*), 5

### R

`reserve()` (*SRT.SRT method*), 6

`reserve_standby()` (*SRT.SRT method*), 6

`reserve_standby_option_settings()` (*SRT.SRT method*), 7

### S

`search_train()` (*SRT.SRT method*), 6

`SRT` (*class in SRT*), 5

### T

`ticket_info()` (*SRT.SRT method*), 7