

# RailGo Webサービス解説書

ExpService04  
ExpService05  
ExpService06  
ExpService07  
ExpService08  
ExpService09  
ExpService10

Version 1.09

2023/2/9 Ver1.09

イースト(株)

# 目次

1. サイト .....	3
2. メソッド .....	4
SearchStation .....	4
GetDefaultNavigation .....	4
GetUserPoints .....	4
SearchCourse .....	5
SearchStationAll .....	5
SearchCourseAll .....	5
3. クラス .....	7
Area クラス .....	7
Course クラス .....	8
Station クラス .....	10
RouteSection クラス .....	12
FareSection クラス .....	13
SurchargeSection クラス .....	14
PassSection クラス .....	15
Line クラス .....	16
Fare クラス .....	17
Corporation クラス .....	19
Ticket クラス .....	20
Navigation クラス .....	21
4. エラーコード .....	24
5. サンプルコード .....	25
通常検索 .....	25
一括検索 .....	26
6. バージョンによる違い .....	27
ExpService05 での変更点 .....	27
ExpService06 での変更点 .....	27
ExpService07 での変更点 .....	28
ExpService08 での変更点 .....	28
ExpService09 での変更点 .....	28
ExpService10 での変更点 .....	28

RailGo Webサービスの解説です。

## 1. サイト

テストサイトの URL は以下になります。

呼び出し URL (ブラウザで見た場合は、解説が表示されます。)

<http://wstest.railgo.jp/ExpService10.asmx>

WSDL

<http://wstest.railgo.jp/ExpService10.asmx?wsdl>

## 2. メソッド

### SearchStation

駅名のヨミから、駅情報を検索します。

書式

```
Station[] Stations = SearchStation( String StationYomi, Area.AreaType AreaType,  
    Station.StationType StationType, int Date)
```

パラメータ

**String StationYomi** 駅名の読みを指定します。漢字・ひらがな・カタカナ・ローマ字が指定できます。指定された読みで始まる駅の情報が返されます。読みの先頭に「?」を付けると、その読みを含むすべての駅の情報が返されます。

**AreaType** 検索対象範囲を指定します。(通常AreaType.Japanを指定します。)

**Station.StationType StationType** 検索対象の駅種別を指定します。(通常は RailRoadを指定します。)

**Int Date** 検索する日付を指定します。(2007/01/17 であれば、20070117 と指定します。)

戻り値

**Station[]** 該当する駅の配列が戻ります。

### GetDefaultNavigation

デフォルトの検索オプション値を取得します。

書式

```
Navigation GetDefaultNavigation()
```

パラメータ

なし。

戻り値

**Navigation** デフォルトの検索オプションが戻ります。

### GetUserPoints

指定月の使用ポイントを取得します。

書式

```
int GetUserPoints(DateTime Date)
```

パラメータ

**DateTime Date** 日の指定は1日にしてください(例:2006/03/01)。

戻り値

使用ポイントが戻ります。使用ポイントが取得できない場合は-1 が戻ります。

## SearchCourse

駅配列から、経路情報を検索します。[0]を発駅、[UpperBound]を着駅とし、その間は経由駅になります。距離単位は 100mです。駅は 20 まで指定できます。

書式

```
Course[] SearchCourse( Station[] Stations, Navigation SearchOption )
```

パラメータ

**Station[] Stations** SearchStationで取得した駅情報を元に、必要なものを指定します。

**Navigation SearchOption** GetDefaultNavigationで取得した検索オプションを必要なら変更して指定します。

戻り値

**Course[]** 検索された経路の配列が戻ります。

## SearchStationAll

駅名のヨミから、すべての駅情報を一括で検索します。

このメソッドは、ExpService04, ExpService05 にはありません。

書式

```
Station[][] Stations = SearchStationAll( String[] StationYomiArray, Area.AreaType AreaType, Station.StationType StationType, int Date)
```

パラメータ

**String[] StationYomiArray** 駅名の読みを指定します。漢字・ひらがな・カタカナ・ローマ字が指定できます。指定された読みで始まる駅の情報が返されます。読みの先頭に「?」を付けると、その読みを含むすべての駅の情報が返されます。配列で複数指定できます。

**AreaType** 検索対象範囲を指定します。(通常AreaType.Japanを指定します。)

**Station.StationType StationType** 検索対象の駅種別を指定します。(通常は RailRoadを指定します。)

**Int Date** 検索する日付を指定します。(2007/01/17 であれば、20070117 と指定します。)

戻り値

**Station[][]** 該当する駅の配列の配列が戻ります。

## SearchCourseAll

発駅・着駅(配列により複数指定可)から、経路情報を一括で検索します。

このメソッドは、ExpService04, ExpService05 にはありません。

#### 書式

Course[] SearchCourse( Station[] DepartureStations, Station[] ArrivalStations, Navigation SearchOption )

#### パラメータ

**Station[]DepartureStations** 発駅です。SearchStationで取得した駅情報を元に、必要なものを指定します。

**Station[]ArrivalStations** 着駅です。SearchStationで取得した駅情報を元に、必要なものを指定します。

**Navigation SearchOption** GetDefaultNavigationで取得した検索オプションを必要なら変更して指定します。

#### 戻り値

**Course[]** 検索された経路の配列が戻ります。

### 3. クラス

#### Area クラス

検索対象範囲の地域の設定です。

AreaType

None = 0

hokkaido = 1

tohoku = 2

kanto = 4

chubu = 8

kinki = 16

chugoku = 32

shikoku = 64

kyusyu = 128

Japan = 255

# Course クラス

経路の情報です。

## a) 経路全体の情報

・駅情報文字列

String Description

・所要時間合計(単位:分)

int TotalTime

1

・乗車時間合計(単位:分)

int TotalBoardTime

2

・徒歩時間合計(単位:分)

int TotalWalkTime

3

・その他時間合計(単位:分)

int TotalOtherTime

乗り継ぎ時間、乗車待ち時間等の時間

4

・乗り換え回数

int TransferCount

5

・距離合計(単位:100m)

int TotalDistance

・定期券運賃合計

PassSection PassFare

1ヵ月定期券運賃合計

int PassFare.OneMonth

3ヵ月定期券運賃合計

int PassFare.ThreeMonth

6ヵ月定期券運賃合計

int PassFare.SixMonth

・運賃合計

Fare TotalFare

片道運賃合計

Decimal TotalFare.OneWayFare

往復運賃合計

Decimal TotalFare.RoundTripFare

・特急料金合計

Fare TotalSurcharge



片道特急料金合計  
Decimal TotalSurcharge.OneWayFare

往復特急料金合計  
Decimal TotalSurcharge.RoundTripFare

b) 区間ごとの情報

・停車駅  
Station [] Stations

・ルートセクション  
RouteSection [] RouteSections;

・料金セクション  
FareSection [] FareSections;

・特急料金セクション  
SurchargeSection [] SurchargeSections;

・定期区間  
PassSection [] PassSections;

## Station クラス

駅の情報です。

### ・駅種別定義 (v09 まで)

StationType

none = 0	未設定状態 (エラー値)
RailRoad = 1	鉄道駅
AirPort = 2	空港
LandMark = 4	ランドマーク (SearchStationでは指定できません。)
BusStop = 8	バス停

### ・駅種別定義 (v10 以降)

StationType

none = 0	未設定状態 (エラー値)
RailRoad = 1	鉄道駅 (高速バス停、連絡バス停、港を含む)
AirPort = 2	空港
LandMark = 4	ランドマーク (SearchStationでは指定できません。)
BusStop = 8	路線バス停
RailRoadOnly = 16	鉄道駅のみ
HighwayBus = 32	高速バス停
ConnectioBus = 64	連絡バス停
Port = 128	港

### ・駅種別

StationType [] Types

上記の駅種別が該当する数分設定されます。

### ・表示用駅名

String Name

### ・探索用駅名

String LongName

### ・路線名 (設定されません)

String LineName

### ・短縮駅名

String ShortName

### ・読み (ひらがな)

String Yomi

### ・ランドマークカテゴリ名

String LandmarkCategory

### ・ランドマーク最寄り駅情報

Station [] LandmarkNearestStation

・ランドマーク最寄り駅経路情報(設定されません)  
Course [] LandmarkNearestStationTransfe

## RouteSection クラス

経路の情報です。

・路線情報

Line Line

・路線区間列車情報

Train Train

この情報は使用できません。

・区間の開始となる駅の駅情報

Station StationFrom

・区間の終了となる駅の駅情報

Station StationTo

・路線種別

LineType LineType

・営業距離(単位:100m)

int Distance

・運営会社名

String CorpName

・乗り入れ路線判定

bool ConnectedSection

指定された路線区間が乗り入れ路線(前の路線区間と接続している)かどうか

True: 乗り入れ路線である

False: 乗り入れ路線でない

・戻り区間判定

bool BackSection

指定された路線区間が戻り区間かどうか

True: 戻り路線である

False: 戻り路線でない

・停車駅(設定されません)

Station [] StopStations

## FareSection クラス

区間運賃の情報です。

- ・区間の開始となる駅の駅情報

Station StationFrom

- ・区間の終了となる駅の駅情報

Station StationTo

- ・運賃情報

Fare [] FareList

片道運賃

Decimal FareList[i].OneWayFare

往復運賃

Decimal FareList[i].RoundTripFare

運賃種別

Fare.AirlineFareType FareList[i].airlineFareType

運賃種別名称

String FareList[i].FareTypeName

- ・運賃に関するコメント

String Comment

- ・ プロパティCommentが返す文字列は下記のとおりです。

文字列	説明
現金のみ	現金で支払った場合の運賃である場合、返されます。
参考運賃	航空時刻表の有効期限より前の発着日の経路の場合、返されます。
前期適用運賃	航空時刻表の有効期限を過ぎた発着日の経路の場合、返されます。
特定便割引	航空機の特定便割引の場合、返されます。

- ・ プロパティFareTypeNameが返す文字列は下記のとおりです。

運賃種別	運賃種別名称の文字列
0	空文字
1	特定便割引
2	普通運賃

## SurchargeSection クラス

区間特急運賃の情報です。

・区間の開始となる駅の駅情報

Station SurchargeFrom

・区間の終了となる駅の駅情報

Station SurchargeTo

・区間特急料金

Fare [] Surcharges

片道特急料金

Decimal Surcharges[i].OneWayFare

往復特急料金

Decimal Surcharges[i].RoundTripFare

特急料金種別

Decimal Surcharges[i].SurchargeSeatType

特急料金種別名称

Decimal Surcharges[i]. SurchargeSeatTypeName

・特急料金種別

0 未設定状態 (エラー)

1 指定席

2 自由席

3 グリーン

4 寝台

5 立席

6 特定特急料金

7 グリーン(車内購入)

8 定期 (ExpService04, ExpService05 にはありません。)

9 回数券 (ExpService04, ExpService05, ExpService06 にはありません。)

15 有料車両(寝台を除く) (ExpService04, ExpService05, ExpService06 にはありません。)

16 グランクラス(東北新幹線はやぶさのグランクラスに対応) (ExpService04, ExpService05, ExpService06 にはありません。)

17 スワローあかぎ料金 (ExpService04, ExpService05, ExpService06, ExpService07, ExpService08 にはありません。)

18 指定席(車内購入時) (ExpService04, ExpService05, ExpService06, ExpService07, ExpService08 にはありません。)

19 座席未指定料金 (ExpService04, ExpService05, ExpService06, ExpService07, ExpService08 にはありません。)

## PassSection クラス

定期の情報です。

・発駅

Station StationFrom

・着駅

Station StationTo

・1ヶ月定期料金

int OneMonth

・3ヶ月定期料金

int ThreeMonth

・6ヶ月定期料金

int SixMonth

・この区間の定期券運賃が他の区間の定期券運賃に含まれるかどうか  
bool FreePass

True: 含まれる

False: 含まれない

# Line クラス

経路の情報です。

## ・路線種別定義

LineType

none = 0	指定なし
RailRoad = 1	鉄道
AirLine = 2	空路
ConnectionBus = 4	連絡バス
Ship = 8	海路
Walk = 16	徒歩
Bus = 32	路線バス
HighwayBus = 64	高速バス
Extern = 128	ランドマーク・最寄駅等
MidnightBus = 256	深夜急行バス

## ・路線種別

LineType lineType

上記参照

## ・表示用路線名

String Name

## ・探索用路線名

String LongName

## ・短縮路線名

String ShortName

## ・運行会社名

Corporation OperationCorporation

## ・路線色(形式 00BBGRRh)

int Color

VC++のCOLORREF型と同じ

## ・設定されません。

Station [] StopStations



## Fare クラス

運賃の情報です。

### ・運賃種別定義

ExpService06 の場合は以下です。

AirlineFareType

other = 0	その他
AirLineRegular = 1	通常
AirLineTokuwari = 2	特割

ExpService04, ExpService05 の場合は以下です。

AirlineFareType

other = 0	その他
AirLineRegular = 1	特割
AirLineTokuwari = 2	通常

定義名称が説明と異なりますが、定義名称の誤りで、説明が正しいです。

### ・定期種別定義

PassType

Commuter = 1	通勤定期
Student = 2	学割
Highschool = 3	学割(高校)

### ・片道運賃

decimal OneWayFare

### ・往復運賃

decimal RoundTripFare

### ・運賃種別

AirlineFareType airlineFareType

### ・運賃種別名称

String FareTypeName

### ・特急料金種別

SeatType SurchaegeSeatType

- ・ プロパティFareTypeNameが返す文字列は下記のとおりです。

運賃種別	運賃種別名称の文字列
0	空文字
1	特定便割引
2	普通運賃

・特急料金種別定義

SeatType

none = 0	未設定(エラー)
Reserved = 1	指定席
Free = 2	自由席
Green = 3	グリーン
Bed = 4	寝台
Stands = 5	立席
SpecialExpress = 6	特定特急料金
unknown = 7	グリーン(車内購入) (ExpService04, ExpService05 の場合。) 定義名称に誤りがあります。
GreenInCar = 7	グリーン(車内購入) (ExpService06 の場合。)
Pass = 8	定期 (ExpService04, ExpService05 にはありません。)
Coupon = 9	回数券 (ExpService04, ExpService05, ExpService06 にはありません。)
Car = 15	有料車両(寝台を除く) (ExpService04, ExpService05, ExpService06 にはありません。)
GranClass = 16	グランクラス(東北新幹線はやぶさのグランクラスに対応) (ExpService04, ExpService05, ExpService06 にはありません。)
Suwarou = 17	スワローあかぎ料金 (ExpService04, ExpService05, ExpService06, ExpService07, ExpService08 にはありません。)
InCar = 18	指定席(車内購入時) (ExpService04, ExpService05, ExpService06, ExpService07, ExpService08 にはありません。)
NoneReserved = 19	座席未指定料金 (ExpService04, ExpService05, ExpService06, ExpService07, ExpService08 にはありません。)

・特急料金種別

0	未設定状態
1	指定席
2	自由席
3	グリーン
4	寝台
5	立席
6	特定特急料金
7	グリーン(車内購入)
8	定期 (ExpService04, ExpService05 にはありません。)
9	回数券 (ExpService04, ExpService05, ExpService06 にはありません。)
15	有料車両(寝台を除く) (ExpService04, ExpService05, ExpService06 にはありません。)
16	グランクラス(東北新幹線はやぶさのグランクラスに対応) (ExpService04, ExpService05, ExpService06 にはありません。)
17	スワローあかぎ料金 (ExpService04, ExpService05, ExpService06, ExpService07, ExpService08 にはありません。)
18	指定席(車内購入時) (ExpService04, ExpService05, ExpService06, ExpService07, ExpService08 にはありません。)
19	座席未指定料金 (ExpService04, ExpService05, ExpService06, ExpService07, ExpService08 にはありません。)

## Corporation クラス

運行会社の情報です。

- ・運行会社名

String Name

- ・設定されません。

LineType LineType

## Ticket クラス

乗車券の情報です。

(ExpService08 から。ExpService04～07 にはありません。)

### ・乗車券システム

TicketSystem

None = 0

未設定

Ordinary = 1

普通乗車券

IC = 2

ICカード乗車券

## Navigation クラス

経路検索時の各種設定をするためのクラスです。

RailGoのGetDefaultNavigation()により取得したものを基にしてください。

必要ならば変更してSearchCourse(), SearchCourseAll()に渡してください。

### ・利用についての設定

UseState

readily = 1	気軽に利用
ordinary = 2	普通に利用(初期値)
avoid = 3	極力利用しない

### ・利用についての考慮の設定

CareState

DontCare = 1	気にならない(初期値)
BitCareAbout = 2	少し気になる
RealyCare = 3	大いに気になる

### ・待ち時間の設定(ExpService04, ExpService05 にはありません。)

WaitTimeState

None = 0	待ち時間なし
Average = 1	平均待ち時間(初期値)

### ・表示順序種別(ExpService04 にはありません。)

SortType

Search = 1	探索順
Time = 2	時間順
Fare = 4	運賃順
Pass = 8	定期順

### ・経路検索要求回答数

int AnswerCount(ExpService04 では AnserCount となります。)  
初期値5

### ・出発/到着日

int TargetDate

初期値は本日です。

### ・出発/到着時刻

int TargetTime

初期値は指定なしです。設定することはできません。平均待ち時間探索を行います。

### ・出発/到着指定

bool Departure

true=出発指定(初期値) false=到着指定

・特急料金種別希望初期値

SeatType SurchargeKind(SurchargeKindDefault)  
Reserved(初期値), Free, Green のみ指定できます。

・定期タイプ

PassType PassType  
初期値 Commuter

・JR 路線の季節料金を無視するかどうか

bool IgnoreSeasonJR  
true=無視 false=無視しない(初期値)

・航空保険特別料金を運賃に含めるかどうか

bool IncludeInsuranceSurcharge  
true=含める false=含めない(初期値)

・乗車券システム種別(ExpService08 から。ExpService04~07 にはありません。)

TicketSystem TicketSystem  
None, Ordinary(初期値), IC  
IC カード乗車券を使用する場合はこの値を IC に指定してください。

・路線名あいまい指定

bool FuzzyLine  
true=無視する false=無視しない(初期値)  
路線を指定した場合、その路線が両端の駅に接続していない路線名の時、無視して探索するか／無視しないでエラーとするかを設定します。

・空路の利用

UseState Airplane  
初期値 ordinary

・特急列車の利用

UseState Express  
初期値 ordinary

・JR 路線の利用

UseState JR  
初期値 ordinary

・駅間徒歩

CareState Walk  
初期値 DontCare

・乗り換え

CareState Change  
初期値 DontCare

・路線バスの利用

bool UseLocalBus  
true=利用する(初期値) false=利用しない

・路線探索結果の表示順(ExpService04 にはありません。)  
SortType Sort  
初期値 Time

以下は ExpService04, ExpService05 にはありません。

・高速バスの利用  
UseState HighwayBus  
初期値 ordinary

・連絡バスの利用  
UseState ConnectionBus  
初期値 ordinary

・出発駅乗車待ち時間  
WaitTimeState WaitTime  
初期値 Average

・特急始発駅  
UseState StartStation  
初期値 ordinary  
avoid は ordinary になります。

・寝台列車の利用  
UseState SleepingTrain  
初期値 avoid  
ordinary は readily になります。

・路線バスのみ利用  
bool LocalBusOnly  
true=する false=しない(初期値)

## 4.エラーコード

例外発生時に、メッセージで以下の形式でコードが返される場合があります。 ”エラー:XXX”  
(XXX はエラーコード)

エラーコードとエラーの内容は以下になります。

- 1:指定駅が接続していない。
- 2:経由駅が不正。
- 3:指定が不正。
- 100:駅の指定がない。
- 101:駅名の指定が不正。
- 102:路線名が不正。
- 103:駅が重複。
- 104:路線バス不使用なのに、バス停が指定されている。
- 105:路線バスのみなのに、バス停以外が指定されている。
- 106:指定が不正。
- 107:乗車専用駅で降車指定されている。
- 108:降車専用駅で乗車指定されている。
- 109:運行日以外の日付が指定されている。
- 253:経路がない。
- 254:探索失敗。
- 255:障害発生。



## 5. サンプルコード

C#のサンプルです。言語・環境により詳細は異なります。

### 通常検索

```
// RailGo
RailGo.ExpService05 expsv = new RailGo.ExpService05();

// 認証情報を設定
RailGo.Authentication at = new RailGo.Authentication();
at.User = "User";
at.Password = "Password";
expsv.AuthenticationValue = at;

// 駅の情報 は SearchStationAll()でも取得できます。

// 出発駅の情報 を取得
RailGo.Station [] DepStations = expsv.SearchStation("新宿", RailGo.AreaType.Japan,
RailGo.StationType.RailRoad, 20070117);

// 到着駅の情報 を取得
RailGo.Station [] ArvStations = expsv.SearchStation("渋谷", RailGo.AreaType.Japan,
RailGo.StationType.RailRoad, 20070117);

// 選択した駅を設定 (例として、見つかった出発駅・到着駅のそれぞれ先頭を指定しています。)
RailGo.Station [] stations = new RailGo.Station[2];
stations[0] = DepStations[0];
stations[1] = ArvStations[0];

// デフォルトの検索オプションを取得
RailGo.Navigation navi = expsv.GetDefaultNavigation();

// 経路情報を取得
RailGo.Course [] cinfo = expsv.SearchCourse(stations, navi);

// 料金を表示
label1.Text = DepStations[0].LongName + "→" + ArvStations[0].LongName + " " +
cinfo[0].TotalFare.OneWayFare + "円";
```

## 一括検索

この機能は ExpService06 以降で使用できます。

```
// RailGo
RailGo.ExpService06 expsv = new RailGo.ExpService06();

// 認証情報を設定
RailGo.Authentication at = new RailGo.Authentication();
at.User = "User";
at.Password = "Password";
expsv.AuthenticationValue = at;

// 駅の候補(発駅3つ、着駅3つ)
string[] stationname = {"新宿", "渋谷", "品川", "東京", "大手町", "京橋"};

// 駅の情報 SearchStation()でも取得できます。

// 駅の情報取得
RailGo.Station [][] Stations = expsv.SearchStationAll(stationname, RailGo.AreaType.Japan,
RailGo.StationType.RailRoad, 20070117);

// 発駅に選択した駅を設定(例として、見つかった駅のそれぞれ先頭1つを指定しています。)
RailGo.Station [] DepStations = new RailGo.Station[3];
DepStations[0] = Stations[0][0];
DepStations[1] = Stations[1][0];
DepStations[2] = Stations[2][0];

// 着駅に選択した駅を設定(例として、見つかった駅のそれぞれ先頭1つを指定しています。)
RailGo.Station [] ArvStations = new RailGo.Station[3];
ArvStations[0] = Stations[3][0];
ArvStations[1] = Stations[4][0];
ArvStations[2] = Stations[5][0];

// デフォルトの検索オプションを取得
RailGo.Navigation navi = expsv.GetDefaultNavigation();

// 経路情報取得
RailGo.Course [] cinfo = expsv.SearchCourseAll(DepStations, ArvStations, navi);

// 料金を表示
label1.Text = DepStations[0].LongName + "→" + ArvStations[0].LongName + " " + " " +
cinfo[0].TotalFare.OneWayFare + "円";
```

## 6. バージョンによる違い

### ExpService05 での変更点

- ・経路検索要求回答数  
Navigation.AnserCount → Navigation.AnswerCount に変更。
- ・表示順序種別 Navigation.SortType 追加

### ExpService06 での変更点

- ・駅一括検索 SearchStationAll()を追加。
- ・経路一括検索 SearchCourseAll()を追加。
- ・待ち時間の設定 Navigation.WaitTimeState を追加。
- ・高速バスの利用 Navigation.HighwayBus を追加。
- ・連絡バスの利用 Navigation.ConnectionBus を追加。
- ・出発駅乗車待ち時間 Navigation.WaitTime を追加。
- ・特急始発駅 Navigation.StartStation を追加。
- ・寝台列車の利用 Navigation.SleepingTrain を追加。
- ・路線バスのみ利用 Navigation.LocalBusOnly を追加。
- ・運賃種別定義 Fare.AirlineFareTypeを以下に変更。  
other = 0            その他  
AirLineRegular = 1 通常  
AirLineTokuwari = 2特割
- ・特急料金種別定義 Fare.SeatType を変更。  
変更 unknown = 7 → GreenInCar = 7      グリーン(車内購入)  
追加 Pass = 8      定期

## ExpService07 での変更点

- ・SurchargeSection クラス

特急料金種別に回数券、有料車両(寝台を除く)、グランクラスを追加。

9 回数券

15 有料車両(寝台を除く)

16 グランクラス

- ・Line クラス

路線種別定義に深夜高速バスを追加。

MidnightBus = 256 深夜急行バス

## ExpService08 での変更点

- ・Ticket クラス

新設。

乗車券についての情報に関するクラス

乗車券システム種別

TicketSystem

を含む

- ・Navigation クラス

乗車券システム種別の設定 Navigation.TicketSystem 追加

## ExpService09 での変更点

- ・SurchargeSection クラス

特急料金種別にスワローあかぎ料金、指定席(車内購入時)、座席未指定料金を追加。

17 スワローあかぎ料金

18 指定席(車内購入時)

19 座席未指定料金

## ExpService10 での変更点

- ・StationType クラス

V10 以降で仕様変更。

V10 以降の StationType クラスの説明を追記。

StationType

none = 0

RailRoad = 1

AirPort = 2

LandMark = 4

BusStop = 8

RailRoadOnly = 16

HighwayBus = 32

ConnectioBus = 64

Port = 128

未設定状態(エラー値)

鉄道駅(高速バス停、連絡バス停、港を含む)

空港

ランドマーク(SearchStationでは指定できません。)

路線バス停

鉄道駅のみ

高速バス停

連絡バス停

港