

**人の流れプロジェクト  
時空間内挿 WebAPI  
仕様書**

2024 年 3 月

**東京大学 空間情報科学研究センター**

## 目次

<b>1. API 仕様</b> .....	<b>1</b>
1.1. WebAPI 一覧 .....	1
1.2. CreateSession .....	2
1.2.1. CreateSession のエラーコード .....	2
1.3. DestroySession .....	3
1.3.1. DestroySession のエラーコード .....	3
1.4. GetNearestRoadPoint .....	4
1.4.1. GetNearestRoadPoint の戻り値詳細 .....	5
1.4.2. GetNearestRoadPoint のエラーコード .....	5
1.5. GetRoadRoute .....	6
1.5.1. GetRoadRoute の戻り値詳細 .....	7
1.5.2. GetRoadRoute のエラーコード .....	7
1.5.3. GetRoadRoute 詳細 .....	8
1.6. GetRailRoute .....	11
1.6.1. GetRailRoute の戻り値詳細 .....	12
1.6.2. GetRailRoute のエラーコード .....	13
1.6.3. GetRailRoute 詳細 .....	13
1.7. GetBusRoute .....	14
1.7.1. GetBusRoute の戻り値詳細 .....	15
1.7.2. GetBusRoute のエラーコード .....	17
1.7.3. GetBusRoute 詳細 .....	17
1.8. GetMixedRoute .....	18
1.8.1. GetMixedRoute 戻り値詳細 .....	19
1.8.2. GetMixedRoute のエラーコード .....	20
1.8.3. GetMixedRoute 詳細 .....	21
1.9. GetSTInterpolatedPoints .....	22
1.9.1. GetSTInterpolatedPoints の戻り値詳細 .....	24
1.9.2. GetSTInterpolatedPoints のエラーコード .....	24
1.10. コード表 .....	26
1.10.1. 交通手段コード一覧 .....	26
1.10.2. 道路種別コード一覧 .....	26
1.11. システムエラーコード .....	27

<b>2. 過去 API（動線解析）からの変更点</b> .....	<b>27</b>
2.1. 道路の経路探索 .....	27
2.2. GetNearestRoadPoint .....	27
2.3. GetRailRoute .....	27

## 1. API 仕様

### 1.1. WebAPI 一覧

API 名称	機能
CreateSession	JoRAS 認証を行い、セッションを取得する。
DestroySession	セッションを破棄しログアウトする。
GetNearestRoadPoint	任意の座標点より、道路ネットワークにおけるその位置の近傍座標を取得する。
GetRoadRoute	任意の点座標（起点、終点、経由点）より、それら地点を結ぶ道路経路を取得する。
GetRailRoute	任意の駅名または駅座標（起点、終点、経由点）より、それら地点を結ぶ鉄道経路を取得する。
GetMixedRoute	任意の点座標（起点、終点、経由点）より、それら地点を結ぶ道路及び鉄道経路を取得する。 鉄道を主とした公共交通手段を基本として経路を選択する。（鉄道の利用が必要条件となる）
GetMixedRouteTime	GetMixedRoute と統合したため、廃止した。
GetSTInterpolatedPoints	任意の時空間位置（起点、終点の時空間位置）より、それらを内挿する時空間位置を取得する。

## 1.2. CreateSession

API名	CreateSession					
URL	http://pflow-api.csis.u-tokyo.ac.jp/webapi/CreateSession					
機能	JoRAS 認証を行い、セッションを取得する。					
Content-Type	text/plain					
文字コード	Shift-JIS					
データ形式	URL エンコード					
リクエスト	POST					
引数	No	変数	項目名	型式	必須	備考
	1	UserID	ユーザ ID	Char	○	JoRAS 認証のアカウント
	2	Password	パスワード	Char	○	
戻り値	<ul style="list-style-type: none"> <li>・ステータス ※正常終了：1、異常終了：エラー番号</li> <li>・セッション ID</li> </ul>					
戻り値の詳細	テキスト形式のテキストデータ。詳細は次の通り					
	行	項目順	項目名	型式	備考	
	1	1	ステータス	Integer	正常終了：1、異常終了：エラー番号	
		2	セッション ID	Char		
解説	<p>受け取った引数を使用して JoRAS 認証を行う。</p> <p>URL : <a href="http://joras.csis.u-tokyo.ac.jp:8080/login">http://joras.csis.u-tokyo.ac.jp:8080/login</a></p> <p>認証が成功：Bearer トークンと、ユーザのフルネームを含む JSON が返答される。</p> <p>認証が失敗：422 の Auth エラーになる。</p>					

### 1.2.1. CreateSession のエラーコード

エラー番号	内容
1999	システムエラー
1001	ユーザ ID が未入力
1002	パスワードが未入力
1011	JoRAS 認証エラー

### 1.3. DestroySession

API名	DestroySession					
URL	http://pflow-api.csis.u-tokyo.ac.jp/webapi/DestroySession					
機能	セッションを破棄しログアウトする。					
Content-Type	text/plain					
文字コード	Shift-JIS					
データ形式	URL エンコード					
リクエスト	POST					
引数	No	変数	項目名	型式	必須	備考
	1	SessionID	セッション ID	Char	○	
戻り値	・ステータス ※正常終了：1、異常終了：エラー番号					
戻り値の詳細	テキスト形式のテキストデータ。詳細は次の通り					
	行	項目順	項目名	型式	備考	
	1	1	ステータス	Integer	正常終了：1、異常終了：エラー番号	
解説	<ul style="list-style-type: none"> <li>・セッションが切れる前に強制的にセッションを破棄する。</li> <li>・引数「セッション ID」には、CreateSession にて取得した値を入力。</li> </ul>					

#### 1.3.1. DestroySession のエラーコード

エラー番号	内容
2999	システムエラー
2001	セッション ID が未入力
2011	セッション ID が存在しない

## 1.4. GetNearestRoadPoint

API名	GetNearestRoadPoint					
URL	http://pflow-api.csis.u-tokyo.ac.jp/webapi/GetNearestRoadPoint					
機能	任意の座標点より、道路ネットワークにおけるその位置の近傍座標を取得する。					
Content-Type	text/plain					
文字コード	UTF-8					
データ形式	URL エンコード					
リクエスト	POST					
引数	No	変数	項目名	型式	必須	備考
	1	UnitTypeCode	座標単位指定	Integer	○	1：度分秒 2：度
	2	PosLongitude	点座標：経度	Double	○	世界測地系
	3	PosLatitude	点座標：緯度	Double	○	
	4	RoadKindCode	道路種別コード	Integer	－	表 1.10.2. 道路種別コード参照
	5	RoadNo	路線番号	Integer	－	道路ネットワークに準拠
戻り値	<ul style="list-style-type: none"> <li>・正常終了時：GeoJSON形式のテキスト、異常終了時：エラー番号</li> <li>・マッチング処理による近傍点座標および道路情報</li> <li>・詳細は GetNearestRoadPoint の戻り値詳細を参照</li> </ul>					
解説	<p>○引数「座標単位指定」で入力した値により、入出力の経度、緯度項目の形式が異なる</p> <ul style="list-style-type: none"> <li>・座標単位指定=1の場合：度分秒単位 dddmmss.sss &lt;例&gt;139° 51' 8.152" → 1395108.152</li> <li>・座標単位指定=2の場合：度単位 ddd.dddddddd &lt;例&gt;139.85226444</li> </ul> <p>○引数「道路種別」、「路線番号」は任意入力とし、未入力の場合は全道路を探索の対象とする。</p> <p>○探索は任意点より 1km 以内の範囲に対して行う。指定した点より半径 1km 以内にネットワークデータが見つからない場合にはエラーとなる。</p> <p>○戻り値の「路線番号」は、路線番号の存在しない路線の場合には 0 を出力する。</p>					

#### 1.4.1. GetNearestRoadPoint の戻り値詳細

項目名	内容	型	備考
status	ステータス	整数	正常終了：1
features	図形の配列	配列	要素数は常に 1
type	Feature	文字	
properties	属性	オブジェクト	近傍点を落とした DRM リンクの情報
dourosyubetsu	道路種別コード	整数	表 1.10.2. 道路種別コード一覧参照
rosen_no	路線番号	整数	路線番号の存在しない路線の場合には 0
mesh2	2 次メッシュコード	整数	
geometry	図形	オブジェクト	
type	Point	文字	
coordinates	座標	配列	・座標は 1 点のみ ・世界測地系緯度経度

#### 1.4.2. GetNearestRoadPoint のエラーコード

エラー番号	内容
3999	システムエラー
3000	近傍点が探索範囲内にない
3001	「座標単位指定」が未入力
3002	「点座標：経度」が未入力
3003	「点座標：緯度」が未入力
3011	「座標単位指定」が 1 または 2 以外の値
3012	「点座標：経度」が数値以外の値
3013	「点座標：緯度」が数値以外の値
3014	「道路種別コード」が数値以外の値
3015	「路線番号」が数値以外の値



## 1.5. GetRoadRoute

API名	GetRoadRoute					
URL	http://pflow-api.csis.u-tokyo.ac.jp/webapi/GetRoadRoute					
機能	任意の点座標（起点、終点、経由点）より、それら地点を結ぶ道路経路を取得する。					
Content-Type	text/plain					
文字コード	UTF-8					
データ形式	URL エンコード					
リクエスト	POST					
引数	No	変数	項目名	型式	必須	備考
	1	UnitTypeCode	座標単位指定	Integer	○	1：度分秒 2：度
	2	StartLongitude	起点：経度	Double	○	世界測地系
	3	StartLatitude	起点：緯度	Double	○	
	4	GoalLongitude	終点：経度	Double	○	
	5	GoalLatitude	終点：緯度	Double	○	
	6	WayLongitude	経由点：経度	Double	－	
	7	WayLatitude	経由点：緯度	Double	－	
	8	RoadKindCode	道路種別コード	Integer	－	表 1.10.2. 道路種別コード参照
	9	RoadNo	路線番号	Integer	－	道路ネットワークに準拠
	10	TransportCode	交通手段コード	Integer	－	表 1.10.1. 交通手段コード参照
11	OutputNum	出力経路数	Integer	－	最大 3 指定可能	
戻り値	<ul style="list-style-type: none"> <li>・正常終了時：GeoJSON 形式のテキスト、異常終了時：エラー番号</li> <li>・マッチング処理による経路</li> <li>・詳細は 1.5.1. GetRoadRoute の戻り値詳細参照</li> </ul>					
解説	<p>○引数「座標単位指定」で入力した値により、入出力の経度、緯度項目の形式が異なる</p> <ul style="list-style-type: none"> <li>・座標単位指定=1 の場合：度分秒単位 dddmmss.sss &lt;例&gt;139° 51' 8.152" → 1395108.152</li> <li>・座標単位指定=2 の場合：度単位 ddd.dddddddd &lt;例&gt;139.85226444</li> </ul> <p>○引数「道路種別」、「路線番号」は任意入力とし、入力された場合、指定した道路種別、路線番号が優先的に使用されるようになる。</p> <p>○引数「交通手段コード」は任意入力とする。未入力の場合は全道路を探索の対象とし、移動</p>					

	<p>速度を無視して距離だけで処理する。(時間優先の最短経路と距離優先の最短経路の出力結果が同じ経路となる)</p> <p>○引数「出力経路数」は任意入力とし、入力値に対する出力は次のとおりである。</p> <p>1：時間優先の最短経路</p> <p>2：時間優先の最短経路、距離優先の最短経路</p> <p>3：時間優先の最短経路、距離優先の最短経路、料金優先の最短経路（有料道路を利用しない経路）未入力の場合は、1 の出力とする。</p> <p>○料金(fare)は通過した高速自動車国道の延長から以下の計算式で算出される。</p> <p>料金 = 通過した高速自動車国道の延長(km) * 24.6 * 1.05 + 150</p>
--	---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

### 1.5.1. GetRoadRoute の戻り値詳細

項目名	内容	型	備考
status	ステータス	整数	正常終了：1
num_route	経路数	整数	
features	図形の配列	配列	出力経路数分の配列
type	Feature	文字	1 経路分のポリラインに相当
properties	属性	オブジェクト	
id	経路 No	整数	
length	延長	実数	m
num_point	座標数	整数	
required_time	所要時間	実数	分
fare	料金	整数	通過した高速自動車国道の延長(km) * 24.6 * 1.05 + 150
geometry	図形	オブジェクト	
type	LineString	文字	
coordinates	座標	配列	世界測地系緯度経度

### 1.5.2. GetRoadRoute のエラーコード

エラー番号	内容
5999	システムエラー
5000	近傍点がないまたはネットワークが繋がっていない
5001	「座標単位指定」が未入力
5002	「起点：経度」が未入力
5003	「起点：緯度」が未入力

5004	「終点：経度」が未入力
5005	「終点：緯度」が未入力
5011	「座標単位指定」が1または2以外の値
5012	「起点：経度」が数値以外の値
5013	「起点：緯度」が数値以外の値
5014	「終点：経度」が数値以外の値
5015	「終点：緯度」が数値以外の値
5016	「経由点：経度」が数値以外の値
5017	「経由点：緯度」が数値以外の値
5018	「道路種別コード」が数値以外の値
5019	「路線番号」が数値以外の値
5020	「交通手段コード」が数値以外の値
5021	「出力経路数」が4以上または数値以外の値

表 1.5-1 GetRoadRoute のエラーコード

### 1.5.3. GetRoadRoute 詳細

#### (1) 交通手段と速度

GetRoadRoute では、TransportCode(交通手段コード)で指定した交通手段とリンクの道路種別に応じて通過時間が算出される。

交通手段コードと道路種別ごとのリンク通過速度は「交通手段と速度.csv」で定義されている。

「交通手段と速度.csv」(一部抜粋)

	道路種別	徒歩	自転車	乗用車	距離優先	料金優先
交通手段コード		1	2	6	100	101
都市間高速	1	-1	-1	100	30	5
都市高速	2	-1	-1	70	30	5
国道	3	6	10	50	30	30
主要地方道	4	6	10	45	30	30
主要市道	5	6	10	45	30	30
県道	6	6	10	40	30	30
市道	7	6	10	30	30	30
街路	8	6	10	30	30	30
その他	9	6	10	30	30	30

例) 交通手段コード 6(乗用車)で国道を走行する場合、時速 50km で走行するものとして所要時間が算出される。

なお、速度が「-1」の場合、そのリンクを走行することはできない。

## (2) リンク属性とリンクの通行可/不可

また、TransportCode(交通手段コード)で指定した交通手段とリンクの属性(リンク種別、リンク通行可・不可)の組み合わせで、リンクを通行できるかどうかが決まる。

### a) リンク種別

リンク種別が「自転車道等リンク」の場合、交通手段が徒歩(交通手段コード 1)または自転車(交通手段コード 2)以外では通行不可とする。

ただし、交通手段が「距離優先(17)」、「料金優先(18)」の場合、上記の制限には該当しないものとする。

以下にリンク種別と交通手段による通行可/不可の一覧を示す。

リンク種別		車両通行 が可能か	徒歩/自転車通行 が可能か
内容	コード		
本線(上下線非分離)リンク	1	○	○
本線(上下線分離)リンク	2	○	○
連結路(本線間の渡り線)リンク	3	○	○
交差点内リンク	4	○	○
連結路(ランプ)リンク	5	○	○
本線と同一路線の側道リンク	6	○	○
S A等側線リンク	7	○	○
自転車道等リンク	8	×	○
本線側道接続リンク	9	○	○
未調査	0	○	○

## b) リンク通行可・不可

交通手段とリンク通行可・不可の組み合わせでリンクの通行可/不可が決まる。

- リンクの通行可・不可コードが「未共用」または「計画決定」のリンク
  - →すべての交通手段で通行不可。
- リンクの通行可・不可コードが「自動車通行不可」のリンク
  - →交通手段が徒歩(交通手段コード1)または自転車(交通手段コード2)以外では通行不可。
  - ただし、交通手段が「距離優先(17)」、「料金優先(18)」の場合、上記の制限には該当しないものとする。

リンク通行可・不可		車両通行	徒歩/自転車通行
内容	コード		
自動車通行可	1	○	○
自動車通行不可	2	×	○
未共用(含工事中)	3	×	×
計画決定	4	×	×
供用中(自動車通行可・不可は未調査)	0	○	○

## 1.6. GetRailRoute

API名	GetRailRoute					
URL	http://pflow-api.csis.u-tokyo.ac.jp/webapi/GetRailRoute					
機能	任意の駅名または駅座標（起点、終点、経由点）より、それら地点を結ぶ鉄道経路を取得する。					
Content-Type	text/plain					
文字コード	UTF-8					
データ形式	URL エンコード					
リクエスト	POST					
引数	No	変数	項目名	型式	必須	備考
	1	UnitTypeCode	座標単位指定	Integer	○	1：度分秒 2：度
	2	GeoOptionCode	位置単位指定	Integer	○	1：駅名指定 2：座標指定
	3	StartStation	起点：駅名	Char	○	GeoOptionCode が 1 の場合に指定
	4	GoalStation	終点：駅名	Char	○	
	5	WayStation	経由点：駅名	Char	－	
	6	StartLongitude	起点：経度	Double	○	GeoOptionCode が 2 の場合に指定 世界測地系
	7	StartLatitude	起点：緯度	Double	○	
	8	GoalLongitude	終点：経度	Double	○	
	9	GoalLatitude	終点：緯度	Double	○	
	10	WayLongitude	経由点：経度	Double	－	
	11	WayLatitude	経由点：緯度	Double	－	
	12	AppDate	日付	Char	－	西暦年月日 YYYYMMDD
	13	AppTime	時刻	Char	－	時分 hhmm
14	StartGoalType	出発／到着指定	Integer	－	1：出発、 2：到着	
戻り値	<ul style="list-style-type: none"> <li>・正常終了時：GeoJSON 形式のテキスト、異常終了時：エラー番号</li> <li>・マッチング処理による経路と駅情報</li> <li>・詳細は 1.6.1. GetRailRoute の戻り値詳細参照</li> </ul>					
解説	<p>○引数「座標単位指定」で入力した値により、入出力の経度、緯度項目の形式が異なる</p> <ul style="list-style-type: none"> <li>・座標単位指定=1 の場合：度分秒単位 dddmmss.sss &lt;例&gt;139° 51' 8.152" → 1395108.152</li> <li>・座標単位指定=2 の場合：度単位 ddd.ddddddd &lt;例&gt;139.85226444</li> </ul>					

	<p>○引数「位置単位指定」で入力した値により、駅の指定方法が異なる。</p> <ul style="list-style-type: none"> <li>・位置単位指定=1 の場合：駅名称により指定</li> <li>・位置単位指定=2 の場合：駅座標（経度、緯度）により指定</li> </ul> <p>○引数「日付」、「時刻」、「出発／到着指定」は任意入力とし、未入力の場合は平均所要時間で探索する。</p> <p>○平均所要時間の場合、戻り値の到着時刻(arrival_time)、出発時刻(departure_time)は空白になる。</p> <p>○引数「日付」、「時刻」、「出発／到着指定」はいずれかが入力された場合は全て必須とする。</p> <p>○戻り値の図形は経路の駅情報であり、起点から経路に沿った順に出力される。</p>
--	---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

### 1.6.1. GetRailRoute の戻り値詳細

項目名	内容	型	備考
status	ステータス	整数	正常終了：1
num_station	利用駅数	整数	
total_time	総所要時間	整数	分
fare	合計運賃	整数	
features	図形の配列	配列	駅の数分の配列
type	Feature	文字	ひとつの駅に相当
properties	属性	オブジェクト	
station	駅名	文字	
railway	路線名	文字	最初の駅では空
arrival_time	到着時刻	文字	<ul style="list-style-type: none"> <li>・最初の駅では空の文字列</li> <li>・平均所要時間検索では空の文字列</li> </ul>
departure_time	出発時刻	文字	<ul style="list-style-type: none"> <li>・最後の駅では空の文字列</li> <li>・平均所要時間検索では空の文字列</li> </ul>
required_time	所要時間	整数	<ul style="list-style-type: none"> <li>・最初の駅では 0</li> <li>・2 番目の駅では到着時刻と最初の駅の出発時刻の差</li> <li>・3 番目以降の駅では到着時刻と前の駅の到着時刻の差</li> </ul>
geometry	図形	オブジェクト	
type	Point	文字	
coordinates	座標	座標	世界測地系緯度経度

### 1.6.2. GetRailRoute のエラーコード

エラー番号	内容
6999	システムエラー
6001	「座標単位指定」が未入力
6002	「起点：駅名」が未入力
6003	「終点：駅名」が未入力
6005	「日付」が未入力（出発／到着指定が1または2の場合のチェック）
6006	「時刻」が未入力（出発／到着指定が1または2の場合のチェック）
6011	「座標単位指定」が1または2以外の値
6015	「日付」がYYYYMMDD形式以外の値
6016	「時刻」がhhmm形式以外の値
6017	「出発／到着指定」が1または2以外の値
6019	「日付」「時刻」「出発／到着指定」が不正（日付、時刻、出発／到着指定のいずれかが入力されている場合のチェック）
6021	「位置単位指定」が未入力
6022	「起点：経度」が未入力
6023	「起点：緯度」が未入力
6024	「終点：経度」が未入力
6025	「終点：緯度」が未入力
6031	「位置単位指定」が1または2以外の値
6032	「起点：経度」が数値以外の値
6033	「起点：緯度」が数値以外の値
6034	「終点：経度」が数値以外の値
6035	「終点：緯度」が数値以外の値
6036	「経由点：緯度」が数値以外の値
6037	「経由点：緯度」が数値以外の値

### 1.6.3. GetRailRoute 詳細

#### (1) 駅名の指定について

駅名指定の経路探索で、複数の県に同じ名前の駅が存在する場合、「駅名(県名)」のように県名を付加して指定する必要がある。また、同じ名称が同一県の場合、「駅名(路線名)」のように路線名を付加する必要がある。



## 1.7. GetBusRoute

API名	GetBusRoute					
URL	http:// pflow-api.csis.u-tokyo.ac.jp /webapi/GetBusRoute					
機能	任意のバス停名またはバス停座標（起点、終点、経由点）より、それら地点を結ぶ路線バス経路を取得する。					
Content-Type	text/plain					
文字コード	UTF-8					
データ形式	URL エンコード					
リクエスト	POST					
引数	No	変数	項目名	型式	必須	備考
	1	UnitTypeCode	座標単位指定	Integer	○	1：度分秒 2：度
	2	GeoOptionCode	位置単位指定	Integer	○	1：バス停名指定 2：座標指定
	3	StartStation	起点：バス停名	Char	○	GeoOptionCode が 1 の場合に指定
	4	GoalStation	終点：バス停名	Char	○	
	5	WayStation	経由点：バス停名	Char	－	
	6	StartLongitude	起点：経度	Double	○	GeoOptionCode が 2 の場合に指定 世界測地系
	7	StartLatitude	起点：緯度	Double	○	
	8	GoalLongitude	終点：経度	Double	○	
	9	GoalLatitude	終点：緯度	Double	○	
	10	WayLongitude	経由点：経度	Double	－	
	11	WayLatitude	経由点：緯度	Double	－	
	12	AppDate	日付	Char	－	西暦年月日 YYYYMMDD
	13	AppTime	時刻	Char	－	時分 hhmm
14	StartGoalType	出発／到着指定	Integer	－	1：出発、 2：到着	
戻り値	<ul style="list-style-type: none"> <li>・ステータス ※正常終了：1、異常終了：エラー番号</li> <li>・マッチング処理による経路とバス停情報</li> <li>・戻り値の詳細は 1.7.1. GetBusRoute の戻り値詳細参照</li> </ul>					
解説	<p>○引数「座標単位指定」で入力した値により、入出力の経度、緯度項目の形式が異なる</p> <ul style="list-style-type: none"> <li>・座標単位指定=1 の場合：度分秒単位 dddmss.sss &lt;例&gt;139° 51' 8.152" →</li> </ul>					

	<p>1395108.152</p> <ul style="list-style-type: none"> <li>・座標単位指定=2 の場合：度単位 ddd.dddddddd &lt;例&gt;139.85226444</li> </ul> <p>○引数「位置単位指定」で入力した値により、バス停の指定方法が異なる。</p> <ul style="list-style-type: none"> <li>・位置単位指定=1 の場合：バス停名称により指定</li> <li>・位置単位指定=2 の場合：バス停座標（経度、緯度）により指定</li> </ul> <p>○引数「日付」、「時刻」、「出発／到着指定」は任意入力とし、未入力の場合は平均所要時間で探索する。</p> <p>○平均所要時間の場合、戻り値の 3～N 行目の到着時刻、出発時刻は空白になる。</p> <p>○引数「日付」、「時刻」、「出発／到着指定」はいずれかが入力された場合は全て必須とする。</p> <p>○戻り値の図形は経路のバス停情報であり、起点から経路に沿った順に出力される。</p>
--	---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

### 1.7.1. GetBusRoute の戻り値詳細

項目名	内容	型	備考	
status	ステータス	整数	正常終了：1	
num_station	利用バス停数	整数		
total_time	総所要時間	整数	分	
fare	合計運賃	整数		
features	図形の配列	配列	バス停の数分の配列	
	type	Feature	文字	ひとつのバス停に相当
	properties	属性	オブジェクト	
	bus_stop	バス停名	文字	
	bus_route	路線名	文字	最初のバス停では空
	arrival_time	到着時刻	文字	<ul style="list-style-type: none"> <li>・最初のバス停では空の文字列</li> <li>・平均所要時間検索では空の文字列</li> </ul>
	departure_time	出発時刻	文字	<ul style="list-style-type: none"> <li>・最後のバス停では空の文字列</li> <li>・平均所要時間検索では空の文字列</li> </ul>

	required_time	所要時間	整数	<ul style="list-style-type: none"> <li>・最初のバス停では0</li> <li>・2番目のバス停では到着時刻と最初のバス停の出発時刻の差</li> <li>・3番目以降のバス停では到着時刻と前のバス停の到着時刻の差</li> </ul>
	geometry	図形	オブジェクト	
	type	Point	文字	
	coordinates	座標	座標	世界測地系緯度経度

## 1.7.2. GetBusRoute のエラーコード

GetRailRoute と同じなので、1.6.2. GetRailRoute のエラーコードを参照のこと。

## 1.7.3. GetBusRoute 詳細

### (1) バス停名の指定について

バス停名指定の経路探索では、駅すばあとの仕様により、「バス停名/会社名」の表記で指定する必要がある。

### (2) バス停名の例

アーツ千代田 3 3 3 1 / 日立自動車交通
青砥駅入口 / 京成バス
青砥駅東交差点 / 京成バス
青戸公団 / 京成バス
青戸車庫前 / 都営バス
青戸中央憩い交流館 / 京成バス
青戸平和公園 / 京成バス
赤土小学校前 / 都営バス
秋葉原駅中央口 / 日立自動車交通
秋葉原駅前 / 都営バス
アクロシティ / 東京都荒川区
浅草一丁目 / 都営バス
浅草駅 / 東京都台東区
浅草雷門 / 都営バス
浅草雷門 / 東京都台東区
浅草雷門南 / 都営バス
浅草菊水通り / 東京都台東区
浅草警察署前 / 都営バス
浅草警察署前 / 東京都台東区
浅草公園六区 / 都営バス

## 1.8. GetMixedRoute

バージョン 2.0 で以下の変更を行った。

- GetBusRoute の追加に伴い、公共交通機関の経路を鉄道、バス、鉄道またはバスから指定できるようにした。
- 戻り値に旧 GetMixedRouteTime の内容を取り込んだ。これによりバージョン 2.0 では GetMixedRouteTime を廃止した。

API 名	GetMixedRoute					
URL	http://pflow-api.csis.u-tokyo.ac.jp/webapi/GetMixedRoute					
機能	任意の点座標（起点、終点、経由点）より、それら地点を結ぶ道路及び鉄道/バス経路を取得する。 鉄道/バスを主とした公共交通手段を基本として経路を選択する。（鉄道/バスの利用が必要条件となる）					
Content-Type	text/plain					
文字コード	UTF-8					
データ形式	URL エンコード					
リクエスト	POST					
引数	No	変数	項目名	型式	必須	備考
	1	UnitTypeCode	座標単位指定	Integer	○	1：度分秒 2：度
	2	StartLongitude	起点：経度	Double	○	世界測地系
	3	StartLatitude	起点：緯度	Double	○	
	4	GoalLongitude	終点：経度	Double	○	
	5	GoalLatitude	終点：緯度	Double	○	
	6	WayLongitude	経由点：経度	Double	—	
	7	WayLatitude	経由点：緯度	Double	—	
8	TransportCode	使用交通機関 コード	Integer	○	1：鉄道のみ 2：バスのみ 3：鉄道またはバス	
戻り値	<ul style="list-style-type: none"> <li>・正常終了時：GeoJSON 形式のテキスト、異常終了時：エラー番号</li> <li>・マッチング処理による経路と駅/バス停情報</li> <li>・戻り値の詳細は 1.8.1. GetMixedRoute 戻り値詳細参照</li> </ul>					
解説	<p>○引数「座標単位指定」で入力した値により、入出力の経度、緯度項目の形式が異なる</p> <ul style="list-style-type: none"> <li>・座標単位指定=1 の場合：度分秒単位 dddmmss.sss &lt;例&gt;139° 51' 8.152" → 1395108.152</li> </ul>					

	<ul style="list-style-type: none"> <li>・座標単位指定=2 の場合：度単位 ddd.dddddddd &lt;例&gt;139.85226444</li> <li>○本 API は、鉄道または路線バスの利用を前提としている。</li> <li>○戻り値の図形は経路の駅/バス停情報であり、起点から経路に沿った順に出力される。</li> <li>○「使用交通機関コード」は座標から最寄りの公共交通機関を検索する際に使用される。 <ul style="list-style-type: none"> <li>「1：鉄道のみ」の場合、検索対象となるのは鉄道駅のみ</li> <li>「2：バスのみ」の場合、検索対象となるのはバス停のみ</li> <li>「3：鉄道またはバス」の場合、鉄道駅とバス停の両方が検索対象となり、座標に近いほうが採用される。</li> </ul> </li> </ul>
--	--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

### 1.8.1. GetMixedRoute 戻り値詳細

バージョン 2.0 で GetMixedRoute の戻り値に旧 GetMixedRouteTime の結果を取り込むようにした。

項目名	内容	型	備考	
status	ステータス	整数	正常終了：1	
num_station	利用駅/バス停数	整数		
num_point	座標数	整数		
fare	合計運賃	整数		
total_time	総所要時間	実数	分	旧 GetMixedRouteTime の内容
length_to_start_station	起点から出発駅/ バス停までの道 路の総延長	整数	m	
time_to_start_station	起点から出発駅/ バス停までの道 路の総所要時間	実数	分	
total_transport_time	鉄道/バスの総所 要時間	実数	分	
length_to_end_station	到着駅/バス停か ら終点までの道 路の総延長	整数	m	
time_to_end_station	到着駅/バス停か ら終点までの道 路の所要時間	実数	分	
features	図形の配列	配列		
type	Feature	文字		
properties	属性	オブジェクト		

	id	座標 No	整数	座標 No
	transportation	移動方法	整数	1:道路、2:鉄道、3:バス、4:徒歩、0:それ以外
	station	通過駅/バス停情報	オブジェクト	道路の場合、null
	station_name	駅/バス停名	文字	
	route	路線名	文字	最初の鉄道駅/バス停では空徒歩の場合、"徒歩"
	geometry	図形	オブジェクト	
	type	Point	文字	
	coordinates	座標	座標	世界測地系緯度経度

### 1.8.2. GetMixedRoute のエラーコード

エラー番号	内容
11999	システムエラー
11000	近傍点がないまたはネットワークがつながっていない
11001	座標単位指定が未入力
11002	起点：経度が未入力
11003	起点：緯度が未入力
11004	終点：経度が未入力
11005	終点：緯度が未入力
11011	座標単位指定が 1 または 2 以外の値
11012	起点：経度が数値以外の値
11013	起点：緯度が数値以外の値
11014	終点：経度が数値以外の値
11015	終点：緯度が数値以外の値
11016	経由点：経度が数値以外の値
11017	経由点：緯度が数値以外の値
11018	使用交通機関コードの書式エラー

### 1.8.3. GetMixedRoute 詳細

#### (1) 処理の流れ

- 最初に「使用交通機関コード」に応じた公共交通機関での経路探索を実施する。公共交通機関の経路探索に失敗した場合、道路の経路探索を行う。
  - 公共交通機関の経路探索に失敗した場合、道路の経路探索では、経由点を指定していても無視され起点～終点の経路探索が実施される。
- 公共交通機関の経路探索に成功した場合、起点～起点駅/バス停、終点駅/バス停～終点の道路の経路探索が行われる。
- 道路の経路探索では交通手段コード 17 (=距離優先、すべての道路が通行可能で速度は一律 30km/h) が使用される。

#### (2) 鉄道の経路探索に使用する日時、出発/到着

鉄道/バスの経路探索に使用する日時、出発/到着は Tomcat の設定ファイル webapi.properties で指定されている。

```
#鉄道の経路探索 日付  
railall.date=20240401
```

```
#鉄道の経路探索 時刻  
railall.time=0830
```

```
#鉄道の経路探索 出発/到着指定  
railall.startgoaltype=1
```

railall.startgoaltype は、railall.date と railall.time を出発日時として扱うか到着日時として扱うかを表しており、上記の設定では 2024/04/01 08:30 出発時の経路が探索される。



## 1.9. GetSTInterpolatedPoints

バージョン 2.0 で以下の変更を行った。

- GetBusRoute の追加に伴い、NetworkOption=4 を任意の座標列からバスの経路座標列に変更し、任意の座標列は NetworkOption=5 とした。

API 名	GetSTInterpolatedPoints					
URL	http://pflow-api.csis.u-tokyo.ac.jp/webapi/GetSTInterpolatedPoints					
機能	任意の時空間位置（起点、終点の時空間位置）より、それらを内挿する時空間位置を取得する。					
Content-Type	text/plain					
文字コード	UTF-8					
データ形式	URL エンコード					
リクエスト	POST					
引数	No	変数	項目名	型式	必須	備考
	1	UnitTypeCode	座標単位指定	Integer	○	1:度分秒 2:度
	2	Resolution	内挿時間間隔	Integer	—	内挿する時間解像度(sec)を指定 未指定の場合は 60(sec)
	3	DtStart	起点日時	Char	○	起点の出発日時を yyyyMMddHHmmss 形式で指定
	4	DtEnd	終点日時	Char	○	終点の出発日時を yyyyMMddHHmmss 形式で指定
	5	StartLatitude	起点緯度	Double	○	世界測地系
	6	StartLongitude	起点経度	Double	○	
	7	GoalLatitude	終点緯度	Double	○	
	8	GoalLongitude	終点経度	Double	○	
	9	NetworkOption	経路座標列指定フラグ	Integer	—	※解説参照。デフォルトは 1 1:鉄道/バス+道路の経路座標列を使用

						2:道路経路座標列を使用 3:鉄道経路座標列を使用 4:バス経路座標列を使用 5:Network に指定した座標列を使用
	10	Network	経路座標列	Char	△	経路座標列の経緯度を" "区切りで指定 NetworkOption が5の場合は必須 例 )network=139.111,39.1111 139.222,39.2222 ...
	11	SubPointOption	補間点有無	Integer	—	1:有り、2:無し。 デフォルトは 1
戻り値	<ul style="list-style-type: none"> <li>・正常終了時：GeoJSON 形式のテキスト、異常終了時：エラー番号</li> <li>・経路ネットワーク(検索経路・指定経路)に沿って、指定した時間間隔に内挿された時空間情報</li> <li>・戻り値の詳細は</li> </ul>					
解説	<p>○経路区間内では等速で移動したものと見なす。 ⇒経路途中の交通手段等は考慮せず、指定された起終点時間に沿って内挿する</p> <p>○NetworkOption の値と起終点座標・ネットワークデータの関係</p> <p>■NetworkOption = 1 の場合 引数の起終点座標から GetMixedRoute で得られる道路+鉄道経路の座標列に沿って内挿する。 このとき Network 引数は無視される。</p> <p>■NetworkOption = 2 の場合 引数の起終点座標から GetRoadRoute で得られる道路経路の座標列に沿って内挿する。 このとき Network 引数は無視される。 経路探索は距離優先で行われる。</p> <p>■NetworkOption = 3 の場合 引数の起終点座標から GetRailRoute で得られる鉄道経路の座標列に沿って内挿する。 このとき Network 引数は無視される。</p> <p>■NetworkOption = 4 の場合</p>					

	<p>引数の起終点座標から GetBusRoute で得られるバス経路の座標列に沿って内挿する。 このとき Network 引数は無視される。</p> <p>■NetworkOption = 5 の場合</p> <p>Network={起点座標(必須), 経路座標列(必須), 終点座標(必須)} の座標列に対して内挿する。</p> <p>座標列全体で、少なくとも座標が 3 点なければならない(Network の指定が必須)。</p>
--	--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

### 1.9.1. GetSTInterpolatedPoints の戻り値詳細

項目名	内容	型	備考
status	ステータス	整数	正常終了：1
features	図形の配列	配列	
type	Feature	文字	
properties	属性	オブジェクト	
id	座標 No	整数	座標 No
pass_time	通過日時	文字	yyyy-mm-dd HH:MM:SS
geometry	図形	オブジェクト	
type	Point	文字	
coordinates	座標	座標	世界測地系緯度経度

### 1.9.2. GetSTInterpolatedPoints のエラーコード

エラー番号	内容
20999	システムエラー
20998	そのほかのエラー
20000	座標単位指定エラー
20001	時間解像度書式エラー
20002	時間解像度指定値エラー
20101	起点日時未指定エラー
20102	起点日時書式エラー
20103	終点日時未指定エラー
20104	終点日時書式エラー
20199	起終点日時順のエラー
20301	起点座標未指定エラー
20302	起点座標書式エラー
20303	起点座標指定値エラー
20304	終点座標未指定エラー

20305	終点座標書式エラー
20306	終点座標指定値エラー
20401	ネットワークデータがない（または足りない）
20402	ネットワークデータ書式エラー
20403	ネットワークオプションエラー
20404	補間点オプションエラー

## 1.10. コード表

### 1.10.1. 交通手段コード一覧

コード	内容
1	徒歩
2	自転車
3	原動機付自転車
4	自動二輪車タクシー
5	タクシー
6	乗用車
7	軽乗用車
8	貨物自動車
9	自家用バス路線バス
10	路線バス

表 1.10.1. 交通手段コード

### 1.10.2. 道路種別コード一覧

コード	内容
1	高速自動車国道
2	都市高速道路（含指定都市高速道路）
3	一般国道
4	主要地方道（都道府県道）
5	主要地方道（指定市道）
6	一般都道府県道
7	指定市の一般市道
9	その他道路
0	未調査

表 1.10.2. 道路種別コード

## 1.11. システムエラーコード

すべての API に共通するエラーコード

エラー番号	内容
10001	セッション ID が未セット
10011	セッション ID が不正または存在しない

## 2. 過去 API（動線解析）からの変更点

### 2.1. 道路の経路探索

- 一方通行が考慮されていなかったため道路を逆走するケースがあったが、リンクの通行規制種別コードを使用して進行方向を考慮するようにした。
- 道路種別を指定した場合、該当するリンクのコストを小さくしすぎていた（通常の 1/1000）ため経路探索が正常に終了しないことがあったが、コストを 1/2 にすることで正常終了するようにした。なお、コストを割る数は設定ファイルの「指定条件一致リンクのコスト算出スケール」で変更可能。

### 2.2. GetNearestRoadPoint

- 路線番号指定時に、指定した路線とは異なる路線上の座標が返されることがあったが、指定した路線上の座標が返るようにした。

### 2.3. GetRailRoute

- 出力経路の最初の行（起点駅に該当する行）の路線名は空で出力するようにした。