

NCM Web Api リファレンス

Ver 1.1

平成 29 年 8 月 15 日

(株) エヌ・シー・エム

目次

1.	インターフェイス定義.....	1
1.1.	eREXWebApi	1
1.1.1.	最適ルート探索 (TSP 込み)	1
1.1.2.	最適ルート探索 (マルチゴール)	3
1.1.3.	到達域ポリゴン取得	5
1.1.4.	探索結果情報	6
1.1.5.	パフォーマンスに関する注意点	8
1.2.	eCapGCWebApi	9
1.2.1.	文字列からのジオコード検索	9
1.2.2.	最近隣住所検索	10
1.2.3.	文字列からの駅情報検索	11
1.2.4.	最近隣駅情報検索	11
1.2.5.	検索結果情報	12

1. インターフェイス定義

1.1. eREXWebApi

1.1.1. 最適ルート探索 (TSP 込み)

最適ルート探索 (TSP 込み) の詳細を表 1.1-1 に示す。

表 1.1-1 最適ルート探索 (TSP 込み)

インターフェイス名	最適ルート探索 (TSP 込み)			
URL	http://asp.ncm-git.co.jp/eREXWebApi/route/result/			
説明	<ul style="list-style-type: none"> 指定された地点間の最適ルート探索を実行する。 出発地、経由地(複数指定可能)、目的地の地点情報を設定して実行可能である。 経由地経由順序の最適化(TSP)が可能である。 探索で使用する道路ネットワークデータの種別を指定可能である。 探索結果ルート形状情報の取得可否を指定可能である。 			
引数	名称	型	必須	説明
	datakind	Int2		道路ネットワークデータの種別 1 : 数値地図 25000(空間データ基盤) ※現在は 1 種類のみ。 既定値 1
	startdate	Double		出発日時(yyyyMMddHHmmss 形式) 例:2008年10月27日10時30分30秒の場合 → 20081027103030 既定値 システム日時
	spotname	String		地点名 地点名はカンマ区切りで入力する。 既定値 空文字
	spotposition	String	●	地点位置 地点数分カンマ区切りで入力する。 【位置情報が緯度経度の場合】 緯度、経度(秒単位)の順に「半角スペース」区切りで入力する。 【位置情報が住所の場合】 住所を入力する。 既定値 なし
	arrivallimit	String		経由地点到着時刻 経由地点到着制限としての開始、終了時刻を入力する。 地点毎の設定値はカンマ区切りで入力し、開始、終了時刻は「-」区切りで入力する。 (日時は yyyyMMddHHmmss 形式) 既定値 0-99999999999999(常時)
	staytime	Int2		経由地点滞在時間 経由地点毎の滞在時間(分)をカンマ区切りで入力する。 既定値 0

	side	Int2	地点配置フラグ 地点毎の地点配置フラグを入力する。 「地点配置フラグ」は以下の通り。 0 : 制限なし 1 : リンクの左側のみ(リンク上を含む) 2 : リンクの右側のみ(リンク上を含む)								
			既定値 0								
	speed	Double	全道路一定走行速度[km/h] 0より大きく500以下の値を設定可能。								
			既定値 30								
	turnlevel	Int2	右左折回避レベル 0 : 全く気にしない 1 : 多少気にする 2 : できるだけ避けたい								
			既定値 0								
	tsp	Int2	経路順路最適化フラグ 0 : 最適化しない 1 : 最適化する								
			既定値 0								
	shape	Int2	形状取得フラグ 0 : 取得しない 1 : 取得する								
			既定値 0								
戻り値	「1.1.4 探索結果情報」を参照										
エラー発生時	<fail>タグにエラーメッセージを設定して返す。										
備考	<ul style="list-style-type: none"> spotname, spotposition, side は下記のように設定する。 <ul style="list-style-type: none"> 【経路地なしの場合】 [出発地点],[目的地点] 【経路地ありの場合】 [出発地点],[経路地点 1],[経路地点 2],[目的地点] arrivallimit, staytime は下記のように設定する。 [経路地点 1],[経路地点 2] <p>【設定例】</p> <table border="1"> <tr> <td>地点名</td> <td>spotname=自社,●●工場,...</td> </tr> <tr> <td>地点位置</td> <td>spotposition=東京都文京区白山 1-28-10,126900 487980,...</td> </tr> <tr> <td>地点到着時刻</td> <td>arrivallimit=20080401100000-20080401110000,...</td> </tr> <tr> <td>地点滞在時間</td> <td>staytime=15,5,...</td> </tr> </table> <ul style="list-style-type: none"> 地点位置情報の地点数を正とし、各地点位置に対応する他の地点関連情報が存在しない場合は、既定値を用いて処理を行う。 経路地点の個数は、5個まで設定可能とする。 			地点名	spotname=自社,●●工場,...	地点位置	spotposition=東京都文京区白山 1-28-10,126900 487980,...	地点到着時刻	arrivallimit=20080401100000-20080401110000,...	地点滞在時間	staytime=15,5,...
地点名	spotname=自社,●●工場,...										
地点位置	spotposition=東京都文京区白山 1-28-10,126900 487980,...										
地点到着時刻	arrivallimit=20080401100000-20080401110000,...										
地点滞在時間	staytime=15,5,...										

1.1.2. 最適ルート探索 (マルチゴール)

最適ルート探索 (マルチゴール) の詳細を表 1.1-2 に示す。

表 1.1-2 最適ルート探索 (マルチゴール)

インターフェイス名	最適ルート探索 (マルチゴール)			
URL	http://asp.ncm-git.co.jp/eREXWebApi/multiroute/result/			
説明	<ul style="list-style-type: none"> 指定された出発地と全ての目的地の間の最適ルートを一括探索する。 出発地、目的地(複数指定可能)の地点情報を設定して実行可能である。 探索で使用する道路ネットワークデータの種別を指定可能である。 探索結果ルート形状情報の取得可否を指定可能である。 			
引数	名称	型	必須	説明
	datakind	Int2		道路ネットワークデータの種別 1: 数値地図 25000 (空間データ基盤) ※現在は 1 種類のみ。 既定値 1
	startdate	Double		出発日時 (yyyyMMddHHmmss 形式) 例: 2008 年 10 月 27 日 10 時 30 分 30 秒の場合 → 20081027103030 既定値 システム日時
	spotposition	String	●	地点位置 地点数分カンマ区切りで入力する。 【位置情報が緯度経度の場合】 緯度、経度(秒単位)の順に「半角スペース」区切りで入力する。 【位置情報が住所の場合】 住所を入力する。 既定値 なし
	side	Int2		地点配置フラグ 地点毎の地点配置フラグを入力する。 「地点配置フラグ」は以下の通り。 0 : 制限なし 1 : リnkの左側のみ (リンク上を含む) 2 : リnkの右側のみ (リンク上を含む) 既定値 0
	speed	Double		全道路一定走行速度[km/h] 0 より大きく 500 以下の値を設定可能。 既定値 30
	turnlevel	Int2		右左折回避レベル 0 : 全く気にしない 1 : 多少気にする 2 : できるだけ避けたい 既定値 0
	shape	Int2		形状取得フラグ 0 : 取得しない 1 : 取得する 既定値 0
戻り値	「1.1.4 探索結果情報」を参照			
エラー発生時	<fail>タグにエラーメッセージを設定して返す。			

備考	<ul style="list-style-type: none"> spotname, spotposition, side は下記のように設定する。 [出発地点],[目的地点 1],[目的地点 2],[目的地点 3] 	
	<p>【設定例】</p> <table border="1" style="width: 100%;"> <tr> <td style="width: 30%;">地点位置</td> <td>spotposition=東京都文京区白山 1-28-10,126900 487980,...</td> </tr> </table> <ul style="list-style-type: none"> 地点位置情報の地点数を正とし、各地点位置に対応する他の地点関連情報が存在しない場合は、既定値を用いて処理を行う。 目的地点の個数は、5 個まで設定可能とする。 	地点位置
地点位置	spotposition=東京都文京区白山 1-28-10,126900 487980,...	

1.1.3. 到達域ポリゴン取得

到達域ポリゴン取得の詳細を表 1.1-3 に示す。

表 1.1-3 到達域ポリゴン取得

インターフェイス名	到達域ポリゴン取得			
URL	http://asp.ncm-git.co.jp/eREXWebApi/reachablezone/result/			
説明	<ul style="list-style-type: none"> 指定された地点間から終了条件を満たす到達域解析を実行する。 終了条件として、到達距離、到達時間を指定可能である。 探索で使用する道路ネットワークデータの種別を指定可能である。 到達域を表すポリゴン形状情報を取得する。 			
引数	名称	型	必須	説明
	datakind	Int2		道路ネットワークデータの種別 1 : 数値地図 25000 (空間データ基盤) ※現在は 1 種類のみ。
				既定値
	startdate	Double		出発日時 (yyyyMMddHHmmss 形式) 例: 2008 年 10 月 27 日 10 時 30 分 30 秒の場合 → 20081027103030
				既定値
	spotposition	String	●	地点位置 【位置情報が緯度経度の場合】 緯度、経度 (秒単位) の順に「半角スペース」区切りで入力する。 【位置情報が住所の場合】 住所を入力する。
				既定値
	speed	Double		全道路一定走行速度 [km/h] 0 より大きく 500 以下の値を設定可能。
				既定値
	turnlevel	Int2		右左折回避レベル 0 : 全く気にしない 1 : 多少気にする 2 : できるだけ避けたい
既定値				0
distancelimit	Double	▲*1	到達域距離終了条件 (km) 2km まで設定可能。	
			既定値	0
timelimit	Double	▲*1	到達域距離終了条件 (分) 距離換算 (speed より) 値 2km まで設定可能。	
			既定値	0
stopruletype	Int2		条件連結方法 1 : OR (終了条件の何れか一方を満たすまで) 2 : AND (全ての終了条件を満たすまで)	
			既定値	1
戻り値	「1.1.4 探索結果情報」を参照			
エラー発生時	<fail>タグにエラーメッセージを設定して返す。			
備考	*1 「到達域距離終了条件」と「到達域距離終了条件」の何れか一方には 0 より大きな値を設定しなければならない。			

1.1.4. 探索結果情報

探索結果情報は下記の XML フォーマットで出力されます。

文字エンコーディング	UTF-8
------------	-------

項番	エレメント名	名称	タイプ	型	個数	説明		
1	eREXResult	eREX 探索結果	element	—	1	ルートエレメント		
2	route	ルート情報	element	—	0..	複数の可能性あり(地点が3点以上の場合) ※ 最適ルート探索(TSP 込み)、最適ルート探索(マルチゴール)の場合に出力される。		
3		startname	出発地名	attribute	string		出発地の名称(地名が設定されていない場合は、設定順序番号が設定される。)	
4		goalname	目的地名	attribute	string		目的地の名称(地名が設定されていない場合は、設定順序番号が設定される。)	
5		startdate	出発日時	attribute	double		yyyyMMddHHmmss 形式	
6		distance	走行距離	attribute	double		当該ルートの走行距離(m)	
7		time	走行時間	attribute	double		当該ルートの走行時間(秒)	
8		shape	ルート形状	element	—	1	1つのルート情報に1つのみ存在する。	
9			xy	xy座標	element	string	1..	緯度、経度の順に「半角空白文字」区切りで設定される(度単位)。 座標数分のデータ数。
10		reachablezone	到達域情報	element	—	0..	※ 到達域ポリゴン取得の場合に出力される。	
11			startdate	出発日時	attribute	double		yyyyMMddHHmmss 形式
12			distance	走行距離	attribute	double		当該ルートの走行距離(m)
13			time	走行時間	attribute	double		当該ルートの走行時間(秒)
14			shape	ポリゴン形状	element	—	1	1つの到達域情報に1つのみ存在する。
15			xy	xy座標	element	string	1..	緯度、経度の順に「半角空白文字」区切りで設定される(秒単位)。 座標数分のデータ数。
16		fail	エラー情報	element	string	0.1	エラーが発生した時のみ出力される。	

サンプル

```
<?xml version="1.0" encoding="utf-8" ?>
<eREXResult>
  <route startname="自社" goalname="A工場" startdate="20080612100000" distance="9910.50560791016" time="35677.8201884766">
    <shape>
      <xy>133939.217736 504107.40168</xy>
      <xy>133939.35 504107.55</xy>
      ...
      <xy>133815.33 504276.63012</xy>
      <xy>133815.219948 504276.91452</xy>
    </shape>
  </route>
  <reachablezone startdate="20080612100000" distance="9910.50560791016" time="35677.8201884766">
    <shape>
      <xy>133939.217736 504107.40168</xy>
      <xy>133939.35 504107.55</xy>
      ...
      <xy>133815.33 504276.63012</xy>
      <xy>133815.219948 504276.91452</xy>
    </shape>
  </reachablezone>
  <fail>ルート探索に失敗しました。</fail>
</eREXResult>
```

1.1.5. パフォーマンスに関する注意点

無償サービスの場合、道路ネットワークデータとして数値地図 25000 (空間データ基盤) を利用します。

ここで困ったことに、この道路ネットワークデータは、全国に渡って信頼性の保証が可能な属性データを持っておりません。それゆえ、多くの最適ルート探索エンジンが処理の高速化のために行なっている、道路ネットワークの階層的探索（例えば、東京から大阪に行く時は、高速道路を利用するであろうと言う仮定の下に高速道路のみを探索する）が出来ません。

そう言う意味では、特に道路の本数が多い地域でちょっとでも距離的に長い探索を行なうと、レスポンスは劣化します。しかし、これは道路ネットワークデータに起因するものあり、eREX の性能が低いためではありません。

市販の道路ネットワークデータを利用すれば、例えば北海道の東北端と鹿児島島の南端を結ぶルートであっても、eREX は2秒以下で答えを返すことが可能です。

1.2. eCapGCWebApi

1.2.1. 文字列からのジオコード検索

文字列からのジオコード検索の詳細を表 1.1-1 に示す。

表 1.2-1 文字列からのジオコード検索

インターフェイス名	文字列からのジオコード検索			
URL	http://asp.ncm-git.co.jp/eCapGCWebApi/keyword/geocode/			
説明	・ 指定されたキーワード文字列からジオコード情報を取得する。			
引数	名称	型	必須	説明
	keyword	String	●	検索のキーワードとなる住所文字列。 既定値 なし
戻り値	「1.2.5 検索結果情報」を参照			
エラー発生時	<fail>タグにエラーメッセージを設定して返す。			
備考				

1.2.2. 最近隣住所検索

最近隣住所検索の詳細を表 1.1-2 に示す。

表 1.2-2 最近隣住所検索

インターフェイス名	最近隣住所検索			
URL	http://asp.ncm-git.co.jp/eCapGCWebApi/point/geocode/			
説明	<ul style="list-style-type: none"> 指定された地点周辺のジオコード情報を取得する。 検索範囲を半径で指定する。 取得対象のジオコードレベルを指定可能である。 			
引数	名称	型	必須	説明
	x	Double	●	検索地点の緯度(秒単位) 既定値 なし
	y	Double	●	検索地点の経度(秒単位) 既定値 なし
	rangelimit	Double	●	検索範囲(円)の半径(単位m) 値の範囲: $0 < \text{rangelimit} \leq 200$ 既定値 なし
	matchinglevel	Int2	●	検索対象ジオコードレベル 0 : 全てのレベル 1 : レベル 01(都道府県名) 2 : レベル 02(郡名) 3 : レベル 03(市町村名) 4 : レベル 04(区名) 5 : レベル 05(大字・町域名) 6 : レベル 06(小字名) 7 : レベル 07(丁目) 8 : レベル 08(街区番号) 9 : レベル 09(住居番号) 10 : レベル 10(建物名称) 11 : レベル 11(階数・号室) 既定値 なし
	countlimit	Int2		検索結果個数の上限値 値の範囲: $0 < \text{countlimit} \leq 10$ 既定値 1
戻り値	「1.2.5 検索結果情報」を参照			
エラー発生時	<fail>タグにエラーメッセージを設定して返す。			
備考				

1.2.3. 文字列からの駅情報検索

文字列からの駅情報検索の詳細を表 1.2-3 に示す。

表 1.2-3 文字列からの駅情報検索

インターフェイス名	文字列からの駅情報検索			
URL	http://asp.ncm-git.co.jp/eCapGCWebApi/keyword/stationcode/			
説明	<ul style="list-style-type: none"> 指定されたキーワード文字列から駅情報を取得する。 			
引数	名称	型	必須	説明
	keyword	String	●	検索のキーワードとなる文字列。 既定値 なし
	matchtype	Int2		検索の種類 1 : 完全一致 2 : 部分一致 3 : 前方一致
				既定値 2
戻り値	「1.2.5 検索結果情報」を参照			
エラー発生時	<fail>タグにエラーメッセージを設定して返す。			
備考				

1.2.4. 最近隣駅情報検索

最近隣駅情報検索の詳細を表 1.2-4 に示す。

表 1.2-4 最近隣駅情報検索

インターフェイス名	最近隣駅情報検索			
URL	http://asp.ncm-git.co.jp/eCapGCWebApi/point/stationcode/			
説明	<ul style="list-style-type: none"> 指定された地点周辺の駅情報を取得する。 検索範囲を半径で指定する。 検索結果個数制限値を指定可能である。 			
引数	名称	型	必須	説明
	x	Double	●	検索地点の緯度(秒単位) 既定値 なし
	y	Double	●	検索地点の経度(秒単位) 既定値 なし
				既定値 なし
	rangelimt	Double	●	検索範囲(円)の半径(単位m) 値の範囲: $0 < rangelimt \leq 200$
				既定値 なし
	countlimit	Int2		検索結果個数の上限値 値の範囲: $0 < countlimit \leq 10$
				既定値 1

1.2.5. 検索結果情報

検索結果情報は下記の XML フォーマットで出力されます。

文字エンコーディング	UTF-8
------------	-------

項番	エレメント名	名称	タイプ	型	個数	説明
1	eCapGCResult	ジオコーディング検索結果	element	—	1	ルートエレメント
2	version	バージョン	attribute	string		eCapGC のバージョン情報
3	coordinateType	座標のタイプ	attribute	int2		座標のタイプは以下の通り。 0 : 未定義 1 : BL(度単位) 2 : UTM(m単位) 3 : TM(m単位)
4	coordinateZone	ゾーン番号	attribute	int2		
5	datum	測地系の種類	attribute	int2		測地系の種類は以下の通り。 0 : 未定義 1 : 旧日本測地系 2 : JGD2000、世界測地系
6	geoCode	ジオコード情報	element	—	1..	
7	matchingRatio	マッチング率	attribute	double		値の範囲は、0-100
8	matchingLevel	マッチングレベル	attribute	int2		
9	xy	位置座標	element	string	1	緯度、経度の順に「半角空白文字」区切りで設定される。 ※ 単位は秒
10	zipCode	郵便番号	element	string	1	「-」なしの7桁の番号が設定される。
11	authorizedName	構造化住所文字列	element	string	1..	index:0 = 都道府県名
12	index	インデックス	attribute	int2		index:1 = 郡名 ...
13	authorizedYomi	構造化住所読み仮名文字列	element	string	1..	index:0 = 都道府県名読み仮名 index:1 = 郡名読み仮名

14		index	インデックス	attribute	int2		...
15	stationCode		駅情報	element	—	1..	
16		stationName	駅名称	element	string	1	
17		stationYomi	駅名読み仮名	element	string	1	全角ひらがな
18		corpName	運営組織名称	element	string	1	
19		lineName	鉄道路線名称	element	string	1	
20		xy	位置座標	element	string	1	緯度、経度の順に「半角空白文字」区切りで設定される。 ※ 単位は秒
21	fail		エラー情報	element	string	0.1	エラーが発生した時のみ出力される。

サンプル

```
<?xml version="1.0" encoding="utf-8" ?>
<eCapGCRresult version="1.1.0" coordinateType="1" coordinateZone="0" datum="2">
  <geoCode matchingRatio="100" matchingLevel="5">
    <xy>128590.848073469 503097.541622449</xy>
    <zipCode>1120000</zipCode>
    <authorizedName index="0">東京都</authorizedName>
    <authorizedName index="1" />
    <authorizedName index="2" />
    <authorizedName index="3">文京区</authorizedName>
    <authorizedName index="4">白山</authorizedName>
    <authorizedYomi index="0">トウキョウト</authorizedYomi>
    <authorizedYomi index="1" />
    <authorizedYomi index="2" />
    <authorizedYomi index="3">ブンキョウク</authorizedYomi>
    <authorizedYomi index="4">ハクサン</authorizedYomi>
  </geoCode>
  <stationCode>
    <stationName>東京</stationName>
    <stationYomi>とうきょう</stationYomi>
  </stationCode>
</eCapGCRresult>
```

```
<corpName>JR</corpName>  
<lineName>東海道新幹線(東京～新大阪)</lineName>  
<xy>128452.254585129 503162.358617167</xy>  
</geoCode>  
<fail>入力パラメータの取得に失敗しました。</fail>  
</eCapGCRresult>
```