

# Web services de l'offre « temps réel » des transports collectifs de Brest métropole (version bêta)

## I. Contexte

La délibération n° B 2010-03-037 du 12 mars 2010 du bureau de communauté proposait de placer la base de données cartographique de la métropole et son contenu en libre-accès.

Cette disposition permet de faciliter la réutilisation de ces données publiques par les citoyens et les acteurs économiques.

Un certain nombre de flux ouverts, orientés système d'information géographique, ont déjà été mis en place par Brest métropole sur le portail d'information géographique du pays de Brest <https://geo.pays-de-brest.fr/donnees/Pages/default.aspx>.

## II. Objectif

En complément de la publication de l'offre théorique (au format GTFS), ce document présente la description technique des données temps réel des bus et des tramways de Brest métropole, qui sont fournies à partir de Web Services.

## III. Versions de l'API

Version	Date de publication	Description
1.3	27/10/2016	Corrections des fonctions <code>getStopsNear()</code> et <code>getStopNear_route()</code>
Beta 1.2	26/10/2015	<p>Ajout des fonctions :</p> <ul style="list-style-type: none"> <li>- <b>getStop2</b></li> <li>- <b>getRoutes_stop2</b></li> <li>- <b>getNextDepartures2</b></li> <li>- <b>getRemainingTimes2</b></li> </ul> <p>qui travaillent avec un <code>stop_id</code> à la place du <code>stop_name</code>.</p> <p>Fonction <b>getDeviations</b> : ajout des <code>stop_id</code> dans le résultat.</p>
Beta 1.1	09/09/2015	<p>Mise à jour des exemples suite à la réorganisation du réseau de transport au 1<sup>er</sup> septembre,</p> <p>Fonctions <b>getNextDepartures</b> et <b>getRemainingTimes</b> : correction d'une faute de frappe, <code>arrvial_time</code> devient <code>arrival_time</code>,</p> <p>Fonctions <b>getStopsNear</b> et <b>getStopNear_route</b>: renommage des paramètres X,Y en longitude, latitude.</p>

Beta 1.0	11/06/2015	Mise en ligne.

## IV. Web services

### 1. Accès aux Web Services

Les Web Services fonctionnent en mode SOAP et REST.

- Description WSDL :  
<https://applications002.brest-metropole.fr/WIPOD01/Transport.svc?singleWsd>

- Accès REST :

Page d'aide <https://applications002.brest-metropole.fr/WIPOD01/Transport/REST/help>  
Deux formats sont proposés en sortie : JSON ou XML (casse non prise en compte).

- Accès SOAP : <https://applications002.brest-metropole.fr/WIPOD01/Transport.svc/SOAP>

La connexion est sécurisée (**https**), elle accepte uniquement les protocoles suivants :

- TLS 1.0
- TLS 1.1
- TLS 1.2

et les ciphers suivants :

- Triple DES 168
- AES 128/128
- AES 256/256

### 2. Les fonctions de l'API

La casse et les accents des paramètres d'entrée ne sont pas pris en compte.  
(head\_trip= « oceanopolis » est identique à head\_trip= « oceAnopoliS »)

En accès SOAP, les paramètres d'entrée sous forme de chiffres ne doivent pas comporter de virgule mais des points (-4,45545345 mais -4.45545345).

Les codes d'arrêts, de lignes et de directions sont strictement identiques à ceux présents dans le lot des données théoriques GTFS du site :

<https://geo.pays-de-brest.fr/donnees/Pages/TempsReel.aspx>

### 3. Les données retournées par l'API

- Les données retournées par l'API proviennent d'une copie de la base de données du système temps réel qui gèrent les bus et les tramways. Dans le pire des cas, les données fournies par L'API ont une minute de retard.
- L'API n'indique pas si le code d'arrêt, de ligne ou de direction est invalide. Une réponse "vide" signifie probablement qu'il n'y a pas de résultat pour votre recherche, ou que les données d'entrées sont invalides.
- L'API ne vous donnera que les départs prévus dans l'heure.



## 4. Sommaire des Web Services disponibles

<b>Données statiques</b> .....	5
getVersion() : obtenir la version de l'application.....	5
getStopsNames() : obtenir le nom de tous les arrêts .....	6
getRoutes() : obtenir toutes les lignes .....	8
getStops_route () : obtenir les arrêts d'une ligne .....	10
getStop () : obtenir un arrêt .....	12
getStop2 () : obtenir un arrêt .....	14
getRoutes_stop() : obtenir toutes les lignes desservant un arrêt .....	16
getRoutes_stop2() : obtenir toutes les lignes desservant un arrêt .....	18
getDestinations() : obtenir les noms des destinations .....	20
<b>Données dynamiques et temps réel</b> .....	23
getStopsNear () : obtenir les arrêts les plus proches .....	23
getStopNear_route() : obtenir l'arrêt le plus proche pour une ligne .....	25
getStopVehiclesPosition() : obtenir la position (arrêt) des véhicules (bus ou tramway).....	27
getGeolocatedVehiclesPosition() : obtenir la position (géographique) des véhicules.....	29
getNextDepartures() : obtenir les horaires des prochains départs .....	32
getNextDepartures2() : obtenir les horaires des prochains départs .....	35
getRemainingTimes () : obtenir les temps d'attente avant les prochains passages .....	38
getRemainingTimes2 () : obtenir les temps d'attente avant les prochains passages .....	41
getDeviations() : obtenir les déviations .....	44
getPerturbations() : obtenir les perturbations.....	46

# Données statiques

---

## getVersion() : obtenir la version de l'application

Paramètres d'entrée
Aucun

Paramètres de sortie				
Name	Type	Description	Attributs	Type
Version	Version	Retourne la version de l'application	Date	String
			Number	String

### Informations sur les données de sortie :

La fonction retourne une liste de versions (Version) de l'application avec pour attributs :

- Date : date du changement de version de l'application
- Number : numéro de la version en cours

### Syntaxe REST

<https://applications002.brest-metropole.fr/WIPOD01/Transport/REST/getVersion?format={format}>

### Exemples d'interrogations:

Requête REST :

<https://applications002.brest-metropole.fr/WIPOD01/Transport/REST/getVersion?format=json>

Résultat

```
{
  "Date": "27/10/2016",
  "Number": "1.3"
}
```

Requête SOAP :

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:tem="http://tempuri.org/">
  <soapenv:Header/>
```

```

    <soapenv:Body>
      <tem:getVersion>
        <!--Optional:-->
        <tem:format?></tem:format>
      </tem:getVersion>
    </soapenv:Body>
  </soapenv:Envelope>

```

### Résultat

```

<s:Envelope xmlns:s="http://schemas.xmlsoap.org/soap/envelope/">
  <s:Body>
    <getVersionResponse xmlns="http://tempuri.org/">
      <getVersionResult
xmlns:a="http://schemas.datacontract.org/2004/07/Dit.od"
xmlns:i="http://www.w3.org/2001/XMLSchema-instance">
        <a:Date>27/10/2016</a:Date>
        <a:Number>1.3</a:Number>
      </getVersionResult>
    </getVersionResponse>
  </s:Body>
</s:Envelope>

```

## getStopsNames() : obtenir le nom de tous les arrêts

Paramètres d'entrée		
Nom	Type	Description
format	String	XML/JSON (XML par défaut)

Paramètres de sortie				
Name	Type	Description	Attributs	Type
StopName[]	StopName enumeration	Retourne le nom de tous les arrêts du réseau de transport de Brest Métropole	Stop_name	String

### Informations sur les données de sortie :

La fonction retourne une liste de noms d'arrêts (StopName) avec pour attributs :

- Stop\_name : nom commercial de la station (une station comprend communément 2 arrêts : un arrêt dans le sens aller et un arrêt dans le sens retour – sauf exceptions)

### Syntaxe REST

<https://applications002.brest-metropole.fr/WIPOD01/Transport/REST/getStopsNames?format={format}>

## Exemples d'interrogations

Requête REST :

<https://applications002.brest-metropole.fr/WIPOD01/Transport/REST/getStopsNames?format=json>

Résultat

```
[ {"Stop_name": "4 Chemins"},
  {"Stop_name": "4 Moulins"},
  {"Stop_name": "8 mai 1945"},
  {"Stop_name": "A.France"},
  {"Stop_name": "Aber-Ildut"},
```

Requête SOAP :

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:tem="http://tempuri.org/">
  <soapenv:Header/>
  <soapenv:Body>
    <tem:getStopsNames>
      <!--Optional:-->
      <tem:format>xml</tem:format>
    </tem:getStopsNames>
  </soapenv:Body>
</soapenv:Envelope>
```

Résultat

```
<s:Envelope xmlns:s="http://schemas.xmlsoap.org/soap/envelope/">
  <s:Body>
    <getStopsNamesResponse xmlns="http://tempuri.org/">
      <getStopsNamesResult
xmlns:a="http://schemas.datacontract.org/2004/07/Dit.od"
xmlns:i="http://www.w3.org/2001/XMLSchema-instance">
        <a:StopName>
          <a:Stop_name>4 Chemins</a:Stop_name>
        </a:StopName>
        <a:StopName>
          <a:Stop_name>4 Moulins</a:Stop_name>
        </a:StopName>
        <a:StopName>
          <a:Stop_name>8 mai 1945</a:Stop_name>
        </a:StopName>
        <a:StopName>
          <a:Stop_name>A.France</a:Stop_name>
        </a:StopName>
        <a:StopName>
          <a:Stop_name>Aber-Ildut</a:Stop_name>
        </a:StopName>
        <a:StopName>
          <a:Stop_name>Aéroport</a:Stop_name>
        </a:StopName>
      </getStopsNamesResult>
    </getStopsNamesResponse>
  </s:Body>
</s:Envelope>
```

## getRoutes() : obtenir toutes les lignes

Paramètres d'entrée		
Nom	Type	Description
format	String	XML/JSON (XML par défaut)

Paramètres de sortie				
Name	Type	Description	Attributs	Type
Route[]	Route enumeration	Retourne les lignes du réseau de transport de Brest Métropole	Route_id	String
			Route_long_name	String

### Informations sur les données de sortie :

La fonction retourne une liste de lignes (Route) avec pour attributs :

- Route\_id : numéro de la ligne permettant de l'identifier
- Route\_long\_name : nom de la ligne

### Syntaxe REST

<https://applications002.brest-metropole.fr/WIPOD01/Transport/REST/getRoutes?format={format}>

### Exemples d'interrogations

Requête REST :

<https://applications002.brest-metropole.fr/WIPOD01/Transport/REST/getRoutes?format=ison>

### Résultat

```
[ {
  "Route_id": "A",
  "Route_long_name": "Tramway Est Ouest"
},
{
  "Route_id": "1",
  "Route_long_name": "Chru < > Montbarrey"
},
{
  "Route_id": "3",
  "Route_long_name": "Oceano < > Lambe"
},
{
  "Route_id": "4",
  "Route_long_name": "Chr < > Elorn"
},
...
]
```



## Requête SOAP :

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:tem="http://tempuri.org/"
  <soapenv:Header/>
  <soapenv:Body>
    <tem:getRoutes>
      <!--Optional:-->
      <tem:format>xml</tem:format>
    </tem:getRoutes>
  </soapenv:Body>
</soapenv:Envelope>
```

## Résultat

```
<s:Envelope xmlns:s="http://schemas.xmlsoap.org/soap/envelope/">
  <s:Body>
    <getRoutesResponse xmlns="http://tempuri.org/">
      <getRoutesResult
xmlns:a="http://schemas.datacontract.org/2004/07/Dit.od"
xmlns:i="http://www.w3.org/2001/XMLSchema-instance">
        <a:Route>
          <a:Route_id>A</a:Route_id>
          <a:Route_long_name>Tramway Est Ouest</a:Route_long_name>
        </a:Route>
        <a:Route>
          <a:Route_id>1</a:Route_id>
          <a:Route_long_name>Chru &lt; > Montbarrey</a:Route_long_name>
        </a:Route>
        <a:Route>
          <a:Route_id>3</a:Route_id>
          <a:Route_long_name>Océano &lt; > Lambe</a:Route_long_name>
        </a:Route>
        <a:Route>
          <a:Route_id>4</a:Route_id>
          <a:Route_long_name>Chr &lt; > Elorn</a:Route_long_name>
        </a:Route>
      </getRoutesResult>
    </getRoutesResponse>
  </s:Body>
</s:Envelope>
```

## getStops\_route () : obtenir les arrêts d'une ligne

Paramètres d'entrée		
Nom	Type	Description
format	String	XML/JSON (XML par défaut)
route_id	String	Identifiant de la ligne (numéro de ligne)
trip_headsign	String	Nom de la destination

Paramètres de sortie				
Name	Type	Description	Attributs	Type
Stop[]	Stop enumeration	Retourne les arrêts dans l'ordre pour une ligne et une destination	Stop_id	String
			Stop_lat	String
			Stop_lon	String
			Stop_name	String

### Informations sur les données de sortie :

La fonction retourne une liste d'arrêts (Stop) avec pour attributs :

- Stop\_id : nom permettant d'identifier l'arrêt
- Stop\_lat : latitude de l'arrêt en WGS84
- Stop\_lon : longitude de l'arrêt en WGS84
- Stop\_name : nom commercial de la station (une station comprend communément 2 arrêts : un arrêt dans le sens aller et un arrêt dans le sens retour – sauf exceptions)

### Syntaxe REST

[https://applications002.brest-metropole.fr/WIPOD01/Transport/REST/getStops\\_route?format=json&route\\_id={route\\_id}&trip\\_headsign={trip\\_headsign}](https://applications002.brest-metropole.fr/WIPOD01/Transport/REST/getStops_route?format=json&route_id={route_id}&trip_headsign={trip_headsign})

### Exemples d'interrogations

Requête REST :

[https://applications002.brest-metropole.fr/WIPOD01/Transport/REST/getStops\\_route?format=json&route\\_id=A&trip\\_headsign=Porte%20de%20Guipavas](https://applications002.brest-metropole.fr/WIPOD01/Transport/REST/getStops_route?format=json&route_id=A&trip_headsign=Porte%20de%20Guipavas)

Résultat

```
[
  {
    "Stop_id": "PTPLZ_TA",
    "Stop_lat": "48.3762763556729",
    "Stop_lon": "-4.55344850018421",
    "Stop_name": "Porte Plouzane"
  },
  {
    "Stop_id": "FMONT_TA",
```

```

    "Stop_lat": "48.3787259257997",
    "Stop_lon": "-4.5439766947035",
    "Stop_name": "Fort Montbarey"
  },
  {
    "Stop_id": "KROUX_TA",
    "Stop_lat": "48.3823702353052",
    "Stop_lon": "-4.53937620053102",
    "Stop_name": "Keranroux"
  },...

```

## Requête SOAP :

```

<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:tem="http://tempuri.org/"
  <soapenv:Header/>
  <soapenv:Body>
    <tem:getStops_route>
      <!--Optional:-->
      <tem:format>?</tem:format>
      <!--Optional:-->
      <tem:route_id>A</tem:route_id>
      <!--Optional:-->
      <tem:trip_headsign>Porte de Guipavas</tem:trip_headsign>
    </tem:getStops_route>
  </soapenv:Body>
</soapenv:Envelope>

```

## Résultat

```

<s:Envelope xmlns:s="http://schemas.xmlsoap.org/soap/envelope/">
  <s:Body>
    <getStops_routeResponse xmlns="http://tempuri.org/">
      <getStops_routeResult
xmlns:a="http://schemas.datacontract.org/2004/07/Dit.od"
xmlns:i="http://www.w3.org/2001/XMLSchema-instance">
        <a:Stop>
          <a:Stop_id>PTPLZ_TA</a:Stop_id>
          <a:Stop_lat>48.3762763556729</a:Stop_lat>
          <a:Stop_lon>-4.55344850018421</a:Stop_lon>
          <a:Stop_name>Porte Plouzane</a:Stop_name>
        </a:Stop>
        <a:Stop>
          <a:Stop_id>FMONT_TA</a:Stop_id>
          <a:Stop_lat>48.3787259257997</a:Stop_lat>
          <a:Stop_lon>-4.5439766947035</a:Stop_lon>
          <a:Stop_name>Fort Montbarey</a:Stop_name>
        </a:Stop>
        <a:Stop>
          <a:Stop_id>KROUX_TA</a:Stop_id>
          <a:Stop_lat>48.3823702353052</a:Stop_lat>
          <a:Stop_lon>-4.53937620053102</a:Stop_lon>
          <a:Stop_name>Keranroux</a:Stop_name>
        </a:Stop>
      </getStops_routeResult>
    </getStops_routeResponse>
  </s:Body>
</s:Envelope>

```

## getStop () : obtenir un arrêt

Paramètres d'entrée		
Nom	Type	Description
format	String	XML/JSON (XML par défaut)
stop_name	String	Nom d'un arrêt

Paramètres de sortie				
Name	Type	Description	Attributs	Type
Stop[]	Stop enumeration	Retourne les arrêts pour un nom d'arrêt	Stop_lat	String
			Stop_lon	String
			Stop_name	String
			Stop_id	String

### Informations sur les données de sortie :

La fonction retourne une liste d'arrêts (Stop) avec pour attributs :

- Stop\_lat : latitude de l'arrêt en WGS84
- Stop\_lon : longitude de l'arrêt en WGS84
- Stop\_name : nom commercial de la station (une station comprend communément 2 arrêts : un arrêt dans le sens aller et un arrêt dans le sens retour – sauf exceptions)
- Stop\_id : nom permettant d'identifier l'arrêt

### Syntaxe REST

[https://applications002.brest-metropole.fr/WIPOD01/Transport/REST/getSTopformat=json&stop\\_name={stop\\_name}](https://applications002.brest-metropole.fr/WIPOD01/Transport/REST/getSTopformat=json&stop_name={stop_name})

### Exemples d'interrogations

Requête REST :

[https://applications002.brest-metropole.fr/WIPOD01/Transport/REST/getStop?format=json&stop\\_name=malakoff](https://applications002.brest-metropole.fr/WIPOD01/Transport/REST/getStop?format=json&stop_name=malakoff)

Résultat

```
[ {
  "Stop_id": "MALAKO_R",
  "Stop_lat": "48.3965384078681",
  "Stop_lon": "-4.48386413164589",
  "Stop_name": "Malakoff"
}, {
  "Stop_id": "MALAKO_1",
  "Stop_lat": "48.3962545062274",
  "Stop_lon": "-4.48424712892053",
  "Stop_name": "Malakoff"
}, {
  "Stop_id": "MALAKO_2",
  "Stop_lat": "48.395918831208",
  "Stop_lon": "-4.48412196331267",
```

```
    "Stop_name": "Malakoff"  
  }  
]
```

### Requête SOAP :

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"  
  xmlns:tem="http://tempuri.org/">  
  <soapenv:Header/>  
  <soapenv:Body>  
    <tem:getStop>  
      <!--Optional:-->  
      <tem:format?></tem:format>  
      <!--Optional:-->  
      <tem:stop_name>malakoff</tem:stop_name>  
    </tem:getStop>  
  </soapenv:Body>  
</soapenv:Envelope>
```

### Résultat

```
<s:Envelope xmlns:s="http://schemas.xmlsoap.org/soap/envelope/">  
  <s:Body>  
    <getStopResponse xmlns="http://tempuri.org/">  
      <getStopResult xmlns:a="http://schemas.datacontract.org/2004/07/Dit.od"  
xmlns:i="http://www.w3.org/2001/XMLSchema-instance">  
        <a:Stop>  
          <a:Stop_id>MALAKO_R</a:Stop_id>  
          <a:Stop_lat>48.3965384078681</a:Stop_lat>  
          <a:Stop_lon>-4.48386413164589</a:Stop_lon>  
          <a:Stop_name>Malakoff</a:Stop_name>  
        </a:Stop>  
        <a:Stop>  
          <a:Stop_id>MALAKO_1</a:Stop_id>  
          <a:Stop_lat>48.3962545062274</a:Stop_lat>  
          <a:Stop_lon>-4.48424712892053</a:Stop_lon>  
          <a:Stop_name>Malakoff</a:Stop_name>  
        </a:Stop>  
        <a:Stop>  
          <a:Stop_id>MALAKO_2</a:Stop_id>  
          <a:Stop_lat>48.395918831208</a:Stop_lat>  
          <a:Stop_lon>-4.48412196331267</a:Stop_lon>  
          <a:Stop_name>Malakoff</a:Stop_name>  
        </a:Stop>  
      </getStopResult>  
    </getStopResponse>  
  </s:Body>  
</s:Envelope>
```

## getStop2 () : obtenir un arrêt

Paramètres d'entrée		
Nom	Type	Description
format	String	XML/JSON (XML par défaut)
stop_id	String	Id d'un arrêt

Paramètres de sortie				
Name	Type	Description	Attributs	Type
Stop[]	Stop enumeration	Retourne les arrêts pour un nom d'arrêt	Stop_lat	String
			Stop_lon	String
			Stop_name	String
			Stop_id	String

### Informations sur les données de sortie :

La fonction retourne un arrêt (Stop) avec pour attributs :

- Stop\_lat : latitude de l'arrêt en WGS84
- Stop\_lon : longitude de l'arrêt en WGS84
- Stop\_name : nom commercial de la station
- Stop\_id : nom permettant d'identifier l'arrêt

### Syntaxe REST

[https://applications002.brest-metropole.fr/WIPOD01/Transport/REST/getStop2?format=json&stop\\_id={stop\\_id}](https://applications002.brest-metropole.fr/WIPOD01/Transport/REST/getStop2?format=json&stop_id={stop_id})

### Exemples d'interrogations

Requête REST :

[https://applications002.brest-metropole.fr/WIPOD01/Transport/REST/getStop2?format=json&stop\\_id=MALAKO\\_1](https://applications002.brest-metropole.fr/WIPOD01/Transport/REST/getStop2?format=json&stop_id=MALAKO_1)

Résultat

```
[ {
  "Stop_id": "MALAKO_1",
  "Stop_lat": "48.3962545062274",
  "Stop_lon": "-4.48424712892053",
  "Stop_name": "Malakoff"
}
```

## Requête SOAP :

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:tem="http://tempuri.org/">
  <soapenv:Header/>
  <soapenv:Body>
    <tem:getStop>
      <!--Optional:-->
      <tem:format?></tem:format>
      <!--Optional:-->
      <tem:stop_name>malakoff</tem:stop_name>
    </tem:getStop>
  </soapenv:Body>
</soapenv:Envelope>
```

## Résultat

```
<s:Envelope xmlns:s="http://schemas.xmlsoap.org/soap/envelope/">
  <s:Body>
    <getStopResponse xmlns="http://tempuri.org/">
      <getStopResult xmlns:a="http://schemas.datacontract.org/2004/07/Dit.od"
xmlns:i="http://www.w3.org/2001/XMLSchema-instance">
        <a:Stop>
          <a:Stop_id>MALAKO_1</a:Stop_id>
          <a:Stop_lat>48.3962545062274</a:Stop_lat>
          <a:Stop_lon>-4.48424712892053</a:Stop_lon>
          <a:Stop_name>Malakoff</a:Stop_name>
        </a:Stop>
      </getStopResult>
    </getStopResponse>
  </s:Body>
</s:Envelope>
```

## getRoutes\_stop() : obtenir toutes les lignes desservant un arrêt

Paramètres d'entrée		
Nom	Type	Description
format	String	XML/JSON (XML par défaut)
stop_name	String	Nom de l'arrêt

Paramètres de sortie				
Name	Type	Description	Attributs	Type
Route[]	Route enumeration	Retourne les lignes desservant l'arrêt choisi	Route_id	String
			Route_long_name	String

### Informations sur les données de sortie :

La fonction retourne une liste de lignes (Route) avec pour attributs :

- Route\_id : numéro de la ligne permettant de l'identifier
- Route\_long\_name : nom de la ligne

### Syntaxe REST

[https://applications002.brest-metropole.fr/WIPOD01/Transport/REST/getRoutes\\_Stop?format={format}&stop\\_name={stop\\_name}](https://applications002.brest-metropole.fr/WIPOD01/Transport/REST/getRoutes_Stop?format={format}&stop_name={stop_name})

### Exemples d'interrogations :

Requête REST :

[https://applications002.brest-metropole.fr/WIPOD01/Transport/REST/getRoutes\\_Stop?format=json&stop\\_name=malakoff](https://applications002.brest-metropole.fr/WIPOD01/Transport/REST/getRoutes_Stop?format=json&stop_name=malakoff)

Résultat

```
[{
  "Route_id": "8",
  "Route_long_name": "Plougastel  Liberte"
},{
  "Route_id": "7",
  "Route_long_name": "Hopital tromeur - Liberte"
},{
  "Route_id": "3",
  "Route_long_name": "Oceano < > Lambe"
},{
  "Route_id": "1",
  "Route_long_name": "Chru < > Montbarrey"}]
```



## Requête SOAP

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:tem="http://tempuri.org/"
  <soapenv:Header/>
  <soapenv:Body>
    <tem:getRoutes_Stop>
      <!--Optional:-->
      <tem:format>xml</tem:format>
      <!--Optional:-->
      <tem:stop_name>Malakoff</tem:stop_name>
    </tem:getRoutes_Stop>
  </soapenv:Body>
</soapenv:Envelope>
```

## Résultat

```
<s:Envelope xmlns:s="http://schemas.xmlsoap.org/soap/envelope/"
  <s:Body>
    <getRoutes_StopResponse xmlns="http://tempuri.org/"
      <getRoutes_StopResult
xmlns:a="http://schemas.datacontract.org/2004/07/Dit.od"
xmlns:i="http://www.w3.org/2001/XMLSchema-instance">
        <a:Route>
          <a:Route_id>8</a:Route_id>
          <a:Route_long_name>Plougastel  Liberte</a:Route_long_name>
        </a:Route>
        <a:Route>
          <a:Route_id>7</a:Route_id>
          <a:Route_long_name>Hopital tromeur - Liberte</a:Route_long_name>
        </a:Route>
        <a:Route>
          <a:Route_id>3</a:Route_id>
          <a:Route_long_name>Océano &lt; > Lambe</a:Route_long_name>
        </a:Route>
        <a:Route>
          <a:Route_id>1</a:Route_id>
          <a:Route_long_name>Chru &lt; > Montbarrey</a:Route_long_name>
        </a:Route>
      </getRoutes_StopResult>
    </getRoutes_StopResponse>
  </s:Body>
</s:Envelope>
```

## getRoutes\_stop2() : obtenir toutes les lignes desservant un arrêt

Paramètres d'entrée		
Nom	Type	Description
format	String	XML/JSON (XML par défaut)
stop_id	String	Id de l'arrêt

Paramètres de sortie				
Name	Type	Description	Attributs	Type
Route[]	Route enumeration	Retourne les lignes desservant l'arrêt choisi	Route_id	String
			Route_long_name	String

### Informations sur les données de sortie :

La fonction retourne une liste de lignes (Route) avec pour attributs :

- Route\_id : numéro de la ligne permettant de l'identifier
- Route\_long\_name : nom de la ligne

### Syntaxe REST

[https://applications002.brest-metropole.fr/WIPOD01/Transport/REST/getRoutes\\_Stop2?format={format}&stop\\_in={stop\\_id}](https://applications002.brest-metropole.fr/WIPOD01/Transport/REST/getRoutes_Stop2?format={format}&stop_in={stop_id})

### Exemples d'interrogations :

Requête REST :

[https://applications002.brest-metropole.fr/WIPOD01/Transport/REST/getRoutes\\_Stop2?format=json&stop\\_id=MALAKO\\_1](https://applications002.brest-metropole.fr/WIPOD01/Transport/REST/getRoutes_Stop2?format=json&stop_id=MALAKO_1)

### Résultat

```
[{
  "Route_id": "8",
  "Route_long_name": "Plougastel Liberte"
},{
  "Route_id": "7",
  "Route_long_name": "Hopital tromeur - Liberte"
},{
  "Route_id": "3",
  "Route_long_name": "Oceano < > Lambe"
},{
  "Route_id": "1",
  "Route_long_name": "Chru < > Montbarrey"}]
```

## Requête SOAP

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:tem="http://tempuri.org/"
  <soapenv:Header/>
  <soapenv:Body>
    <tem:getRoutes_Stop>
      <!--Optional:-->
      <tem:format>xml</tem:format>
      <!--Optional:-->
      <tem:stop_name>Malakoff</tem:stop_name>
    </tem:getRoutes_Stop>
  </soapenv:Body>
</soapenv:Envelope>
```

## Résultat

```
<s:Envelope xmlns:s="http://schemas.xmlsoap.org/soap/envelope/"
  <s:Body>
    <getRoutes_StopResponse xmlns="http://tempuri.org/"
      <getRoutes_StopResult
xmlns:a="http://schemas.datacontract.org/2004/07/Dit.od"
xmlns:i="http://www.w3.org/2001/XMLSchema-instance">
        <a:Route>
          <a:Route_id>8</a:Route_id>
          <a:Route_long_name>Plougastel  Liberte</a:Route_long_name>
        </a:Route>
        <a:Route>
          <a:Route_id>7</a:Route_id>
          <a:Route_long_name>Hopital tromeur - Liberte</a:Route_long_name>
        </a:Route>
        <a:Route>
          <a:Route_id>3</a:Route_id>
          <a:Route_long_name>Océano &lt; > Lambe</a:Route_long_name>
        </a:Route>
        <a:Route>
          <a:Route_id>1</a:Route_id>
          <a:Route_long_name>Chru &lt; > Montbarrey</a:Route_long_name>
        </a:Route>
      </getRoutes_StopResult>
    </getRoutes_StopResponse>
  </s:Body>
</s:Envelope>
```

## getDestinations() : obtenir les noms des destinations

Paramètres d'entrée		
Nom	Type	Description
format	String	XML/JSON (XML par défaut)
route_id	String	Identifiant de la ligne (numéro de ligne)

Paramètres de sortie				
Name	Type	Description	Attributs	Type
Destination[]	Destination enumeration	Retourne les destinations pour la ligne choisie	Trip_headsign	String

### Informations sur les données de sortie :

La fonction retourne une liste de destinations (Destination) avec pour attributs :

- Trip\_headsign : nom de la destination

Remarque : les destinations d'une ligne correspondent à ses terminus. Une ligne peut avoir plus de deux destinations : en cas de ligne en branches (exemple : ligne de tramway), ou de terminus partiel.

### Syntaxe REST

[https://applications002.brest-metropole.fr/WIPOD01/Transport/REST/getDestinations?format={format}&route\\_id={route\\_id}](https://applications002.brest-metropole.fr/WIPOD01/Transport/REST/getDestinations?format={format}&route_id={route_id})

### Exemples d'interrogations:

Requête REST :

[https://applications002.brest-metropole.fr/WIPOD01/Transport/REST/getDestinations?format=json&route\\_id=A](https://applications002.brest-metropole.fr/WIPOD01/Transport/REST/getDestinations?format=json&route_id=A)

Résultat:

```
[
  {"Trip_headsign": "Porte de Gouesnou"},
  {"Trip_headsign": "Porte de Guipavas"},
  {"Trip_headsign": "Porte de Plouzané"}
]
```

Requête SOAP :

```

<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:tem="http://tempuri.org/">
  <soapenv:Header/>
  <soapenv:Body>
    <tem:getDestinations>
      <!--Optional:-->
      <tem:format?></tem:format>
      <!--Optional:-->
      <tem:route_id>A</tem:route_id>
    </tem:getDestinations>
  </soapenv:Body>
</soapenv:Envelope>

```

**Résultat :**

```

<s:Envelope xmlns:s="http://schemas.xmlsoap.org/soap/envelope/">
  <s:Body>
    <getDestinationsResponse xmlns="http://tempuri.org/">
      <getDestinationsResult
xmlns:a="http://schemas.datacontract.org/2004/07/Dit.od"
xmlns:i="http://www.w3.org/2001/XMLSchema-instance">
        <a:Destination>
          <a:Trip_headsign>Porte de Gouesnou</a:Trip_headsign>
        </a:Destination>
        <a:Destination>
          <a:Trip_headsign>Porte de Guipavas</a:Trip_headsign>
        </a:Destination>
        <a:Destination>
          <a:Trip_headsign>Porte de Plouzané</a:Trip_headsign>
        </a:Destination>
      </getDestinationsResult>
    </getDestinationsResponse>
  </s:Body>
</s:Envelope>

```



# Données dynamiques et temps réel

## getStopsNear () : obtenir les arrêts les plus proches

Paramètres d'entrée		
Nom	Type	Description
format	String	XML/JSON (XML par défaut)
longitude	Double	Longitude en WGS84
latitude	Double	Latitude en WGS84

Paramètres de sortie				
Name	Type	Description	Attributs	Type
Stop[]	Stop enumeration	Retourne les arrêts les plus proches du point	Stop_id	String
			Stop_lat	String
			Stop_lon	String
			Stop_name	String

### Informations sur les données de sortie :

La fonction retourne une liste d'au maximum 10 arrêts (Stop), classés du plus proche au plus éloigné, avec pour attributs :

- Stop\_id : identifiant de l'arrêt
- Stop\_lat : latitude de l'arrêt en WGS84
- Stop\_lon : longitude de l'arrêt en WGS84
- Stop\_name : nom commercial de la station (une station comprend communément 2 arrêts : un arrêt dans le sens aller et un arrêt dans le sens retour – sauf exceptions)

### Syntaxe REST

<https://applications002.brest-metropole.fr/WIPOD01/Transport/REST/getStopsNear?format={format}&longitude={ longitude }&latitude={latitude}>

### Exemples d'interrogations :

Requête REST :

<https://applications002.brest-metropole.fr/WIPOD01/Transport/REST/getStopsNear?format=json&latitude=48.403&longitude=-4.489>

Résultat

```
[{"Stop_id":"KERINO_1","Stop_lat":"48.4034343402479","Stop_lon":"-4.48898027270084","Stop_name":"Kerinou"}, {"Stop_id":"KERINO_2","Stop_lat":"48.4039242331636","Stop_lon":"-4.48893591152631","Stop_name":"Kerinou"}, {"Stop_id":"PEGUY_1","Stop_lat":"48.4020266012991","Stop_lon":"-4.48882328891027","Stop_name":"Peguy"}, {"Stop_id":"PEGUY_2","Stop_lat":"48.4018338
```

```
34137", "Stop_lon": "-4.48887939823506", "Stop_name": "Peguy"}, {"Stop_id": "PASTEU_2",  
...
```

### Requête SOAP :

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"  
xmlns:tem="http://tempuri.org/">  
  <soapenv:Header/>  
  <soapenv:Body>  
    <tem:getStopsNear>  
      <!--Optional:-->  
      <tem:format?></tem:format>  
      <!--Optional:-->  
      <tem:longitude>-4.489</tem:longitude>  
      <!--Optional:-->  
      <tem:latitude>48.403</tem:latitude>  
    </tem:getStopsNear>  
  </soapenv:Body>  
</soapenv:Envelope>
```

### Résultat

```
<s:Envelope xmlns:s="http://schemas.xmlsoap.org/soap/envelope/">  
  <s:Header />  
  <s:Body>  
    <getStopsNearResponse xmlns="http://tempuri.org/">  
      <getStopsNearResult  
xmlns:a="http://schemas.datacontract.org/2004/07/Dit.od"  
xmlns:i="http://www.w3.org/2001/XMLSchema-instance">  
        <a:Stop>  
          <a:Stop_id>KERINO_1</a:Stop_id>  
          <a:Stop_lat>48.4034343402479</a:Stop_lat>  
          <a:Stop_lon>-4.48898027270084</a:Stop_lon>  
          <a:Stop_name>Kerinou</a:Stop_name>  
        </a:Stop>  
        <a:Stop>  
          <a:Stop_id>KERINO_2</a:Stop_id>  
          <a:Stop_lat>48.4039242331636</a:Stop_lat>  
          <a:Stop_lon>-4.48893591152631</a:Stop_lon>  
          <a:Stop_name>Kerinou</a:Stop_name>  
        </a:Stop>...
```



## getStopNear\_route() : obtenir l'arrêt le plus proche pour une ligne

Paramètres d'entrée		
Nom	Type	Description
format	String	XML/JSON (XML par défaut)
route_id	String	Identifiant de la ligne (numéro de ligne)
longitude	Double	Longitude en WGS84
latitude	Double	Latitude en WGS84

Paramètres de sortie				
Name	Type	Description	Attributs	Type
Stop[]	Stop enumeration	Retourne l'arrêt le plus proche du point de coordonnées X,Y desservi par la ligne choisie	Stop_id	String
			Stop_lat	String
			Stop_lon	String
			Stop_name	String

### Informations sur les données de sortie :

La fonction retourne une liste contenant l'arrêt le plus proche (Stop) pour la ligne indiquée avec pour attributs :

- Stop\_id : identifiant de l'arrêt
- Stop\_lat : latitude de l'arrêt en WGS84
- Stop\_lon : longitude de l'arrêt en WGS84
- Stop\_name : nom commercial de la station (une station comprend communément 2 arrêts : un arrêt dans le sens aller et un arrêt dans le sens retour – sauf exceptions)

### Syntaxe REST

[https://applications002.brest-metropole.fr/WIPOD01/Transport/REST/getStopNear\\_route?format={format}&route\\_id={route\\_id}&latitude={latitude}&longitude={longitude}](https://applications002.brest-metropole.fr/WIPOD01/Transport/REST/getStopNear_route?format={format}&route_id={route_id}&latitude={latitude}&longitude={longitude})

### Exemples d'interrogations :

Requête REST :

[https://applications002.brest-metropole.fr/WIPOD01/Transport/REST/getStopNear\\_route?format=json&route\\_id=3&latitude=48.403&longitude=-4.489](https://applications002.brest-metropole.fr/WIPOD01/Transport/REST/getStopNear_route?format=json&route_id=3&latitude=48.403&longitude=-4.489)

Résultat

```
[{"Stop_id":"KERINO_1","Stop_lat":"48.4034343402479","Stop_lon":"-4.48898027270084","Stop_name":"Kerinou"}]
```

## Requête SOAP :

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:tem="http://tempuri.org/"
  <soapenv:Header/>
  <soapenv:Body>
    <tem:getStopNear>
      <!--Optional:-->
      <tem:format>xml</tem:format>
      <!--Optional:-->
      <tem:route_id>3</tem:route_id>
      <!--Optional:-->
      <tem:X>-4.489</tem:X>
      <!--Optional:-->
      <tem:Y>48.403</tem:Y>
    </tem:getStopNear>
  </soapenv:Body>
</soapenv:Envelope>
```

## Résultat:

```
<s:Envelope xmlns:s="http://schemas.xmlsoap.org/soap/envelope/">
  <s:Header />
  <s:Body>
    <getStopNear_routeResponse xmlns="http://tempuri.org/">
      <getStopNear_routeResult
xmlns:a="http://schemas.datacontract.org/2004/07/Dit.od"
xmlns:i="http://www.w3.org/2001/XMLSchema-instance">
        <a:Stop>
          <a:Stop_id>KERINO_1</a:Stop_id>
          <a:Stop_lat>48.4034343402479</a:Stop_lat>
          <a:Stop_lon>-4.48898027270084</a:Stop_lon>
          <a:Stop_name>Kerinou</a:Stop_name>
        </a:Stop>
      </getStopNear_routeResult>
    </getStopNear_routeResponse>
  </s:Body>
</s:Envelope>
```

## getStopVehiclesPosition() : obtenir la position (arrêt) des véhicules (bus ou tramway)

Paramètres d'entrée		
Nom	Type	Description
format	String	XML/JSON (XML par défaut)
route_id	String	Identifiant de la ligne (numéro de ligne)
trip_headsign	String	Nom de la destination

Paramètres de sortie				
Name	Type	Description	Attributs	Type
Stop[]	Stop enumeration	Retourne les arrêts où se situent présentement (ou à défaut qui viennent d'être desservis) les véhicules sur la ligne et pour la direction choisies	Stop_id	String
			Stop_lat	String
			Stop_lon	String
			Stop_name	String

### Informations sur les données de sortie :

La fonction retourne une liste d'arrêts (Stop) avec pour attributs :

- Stop\_id : numéro de la ligne permettant de l'identifier
- Stop\_lat : latitude de l'arrêt en WGS84
- Stop\_lon : longitude de l'arrêt en WGS84
- Stop\_name : nom commercial de la station (une station comprend communément 2 arrêts : un arrêt dans le sens aller et un arrêt dans le sens retour – sauf exceptions)

### Syntaxe REST

[https://applications002.brest-metropole.fr/WIPOD01/Transport/REST/getStopVehiclesPosition?format={format}&route\\_id={route\\_id}&trip\\_headsign={trip\\_headsign}](https://applications002.brest-metropole.fr/WIPOD01/Transport/REST/getStopVehiclesPosition?format={format}&route_id={route_id}&trip_headsign={trip_headsign})

### Exemples d'interrogations:

Requête REST :

[https://applications002.brest-metropole.fr/WIPOD01/Transport/REST/getStopVehiclesPosition?format=json&route\\_id=3&trip\\_headsign=Lambezellec](https://applications002.brest-metropole.fr/WIPOD01/Transport/REST/getStopVehiclesPosition?format=json&route_id=3&trip_headsign=Lambezellec)

Résultat

```
[{"Stop_id":"MALAKO_2","Stop_lat":"48.395918831208","Stop_lon":"-4.48412196331267","Stop_name":"Malakoff"}, {"Stop_id":"OCTROI_2","Stop_lat":"48.3969406524418","Stop_lon":"-4.47457978866822","Stop_name":"Octroi"},
```

## Requête SOAP :

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:tem="http://tempuri.org/"
  <soapenv:Header/>
  <soapenv:Body>
    <tem:getStopVehiclesPosition>
      <!--Optional:-->
      <tem:format?></tem:format>
      <!--Optional:-->
      <tem:route_id>3</tem:route_id>
      <!--Optional:-->
      <tem:trip_headsign>Lambezellec</tem:trip_headsign>
    </tem:getStopVehiclesPosition>
  </soapenv:Body>
</soapenv:Envelope>
```

## Résultat :

```
<s:Envelope xmlns:s="http://schemas.xmlsoap.org/soap/envelope/"
  <s:Body>
    <getStopVehiclesPositionResponse xmlns="http://tempuri.org/"
      <getStopVehiclesPositionResult
xmlns:a="http://schemas.datacontract.org/2004/07/Dit.od"
xmlns:i="http://www.w3.org/2001/XMLSchema-instance">
        <a:Stop>
          <a:Stop_id>F</a:Stop_id>
          <a:Stop_lat>48.3911382025031</a:Stop_lat>
          <a:Stop_lon>-4.48626159870198</a:Stop_lon>
          <a:Stop_name>Liberte Place</a:Stop_name>
        </a:Stop>
        <a:Stop>
          <a:Stop_id>DOURIC_2</a:Stop_id>
          <a:Stop_lat>48.3979859447856</a:Stop_lat>
          <a:Stop_lon>-4.46586703797156</a:Stop_lon>
          <a:Stop_name>Douric</a:Stop_name>
        </a:Stop>
        <a:Stop>
          <a:Stop_id>BERLIO_2</a:Stop_id>
          <a:Stop_lat>48.4130916724363</a:Stop_lat>
          <a:Stop_lon>-4.49767200144952</a:Stop_lon>
          <a:Stop_name>Berlioz</a:Stop_name>
        </a:Stop>
        <a:Stop>
          <a:Stop_id>PORT_P_2</a:Stop_id>
          <a:Stop_lat>48.3949388700467</a:Stop_lat>
          <a:Stop_lon>-4.43388398484272</a:Stop_lon>
          <a:Stop_name>Port Plaisance</a:Stop_name>
        </a:Stop>
      </getStopVehiclesPositionResult>
    </getStopVehiclesPositionResponse>
  </s:Body>
</s:Envelope>
```

## getGeolocatedVehiclesPosition() : obtenir la position (géographique) des véhicules

Paramètres d'entrée		
Nom	Type	Description
format	String	XML/JSON (XML par défaut)
route_id	String	Identifiant de la ligne (numéro de ligne)
trip_headsign	String	Nom de la destination

Paramètres de sortie				
Name	Type	Description	Attributs	Type
Vehicle[]	Vehicle enumeration	Retourne les positions (géographiques) où sont situés présentement les véhicules (bus ou tramways) sur la ligne et pour la direction choisies	Advance	String
			Cape	String
			Delay	String
			Lat	String
			Lon	String
			Pos	String
			State	String
			VehicleType	String

### Informations sur les données de sortie :

La fonction retourne une liste de véhicules(Vehicle) avec pour attributs :

- Advance : avance du véhicule par rapport à l'horaire théorique
- Cape : cap en degrés
- Delay : retard du véhicule par rapport à l'horaire théorique
- Lat : latitude du véhicule en WGS84
- Lon : longitude du véhicule en WGS84
- Pos : position curviligne sur la course
- State : état du véhicule (en ligne, défaut, ...)
- VehicleType : libellé de type de véhicule (bus standard, articulé, ou tramway)

### Syntaxe REST :

[https://applications002.brest-metropole.fr/WIPOD01/Transport/REST/getGeolocatedVehiclesPosition?format={format}&route\\_id={route\\_id}&trip\\_headsign={trip\\_headsign}](https://applications002.brest-metropole.fr/WIPOD01/Transport/REST/getGeolocatedVehiclesPosition?format={format}&route_id={route_id}&trip_headsign={trip_headsign})

### Exemples d'interrogations:

Requête REST :

[https://applications002.brest-metropole.fr/WIPOD01/Transport/REST/getGeolocatedVehiclesPosition?format=json&route\\_id=3&trip\\_headsign=Oceanopolis](https://applications002.brest-metropole.fr/WIPOD01/Transport/REST/getGeolocatedVehiclesPosition?format=json&route_id=3&trip_headsign=Oceanopolis)

## Résultat

```
[
  {
    "Advance": "00:03:12",
    "Cape": "64",
    "Delay": "00:00:00",
    "Lat": "48.3953878270202",
    "Lon": "-4.46085349437291",
    "Pos": "8375",
    "State": "LIGN",
    "VehicleType": "BUSSTD"
  },
  {
    "Advance": "00:00:00",
    "Cape": "117",
    "Delay": "00:01:05",
    "Lat": "48.3972798991508",
    "Lon": "-4.44353768850528",
    "Pos": "10155",
    "State": "LIGN",
    "VehicleType": "BUSSTD"
  },
  ...
]
```

## Requête SOAP :

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:tem="http://tempuri.org/"
  <soapenv:Header/>
  <soapenv:Body>
    <tem:getGeolocatedVehiclesPosition>
      <!--Optional:-->
      <tem:format?></tem:format>
      <!--Optional:-->
      <tem:route_id>3</tem:route_id>
      <!--Optional:-->
      <tem:trip_headsign>Oceanopolis</tem:trip_headsign>
    </tem:getGeolocatedVehiclesPosition>
  </soapenv:Body>
</soapenv:Envelope>
```

## Résultat :

```
<s:Envelope xmlns:s="http://schemas.xmlsoap.org/soap/envelope/"
  <s:Body>
    <getGeolocatedVehiclesPositionResponse xmlns="http://tempuri.org/"
      <getGeolocatedVehiclesPositionResult
xmlns:a="http://schemas.datacontract.org/2004/07/Dit.od"
xmlns:i="http://www.w3.org/2001/XMLSchema-instance">
        <a:Vehicle>
          <a:Advance>00:01:54</a:Advance>
          <a:Cape>355</a:Cape>
          <a:Delay>00:00:00</a:Delay>
          <a:Lat>48.393275035126</a:Lat>
          <a:Lon>-4.47445386393795</a:Lon>
          <a:Pos>6595</a:Pos>
          <a:State>LIGN</a:State>
          <a:VehicleType>BUSSTD</a:VehicleType>
        </a:Vehicle>
        <a:Vehicle>
          <a:Advance>00:00:00</a:Advance>
          <a:Cape>64</a:Cape>
          <a:Delay>00:01:45</a:Delay>
          <a:Lat>48.3952383524736</a:Lat>
          <a:Lon>-4.46125410249057</a:Lon>
```

```
<a:Pos>8339</a:Pos>  
<a:State>LIGN</a:State>  
<a:VehicleType>BUSSTD</a:VehicleType>  
</a:Vehicle>
```

...

## getNextDepartures() : obtenir les horaires des prochains départs

Paramètres d'entrée		
Nom	Type	Description
format	String	XML/JSON (XML par défaut)
route_id	String	Identifiant de la ligne (numéro de ligne)
stop_name	String	Nom de l'arrêt
trip_headsign	String	Nom de la destination

Paramètres de sortie				
Name	Type	Description	Attributs	Type
Time[]	Times enumeration	Retourne l'horaire des prochains départs pour l'arrêt, la ligne et la destination choisis	Applicable_Arrival_Time	String
			Applicable_Departure_Time	String
			Arrival_time	String
			Departure_time	String
			EstimateTime_arrivalRealized	String
			EstimateTime_departureRealized	String
			Reliable	String
Stop_type	String			

### Informations sur les données de sortie :

La fonction retourne une liste d'horaires (time) avec pour attributs :

- Applicable\_arrival\_time : heure réelle d'arrivée
- Applicable\_departure\_time : heure réelle de départ
- Arrival\_time : heure théorique d'arrivée
- Departure\_time : heure théorique de départ
- EstimateTime\_arrivalRealized : estimation de l'heure d'arrivée réelle
- EstimateTime\_departureRealized : estimation de l'heure de départ réelle
- Reliable : fiabilité de l'heure estimée (fiable si l'horaire estimé a été calculé, horaire théorique sinon)
- Stop\_type : départ, arrivée, référence, normal

### Syntaxe REST

[https://applications002.brest-metropole.fr/WIPOD01/Transport/REST/getNextDepartures?format={format}&route\\_id={route\\_id}&trip\\_headsign={trip\\_headsign}&stop\\_name={stop\\_name}](https://applications002.brest-metropole.fr/WIPOD01/Transport/REST/getNextDepartures?format={format}&route_id={route_id}&trip_headsign={trip_headsign}&stop_name={stop_name})

### Exemples d'interrogations:

Requête REST :



[https://applications002.brest-metropole.fr/WIPOD01/Transport/REST/getNextDepartures?format=json&route\\_id=3&stop\\_name=giloux&trip\\_headsign=lambezellec](https://applications002.brest-metropole.fr/WIPOD01/Transport/REST/getNextDepartures?format=json&route_id=3&stop_name=giloux&trip_headsign=lambezellec)

Résultat :

```
[
  {
    "Applicable_Arrival_Time": "14:47:15",
    "Applicable_Departure_Time": "14:47:15",
    "Arrival_time": "14:47:15",
    "Departure_time": "14:47:15",
    "EstimateTime_arrivalRealized": "14:45:28",
    "EstimateTime_departureRealized": "14:45:28",
    "Reliable": "F",
    "StopType": "N"
  },
  {
    "Applicable_Arrival_Time": "14:57:15",
    "Applicable_Departure_Time": "14:57:15",
    "Arrival_time": "14:57:15",
    "Departure_time": "14:57:15",
    "EstimateTime_arrivalRealized": "14:57:24",
    "EstimateTime_departureRealized": "14:57:24",
    "Reliable": "F",
    "StopType": "N"
  },
  ...
]
```

Requête SOAP :

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:tem="http://tempuri.org/">
  <soapenv:Header/>
  <soapenv:Body>
    <tem:getNextDepartures>
      <!--Optional:-->
      <tem:format?></tem:format>
      <!--Optional:-->
      <tem:route_id>3</tem:route_id>
      <!--Optional:-->
      <tem:trip_headsign>lambezellec</tem:trip_headsign>
      <!--Optional:-->
      <tem:stop_name>giloux</tem:stop_name>
    </tem:getNextDepartures>
  </soapenv:Body>
</soapenv:Envelope>
```

Résultat :

```
<s:Envelope xmlns:s="http://schemas.xmlsoap.org/soap/envelope/">
  <s:Body>
    <getNextDeparturesResponse xmlns="http://tempuri.org/">
      <getNextDeparturesResult
xmlns:a="http://schemas.datacontract.org/2004/07/Dit.od"
xmlns:i="http://www.w3.org/2001/XMLSchema-instance">
        <a:Time>
          <a:Applicable_Arrival_Time>14:09:15</a:Applicable_Arrival_Time>
          <a:Applicable_Departure_Time>14:09:15</a:Applicable_Departure_Time>
          <a:Arrival_time>14:09:15</a:Arrival_time>
          <a:Departure_time>14:09:15</a:Departure_time>
        </a:Time>
        <a:EstimateTime_arrivalRealized>14:07:09</a:EstimateTime_arrivalRealized>
      </getNextDeparturesResult>
    </getNextDeparturesResponse>
  </s:Body>
</s:Envelope>
```

```
<a:EstimateTime_departureRealized>14:07:09</a:EstimateTime_departureRealized>
  <a:Reliable>F</a:Reliable>
  <a:StopType>N</a:StopType>
</a:Time>
<a:Time>
  <a:Applicable_Arrival_Time>14:18:15</a:Applicable_Arrival_Time>
  <a:Applicable_Departure_Time>14:18:15</a:Applicable_Departure_Time>
  <a:Arrival_time>14:18:15</a:Arrival_time>
  <a:Departure_time>14:18:15</a:Departure_time>

<a:EstimateTime_arrivalRealized>14:18:59</a:EstimateTime_arrivalRealized>

<a:EstimateTime_departureRealized>14:18:59</a:EstimateTime_departureRealized>
  <a:Reliable>F</a:Reliable>
  <a:StopType>N</a:StopType>
</a:Time>

...
```

## getNextDepartures2() : obtenir les horaires des prochains départs

Paramètres d'entrée		
Nom	Type	Description
format	String	XML/JSON (XML par défaut)
route_id	String	Identifiant de la ligne (numéro de ligne)
stop_id	String	Id de l'arrêt
trip_headsign	String	Nom de la destination

Paramètres de sortie				
Name	Type	Description	Attributs	Type
Time[]	Times enumeration	Retourne l'horaire des prochains départs pour l'arrêt, la ligne et la destination choisis	Applicable_Arrival_Time	String
			Applicable_Departure_Time	String
			Arrival_time	String
			Departure_time	String
			EstimateTime_arrivalRealized	String
			EstimateTime_departureRealized	String
			Reliable	String
Stop_type	String			

### Informations sur les données de sortie :

La fonction retourne une liste d'horaires (time) avec pour attributs :

- Applicable\_arrival\_time : heure réelle d'arrivée
- Applicable\_departure\_time : heure réelle de départ
- Arrival\_time : heure théorique d'arrivée
- Departure\_time : heure théorique de départ
- EstimateTime\_arrivalRealized : estimation de l'heure d'arrivée réelle
- EstimateTime\_departureRealized : estimation de l'heure de départ réelle
- Reliable : fiabilité de l'heure estimée (fiable si l'horaire estimé a été calculé, horaire théorique sinon)
- Stop\_type : départ, arrivée, référence, normal

### Syntaxe REST

[https://applications002.brest-metropole.fr/WIPOD01/Transport/REST/getNextDepartures2?format={format}&route\\_id={route\\_id}&trip\\_headsign={trip\\_headsign}&stop\\_id={stop\\_id}](https://applications002.brest-metropole.fr/WIPOD01/Transport/REST/getNextDepartures2?format={format}&route_id={route_id}&trip_headsign={trip_headsign}&stop_id={stop_id})

## Exemples d'interrogations:

Requête REST :

[https://applications002.brest-metropole.fr/WIPOD01/Transport/REST/getNextDepartures2?format=json&route\\_id=3&stop\\_id=GILOUX\\_2&trip\\_headsign=lambezellec](https://applications002.brest-metropole.fr/WIPOD01/Transport/REST/getNextDepartures2?format=json&route_id=3&stop_id=GILOUX_2&trip_headsign=lambezellec)

Résultat :

```
[
  {
    "Applicable_Arrival_Time": "14:00:13",
    "Applicable_Departure_Time": "14:00:13",
    "Arrival_time": "14:00:13",
    "Departure_time": "14:00:13",
    "EstimateTime_arrivalRealized": "14:02:55",
    "EstimateTime_departureRealized": "14:02:55",
    "Reliable": "F",
    "StopType": "N"
  },
  {
    "Applicable_Arrival_Time": "14:09:13",
    "Applicable_Departure_Time": "14:09:13",
    "Arrival_time": "14:09:13",
    "Departure_time": "14:09:13",
    "EstimateTime_arrivalRealized": "14:09:26",
    "EstimateTime_departureRealized": "14:09:26",
    "Reliable": "F",
    "StopType": "N"
  },
  ...
]
```

Requête SOAP :

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:tem="http://tempuri.org/">
  <soapenv:Header/>
  <soapenv:Body>
    <tem:getNextDepartures>
      <!--Optional:-->
      <tem:format?></tem:format>
      <!--Optional:-->
      <tem:route_id>3</tem:route_id>
      <!--Optional:-->
      <tem:trip_headsign>lambezellec</tem:trip_headsign>
      <!--Optional:-->
      <tem:stop_name>giloux</tem:stop_name>
    </tem:getNextDepartures>
  </soapenv:Body>
</soapenv:Envelope>
```

Résultat :

```
<s:Envelope xmlns:s="http://schemas.xmlsoap.org/soap/envelope/">
  <s:Body>
    <getNextDepartures2Response xmlns="http://tempuri.org/">
      <getNextDepartures2Result
xmlns:a="http://schemas.datacontract.org/2004/07/Dit.od"
xmlns:i="http://www.w3.org/2001/XMLSchema-instance">
        <a:Time>
          <a:Applicable_Arrival_Time>14:00:13</a:Applicable_Arrival_Time>
          <a:Applicable_Departure_Time>14:00:13</a:Applicable_Departure_Time>
        </a:Time>
      </getNextDepartures2Result>
    </getNextDepartures2Response>
  </s:Body>
</s:Envelope>
```

```
        <a:Arrival_time>14:00:13</a:Arrival_time>
        <a:Departure_time>14:00:13</a:Departure_time>
<a:EstimateTime_arrivalRealized>14:01:35</a:EstimateTime_arrivalRealized>
<a:EstimateTime_departureRealized>14:01:35</a:EstimateTime_departureRealized>
    <a:Reliable>F</a:Reliable>
    <a:StopType>N</a:StopType>
  </a:Time>
  <a:Time>
    <a:Applicable_Arrival_Time>14:09:13</a:Applicable_Arrival_Time>
    <a:Applicable_Departure_Time>14:09:13</a:Applicable_Departure_Time>
    <a:Arrival_time>14:09:13</a:Arrival_time>
    <a:Departure_time>14:09:13</a:Departure_time>
<a:EstimateTime_arrivalRealized>14:09:26</a:EstimateTime_arrivalRealized>
<a:EstimateTime_departureRealized>14:09:26</a:EstimateTime_departureRealized>
    <a:Reliable>F</a:Reliable>
    <a:StopType>N</a:StopType>
  </a:Time>
  ...
```

## getRemainingTimes () : obtenir les temps d'attente avant les prochains passages

Paramètres d'entrée		
Nom	Type	Description
format	String	XML/JSON (XML par défaut)
route_id	String	Identifiant de la ligne (numéro de ligne)
stop_name	String	Nom de l'arrêt
trip_headsign	String	Nom de la destination

Paramètres de sortie				
Name	Type	Description	Attributs	Type
RemainingTime[]	Stop enumeration	Retourne les temps d'attente avant les prochains passages pour la ligne, l'arrêt et la destination choisis	Advance route_id	String
			Arrival_time	String
			Delay	String
			EstimateTime_arrivalRealized	String
			Remaining_time	String

### Informations sur les données de sortie :

La fonction retourne une liste de temps d'attente (RemainingTimes) avec pour attributs :

- Advance : avance du véhicule par rapport à l'horaire théorique
- Arrival\_time : heure théorique d'arrivée
- Delay : retard du véhicule par rapport à l'horaire théorique
- EstimateTime\_arrivalRealized : heure d'arrivée estimée
- Remaining\_time : temps d'attente restant

Remarque : les temps d'attente avant un prochain passage se limitent environ à une heure.

### Syntaxe REST

[https://applications002.brest-metropole.fr/WIPOD01/Transport/REST/getRemainingTimes?format={format}&route\\_id={route\\_id}&trip\\_headsign={trip\\_headsign}&stop\\_name={stop\\_name}](https://applications002.brest-metropole.fr/WIPOD01/Transport/REST/getRemainingTimes?format={format}&route_id={route_id}&trip_headsign={trip_headsign}&stop_name={stop_name})

### Exemples d'interrogations:

Requête REST :

[https://applications002.brest-metropole.fr/WIPOD01/Transport/REST/getRemainingTimes?format=json&route\\_id=3&trip\\_headsign=oceanopolis&stop\\_name=Malakoff](https://applications002.brest-metropole.fr/WIPOD01/Transport/REST/getRemainingTimes?format=json&route_id=3&trip_headsign=oceanopolis&stop_name=Malakoff)

## Résultat

[

```
{
  "Advance": "00:01:07",
  "Arrival_time": "14:49:56",
  "Delay": "00:00:00",
  "EstimateTime_arrivalRealized": "14:49:03",
  "Remaining_time": "00:06:07"
},
{
  "Advance": "00:00:00",
  "Arrival_time": "15:00:56",
  "Delay": "00:01:04",
  "EstimateTime_arrivalRealized": "15:02:01",
  "Remaining_time": "00:19:05"
},
{
  "Advance": "00:00:00",
  "Arrival_time": "15:10:56",
  "Delay": "00:00:00",
  "EstimateTime_arrivalRealized": "15:10:56",
  "Remaining_time": "00:28:00"
}
]
```

## Requête SOAP :

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:tem="http://tempuri.org/"
  <soapenv:Header/>
  <soapenv:Body>
    <tem:getRemainingTimes>
      <!--Optional:-->
      <tem:format?></tem:format>
      <!--Optional:-->
      <tem:route_id>3</tem:route_id>
      <!--Optional:-->
      <tem:trip_headsign>oceanopolis</tem:trip_headsign>
      <!--Optional:-->
      <tem:stop_name>malakoff</tem:stop_name>
    </tem:getRemainingTimes>
  </soapenv:Body>
</soapenv:Envelope>
```

## Résultat

```
<s:Envelope xmlns:s="http://schemas.xmlsoap.org/soap/envelope/">
  <s:Body>
    <getRemainingTimesResponse xmlns="http://tempuri.org/"
      <getRemainingTimesResult
xmlns:a="http://schemas.datacontract.org/2004/07/Dit.od"
xmlns:i="http://www.w3.org/2001/XMLSchema-instance">
        <a:RemainingTime>
          <a:Advance>00:00:00</a:Advance>
          <a:Arrival_time>15:00:56</a:Arrival_time>
          <a:Delay>00:01:44</a:Delay>
        </a:RemainingTime>
        <a:EstimateTime_arrivalRealized>15:02:42</a:EstimateTime_arrivalRealized>
        <a:Remaining_time>00:11:12</a:Remaining_time>
      </a:RemainingTime>
    </s:Body>
  </s:Envelope>
```

```
        <a:Advance>00:00:00</a:Advance>
        <a:Arrival_time>15:10:56</a:Arrival_time>
        <a:Delay>00:00:00</a:Delay>
    <a:EstimateTime_arrivalRealized>15:10:58</a:EstimateTime_arrivalRealized>
        <a:Remaining_time>00:19:28</a:Remaining_time>
    </a:RemainingTime>
    <a:RemainingTime>
        <a:Advance>00:00:00</a:Advance>
        <a:Arrival_time>15:20:56</a:Arrival_time>
        <a:Delay>00:00:00</a:Delay>
    <a:EstimateTime_arrivalRealized>15:20:56</a:EstimateTime_arrivalRealized>
        <a:Remaining_time>00:29:26</a:Remaining_time>
    </a:RemainingTime>
</getRemainingTimesResult>
</getRemainingTimesResponse>
</s:Body>
</s:Envelope>
```



## getRemainingTimes2 () : obtenir les temps d'attente avant les prochains passages

Paramètres d'entrée		
Nom	Type	Description
format	String	XML/JSON (XML par défaut)
route_id	String	Identifiant de la ligne (numéro de ligne)
stop_id	String	Id de l'arrêt
trip_headsign	String	Nom de la destination

Paramètres de sortie				
Name	Type	Description	Attributs	Type
RemainingTime[]	Stop enumeration	Retourne les temps d'attente avant les prochains passages pour la ligne, l'arrêt et la destination choisis	Advance route_id	String
			Arrival_time	String
			Delay	String
			EstimateTime_arrivalRealized	String
			Remaining_time	String

### Informations sur les données de sortie :

La fonction retourne une liste de temps d'attente (RemainingTimes) avec pour attributs :

- Advance : avance du véhicule par rapport à l'horaire théorique
- Arrival\_time : heure théorique d'arrivée
- Delay : retard du véhicule par rapport à l'horaire théorique
- EstimateTime\_arrivalRealized : heure d'arrivée estimée
- Remaining\_time : temps d'attente restant

Remarque : les temps d'attente avant un prochain passage se limitent environ à une heure.

### Syntaxe REST

[https://applications002.brest-metropole.fr/WIPOD01/Transport/REST/getRemainingTimes2?format={format}&route\\_id={route\\_id}&trip\\_headsign={trip\\_headsign}&stop\\_id={stop\\_id}](https://applications002.brest-metropole.fr/WIPOD01/Transport/REST/getRemainingTimes2?format={format}&route_id={route_id}&trip_headsign={trip_headsign}&stop_id={stop_id})

### Exemples d'interrogations:

Requête REST :

[https://applications002.brest-metropole.fr/WIPOD01/Transport/REST/getRemainingTimes2?format=json&route\\_id=3&trip\\_headsign=oceanopolis&stop\\_id=MALAKO\\_1](https://applications002.brest-metropole.fr/WIPOD01/Transport/REST/getRemainingTimes2?format=json&route_id=3&trip_headsign=oceanopolis&stop_id=MALAKO_1)

## Résultat

```
[
  {
    "Advance": "00:00:05",
    "Arrival_time": "14:08:59",
    "Delay": "00:00:00",
    "EstimateTime_arrivalRealized": "14:08:54",
    "Remaining_time": "00:02:19"
  },
  {
    "Advance": "00:01:00",
    "Arrival_time": "14:18:59",
    "Delay": "00:00:00",
    "EstimateTime_arrivalRealized": "14:17:59",
    "Remaining_time": "00:11:24"
  },
  {
    "Advance": "00:00:00",
    "Arrival_time": "14:29:59",
    "Delay": "00:00:00",
    "EstimateTime_arrivalRealized": "14:29:59",
    "Remaining_time": "00:23:24"
  }
]
```

## Requête SOAP :

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:tem="http://tempuri.org/">
  <soapenv:Header/>
  <soapenv:Body>
    <tem:getRemainingTimes>
      <!--Optional:-->
      <tem:format?></tem:format>
      <!--Optional:-->
      <tem:route_id>3</tem:route_id>
      <!--Optional:-->
      <tem:trip_headsign>oceanopolis</tem:trip_headsign>
      <!--Optional:-->
      <tem:stop_name>MALAKO_1</tem:stop_name>
    </tem:getRemainingTimes>
  </soapenv:Body>
</soapenv:Envelope>
```

## Résultat

```
<s:Envelope xmlns:s="http://schemas.xmlsoap.org/soap/envelope/">
  <s:Body>
    <getRemainingTimes2Response xmlns="http://tempuri.org/">
      <getRemainingTimes2Result
xmlns:a="http://schemas.datacontract.org/2004/07/Dit.od"
xmlns:i="http://www.w3.org/2001/XMLSchema-instance">
        <a:RemainingTime>
          <a:Advance>00:01:03</a:Advance>
          <a:Arrival_time>14:18:59</a:Arrival_time>
          <a:Delay>00:00:00</a:Delay>
        </a:RemainingTime>
        <a:EstimateTime_arrivalRealized>14:18:14</a:EstimateTime_arrivalRealized>
        <a:Remaining_time>00:06:26</a:Remaining_time>
      </a:RemainingTime>
    </a:RemainingTime>
  </s:Body>
</s:Envelope>
```

```
<a:Advance>00:00:00</a:Advance>
<a:Arrival_time>14:29:59</a:Arrival_time>
<a:Delay>00:01:14</a:Delay>
<a:EstimateTime_arrivalRealized>14:31:14</a:EstimateTime_arrivalRealized>
  <a:Remaining_time>00:19:26</a:Remaining_time>
    </a:RemainingTime>
  </getRemainingTimes2Result>
</getRemainingTimes2Response>
</s:Body>
</s:Envelope>
```

## getDeviations() : obtenir les déviations

Paramètres d'entrée		
Nom	Type	Description
format	String	XML/JSON (XML par défaut)
route_id	String	Identifiant de la ligne (numéro de ligne)
trip_headsign	String	Nom de la destination

Paramètres de sortie				
Name	Type	Description	Attributs	Type
Deviation[]	Deviation enumeration	Retourne la déviation pour la ligne et la destination choisies	active	String
			stop_name_end	String
			stop_name_start	String
			Stop_id_end	String
			Stop_id_start	String

### Informations sur les données de sortie :

La fonction retourne une liste de déviations (Deviation) avec pour attributs :

- Active : la déviation est active ou non
- stop\_name\_end : nom de l'arrêt de fin de déviation
- stop\_name\_start : nom de l'arrêt de début de déviation
- stop\_id\_end : Id de l'arrêt de fin de déviation
- Stop\_id\_start : Id de l'arrêt de début de déviation

### Syntaxe REST

[https://applications002.brest-metropole.fr/WIPOD01/Transport/REST/getDeviations?format={format}&route\\_id={route\\_id}&trip\\_headsign={trip\\_headsign}](https://applications002.brest-metropole.fr/WIPOD01/Transport/REST/getDeviations?format={format}&route_id={route_id}&trip_headsign={trip_headsign})

### Exemples d'interrogations:

Requête REST :

[https://applications002.brest-metropole.fr/WIPOD01/Transport/REST/getDeviations?format=json&route\\_id=5&trip\\_headsign=Provence](https://applications002.brest-metropole.fr/WIPOD01/Transport/REST/getDeviations?format=json&route_id=5&trip_headsign=Provence)

### Résultat

```
[{
  "Active": "0",
  "Stop_id_end": "J_LESV_2",
  "Stop_id_start": "BRETAG_5",
  "Stop_name_end": "Jules Lesven",
  "Stop_name_start": "Place Bretagne"
}]
```

## Requête SOAP :

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:tem="http://tempuri.org/">
  <soapenv:Header/>
  <soapenv:Body>
    <tem:getDeviations>
      <!--Optional:-->
      <tem:format?></tem:format>
      <!--Optional:-->
      <tem:route_id>5</tem:route_id>
      <!--Optional:-->
      <tem:trip_headsign>Provence</tem:trip_headsign>
    </tem:getDeviations>
  </soapenv:Body>
</soapenv:Envelope>
```

## Résultat :

```
<s:Envelope xmlns:s="http://schemas.xmlsoap.org/soap/envelope/">
  <s:Body>
    <getDeviationsResponse xmlns="http://tempuri.org/">
      <getDeviationsResult
xmlns:a="http://schemas.datacontract.org/2004/07/Dit.od"
xmlns:i="http://www.w3.org/2001/XMLSchema-instance">
        <a:Deviation>
          <a:Active>0</a:Active>
          <a:Stop_id_end>J_LESV_2</a:Stop_id_end>
          <a:Stop_id_start>BRETAG_5</a:Stop_id_start>
          <a:Stop_name_end>Jules Lesven</a:Stop_name_end>
          <a:Stop_name_start>Place Bretagne</a:Stop_name_start>
        </a:Deviation>
      </getDeviationsResult>
    </getDeviationsResponse>
  </s:Body>
</s:Envelope>
```

## getPerturbations() : obtenir les perturbations

Paramètres d'entrée		
Nom	Type	Description
format	String	XML/JSON (XML par défaut)

Paramètres de sortie				
Name	Type	Description	Attributs	Type
Perturbation[]	Perturbation enumeration	Retourne les perturbations à partir des informations présentes dans le flux RSS : <a href="http://www.bibus.fr/rssInfoTrafic.aspx">http://www.bibus.fr/rssInfoTrafic.aspx</a>	Description	String
			Link	
			PubDate	String
			Title	String

### Informations sur les données de sortie :

La fonction retourne une liste des perturbations (Perturbation) avec pour attributs :

- Description : description de la perturbation
- Link : lien renvoyant vers le détail de la perturbation
- PubDate : date de publication de la perturbation
- Title : nom de la perturbation

### Syntaxe REST

<https://applications002.brest-metropole.fr/WIPOD01/Transport/REST/getPerturbations?format={format}>

### Exemples d'interrogations:

Requête REST :

<https://applications002.brest-metropole.fr/WIPOD01/Transport/REST/getPerturbations?format=json>

### Résultat

```
[
  {
    "Description": "Les travaux continuent Place du Calvaire dans les 2 sens, l'itinéraire des lignes 8 et 54 sont déviés à compter du lundi 7 septembre pour la semaine.\nVers Plougastel :\n\n    L'arrêt Méheut vers Ty-ar-Ménez est reporté à l'arrêt Fontaine Blanche.\n    L'arrêt Champ de Foire est reporté à l'arrêt Kerdanguy.\n\nVers Liberté :\n\n    L'arrêt Champ de Foire est reporté soit à l'arrêt Méheut soit à l'arrêt Kerdanguy.\n\nNous vous prions de nous excuser pour ces désagréments.\n ",
    "Link": "http://www.bibus.fr/Actualite.aspx?ActualiteID=1492#1492",
    "PubDate": "07/09/2015 00:00:00 +00:00",
```

```

    "Title": "Info trafic : Les travaux continuent Place du Calvaire : Ligne 8
déviiée."
  },
  {
    "Description": "En raison de travaux de réfection de chaussée sur la RD05 au
niveau de l'entrée de Guilers, la ligne 10 est déviée dans les 2 sens à compter du
lundi 7 septembre (8h00) pour une durée prévue de 2 semaines.\n \nEn direction de
Guilers :\n\n l'arrêt Castelmein est reporté à l'arrêt Penfeld.\n les arrêts
Coat ty Ogant, Coat-Mez et Lancelot sont reportés au poteau provisoire situé rue
Lemonnier, à côté de l'école Ste Thérèse.\n\nEn direction de Liberté :\n\n les
arrêts Lancelot et Coat-Mez sont reportés au poteau provisoire situé rue
Lemonnier, à côté de l'école Ste Thérèse.\n l'arrêt Penfeld est reporté au
poteau provisoire avant le Rdp du Parc des Expositions sur la route de Pont
Cabioch, en venant de la route de St-Pierre.\n\nNous vous prions de nous excuser
pour ces désagréments.\n ",
    "Link": "http://www.bibus.fr/Actualite.aspx?ActualiteID=1493#1493",
    "PubDate": "07/09/2015 00:00:00 +00:00",
    "Title": "Info trafic : Travaux sur RD05 Guilers"
  }
]

```

### Requête SOAP :

```

<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:tem="http://tempuri.org/"
  <soapenv:Header/>
  <soapenv:Body>
    <tem:getPerturbations>
      <!--Optional:-->
      <tem:format?></tem:format>
    </tem:getPerturbations>
  </soapenv:Body>
</soapenv:Envelope>

```

### Résultat :

```

<s:Envelope xmlns:s="http://schemas.xmlsoap.org/soap/envelope/">
  <s:Body>
    <getPerturbationsResponse xmlns="http://tempuri.org/">
      <getPerturbationsResult
xmlns:a="http://schemas.datacontract.org/2004/07/Dit.od"
xmlns:i="http://www.w3.org/2001/XMLSchema-instance">
        <a:Perturbation>
          <a:Description>Les travaux continuent Place du Calvaire dans les 2
sens, l'itinéraire des lignes 8 et 54 sont déviés à compter du lundi 7 septembre
pour la semaine.
Vers Plougastel :

          L'arrêt Méheut vers Ty-ar-Ménez est reporté à l'arrêt Fontaine Blanche.
          L'arrêt Champ de Foire est reporté à l'arrêt Kerdanguy.

Vers Liberté :

          L'arrêt Champ de Foire est reporté soit à l'arrêt Méheut soit à l'arrêt
Kerdanguy.

          Nous vous prions de nous excuser pour ces désagréments.
        </a:Description>
        <a:Link>http://www.bibus.fr/Actualite.aspx?ActualiteID=1492#1492</a:Link>
          <a:PubDate>07/09/2015 00:00:00 +00:00</a:PubDate>
          <a:Title>Info trafic : Les travaux continuent Place du Calvaire :
Ligne 8 déviée.</a:Title>
        </a:Perturbation>
      </a:Perturbation>
    </getPerturbationsResult>
  </s:Body>
</s:Envelope>

```

<a:Description>En raison de travaux de réfection de chaussée sur la RD05 au niveau de l'entrée de Guilers, la ligne 10 est déviée dans les 2 sens à compter du lundi 7 septembre (8h00) pour une durée prévue de 2 semaines.

En direction de Guilers :

l'arrêt Castelmein est reporté à l'arrêt Penfeld.  
les arrêts Coat ty Ogant, Coat-Mez et Lancelot sont reportés au poteau provisoire situé rue Lemonnier, à côté de l'école Ste Thérèse.

En direction de Liberté :

les arrêts Lancelot et Coat-Mez sont reportés au poteau provisoire situé rue Lemonnier, à côté de l'école Ste Thérèse.

l'arrêt Penfeld est reporté au poteau provisoire avant le Rdp du Parc des Expositions sur la route de Pont Cabioch, en venant de la route de St-Pierre.

Nous vous prions de nous excuser pour ces désagréments.

</a:Description>

<a:Link><http://www.bibus.fr/Actualite.aspx?ActualiteID=1493#1493></a:Link>

<a:PubDate>07/09/2015 00:00:00 +00:00</a:PubDate>

<a:Title>Info trafic : Travaux sur RD05 Guilers</a:Title>

</a:Perturbation>

</getPerturbationsResult>

</getPerturbationsResponse>

</s:Body>

</s:Envelope>