

Api RDEX

Documentation v1.2.1



Rédaction : Covivo pour le compte de la Feduco

Table des matières

Description générale.....	4
Architecture de base.....	4
Accès à l'api.....	4
Format des données.....	5
Gestion des erreurs.....	5
Accès aux ressources.....	7
Méthode GET journeys.....	7
Méthode POST connections.....	11
Annexes.....	13
Annexe 1 : Structure de journeys.....	13
Annexe 2 : Exemple de journeys régulier aller-retour.....	15
Annexe 3 : Liste des codes d'erreur.....	18
GLOSSAIRE.....	19

Historique des versions :

Version	Modifications apportées par rapport à la version précédente.
1.1.3	<ul style="list-style-type: none"> – Ajout de deux champs dans L'Annexe 1 STRUCTURE DES JOURNEYS pour la prise en compte du temps réel : champ real_time et stopped. – Une indication supplémentaire en bas de chaque requête : « Remplacer : <code>www.operator.com</code> par le site de l'opérateur concerné ».
1.1.4	<ul style="list-style-type: none"> – Ajout d'un tableau contenant les paramètres de recherche acceptés par la méthode GET JOURNEYS.
1.2.0	<ul style="list-style-type: none"> – Ajout d'exemples – Mise en forme pour rendre les spécifications plus claires
1.2.1	<ul style="list-style-type: none"> – Annexe structure connections supprimée et remplacée par « Paramètres » dans « Méthode POST connections »

1 Description générale

Dans la suite du document, est désigné par `www.operator.com` le site de covoiturage souhaitant mettre en place RDEX.

L'encodage de caractères est UTF-8 aussi bien en entrée qu'en sortie.

1.1. Architecture de base

L'api RDEX est une api REST utilisant les 4 opérations du modèle CRUD (Create, Read, Update, Delete).

Méthode	Action
GET	On lit une ressource
POST	On crée une ressource
PUT	On modifie une ressource
DELETE	On supprime une ressource

Une requête à l'api RDEX respecte le format de base suivant :

`https://www.operator.com/rdexapi/ressource.json?`

`timestamp=timestamp&apikey=cléPublique&p=paramètres&signature=signature`

```
'https://www.operator.com/rdexapi/journeys.json?  
timestamp=1326359986&apikey=operator_public_key&signature=b249c7e56cd8b7c3caf2  
c3333cb446bca5d1c51bd35bf824528daacea9dc4ae6'
```

1.2. Accès à l'api

L'accès à l'api se fait en https (SSL). Il est restreint : les utilisateurs de l'api doivent être au préalable enregistrés auprès de l'opérateur de covoiturage. La demande d'enregistrement se fait par envoi d'un email à l'adresse email de contact de l'opérateur de covoiturage.

1.2.1 Restriction sur adresse ip

L'accès à l'api n'est autorisé que pour une liste d'adresse ip définie. L'utilisateur doit communiquer l'adresse ip du serveur qui interrogera l'api afin de pouvoir l'utiliser.

1.2.2 Authentification

Suivant le paradigme REST, l'api RDEX est sans-état (stateless). Il n'y a donc pas de système de session.

Chaque utilisateur se voit attribuer une clé publique (apikey) et une clé privée (privatekey).

L'apikey étant publique, elle n'est soumise à aucune politique de confidentialité particulière. Elle permet de s'identifier auprès de l'api RDEX.

La privatekey est personnelle et strictement confidentielle. Elle permet de vérifier l'identité de l'utilisateur par l'intermédiaire du mécanisme de signature.

1.2.3 Signature

Chaque appel à l'api doit être signé. Pour cela, une clé cryptographique privée (privatekey) est communiquée à chaque utilisateur. Cette clé est unique pour chaque utilisateur et doit rester absolument confidentielle. En cas de divulgation de la clé cryptographique, l'utilisateur en sera tenu pour seul responsable. Une nouvelle clé peut néanmoins être attribuée, sur demande, en cas de corruption de celle-ci.

Concrètement, la signature est obtenue en appliquant l'algorithme de hachage 'sha256' à l'url non signée et tous ses paramètres (sauf le paramètre signature) en utilisant la privatekey comme valeur de salaison et ceci après un url encode (conforme à la RFC 1738 ou à la RFC 3986, les deux sont supportées).

Exemple php :

```
$unsigned_url = 'https://www.operator.com/rdexapi/journeys.json?
timestamp=1326359986&apikey=operator_public_key';
$privateKey = 'operator_private_key';
$signature = hash_hmac('sha256', $unsigned_url, $privatekey);
$signed_url = $unsigned_url.'&signature='.$signature;
```

1.3.Format des données

Les données sont retournées par défaut au format json. On peut toutefois demander à retourner les données au format xml en modifiant l'extension en .xml.

L'utilisation du format json est conseillé.

Exemple :

```
'https://www.operator.com/rdexapi/journeys.xml?
timestamp=1326359986&apikey=operator_public_key&signature=b249c7e56cd8b7c3ca
f2c3333cb446bca5d1c51bd35bf824528daacea9dc4ae6'
```

1.4.Gestion des erreurs

Les erreurs sont basées sur les codes http ([http status code](#)). Elles sont précisées grâce à une réponse en json.

Structure erreur :

Champ	Détails
name	Code d'erreur
message_debug	Message à destination des développeurs pour préciser l'erreur
message_user	Message à destination de l'utilisateur final (en

	cours d'implémentation)
field	Champ sur lequel porte l'erreur

Exemple :

```
{
  "error": {
    "name": "invalid_input",
    "message_debug": "must be a float between -90 and 90",
    "message_user": "An unexpected error occurred. Please try again. If the
problem persists, please contact Covivo.",
    "field": "latitude"
  }
}
```

2 Accès aux ressources

2.1.Méthode *GET journeys*

La ressource « journeys » correspond aux annonces de covoiturage (conducteur ou passager).

La méthode GET journeys permet de récupérer les annonces de covoiturage.

Paramètres de recherche :

Paramètres		Obligatoire	Détails	
driver	state	Oui	Si = 1 on cherche des conducteurs, sinon 0.	
passenger	state	Oui	Si = 1 on cherche des passagers, sinon 0.	
from	longitude	Oui	Longitude de départ (WGS84 degrés décimaux)	
	latitude		Latitude de départ (WGS84 degrés décimaux)	
to	longitude	Oui	Longitude d'arrivée (WGS84 degrés décimaux)	
	latitude		Latitude d'arrivée (WGS84 degrés décimaux)	
frequency		Non	regular/punctual (vide si on ne veut pas restreindre la recherche)	
outward	mindate		Non	Date départ mini
	maxdate			Date départ maxi
	monday	mintime	Non	Heure départ mini
		maxtime		Heure départ maxi
	tuesday	mintime	Non	Heure départ mini
		maxtime		Heure départ maxi
	wednesday	mintime	Non	Heure départ mini
		maxtime		Heure départ maxi
	thursday	mintime	Non	Heure départ mini
		maxtime		Heure départ maxi
	friday	mintime	Non	Heure départ mini
		maxtime		Heure départ maxi
	saturday	mintime	Non	Heure départ mini

		maxtime		Heure départ maxi
	sunday	mintime	Non	Heure départ mini
		maxtime		Heure départ maxi

Exemple :

Récupérer toutes les annonces conducteurs entre 2 points pour une date donnée.

Requête GET envoyée

```
'http://www.operator.com/rdexapi/journeys.json?  
timestamp=14484452833855&apikey=operator_public_key&p[driver]  
[state]=1&p[passenger][state]=0&p[from][latitude]=45.128038&p[from]  
[longitude]=5.587706&p[to][latitude]=45.172348&p[to]  
[longitude]=5.62839&p[outward][mindate]=2015-12-14&p[outward][maxdate]=2015-12-  
14&signature=44f2a639825c3acf5698cb14e0886f1ba1ab1b24c625eec39b42e8d6aa4b3158'
```

Réponse json reçue :

```

"journeys": {
  "uuid": 10945310,
  "operator": "operator",
  "origin": "www.operator.com",
  "url": "http://www.operator.com/65321324",
  "driver": {
    "uuid": 1011327,
    "alias": "jean",
    "image": null,
    "gender": "male",
    "seats": 3,
    "state": 1
  },
  "passenger": {
    "uuid": 1011327,
    "alias": "jean",
    "image": null,
    "gender": "male",
    "persons": 0,
    "state": 0
  },
  "from": {
    "address": "",
    "city": "Lans-en-Vercors",
    "postalcode": "",
    "country": "France",
    "latitude": 45.127976,
    "longitude": 5.58827
  },
  "to": {
    "address": "",
    "city": "Saint-Nizier-du-Moucherotte",
    "postalcode": "",
    "country": "France",
    "latitude": 45.169686,
    "longitude": 5.629263
  },
  "distance": 8998,
  "duration": 725,
  "cost": {
    "fixed": "",
    "variable": 0.08
  },
  "details": "",
  "frequency": "punctual",
  "type": "one-way",
  "days": {
    "monday": 1,
    "tuesday": 0,
    "wednesday": 0,
    "thursday": 0,
    "friday": 0,
    "saturday": 0,
    "sunday": 0
  },
  "outward": {
    "mindate": "2015-12-14",
    "maxdate": "2015-12-14",
    "monday": {
      "mintime": "06:30:00",
      "maxtime": "07:30:00"
    }
  }
}

```

2.2.Méthode *POST connections*

La ressource « connections » permet de contacter un inscrit d'une autre base de données à partir des informations indiquées dans les journeys. L'opérateur destinataire de la requête POST connections se charge d'envoyer un message à l'inscrit concerné.

Paramètres :

Champs		Obligatoire	Détails
operator		Non	Nom de l'opérateur effectuant la requête
origin		Non	Nom du site internet effectuant la requête
driver	uuid	Oui si state = recipient	
	alias	Oui si state = recipient et uuid vide	
	state	Oui	'sender', 'recipient'
passenger	uuid	Oui si state = recipient	
	alias	Oui si state = recipient et uuid vide	
	state	Oui	'sender', 'recipient'
journeys	uuid	Oui	
details		Oui	Champ texte libre contenant le message de contact (500 caractères)

Exemple :

Requête POST sur

```
'http://www.operator.com/restapi/connections.json?
timestamp=14484486514251&apikey=operator_public_key&signature=442264aa0840bb12b
804cca6398ea9947a1d4381f2a66b2c8353ae14492d614d'
```

Contenu du tableau POST

```
$post_data = array (
  'details' => 'Bonjour
                    J\'ai vu votre annonce de covoiturage pour un trajet
Valence / Grenoble, et je souhaiterais rentrer en contact avec vous.
                    Mon email : monsieurtoutlemonde@gmail.com

                    Cordialement',
  'journeys' =>
  array (
    'uuid' => 10945310,
  ),
  'driver' =>
  array (
    'state' => 'recipient',
    'uuid' => 1011327,
  ),
  'passenger' =>
  array (
    'state' => 'sender',
  ),
);
```

Réponse :

Réponse vide (code http 201).

3 Annexes

3.1. Annexe 1 : Structure de journeys

Champs		Obligatoire	Détails
uuid		Oui	ID de la ressource
operator		Oui	Nom de l'opérateur
origin		Oui	Url du site internet
url		Non	Url de l'annonce
driver	uuid	Non si state = 0	
	alias		Pseudo utilisateur
	image		Url de la photo de profil
	seats		Nombre de places disponibles
	state	Oui	Conducteur ? 0 :non 1 :oui
passenger	uuid	Non si state = 0	
	alias		Pseudo utilisateur
	image		Url de la photo de profil
	persons		Nombre de passagers
	state	Oui	Passenger ? 0 : non 1 : oui
from	address	Oui	Le code postal peut être laissé vide
	city		
	postalcode		
	country		
	latitude		
	longitude		
to	address	Oui	Le code postal peut être laissé vide
	city		
	postalcode		
	country		
	latitude		
	longitude		
distance		Oui	Distance en mètres
duration		Oui	Durée en secondes
route		Non	Itinéraire
number_of_waypoints		Non	
waypoints	0 address	Non (sauf si	step_distance : from previous waypoint

	city	number_of_waypoints > 0)		
	postalcode			
	country			
	latitude			
	longitude			
	step_distance			
	step_duration			
	type			pick-up/drop-off
	mandatory	0 : non 1 :oui		
	1	address		
	city ...			
...	...			
cost	fixed	Oui si variable vide	Prix forfaitaire	
	variable	Oui si fixed vide	Prix au kilomètre	
details		Non	Commentaire libre	
vehicle	vehicle_image	Non		
	model			
	color			
frequency		Oui	punctual/regular	
type		Oui	one-way/round-trip	
real_time		Non	0 : non 1 : oui	
stopped		Non	0 : en cours 1 : arrêté	
days	monday	Oui	Si punctual : jour de départ mini Si regular : jours concernés	
	tuesday			
	wednesday			
	thursday			
	friday			
	saturday			
	sunday			
outward	mindate	Oui	Si punctual : date départ mini Si regular : date de validité mini	
	maxdate	Oui	Si punctual : date départ maxi Si regular : date de validité maxi	
	monday	mintime	Oui si lundi	Dans la même journée pour un régulier
		maxtime		
tuesday	mintime	Oui si mardi	Dans la même journée pour un régulier	

		maxtime		
	wednesday	mintime maxtime	Oui si mercredi	Dans la même journée pour un régulier
	thursday	mintime maxtime	Oui si jeudi	Dans la même journée pour un régulier
	friday	mintime maxtime	Oui si vendredi	Dans la même journée pour un régulier
	saturday	mintime maxtime	Oui si samedi	Dans la même journée pour un régulier
	sunday	mintime maxtime	Oui si dimanche	Dans la même journée pour un régulier
return	mindate		Oui si retour	Punctual : date départ mini Regular : date de validité mini
	maxdate		Oui si retour	Punctual : date départ maxi Regular : date de validité maxi
	monday	mintime maxtime	Oui si lundi	Dans la même journée pour un régulier
	tuesday	mintime maxtime	Oui si mardi	Dans la même journée pour un régulier
	wednesday	mintime maxtime	Oui si mercredi	Dans la même journée pour un régulier
	thursday	mintime maxtime	Oui si jeudi	Dans la même journée pour un régulier
	friday	mintime maxtime	Oui si vendredi	Dans la même journée pour un régulier
	saturday	mintime maxtime	Oui si samedi	Dans la même journée pour un régulier
	sunday	mintime maxtime	Oui si dimanche	Dans la même journée pour un régulier

3.2. Annexe 2 : Exemple de journées régulier aller-retour

```
{
  "journeys": {
    "uuid": 11031710,
    "operator": "operator",
    "origin": "www.operator.com",
    "url": "http://www.operator.com/546513256",
    "driver": {
      "uuid": 1011327,
      "alias": "jean",
      "image": "null",
      "gender": "male",
      "seats": 3,
      "state": 1
    },
    "passenger": {
      "uuid": 1011327,
      "alias": "jean",
      "image": "null",
      "gender": "male",
      "persons": 0,
      "state": 0
    }
  },
}
```

```

"from": {
  "address": "",
  "city": "Grenoble",
  "postalcode": "",
  "country": "France",
  "latitude": 45.188529,
  "longitude": 5.724524
},
"to": {
  "address": "",
  "city": "Lyon",
  "postalcode": "",
  "country": "France",
  "latitude": 45.764043,
  "longitude": 4.835659
},
"distance": 114130,
"duration": 4500,
"route": "",
"number_of_waypoints": 0,
"waypoints": "",
"cost": {
  "fixed": "",
  "variable": "0.08"
},
"details": "",
"frequency": "regular",
"type": "round-trip",
"days": {
  "monday": 1,
  "tuesday": 1,
  "wednesday": 1,
  "thursday": 1,
  "friday": 1,
  "saturday": 1,
  "sunday": 1
},
"outward": {
  "mindate": "2015-12-10",
  "maxdate": "2016-12-10",
  "monday": {
    "mintime": "07:30:00",
    "maxtime": "08:30:00"
  },
  "tuesday": {
    "mintime": "07:30:00",
    "maxtime": "08:30:00"
  },
  "wednesday": {
    "mintime": "07:30:00",
    "maxtime": "08:30:00"
  },
  "thursday": {
    "mintime": "07:30:00",
    "maxtime": "08:30:00"
  },
  "friday": {
    "mintime": "07:30:00",
    "maxtime": "08:30:00"
  },
}

```


3.3. Annexe 3 : Liste des codes d'erreur

N°	Code erreur	Code Http	Signification
1	access_denied	401	
2	already_exists	409	
3	insufficient_permissions	403	
4	internal_error	500	
5	invalid_input	400	
6	invalid_uri	400	
7	invalid_uuid	400	
8	missing_mandatory_field	400	
9	missing_required_query_parameter	400	
10	no_post_data	400	
11	no_put_data	400	
12	not_implemented	501	
13	origin_mismatch	403	
14	resource_not_found	404	
15	signature_mismatch	401	
16	timestamp_too_skewed	401	
17	too_late	400	
18	too_many_post	429	
19	too_many_queries	429	
20	undefined_error	400	
21	unknown_user	400	
22	unsupported_http_verb	405	

4 GLOSSAIRE

- **apikey** : clé publique identifiant l'utilisateur de l'api.
- **extension** : ajoutée à la suite du nom d'une ressource, elle spécifie le format dans lequel les résultats seront retournés.
- **p** : tableau de paramètres permettant de spécifier des conditions sur la ressource retournée.
- **privatekey** : clé secrète attribué de façon unique à chaque utilisateur.
- **resource** : élément auquel on souhaite accéder sur le serveur.
- **signature** : Générée à partir de l'url appelée et de la privatekey, elle permet de garantir l'identité de l'utilisateur.
- **timestamp** : nombre de secondes écoulées depuis une date donnée (1er janvier 1970 00:00:00 GMT)