Api RDEX

Documentation

Rédaction: Covivo dans le cadre d'un groupe de travail avec Roulez-Malin et Ecolutis

Table des matières

Modifications apportées:

Nouvelle version	Modifications apportées par rapport à la version précédente.
Spécification1.1.3	* 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 : www.operator.com par le site de l'opérateur concerné ».

1 DESCRIPTION GENERALE

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

NB : La logique voudrait que l'on utilise PUT pour créer une ressource et POST pour modifier une ressource. La méthode POST étant plus couramment utilisée, le choix a été fait d'utiliser POST pour la création d'une ressource et PUT pour la modification.

Une requête à l'api RDEX respecte le format de base suivant :
https://www.operator.com/rdexapi/resource[.extension] ?timestamp=timestamp
&apikey=cléPublique[&p=tableauDeParamètres][&limit=nbMaxDeRésultatsSouhaité]
&signature=signature
Les données entre crochets sont optionnelles.

```
https://www.operator.com/rdexapi/journeys?timestamp=13263599860973
&apikey=s5fg8js9856
&signature=b249c7e56cd8b7c3caf2c3333cb446bca5d1c51bd35bf824528daacea9dc4ae6
```

Remplacer : www.operator.com par le site de l'opérateur concerné.

1.1.1 Méthode GET

Soit 'ress' une ressource.

Soit 'id' un identifiant.

restapi/ress : On lit toutes les ressources 'ress'.

restapi/ress?limit=5 : On lit les 5 premières ressources 'ress'.

restapi/ress/id : On lit la ressource 'res' dont l'identifiant est 'id'.

restapi/ressa/id/ressb : On lit toutes les ressources 'ressb' ayant pour parente la ressource 'ressa' dont l'identifiant est id

1.1.2 Méthode POST

Soit 'ress' une ressource.

restapi/ress + données POST : On ajoute une ressource 'ress'.

1.1.3 Méthode PUT

Soit 'ress' une ressource.

Soit 'id' un identifiant.

restapi/ress/id + données POST : On modifie une ressource 'ress'.

1.1.4 Méthode DELETE

Soit 'ress' une ressource.

Soit 'id' un identifiant.

restapi/ress/id : On supprime une ressource 'ress'.

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 de mail à l'adresse mail 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 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).
sign = sha256(urlNonSignée, privatekey)
urlSignée = urlNonSignée&signature=sign

```
$urlNonSignee = 'https://www.operator.com/rdexapi/journeys.json?
timestamp=13263599860973&apikey=operator2011';
$sign = hash_hmac('sha256', $urlNonSignee, $privatekey);
$urlSignee = $urlNonSignee.'&signature='.$sign;
```

Remplacer : www.operator.com par le site de l'opérateur concerné.

Test de la signature :
La validité de la méthode de génération de la signature peut être testée en effectuant une requête GET à https://www.operator.org/rdexapi/signature avec apiKey = testApiKey et privatekey = testPrivateKey.

```
https:// www.operator.com /rdexapi/signature.json?timestamp
=13263596831637&apikey=testApiKey&signature=78242784bf5de
3ca0c8d184075d95fad91a9fe0d49163cc60c3a6403804663ad
```

Remplacer : www.operator.com par le site de l'opérateur concerné.

Valeur de retour : code erreur http + « signature OK ».

1.2.4 Droits d'accès

Chaque utilisateur se voit attribuer des droits d'accès aux ressources en fonction de ses besoins et des accords passés avec l'opérateur. Il y a un droit d'accès par ressource et par méthode.

1.3 Format des données

Les données sont retournées par défaut au format json. On peut toutefois spécifier un format différent en ajoutant une extension (ex : .xml) ou en spécifiant le HTTP_ACCEPT header ('application/xml'). L'extension ayant la priorité sur le HTTP_ACCEPT header.

1.4 Gestion des erreurs

Les erreurs sont basées sur les codes http (http status code).

Dans le cas d'une erreur du client, le champ « Warning » du header permet de préciser la cause de l'erreur à l'aide d'un code d'erreur personnalisé.

2 ACCES AUX RESSOURCES

2.1 Méthode GET JOURNEYS

La ressource « JOURNEYS » correspond aux offres (trajets conducteur) regroupées par jours de la semaine ou non et afficher ainsi une régularité ou non des trajets: trajets réguliers et aller-retour ou trajet occasionnel.

A noter: On appelle journeyparts la ressource qui correspond à chaque élément d'un journeys: chaque trajet.

restapi/journeys.json + params : On effectue une recherche

Exemples simples:

Pour récupérer tous les journeys au départ ou à l'arrivée autour d'un point dans un rayon (en mètre) entre deux dates:

```
http:// www.operator.com /rdexapi/journeys.json?timestamp=13245714598658
&apikey=operator2011&p[from][latitude]=45.478353&p[from][longitude]=1.22744
&p[radius]=10000&p[outward][mindate]=2012-08-24&p[outward][maxdate]=2012-09-26
&signature=3e6950d08bb9d7f4e9a4f505e2747d73cd692d39
```

Remplacer : www.operator.com par le site de l'opérateur concerné.

Pour récupérer tous les journeys au départ ou à l'arrivée dans une zone géographique entre deux dates (ces requêtes n'utilisent pas la capacité de calcul):

```
http://www.operator.com/rdexapi/journeys.json?timestamp=13245714598658
&apikey=operator2011&p[bounds][southwest][latitude]=34.0523600&p[bounds]
[southwest][longitude]=-118.2435600&p[bounds][northeast][latitude]=41.8781100
&p[bounds][northeast][longitude]=-87.6297900&p[outward][mindate]=2012-08-24
&p[outward][maxdate]=2012-09-26
&signature=3e6950d08bb9d7f4e9a4f505e2747d73cd692d39
```

Remplacer : www.operator.com par le site de l'opérateur concerné.

Pour récupérer tous les journeys dans une zone liée aux deux points à une date donnée:

```
http://www.operator.com/rdexapi/journeys.json?
timestamp=13245726085787&apikey=operator2011
&p[from][latitude]=45.478353&p[from][longitude]=1.22744&p[to][latitude]=45.159555
&p[to][longitude]=1.533937&p[outward][mindate]=2012-08-24
&p[outward][maxdate]=2012-08-24
&signature=33996b6bd358a73b0f86f95ac93f9144c776f819
```

Remplacer : www.operator.com par le site de l'opérateur concerné.

Valeur de retour : code http 200 + tous les journeys correspondant aux critères de recherche.

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 envoie automatiquement un message à l'inscrit à interroger.

rdexapi/connections + données POST : On publie une nouvelle connexion.

Valeur de retour : code http 201 + ID de connections + location.

3 ANNEXES

3.1 Annexe 1 : Structure des JOURNEYS

uuid		Obligatoire	ID de la ressource
operator		Obligatoire	Nom de l'opérateur
origin		Obligatoire	Nom du site internet
logo_supplier		Facultatif	
url		Facultatif	
driver	uuid	Facultatif si state = 0	Mettre 0 si la personne n'est pas inscrite dans la base de données du partenaire
	alias		Identique à alias passenger
	image		Url de la photo de profil
	seats		
	state	Obligatoire	Conducteur ? 0 :non 1 :oui
from	address	Obligatoire	
	city		
	postalcode		
	country		
	latitude		
	longitude		
to	address	Obligatoire	
	city		
	postalcode		
	country		
	latitude		
	longitude		
distance		Obligatoire	Distance en mètres
duration		Obligatoire	Durée en secondes
route		Facultatif	Itinéraire
number_of_waypoints		Obligatoire	

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

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

3.2 Annexe 2 : Structure des CONNECTIONS

uuid		Obligatoire	ID de la ressource
operator		Obligatoire	Nom de l'opérateur
origin		Obligatoire	Nom du site internet
driver	uuid	Obligatoire si state = recipient	
	alias	Obligatoire si state = recipient et uuid vide	
	seats	Obligatoire	Nombre de places disponibles
	state		‘sender’, ‘recipient’
	operator		
	origin		
passenger	uuid	Obligatoire si state = recipient	
	alias	Obligatoire si state = recipient et uuid vide	
	persons	Obligatoire	Nombre de personnes
	state		‘sender’, ‘recipient’
	operator		
	origin		
journeys	uuid	Obligatoire	
details		Obligatoire	Champ texte libre contenant les infos de contact (500 caractères max)

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.

limit : permet de spécifier le nombre maximum de résultats souhaité.

params : 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)