



ANGE GARDIEN DES MOTARDS

API – Interface Emergency

Serveur HTTP haute disponibilité pour les traitements critiques. Les applications mobiles le contactent lorsqu'elles détectent un accident pour déclencher l'intervention des secours. Les traitements non critiques sont à éviter pour éviter que des bugs futiles cassent toute la chaîne.

Error Codes

- Timeout ou erreur : premier retry au bout de
 - 0 secondes, puis retry au bout de
 - 5 secondes, puis retry au bout de
 - 15 secondes, puis retry au bout de
 - 30 secondes, puis retry au bout de
 - 60 secondes, puis retry au bout de
 - 60 secondes, puis retry au bout de
 - 120 secondes

Endpoint /emergency/accident

Add to the header of the POST request : Content-Type: application/json

Input :

- id: string au format UUID
- uid: string
- firstName: string
- lastName: string
- email: string
- phoneNumber: string
- isDuo: boolean // if user is in duo
- isTest: boolean // if accident is a test
- sessionId: string // can also be sent as rideId
- shockId: string au format UUID
- appVersion: string
- appPlatform: string
- clientTime : double // ms
- clientUploadTime : double // ms
- lastLocations: Location[] // Les **10** dernières locations, au moins **1** location. L'ordre est important. Les locations les plus **récentes** doivent être à la **fin**. Le timestamp de location doit être arrondi à la minute.
 - latitude: decimal
 - longitude: decimal
 - accuracy: decimal
 - timestamp: int // ms
- brandName: string ("Kawasaki")
- modelName: string ("ER6")

Response status code: 204

Pas de data en output

Traitements déclenchés :

- Enregistrer serverUploadTime
- Si erreur imprévue, renvoyer 500
- Vérifier l'idToken
- Si erreur id token: renvoyer 401
- Vérifier intégrité des données (bon format, type, pas vide)
- Si erreur de données: renvoyer 400
- Build XML pour IMA
- Check XML pour IMA
- Si erreur : renvoyer 400
- Envoyer le XML à IMA (config staging seulement) CRITICAL
- Envoyer un email sur alarms@liberty-rider.com (pour déclencher la chaine Zapier), (config staging et production seulement), comme dans le worker. CRITICAL
- Contacter healthchecks.io si test=true (config différentes par environnement)
- Renvoyer 204 (succès) au client
- Ensuite, poster les infos sur l'interface algorithme (on fait ça après avoir retourné 204 car la télémétrie n'est pas critique), cf [API - Interface algorithme](#)

Architecture Interne

Liste des services :

- IMA service : build, check, send

- Email service : send
- angel-api service : send
- http service : send response

Tests unitaire :

- Test de chaque service
- Test des controllers

Tests système :

- Teste l'intégralité de la chaîne sous forme de scénarios. Envoi des requêtes http, permet d'émuler le comportement d'un client (mobile, web, etc.).

Environments Variations

Paramètre REST isHealthcheck : Le body de la requête emergency contient le paramètre isHealthcheck=true lorsqu'on veut faire un **test automatique** de bout en bout. Envoyé uniquement par angel-scheduler (toutes les 5 min). Comme l'envoi est très fréquent les données ne sont pas stockées. Donc ça se comporte pas à 100% comme une vraie alerte.

Paramètre REST isTest : Le body de la requête emergency contient le paramètre isTest=true lorsqu'on veut faire un **test manuel** de la chaîne de bout en bout. La seule différence de traitement avec une vraie alerte c'est qu'on transmet le paramètre isTest. C'est donc plus réaliste que isHealthcheck. Le seul moyen de faire tourner ce test est d'utiliser un APK dédié.

Interdit d'envoyer à la fois isTest=1 et isHealthcheck=1.

	NODE_ENV=test (circleci et TU) les services externes mockés en TU			NODE_ENV=development (développeur)		
	isTest=0 isHealth=0	isTest=1 isHealth=0	isTest=0 isHealth=1	isTest=0 isHealth=0	isTest=1 isHealth=0	isTest=0 isHealth=1
IMA						
Email						
healthchecks.io						✓
angel-api				✓	✓	

	NODE_ENV=staging info "staging" ajoutée ?			NODE_ENV=production		
	isTest=0 isHealth=0	isTest=1 isHealth=0	isTest=0 isHealth=1	isTest=0 isHealth=0	isTest=1 isHealth=0	isTest=0 isHealth=1
IMA	✓	✓	✓	2018	2018	2018
Email	✓	✓		✓	✓	
healthchecks.io			✓			✓
angel-api	✓	✓		✓	✓	

IMA : envoi de l'alerte à IMA via une API REST en XML. Ils ont un paramètre "isTest" qui correspond au notre, et un paramètre "supervision" qui correspond à notre isHealthcheck.

NOTE: Actuellement IMA est activé seulement en staging !

NOTE TESTS IMA DEBUTANT 28/03 :

- Création d'une application Android flavor IMA TEST
- Envoi d'un paramètre spécial : isIMATest
- Notifier l'environnement de production IMA des alertes uniquement si l'accident est flaggé isIMATest = true
 - Asana : <https://app.asana.com/0/606735458576959/600688449298686>
 - PR : <https://github.com/liberty-rider/emergency-api/pull/42>

Email : envoi d'un email à un Google group dédié alarms@liberty-rider.com, auquel les fondateurs et Zapier sont inscrits. Zapier déclenche l'envoi d'un message Slack sur **#alertes** et l'appel du téléphone d'alerte. C'est le système qu'on utilisait avant IMA, et on le garde en parallèle.

En staging, l'email arrive sur debug+staging+alarms@liberty-rider.com et ne déclenche pas zapier.

healthchecks.io : envoi à <https://healthchecks.io/checks/>, un service qui nous alerte sur Slack dans le canal #monitoring lorsque le healthcheck est en retard par rapport à l'intervalle de 5 minutes qui est prévu.

angel-api : envoi des détails de l'alerte à notre API pour être stocké dans notre base de donnée, et affiché sur la carte de la session.