

# [NOM DU PROJET] DOCUMENTATION SERVEUR NODE

MAJ : 03/01/2018

## Table des matières

CONNEXION .....	2
auth_attempt.....	2
sign_up_attempt.....	2
auth_success.....	2
auth_failed .....	2
registration_success.....	2
error.....	3
node_error .....	3
registration_failed .....	3
FAMILIES.....	3
request_family .....	3
select_family.....	3
switch_family .....	3
new_family.....	4
add_family_to_user .....	4
request_family_reply .....	4
error.....	4
PROFIL .....	5
request_profile.....	5
update_user_profil.....	5
request_profile_reply .....	5
error .....	5
CHAT .....	6
new_message .....	6
load_messages .....	6
new_message_available .....	6
load_messages_reply.....	6



# CONNEXION

## EVENEMENTS ENTRANT

### auth\_attempt

**param:** JSON { 'email': "", 'password': "" }

**reply :** EVENT auth\_success  
EVENT node\_error  
EVENT auth\_failed

*Evènement à émettre pour vérifier l'authentification d'un utilisateur.*

### sign\_up\_attempt

**param:** JSON { 'email': "", 'password': "", 'name': "", 'surname': "", 'address': "", 'cp': "", 'city': "", 'country': "", 'birthday': "" }

**reply :** EVENT registration\_success  
EVENT node\_error  
EVENT registration\_failed

*Evènement à émettre pour procéder à l'enregistrement d'un nouvel utilisateur.*

## EVENEMENTS SORTANT

### auth\_success

**param:** Aucun

**reply :** JSON { 'token': }

*Evènement émis quand la connexion est valide. Si l'utilisateur est présent dans la base de données JEE, le serveur NODE renvoi un token utilisé pour l'authentification. Ce token doit être utilisé pour toutes les autres requêtes.*

### auth\_failed

**param:** Aucun

**reply :** <STRING> Error

*Evènement émis quand l'authentification a échoué. Dans le cas de la connexion, les données ne sont pas bonnes ou n'existent pas en base de données. Pour le reste, le token est invalide.*

### registration\_success

**param:** Aucun

**reply :** Aucune

*Evènement émis quand la création d'un nouvel utilisateur est valide et enregistrées dans les bases de données JEE et MongoDB.*



## error

**param:** Aucun

**reply :** <STRING> Error

*Evènement émis quand une erreur survient dans le serveur NODE.*

## node\_error

**param:** Aucun

**reply :** <STRING> Error

*Evènement émis quand la connexion avec le serveur JEE est impossible.*

## registration\_failed

**param:** Aucun

**reply :** <STRING> Error

*Evènement émis quand l'enregistrement d'un nouvel utilisateur a échoué.*

# FAMILIES

## EVENEMENTS ENTRANT

### request\_family

**param:** JSON { 'token': "", 'email': "" }

**reply :** EVENT request\_family\_reply  
EVENT error

*Evènement à émettre pour obtenir la liste des familles associées à un utilisateur.*

### select\_family

**param:** JSON { 'token': "", 'code': "" }

**reply :** EVENT error

*Evènement à émettre pour sauvegarder le choix de l'utilisateur sur la famille à afficher. Le code correspondant à la famille choisie est enregistré dans une map avec la socket.*

### switch\_family

**param:** JSON { 'token':, 'email':, 'current\_family\_id':, 'next\_family\_id' }

current\_family\_id : id de la famille affichée.  
next\_family\_id : id de la famille à afficher.



**reply :** EVENT error

*Evènement à émettre pour changer la famille enregistré en mémoire pour un utilisateur.*

### new\_family

**param:** JSON { 'token':, 'email':, 'family': }

**reply :** EVENT error

*Evènement à émettre pour créer une nouvelle famille. La famille est automatiquement associée à l'utilisateur référencé par son email.*

### add\_family\_to\_user

**param:** JSON { 'token':, 'email':, 'code' }

**reply :** EVENT error

*Evènement à émettre pour associer une famille a un utilisateur.*

## EVENEMENTS SORTANT

### request\_family\_reply

**param:** Aucun

**reply :** JSON { 'email' : , 'families' : [] }

Families : <Array> Contient tous les JSON de familles associées à un utilisateur.

*Evènement émis après une demande de familles associées à un utilisateur.*

### error

**param:** Aucun

**reply :** <STRING> Error

*Evènement émis quand une erreur survient dans le serveur NODE.*



# PROFIL

## EVENEMENTS ENTRANT

### request\_profile

**param:** JSON {'token': "", 'email': ""}

**reply :** EVENT request\_profile\_reply  
EVENT error

*Evènement à émettre pour obtenir les informations du profil d'un utilisateur.*

### update\_user\_profil

**param:** JSON {'token': "", 'email': "", 'profile':{'cp':'','city':''}}

*Le champ profile doit contenir tous les changements sur le profil associé à l'email.*

**reply :** EVENT error

*Evènement à émettre pour mettre à jour un profil utilisateur.*

## EVENEMENTS SORTANT

### request\_profile\_reply

**param:** Aucun

**reply :** JSON { 'email': "", 'name': "", 'surname': "", 'address': "", 'cp': "", 'city': "", 'country': "", 'birthday': "" }

*Evènement émis après une requête demandant le profil d'un utilisateur.*

### error

**param:** Aucun

**reply :** <STRING> Error

*Evènement émis quand une erreur survient dans le serveur NODE.*



# CHAT

## EVENEMENTS ENTRANT

### new\_message

**param:** JSON {'token': "", 'msg': ""}

Msg : JSON {code: , user:, date: , content: }

**reply :** EVENT new\_message\_available  
EVENT error

*Evènement à émettre lors d'un nouveau message.*

### load\_messages

**param:** JSON {'token': "", 'code': ""}

code : Code of the current family

**reply :** EVENT load\_messages\_reply  
EVENT error

*Evènement à émettre pour charger tous les messages relatifs à une famille.*

## EVENEMENTS SORTANT

### new\_message\_available

**param:** Aucun

**reply :** JSON Message {code: , user:, date: , content: }

*Evènement émis à toutes les sockets authentifiées. L'évènement contient le nouveau message. Cet événement est émis après l'émission d'un nouveau message.*

### load\_messages\_reply

**param:** Aucun

**reply :** <Array> Messages

*Evènement émis après une demande de chargement des messages. Cet événement renvoi un tableau contenant tous les messages d'un chat de famille.*

