

## Routes :

Default config: <http://localhost:2299>

## Entrées API

*params which must be set\**

*// = or*

*params are stored on req.body*

### POST /Message

*idUser\* , [ idChatLog || nameChatLog ]\* , isVisible , message , color , effect , font , pseudo , password*

### GET /Messages

*[ idChatLog || nameChatLog ]\* , password*

### GET /Message

*[ idChatLog || nameChatLog ]\* || nameChatLog\* , password*

### UPDATE /Message

*idMessage\* , [ idChatLog || nameChatLog ]\* , idUser\* , password , isVisible , message , color , effect , font*

### DELETE /Message

*idMessage\* , [ idChatLog || nameChatLog ]\* , idUser\* , password*

### POST /Chatlog

*[ idChatLog || nameChatLog ]\* , idUser\* , hasPassword , password , isVisible , name , description , color , effect , font , icon , colorMessages , effectMessages , fontMessages*

### GET /Chatlogs

*(Nothing)*

### GET /Chatlog

*[ idChatLog || nameChatLog ]\**

### UPDATE /Chatlog

*[ idChatLog || nameChatLog ]\* , idUser\* , hasPassword , password , isVisible , name , description , color , effect , font , icon , colorMessages , effectMessages , fontMessages*

### DELETE /Chatlog

*[ idChatLog || nameChatLog ]\* , idUser\* ,*

# Sorties API

*params which must be set\**

*// = or*

*params are stored on req.body*

## POST /Message

```
{
  "idMessage": string / number
  "version": string
  "idUser": string / number
  "isVisible": bool
  "idChatlog": string
  "pseudo": string
  "color": string format : "#123456"
  "effect": string,
  "font": string,
  "message": string
  "timestamp": int
}
```

## GET /Messages

```
[
  {
    "idMessage": "20",
    "idMessage": string / number
    "version": string
    "idUser": string / number
    "isVisible": bool
    "idChatlog": string
    "pseudo": string
    "color": string format : "#123456"
    "effect": string,
    "font": string,
    "message": string
    "timestamp": int
  },
  ...
]
```

## GET /Message

```
{
  "idMessage": string / number
  "version": string
  "idUser": string / number
  "isVisible": bool
  "idChatlog": string
  "pseudo": string
  "color": string format : "#123456"
  "effect": string,
  "font": string,
  "message": string
  "timestamp": int
}
```

## UPDATE /Message

*Comme Le POST /Message et Le GET /Message - L'objet modifié est retourné*

## DELETE /Message

*Comme Le POST /Message et Le GET /Message - L'objet retiré est retourné*

## POST /Chatlog

```
{
  "idChatlog": string / integer,
  "version": string
  "idUser": string / integer,
  "isVisible": bool,
  "name": string,
  "description": string
  "color": string    format :"#123456"
  "effect": string
  "font": string
  "icon": string
  "colorMessages": string    format :"#123456"
  "effectMessages": string
  "fontMessages": string
  "hasPassword": bool
  "password": string
  "timestamp": integer
}
```

## GET /Chatlogs

```
[
  {
    "idChatlog": string / integer,
    "version": string
    "idUser": string / integer,
    "isVisible": bool,
    "name": string,
    "description": string
    "color": string    format :"#123456"
    "effect": string
    "font": string
    "icon": string
    "colorMessages": string    format :"#123456"
    "effectMessages": string
    "fontMessages": string
    "hasPassword": bool
    "password": string
    "timestamp": integer
  },
  ...
]
```

## GET /Chatlog

```
{
  "idChatlog": string / integer,
  "version": string
  "idUser": string / integer,
  "isVisible": bool,
  "name": string,
  "description": string
  "color": string    format :"#123456"
  "effect": string
  "font": string
  "icon": string
  "colorMessages": string    format :"#123456"
  "effectMessages": string
  "fontMessages": string
  "hasPassword": bool
  "password": string
  "timestamp": integer
}
```

## UPDATE /Chatlog

Comme Le POST /ChatLog et Le GET /ChatLog - L'objet modifié est retourné

## DELETE /Chatlog

Comme Le POST /ChatLog et Le GET /ChatLog - L'objet retiré est retourné

La configuration globale du projet est dans ./env

Il y a 2 fichiers de conf, la version local (dev) & la version prod

Pour écrire dans les fichiers texte, on utilise les variables d'environnements afin de « bloquer » l'accès de l'API durant l'écriture, peu importe où on se trouve.

POUR EXECUTER LE PROJET

**npm run dev**

(ou "npm run prod")

(((((((todo SOMEONE\_WRITING\_ON\_TXTFILE=false prendre le cas des 2 fichiers séparément)))))))