

## Présentation et installation de Telegram

Telegram est une application de messagerie sur Smartphone ou ordinateur.

Un Bot Telegram est une application qui tourne dans Telegram, et les utilisateurs peuvent interagir avec les bots en envoyant des messages, des commandes ou des requêtes.

Une palette Telegram est proposée sous Node Red pour gérer les messages (envoi, réception...).

### Installer Telegram sur son Smartphone et configurer un Bot

⇒ Installer Telegram

⇒ Faire une recherche de BotFather puis démarrer

⇒ Taper sur la commande /start (ou envoyer la commande /start)

NB : les commandes commencent par /

⇒ Créer un nouveau Bot en tapant la commande : /newbot

⇒ Donner un nom, par exemple : Maintenance

⇒ Donner un nom utilisateur qui termine par Bot , exemple : DupontJeanBot

NB : Le nom utilisateur doit être unique.

⇒ Récupérer le « token » pour pouvoir établir la communication (le token peut être transmis par mail par exemple pour être récupérer sur le PC)

### Installation sur NODE RED

Sous Node Red

⇒ Vérifier la présence de la palette Telegram



The image shows a screenshot of the Node-RED interface. On the left, there is a text instruction: "⇒ Vérifier la présence de la palette Telegram". On the right, a palette titled "telegram" is expanded, displaying six nodes: "receiver", "command", "event", "sender", "reply", and "control". Each node is represented by a blue rectangular block with a white arrow icon on the left and a small square handle on the right.

### Réception de message :

⇒ Réaliser le FLOW ci-contre



The image shows a screenshot of a Node-RED flow. On the left, there is a text instruction: "⇒ Réaliser le FLOW ci-contre". On the right, a flow is visible on a grid background. It consists of two nodes: a blue "Telegram receiver" node on the left and a green "msg.payload" node on the right. A grey line connects the output of the "Telegram receiver" node to the input of the "msg.payload" node. Below the "Telegram receiver" node, there is a small green square icon and the text "connected".

### ⇒ Paramétrer le compte Bot Telegram

Bot

Name

Download Directory

### ⚙ Properties

Bot-Name

Token

**Tip:** If you don't have a token yet, you can create a new [@BotFather](#).

---

Users

ChatIds

Server URL

⇒ Dans le menu principal de Telegram (sur Smartphone), faire une recherche du nom du bot créé (dans l'exemple ici c'est Maintenance).

⇒ Envoyer la commande /start puis envoyer un message (Test par exemple)

⇒ Retrouver les informations reçues sous Node Red, dans la fenêtre DEBUG

```
02/01/2023 17:40:32 node: 81c483d0.e8d898
```

```
msg.payload : Object
```

```
▼ object
```

```
chatId: 5932801470
```

```
messageId: 1
```

```
type: "message"
```

```
content: "/start"
```

```
date: 1672677633
```

```
02/01/2023 17:40:49 node: 81c483d0.e8d898
```

```
msg.payload : Object
```

```
▼ object
```

```
chatId: 5932801470
```

```
messageId: 2
```

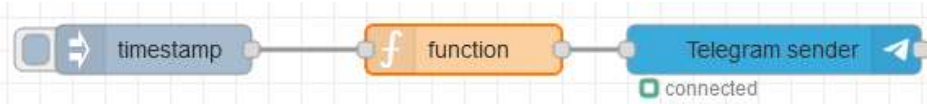
```
type: "message"
```

```
content: "Test"
```

```
date: 1672677650
```

## Envoi de message

⇒ Ajouter le FLOW suivant au précédent



⇒ Editer la fonction de la manière suivante, en remplaçant le **chatId** par celui reçu précédemment.

### Edit function node

Delete

⚙️ Properties

Name: Name

⚙️ Setup   On Start   **On Message**

```
1 msg.payload={
2   "content" : "test",
3   "type" : "message",
4   "chatId":5932801470
5 }
6 return msg;
```

Le Telegram sender est paramétré de la même manière que pour la réception.

Le node timestamp est laissé avec ses valeurs par défaut

⇒ Vérifier l'envoi des messages sur le Smartphone à chaque appui sur le bouton du node timestamp.

NB : Les notifications peuvent être validées pour un affichage sur l'écran d'accueil du Smartphone.

### Exercice :

On souhaite extraire la valeur de la température d'un fichier au format JSON et l'envoyer sur la messagerie Telegram.

Le test s'effectuera avec le node inject pour simuler la réception d'un fichier JSON.

Réaliser le flow suivant (chatId à modifier):

The image shows a Node-RED flow with three nodes: an inject node named 'Temperature', a function node, and a Telegram sender node. Below the flow are two configuration panels.

**Edit inject node**

Delete Cancel

Properties

Name: Temperature

msg.payload: {"temperature":19}

msg.topic: a\_z

**Edit function node**

Delete

Properties

Name: Name

Setup On Start On Me

```
1 msg.payload={
2   "content" : msg.payload.temperature,
3   "type" : "message",
4   "chatId":5328401316
5 }
6 return msg;
```

⇒ Tester la réception sur Smartphone

⇒ Modifier la fonction de la manière suivante et tester (chatId à modifier)

```
Setup On Start On Message
1 msg.payload={
2   "content" : "Température actuelle: "
3   + msg.payload.temperature
4   + "°C",
5   "type" : "message",
6   "chatId":5328401316
7 }
8 return msg;
```

⇒ Modifier la fonction de la manière suivante. Analyser le comportement de la fonction et tester en modifiant le fichier JSON injecté.

```
Setup On Start On Message
1 if (msg.payload.temperature>30){
2   msg.payload={
3     "content" : "Température actuelle: "
4     + msg.payload.temperature
5     + "°C",
6     "type" : "message",
7     "chatId":5328401316
8   }
9   return msg;
10 }
11 else return null;
```

⇒ Modifier le message envoyé pour afficher  
Température dépassée  
Température actuelle :  °C

NB : le caractère \n permet d'ajouter un saut de ligne

On souhaite bloquer la transmission à 3 messages au maximum lorsque que la température de 30°C est dépassée.

⇒ Modifier la fonction de la manière suivante :

⚙️ Setup	On Start	On Message
----------	----------	------------

```
1 var ack= flow.get("ACQUIT") || 0;
2 if (msg.payload.temperature>30 && ack<3){
3     msg.payload={
4         "content" : "Température dépassée\n"
5         + "Température actuelle: "
6         + msg.payload.temperature
7         + "°C",
8         "type" : "message",
9         "chatId":5328401316
10    }
11    ack++;
12    flow.set("ACQUIT",ack);
13    return msg;
14 }
15 else return null;
```

Explications :

- Sous NODE RED il est possible de déclarer des variables « globales » de 2 types : soit de type flow et ces variables sont reconnues dans le flow (une page ) ou soit de type global et ces variables sont reconnues dans tous les flows (plusieurs pages).
  - L’instruction flow.get(« ACQUIT » ) permet de retourner la valeur de la variable ACQUIT qui est de type flow (donc reconnue dans tous les nodes du même flow).
  - L’instruction flow.set(« ACQUIT », 0) permet de mettre la variable ACQUIT à la valeur de 0.
  - L’instruction var ack permet de déclarer une variable appelée ack (var est une instruction javascript). La variable ack est ici reconnue dans le node function uniquement.
  - L’instruction var ack = flow.get(“ACQUIT”) || 0; déclare une variable ack qui sera initialisée avec la valeur de la variable ACQUIT ou avec la valeur 0 la première fois (tant que la valeur de ACQUIT n’a pas été définie). (en javascript ou en C++ , | correspond à un OU).
- L’intérêt de passer par une variable globale de type flow (ou global) c’est que cette variable peut être modifiée ou lue par un autre node.

⇒ Tester et vérifier le nombre maximum de messages envoyé.

On souhaite remettre à 0 la variable ACQUIT lorsqu’on envoie, à l’aide du smartphone, le message « /acquittement ».

⇒ Ajouter la programmation suivante et tester. Faire constater le fonctionnement.

	<p>L’algorithme de la fonction :</p> <p>SI le contenu du message reçu = « /acquittement » Mettre la variable ACQUI à 0 FIN SI</p>
---	---