# A- Introduction :

Ce document définit les spécificités techniques du GUI servant à configurer les networks servers.

# B- Les fonctions :

## 1. conversion :

### 1.1 Description :

Sert à créer le formulaire en html selon l’index de l’élément choisit dans la listBox. Sert uniquement dans l’onglet Ajouter.

### 1.2 Fonctionnement :

Tel = index de l’élément sélectionné dans la listBox.

Ainsi selon cet index la fonction créer un formulaire. A chaque création n prends une valeur, ainsi le formulaire 1 correspondant n’est pas recréer si n est égale à 1, évite ainsi toute erreur utilisateur en ne recréant pas le même formulaire et évite perte de saisie. Se lance lors du clic sur le bouton suivant.

## 2. preload :

### 2.1 Description :

Sert à pré remplir les cases du formulaire existant.

### 2.2 Fonctionnement :

indexSelected = index de l’élément sélectionné dans la listBox.

Ainsi selon cet index la fonction remplit le formulaire. Fonctionne selon l’index grâce à un switch. Se lance lors du clic sur le bouton preload.

## 3. disableField :

### 3.1 Description :

Sert à bloquer la case NSCONFT lorsque l’utilisateur saisie quelque chose dans la case NSCONFB.

### 3.2 Fonctionnement :

Test si la longueur de la valeur de la case est plus grande que 0, si oui bloque la case NSCONFT

## 4. Suppression :

### 4.1Description :

Même fonctionnement et utilité que conversion, sauf que là c’est uniquement pour les delete.

## 5. Visual :

### 5.1 Description :

Même fonctionnement et utilité que conversion, sauf que là c’est uniquement pour l’onglet Aperçu.

### 5.2 Fonctionnement :

Petite subtilité, selon l’index sélectionné, définit méthode et cible à chaque cas, et à la fin lance une requête makeCorsRequest.

## 6. Add (AddBr, AddGw, AddDv, AddNs, AddAsa) :

### 6.1 Description :

Ces fonctions servent à récupérer les données saisies par l’utilisateur dans un objet, avec quelques vérifications, et de lancer la requête schéma. Se lance lors du clic sur le bouton Terminer la configuration (sauf formulaire gw, RecModif sera lancé).

AddBr -> Broker

AddGw -> Gateway

AddDv -> Device

AddNs -> Network server (Oam, Cam et NsContr)

AddAsa -> AsaRouter

### 6.2 Fonctionnement :

Pour cela cette fonction créer un objet et ensuite récupère toutes les valeurs en les mettant une à une dans une propriété de l’objet, ces propriétés sont nommées selon le Schema Json. Enfin la fonction fait appel à la fonction schema (cf schema), avec comme paramètres :

- méthode (POST, GET, LIST,…),

- NSNOW (correspondant à l’url du Ns en cours),

- ’’ADD’’ correspondant au type dans makeCorsRequest,

- cible (mqttBroker, gateway,…), idd (l’objet précédemment créé),

- url (l’url du schéma correspondant à la requête).

Le for sert à récupérer les potentiels donnés de channels et paramètres et les rajouter à l’objet (via tableau d’objet pour les channels)

## 7. Ajout (1, 2, et 3) :

### 7.1 Description :

Ces fonctions servent à ajouter des channels/paramètres au formulaire existant lors du clic sur le bouton ajouter un channel ou ajouter un paramètre.

### 7.2 Fonctionnement :

Chaque fonction a son indice, à chaque appuie sur le bouton ajoute une ligne html avec comme identifiant le nom de la case + l’indice, pour ainsi pouvoir les récupérés précisément lors de la fonction Add. L’indice permet aussi de bloqué le nombre maximum d’ajout dans certains cas.

Ajout1 -> Gateway

Ajout2 -> Device

Ajout3 -> NetworkServer

## 8. RecModif :

### 8.1 Description :

Cette fonction sert à vérifier avant comparaison au schéma si centerFrequency est bien remplit si ASYNCHRONOUS est sélectionné. Lance AddGw si c’est le cas.

### 8.2 Fonctionnement :

Récupère d’abord la valeur sélectionné de la listbox mode, ensuite vérifie la condition énoncée dans la description, et si concluant lance AddGw, sinon avertie l’utilisateur, se lance lors du clic sur Terminer la configuration du formulaire Gw.

## 9. Delete :

### 9.1 Description :

### Cette fonction a pour but de récupérer les valeurs entrées par l'utilisateur dans le formulaire supprimer et d'envoyer la requête de delete.

### 9.2 Fonctionnement :

En fonction de l’index sélectionné de la listBox dans un switch, récupère les données et les mets dans un objet puis dans idd, change la méthode, puis après le switch lance makeCorseRequest avec les paramètres adaptés. Se lance lors de l’appuie sur le bouton Supprimer! de l’onglet Supprimer.

## 10. ajoutNameValue :

### 10.1 Description :

### Cette fonction sert dans la fonction AddNs, elle sert à récupérer les données en fonction du paramètre, de les mettre dans un object et de retourner celui-ci

### 10.2 Fonctionnement :

En fonction du paramètre renseigner, la fonction va chercher la valeur et va la mettre dans la propriété value de l’objet, dans quelconque cas de truststore, keyfile, etc, modifie la valeur name(paramètre renseigné) en ce qui est attendu dans les schémas (ex : truststoreFileOam devient truststoreFile). Ensuite retourne l’objet.

## 11. restrictCharacters :

### 11.1 Description :

Cette fonction sert à assurer que l’utilisateur rentre un int conformément au schéma.

### 11.2 Fonctionnement :

Cette fonction n’autorise donc que les chiffres et un seul . dans la chaine.

## 12. CreateCORSRequest :

### 12.1 Description :

Cette fonction instancie une requête CORS

### 12.2 Fonctionnement :

Elle crée une variable xhr qu’elle définit au type d’envoi de la requête.

## 13. MakeCORSRequest :

### 13.1 Description :

Cette fonction récupère toutes les données qui lui sont mises à disposition (URI du NS, Type de méthode, cible d’envoi, type d’envoi..) et envoie la requête CORS qu’elle doit envoyer en fonction de ces paramètres.

### 13.2 Fonctionnement :

Elle fonctionne par étapes, dans un premier temps elle vient récupérer l’URI, la cible et l’idd de requêtes qui lui a été fourni. Dans le cas où l’idd est null elle génère un IDD aléatoire.

Cet IID sert de nom à la requête, toutes les informations la concernant serons désormais transmis entre User & Server sous cette forme.

L’URI est le lien « générique » du Network server qui a été entré par l’utilisateur lors de la connexion à ce dernier.

Idds est une variable globale ayant été définie auparavant qui définit l’idd à rentrer DANS le lien.

Tous ces paramètres sont dans un premier temps rassemblés pour créer l’URI finale d’envoi de la requête

Dans un second temps la requête va étudier les méthodes et en fonction du type et de la méthode elle va l’attribuer à un ou plusieurs cas du switch qui effectuera des actions en fonction.

## 14. scookie :

### 14.1 Description :

La fonction scookie a pour but

### 14.2 Fonctionnement :