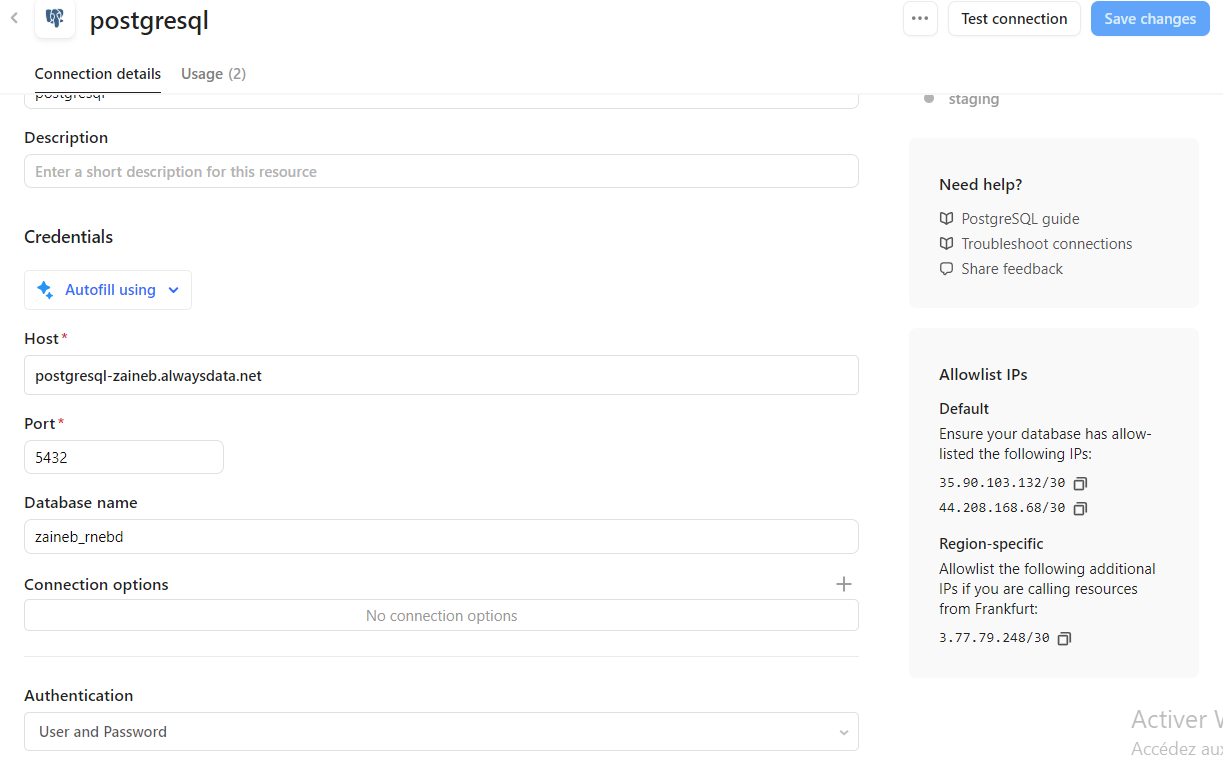
**Solution Low Code :Retool**

**Definition :**Retool est une plateforme de développement qui permet de créer rapidement des applications internes personnalisées en utilisant des composants prédéfinis et en intégrant facilement des bases de données et des API **Connexion avec la base de donnée externe :**  
Pour connecter Retool à une base de données PostgreSQL hébergée sur un serveur distant, tel que alwaysData, commencez par héberger votre base de données PostgreSQL sur alwaysData. Ensuite, configurez le fichier application.properties de votre application Spring Boot pour établir la connexion :  
1)Aller Ressource 🡪 Create New Ressource 🡪Ajouter la configuration de la base de donnée PostgreSQL hébergée

2)Tester Connexion 🡪Save Change

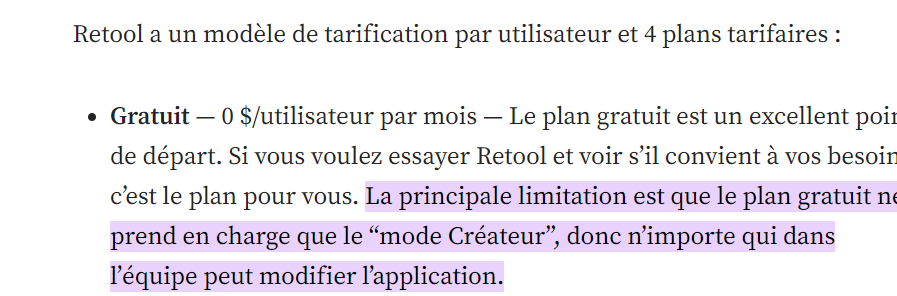
Configuration dans Retool :

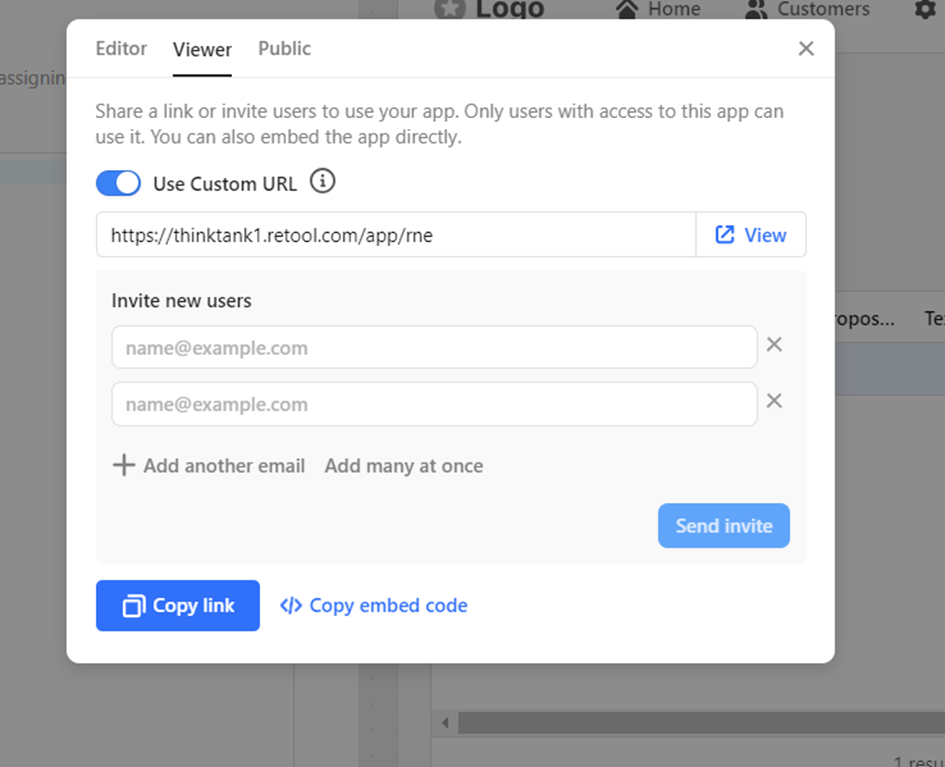


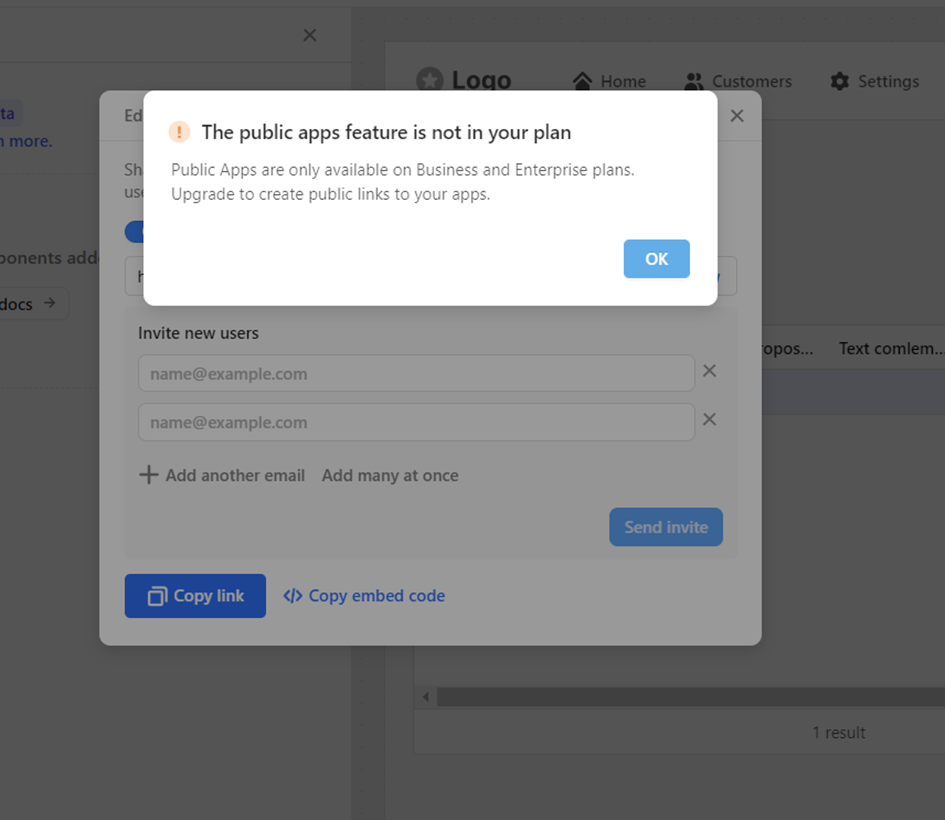
Partie Spring Boot :  


**Integration de Retool avec Angular :**

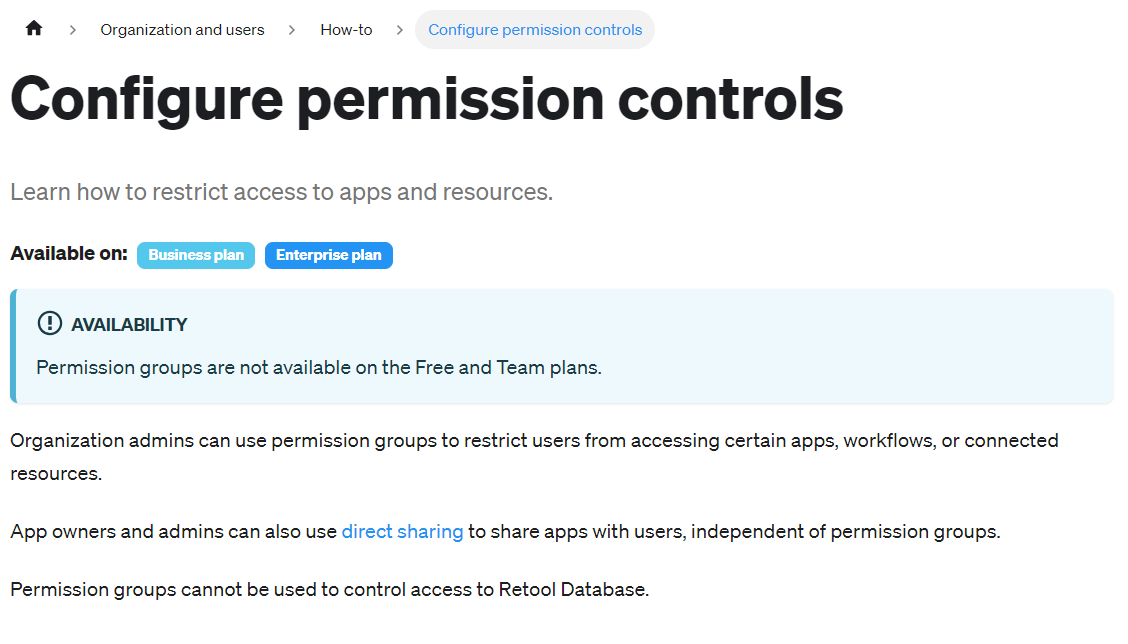
Après avoir connecté votre application Retool à Angular, vous obtenez un éditeur interactif. Cela signifie que toute personne ayant accès à votre compte Retool peut modifier le contenu de votre application. Cependant, après des recherches, j'ai découvert que si vous souhaitez fournir une URL permettant à un utilisateur simple d'accéder à votre plateforme sans le bouton d'édition, cette fonctionnalité est disponible dans le plan Entreprise de Retool. Cette option permet de sécuriser votre application en limitant les droits d'édition uniquement aux utilisateurs autorisés, tout en offrant une expérience utilisateur fluide pour les utilisateurs finaux.





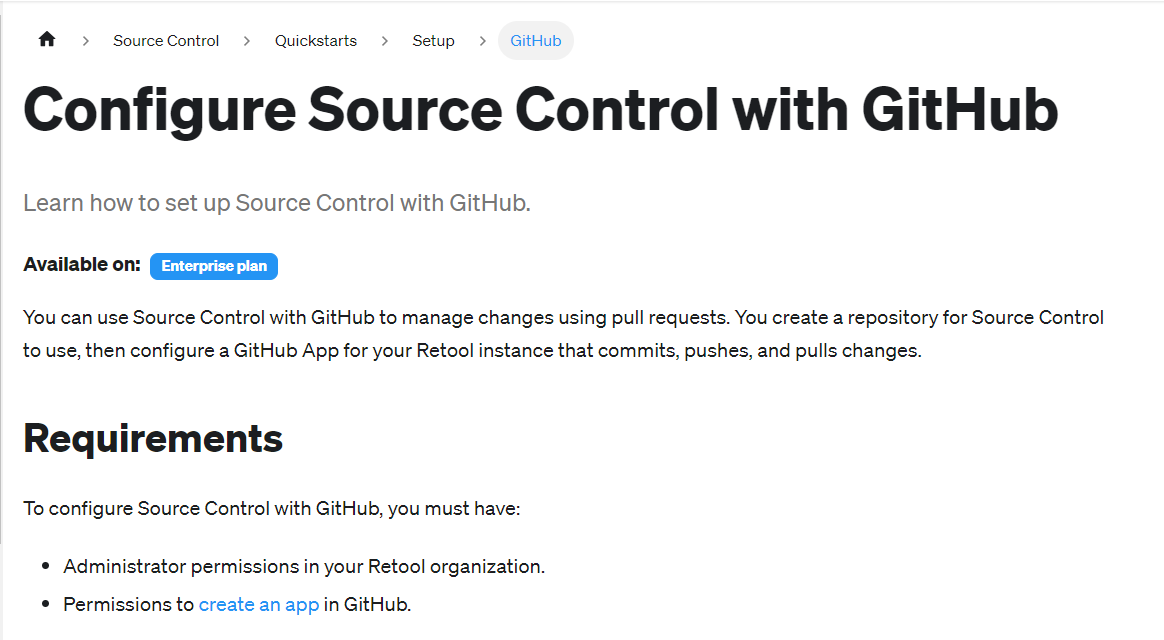


**Configuration des contrôles d'autorisation**



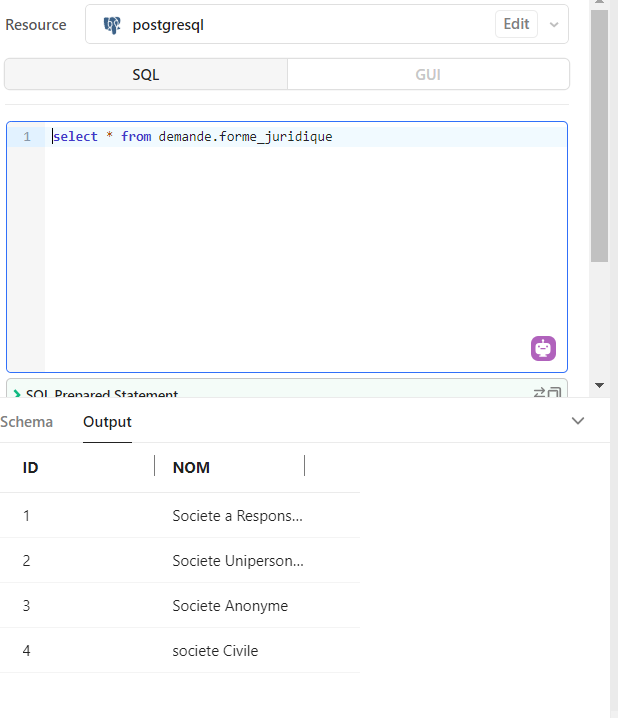
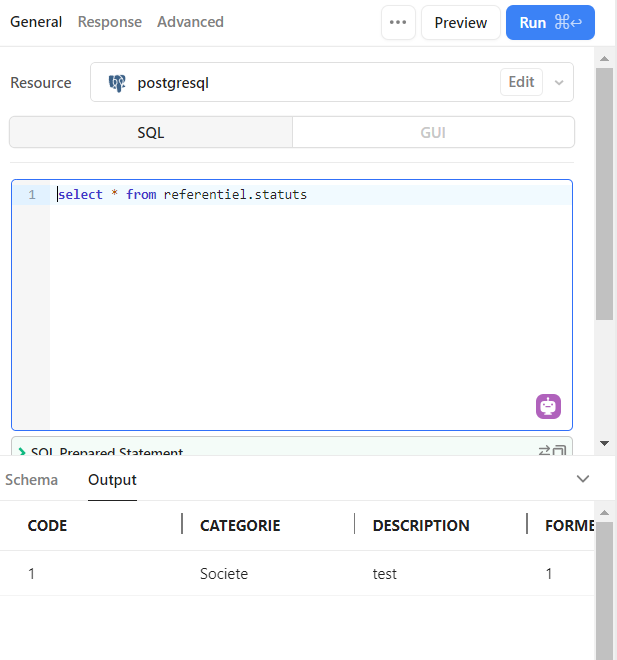
**Git avec Retool**

le plan Entreprise de Retool inclut également l'intégration avec Git. Cette fonctionnalité permet de versionner et de gérer le code source de vos applications Retool, facilitant la collaboration entre les développeurs et le suivi des modifications. Grâce à cette intégration, vous pouvez synchroniser votre application Retool avec votre dépôt Git, assurant ainsi une gestion de version cohérente et une meilleure organisation du développement.

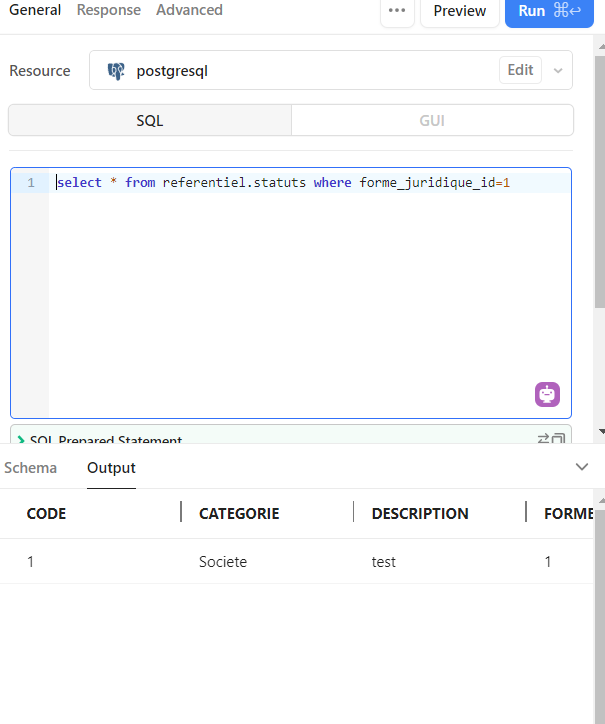


Cas d’utilisation:

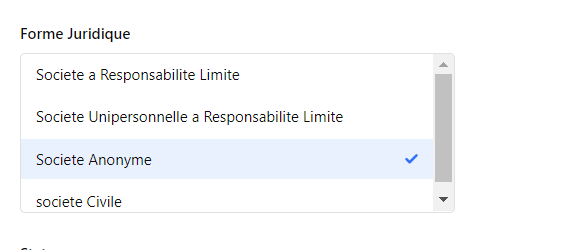
1) Récupérer les deux entités Forme Juridique et Status :

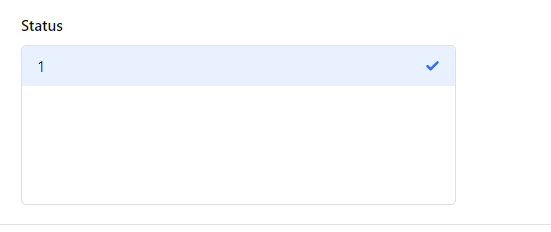
 

2) Récupérer Status By Forme Juridique :



3) Récupérer list des Forme Juridique dans un listBox :



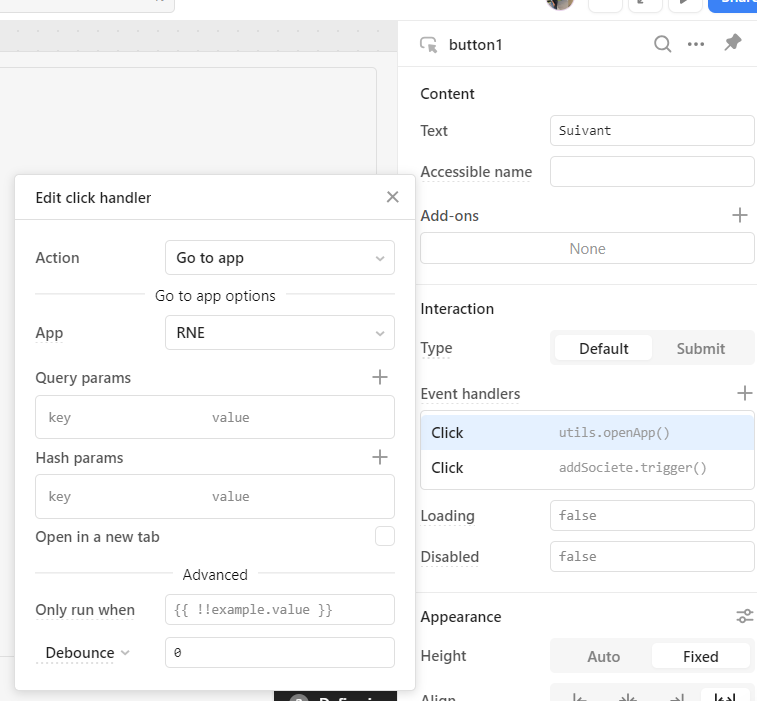
4) Récupérer list des status By Forme Juridique dans un listBox :  


5)Faire la dependance entre ces deux listBox :

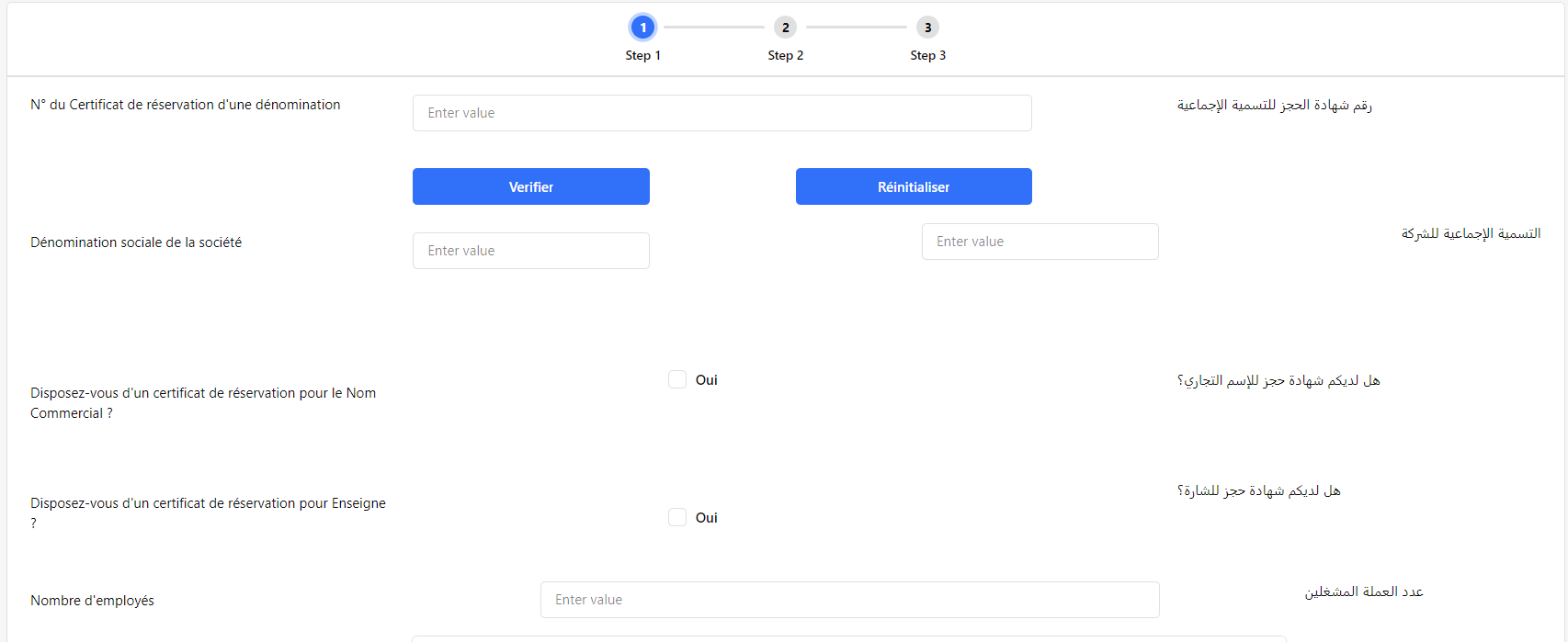
-Aller a la premiere listBox FormeJuridique

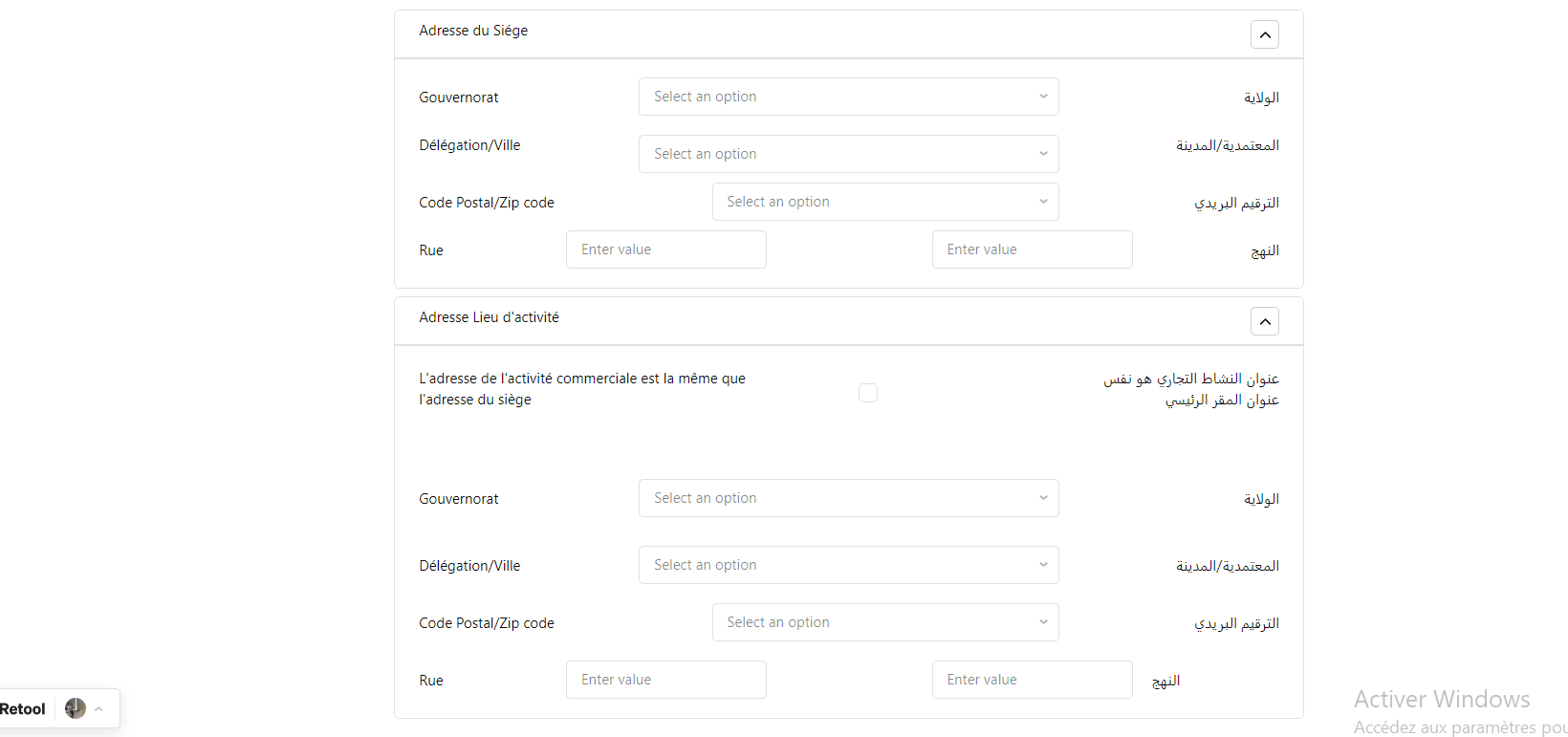
-Cliquer Ajouter Event :

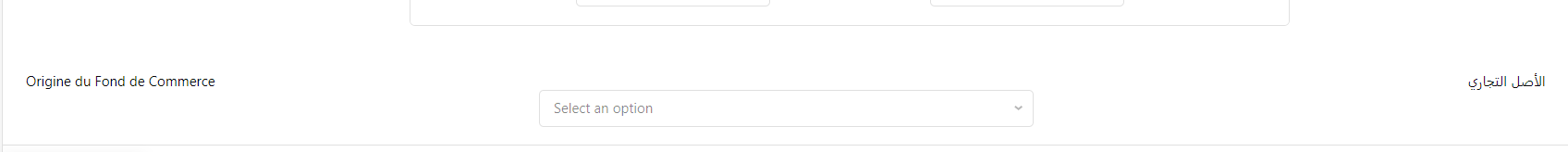
* Event :Change
* Action :query
* Query :getStatusByFormeJuridique

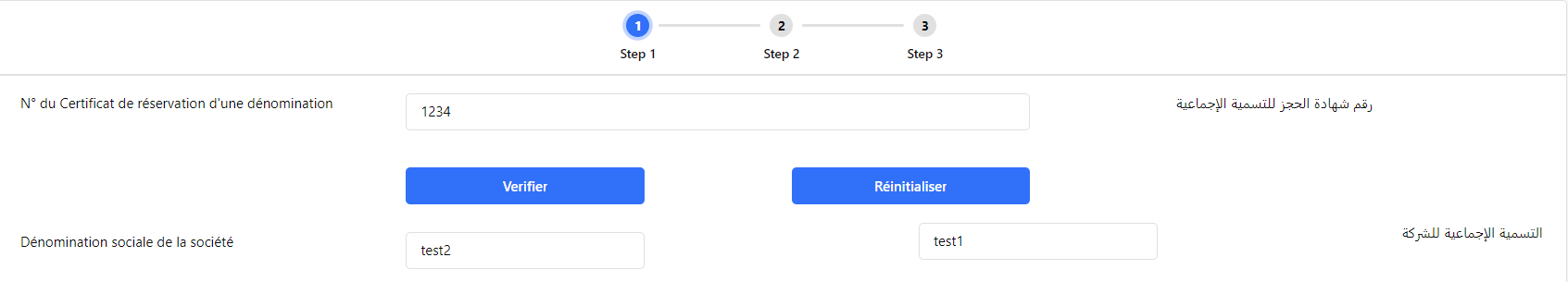
**Naviguer d’une page a une autre :**  


1)Button Suivant 🡪Event handlers 🡪 click 🡪 Action 🡪 Go to app

Step1:  
à l'étape 1, nous remplirons le formulaire ci-dessous :  


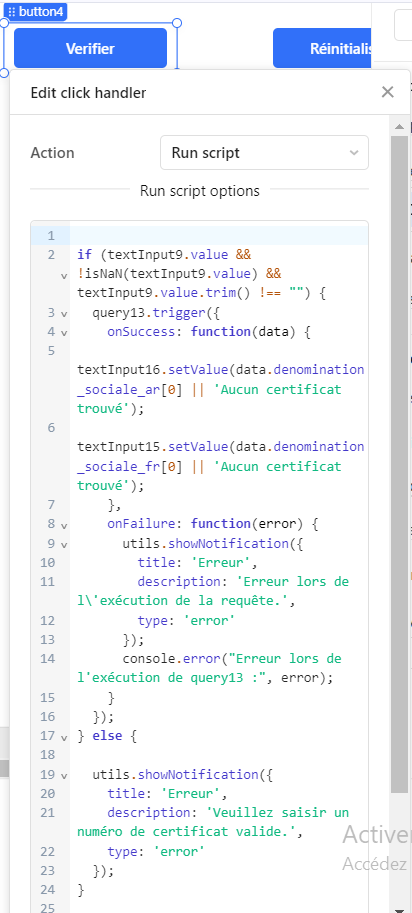




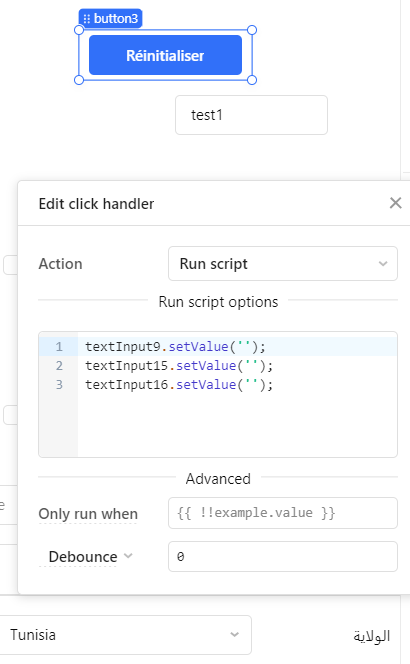
Commençons par la première étape : Vérification du certificat :  


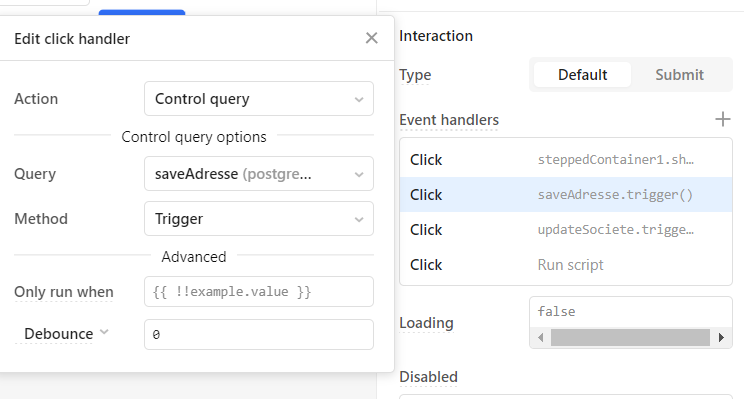
Vous devez d'abord mettre votre widget de saisie:certifNum , Certificat.denomination\_social\_fr et Certificat.denomination\_social\_ar.

En cliquant sur le bouton de vérification, nous mettrons en œuvre ce script:

  
query13 :  
SELECT "denomination\_sociale\_ar", "denomination\_sociale\_fr"

FROM "demande"."certificat"

WHERE "numCertificat" = {{textInput9.value}};  
  
  
et pour la button Réinitialiser : je viens d’exécuter ce script dans Event handlers de la button :  


Pour la button Suivant de step 1 :  
Ajouter des événements :  
1 :la première événement c’est le sauvegarde des adresse :  


Pour la deuxième événement :  
update de la société que je viens de l’ajouter dans l’étape 1 :

