

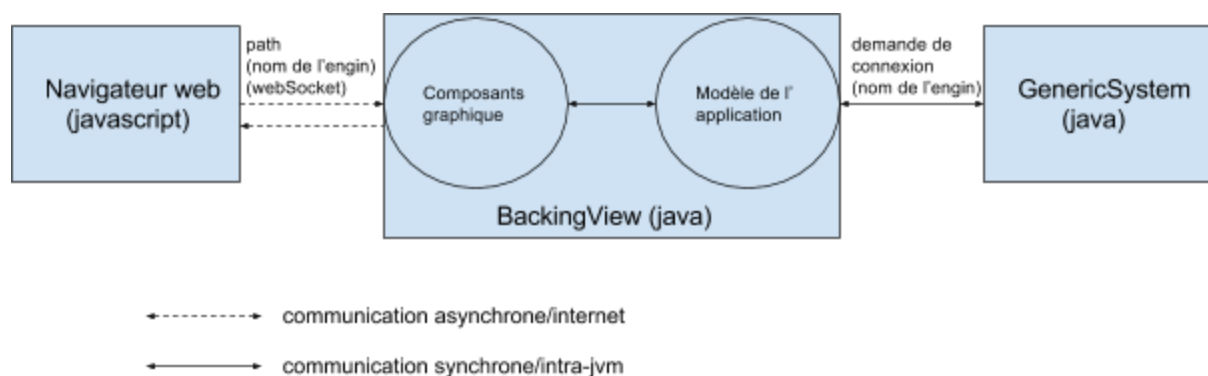
Compte-rendu d'activités - Juba MEHDI

Janvier - 18 février 2016

GS administration réactive et persistante (webBrowser javascript) : l'intérêt du projet est de mettre en place, une application d'administration (sur un navigateur web) réactive et persistante pour gérer le moteur GS. Un protocole de communication est alors mis en place, entre le navigateur web , et le serveur GS.

Architecture et mode de fonctionnement :

Le schéma suivant décrit l'architecture globale de l'application :



Architecture du projet

Dans le but de rendre facile la communication entre le navigateur web et la BackingView (coté serveur), un protocole de communication est mis en place en utilisant les webSocket et le format JSON. Le protocole en question est décrit comme suit :

Trois types de message sont transmissible de la vue d'arrière plan (BackingView) vers le navigateur (message d'ajout, de modification et des message de suppression de noeuds), les messages sont envoyés sous le format JSON, et contient les informations suivantes (*le type du message, l'id du noeud père, l'id du noeud, le tag du noeud et le contenu texte du noeud*).

- message d'ajout : ajout d'un noeud graphique.
- message de modification : modifier un noeud.
- message de suppression : supprimer un noeud.

De sont coté le navigateur web traite chaque information reçu, selon le type du message.

Le navigateur web peut à son tour transmettre de l'information à la vue d'arrière plan grâce au protocole, deux type de message sont possible (*message de modification, message d'action*), les messages sont envoyés sous le format JSON et contient les

information suivante (*type du message ,id du noeud et contenu texte, selon le type du message, le contenu texte peut être null*)

- message de modification : modifier le contenu texte du noeud.
- message d'action : déclencher une action.

Construction du modèle et de l'interface graphique

Le client demande via le navigateur web, d'avoir une connexion vers une base de donnée de son choix, le nom de la base (engin) est spécifié dans le lien de la requête http. A l'envoi de la requête, une websocket est ouverte, permettant ainsi une communication asynchrone entre le navigateur et le serveur GS.

A la réception de la requête, le serveur récupère le nom de la base de donnée (nom de l'engin), le nom de l'engin permet au serveur, de demander à GS d'ouvrir une connexion vers la base en question, et de créer un cache (*communication intra-jvm*). Une fois que la connexion est établie, le serveur commence l'initialisation des arbres de modèle et de composants graphiques. Pour chaque modèle créé un composant graphique lui est associé et initialisé. A la fin de l'initialisation, un message d'ajout est envoyé au navigateur web, via une communication asynchrone utilisant la websocket ouverte au début. Ce message d'ajout, permet au client web (javascript) de créer et de rendre visible le composant graphique.

Dans le cas où GS est impacté, celui-ci réagit jusqu'au navigateur. En premier lieu, L'information est remontée, en synchrone (intra-jvm), de GS aux noeuds graphique en passant par le modèle. En deuxième lieu, en asynchrone, à partir des noeuds graphique au navigateur web en utilisant la websocket.