



Comment utiliser Atom avec NetSuite

17 septembre 2024

Introduction

Lorsqu'on commence à travailler avec NetSuite, il arrivera inévitablement un jour où il sera nécessaire de créer un script basé sur le cadre SuiteCloud. Il existe quelques environnements de développement intégrés (IDE) compatibles avec le cadre SuiteCloud et permettant un codage rapide et efficace.

Pourquoi est-ce important ? Parce que vous voudrez être correctement équipé pour le développement SuiteCloud de NetSuite. Dans l'ensemble, si vous n'êtes pas bien équipé, il peut être assez fastidieux de créer des scripts pour NetSuite.

Si vous lisez ceci, on suppose que vous utilisez déjà NetSuite d'une manière ou d'une autre (soit en tant que client ou fournisseur de solutions) et que vous avez commencé votre première personnalisation. Ou, peut-être avez-vous utilisé un autre IDE et souhaitez passer à Atom.

Dans tous les cas, l'extension des performances et des capacités de NetSuite avec SuiteCloud peut faire une grande différence lorsqu'elle est bien exécutée.

Dans cet article, nous allons couvrir comment utiliser Atom avec NetSuite et comment le configurer pour utiliser GitHub et un plugin d'autocomplétion pour vous aider dans votre saga de script.

Qu'est-ce qu'Atom et pourquoi est-ce important?

Atom est un éditeur de texte gratuit et open-source pour macOS, Linux et Windows. Il est préféré par de nombreux développeurs pour son excellent design, son interface épurée, et l'écosystème exceptionnel de packages communautaires.

La plupart des packages d'extension sont gratuits et développés par la communauté. Atom est basé sur Electron, un cadre qui permet des applications de bureau multiplateformes utilisant Chromium et Node.js. Il est écrit en CoffeeScript et Less.

Atom est une excellente option car il contient un connecteur intégré à GitHub utilisant Git, ainsi que certains packages communautaires pour l'autocomplétion NetSuite. Ces deux facteurs en font un candidat viable pour toute organisation souhaitant l'adopter comme IDE de développement.

Comment configurer Atom pour NetSuite

La configuration d'Atom pour une utilisation avec NetSuite est assez simple.

Télécharger Atom

La première étape consiste à télécharger Atom. Rendez-vous sur ce [site](#) et téléchargez la version la plus récente d'Atom.

Installation d'Atom sur Windows

Atom est disponible avec des installateurs Windows qui peuvent être téléchargés [ici](#) ou à partir du [page des versions d'Atom](#). Utilisez **AtomSetup.exe** pour les systèmes 32 bits et **AtomSetup-x64.exe** pour les systèmes 64 bits. Ce programme d'installation installera Atom, ajoutera les commandes **atom** et **apm** dans votre **chemin**, et créera des raccourcis sur le bureau et dans le menu démarrer.

Le menu contextuel **ouvrir avec Atom** dans l'Explorateur de fichiers et l'option de rendre Atom disponible pour l'association de fichiers via **Ouvrir avec...**, sont contrôlés par le panneau Paramètres système comme indiqué ci-dessus.

Avec Atom ouvert, cliquez sur **Fichier -> Paramètres**, puis sur l'onglet **système** à gauche. Cochez les cases à côté de **montrer dans les menus contextuels de fichier**, ainsi que **montrer dans les menus contextuels de dossier**. Et voilà, tout est prêt.

Installation d'Atom sur Mac

Atom suit le processus d'installation Mac standard via un fichier zip. Vous pouvez soit appuyer sur le bouton de téléchargement depuis le [site](#) ou vous pouvez aller sur la [page des versions d'Atom](#) pour télécharger le fichier **atom-mac.zip** explicitement. Une fois que vous avez ce fichier, vous pouvez cliquer dessus pour extraire l'application, puis faire glisser la nouvelle application **Atom** dans votre dossier "Applications".

Lorsque vous ouvrez Atom pour la première fois, il essaiera d'installer les commandes **atom** et **apm** pour une utilisation dans le terminal. Dans certains cas, Atom pourrait ne pas être en mesure d'installer ces commandes car il a besoin d'un mot de passe administrateur. Pour vérifier si Atom a pu installer la commande **atom**, par exemple, ouvrez une fenêtre de terminal et tapez **which atom**. Si la commande **atom** a été installée, vous verrez quelque chose comme ceci :

Si la commande **atom** n'a pas été installée, la commande **which** ne renverra rien :

Mise à jour d'Atom

La **mise à jour automatique** est activée par défaut dans les paramètres principaux de la vue des paramètres, ce qui permet à Atom de vérifier automatiquement les mises à jour.

Pour effectuer une mise à jour manuelle :

- Cliquez sur l'élément **aide -> vérifier les mises à jour** dans la barre de menu.
- Recherchez **Application: À propos** dans la palette de commandes (ctrl-shift-p) et cliquez sur le bouton **Vérifier maintenant**.

Installer le package autocomplete-netsuite

Ce package est un package d'autocomplétion pour Atom qui complète automatiquement les fonctions de l'API NetSuite SuiteScript.

Vous pouvez l'installer en recherchant **autocomplete-netsuite** dans l'onglet d'installation dans la vue des paramètres. Plus de détails sur la vue des paramètres peuvent être lus [ici](#). Ou, vous pouvez l'installer via la commande suivante : **apm install autocomplete-netsuite**.

Certaines des fonctionnalités incluses sont :

- Prise en charge de SuiteScript 1.0
 - Toutes les fonctions `nlapi*`
 - Tous les constructeurs et fonctions membres `nlobj*`
- Prise en charge de SuiteScript 2.0
 - Toutes les fonctions membres
 - Tous les objets globaux `log` et `util`
 - Toutes les fonctions globales `define` et `require`
 - Toutes les valeurs d'énumération **record.type** (ex: **record.type.employee**)

Il comprend également un extrait de commentaire d'en-tête de fichier NetSuite pratique. Si vous tapez **suiteletcomment** et appuyez sur tab, il insérera un modèle pour l'en-tête de fichier. Vous pouvez ensuite utiliser tab pour naviguer à travers les options du modèle.

Remarque : cet autocomplétion n'active que les fichiers avec l'extension ".js"

Connecter Atom à votre dépôt GitHub

Une fois que vous avez Atom en fonctionnement avec autocomplete-netsuite, l'étape suivante consiste à le connecter à votre dépôt GitHub.

Si vous n'avez pas encore de dépôt GitHub, vous devrez en créer un avant de passer à l'étape suivante.

Une fois que vous avez configuré votre dépôt GitHub, vous verrez un lien comme celui-ci :

Vous devrez copier le petit lien se terminant par ".git". Une fois que vous l'avez copié, revenez à Atom et tapez **"ctrl-shift-p"** puis **"clone"** et vous devriez voir :

Sélectionnez l'option **"GitHub: Clone"** et collez l'URL que vous aviez copiée depuis GitHub.

Une fois que votre dépôt est cloné dans Atom, l'étape suivante sera d'autoriser l'utilisation de GitHub avec Atom. Pour ce faire, allez sur github.atom.io/login.

Assurez-vous de copier votre jeton.

Une fois que vous avez votre jeton, revenez à Atom et cliquez en bas à droite où il est indiqué **"GitHub"**, entrez votre jeton et cliquez sur **"Connexion"**.

Une fois cette étape terminée, votre onglet GitHub devrait ressembler à ceci :

Et voilà, vous avez maintenant un IDE Atom complètement fonctionnel, avec un autocomplétion NetSuite et connecté à votre dépôt GitHub.

2 conseils et rappels pour utiliser Atom avec NetSuite

1. Assurez-vous d'utiliser les commandes "fetch" et "push" pour les modifications de script que vous effectuez afin de garantir que vous utilisez la version la plus récente et la plus précise.
2. Un autre package très utile est linter-jshint. Ce package peut être utilisé pour valider vos fichiers javascript en temps réel.

Conclusion

Si vous avez utilisé Eclipse et que vous en avez assez, cela pourrait être une bonne alternative à essayer. Alternativement, si vous commencez tout juste à écrire des scripts dans NetSuite et que vous cherchez un package de départ, cela vous sera certainement utile.

Essayez-le et dites-moi ce que vous en pensez !

Utilisez-vous un autre IDE avec de bons résultats ? Si c'est le cas, faites-le nous savoir ci-dessous !