



Programmation *Web* Côté Client en *JS*

Petit Tutoriel n°

Tester et déboguer un programme *HTML/JavaScript* avec *Firefox*

Une application *WEB* incluant du code *JavaScript* va s'exécuter dans le navigateur, et éventuellement générer du code *HTML* ou modifier dynamiquement le document via l'accès au *DOM*.

Pour déboguer un tel programme en l'exécutant, un certain nombre d'outils ont été intégrés au navigateur *Firefox*, qui sont accessible notamment dans le menu :

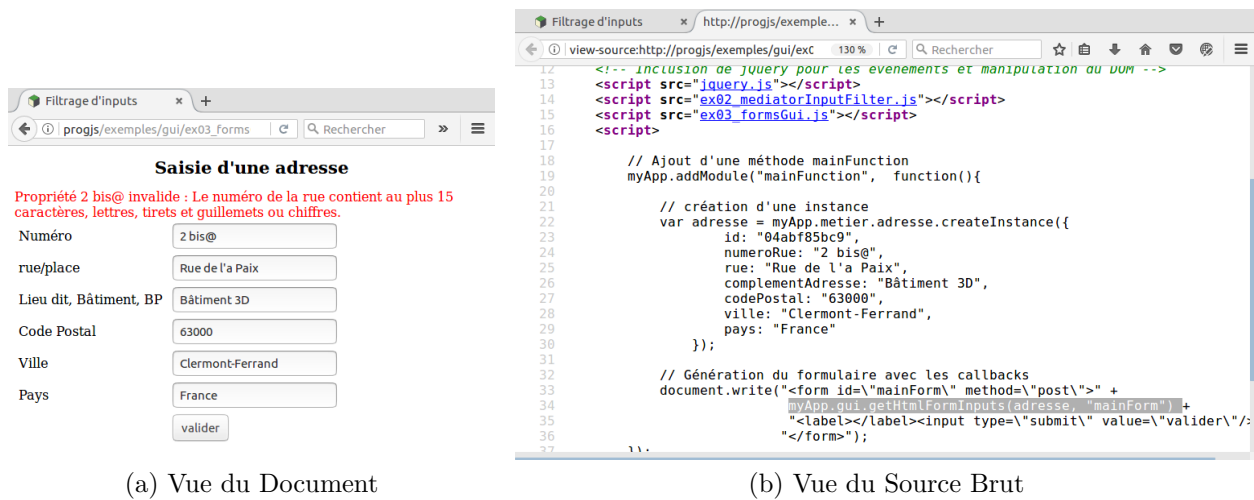
Outils → *Développement Web* → *Console Web*

Table des matières

1	Voir le Code Source <i>HTML/JavaScript</i>	1
2	Inspecter le Code <i>HTML</i> du Document	2
3	Afficher les erreurs <i>JavaScript</i>	2
4	Débuggage, points d'arrêt, contenu des variables	3
5	Débugger une <i>API Restful</i> avec l'extension <i>RESTClient</i>	3

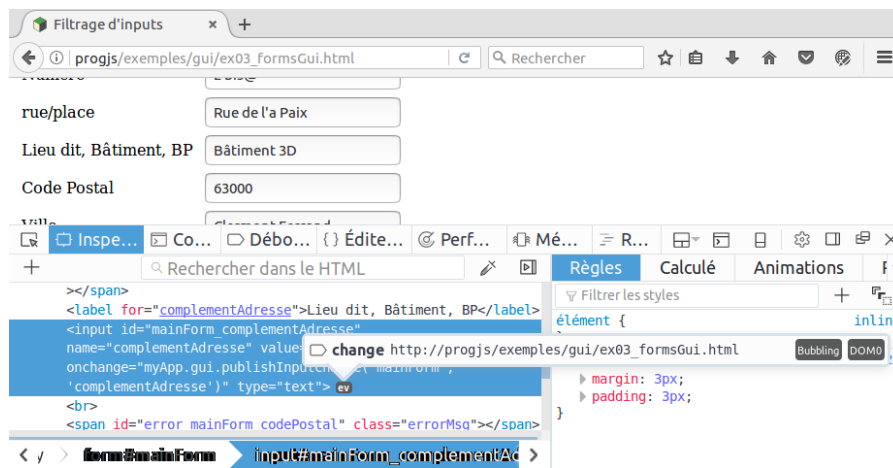
1 Voir le Code Source *HTML/JavaScript*

Pour voir le code source brut *HTML/JavaScript* (voir la figure 1), utiliser le menu contextuel (faire un *click* droit sur la page).

FIGURE 1 : Vue du document *HTML* généré dynamiquement et du code source brut

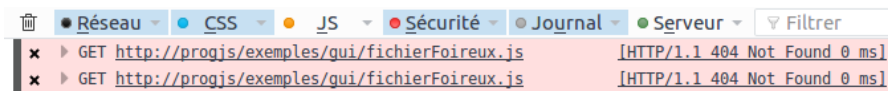
2 Inspecter le Code *HTML* du Document

Pour voir le code *HTML* du document généré dynamiquement en *JavaScript* (via `document.write` ou via les méthode *jQuery* de manipulation du *DOM*), faire un *click* droit sur l'élément *HTML* du document qui vous intéresse (ici un élément `input`). On peut aussi voir les événements *JavaScript* (*click*, *change*, *hover*, etc.) associés à ces éléments *HTML* (voir la figure 2).

FIGURE 2 : Vue code *HTML* du document généré dynamiquement et des événements *JS*

3 Afficher les erreurs *JavaScript*

Pour afficher les erreurs de syntaxe *JavaScript*, ouvrir la console *WEB* (onglet *JS* et recharger la page (voir la figure 3).

FIGURE 3 : Vue du document *HTML* généré dynamiquement et des événements *JS*

4 Débuggage, points d'arrêt, contenu des variables

En cliquant sur l'onglet *Debugger* de la console *Web*, on accède aux fonctionnalités traditionnelles d'un débogueur (voir la figure 4). On spécifie un point d'arrêt dans le code *JavaScript* par un *click* droit sur le numéro de ligne sur la marge de gauche. L'exécution du code (bouton *play* ou "exécuter") permet ensuite d'exécuter pas à pas le code, en visualisant l'état des variables de manière dynamique en cours d'exécution.

Notez que la visualisation de la portée des variables permet de **recenser** très simplement les **variables globales**, pour s'assurer qu'il n'y aura pas de couplage involontaire dans le code du à l'**oubli du mot réservé var**, créant involontairement une variable globale.

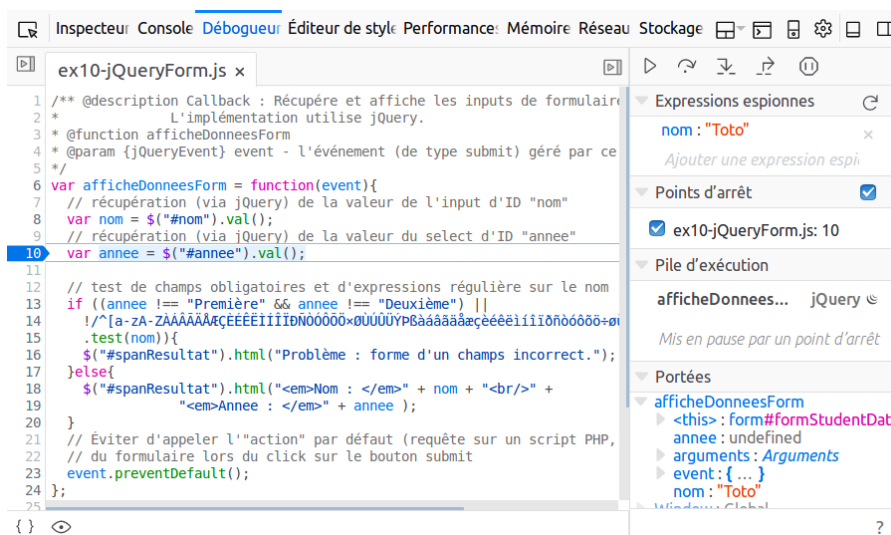


FIGURE 4 : Vue du débogueur en pause sur un point d'arrêt (ligne 10)

5 Débugger une *API Restful* avec l'extension *RESTClient*

L'extension *Firefox RESTClient* (voir la figure 5) permet de tester des requêtes sur une *API HTTP* en choisissant les verbes *HTTP* (GET, PUT, POST, OPTIONS, etc.) et d'examiner le résultat, sur la réponse et sur l'état du serveur. Cela permet de décomposer le débugeage de l'architecture client/serveur, en s'assurant que le *Web Service* est fonctionnel, avant de tester la couche persistance *AJAX* du client.

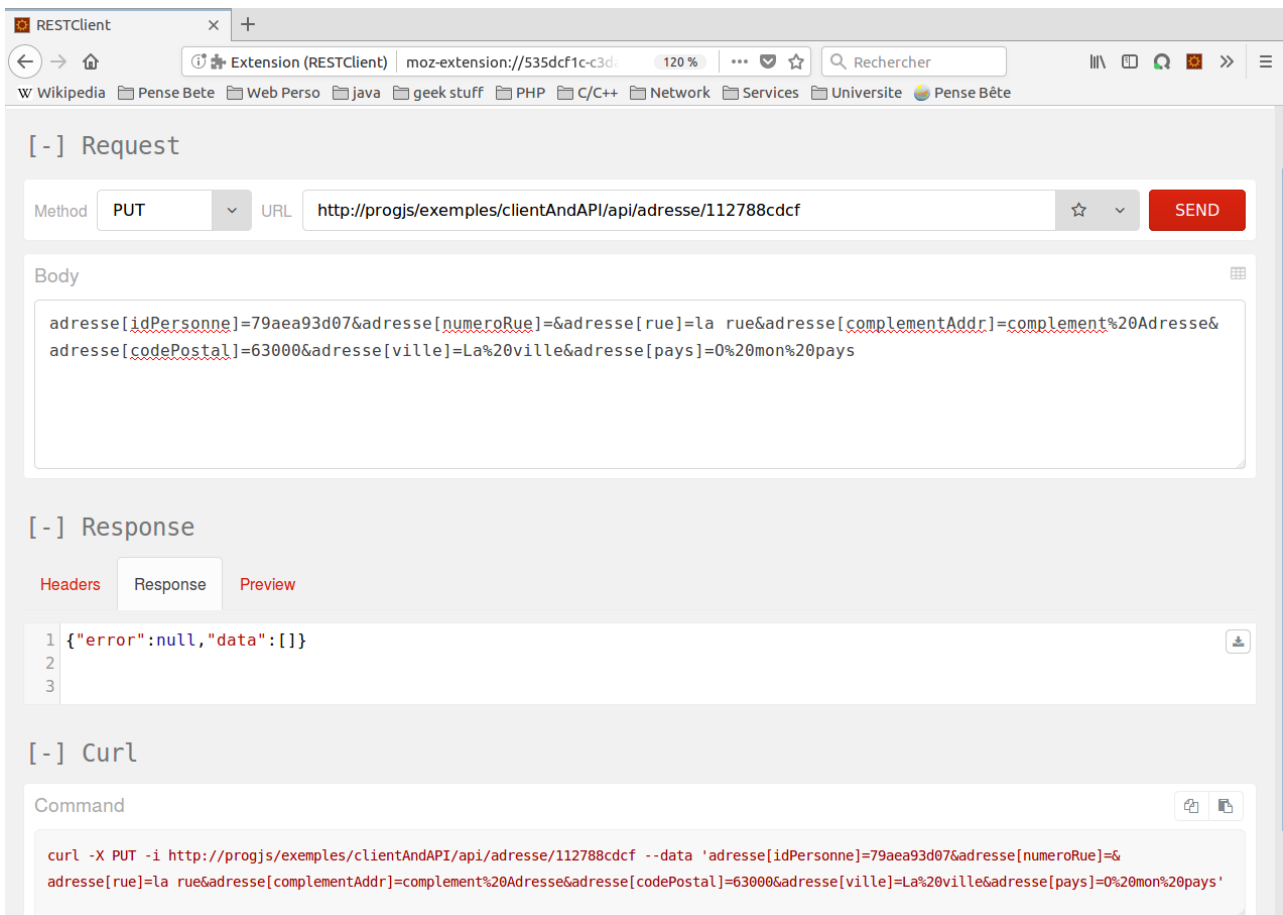


FIGURE 5 : Vue de l'extension *Firefox* permettant de tester des requêtes sur une *API Restful*