# Tanaguru (Webextension)

[Tanaguru (Webextension) 1](#_Toc507426170)

[A propos de ce document 1](#_Toc507426171)

[Installer la webextension 1](#_Toc507426172)

[Utiliser la webextension 2](#_Toc507426173)

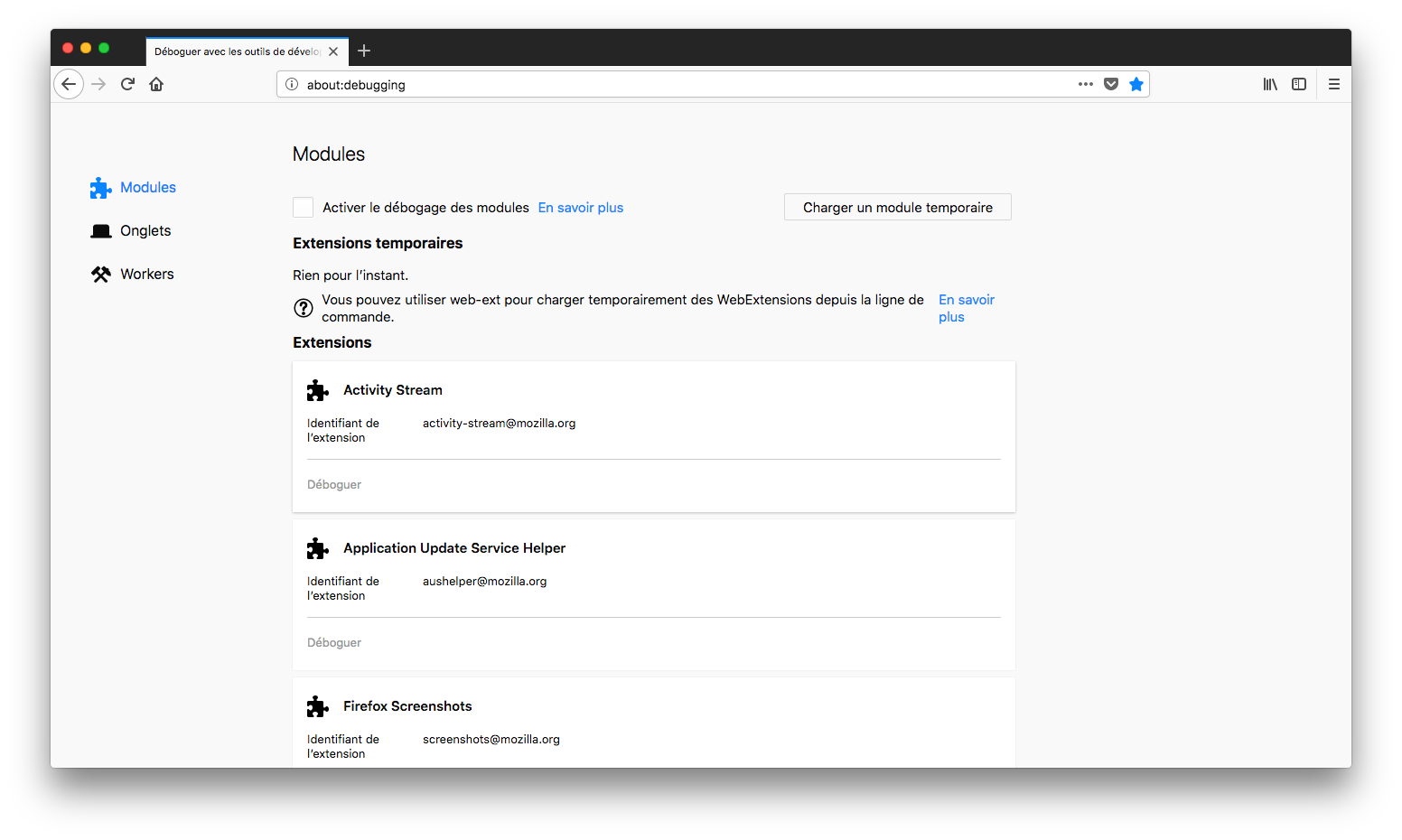
[Écrire un test 4](#_Toc507426174)

## A propos de ce document

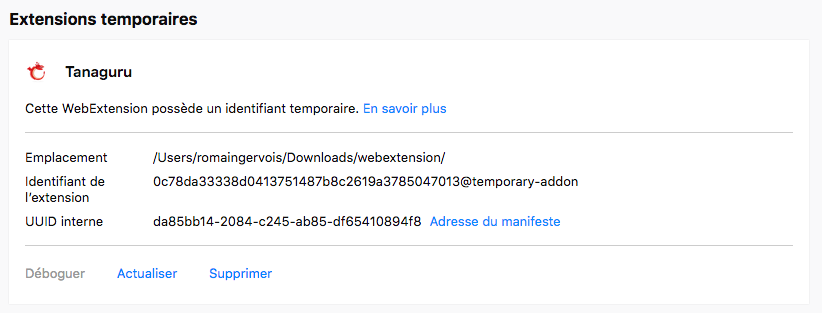
* Auteur : Romain Gervois.
* Projet : Tanaguru Webextension.
* Date de rédaction : 26/02/2018.

## Installer la webextension

Pour installer la webextension, **dans la barre d’adresse** de Firefox 57 ou supérieur, saisissez « **about:debugging** ». La page « Modules » s’affiche :



Sur cette page, activez le bouton « **Charger un module temporaire** ». Une boîte de dialogue de fichier s’affiche alors. Depuis cette boîte, sélectionnez sur votre disque local, le fichier « **manifest.json** ». Cette sélection termine l’installation de la webextension.



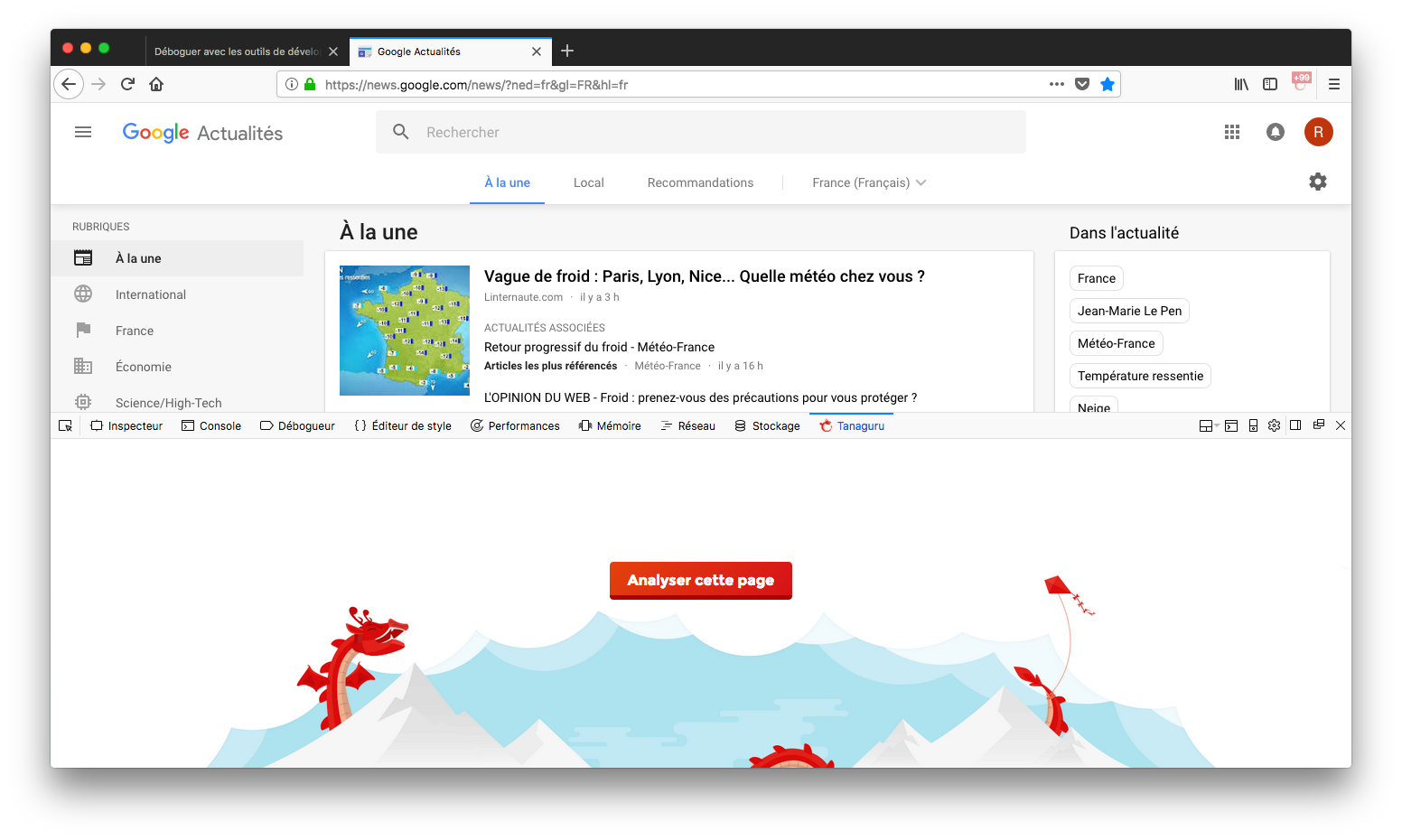
Note : la webextension est installée temporairement. C’est-à-dire qu’elle sera désinstallée à la fermeture de Firefox.

## Utiliser la webextension

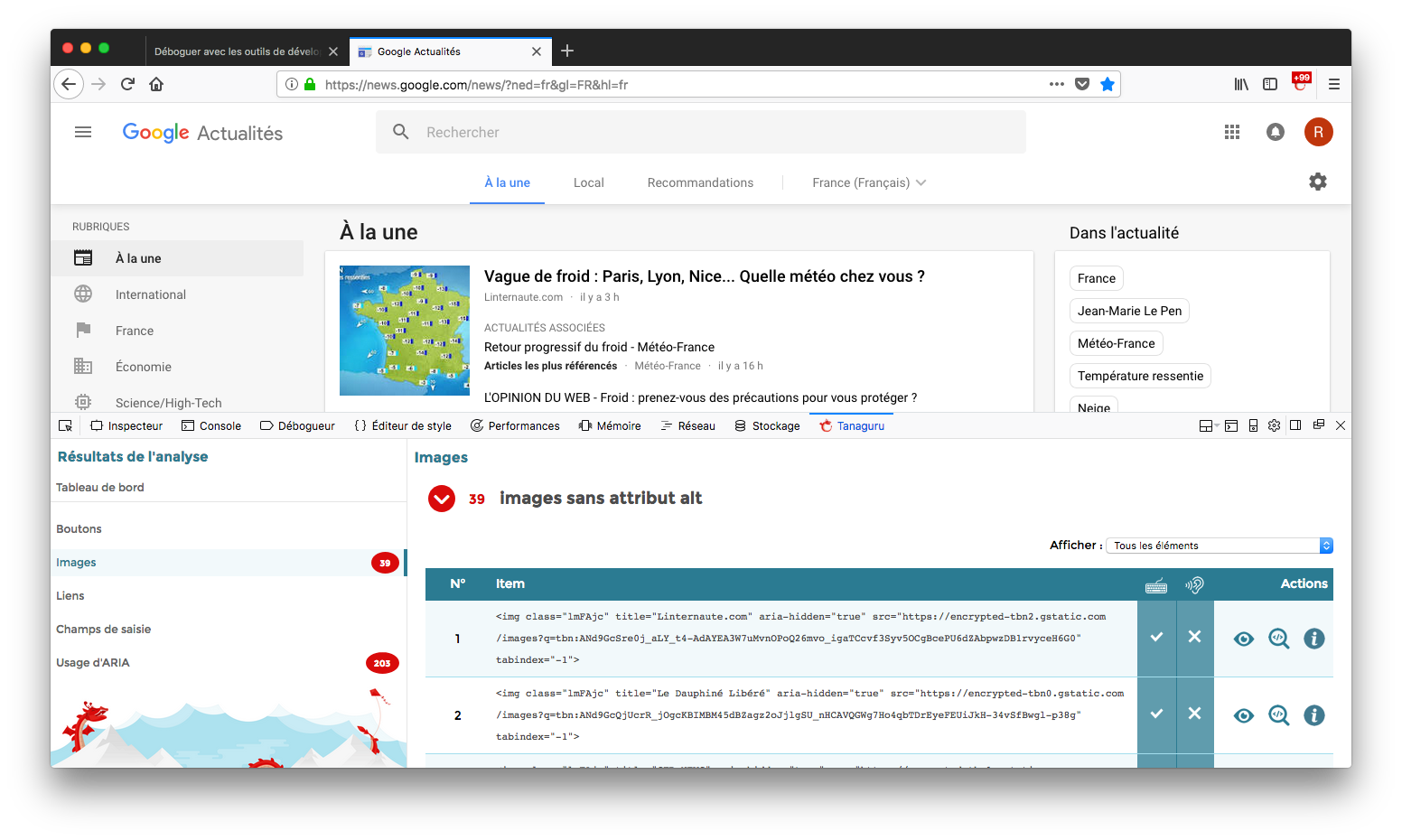
Une fois, l’installation réalisée, un bouton « Tanaguru » apparait dans la barre d’outils de Firefox. Ce bouton permet à l’utilisateur via l’affichage d’une fenêtre popup de connaître la version de la webextension, la procédure pour démarrer les tests et d’aller consulter le site de Tanaguru pour plus d’informations.



Pour démarrer les tests, **rendez-vous sur la page à analyser** puis, si l’outil de développement n’est pas déjà affiché, allez dans « **Outils > Développement Web > Outils de développement** » puis activez l’onglet « **Tanaguru** ».



Activez le bouton « **Analyser cette page** ». Selon la « complexité » du DOM de la page à analyser, un message peut vous demander si vous souhaitez « Arrêter le script » ou « **Patienter** » (choisissez cette option pour continuer). Les résultats s’affichent enfin en lieu et place du bouton « Analyser cette page ».



Chaque résultat est accompagné de trois boutons :

Trois boutons par élément détecté : mettre en évidence sur la page, révéler dans l'inspecteur et à propos de cet élément.

* « **Mettre en évidence sur la page** » permet d’appliquer un style pour faire ressortir visuellement l’élément sur la page.
* « **Révéler dans l'inspecteur** » permet de sélectionner le nœud correspondant dans l’inspecteur de la page.
* « **À propos de cet élément** » permet de consulter et récupérer les représentations XPath et CSS de l’élément.

## Écrire un test

L’écriture d’un test s’effectue depuis le fichier Javascript « **/common/scripts/analyze.js** ».

**Intégrer entre la ligne de création de la variable json :**

var json = {};

**Et la ligne de renvoi de valeur (de la variable json) :**

json;

**Le test que vous souhaitez réaliser et ajouter aux résultats de la webextension.**

Par exemple, si vous souhaitez lister tous les liens possédant un attribut target=“\_blank“, utilisez d’abord la méthode DOM permettant de rechercher ces éléments :

var linksWithTargetBlank = document.querySelectorAll(‘a[href][target=“\_blank“]’);

Créez un tableau vide puis remplissez-le en parcourant chacun des éléments trouvés :

var linksWithTargetBlankSet = [];

for (var i = 0; i < linksWithTargetBlank.length; i++) { linksWithTargetBlankSet.push(manageOutput(linksWithTargetBlank[i])); }

Note : la fonction « **manageOutput** » génère un objet JSON avec les différentes informations nécessaires pour exploitation par l’interface de la webextension.

Enfin, appelez la fonction **addResultSet** avec :

* En **premier paramètre**, le nom du panneau où les résultats seront affichés ;
* En **second paramètre**, un objet JSON avec :
  + La propriété « **name** » correspondant au nom du test réalisé ;
  + La propriété « **type** » correspondant à la nature des résultats (valeur « **failure** » s’il s’agit d’éléments en erreur ou valeur « **humanneeded** » s’il s’agit d’éléments à expertiser) ;
  + La propriété « **data** » correspondant à l’objet contenant les résultats ;
  + Éventuellement, la propriété « **description** » correspondant à un texte accompagnant les résultats (consignes, explications, etc.).
  + Éventuellement la propriété « **mark** » correspondant à une expression régulière qui sera utilisée pour mettre en valeur certains passages de code dans l’aperçu depuis l’interface de la webextension.

Pour que les résultats soient chargés et exploités par l’interface de la webextension :

addResultSet(‘Liens’, { name: 'Lien' + (linksWithTargetBlankSet.length > 1 ? 's' : '') + ' avec attribut target=”\_blank”', type: 'humanneeded', data: linksWithTargetBlankSet, mark: '(target=&quot;(?:(?!&quot;).)\*&quot;)' });

Note : si le premier paramètre (« Liens ») a déjà été utilisé, l’objet JSON spécifié viendra compléter le (ou les) objet(s) JSON précédemment spécifié(s) et l’ensemble de ces objets seront chargés et affichés dans le même panneau.

