# Documentation développeur :

## Outils à installer

Un environnement de développement html/javascript quelconque est suffisant. Il faut installer les navigateur internet Mozilla Firefox et Chrome.

## Configuration de l’environnement de développement

Rien de particulier à signaler.

## Procédure de build

Le test de la web-extension se fait via les deux navigateurs :

* Pour Firefox :

Dans la barre d’url rentrez le texte suivant : « about:debugging », puis cliquez sur « Charger un module temporaire » et sélectionnez l’un des fichier de votre projet. La web-extension sera charger dans votre navigateur.

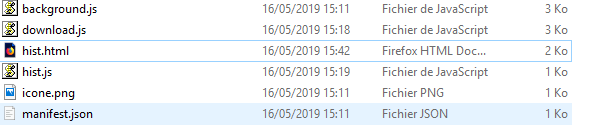
* Pour Chrome :

Dans la barre d’url rentrez l’adresse suivante : « [chrome://extensions](chrome://extensions/) », puis cliquez sur « Charger l’extension non empaquetée » et sélectionner le dossier racine de votre projet (celui contenant manifest.json)

Une fois la web-extension chargée, il suffit d’appuyer sur actualiser pour remettre à jour au niveau du code.

Le build à proprement parler n’est possible d’après vérification par Mozilla Firefox et Chrome de l’extension temporaire, et ce pour des raisons de sécurités.

## Présentation de l'architecture, des concepts



L’architecture d’une extension web est centrée sur un fichier manifest.json contenant toutes les informations nécessaires pour que le navigateur puisse charger l’extension. Celui-ci contient entre autres :

* Les permissions nécessaires, ce qui déclenchera ou non suivant leur gravité une demande d’acceptation par l’utilisateur.
* Les fichiers à appeler lors du lancement du navigateur, d’une page, ou d’un appui sur le bouton de l’extension.
* Le lien vers l’icône de l’extension.

Dans le cas de notre extension, manifest.json appelle background.js lors de l’ouverture du navigateur (ou mise en route de l’extension) et hist.html lors d’un appui sur le bouton, ce dernier utilisant les deux fichiers download.js et hist.js.

On dispose des permissions suivantes : "webNavigation", et "storage". La première est nécessaire pour accéder aux données de navigation (url visitées), la deuxième permet d’avoir accès au stockage du navigateur et ainsi sauvegarder des données d’une page à l’autre.

Le concept principale repose sur l’utilisation de chrome.storage, compatible pour les deux navigateurs. Dans background.js, on sauvegarde ainsi les données que l’on renvoie au besoin à la page hist.html qui est un popup ouvert lors de l’appui sur le bouton. On peut ainsi afficher l’historique de navigation. La création et le téléchargement du fichier export se fait dans le fichier download.js.