

 eBook Gratuit

APPRENEZ

firefox-addon

eBook gratuit non affilié créé à partir des
contributeurs de Stack Overflow.

#firefox-
addon

Table des matières

À propos.....	1
Chapitre 1: Démarrer avec firefox-addon.....	2
Exemples.....	2
introduction.....	2
Modules complémentaires:.....	2
Les extensions.....	2
Déchéance et suppression de tous les types d'extensions autres que WebExtensions.....	2
Types d'extensions.....	2
Autres types de modules complémentaires Firefox.....	4
Installation d'un module complémentaire temporaire.....	5
WebExtensions.....	6
Firefox SDK complémentaire.....	6
Bootstrap / Restartless.....	7
Legacy / Overlay / XUL.....	7
Installer des modules complémentaires non signés.....	7
Installation de modules complémentaires pour le développement.....	7
Modules complémentaires normaux.....	8
Limitations sur l'installation des add-ons normaux: Add-on Signing.....	8
Add-ons temporaires (extensions uniquement).....	9
Crédits.....	11

À propos

You can share this PDF with anyone you feel could benefit from it, downloaded the latest version from: [firefox-addon](#)

It is an unofficial and free firefox-addon ebook created for educational purposes. All the content is extracted from [Stack Overflow Documentation](#), which is written by many hardworking individuals at Stack Overflow. It is neither affiliated with Stack Overflow nor official firefox-addon.

The content is released under Creative Commons BY-SA, and the list of contributors to each chapter are provided in the credits section at the end of this book. Images may be copyright of their respective owners unless otherwise specified. All trademarks and registered trademarks are the property of their respective company owners.

Use the content presented in this book at your own risk; it is not guaranteed to be correct nor accurate, please send your feedback and corrections to info@zzzprojects.com

Chapitre 1: Démarrer avec firefox-addon

Exemples

introduction

Modules complémentaires:

Les modules complémentaires de Firefox sont généralement regroupés dans les extensions, puis dans les "autres types" de modules complémentaires de Firefox.

Les extensions

Les extensions permettent de personnaliser Firefox en ajoutant ou en modifiant les fonctionnalités de Firefox. Certains des types de choses qui peuvent être faites avec des extensions incluent:

- Modifiez l'apparence de sites Web spécifiques, leur contenu ou leur interaction.
- Personnaliser l'interface utilisateur de Firefox
- Ajouter des fonctionnalités supplémentaires à Firefox
- Modifier le fonctionnement des fonctionnalités existantes de Firefox

Les extensions Firefox sont principalement écrites en JavaScript avec l'ajout de certaines API JavaScript.

Déchéance et suppression de tous les types d'extensions autres que WebExtensions

Les modules complémentaires de Firefox, en particulier les extensions, sont en pleine mutation pour le moment. Mozilla a annoncé et confirmé avoir désapprouvé tous les types d'extensions de Firefox, à l'exception de WebExtensions, et que [toutes les extensions non basées sur WebExtensions seraient désactivées dans Firefox 57, prévu pour 2017-11-14](#) .

Types d'extensions

Firefox dispose de quatre types d'extensions (qui sont communément appelées add-ons):

- [WebExtensions](#) : À l'avenir, WebExtensions est le seul type d'extension de Firefox qui sera pris en charge. Ces add-ons sont décrits par un fichier [manifest.json](#) . Cette API est similaire à celle utilisée pour les extensions Google Chrome. Ces add-ons utilisent HTML et CSS en plus de Javascript. Alors que Mozilla a déclaré que cette API est l'avenir des extensions Firefox, cette API est toujours en développement. Pour l'instant, il est probablement

préférable de développer et de tester votre module complémentaire WebExtension avec [Firefox Developer Edition](#) ou [Firefox Nightly](#) . Vous devez également prendre note de la version de Firefox requise pour les fonctionnalités que vous souhaitez utiliser. Ces informations sont contenues dans la section "Compatibilité du navigateur" des pages de documentation de MDN.

WebExtensions utilise une API significativement différente des trois autres types d'extensions. Il n'y a, intentionnellement, aucune possibilité d'utiliser les interfaces fournies par les autres types de modules complémentaires.

- **Add-on SDK** : [**déconseillé; prévu pour la suppression**] Ces add-ons sont décrits par un fichier `package.json` qui est initialement généré en exécutant `jpm init` . Ces extensions utiliseront souvent `require()` pour charger des API de [haut niveau](#) ou de [bas niveau](#) afin d'interfacer Firefox. Ces add-ons utilisent HTML et CSS en plus de Javascript. Actuellement, ces modules complémentaires sont intégrés dans une extension `jpm run` lorsqu'ils sont chargés pour être testés par `jpm run` ou consolidés dans un fichier `.xpi` par `jpm xpi` pour la distribution (par exemple, téléchargement vers AMO / Mozilla). En d'autres termes, ce sont des extensions bootstrap avec un wrapper SDK.

Mozilla semble être déterminé à continuer à prendre en charge les extensions basées sur un SDK complémentaire tant que l'extension n'utilise pas la configuration `require("chrome")` ou dépend de [XUL](#) , [XPCOM](#) et [XBL](#) .

La plupart des choses qui peuvent être faites dans une extension amorcée peuvent être effectuées dans une extension basée sur un SDK complémentaire. Cependant, beaucoup de ces facteurs contournent le kit de développement logiciel (SDK), qui perd une partie importante des avantages de l'utilisation du SDK complémentaire.

- **Amorcé** : [**déconseillé; planifié pour la suppression**] Ces extensions sont aussi communément appelées "restartless" car elles étaient le premier type d'extension Mozilla qui ne nécessitait pas le redémarrage de l'application pour charger / décharger le module complémentaire. Toutefois, sans redémarrage est un descripteur de leur fonctionnement. L'utilisation de "restartless" comme nom pour ce type de module complémentaire est source de confusion car les modules complémentaires SDK et WebExtension ne nécessitent pas non plus le redémarrage de l'application lors du chargement ou du déchargement du module complémentaire. Pour cette raison, il y a une tendance à ne plus utiliser "restartless" comme nom pour ce type d'add-on.

Ces add-ons utilisent HTML et CSS en plus de Javascript. Beaucoup interagissent également avec Firefox en utilisant [XUL](#) .

Ces add-ons ont un fichier JavaScript appelé `bootstrap.js` qui doit contenir [des points d'entrée \(fonctions\)](#) appelés pour les modules complémentaires `startup()` , `shutdown()` , `install()` et `uninstall()` .

Ces modules complémentaires contiennent un fichier `install.rdf` qui décrit le module complémentaire. Ils contiennent généralement, mais pas toujours, un fichier `chrome.manifest` qui décrit comment les fichiers et les répertoires de l'extension sont liés à

l'application Mozilla (par exemple, Firefox).

La plupart des tâches qui peuvent être effectuées dans les extensions / XUL / Legacy peuvent être accomplies dans des modules complémentaires bootstrap. Tout ce qui peut être fait dans le SDK complémentaire peut être fait dans une extension amorcée (les extensions du SDK additionnel sont des add-ons bootstrap avec certaines couches d'API basées sur JavaScript).

Mozilla a déclaré [qu'ils envisageaient de désapprouver les](#) «modules complémentaires qui dépendent de [XUL](#) , [XPCOM](#) et [XBL](#) ». Bien que tous les modules complémentaires démarrés ne dépendent pas de ces technologies, les modules complémentaires démarrés ont tendance à fonctionner à un niveau inférieur à celui des modules complémentaires SDK et WebExtension. Ils sont donc plus susceptibles d'utiliser ces technologies. Bien que certains affirment que tous les modules complémentaires bootstrap sont censés être obsolètes, il n'est pas certain que ce soit le cas. Après tout, les extensions du SDK complémentaire ne sont pas obsolètes (sauf si elles utilisent `require("chrome")` ou dépendent de XUL, XPCOM ou XBL) et que toutes les extensions du SDK complémentaire sont des extensions amorcées, avec un wrapper SDK uniquement .

- [Overlay / XUL / Legacy](#) : [**déconseillé; planifié pour la suppression**] Ces add-ons contiennent un fichier [install.rdf](#) qui décrit le complément et un fichier [chrome.manifest](#) pour décrire la relation entre les fichiers du module complémentaire (par exemple, superposition) et les fichiers de l'application. Comment les fonctions complémentaires à l'application dépendent complètement des relations décrites dans le fichier [chrome.manifest](#) . Les seules exceptions à cela sont quelques éléments tels que les icônes de l'extension et le fichier décrivant les options de l'extension, qui sont indiqués dans le fichier [install.rdf](#) . Ces extensions interagissent avec l'application (par exemple Firefox) à un niveau très bas. Cela les rend plus susceptibles de se briser lorsque des modifications sont apportées à l'application.

Ces add-ons utilisent XUL, HTML et CSS en plus de Javascript. Certains utilisent également [XPCOM](#) et [XBL](#) .

Toutes les extensions Overlay / XUL / Legacy devraient être obsolètes.

Autres types de modules complémentaires Firefox

Lorsque la plupart des gens pensent aux modules complémentaires de Firefox, ils pensent aux extensions décrites ci-dessus. Cependant, il existe d'autres types de modules complémentaires pour Firefox:

- [Les thèmes](#) [**partiellement déconseillés**] permettent de personnaliser l'apparence de Firefox. Principalement en fournissant différentes règles CSS à toutes les parties du navigateur. [Les "thèmes complets" sont obsolètes](#) et [prévus pour être partiellement remplacés par une interface inexistante](#) .

- [Les modules complémentaires mobiles](#) sont pour Firefox pour Android. Actuellement, tous les types d'extension Firefox sans redémarrage sont pris en charge.
- [Les plug-ins de moteur de recherche](#) sont utilisés pour ajouter des moteurs de recherche supplémentaires à la barre de recherche du navigateur.
- [Les dictionnaires utilisateur](#) permettent d'utiliser le correcteur orthographique dans d'autres langues.
- [Les modules linguistiques](#) ajoutent des langues supplémentaires à l'interface utilisateur de Firefox.
- [Les plug-ins](#) sont des bibliothèques partagées pour afficher le contenu que l'application elle-même ne peut pas afficher en mode natif.

Cet "exemple" est principalement copié, avec quelques modifications, de [ma réponse Makyen sur une question de stackoverflow](#) .

Certaines parties de ce contenu ont été copiées, ou du moins basées sur la [page Add-on sur](#) le réseau de développeurs Mozilla (MDN).

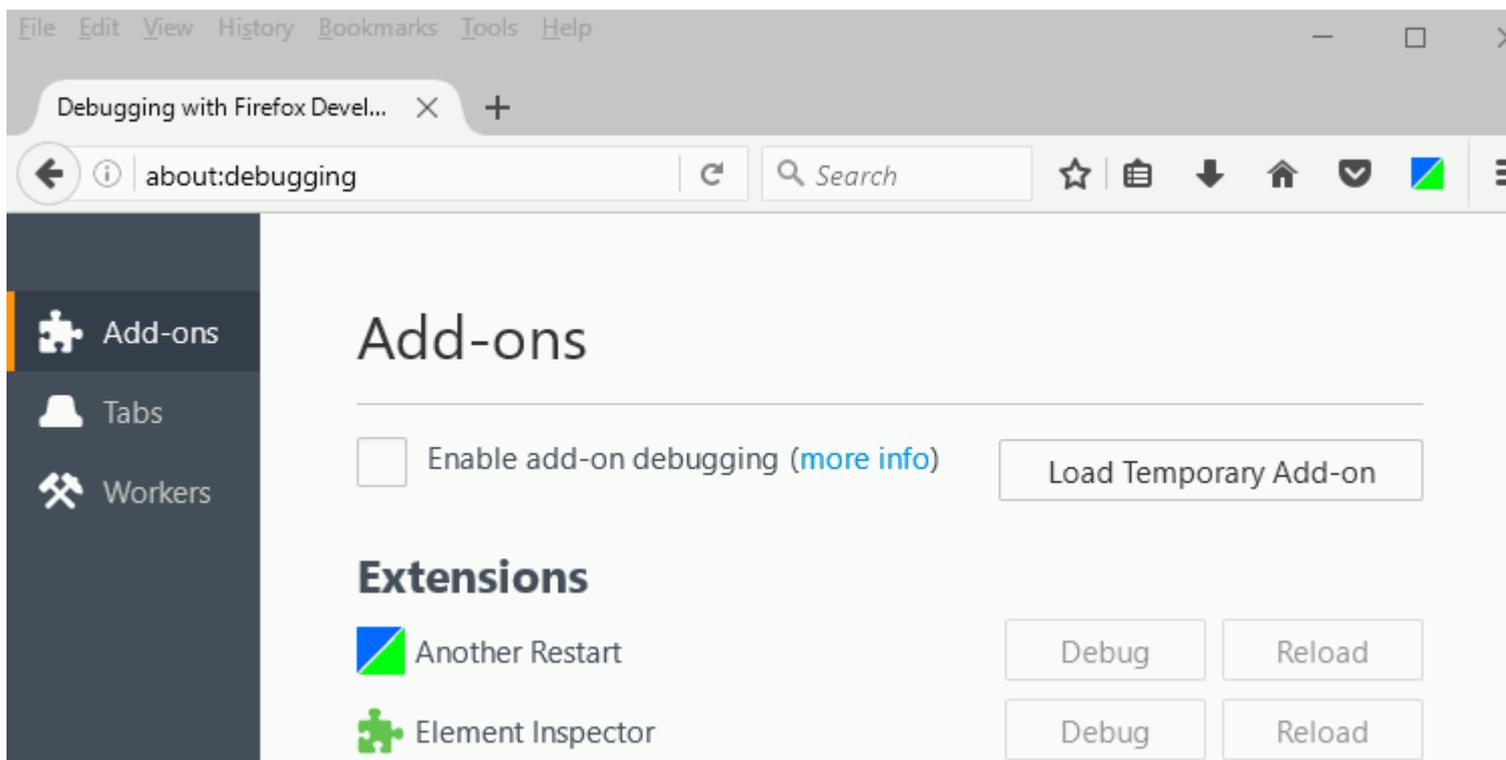
Ceci a été initialement publié par [Makyen](#) dans la section "Remarques" de la [balise firefox-addon](#) . Il a ensuite été modifié par [Ondřej Doněk](#) , qui a retiré un "pour" supplémentaire. Dans une seule édition effectuée par [nus](#) , il a été déplacé de "Remarques" vers un "exemple" épinglé intitulé "Introduction". Malheureusement, le système a perdu des informations d'attribution.

Installation d'un module complémentaire temporaire

Pour tester un module que vous développez, vous souhaitez probablement l'installer temporairement dans Firefox. Vous pouvez le faire en le chargeant comme [add-on temporaire](#) . Faire cela:

1. Aller à [about:debugging](#)
2. Cliquez sur "Load Temporary Add-on"
3. Dans le sélecteur de fichiers, accédez au répertoire contenant les fichiers complémentaires
4. Sélectionnez n'importe quel fichier dans le dossier
5. Cliquez "Ouvrir"

L'animation suivante montre le chargement d'un add-on nommé "aaaaaaaaaaaaaaaaaaaa - add-on démo" à [about:debugging](#) et que le complément apparaît ensuite dans [about:addons](#) :



En tant que module complémentaire temporaire, vous pouvez charger un module complémentaire décompressé (un répertoire contenant tous les fichiers de l'add-on que vous placerez dans un fichier `.xpi`), ou un module ajouté dans un fichier `.xpi` archive. Il n'est pas nécessaire de [signer les modules complémentaires temporaires](#). Le module complémentaire temporaire reste installé jusqu'à sa désinstallation manuelle ou le redémarrage de Firefox.

Documentation Mozilla: [Installation temporaire dans Firefox](#)

WebExtensions

WebExtensions peut être chargé en tant que modules complémentaires temporaires. Cela peut être fait avec les fichiers additionnels décompressés ou empaquetés dans un fichier `.xpi`.

Firefox SDK complémentaire

Vous ne pouvez pas charger une extension Firefox SDK complémentaire en tant que module complémentaire temporaire sans la mettre au préalable dans un fichier `.xpi` avec `jpm xpi`. En général, vous utiliserez `jpm run` pour tester votre extension Firefox SDK complémentaire.

Les fichiers généralement modifiés pour une extension SDK de module complémentaire ne constituent pas une extension complète sans certaines fonctions d'encapsulation supplémentaires et le fichier `package.json` est traduit dans un fichier `install.rdf` et, éventuellement, dans un fichier `chrome.manifest`. Ce processus encapsule l'extension du SDK complémentaire dans un module complémentaire Bootstrap / Restartless, compris par Firefox. Sans ce processus, Firefox ne pourra pas charger le module complémentaire. Ce processus est effectué par `jpm xpi` ce `jpm xpi` donne un fichier `.xpi` compressé. L'exécution de `jpm run` exécute également ce processus, mais stocke les fichiers résultants dans un emplacement temporaire et appelle Firefox avec le module complémentaire installé.

Bootstrap / Restartless

Les modules complémentaires Bootstrap / Restartless peuvent être chargés en tant que modules complémentaires temporaires. Cela peut être fait avec les fichiers additionnels décompressés ou empaquetés dans un fichier `.xpi`.

Legacy / Overlay / XUL

Les modules complémentaires Legacy / Overlay / XUL **ne** peuvent **pas** être chargés en tant que modules complémentaires temporaires.

Installer des modules complémentaires non signés

Pour installer une extension en tant que module complémentaire normal dans les versions Release ou Beta de Firefox version supérieure ou égale à la version 48, l'extension *doit* être [signée par Mozilla](#). Une extension est signée en la [soumettant à AMO](#). Une fois signé, l'extension peut être installée sur toute version de Firefox prise en charge. Pour les versions de Firefox antérieures à la version 43, il n'était pas nécessaire que les extensions soient signées par Mozilla. [Les types de modules complémentaires autres que les extensions](#) ne doivent pas nécessairement être signés par Mozilla.

Vous pouvez installer des extensions non signées en tant qu'add-ons normaux dans d'autres versions de Firefox (par exemple, [Firefox Developer Edition](#), [Firefox Nightly](#), [Unbranded Beta ou Unbranded Release](#)) en définissant `xpinstall.signatures.required` sur `false` dans `about:config`. La définition de cette option était également efficace dans les versions 43 et 48 de Release et Beta Firefox. La définition de cette option n'est pas effective dans les versions Release et Beta de Firefox à partir de la version 48.

La nécessité de pouvoir installer des modules complémentaires non signés lors du développement d'un module complémentaire a été considérablement réduite par la disponibilité des extensions d'installation en tant que [modules complémentaires temporaires](#). Les modules complémentaires temporaires n'ont pas besoin d'être signés et peuvent être chargés dans n'importe quelle version actuelle de Firefox. Comme son nom l'indique, le principal inconvénient des modules complémentaires temporaires est qu'ils sont temporaires. Ils doivent être réinstallés à chaque redémarrage de Firefox. Cependant, il existe des cas d'utilisation où il est préférable d'installer un module complémentaire en tant que module complémentaire normal, mais non signé, plutôt qu'en tant que module complémentaire temporaire. Un module complémentaire doit être installé en tant que module complémentaire non signé s'il est nécessaire que le module reste installé même après le redémarrage de Firefox. Cela peut être souhaitable pour diverses raisons, y compris: des tests d'utilisation à plus long terme ou pour tester le fonctionnement du module complémentaire au démarrage de Firefox.

Installation de modules complémentaires pour le développement

Les modules complémentaires peuvent être installés en tant que:

1. Modules complémentaires normaux, installés jusqu'à leur désinstallation
2. Modules complémentaires temporaires ([extensions](#) uniquement): ne sont installés que lorsque Firefox est redémarré ou peuvent être désinstallés manuellement plus tôt.
3. Utilisation de `jpm run` ([Kit de développement jpm run](#) uniquement): `jpm run` automatiquement Firefox en utilisant un profil temporaire avec votre module complémentaire chargé.
4. Utilisation de `web-ext run` ([WebExtensions](#) uniquement): exécute automatiquement Firefox en utilisant un profil temporaire avec votre module complémentaire chargé en tant que module complémentaire temporaire. Par défaut, surveille vos fichiers d'extension afin de détecter les modifications et recharge automatiquement votre extension lorsque les fichiers sont modifiés.

Modules complémentaires normaux

L'[installation d'extensions empaquetées](#) (c.-à-d. Le fichier `.xpi`) peut être une simple opération de glisser-déposer sur une fenêtre Firefox exécutant le profil dans lequel vous souhaitez l'installer. Ils peuvent également être installés directement en téléchargeant l'extension depuis [AMO](#) . Selon votre objectif (un profil, tous les profils, tous les utilisateurs, quel système d'exploitation, etc.), il existe d'autres options pour [installer des extensions](#) .

Ces autres options incluent divers répertoires en dehors du répertoire du profil dans lequel vous pouvez placer le fichier `.xpi` pour qu'il soit installé pour tous les utilisateurs d'une version particulière de Firefox ou de tous les profils d'un utilisateur particulier. Sous Windows, vous pouvez également installer une extension en ajoutant une clé au registre Windows. En général, ces autres répertoires ne sont pas ceux dans lesquels vous souhaitez installer un module complémentaire sur lequel vous écrivez actuellement. Cependant, ils peuvent être utilisés pour vous assurer que vous utilisez des modules complémentaires pour prendre en charge votre test / développement chargé dans tout profil Firefox ou version Firefox que vous utilisez. Par exemple, en plaçant un fichier `.xpi` dans `<Firefox install directory>/browser/extensions` vous pouvez avoir une extension disponible même dans le profil temporaire créé par `jpm run` (utilisé pour tester les extensions basées sur Firefox SDK).

Pour le [développement / test](#) , vous pouvez avoir l'extension dans n'importe quel répertoire de votre lecteur local en utilisant un [fichier proxy d'extension Firefox](#) (créez un fichier nommé `<em:id>` dans le *fichier install.rdf* pour Bootstrap / Restartless et Overlay / Legacy) dans le répertoire des *extensions* du profil contenant une ligne avec le chemin d'accès complet au répertoire contenant les fichiers de l'extension). Les extensions installées de cette manière seront presque toujours non signées (voir ci-dessous). Ainsi, cette méthode n'est pas très utile si vous souhaitez installer l'extension dans une version Release ou Beta de Firefox.

Limitations sur l'installation des add-ons normaux: Add-on Signing

Pour installer une extension en tant que module complémentaire normal dans les versions Release ou Beta de Firefox version supérieure ou égale à la version 48, l'extension *doit* être [signée par Mozilla](#) . Une extension est signée en la [soumettant à AMO](#) . Une fois signé,

l'extension peut être installée sur toute version de Firefox prise en charge. Pour les versions de Firefox antérieures à la version 43, il n'était pas nécessaire que les extensions soient signées par Mozilla. [Les types de modules complémentaires autres que les extensions](#) ne doivent pas nécessairement être signés par Mozilla.

Vous pouvez installer des extensions non signées en tant qu'add-ons normaux dans d'autres versions de Firefox (par exemple, [Firefox Developer Edition](#) , [Firefox Nightly](#) , [Unbranded Beta ou Unbranded Release](#)) en définissant `xpinstall.signatures.required` sur `false` dans `about:config` . La définition de cette option était également efficace dans les versions 43 et 48 de Release et Beta Firefox. La définition de cette option n'est pas effective dans les versions Release et Beta de Firefox à partir de la version 48.

Cependant, vous pouvez désactiver complètement la vérification des signatures dans Firefox sur toutes les versions, y compris la version. La réponse à la question Stack Overflow [Comment puis-je désactiver la vérification de signature pour les modules complémentaires Firefox?](#) décrit comment faire.

La nécessité d'installer des modules complémentaires non signés à des fins de développement a été considérablement réduite par la disponibilité des extensions d'installation en tant que [modules complémentaires temporaires](#) . Les modules complémentaires temporaires n'ont pas besoin d'être signés et peuvent être chargés dans n'importe quelle version actuelle de Firefox. Comme son nom l'indique, le principal inconvénient des modules complémentaires temporaires est qu'ils sont temporaires. Ils doivent être réinstallés à chaque redémarrage de Firefox. Cependant, il existe des cas d'utilisation où il est préférable d'installer un module complémentaire en tant que module complémentaire normal, mais non signé, plutôt qu'en tant que module complémentaire temporaire. Un module complémentaire doit être installé en tant que module complémentaire non signé s'il est nécessaire que le module reste installé même après le redémarrage de Firefox. Cela peut être souhaitable pour diverses raisons, y compris: des tests d'utilisation à plus long terme ou pour tester le fonctionnement du module complémentaire au démarrage de Firefox.

Add-ons temporaires (extensions uniquement)

Les extensions WebExtensions, les extensions basées sur SDK Firefox et les extensions Restartless / Bootstrap peuvent être installées en tant que [modules complémentaires temporaires](#) à partir `about:debugging` . Les extensions peuvent être chargées en tant que modules complémentaires temporaires dans toute version actuelle de Firefox. Il n'est pas nécessaire que les modules complémentaires temporaires soient signés et peuvent être chargés à partir de fichiers décompressés (un répertoire contenant des fichiers) ou emballés (par exemple, un fichier `.xpi`).

Pour plus d'informations sur l'installation temporaire d'extensions, voir [Installation d'un module complémentaire temporaire](#) .

Lire Démarrer avec firefox-addon en ligne: <https://riptutorial.com/fr/firefox->

[addon/topic/3235/demarrer-avec-firefox-addon](#)

Crédits

S. No	Chapitres	Contributeurs
1	Démarrer avec firefox-addon	Community , Makyen , Martin Zhai , nus , Ondřej Doněk , Priya