

 eBook Gratuit

APPRENEZ

v8

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

#v8

Table des matières

À propos	1
Chapitre 1: Démarrer avec v8	2
Remarques.....	2
Versions.....	2
v8 Communiqués.....	2
Exemples.....	2
Installation ou configuration.....	2
Chapitre 2: Démarrer avec v8	4
Exemples.....	4
Exécution de v8 sur un fichier.....	4
Fonctions et objets intégrés utiles dans d8.....	4
Chapitre 3: Faibles rappels	7
Remarques.....	7
Exemples.....	7
Exécution d'un code spécifié par l'utilisateur lorsqu'un objet est nettoyé.....	7
Crédits	9

À propos

You can share this PDF with anyone you feel could benefit from it, downloaded the latest version from: [v8](#)

It is an unofficial and free v8 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 v8.

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 v8

Remarques

Cette section fournit une vue d'ensemble de ce qu'est la v8 et de la raison pour laquelle un développeur peut vouloir l'utiliser.

Il devrait également mentionner tous les grands sujets de la v8 et établir un lien avec les sujets connexes. La documentation de la version 8 étant nouvelle, vous devrez peut-être créer des versions initiales de ces rubriques connexes.

Versions

v8 Communiqués

Version	Date de sortie
0.1.0	2008-07-03
1.0.0	2009-02-09
2.0.0	2009-11-18
3.0.0	2010-12-07
3.10.0	2012-10-30
3.20.0	2013-06-28
4.2.1	2015-01-14
4.10,1	2016-01-16

Exemples

Installation ou configuration

v8 utilise Google `depot_tools` pour obtenir la source et créer la bibliothèque. Pour installer `depot_tools`, suivez les instructions ici:

<https://www.chromium.org/developers/how-tos/install-depot-tools>

Obtenez la source v8 en exécutant

```
/path/to/depot_tools/fetch v8
```

Après avoir exécuté cette commande, pour extraire de nouvelles versions, exécutez

```
/path/to/depot_tools/gclient sync
```

Les instructions de construction sont fournies ici:

<https://github.com/v8/v8/wiki/Building%20with%20GN>

Lire Démarrer avec v8 en ligne: <https://riptutorial.com/fr/v8/topic/7639/demarrer-avec-v8>

Chapitre 2: Démarrer avec v8

Exemples

Exécution de v8 sur un fichier

Utilisez le shell d8 pour exécuter le moteur v8. Utilisez le modèle suivant pour exécuter sur un fichier:

```
/path/to/d8 [flags] [file].js
```

Par exemple:

```
./d8 --log-gc script.js
```

exécutera d8 sur script.js avec la journalisation de la récupération de place activée

Fonctions et objets intégrés utiles dans d8

En plus des bibliothèques définies dans la spécification du langage EcmaScript et de la spécification API de l'internationalisation EcmaScript, d8 implémente également les fonctions et objets suivants.

- `print(args...): function` . Imprimer sur `stdout` .
- `printErr(args...): function` . Imprimer à `stderr` .
- `write(args...): function` . Identique à l' `print` mais pas de nouvelle ligne à la fin.
- `read(filename): function` . Lisez le texte du fichier et retournez-le en tant que `String` .
- `readbuffer(filename): function` . Lisez le fichier binaire à partir du fichier et retournez-le sous le nom `ArrayBuffer` .
- `readline(): function` . Lisez la ligne de `stdin` . Utilisez `'\n'` pour la saisie de plusieurs lignes.
- `load(filename): function` . Chargez et exécutez le fichier JavaScript.
- `quit([exitCode]): function` . Quittez avec le code de sortie facultatif.
- `version(): function` . Renvoie le code de la version en tant que `String` .
- `os: object` . Utilitaires liés au système d'exploitation, uniquement disponibles pour POSIX.
 - `os.system(command): function` : Exécute la commande système.
 - `os.chdir(path): function` : Change le répertoire actuel.
 - `os.setenv(name, value): function` : Définit la variable d'environnement.
 - `os.unsetenv(name): function` : variable d'environnement non définie.
 - `os.umask(alue) function` : Appelle l'appel système `umask` et renvoie l'ancien `umask`.
 - `os.mkdirp(path[, mask]): function` : Crée un répertoire. Le masque (le cas échéant) est associé au masque actuel. Les répertoires intermédiaires sont créés si nécessaire.
 - `os.rmdir(path): function` : Supprimer le répertoire.
- `performance: object` : à utiliser pour l'analyse des performances.
 - `performance.now(): function` . Renvoie un horodatage en double, mesuré en millisecondes.

- `Worker: object` : modifié à partir de [HTML5 Web Worker](#) (voir exemple ci-dessous pour plus de détails)
- `Realm: object` : crée et gère un environnement isolé (domaine).
 - `Realm.create(): fonction` : `Realm.create(): fonction` un nouveau domaine avec un jeton de sécurité distinct et renvoie son index.
 - `Realm.createAllowCrossRealmAccess(): fonction` : Créez un nouveau domaine avec le même jeton de sécurité que le domaine actuel et retournez son index.
 - `Realm.current(): fonction` : Renvoie l'index du domaine actuellement actif. L'indice du domaine global est 0.
 - `Realm.global(i): fonction` : Retourne l'objet global du domaine i.
 - `Realm.owner(globalObj): fonction` : Renvoie l'index du domaine qui a créé `globalObj`.
 - `Realm.eval(i, s): fonction` : évalue `s` dans le domaine `i` et renvoie le résultat.
 - `Realm.switch(i): fonction` : Passe au domaine `i` pour les entrées interactives consécutives.
 - `Realm.dispose(i): fonction` : Supprime la référence au domaine `i`.
 - `Realm.shared: object` : Un accesseur pour une valeur partagée unique entre des domaines.

Des implémentations détaillées et des commentaires peuvent être trouvés dans [d8.h](#) , [d8.cc](#) , [d8-posix.cc](#) et [d8-windows.cc](#) .

Exemple:

```
print("Hello World!");
write("Hello ");
write("again!\n");
printErr("Nothing went wrong.");

write("Your name: ");
var name = readline();
print("Hello, ", name, "!");

load("external.js");

var string = read("text.txt");
var buffer = readbuffer("binary.bin");

print("Version: ", version());

quit(0); // bye
```

Exemple de travailleur:

main.js

```
var workerScript = read("worker.js");
var worker = new Worker(workerScript);
worker.postMessage(12);
```

travailleur.js

```
onmessage = ev => {
  print(ev); // 12
};
```

Exemple de domaine:

```
print(Realm.current()); // 0

var rIndex = Realm.create();
print(rIndex); // 1

Realm.eval(rIndex, "var x = 100");
Realm.eval(rIndex, "print(x)"); // 100
var result = Realm.eval(rIndex, "x * 2");
print(result); // 200

Realm.eval(rIndex, "var rIndex2 = Realm.create()");
Realm.eval(rIndex, "print(rIndex2)"); // 2
Realm.eval(rIndex, "print(Realm.owner(this))"); // 1

try {
  var childGlobal = Realm.global(rIndex);
  print(childGlobal); // error
} catch (e) {
  print("Global object cannot be read/written cross-realm.");
}

var rIndex3 = Realm.createAllowCrossRealmAccess();
Realm.eval(rIndex, "var x = 50");
var childGlobal = Realm.global(rIndex3);
childGlobal.x++;
Realm.eval(rIndex3, "print(x)"); // 51

try {
  Realm.dispose(rIndex3);
  Realm.eval(rIndex3, "print(x)"); // error
} catch (e) {
  print("The realm is dereferenced");
}

Realm.shared = "Hello from another world";
Realm.eval(rIndex, "print(Realm.shared)");
```

Lire Démarrer avec v8 en ligne: <https://riptutorial.com/fr/v8/topic/7745/demarrer-avec-v8>

Chapitre 3: Faibles rappels

Remarques

Les rappels faibles sont principalement utilisés pour nettoyer les objets C++ incorporés dans `InternalField` d'un `v8::Object` créé à partir d'un `v8::ObjectTemplate`. Lorsque l'objet JavaScript est nettoyé, l'objet C++ doit souvent être également supprimé. En définissant un faible rappel, vous pouvez être averti qu'un objet JavaScript a été collecté et prendre les mesures appropriées.

Il est *très* important de se rappeler que la récupération de la mémoire n'est *PAS* déterministe. Votre programme peut quitter avec des objets avec des rappels de référence faibles enregistrés qui ne sont jamais appelés. Ces rappels sont importants pour un programme de longue durée se comportant correctement, mais ne doivent pas être utilisés pour libérer des ressources de chemin critique de manière cohérente ou rapide.

Pour que le ramasse-miettes sache à quel moment il doit s'exécuter, vous devez indiquer la quantité d'espace que vos objets C++ utilisent via l'appel à `v8::Isolate::AdjustAmountOfExternalAllocatedMemory`. Le paramètre de cet appel est le *changement* en octets, de sorte que lorsque vous l'allouez, vous envoyez souvent `sizeof(T)` et lorsque vous nettoyez votre faible rappel de référence, vous envoyez `-sizeof(T)`.

Exemples

Exécution d'un code spécifié par l'utilisateur lorsqu'un objet est nettoyé.

```
/**
 * Runs user-specified code when the given javascript object is garbage collected
 */
template<class CALLBACK_FUNCTION>
void global_set_weak(v8::Isolate * isolate, const v8::Local<v8::Object> & javascript_object,
CALLBACK_FUNCTION function)
{
    struct SetWeakCallbackData{
        SetWeakCallbackData(CALLBACK_FUNCTION function, v8::Isolate * isolate, const
v8::Local<v8::Object> & javascript_object) :
            function(function) {
                this->global.Reset(isolate, javascript_object);
            }
        // function to call for cleanup
        CALLBACK_FUNCTION function;

        // this is the weak reference
        v8::Global<v8::Object> global;
    };

    // This must be dynamically allocated so it sticks around until the object
    // is garbage collected. It cleans itself up in the callback.
    auto callback_data = new SetWeakCallbackData(function, isolate, javascript_object);

    // set the callback on the javascript_object to be called when it's garbage collected
```

```
callback_data->global.template SetWeak<SetWeakCallbackData>(callback_data,
  [](const v8::WeakCallbackInfo<SetWeakCallbackData> & data) {
    SetWeakCallbackData * callback_data = data.GetParameter();
    callback_data->function(); // run user-specified code
    callback_data->global.Reset(); // free the V8 reference
    delete callback_data; // delete the heap variable so it isn't leaked
  }, v8::WeakCallbackType::kParameter);
}
```

Lire Faibles rappels en ligne: <https://riptutorial.com/fr/v8/topic/7655/faibles-rappels>

Crédits

S. No	Chapitres	Contributeurs
1	Démarrer avec v8	Community , Paul Sweatte , RamenChef , xaxxon
2	Faibles rappels	RamenChef , tbodt , xaxxon