

VSCODE PLUGIN 開発の心得

The screenshot shows the VS Code interface with the Marketplace extension search results displayed. The search bar at the top contains the text "Extension: Save your plugin — myyoutubevideolog". The results list includes the "Save your plugin" extension by pengfeit, which is currently selected. Other visible extensions include "Save Your Eyes", "Save As Snippet", "Gate plugin", "coala plugin for VS Code", "PocketMine Plugin AutoPac...", "Jira Plugin", and "Logitech Craft VS Code plugin".

Extension: Save your plugin — myyoutubevideolog

Code ファイル 編集 選択 表示 移動 デバッグ ターミナル ウィンドウ ヘルプ

拡張機能: MARKETPLACE

save your plugin

Save your plugin 0.0.9
Help you bakup your favorite plugin... pengfeit

Save Your Eyes 1.2.2
A theme to save your eyes alexlab

Save As Snippet 0.0.1
simply create your snippet zeakd

Gate plugin 0.1.0
Add gate development support to ... Sergiy Samborskiy

coala plugin for VS Code 0.0.20
Provides a unified interface for linting Coala Devs

PocketMine Plugin AutoPac... 0.0.1
Package your pocketmine plugins ... Ad5001

Jira Plugin 0.16.0
Jira integration for vscode gioboa

Logitech Craft VS Code plugin 1.0.1

Extension: Save your plugin pengfeit.save-your-plugin

pengfeit | 87 | ★★★★★ | リポジトリ | ライセンス

Help you bakup your favorite plugin on Cloud

無効にする ▾ アンインストール

詳細 コントリビューション 変更ログ

Save Your Plugin

v.0.0.9 by Pengfei Tang 2019.04.24

This Plugin can help you to save your plugin list on Cloud and Download your plugin list when you will set another computer.

There are three command you can use(Press Ctrl + Shift + P, and input that):

1. You can use this command to get plugin list which you installed in this computer.
`saveplugin:Get Plugin List from this computer`
2. Sync your plugin list from Cloud.If there are some new data in Cloud, that will install new plugin for you.

HOW TO START

vscode plugin development - Y X https://www.youtube.com/results?search_query=vscode+plugin+development

YouTube JP

vscode plugin development

ホーム フィルタ

急上昇

登録チャンネル

ライブラリ

履歴

後で見る

My Music

高く評価した動画

もっと見る

登録チャンネル

- 仙人小阿掌
- 四季萌芽
- 袁腾飞官方频道 ...
- 文曰小强——官 ...
- 键客行
- 無臉
- 李永乐老师

YOUTUBE の他のサービス

YouTube Premium

映画と番組

Creating A Simple VSCode Extension
Traversy Media 1.9万回視聴・6か月前
In this video we will explore what goes into creating a simple VSCode extension. We'll learn how to create a simple Bootstrap ...

Get Started Creating Extensions for Visual Studio Code
Mark Brummel 7542回視聴・1年前
This webinar explains how to get started creating Extensions for Visual Studio Code and publishing them to the Marketplace.

Creating Your First Visual Studio Code Extension
James Q Quick 1058回視聴・1か月前
When Visual Studio Code doesn't do what you want it to, you can always create your own extension! In this video, James shows you how to create a simple extension that does ...

VSCode - 10 Useful Extensions for Web Developers
GTCoding 3.4万回視聴・1年前
My Skillshare course (2 months free trial): <https://skl.sh/2DfX>
In this video, I'll introduce you to my favorite extensions for VS Code. These extensions will help you work more efficiently and effectively ...

Visual Studio Code Extensions
Microsoft Visual Studio 3万回視聴・3年前
Visual Studio Code now supports extension! In this session, Sean McBreen, Principal Program Manager, will demonstrate how to extend core functionality by ...

15 VS Code Extensions For Front-End Developers
Traversy Media 5.9万回視聴・1週間前
In this video, we'll explore 15 useful VS Code extensions for front-end developers. These extensions will help you work more efficiently and effectively ...

エクスプローラー

開いているエディター

- グループ1
 - package.json
- グループ2
- extension.js M

SAVE-YOUR-PLUGIN

- .vscode
- test
- .eslintrc.json
- .gitattributes
- .gitignore
- .vscodeignore
- CHANGELOG.md
- extension.js M
- icon.png
- jsconfig.json
- LICENSE
- package-lock.json
- package.json
- README.md
- vsc-extension-quickstart.md

アウトライン

- activate
 - [!] error
 - [!] getpluginlist_subscription
 - [!] loginorRegist_subscription
 - [!] removeAllPlugin_subscription
 - [!] rewritePluginlistfromThisWor...
 - deactivate
 - [!] getPluginList
 - [!] installPlugin
 - [!] removePlugin
 - [!] uploadyourplugin
 - [!] activate
 - [!] deactivate
 - [!] exec
 - [!] execSync
 - [!] request
 - [!] vscode

package.json

```
18 "activationEvents": [
19     "onCommand:extension.getPluginList",
20     "onCommand:extension.loginorRegist",
21     "onCommand:extension.rewritePluginListOnServer",
22     "onCommand:extension.removeAllPlugin"
23 ],
24 "main": "./extension",
25 "contributes": {
26     "commands": [
27         {
28             "command": "extension.getPluginList",
29             "title": "saveplugin:Get Plugin List from this computer"
30         },
31         {
32             "command": "extension.loginorRegist",
33             "title": "saveplugin:Download Your Plugin from Cloud Server"
34         },
35         {
36             "command": "extension.rewritePluginListOnServer",
37             "title": "saveplugin:Upload Your Plugin List Base from this computer"
38         },
39         {
40             "command": "extension.removeAllPlugin",
41         }
42     ]
43 }
```

extension.js

```
12 function activate(context) {
13
14     // Use the console to output diagnostic information (console.log) and errors (console.error)
15     // This line of code will only be executed once when your extension is activated
16     console.log('Congratulations, your extension "save-your-plugin" is now active!');
17
18     // Process Init for get Extension List
19     try {
20         process.chdir('bin');
21     } catch (error) {
22         console.log("Run from no windows");
23     }
24     console.log("CWD is Change:",process.cwd());
25
26
27     let getpluginlist_subscription = vscode.commands.registerCommand('extension.getPluginList', function(){
28         vscode.window.showInformationMessage(`You have ${getPluginList().length-1} plugins in this machine.`);
29     });
30     context.subscriptions.push(getpluginlist_subscription);
31
32     /**
33      *
34     */
```

エクスプローラー

- 開いているエディター
 - グループ 1
 - package.json
 - グループ 2
 - extension.js M
- SAVE-YOUR-PLUGIN
 - .vscode
 - test
 - .eslintrc.json
 - .gitattributes
 - .gitignore
 - .vscodeignore
 - CHANGELOG.md
 - extension.js M
 - icon.png
 - jsconfig.json
 - LICENSE
 - package-lock.json
 - package.json
 - README.md
 - vsc-extension-quickstart.md
- アウトライン
 - activate
 - error
 - getpluginlist_subscription
 - loginorRegist_subscription
 - removeAllPlugin_subscription
 - rewritePluginlistfromThisWor...
 - deactivate
 - getPluginList
 - installPlugin
 - removePlugin
 - uploadyourplugin
 - activate

package.json

```

18 "activationEvents": [
19   "onCommand:extens
20   "onCommand:extens
21   "onCommand:extens
22   "onCommand:extens
23 ],
24 "main": "./extension",
25 "contributes": {
26   "commands": [
27     {
28       "command": "extension.getPluginList",
29       "title": "saveplugin:Get Plugin List from this computer"
30     },
31     {
32       "command": "extension.loginorRegist",
33       "title": "saveplugin:Download Your Plugin from Cloud Server"
34     },
35     {
36       "command": "extension.rewritePluginListOnServer",
37       "title": "saveplugin:Upload Your Plugin List Base from this computer"
38     },
39     {
40       "command": "extension.removeAllPlugin",

```

最近使用したもの

- extension.getPlu... Aa Ab! * 1 / 2 件

extension.js

```

12 function activate(context) {
13
14   // Use the console to output diagnostic information (console.log) and errors (console.error)
15   // This line of code will only be executed once when your extension is activated
16   console.log('Congratulations, your extension "save-your-plugin" is now active!');
17
18   // Process Init for get Extension List
19   try {
20     process.chdir('bin');
21   } catch (error) {
22     console.log("Run from no windows");
23   }
24   console.log("CWD is Change:",process.cwd());
25
26
27   let getpluginlist_subscription = vscode.commands.registerCommand('extension.getPluginList', function(){
28     vscode.window.showInformationMessage(`You have ${getPluginList().length-1} plugins in this machine.`);

```

extension.getPlu... Aa Ab! * 1 / 1 件

extension.js

```
// Import the module and reference it with the alias vscode in your code below
const vscode = require('vscode');
// const path = require('path');
const request = require('request');
const execSync = require('child_process').execSync;
const exec = require('child_process').exec;

// this method is called when your extension is activated
// your extension is activated the very first time the command is executed
function activate(context) {
    // Use the console to output diagnostic information (console.log) and errors (console.error)
    // This line of code will only be executed once when your extension is activated
    console.log('Congratulations, your extension "save-your-plugin" is now active!');

    // Process Init for get Extension List
    try {
        process.chdir('bin');
    } catch (error) {
        console.log("Run from no windows");
    }
    console.log("CWD is Change:",process.cwd());

    let getpluginlist_subscription = vscode.commands.registerCommand('extension.getPluginList', function(){
        vscode.window.showInformationMessage(`You have ${getPluginList().length-1} plugins in this machine.`);
    });
    context.subscriptions.push(getpluginlist_subscription);

    /**
     */
    let loginorRegist_subscription = vscode.commands.registerCommand('extension.loginorRegist', function (){
        var URL = 'https://us-central1-vscode-save-your-plugin.cloudfunctions.net/logincreate';

        vscode.window.showInputBox({ placeHolder: "Please input your Email for Login or Register" }).then(useremail => {
            vscode.window.showInputBox({ placeHolder: "Please input your password for Login or Register" }).then(password => {
                request.post({
                    uri: URL,
                    headers: { "Content-type": "application/json" },
                    json: {
                        email:useremail,
                        password:password
                    }
                }, (err, res, data) => {
                    console.log(data);
                    if (data.hasOwnProperty('code')) {
                        vscode.window.showErrorMessage("Sorry, Register is failed. Please change other email and try again.");
                        return;
                    }
                })
            })
        })
    })
}

// deactivate
// executeCommand
// execSync
// request
// vscode
```

i You have 48 plugins in this machine.

```
.loginorRegist', function () {
  get/logincreate';
  "Login or Register" }).then(useremail => {
  "Password for Login or Register" }).then(password => {
```

通知BOXの使い方

```
ed. Please change other email and try again.");
```

 You have 48 plugins in this machine.

extension.js — save-your-plugin

エクスプローラー

開いているエディター

- グループ1
 - extension.js M
- グループ2
 - package.json

SAVE-YOUR-PLUGIN

- .vscode
- test
- .eslintrc.json
- .gitattributes
- .gitignore
- .vscodeignore
- CHANGELOG.md
- extension.js M
- icon.png
- {...} jsconfig.json
- LICENSE
- package-lock.json
- package.json
- README.md
- vsc-extension-quickstart.md

extension.js

```
1 // The module 'vscode' contains the VS Code extensibility API
2 // Import the module and reference it with the alias vscode
3 const vscode = require('vscode');
4 // const path = require('path');
5 const request = require('request');
6 const execSync = require('child_process').execSync;
7 const exec = require('child_process').exec;
8
9
10 // this method is called when your extension is activated
11 // your extension is activated the very first time the command
12 function activate(context) {
13     // Use the console to output diagnostic information (console.log) and errors (console.error)
14     // This line of code will only be executed once when your extension is activated
15     context.subscriptions.push(vscode.window.showInformationMessage(`Congratulations, your extension "save-your-plugin" has been activated`));
16     console.log('Congratulations, your extension "save-your-plugin" has been activated');
17
18     // Process Init for get Extension List
19     try {
20         process.chdir('bin');
21     } catch (error) {
22         console.log("Run from no windows");
23     }
24     console.log("CWD is Change:",process.cwd());
25
26     let getpluginlist_subscription = vscode.commands.registerCommand('extension.getpluginlist', () => {
27         vscode.window.showInformationMessage(`You have ${getpluginlist_subscription}`);
28     });
29     context.subscriptions.push(getpluginlist_subscription);
30
31     /**
32      * 
33      */
34     let loginorRegist_subscription = vscode.commands.registerCommand('extension.loginorRegist', () => {
35         var URL = 'https://us-central1-vscode-save-your-plugin.cloudfunctions.net';
36
37         vscode.window.showInputBox({ placeHolder: "Please input your email" });
38         vscode.window.showInputBox({ placeHolder: "Please input your password" });
39         request.post({
40             uri: URL,
41             form: {
42                 email: vscode.window.showInputBox({ placeHolder: "Please input your email" }),
43                 password: vscode.window.showInputBox({ placeHolder: "Please input your password" })
44             }
45         });
46     });
47     context.subscriptions.push(loginorRegist_subscription);
48
49     // This method is called when your extension is deactivated
50     context.subscriptions.push(vscode.commands.registerCommand('extension.deactivate', () => {
51         console.log('Your extension "save-your-plugin" has been deactivated');
52     }));
53 }
54
55
56 package.json
```

package.json

```
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60
```

Command : EXTENSION:CREATE YOUR PLUGIN

```
"title": "saveplugin:Upload Your Plugin List",
},
{
"command": "extension.removeAllPlugin",
"title": "saveplugin:Remove all plugin"
}
],
},
"scripts": {
    "postinstall": "node ./node_modules/vscode/bin/install",
    "test": "node ./node_modules/vscode/bin/test"
},
"devDependencies": {
    "typescript": "^2.6.1",
    "vscode": "^1.1.21",
    "eslint": "^4.11.0",
    "@types/node": "^8.10.25",
    "@types/mocha": "^2.2.42"
},
"dependencies": {
    "request": "^2.88.0"
}
```

...
...



エクスプローラー

開いているエディター

extension.js M

SAVE-YOUR-PLUGIN

.vscode

test

.eslintrc.json

.gitattributes

.gitignore

.vscodeignore

CHANGELOG.md

extension.js M

icon.png

jsconfig.json

LICENSE

package-lock.json

package.json

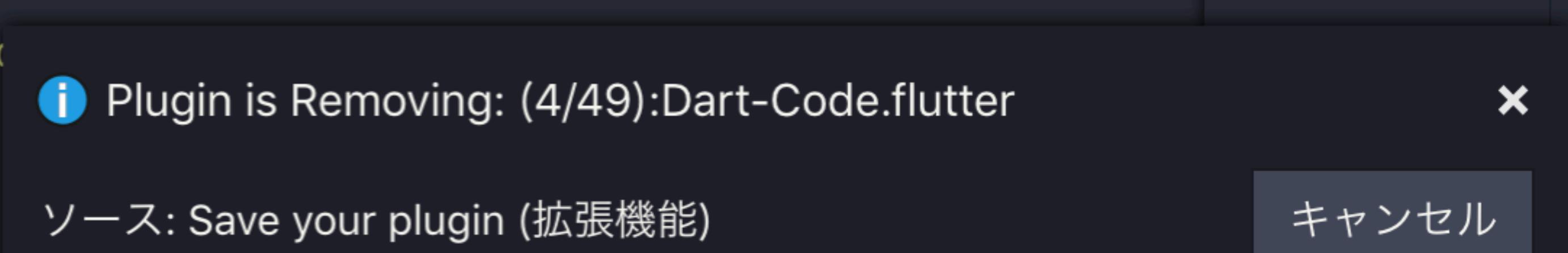
README.md

vsc-extension-quickstart.md

extension.js x

```
12 function activate(context) {  
13     // Use the console to output diagnostic information (console.log) and errors (console.error)  
14     // This line of code will only be executed once when your extension is activated  
15     console.log('Congratulations, your extension "save-your-plugin" is now active!');  
16  
17     // Process Init for get Extension List  
18     try {  
19         process.chdir('bin');  
20     } catch (error) {  
21         console.log("Run from no windows");  
22     }  
23  
24     console.log("CWD is Change:",process.cwd());  
25  
26  
27     let getpluginlist_subscription = vscode.commands.registerCommand('extension.getPluginList', function(){  
28         vscode.window.showInformationMessage(`You have ${getPluginList().length-1} plugins in this machine.`);  
29     });  
30     context.subscriptions.push(getpluginlist_subscription);  
31  
32     /**  
33      *  
34      */  
35     let loginorRegist_subscription = vscode.commands.registerCommand('extension.loginorRegist', function (){  
36         var URL = 'https://us-central1-vscode-save-your-plugin.cloudfunctions.net/logincreate';  
37  
38         vscode.window.showInputBox({ placeHolder: "Please input your Email for Login or Register" }).then(useremail =>  
39             vscode.window.showInputBox({ placeHolder: "Please input your password for Login or Register" }).then(password =>  
40                 request.post({  
41                     uri: URL,  
42                     headers: { "Content-type": "application/json" },  
43                     json: {  
44                         email:useremail,  
45                         password:password  
45                     }  
46                 })  
47             )  
48         );  
49         context.subscriptions.push(loginorRegist_subscription);  
50     }  
51 }  
52  
53 // This file can be edited directly by opening it in a text editor.  
54 //vscode.setThemeColor('background-color', '#000000');
```

```
sn(getpluginlist_subscription);  
  
option = vscode.commands.registerCommand('extension.loginorRegist', function (){  
  s-central1-vscode-save-your-plugin.cloudfunctions.net/logincreate';  
  
  inputBox({ placeHolder: "Please input your Email for Login or Register" }).then(useremail => {  
    nowInputBox({ placeHolder: "Please input your password for Login or Register" }).then(password => {  
      ({  
        ,  
        { "Co  
l:user  
word:password  
  
, data) => {  
  log(data);  
.hasOwnProperty('code')) {  
de.window.showErrorMessage("Sorry, Register is failed");  
rn;  
  
window.showInformationMessage("Login Successful: "+use  
ソース: Save your plugin (拡張機能)  
rn;
```



extension.js

```

62
63 let removeAllPlugin_subscription = vscode.commands.registerCommand('extension.removeAllPlugin', function(){
64     let pluginlist = getPluginList();
65     vscode.window.withProgress({
66         location: vscode.ProgressLocation.Notification,
67         title: "Plugin is Removing",
68         cancellable: true
69     }, async (progress, token) => {
70         token.onCancellationRequested(() => {
71             console.log("User canceled the long running operation")
72         });
73
74         progress.report({ increment: 0 });
75
76         for (let index = 0; index <= pluginlist.length; index++) {
77             //Forが終わったら、非同期操作を終わらせるための処理
78             if(index == pluginlist.length){
79                 return new Promise(resolve => resolve('Over'));
80             }
81             const element = pluginlist[index];
82
83             // await new Promise((resolve,reject)=>{
84             //     setTimeout(() => {
85                 //         progress.report({ increment: (index+1)*3/pluginlist.length, message: `I am long running! - still going
86                 //         //         resolve();
87                 //         //     }, 1000);
88                 // })
89                 console.log('Start-----');
90                 await removePlugin(element);
91                 progress.report({ increment: (index+1)*3/pluginlist.length, message: `(${index+1}/${pluginlist.length}):${element}` });
92                 console.log('End-----');
93
94             })
95
96         }
97         .then((val)=>{
98             vscode.window.showInformationMessage("All Plugin was Removed. Please Reload Window.", { modal: false }, 'Reload','Close')
99             .then(result => {
100                 if(result == "Reload"){
101                     console.log("Windows will reload");
102                     vscode.commands.executeCommand('workbench.action.reloadWindow');
103                 }
104             });
105         });
106
107     });
108 }
109 context.subscriptions.push(removeAllPlugin_subscription);
110

```

withProgress<R>(options: ProgressOptions, task: (progress: Progress<{increment: number, message: string}>, token: CancellationToken) => Thenable<R>): Thenable<R>

Show progress in the editor. Progress is shown while running the given callback and while the promise it returned isn't resolved nor rejected. The location at which progress should show (and other details) is defined via the passed `ProgressOptions`.

Parameter	Description
options: ProgressOptions	
task: (progress: Progress<{increment: number, message: string}>, token: CancellationToken) => Thenable<R>	A callback returning a promise. Progress state can be reported with the provided <code>progress</code> -object. To report discrete progress, use <code>increment</code> to indicate how much work has been completed. Each call with a <code>increment</code> value will be summed up and reflected as overall progress until 100% is reached (a value of e.g. <code>10</code> accounts for <code>10%</code> of work done). Note that currently only <code>ProgressLocation.Notification</code> is capable of showing discrete progress.
Returns	The thenable the task-callback returned.

```
155 function getPluginList(){
156     var result = execSync('code --list-extensions',{cwd:process.cwd()});
157     return result.toString().split('\n');
158 }
159
160 // --install-extension <extension-id>
161 function installPlugin(str_extension_id = null){
162     return new Promise(function(resolve,reject){
163         exec('code --install-extension '+str_extension_id,{cwd:process.cwd()}, function (error, stdout, stderr) {
164             if(stderr){
165                 console.log('stderr: ' + stderr.toString());
166                 resolve(stderr.toString());
167                 return;
168             }
169             if (error !== null) {
170                 console.log('Exec error: ' + error.toString());
171                 resolve(error.toString());
172                 return;
173             }
174             console.log('stdout: ' + stdout.toString());
175             resolve(stdout.toString());
176         });
177     });
178 }
179
180
181 // --uninstall-extension <extension-id>
182 function removePlugin(str_extension_id = null){
183     return new Promise(function(resolve,reject){
184         if(str_extension_id != 'pengfeit.save-your-plugin'){
185             console.log("str_extension_id:",str_extension_id);
186             exec('code --uninstall-extension '+str_extension_id,{cwd:process.cwd()}, function (error, stdout, stderr) {
187                 if(stderr){
188                     console.log('stderr: ' + stderr.toString());
189                     resolve(stderr.toString());
190                     return;
191                 }
192                 if (error !== null) {
193                     console.log('Exec error: ' + error.toString());
194                     resolve(error.toString());
195                     return;
196                 }
197                 console.log('stdout: ' + stdout.toString());
198                 resolve(stdout.toString());
199             });
200         }
201     });
202     else{
203         resolve("Save your plugin");
204     }
205 }
206 }
```

```
ess.report({ increment: 0 });

let index = 0; index <= pluginlist.length; index++) {
    //Forが終わったら、非同期操作を終わらせるための処理
    if(index == pluginlist.length){
        return new Promise(resolve => resolve('Over'));
    }

    const element = pluginlist[index];

    await new Promise((resolve, reject)=>{
        setTimeout(() => {
            })
    })

    console.log('Start-----');
    await removePlugin(element);
    progress.report({ increment: (index+1)*3/pluginlist.length, message: `(${index+1})/${pluginlist.length}):${element}`);
    console.log('End-----');
}

) => {
    e.window.showInformationMessage("All Plugin was Removed. Please");
    result => {

```

VSCODEを再起動するBOX

i Plugin is installed. Please Reload Window.

ソース: Save your plugin (拡張機能)

Reload Close

extension.js

```

62
63 let removeAllPlugin_subscription = vscode.commands.registerCommand('extension.removeAllPlugin', function(){
64     let pluginlist = getPluginList();
65     vscode.window.withProgress({
66         location: vscode.ProgressLocation.Notification,
67         title: "Plugin is Removing",
68         cancellable: true
69     }, async (progress, token) => {
70         token.onCancellationRequested(() => {
71             console.log("User canceled the long running operation")
72         });
73
74         progress.report({ increment: 0 });
75
76         for (let index = 0; index <= pluginlist.length; index++) {
77             //Forが終わったら、非同期操作を終わらせるための処理
78             if(index == pluginlist.length){
79                 return new Promise(resolve => resolve('Over'));
80             }
81             const element = pluginlist[index];
82
83             // await new Promise((resolve,reject)=>{
84             //     setTimeout(() => {
85                 //         progress.report({ increment: (index+1)*3/pluginlist.length, message: `I am long running! - still going
86                 //         //         resolve();
87                 //         //     }, 1000);
88                 // })
89                 console.log('Start-----');
90                 await removePlugin(element);
91                 progress.report({ increment: (index+1)*3/pluginlist.length, message: `(${index+1}/${pluginlist.length}):${element}` });
92                 console.log('End-----');
93
94             })
95
96         }
97         .then((val)=>{
98             vscode.window.showInformationMessage("All Plugin was Removed. Please Reload Window.", { modal: false }, 'Reload','Close')
99             .then(result => {
100                 if(result == "Reload"){
101                     console.log("Windows will reload");
102                     vscode.commands.executeCommand('workbench.action.reloadWindow');
103                 }
104             });
105         });
106
107     });
108 }
109 context.subscriptions.push(removeAllPlugin_subscription);
110

```

withProgress<R>(options: ProgressOptions, task: (progress: Progress<{increment: number, message: string}>, token: CancellationToken) => Thenable<R>): Thenable<R>

Show progress in the editor. Progress is shown while running the given callback and while the promise it returned isn't resolved nor rejected. The location at which progress should show (and other details) is defined via the passed `ProgressOptions`.

Parameter	Description
options: ProgressOptions	
task: (progress: Progress<{increment: number, message: string}>, token: CancellationToken) => Thenable<R>	A callback returning a promise. Progress state can be reported with the provided <code>progress</code> -object. To report discrete progress, use <code>increment</code> to indicate how much work has been completed. Each call with a <code>increment</code> value will be summed up and reflected as overall progress until 100% is reached (a value of e.g. <code>10</code> accounts for <code>10%</code> of work done). Note that currently only <code>ProgressLocation.Notification</code> is capable of showing discrete progress.
token: CancellationToken	To monitor if the operation has been cancelled by the user, use the provided <code>CancellationToken</code> . Note that currently only <code>ProgressLocation.Notification</code> is supporting to show a cancel button to cancel the long running operation.
Returns	Thenable<R>
	The thenable the task-callback returned.

最後：VSCodeストアへ登録する

A screenshot of the VSCode interface. The left sidebar shows a tree view of files and folders. The main editor area has two tabs open: 'package.json' and 'README.md'. The 'package.json' tab contains the following JSON code:

```
1  {  
2    "name": "save-your-plugin",  
3    "displayName": "Save your plugin",  
4    "publisher": "pengfeit",  
5    "description": "Help you backup your favorite plugin on Cloud",  
6    "icon": "icon.png",  
7    "version": "0.0.9",  
8    "engines": {  
9      "vscode": "^1.29.0"  
10     },  
11    "repository": {  
12      "type": "git",  
13      "url": "https://github.com/FlySnowIII/save-your-plugin.git"  
14    },  
15    "categories": [  
16      "Other"  
17    ],  
18    "activationEvents": [  
19      "onCommand:extension.getPluginList",  
20      "onCommand:extension.loginorRegist",  
21      "onCommand:extension.rewritePluginListOnServer".  
22    ]  
23  }  
24  
25  # Save Your Plugin  
26  
27 v.0.0.9 by Pengfei Tang 2019.04.24  
28  
29 This Plugin can help you to save your plugin list on Cloud and Download your plu  
30
```

```
sudo npm install -g vsce
```

```
vsce package --yarn
```

The screenshot shows a browser window displaying the Microsoft Docs page for the Publishing Extension. The URL is <https://docs.microsoft.com/en-us/api/working-with-extensions/publishing-extension>. The page header includes the Microsoft logo and navigation links for Code, Docs, Updates, Blog, API, Extensions, and FAQ.

The main content features a heading 'Publishing Extension' with an 'Edit' button. Below it, a paragraph explains the purpose of publishing extensions:

Once you have made a high-quality extension, you can publish it to the [VS Code Extension Marketplace](#) so others can find, download and use your extension. Alternatively, you can [package](#) an extension into the installable VSIX format and share it with other users.

The 'This topics covers:' section lists the following:

- Using `vsce`, the CLI tool for managing VS Code extensions
- Packaging, publishing and [unpublishing](#) extensions
- Registering a `publisherId` necessary for publishing extensions

VSCE

`vsce`, short for "Visual Studio Code Extensions", is a command line tool for packaging, publishing and managing VS Code extensions.

Installation

Make sure you have `Node.js` installed. Then run:

```
npm install -g vsce
```

Usage

You can use `vsce` to easily [package](#) and [publish](#) your extensions:

```
$ cd myExtension  
$ vsce package  
# myExtension.vsix generated  
$ vsce publish  
# <publisherID>.myExtension published to VS Code MarketPlace
```

追加：ES6で二つの数組を計算する

```
209 function uploadyourplugin(dataObj){  
210   var localpluginlist = getPluginList();  
211   let firebasepluginlist = dataObj.data;  
212   let difference = firebasepluginlist.concat(localpluginlist).filter(v => firebasepluginlist.includes(v) && !localpluginlist.includes(v)) // [1,3]  
213   let union = firebasepluginlist.concat(localpluginlist.filter(v => !firebasepluginlist.includes(v)))  
214   difference.splice(difference.findIndex(v => v === ""), 1);  
215   union.splice(union.findIndex(v => v === ""), 1);  
216  
217   //2. Firebaseとローカルのプラグインリストを照合し、もし差分がなかったらプログラムを中止する  
218   if(difference.length == 0){  
219     vscode.window.showInformationMessage("Your workplace is the newest.");  
220     return ;  
221   }  
222 }
```

Thank you for your time and attention.

<https://github.com/FlySnowIII/save-your-plugin>