

```

1 ///////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////
2 //
3 // 預設 worker 會彈性隨效能能在(max_workers, min_workers)之間調整數量
4 //
5 // 但若不想 worker 的數量自動彈性調整，可以設定(max_workers = min_workers)
6 //
7 ///////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////
8
9
10
11 const defaultSetting = {};
12
13 (function () {
14     const $d = defaultSetting;
15
16     // 同時能運行最大的 worker 數量
17     $d.max_workers = 2;
18
19     // idle 時要維持幾個 workers 待命
20     $d.min_workers = 0;
21
22     // 當 worker 沒任務可接時
23     // 閒置多久後會被銷毀
24     $d.idleTime = (1000 * 30);
25
26     // (underscore, lodash) url
27     $d.sourceScriptPath;
28
29     // extension1.url
30     $d.extension1ScriptPath;
31
32     // 要被 import 的 script
33     $d.importScriptList = [];
34 })();
35 //-----
36 const SettingProxy = {
37     addImportScript: function (script) {
38
39         if (!Array.isArray(script)) {
40             script = [script];
41         }
42         script.forEach(function (s) {
43             defaultSetting.importScriptList.push(s);
44         });
45     },
46 };
47
48 SettingProxy.GModules = {};
49
50 export { SettingProxy as workerSettings };
51
52
53 (function () {
54
55     Object.defineProperty(SettingProxy, 'max_workers', {
56         enumerable: true,
57         configurable: false,
58         // writable: false,
59         get: function () {
60             return defaultSetting.max_workers;
61         },
62         set: function (count) {

```

```

63
64         if (count < defaultSetting.min_workers) {
65             throw new Error('worker maxWorkersCount <
minWorkersCount(${defaultSetting.min_workers})');
66         }
67
68         if (count < 1) {
69             errorList.push('max_workers must >= 1');
70         }
71
72         defaultSetting.max_workers = count;
73     }
74 });
75
76 //-----
77 Object.defineProperty(SettingProxy, 'min_workers', {
78     enumerable: true,
79     configurable: false,
80     // writable: false,
81     get: function () {
82         return defaultSetting.min_workers;
83     },
84     set: function (count) {
85         if (count > defaultSetting.max_workers) {
86             throw new Error('workers count >
maxWorkersNum(${defaultSetting.max_workers})');
87         }
88         workerSettings.min_workers = count;
89     }
90 });
91 //-----
92 Object.defineProperty(SettingProxy, 'idleTime', {
93     enumerable: true,
94     configurable: false,
95     // writable: false,
96     get: function () {
97         return defaultSetting.idleTime;
98     },
99     set: function (time) {
100
101         if (time < 0) {
102             throw new Error("idleTime must be >= 0");
103         }
104         defaultSetting.idleTime = time;
105     }
106 });
107 //-----
108 Object.defineProperty(SettingProxy, 'sourceScriptPath', {
109     enumerable: true,
110     configurable: false,
111     // writable: false,
112     get: function () {
113         const root = SettingProxy.GModules["root"];
114         const _ = root.GModules["_"];
115
116         let info = _.$extension1.info;
117
118         let path = (defaultSetting.sourceScriptPath != null) ?
119             defaultSetting.sourceScriptPath : info.sourceScriptPath;
120
121         return path;
122     },

```

```
123     set: function (path) {
124         defaultSetting.sourceScriptPath = path;
125     }
126 });
127 //-----
128 Object.defineProperty(SettingProxy, 'extension1ScriptPath', {
129     enumerable: true,
130     configurable: false,
131     // writable: false,
132     get: function () {
133         // debugger;
134         const root = SettingProxy.GModules["root"];
135         const _ = root.GModules["_"];
136
137         let info = _.$extension1.info;
138
139         let path = (defaultSetting.extension1ScriptPath != null) ?
140             defaultSetting.extension1ScriptPath : info.extension1ScriptPath;
141
142         return path;
143     },
144     set: function (path) {
145         defaultSetting.extension1Path = path;
146     }
147 });
148
149 //-----
150 Object.defineProperty(SettingProxy, 'importScriptList', {
151     enumerable: true,
152     configurable: false,
153     // writable: false,
154     get: function () {
155         return defaultSetting.importScriptList;
156     },
157     set: function () {
158
159     }
160 });
161
162 }());
```