

Web Workerと、それを便利にするComlinkライブラリの紹介

Web Workerとは

- 従来シングルスレッドのみだったWebブラウザにマルチスレッドを持ち込む仕組み
 - 元々Webのスクリプト用途にはシングルスレッドで十分と考えられていた
 - リッチなWebアプリやブラウザゲームなど高パフォーマンスを要求するようになった
- DOMを処理しているスレッドを妨げずに高負荷な処理を実行できる
- 基本的には一対一のメッセージチャネルを介してのみスレッド間通信を行う

欠点

- メッセージチャンネルが一つあるだけなので通信が面倒
 - RPC的なことをやりたい場合自分で実装しないといけない
 - SharedArrayBufferで共有メモリを作れるがより複雑に
- 構造化複製可能な型しかやり取りできない
 - 完全に等価ではないが、 $f(x) = \text{JSON.parse}(\text{JSON.serialize}(x))$ において不動点である x は転送可能だと思えばいい

Comlinkとは

- Web Workerをenjoyableにするライブラリ
- 双方のスレッドが持つオブジェクトを簡単に公開できる
 - 構造化複製不可能なものもOK
 - function, class, ...
 - Proxyで包むことによって転送可能にしている
 - これらはオブジェクトの所有権を相手スレッド側に移さない
- もちろん所有権移動を伴うコピーや、大きなArrayBuffer等をコピーせずに移譲することも可能(C++で言うところのstd::move)
- TypeScript対応！(generics周りがちょっと弱め)

試してみる

```
$ npm install comlink
```

worker.js

```
import { expose } from 'comlink';  
function add(a, b) {  
  return a + b;  
}  
expose(add);
```

main.js

```
import { wrap } from 'comlink';  
const worker = new Worker(new URL('./worker.js', import.meta.url), {  
  type: 'module',  
});  
const add = wrap(worker);  
const result = await add(3, 4);
```