

Promise Error Handling

約束されたエラー

Promiseとエラーの基本

- Promise内で起きた例外は自動でキャッチされる
- エラー処理は`.catch(fn)`で行う

```
var promise = new Promise(function(){
    throw new Error("例外");
});
promise.catch(function(error){
    // 例外をキャッチできる
});
```

よくある問題

- `.catch(fn)` をしないとエラーログも出せない
- `.catch(fn)` をしわすれてエラーの握りつぶしが発生する
- = **unhandled rejection** (`.catch`をしてないpromise)
- [4.6. Promise.prototype.done とは何か?](#)

現状のunhandled rejectionへの対応

- **unhandled rejection**が発生した時にコンソールに出すかは実装依存
- FirefoxはGCのタイミング
- Chromeは開発者ツールが有効の場合
 - [v8-git-mirror/promise.js at 3709b9254e9d054796f6735b0f5cefed65ce69d3 · v8/v8-git-mirror](https://v8.dev/docs/faq#unhandled-rejection)

今日のテーマ: **unhandledRejection**

- unhandled rejectionは予期せぬ出来事
- 少なくともエラーのログは取りたい
 - `Promise.prototype.done`を実装するのは本質的ではない
- `'unhandledRejection'`, `'rejectionHandled'` というイベントの実装が進められているという話
- ECMAScript 6の仕様ではありません

unhandledRejection / rejectionHandled

bluebirdが主導してるの例はbluebird

unhandledRejection イベント

```
var bluebird = require("bluebird");
process.on("unhandledRejection", function (reason, promise) {
  console.log("unhandledRejection");
});
```

```
var resolved = bluebird.resolve();
resolved.then(function () {
  throw new Error("Yay!");
});
```

unhandledRejection イベント

- catch してないPromiseでエラーが発生すると発行されるイベント
- .catch してないpromiseオブジェクトを見つけるのを助ける
- window.onerror みたいなもの

```
process.on("unhandledRejection", function (reason, promise) {  
    // エラー理由とpromiseがやってくる  
});
```


rejectionHandled イベント

```
var Promise = require("bluebird");
process.on("rejectionHandled", function (promise) {
  console.log("rejectionHandled");
});
var rejected = Promise.reject(new Error("Error Promise"));
setTimeout(function () {
  rejected.catch(function () {
    // rejected済みのpromiseに`catch`する
  });
}, 100);
```

rejectionHandled イベント

- rejected済みのpromiseにcatchした時に起きるイベント
- 呼ばれることがないcatchの発見に役立つ

実際の使い方

- unhandledRejectionのログを取りたい場合

```
var unhandledRejections = new Set();
process.on('unhandledRejection', function(reason, p) {
  unhandledRejections.add(p);
});
process.on('rejectionHandled', function(p) {
  unhandledRejections.delete(p);
});
```

unhandledRejection & rejectionHandled

- なぜunhandledRejectionだけ欲しいのに rejectionHandledも見ろの？
- => rejectionHandled が起きるケースは unhandledRejectionが先に起きてる事がある

unhandledRejection & rejectionHandled パターン

```
var rejected = Promise.reject();
setTimeout(()=>{
  // 2. rejectionHandled イベント
  rejected.catch(()=>{});
}, 100);
// 1. unhandledRejection イベント
```

unhandledRejection & rejectionHandled

- unhandledRejection と rejectionHandled は基本セットで使う
- rejectionHandled単体の使い道はあんまりなさそう？

使い方とドキュメント

参考資料

- [process io.js Manual & Documentation](#)
- [Promise unhandled rejection tracking global handler hook](#)
- [Global rejection events - bluebird](#)

実装

ことのはじまり

- bluebirdの実装提案
 - Difficult to get onPossiblyUnhandledRejection to work all of the time in Node.js due to submodules · Issue #357 · petkaantonov/bluebird
- @benjamingrさんが色々利用状況を調べてプロポーサルを書いた
- Promise unhandled rejection tracking global handler hook

ライブラリの実装

- [bluebird v2.7.0](#)で実装
- [when.js v3.7.0](#)で実装
- [io.js v1.4.1](#)で実装 by [@petkaantonov](#)
 - [Consider exposing promise unhandled rejection hook · Issue #256 · iojs/io.js](#)
 - [Implement unhandled rejection tracking by petkaantonov · Pull Request #758 · iojs/io.js](#)

小さいプロポーサルからの実装

- Promise/A+の頃から同じような話があった
 - [Library hooks · Issue #3 · promises-aplus/unhandled-rejections-spec](#)
- DOM/ECMAScript Promiseでも話があった程度
 - 実際に仕様としては入ることはなかった
 - [\[whatwg\] An API for unhandled promise rejections from Domenic Denicola on 2014-09-12 \(public-whatwg-](#)

Implementation in userland

Implementation in userland

– Consider exposing promise unhandled rejection hook · Issue #256 · iojs/io.js

Implementation in userland

- ユーザランドでの実装から始まっている面白い動き
 - Promise自体もコミュニティ仕様からECMAScript仕様に入った
- io.jsにも入ったため、他のPromiseライブラリにも実装が進んでいきそうな空気がある
 - コミュニティ標準から仕様へ？

まとめ

- Promiseでエラーの握りつぶしがよく起きてる
- 現状ではunhandled rejectionの扱いは実装依存
- unhandled rejectionが起きた時に発行するイベントを定義したコミュニティプロポーサルがでた
- bluebirdやio.jsなどで実装された
- ECMAScript仕様の話はまだない