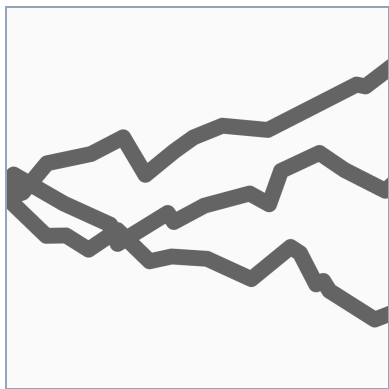


# npmパッケージじゃない仕組みで 共有ライブラリを管理する

Webフロントエンドを軸に、幅を広げたエンジニアたちの仕事

# yuhei (@\_yuheiy)

プレイド / デザインエンジニア



所属:

- Developer Experience & Performance
- デザインシステム

plaidev/karte-io-systems/

- ├── systems/
  - │ ├── academy2/
  - │ ├── action-editor/
  - │ ├── action-table/
  - │ ├── admin/
  - │ ├── apiv2/
  - │ ├── baisu/
  - │ ├── **communication/**
  - │ ├── craft/
  - │ ├── datahub/
  - │ ├── demo-sites/
  - │ ├── developers/
  - │ ├── edge/
  - │ ├── envelope-service/
  - │ └── (略。あと30ディレクトリくらいある)
- └── README.md

plaidev/karte-io-systems/

└─ systems/

└─ communication/

└─ packages/

└─ front-react/

└─ package.json

└─ front/

└─ package.json

└─ lib/

└─ package.json

└─ web/

└─ package.json

└─ package.json

```
const EVENT_NAMES = [  
  {  
    id: 'view',  
    event_name: '閲覧',  
    tag_name: '閲覧',  
    is_auto_send: false,  
  },  
  {  
    id: 'buy',  
    event_name: '購入',  
    tag_name: '購入',  
    is_auto_send: false,  
  },  
  {  
    id: 'identify',  
    event_name: 'ユーザー情報',  
    tag_name: 'ユーザー情報',  
    is_auto_send: false,  
  },  
]
```

systems/communication/packages/web/package.json :

```
{  
  "name": "communication/web",  
  "dependencies": {  
    "@plaidev/nodejs-constants": "x.x.x"  
  }  
}
```

systems/communication/packages/front/package.json :

```
{  
  "name": "communication/front",  
  "dependencies": {  
    "@plaidev/frontend-constants": "x.x.x"  
  }  
}
```

chore(deps): Update Packages @plaidv/frontend-constants:3.19.1 @plaidv/nodejs-constants:6.1.10 #86223

Edit <> Code

Merged merged 1 commit into develop from feature/update-package-1689134653 on Jul 12, 2023

Conversation 0 Commits 1 Checks 0 Files changed 25 +22,381 -22,836

Changes from all commits File filter Conversations 0 / 25 files viewed Review in codespace Review changes

Filter changed files

systems

admin/packages

common

package-lock.json

package.json

front-react

package-lock.json

karte-house

package-lock.json

package.json

migration-batch

package-lock.json

package.json

navigation-front

package-lock.json

package.json

navigation-rpc

package-lock.json

package.json

navigation-v5

package-lock.json

package.json

masspush/packages

rpc

package-lock.json

package.json

2,924 systems/admin/packages/common/package-lock.json

Load diff

Large diffs are not rendered by default.

2 systems/admin/packages/common/package.json

@@ -25,7 +25,7 @@

25 "license": "ISC",

26 "dependencies": {

27 "@plaidv/config": "5.0.1",

28 - "@plaidv/frontend-constants": "3.7.0",

28 + "@plaidv/frontend-constants": "3.19.1",

29 "@plaidv/logger": "9.9.0",

30 "@plaidv/mongo-client": "7.0.0"

31 },

6,116 systems/admin/packages/front-react/package-lock.json

Load diff

Large diffs are not rendered by default.

11,074 systems/admin/packages/karte-house/package-lock.json

Load diff

Large diffs are not rendered by default.

2 systems/admin/packages/karte-house/package.json

Load diff

Large diffs are not rendered by default.

**定数ファイルを  
単純にコピーして配布する**



# 1. JSONファイルを生成する

```
plaidev/karte-io-systems/
```

```
|— shared-constants/
```

```
| |— __generated__/
```

```
| | |— common/products.json
```

```
| | |— ...
```

```
| |— scripts/
```

```
| | |— build.ts
```

```
| |— src/
```

```
| | |— index.ts
```

```
| |— sync-config.ts
```

```
|— systems/
```

```
| |— communication/packages/
```

```
| | |— front/src/generated-shared-constants/
```

```
| | |— web/src/generated-shared-constants/
```

```
| |— craft/packages/
```

## 2. 設定に応じて対象ディレクトリへ一斉にコピー

```
plaidev/karte-io-systems/  
├─ shared-constants/  
│   ├── __generated__/  
│   │   ├── common/products.json  
│   │   └── ...  
│   ├── scripts/  
│   │   └── build.ts  
│   ├── src/  
│   │   └── index.ts  
│   └── sync-config.ts  
└─ systems/  
    ├── communication/packages/  
    │   ├── front/src/generated-shared-constants/  
    │   └── web/src/generated-shared-constants/  
    └── craft/packages/
```

```
import type { SyncConfig } from './scripts/lib/sync-config';

const config: SyncConfig = [
  {
    name: 'communication/front',
    needs: [
      'common/admin_role',
      'common/chat_status',
      'common/event_name',
      'common/products',
      // preserve from formatting
    ],
    outputDir:
      '../systems/communication/packages/front/src/generated-shared-constants',
    outputExt: 'json',
  },
  {
    name: 'communication/front-react',
```

## 2. 設定に応じて対象ディレクトリへ一斉にコピー

```
plaidev/karte-io-systems/
```

```
|— shared-constants/
|   |— __generated__/
|   |   |— common/products.json
|   |   |— ...
|   |— scripts/
|   |   |— build.ts
|   |— src/
|   |   |— index.ts
|   |— sync-config.ts
|— systems/
|   |— communication/packages/
|   |   |— front/src/generated-shared-constants/
|   |   |— web/src/generated-shared-constants/
|   |— craft/packages/
```

```
import type { SyncConfig } from './scripts/lib/sync-config';

const config: SyncConfig = [
  {
    name: 'communication/front',
    needs: [
      'common/admin_role',
      'common/chat_status',
      'common/event_name',
      'common/products',
      // preserve from formatting
    ],
    outputDir: ' ../systems/communication/packages/front/src/generated-shared-cor
    outputExt: 'json',
  },
  {
    name: 'communication/front-react',
    needs: [
```

### 3. 対象の定数ファイルだけがコピーされる

```
plaidev/karte-io-systems/  
└─ systems/  
    └─ communication/packages/  
        └─ front/src/generated-shared-constants/  
            ├── common/  
            │   ├── admin_role.json  
            │   ├── chat_status.json  
            │   ├── event_name.json  
            │   └─ products.json  
            ├── .gitattributes  
            └─ README.md
```

## 4. git commitする

```
$ cd plaidev/karte-io-systems/shared-constants  
$ npm run build  
$ git add --all  
$ git commit -m "Update shared-constants"
```

# Pros

- すべての依存箇所を確実にアップデートできる
- publishの手順を経由せずにテストできる
- 定数の利用箇所が明確になる
- 変更履歴を利用者側から追跡しやすくなる



# publishの手順を経由せずにテストできる

- パッケージ単体では意味のあるテストができないので、利用側に適用してテストするのが妥当
- 新しい仕組みでは、パッケージをリリースしないでも動作確認ができる
- pull requestを作成したタイミングで、影響する別システムのCIも実行される

```
name: "[communication] CI"
on:
  pull_request:
    paths:
      - systems/communication/**
```

# 定数の利用箇所が明確になる

- 影響箇所を意識しながら変更できるので、不用意な問題の混入を防げる

```
yuhei.yasuda@plaid-yuhei-yasuda shared-constants % npm run build
```

```
> shared-constants@0.0.0 build
```

```
> tsx scripts/build.ts
```

(中略)

Constants Scope	Needed By
common/admin_role	<ul style="list-style-type: none"><li>- fuzzy-adventure</li><li>- glowing-octo-computing-machine</li><li>- legendary-enigma</li><li>- literate-train</li><li>- miniature-parakeet</li><li>- shiny-robot</li><li>- special-octo-giggle</li></ul>
common/chat_status	<ul style="list-style-type: none"><li>- redesigned-goggles</li><li>- upgraded-octo-telegram</li></ul>
common/contract_status	<ul style="list-style-type: none"><li>- fantastic-happiness</li></ul>
common/data_types	<ul style="list-style-type: none"><li>- animated-tribble</li><li>- congenial-goggles</li><li>- friendly-enigma</li><li>- ideal-adventure</li><li>- psychic-engine</li><li>- redesigned-octo-system</li><li>- refactored-memory</li><li>- super-memory</li><li>- upgraded-telegram</li></ul>

# 変更履歴を利用者側から追跡しやすくなる

- 定数が変更された経緯をgitで調べられる
- 自分に関係のない定数の変更は無視できる

# Cons

- パッケージからの移行に手間がかかる
- ビルドされたファイルが確実にコミットされることを保証できない
- pull requestのdiffがうるさい
- 外部リポジトリと連携できない
- 独自の仕組みを運用できるか懸念がある

# Cons

- パッケージからの移行に手間がかかる
  - ➡ 移行用のスクリプトを書いて自動化する
- ビルドされたファイルが確実にコミットされることを保証できない
  - ➡ 保証する仕組みを作る
- pull requestのdiffがうるさい
  - ➡ ノイズを軽減する
- 外部リポジトリと連携できない
  - ➡ 連携する仕組みを作る
- 独自の仕組みを運用できるか懸念がある
  - ➡ 周知 & ドキュメントでカバーする

後日ブログ書きます

# パッケージからの移行プロセス

- パッケージはおよそ1200ファイルから参照されていた
- 移行用スクリプトを実装して作業を自動化
  - パッケージを参照するパスを書き換え

```
import { PRODUCT_NAMES } from '@plaidev/nodejs-constants/build/products.json';  
//                                     ↑ Before / After ↓  
import { PRODUCT_NAMES } from './path/to/generated-shared-constants/common/products.json';
```

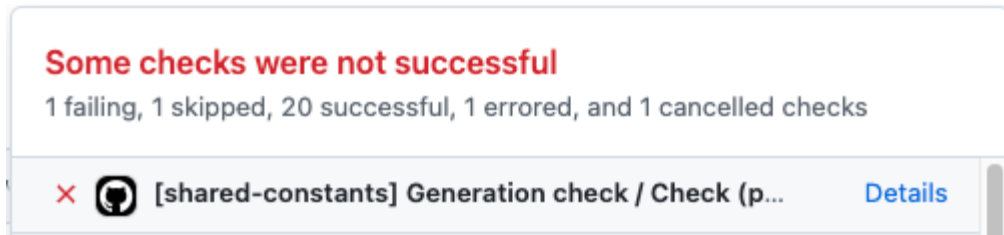
- システムごとに必要な設定を生成

```
{  
  name: ... ,  
  needs: [  
    ...  
  ],  
  outputDir: ... ,  
  outputExt: ... ,  
},
```

- スコープを区切ってpull requestを作成し、担当者にレビューと動作確認依頼
- 新しい仕組みと旧パッケージを同時にビルドする仕組みを作成
  - 移行が完了するまで両方を共存させる
- 移行完了後にパッケージを廃止

# ビルドされたファイルのコミットを保証する仕組み

- CIでビルドを実行して差分が出れば、CIを落としてpull requestにコメントで警告する





# diffのノイズを減らす

- .gitattributes に linguist-generated=true を指定する

The image displays three sequential screenshots of a GitHub diff interface, illustrating the effect of the `linguist-generated=true` attribute in `.gitattributes`. Each screenshot shows a file path, a 'Load diff' button, and a message about generated files not being rendered by default.

**Top Screenshot:** File path: `systems/action-table/packages/web/src/generated-shared-constants/common/products.json`. The diff view shows a message: "Some generated files are not rendered by default. Learn more about [how customized files appear on GitHub](#)."

**Middle Screenshot:** File path: `systems/admin/packages/navigation-rpc/src/generated-shared-constants/common/products.json`. The diff view shows a message: "Some generated files are not rendered by default. Learn more about [how customized files appear on GitHub](#)."

**Bottom Screenshot:** File path: `systems/admin/packages/navigation-v5/src/generated-shared-constants/common/products.json`. The diff view shows a message: "Some generated files are not rendered by default. Learn more about [how customized files appear on GitHub](#)."

# 外部リポジトリに変更を同期する仕組み

- 定数を変更するpull requestがマージされたら、外部リポジトリにも同期するpull requestを作成する



独自の仕組みなのでドキュメントは手厚めに

constantsをpublishして共有する運用をやめて、定数ファイルを：  
配布する仕組みに変更する #110257

➔ Merged yuhely merged 39 commits into develop from feature/shared-constants-setup on Sep 20

Conversation 15    Commits 39    Checks 12    Files changed 125

### 目的

- 定数を共有するために `github.com/medjs-constants / @plaiddev/frontend-constants` を `publish` する運用をやめて、定数ファイルのソースを `src` に置いて配布する仕組みに実装する
- 各システムで必要な定数の定義を明示する
- 共有のシステムで使っている定数が変更されたら検知できるようにする
- 定数ファイルのソースを `JavaScript` から `TypeScript` に書き換える
- refs
  - [github.com/medjs-constants](https://github.com/medjs-constants) の `config` を開いたら #104592
  - <https://plaiddev.slack.com/archives/C0K20K375AA/p17234514106896839>

経緯

- `node/express-server`をpublishして配布することによって、`node/express-server`が依存するライブラリも自動的に、`node/express`パッケージとして同様に配布される。publishが成功した場合は
- ①し、数秒を要するたびにパッケージをpublishするのを禁止する方法として自動的に15秒の冷却時間がある
- ② `node/express`の変更が必要ならば、各システムごとの環境構築に必要になるが、各システムで公開される公開ファイルが`publish`しなればならぬことになる
- ③ ①し、理想的には、各システムごとの環境構築に必要になるが、各システムで公開される公開ファイルが`publish`しなればならぬことになる
- ④ ②、数秒を要するたびにパッケージをpublishするのではなく、1回だけパッケージが完了したら全システムの変更を確定させる
- ⑤ `npm publish`を適用しないことを希望するために、`Signal`は`node/express`に`node/express`をインストールする仕組みを実装していた
- ⑥ `signals`の`shared.config.constants`と`node/express`の`node/express`を参照する
- `node/express`がpublishして1週間することによって自動的に15秒の冷却時間がある

関連する問題

- 定数の変更は各システムに速やかに反映させるべきなので、publish後にはシステム側の依存バージョンを上げる必要がある。この依存バージョンを上げるための仕組みがworkflowとして実装されていたが、実装当時の環境の変化によって現在は動作しなくなっている。修正するにはいくつかの問題の解決が必要な状況

- [shared/frontendのビルドとリリース周りの雑々な問題を解説する](#) #109474 [comment]
- 定数ファイルはJavaScriptで実装されているが、過去にタイプによる実装ミスが発生したことがあるので、TypeScriptに書き換えた
- どのシステムからどの定数が参照されているかが不明瞭のため、不意に変更すると事故が起こる可能性が高い

[illegible]

## 定数并

## 定数共有のための新しい仕組みと運用フロー

1. 現状の `static/src/constants` と同様のファイル群を、`shared-constants/src` に作成する。この時、オブジェクトのキーとして定数のスコープ名を明示する形で値を定義しておく。

[illegible][illegible]

shared-constants

システムを置いて利用されるお客様もシステムに接続して閲覧するための仕組みです。

ここに定義を定義した上で、定義をどのシステムから利用するかを設定することで、必要な箇所に定義ファイルがコピーされるようになります。

### 1. 定数の定義

定義は [https://en.cppreference.com/w/cpp/string/basic\\_string\\_view](https://en.cppreference.com/w/cpp/string/basic_string_view) に記載します。定義のスコープ名とその内容をオブジェクトとして記述します

shared-constants/lpc/common/ln

```
report admin_role from "/admin_role";
report chat_status from "/chat_status";
report contract_status from "/contract_status";
// ...

export default {
  "username/admin_role": admin_role,
  "username/chat_status": chat_status,
  "username/contract_status": contract_status,
  // ...
}
```

定数は用途に応じて適宜ファイルを分割して定義します。分割されたすべての定数をまとめた `index.ts` がエントリポイントになっています。

shared-constants/src/index.ts :

```
import common from './common';
import front from './front';
import locales from './locales';
import server from './server';
```

## 2. 定数の参照場所の登録

定義された変数をシステム側から参照するには、`sync-config.txt` にその設定を記述します。

shared-constants/sync-config.t

[illegible]

```
export default config;
```

- **name**: 参照箇所のコンテキストを示す名前
- **needs**: 必要な数のスコープ
- **separator**: 文字がコピーされる先のディレクトリ
- **outputdate**: 出力ファイルの形式 ( `ts / sjs / cjs / js / json` )。 `ts / sjs / cjs` として出力することで、[Kvno](#)などの静的解析ツールで使われる場合チャックが容易になります

### 3. 定数の配布

定数の値を入力すると、1990-1991年度、年度内情にもとづいて定数ファイルが出力されます。

```
path/to/char0-constants
```

これによって、すべての定数ファイルが `...generated...` に出力された上で、必要な定数ファイルだけが各システムにコピーされます。

システムにコピーされた変更ファイルは次のようにして読み込みます。

```
import {
  EVENT_NAME_MAP,
  INIT_DISPLAY_EVENTS,
} from '@ngrx/store-devtools-components/src/constants/event_name_map'
```

ビルドによって生成されたファイルは、すべてまとめて作業中のpull requestにコミットします。この際、生成されたファイルは、gitignoreなどで除外しないで行ってください。

また、内容が変更されないらば、それを管理するシステム上で数分が経過します。引システム担当者へのレビューが必要であれば、そのpull requestにレビューのコメントを返信を得ることでバージョンをうまく行うことができます。

また、変更の度合いがビルドをせずpushされた場合は、ビルド実行が完了して検閲されたファイルになります。ローカルでビルドしてpushし直してください。

#### 4. 外部リポジトリへの同期

※数ファイルは一部の外部リボソトリからも参照されています。そのため、数値ファイルが変更された際には、それらのリボソトリにも変更を反映する仕組みが働いています。

外部ツボゾトリ向けの定義ファイルは、ビルド時に `shared-configs/for-external-repositories` に配置されます。これらのファイルの変更が検知されると、karate.io-systems/pull-requestのマージ後、対象のツボゾトリにも変更を適用するpull requestが作成されます。