

日本版簡易 PNP(Personal Needs and Preferences) の設定と QR コードの生成・解析を行う JavaScript コンポーネント

version 1.0

有限会社サイパック

2025 年 2 月 12 日

1. はじめに

このソフトウェアは、日本版簡易 PNP(Personal Needs and Preferences) の設定と QR コードの生成・解析を行う Vue.js 3 の環境下で利用できる JavaScript コンポーネントです。

2. 機能

本コンポーネントには、以下の機能があります。

- 表示設定
- データ保存
- JSON データ登録
- QR 表示
- QR 読み込み
- QR アップロード
- 表示サンプル
- About
- Labo

各機能はそれぞれ独立したコンポーネントとして実装しています。それぞれの機能の詳細は以下の通りです。

2.1 表示設定

JPNP で規定されている設定項目を UI で変更する機能です。設定できる項目は以下の通りです。

- 背景色
- 文字色

- ルビ色
- フォント :教科書体/ゴシック体/明朝体
- フォントサイズ
- 文字間
- ルビフォント
- ルビフォントサイズ
- ルビ文字間
- 行間
- ワードラップ :あり/なし
- ルビ表示 :指定無し/総ルビ/パラルビ/ルビ無し
- 分かち書き :指定無し/空白/無し
- 表示方向 :指定無し/縦組/横組
- 表示方向による UI の変更 :あり/なし

指定項目の値を変更すると、表示設定画面は指定結果を元にした表示方法にダイナミックに更新されます。



次の画面は表示方向を縦組に指定して、フォントサイズを 20 ポイントに変更したものです。



次の3つのプリセットボタンが用意されており、プリセットボタンを押すと代表的な色使いの設定に即座に切り替えられます。

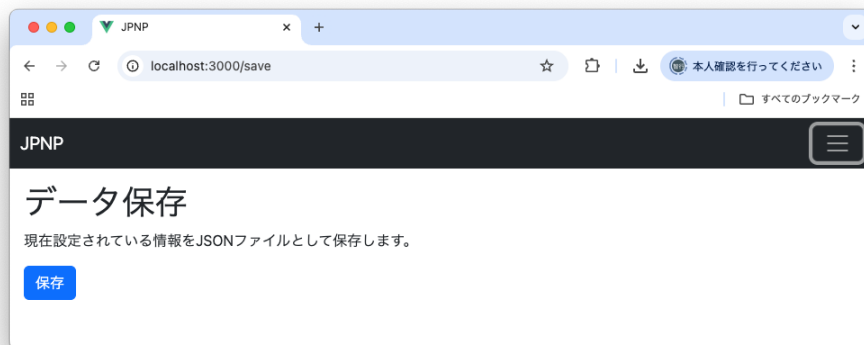
- プリセット 1
- プリセット 2
- プリセット 3

プリセット 1 は白地に黒の文字、プリセット 2 は黒地に白の文字、プリセット 3 は青地に黄色の文字の色使いになっています。本システムでのデフォルトの設定値はプリセット 1 のものなので、プリセット 1 のボタンは設定のリセット機能を兼ねています。

サンプル表示ボタンを押すと、設定値を適用した文書のサンプルが表示されます。

2.2 データ保存

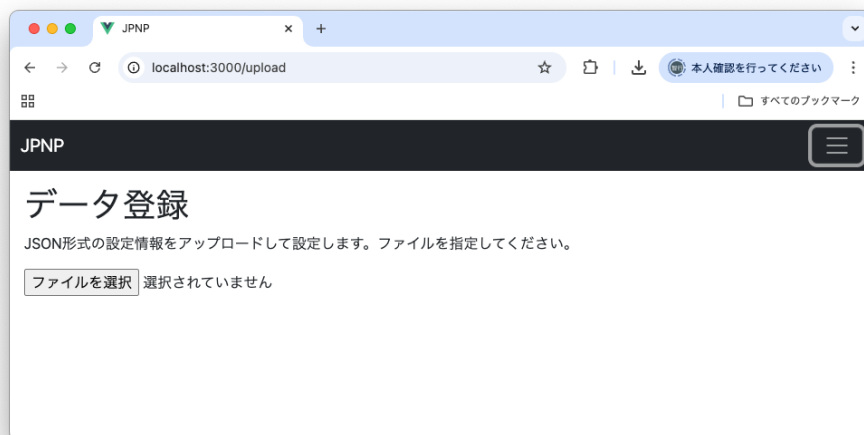
設定した JPNP の内容を json ファイルの形式で保存する機能を提供します。



「保存」ボタンを押すと、ローカルのデバイスに json 形式の JPNP 設定ファイル (jnpn.json) を保存できます。

2.3 JSON データ登録

データ保存でローカルに保存した json 形式の JPNP 設定ファイルをブラウザにアップロードして設定値とする機能です。



2.4 QR 表示

設定内容を QR コードで表示します。



この画面はブラウザの印刷機能を使って印刷できます。また、QR コードを右クリックすると表示されるポップアップメニューから QR コードの画像をファイルとして保存できます。

2.5 QR 読み込み



デバイスのカメラから JPNP の QR コードを読み取ります。カメラに前述の機能を使って印刷した QR コードをかざして、以前の設定値を復元できます。こちらの機能を利用するには、デバイスにカメラが必要です。

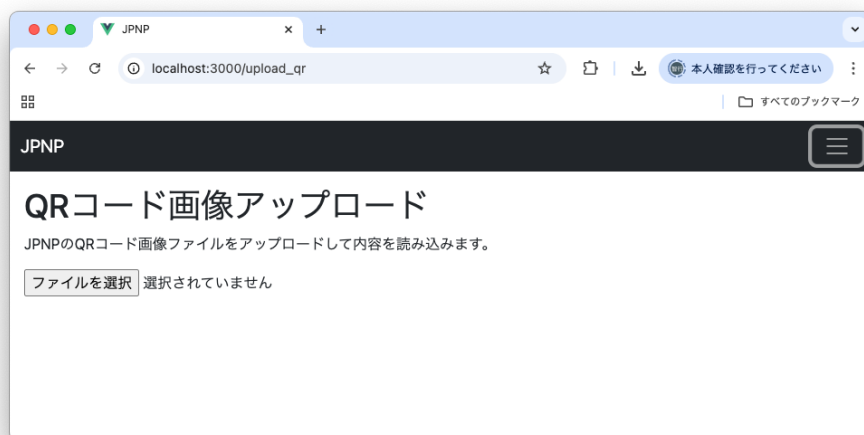
カメラへのアクセスについてはブラウザによるセキュリティ上の縛りが存在し、以下の 2 通りについてのみ許可されています。

- <https://> の URL で動作していて、正しい電子証明書が設定されている WEB サーバ
- ローカルで動作している WEB サーバ([http://localhost: ポート番号/](http://localhost:ポート番号/))

<http://> の WEB サーバで提供されているコンテンツからは、カメラへのアクセスが行えません。

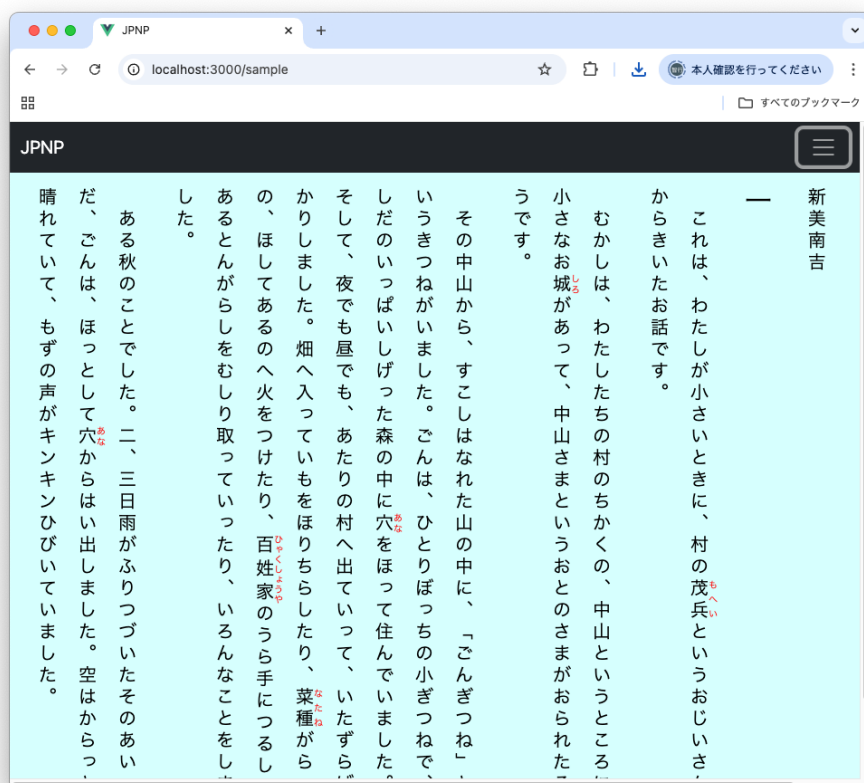
2.6 QR アップロード

QR コードの画像ファイルをアップロードして、以前の設定値を復元する機能です。



2.7 表示サンプル

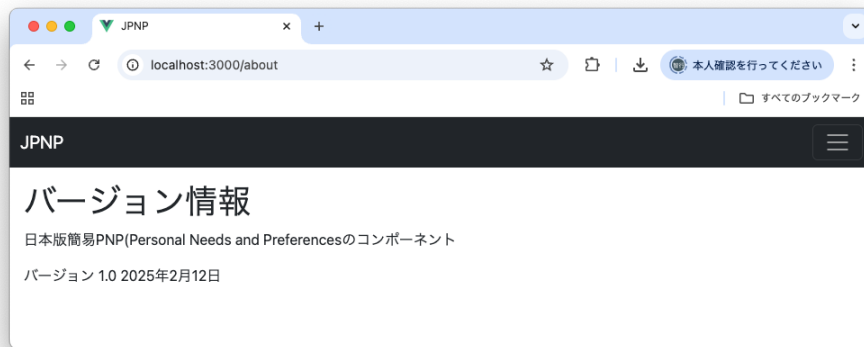
JPNP の設定値を適用した文書を表示するサンプルです。



表示設定画面のサンプル文書表示は、こちらのコンポーネントを内部に埋め込んで実現しています。

2.8 About

本コンポーネントのバージョン番号を表示します。



2.9 Labo

開発用の機能です。内部に読み込まれている JPNNP の json 情報がリアルタイムに表示されます。


```
$ rm -rf node_modules
$ npm install
```

上記コマンドは Makefile の clean と install のルールを用いても実行できます。

これで Vue.js を始めとする動作に必要な JavaScript のモジュールが node_modules のディレクトリ下にインストールされます。インストールが完了したら npm run dev のコマンドで開発用サーバーを起動できます。

```
$ npm run dev
```

```
> jpn@0.0.0 dev
> vite
```

The CJS build of Vite's Node API is deprecated. See <https://vite.dev/guide/troubleshooting.html#vite-cjs-node-api-deprecated> for more details.

```
VITE v6.1.0  ready in 124 ms
```

```
Local:   http://localhost:3000/
Network: use --host to expose
press h + enter to show help
```

上記コマンドは Makefile の run のルールを用いても実行できます。

ブラウザから画面に表示される URL にアクセスすると、本コンポーネントのページが表示されます。

3.4 ビルド

開発用サーバではビルドは不要ですが、本番のデプロイメントのためにはビルドが必要です。Vue.js は大量のコンポーネントで構成されていますが、これを 1 つの JavaScript ファイルにコンパイルするのがビルドのプロセスです。

以下のコマンドでビルドを行ってください。

```
$ rm -rf dist
$ npm run build
```

上記コマンドは Makefile の build のルールを用いても実行できます。

ビルドが完了すると、dist ディレクトリ下に次のようなコンパイル結果のファイルが生成されます。

```
$ ls -lR dist
total 24
drwxr-xr-x  4 kudou  staff   128 Feb 12 22:32 assets
```

```
-rw-r--r-- 1 kudou staff 4286 Feb 12 22:32 favicon.ico
-rw-r--r-- 1 kudou staff 833 Feb 12 22:32 index.html
```

dist/assets:

total 928

```
-rw-r--r-- 1 kudou staff 46 Feb 12 22:32 index-Bwborzv7.css
-rw-r--r-- 1 kudou staff 470945 Feb 12 22:32 index-Cf1Q3Tj3.js
```

このツリーをデプロイメント用のサーバにセットすれば、本機能が利用できます。

3.5 Docker を使ったローカルのサーバ

Docker を使って nginx のコンテナを動作させ WEB サーバを起動する設定ファイルを用意しました。以下のコマンドで Docker のコンテナをビルドし、起動できます。

```
$ make docker-build
```

```
$ make docker-up
```

これで 8400 番のポートを通じて dist ディレクトリ(上記のビルド結果が格納されたツリー)の内容を公開する WEB サーバが立ち上がります。

Docker コンテナのサーバの停止は、次のようにします。

```
$ make docker-down
```

3.6 ソースファイル

今回開発した Vue.js のコンポーネントの主たるソースファイルは次の通りです。

- index.html : 初回起動用ページ
- src/App.vue : 全体を包括するトップレベルのコンポーネント
- src/index.css : css
- src/main.js : Vue.js の起動用スクリプト
- src/components/SettingJPNPComponent.vue : 表示設定用コンポーネント
- src/components/SampleComponent.vue : サンプル表示用コンポーネント
- src/components/Mbr.vue : < mbr/>タグの表示用コンポーネント
- src/components/SaveJPNPComponent.vue : JSON 形式の JPNP ファイル保存機能コンポーネント
- src/components/UploadJPNPComponent.vue : JSON 形式の JPNP ファイルアップロード機能コンポーネント
- src/components/ShowJPNPQRComponent.vue : QR コード表示用コンポーネント
- src/components/ReadJPNPQRComponent.vue : JPNP QR コード読み込み用コンポーネント
- src/components/UploadJPNPQRComponent.vue : QR コード画像アップロード機能コンポーネント

- src/components/AboutComponent.vue : バージョン表示用コンポーネント
- src/components/LaboComponent.vue : 開発用コンポーネント
- src/components/Navbar.vue : ナビゲーションバーのコンポーネント
- src/assets/logo.png : ロゴ
- src/store/index.js : モデル部分のコンポーネント (本システムの核)
- src/router/index.js : ルーター (ナビゲーションメニュー用)
- package.json : 依存パッケージ情報
- vite.config.js : vite の設定情報
- Makefile : ビルドルール
- docker-compose.yml : Docker Compose 設定ファイル
- nginx/Dockerfile : nginx コンテナビルド用設定ファイル
- nginx/default.conf : nginx サーバ設定ファイル
- .vscode/launch.json : VScode+Chrome でのデバッカー動作に必要な設定ファイル
- doc/readme.md : 本ドキュメントソース (マークダウン形式)