

第2章 マニフェストファイルを作る

マニフェストファイルはアプリの仕様が書かれたJSONファイルです。多くの場合 `manifest.json` という名前で作成します。これをHTMLの中で読み込みます。

元：

```
<!-- <link rel="manifest" href="./manifest.json"> -->
```

修正後：

```
<link rel="manifest" href="./manifest.json">
```

今回はTodoアプリを作っている最中にインストールされてしまうと問題があるのでコメントアウトしています。アプリ化を体験する際にはコメントアウトを外してください。

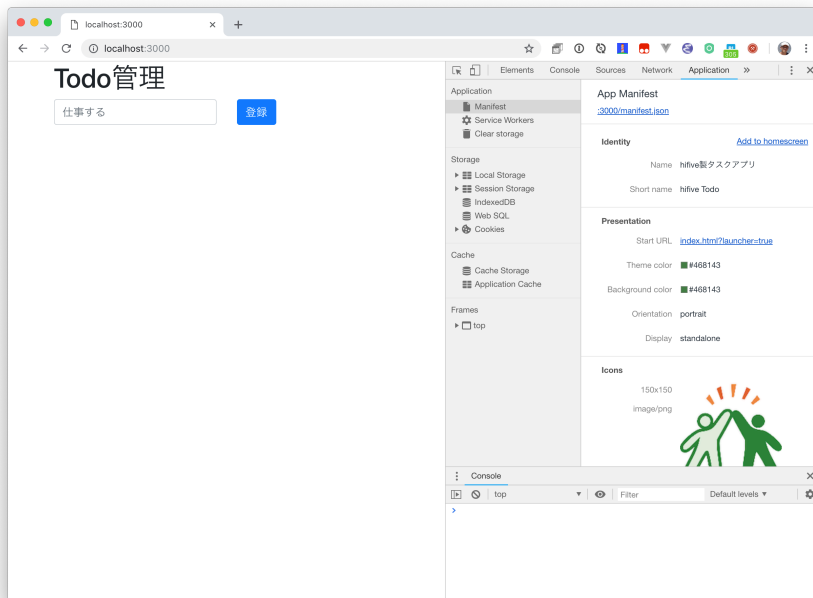
マニフェストファイルの内容

マニフェストファイルは以下のような内容になっています。

| キー | 内容 |
|-------------------------------|----------------------------|
| <code>short_name</code> | 短いアプリ名 |
| <code>name</code> | アプリ名 |
| <code>icons</code> | アイコン。サイズに応じて複数指定可能 |
| <code>display</code> | フルスクリーン、スタンドアローン、Webブラウザなど |
| <code>background_color</code> | アプリが立ち上がる際の背景色 |
| <code>theme_color</code> | テーマカラー。ヘッダーバーの色の適用 |
| <code>orientation</code> | 回転方向 |
| <code>start_url</code> | PWAを表示する際のURL |

内容を確認する

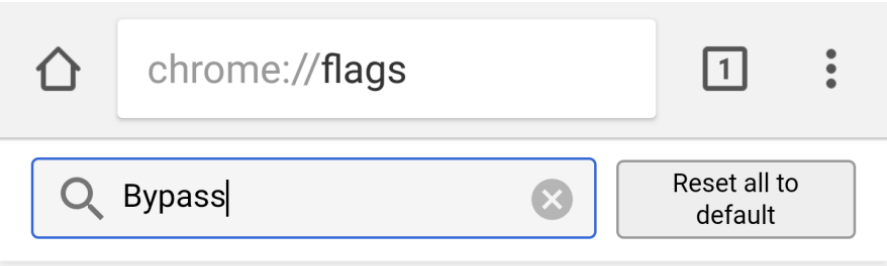
`manifest.json` がきちんと書かれているかどうかは Google Chrome で確認するのが一番簡単です。開発者ツールを開いて、Application タブに切り替えます。以下のように `manifest.json` ファイルの内容が表示されます。



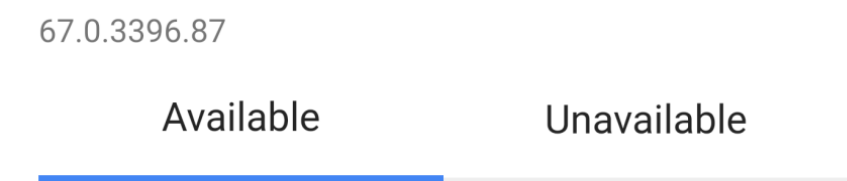
この内容を編集して、Webブラウザで再読込すると表示に反映されます。編集して確認してみましょう。

Androidで設定を変更する

PWAとしてインストールするか確認するバナー（A2H = Add to Home Screen）は5分以上の時間をおいて、2回目以降のアクセスで表示されます。しかし開発中ではこの状態では不便なので、Google Chromeの設定変更をお勧めします。Androidで `chrome://flags` を開きます。そしてBypass user engagement checksと検索して有効にします。



Experiments

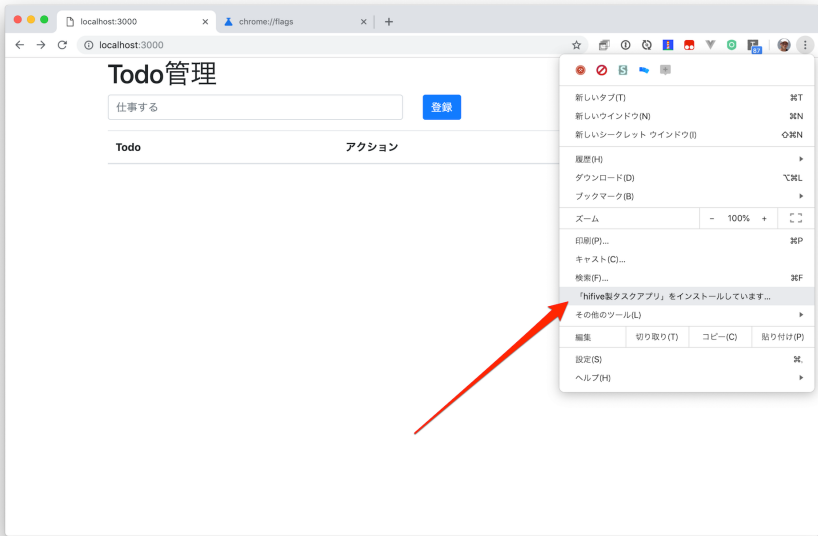


Bypass user engagement checks

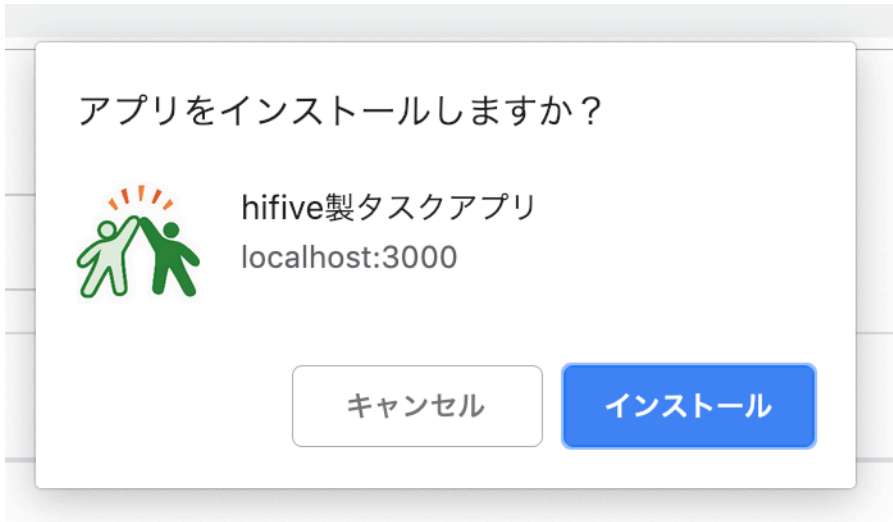
Bypasses user engagement checks for displaying app banners, such as requiring that users have visited the

これで一度目のアクセスでインストールバナーが表示されるようになります。

再起動後、メニュー（右上の縦型の三点リーダー）をクリックすると「**hifive製タスクアプリ**」をインストール... と表示が追加されています。これを選びます。



選ぶとインストールを行うダイアログが出ます。インストールを押せば、デスクトップアプリとしてインストールされます。



Windowsの場合はデスクトップに、macOSの場合は `~/Applications/Chrome` アプリの中にインストールされます。ここから立ち上げると、アドレスバーだけが表示され、シンプルなUIでウィンドウが開きます。



Service Workerについて

アプリ化はマニフェストファイルだけでは利用できません。次の第3章 Service Workerのインストールと有効化と第4章 Service Workerを使った表示高速化、オフライン対応についてを行うとアプリとしてインストールできるようになります。では第3章 Service Workerのインストールと有効化を行いましょう。