

# Firefox OS アプリ開発のご紹介

2015/02/13

一般社団法人 Mozilla Japan

技術部 ソフトウェアエンジニア

吉永 満太郎

mozilla

Do you know Firefox or Firefox OS ?



# Have you Web Content Service ?



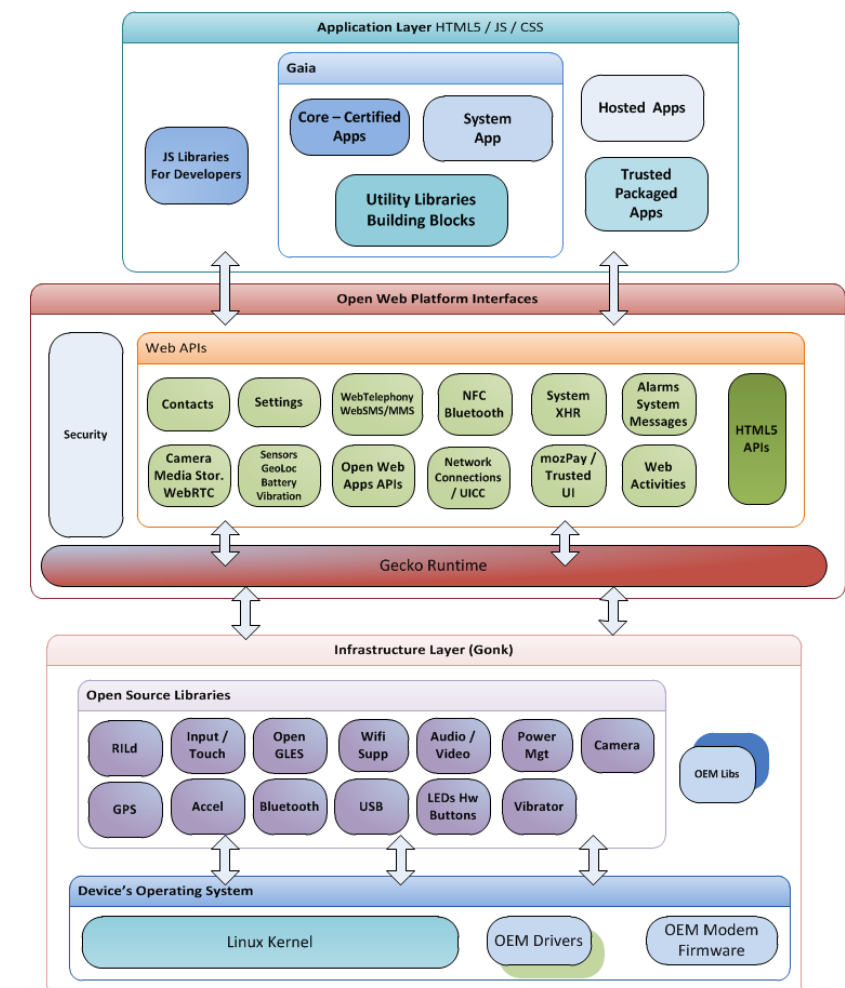
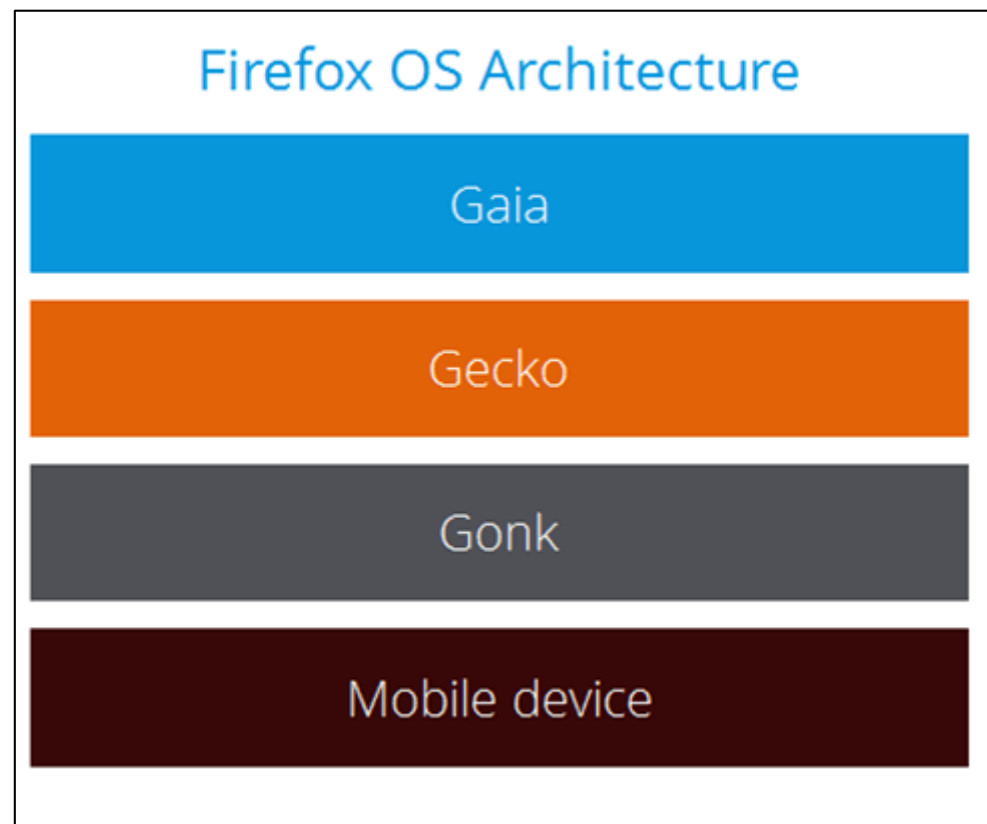
# Agenda

- Firefox OS アプリの概要
- 開発に必要なもの
- 開発ドキュメント
- Firefox OS アプリ開発
  - アプリの種類
    - ホスト型
    - パッケージ型
  - 基本構成
  - Manifest
  - パーミッション
  - 利用可能な API
- クロスプラットフォーム
- 開発Tips
  - オフライン動作
  - クロスバージョン
  - CSP
  - 実機 / エミュレータの違い

# Firefox OS プラットフォームの概要

Gonk と呼ばれるカーネル層に Firefox と同じエンジン (Gecko) を載せ、そのうえで Gaia と呼ばれる UI 層を載せたもの

Firefox OS = Gonk + Gecko + Apps (Gaia)



# Firefox OS プラットフォームの概要

Gaia 上では、すべてのアプリが Web の技術 (HTML / CSS / JavaScript) で動作しています。



一部、アクセス権が必要な API を利用していること以外は Web アプリと一緒にです

# 開発に必要なもの

- 必要なスキル

HTML / CSS / JavaScript のみ

Mozilla の開発ドキュメントでも紹介しています。

<https://developer.mozilla.org/ja/Apps/Quickstart>

<https://developer.mozilla.org/ja/docs/Web/Tutorials>

# 開発に必要なもの

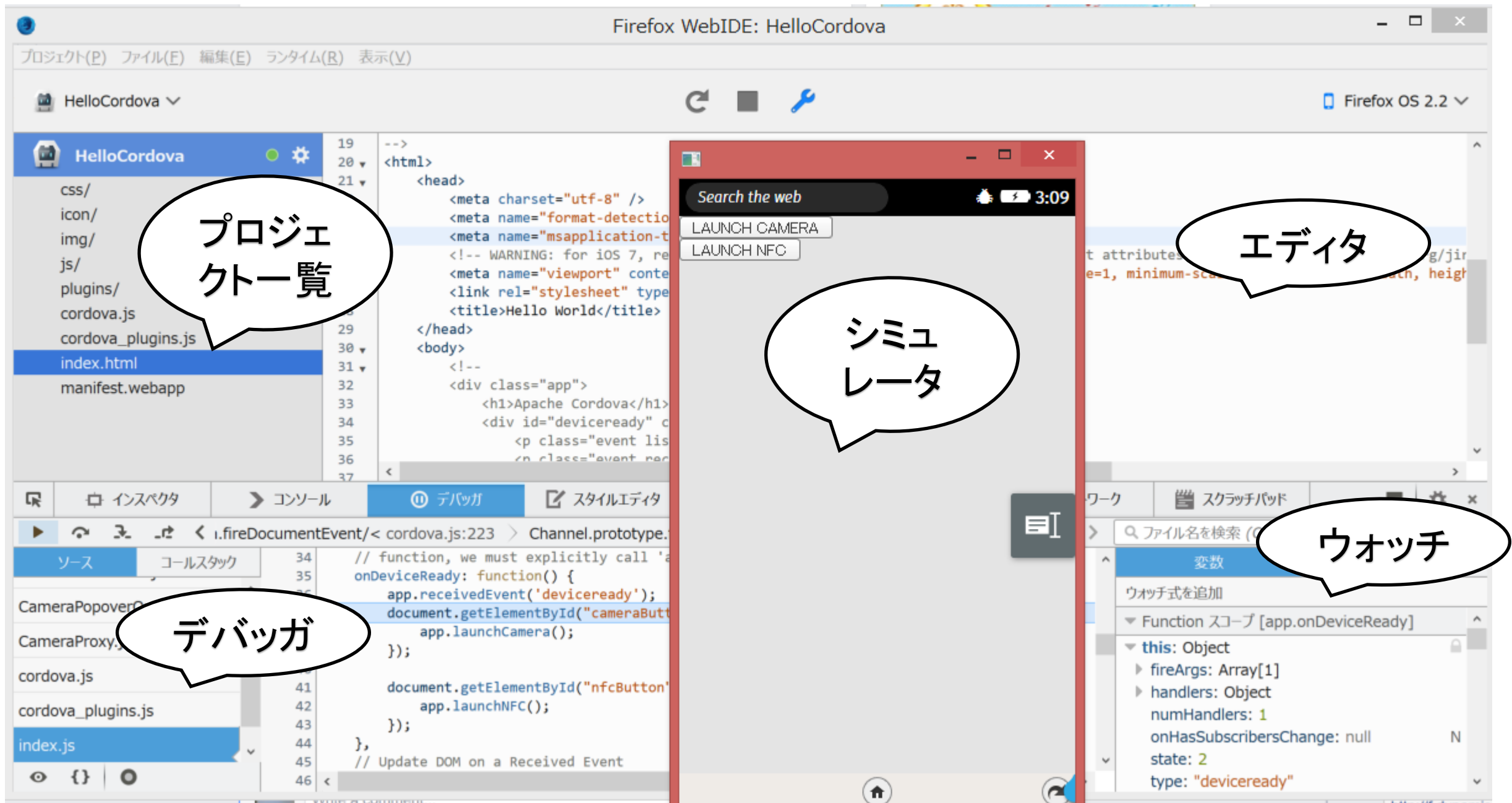


Firefox のみでアプリケーション開発は可能  
(ほかにツールなどがあれば便利になる)



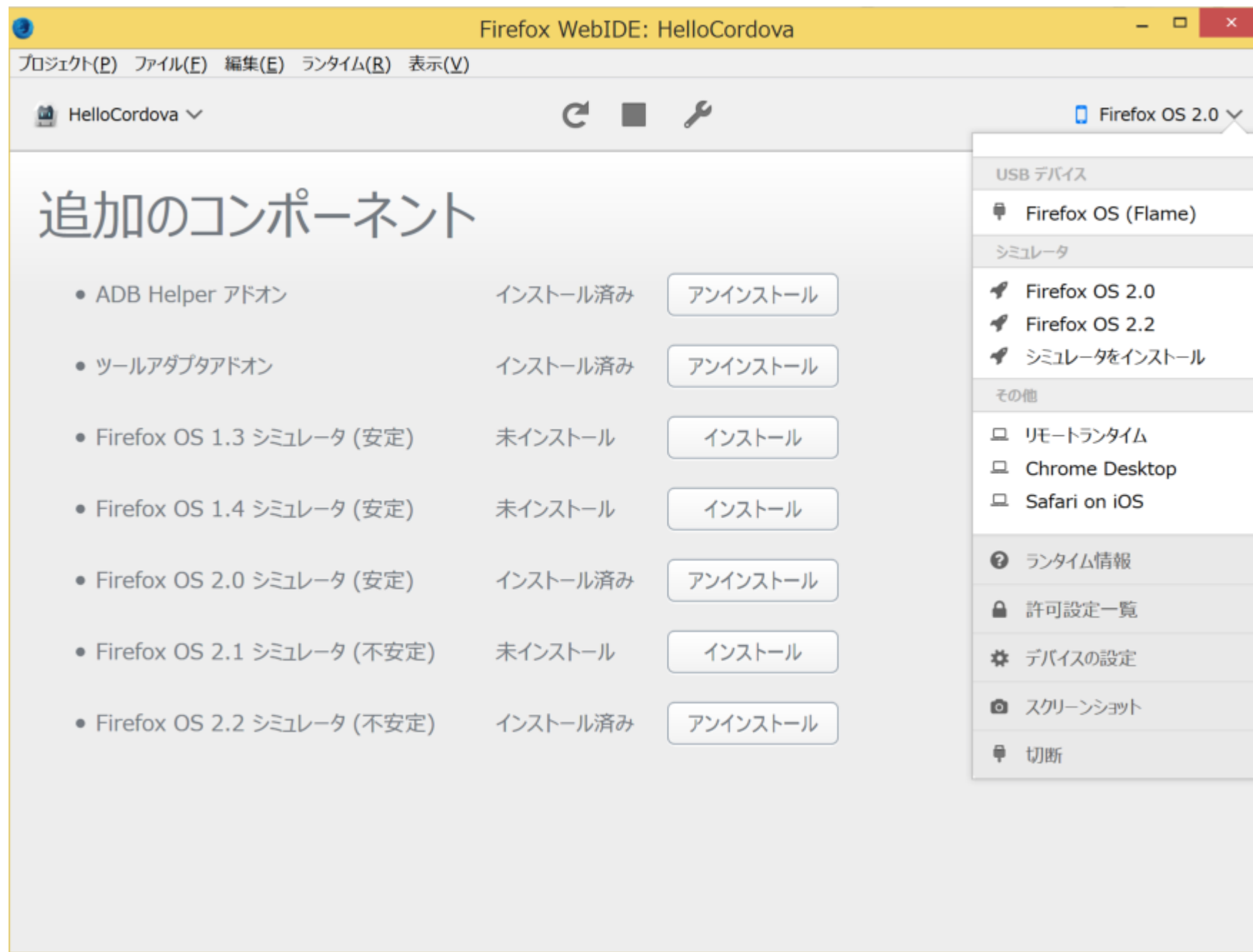
# 開発に必要なもの

デバッグ環境は、Firefox 付属の Web IDE を利用する



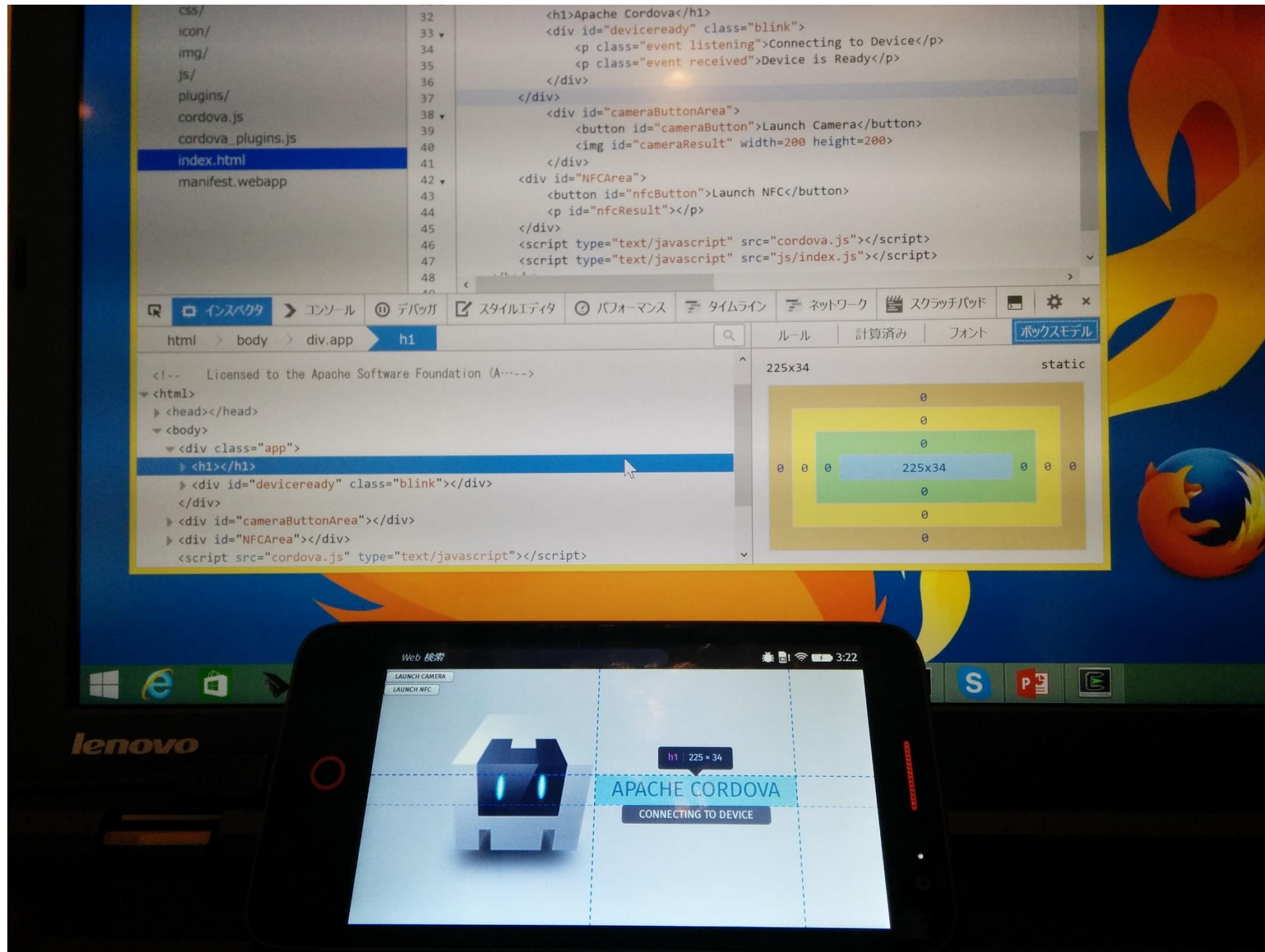
# 開発に必要なもの

シミュレータのインストール起動も WebIDE から実行



# 開発に必要なもの

実機接続後も WebIDE から接続しデバッグ



# 開発に必要なもの

必須ではないが、環境に応じて IDE ・ JavaScript ライブラリ ・ ツールチェーンなどが制限なく利用できます。

※後述する CSP で一部制限が発生する可能性がある





# Firefox OS 開発ドキュメント

Mozilla Developer Network (MDN) に Firefox OS および Firefox OS アプリに関するドキュメントがあります。

<https://developer.Mozilla.org>



## Hacks Blog [hacks.mozilla.org](https://hacks.mozilla.org/) でもっと読む →

### Embedding an HTTP Web Server in Firefox OS

Nearing the end of last year, Mozilla employees were gathered together for a week of collaboration and planning. During that week, a group was formed to envision what the future of Firefox OS might be surrounding a more P2P-focused Web. In particular, we've been looking at harnessing technologies to collectively ...

February 9, 2015 by Justin D'Arcangelo

### Open Web Apps feedback: Consolidating our channels

In August 2014 we announced the opening of a new feedback channel for web apps on UserVoice. It has led to some good discussions and here are a few highlights: This post showed the importance of the nascent FileSystem API. A suggestion about background services reinforced the need for something ...

February 6, 2015 by Andrew Overholt

### Introducing node-firefox

NOTE: we presented this project last Sunday at FOSDEM, but not everyone could make it to Brussels, so here's a post explaining what node-firefox is and how can it help you superturbocharge your Firefox OS app development! At Mozilla we're always looking for ways in which we can make developers' ...

February 5, 2015 by Soledad Penades

### BroadcastChannel API in Firefox 38

Recently the BroadcastChannel API landed in Firefox 38. This API can be used for simple messaging between browser contexts that have the same user agent and origin. This API is exposed to both Windows and Workers and allows communication between iframes, browser tabs, and worker threads. The intent of the ...

February 2, 2015 by Jason Weatherly

## 参加する

Join **6355** contributors

in **42** languages and locales

around the world.

[MDN の改善に貢献する](#)

# Firefox OS 開発ドキュメント

今回対象としている Firefox OS アプリについては、MDN 内「App センター」に情報がありません。

<https://developer.mozilla.org/ja/Apps>

The screenshot shows the MDN App Center page. The header includes the MDN logo and navigation links for 'ゾーン', 'WEB プラットフォーム', 'ツール', 'MDN デモ', and 'つながり'. The main heading is 'アプリセンター'. A sidebar on the left lists categories: 'クイックスタート', '設計', '開発', '公開', 'ツールとフレームワーク', 'リファレンスアプリ', and 'リファレンス'. The main content area features three columns: '設計' (Design), '開発' (Development), and '公開' (Release). Each column contains a brief description and a list of links. At the bottom, there are three more sections: 'Firefox OS', 'デバッグ', and 'コンポーネント'.

MDN MOZILLA DEVELOPER NETWORK

ゾーン WEB プラットフォーム ツール MDN デモ つながり

MDN > 開発者向けのWeb技術 > アプリセンター

言語 編集

## アプリセンター

クイックスタート

既存の Web 標準とオープンなテクノロジーを活用して、Firefox OS を含む様々な環境で動作し、リッチな体験ができる Open Web Apps の作成方法を学びましょう。

### 設計

クロスプラットフォームな優れたユーザ体験を提供するインストール可能な Open Web Apps の設計方法を学びましょう。

- アプリの構想
- UI レイアウトの基本
- Firefox OS スタイルガイド

### 開発

実際の開発にあたっての問題解決に必要なアドバイスとチュートリアルをまとめました。

- クイックスタート
- インストール可能な Firefox OS アプリ
- アプリ開発 FAQ

### 公開

ユーザと開発者を第一に考えたオープンなマーケットプレイスでアプリを配布しましょう。詳しくは [Marketplace ゾーン](#) を参照してください。

- アプリ公開の選択肢
- Marketplace への登録
- 決済

### Firefox OS

Mozilla が開発しているオープンな Web ベースのモ

### デバッグ

あらゆるデバッグの必要性に応じて 開発者の作業に

### コンポーネント

アプリ開発の高速化に役立つ様々なツールと既成のフ



# Firefox OS 開発ドキュメント

MDN は全世界から参加しているボランティアが、ドキュメントのライティング・翻訳をしているプロジェクトです。皆様のご参加をお待ちしております。

<https://developer.mozilla.org/en-US/docs/MDN/Contribute>

The screenshot shows the MDN website's 'Contributing to MDN' page. The header features the MDN logo, a user profile 'mantaroh' with a 'Sign out' link, and a 'mozilla' dropdown menu. Navigation links include 'ZONES', 'WEB PLATFORM', 'TOOLS', 'DEMOS', and 'CONNECT'. A search bar is also present. The main heading is 'Contributing to MDN'. A sidebar on the left lists various topics under 'CONTRIBUTING TO MDN', with 'Contributing to MDN' currently selected. The main content area includes a welcome message, a 'Visit the MDN inbox' button, and sections for 'Contributor guides' and 'How to...'. The 'Contributor guides' section lists links for 'Getting started', 'Content and style guide', 'Editor guide', 'Reviewing articles', 'Terminology and conventions', and 'Working with the MDN community'. The 'How to...' section lists links for 'How to document a CSS property', 'How to document an HTML element', 'How to properly tag pages', and 'How to interpret specifications'.

MDN MOZILLA DEVELOPER NETWORK

mantaroh | Sign out mozilla

ZONES WEB PLATFORM TOOLS DEMOS CONNECT

The MDN project

MDN > The MDN project > Contributing to MDN

LANGUAGES EDIT

## Contributing to MDN

▲ HIDE SIDEBAR

ABOUT MDN

GETTING STARTED ON MDN

CONTRIBUTING TO MDN

Contributing to MDN

Creating and editing pages

Cross-team collaboration tactics for documentation

Documentation processes

Does this belong on MDN?

FAQ

How-to guides

Localization driver role

Localizing MDN

MDN administrator role

MDN content and style guides

MDN editor guide

MDN mentors

Mentor role

Page types

Welcome! By visiting this page, you've taken the first step toward becoming a contributor to MDN. The guides listed here cover all aspects of contributing to MDN, including style guides, guides to using our editor and tools, and more. Please make sure you have read (and are compliant with) the [Mozilla Terms](#) before editing or creating any pages.

Visit the MDN inbox

Add a new article

### Contributor guides

**Getting started**  
A quick start guide to getting set up and making your first contribution.

**Content and style guide**  
The MDN content and style guide provides details about writing style, page layout, and content styles, so that the content you write fits in with the rest of MDN's content.

**Editor guide**  
A complete guide to using MDN's editor.

**Reviewing articles**  
A guide to performing technical and editorial reviews of articles' content, to help us ensure that all content on MDN is as useful and readable as can be!

**Terminology and conventions**  
Our terminology and conventions guide provides information you can use to ensure that you use the correct terminology to describe things.

**Working with the MDN community**

### How to...

Our [how-to guides](#) provide step-by-step instructions to help you accomplish specific tasks when contributing to MDN.

**How to document a CSS property**  
A guide to writing documentation of CSS properties. All CSS property documents should match the style and layout described in this article.

**How to document an HTML element**  
This guide to documenting HTML elements will ensure that the documents you write match others on MDN.

**How to properly tag pages**  
This guide to tagging pages provides information about our standards for tagging, including lists of tags that have standard meanings on MDN. Following this guide will ensure that your content is properly categorized, more easily searchable, and that our search filtering mechanism works properly with your articles.

**How to interpret specifications**  
This guide will help you to properly interpret standard Web specifications; being able to read these can be an art form, and knowing how to do it will help you produce better documentation.

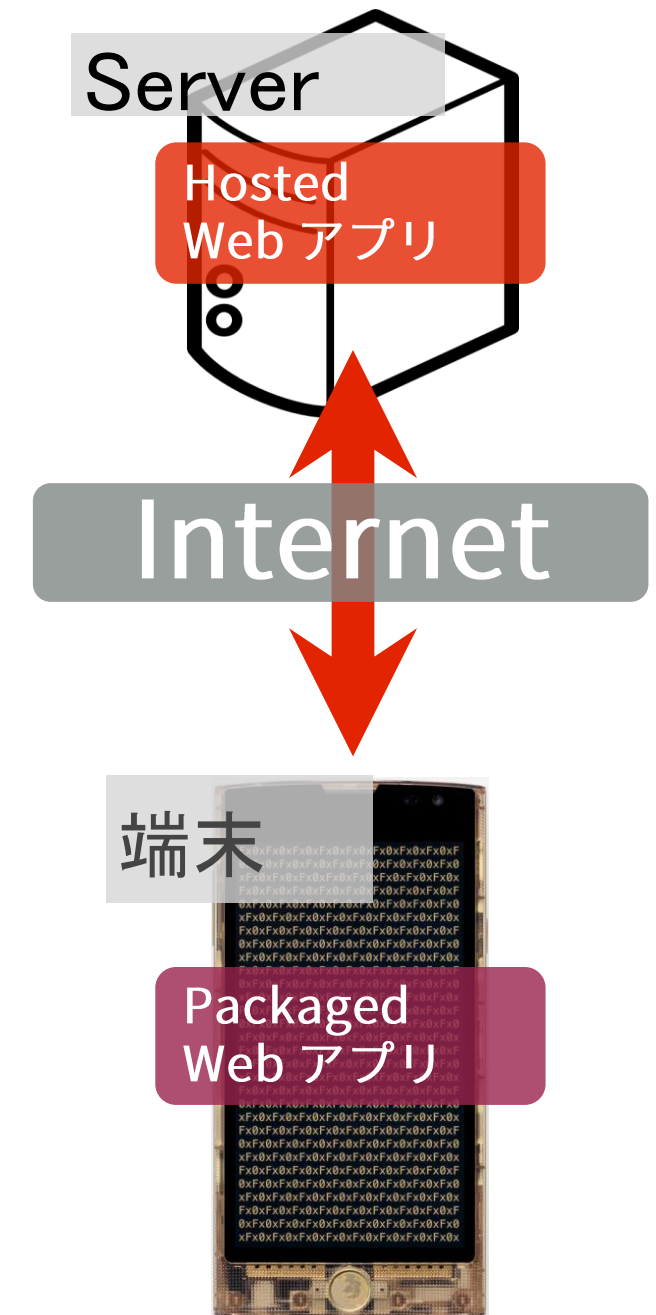
# Firefox OS アプリの種類

## ▪ Hosted (Web 読み込み型)

- 従来の OS ではブラウザ上で動作
- 動作や権限は従来の Web と同じ
- オフライン対応アプリも開発可能

## ▪ Packaged (ダウンロード型)

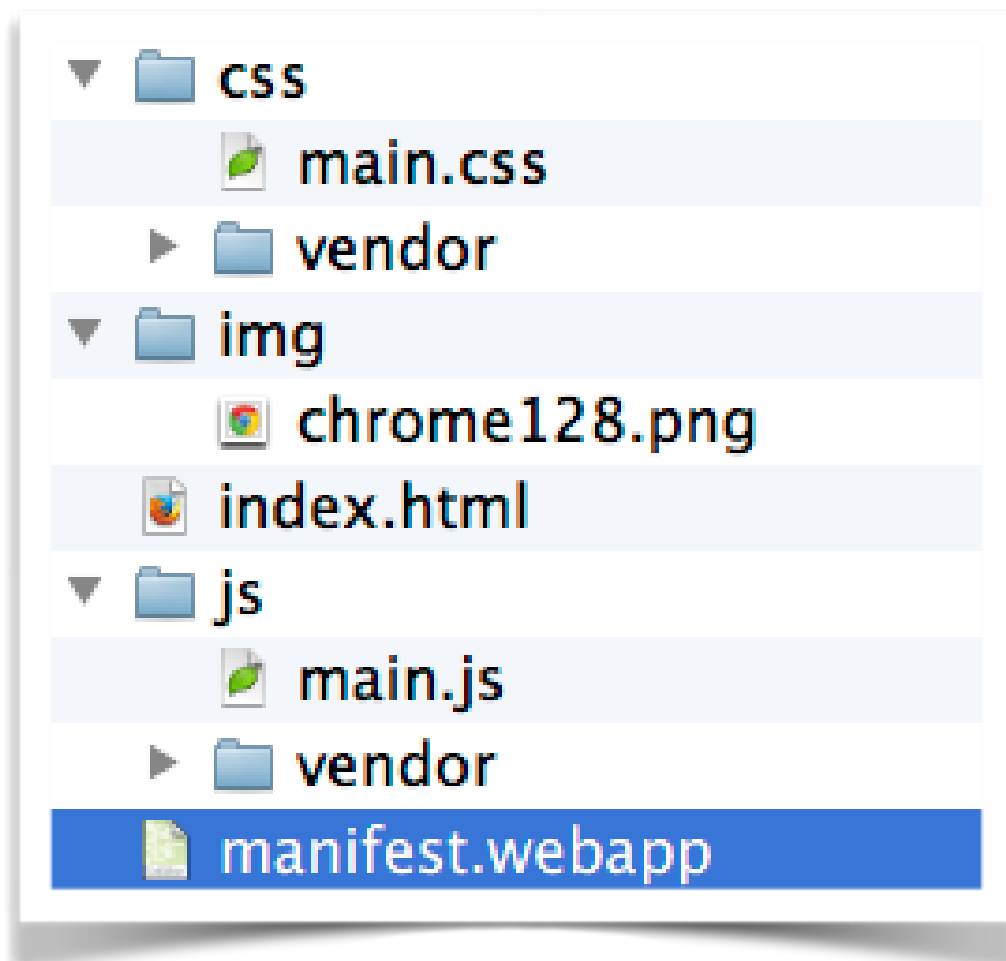
- 従来のスマホアプリに相当する
- マーケット審査を経て追加権限取得
- サイト全体を ZIP して配布する形式



詳細: [https://developer.mozilla.org/ja/docs/Web/Apps/Packaged\\_apps](https://developer.mozilla.org/ja/docs/Web/Apps/Packaged_apps)



# Firefox OS アプリの基本構成



- サイト + manifest.webapp
- manifest.webapp 作成
- アプリ情報書くだけで終了

詳細: [https://developer.mozilla.org/ja/docs/Web/Apps/Packaged\\_apps](https://developer.mozilla.org/ja/docs/Web/Apps/Packaged_apps)

# Manifest (アプリマニフェスト)

- アプリに関する情報(名称、作者、アイコン、説明など)を記載する JSON ファイル
- 最小構成は以下の通り

```
{  
  "name": "マイアプリ",  
  "description": "アプリの簡素な説明",  
  "launch_path": "/index.html",  
  "icons": {  
    "512": "/icons/icon-512.png",  
    "128": "/icons/icon-128.png"  
  },  
  "developer": {  
    "name": "あなたの名前もしくは組織名",  
    "url": "http://your-homepage-here.org"  
  }  
}
```

詳細: <https://developer.mozilla.org/ja/Apps/Manifest>

# Manifest(アプリマニフェスト)

- 戻る / 進むボタンが欲しい場合や、画面の回転の指定などは Manifest でオプションを指定

```
{  
  "name": "マイアプリ",  
  << 中略 >>  
  // 戻る進むボタンを表示する  
  "chrome": { "navigation": true },  
  
  // 常に ランドスケープ(横回転)で表示する  
  "orientation": [ "landscape" ]  
}
```



詳細: <https://developer.mozilla.org/ja/Apps/Manifest>

# アプリの種類 – ホスト型 –

## コミュニケーション API

### Network Information API

接続速度など、現在のネットワーク接続に関する基本的な情報を提供します。

### Bluetooth 非標準

WebBluetooth API は、デバイスのBluetooth ハードウェアに対する低水準のアクセス手段を提供します。

### Mobile Connection API 非標準

信号強度やオペレーター情報など、デバイスのセルラー接続に関する情報を公開します。

### Network Stats API 非標準

データの使用状況をモニタリングして、そのデータの特権アプリケーションに公開します。

### TCP Socket API 非標準

低水準のソケットおよび SSL をサポートします。

### Telephony 非標準

アプリが電話を発着信したり、内蔵の電話機能のユーザインターフェイスを使用したりすることが可能です。

### WebSMS 非標準

アプリに対して、デバイスに保存しているメッセージのアクセスや管理だけでなく、SMS テキストメッセージの送受信を可能にします。

### WiFi Information API 非標準

信号強度、現在接続しているネットワークの名称、利用可能な WiFi ネットワークなどの情報を提供する特権 API です。

## ハードウェアアクセス API

### Ambient Light Sensor API

環境光センサーへのアクセス手段を提供します。これにより、アプリがデバイス付近の環境光のレベルを検出できます。

### Battery Status API

バッテリーの残量や、デバイスが電源に接続されて充電中であるかといった情報を提供します。

### Geolocation API

デバイスの物理的な位置に関する情報を提供します。

### Pointer Lock API

アプリがマウスへのアクセスをロックして、絶対座標ではなく移動量の差分にアクセスできます。これはゲームに対して重要です。

### Proximity API

ユーザの顔など、近くにある物体とデバイスとの近接度を検出できます。

### Device Orientation API

デバイスの向きが変化したときに通知します。

### Screen Orientation API

スクリーンの向きが変化したときに通知します。この API は、アプリにとってどの向きが望ましいかを示すことにも使用できます。

### Vibration API

ゲームでの触覚フィードバックなどのために、アプリがデバイスのバイブレーションハードウェアを制御できます。これは通知のためのバイブレーションといった用途向けではありません。通知については **Alarm API** をご覧ください。

### WebFM API 非標準

使用可能であれば、デバイスの FM ラジオ機能をサポートします。

### Camera API 非標準

デバイス内蔵のカメラを使用して、アプリが写真や動画を撮影できます。

### Power Management API 非標準

アプリがスクリーン、CPU、デバイスの電源などをオン・オフできるようにします。また、リソースロックイベントのリスニングや調査もサポートします。

[すべて見る...](#)

## データ管理 API

### FileHandle API 非標準

ロック機能とともに、書き込み可能なファイルをサポートします。

### IndexedDB

パフォーマンスが高い検索機能をサポートする、クライアントサイドの構造化データ用ストレージです。

### Contacts API 非標準

ユーザの連絡先データベースへのアクセス機能を提供します。連絡先情報の追加、読み取り、編集をサポートします。

### Device Storage API 非標準

現代のデスクトッププラットフォームにおける "ピクチャ" フォルダやモバイルデバイスにおけるフォトストレージなど、デバイスの中心部にアプリがファイルを作成、読み取り、変更することが可能です。

### Settings API 非標準

デバイスへ永続的に保存されるシステム全体の設定オプションを、アプリが調査したり変更したりすることが可能です。

## その他の API

### Alarm API

アプリが通知を予定することが可能です。特定の時刻にアプリを自動起動する機能もサポートします。

### Simple Push API

プラットフォームが特定のアプリケーションに通知メッセージを送信できます。

### Web Notifications

アプリケーションが、システムレベルで表示される通知を送信できます。

### Apps API 非標準

Open WebApps API は、Web アプリのインストールや管理をサポートします。また、アプリが課金情報を調べることができるにします。

### Web Activities 非標準

あるアプリが別のアプリにアクティビティを渡すことができます。例えばあるアプリが別のアプリに対して、写真を選択 (または作成) して返すように依頼するでしょう。一般的に、ユーザはどのアクティビティにどのアプリを使用するかを設定できます。

### WebPayment API 非標準

Web コンテンツが、仮想的な物品に対する支払いや払い戻しの処理を開始できます。

### Browser API 非標準

Web 技術を完全に使用する Web ブラウザの構築をサポートします (要するに、ブラウザの中にあるブラウザです)。

### Idle API 非標準

ユーザが能動的にデバイスを使用していないときに、アプリが通知を受け取れます。

### Permissions API 非標準

集中化された場所で、アプリの許可設定を管理します。設定アプリで使用されます。

### Time/Clock API 非標準

現在時刻の設定をサポートします。タイムゾーンは **Settings API** で設定します。

## WebAPI コミュニティ

これらの API について支援が必要になったとしても、それらの使用方法について他の開発者と語る手段がいくつかあります。

- WebAPI フォーラムで相談する:
  - [メーリングリストとして](#)
  - [Twitter](#)
  - [Stack Overflow](#)
  - [Google Group](#) として
  - [フィードとして](#)
- WebAPI IRC チャンネルを訪問する: [#webapi](#)

[📄 ネットワークを忘れないでください...](#)

■ 既存の Web アプリをそのまま利用できる

■ アプリとしてアイコンを端末にインストールすることが可能

■ 利用できる API が限定

■ オフラインも考慮したアプリを作成可能

詳細: <https://developer.mozilla.org/ja/docs/WebAPI>



# アプリの種類 – ホスト型 –

- Firefox OS にこだわらず、Firefox が動作しているプラットフォームで実行可能
- アプリとしてインストールするための API が用意されている
- Firefox OS ホーム画面にアイコンとして登録されるが、アプリはサーバー上に配置されている

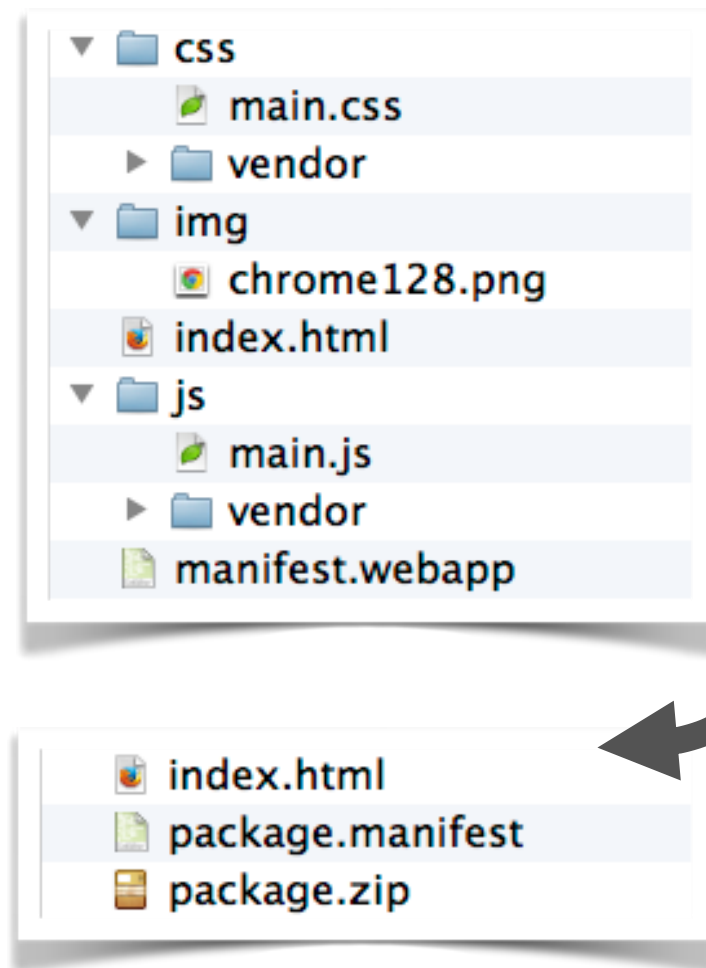
```
var installResult = navigator.mozApps.install(manifest_url);
installResult.onsuccess = function(data) {
    //成功処理
}
installResult.onerror = function(data) {
    //失敗処理
}
```



詳細: <https://developer.mozilla.org/ja/Apps/Quickstart>

# アプリの種類 – パッケージ型–

ホスト型アプリを zip 圧縮し、manifest を付け加えたもの  
端末に全リソースがインストールされる  
アプリの種類に応じた API が利用可能(特権付 or 内部[認定])



- ZIP
- 1. manifest.webapp 作成
- 2. サイト全体を ZIP する
- 3. package.manifest を作成

詳細: [https://developer.mozilla.org/ja/docs/Web/Apps/Packaged\\_apps](https://developer.mozilla.org/ja/docs/Web/Apps/Packaged_apps)

# アプリの種類 – パッケージ型 –

- パッケージ型では利用できる API は、パーミッションの種類に応じて異なる

## コミュニケーション API

### Network Information API

接続速度など、現在のネットワーク接続に関する基本的な情報を提供します。

### Bluetooth

WebBluetooth API は、デバイスの Bluetooth ハードウェアに対する低水準のアクセス手段を提供します。

### Mobile Connection API

信号強度やオペレーター情報など、デバイスのセルラー接続に関する情報を公開します。

### Network Stats API

データの使用状況をモニタリングして、そのデータを特権アプリケーションに公開します。

### TCP Socket API

低水準のソケットおよび SSL をサポートします。

### Telephony

アプリが電話を発着信したり、内蔵の電話機能のユーザインターフェイスを使用したりすることが可能です。

### WebSMS

アプリに対して、デバイスに保存しているメッセージのアクセスや管理だけでなく、SMS テキストメッセージの送受信を可能にします。

### WiFi Information API

信号強度、現在接続しているネットワークの名称、利用可能な WiFi ネットワークなどの情報を提供する特権 API です。

## ハードウェアアクセス API

### Ambient Light Sensor API

環境光センサーへのアクセス手段を提供します。これにより、アプリがデバイス付近の環境光のレベルを検出できます。

### Battery Status API

バッテリーの残量や、デバイスが電源に接続されて充電中であるかといった情報を提供します。

### Geolocation API

デバイスの物理的な位置に関する情報を提供します。

### Pointer Lock API

アプリがマウスへのアクセスをロックして、絶対座標ではなく移動量の差分にアクセスできます。これはゲームに対して重要です。

### Proximity API

ユーザの顔など、近くにある物体とデバイスとの近接度を検出できます。

### Device Orientation API

デバイスの向きが変化したときに通知します。

### Screen Orientation API

スクリーンの向きが変化したときに通知します。この API は、アプリにとってどの向きが望ましいかを示すことにも使用できます。

### Vibration API

ゲームでの触覚フィードバックなどのために、アプリがデバイスのバイブレーションハードウェアを制御できます。これは通知のためのバイブレーションといった用途向けではありません。通知については **Alarm API** をご覧ください。

### WebFM API

使用可能であれば、デバイスの FM ラジオ機能をサポートします。

### Camera API

デバイス内蔵のカメラを使用して、アプリが写真や動画を撮影できます。

### Power Management API

アプリがスクリーン、CPU、デバイスの電源などをオン・オフできるようにします。また、リソースロックイベントのリスニングや調査もサポートします。

[すべて見る...](#)

## データ管理 API

### FileHandle API

ロック機能とともに、書き込み可能なファイルをサポートします。

### IndexedDB

パフォーマンスの高い検索機能をサポートする、クライアントサイドの構造化データ用ストレージです。

### Contacts API

ユーザの連絡先データベースへのアクセス機能を提供します。連絡先情報の追加、読み取り、編集をサポートします。

### Device Storage API

現代のデスクトッププラットフォームにおける "ピクチャ" フォルダやモバイルデバイスにおけるフォトストレージなど、デバイスの中心部にアプリがファイルを作成、読み取り、変更することが可能です。

### Settings API

デバイスへ永続的に保存されるシステム全体の設定オプションを、アプリが調査したり変更したりすることが可能です。

## その他の API

### Alarm API

アプリが通知を予定することが可能です。特定の時刻にアプリを自動起動する機能もサポートします。

### Simple Push API

プラットフォームが特定のアプリケーションに通知メッセージを送信できます。

### Web Notifications

アプリケーションが、システムレベルで表示される通知を送信できます。

### Apps API

Open WebApps API は、Web アプリのインストールや管理をサポートします。また、アプリが課金情報を調べることができるにします。

### Web Activities

あるアプリが別のアプリにアクティビティを渡すことができます。例えばあるアプリが別のアプリに対して、写真を選択 (または作成) して返すように依頼するでしょう。一般的に、ユーザはどのアクティビティにどのアプリを使用するかを設定できます。

### WebPayment API

Web コンテンツが、仮想的な物品に対する支払いや払い戻しの処理を開始できます。

### Browser API

Web 技術を完全に使用する Web ブラウザの構築をサポートします (要するに、ブラウザの中にあるブラウザです)。

### Idle API

ユーザが能動的にデバイスを使用していないときに、アプリが通知を受け取れます。

### Permissions API

集中化された場所で、アプリの許可設定を管理します。設定アプリで使用されます。

### Time/Clock API

現在時刻の設定をサポートします。タイムゾーンは **Settings API** で設定します。

## WebAPI コミュニティ

これらの API について支援が必要になったとしても、それらの使用方法について他の開発者と語る手段がいくつかあります。

- WebAPI フォーラムで相談する:
  - [メーリングリストとして](#)
  - [Twitter](#)
  - [Stack Overflow](#)
  - [Google Group として](#)
  - [フィードとして](#)
- WebAPI IRC チャンネルを訪問する: [#webapi](#)

[🔗 ネットワークを忘れないでください...](#)

詳細: <https://developer.mozilla.org/ja/docs/WebAPI>

# アプリ種別とパーミッション

- 基本はブラウザと同じで、アクセス権を求めない限りは使える機能はブラウザと同等。
- 機能ごとに必要なパーミッションを要求。
- バージョン毎に利用できる API は異なる。  
例: Firefox OS 2.0 から 特権付でもカメラが使える等
- 一覧は WebIDE から閲覧できます。

名前	type:web	type:privileged	type:certified
geolocation	!	!	!
geolocation-noprompt	×	×	✓
camera	×	!	✓
alarms	✓	✓	✓
tcp-socket	×	✓	✓
network-events	×	×	✓
contacts	×	!	✓
device-storage:apps	×	×	✓
device-storage:crashes	×	×	✓
device-storage:pictures	×	!	✓

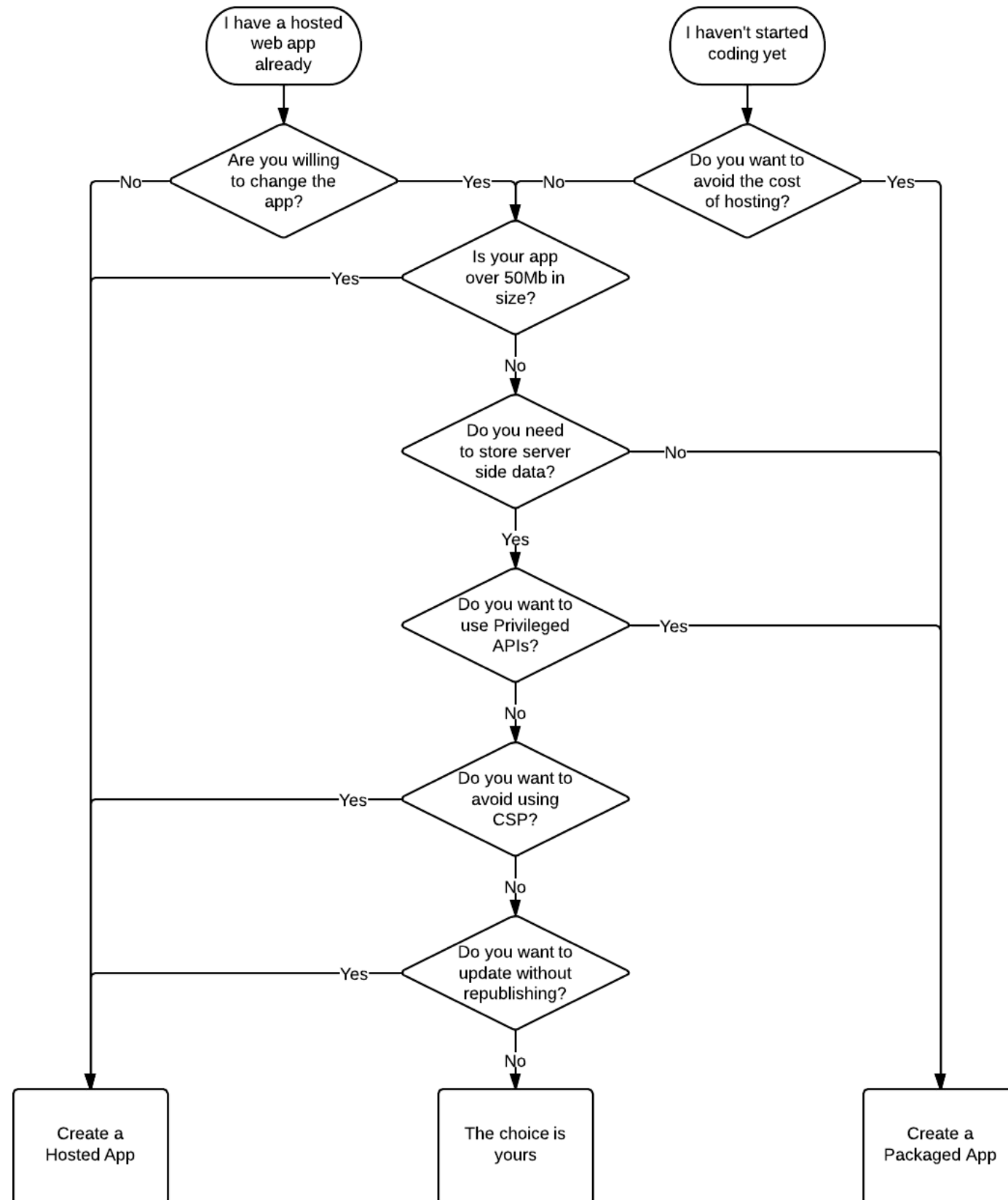
詳細: <https://developer.mozilla.org/ja/docs/WebAPI>



# アプリ種別とパーミッション

方式	種類	概要
ホスト型 (Web から読み込む)	単純な Web ページ	Web サイトそのもの。 ブラウザ内でアクセスして使う。
	インストール可能な Web ページ	Web サイトをインストールしたもの。 ブラウザの UI なしで単独起動する。
パッケージ型 (従来の OS 同様端末にインストールする)	単純なパッケージ型アプリ	Web サイト全体を ZIP パッケージにしているが追加権限を要求しないもの
	特権アプリ	マーケット審査を受けて追加の権限を許可されたアプリ
	内部[認定]アプリ	通信事業者や端末メーカーが特別に高い権限を許可している組み込みアプリ

# アプリごとのアプリ種別選択方針



詳細: <https://developer.mozilla.org/ja/Marketplace/Options/Packaged or hosted>

# アプリごとのアプリ種別選択方針

- すでに Web アプリを持っていて Web 側の変更をしたくない場合  
⇒ ホスト型アプリ
- ホスティングに費用をかけたくなかったり  
サーバーに情報を蓄積する必要がなかったり  
特権 API (カメラなど) を使いたい場合  
⇒ パッケージ型アプリ
- それ以外は開発の好きな方を選んでください  
※ただし、途中からアプリ種別を変更すると CSP などの理由から変更が難しくなることもあります。

# クロスプラットフォーム

- Cordova(PhoneGap)を利用することで、Firefox OS / Android / iOS / Amazon-Fire / Windows8 などの異なるプラットフォームを1つのソースコードから生成可能



# Cordova – Firefox OS 作成 –

- Cordova をダウンロードして、プロジェクト作成・ビルドが簡単にできます。

```
$ npm install -g cordova
$ cordova create test-app
$ cd test-app
$ cordova platform add firefoxos
$ cordova platform add android
$ cordova platform add ios
$ cordova prepare
$ cordova build firefoxos
```

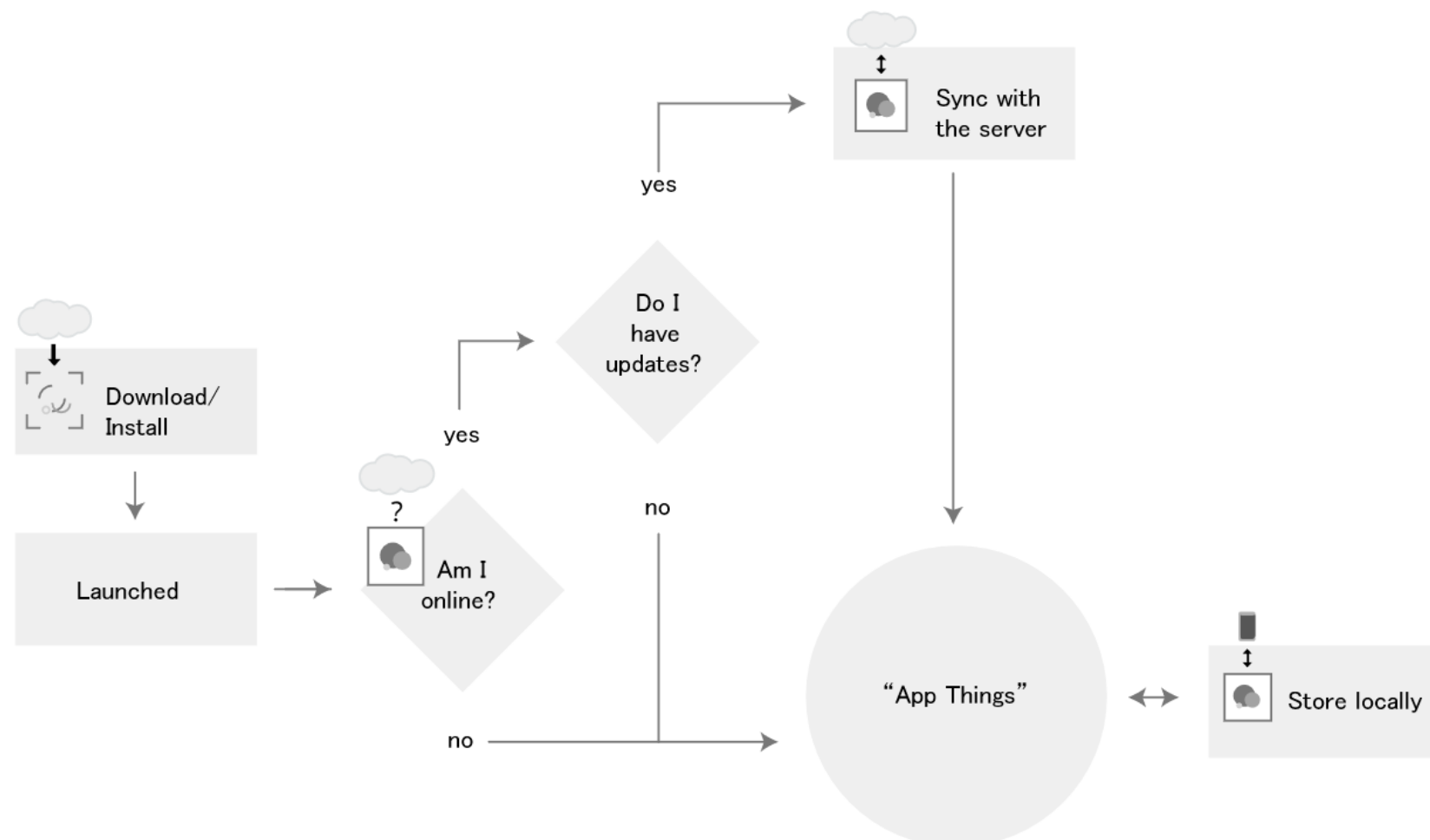
簡単にマルチプラットフォームのアプリができます！

※利用できる API 数がプラットフォームごとに違うので注意は必要(<http://mozilla-cordova.github.io/#pluginStatus>)

たとえば、ステータスバーのカスタマイズは Firefox OS では提供されていません。

# オフライン動作

- ホスト型だとオフライン時の動作に注意が必要  
(パッケージ型でもオフラインを考慮する必要あり)
- オフライン検知には NetworkInformation API
- オフライン時のデータなどは localStorage や ApplicationCache を利用してください



# CSP

- CSP (Content Security Policy):  
クロスサイトスクリプティングなどのセキュリティの脅威から保護するためのレイヤー

Firefox OS では、特権・内部[認定]アプリで適用される。

- CSP の制限を緩くすることはできない
- CSP の制限に引っかかると、以下の様にブロックされる。

Error: call to eval() blocked by CSP

- 詳細: <https://developer.mozilla.org/ja/Apps/CSP>

# CSP

- CSP で制限されること
  - リモートスクリプトの禁止
  - インラインスクリプトの禁止
  - javascript: URI の禁止
  - eval の禁止
  - 関数コンストラクタの禁止
  - setTimeout や setInterval による動的コード実行の禁止
  - リモート Web ワーカーの禁止
  - リモート URL を指すスクリプトタグ生成の禁止
  - プラグインの禁止
  - リモートスタイルの禁止



# CSP

## CSP 注意点

- リモートスクリプトの禁止  
CDN (Contents Deliver Network)からダウンロードする場合にエラーなるため、パッケージ内に同梱する
- インラインスクリプトの禁止  
HTML ページに JavaScript が記載されている場合は実行できないため、HTML と JavaScript のコードは分離する

例:

```
<input type="button" onClick="javascriptFunction();" id="btn1">  
=> document.getElementById( 'btn1' ).addEventListener( 'click', javascriptFunction);
```

- eval の禁止  
利用しているライブラリ内で、eval() にて文字列を式として評価している物は利用できないので、利用するフレームワークの調査をしてください。(CSP対応有無)  
例: Reactive.js / VueJS / Kendo UI
- リモートスタイルの禁止  
スクリプト同様、パッケージ外にある CSS のロードができないため、パッケージ内に同梱する

# シミュレータの注意点

ほかのモバイル開発環境と同じく、ハード等の制限があります

- メモリ容量 / CPU 速度などは実機と異なる
- 動作環境により利用できないコーデックがあります  
MP3 / AAC / H.264 / WebM
- 非対応 API があります  
Network Information / Proximity / Ambient Light ..etc

参考: [https://developer.mozilla.org/ja/docs/Tools/Firefox\\_OS\\_Simulator](https://developer.mozilla.org/ja/docs/Tools/Firefox_OS_Simulator)

まとめ

# まとめ

- Firefox OS アプリ開発は、Firefox があればすぐに開始できる
- Web コンテンツに Manifest ファイルを追加するだけ
- シミュレータ等のデバッグ環境もそろっている
- 利用するフレームワークは自由
- アプリの権限種類によって制約がある
- セキュリティ上の制約（CSP）が存在するが設計方針さえ立てていればアプリ作成可能
- クロスプラットフォームで作成するフレームワークも用意している

さいごに、

# ご紹介

## コミュニティ勉強会の紹介

### 「ついに Firefox OS が日本にやって来た！ — 関東 Firefox OS 勉強会 8<sup>th</sup>」

<http://firefox-os.doorkeeper.jp/events/20518>

### 「Firefox OS アプリ作成ハンズオン」

<http://firefox-os.doorkeeper.jp/events/20401>

Doorkeeper  
Firefox OS コミュニティー

イベント 2  
メンバー 79  
お問い合わせ

次のイベント  
Firefox OS アプリ作成ハンズオン

Firefox OS

Fx0 スペシャルナイトセミナー: ついに Firefox OS が日本にやって来た！ — 関東 Firefox OS 勉強会 8th

◎ 2015-02-20 (金) 19:30 - 21:00

Google カレンダーに追加

締め切り 2015-02-20  
参加費 無料

あと49人参加できます。

申し込む

次回イベント情報をメールで受け取る

参加者 51

kenta iyadomi  
池口 実有子  
菊地宗  
Makoto  
ega  
アヤト  
松田謙司  
ignisdios  
Hiroyuki Nakamura  
Noritada Shimizu  
星影

東京都渋谷区道玄坂1丁目12番1号 株式会社サイバーエージェント マークシティ13F セミナールームB

Doorkeeper  
Firefox OS コミュニティー

イベント 2  
メンバー 79  
お問い合わせ

次のイベント  
Fx0 スペシャルナイトセミナー: ついに Firefox OS が日本にやって来た！ — 関東 Firefox OS 勉強会 8th

BE THE FUTURE

Firefox OS

Firefox OS アプリ作成ハンズオン

◎ 2015-02-28 (土) 10:00 - 16:00

Google カレンダーに追加

締め切り 2015-02-28  
参加費 無料

事前登録不要(どなたでもご参加いただけます)

参加者 15

Makoto Oomori  
藤下正美