FIREFOX OS

WebAPIs & Apps

Using HTML5, CSS and JavaScript together with a number of APIs to build apps and customize the UI.

Firefox OS

Firefox OS is a new mobile operating system developed by Mozilla's **Boot to Gecko** (**B2G**) project. It uses a Linux kernel and boots into a Gecko-based runtime engine, which lets users run applications developed entirely using HTML, JavaScript, and other open web application APIs.

Firefox OS is currently under heavy development; we are constantly working on ways to make it easier for you to use and hack on Gaia and create apps. However, you need knowledge about systems in order to do things like build the entire Firefox OS stack, or flash a phone with a build of Firefox OS. Linked below are guides meant for Web developers interested in running and making changes to Gaia or developing apps to run on Firefox OS devices.

DOCUMENTATION ABOUT FIREFOX OS

Introduction to Firefox OS

Introductory information about what Firefox OS is and how it works.

Writing apps for Firefox OS

A tutorial guide to developing applications to run on Firefox OS devices.

Building and installing Firefox OS

A guide to building Firefox OS and installing it on your compatible device. This guide also covers building the Firefox OS emulator, for running Firefox OS on a computer.

Gaia

Documentation about Gaia, the user interface application for Firefox OS devices; this is a Web application running atop the Firefox OS software stack.

Gonk

Documentation about Gonk, the operating system layer underneath Gaia. This consists of a Linux kernel and a hardware abstraction layer to which Gecko communicates.

Gecko

Gecko is the layer of Firefox OS that provides the same open web standards implementation used by Firefox and Thunderbird, as well as many other applications.

Security

Documentation about security in Firefox OS; this includes topics about security devices from every perspective: for app developers, device integrators, and so forth.

Performance

Articles about optimizing Firefox OS applications.

Firefox OS architecture overview

An overview of how Firefox OS is structured.

GETTING HELP FROM THE COMMUNITY

If you're working with Firefox OS, or developing applications you'd like to run on Firefox OS devices, there are community resources to help you!

- Consult the Boot to Gecko project forum:
  - as a mailing list
  - as a newsgroup
  - as a Google Group
  - as a Web feed
- Ask your question on Mozilla's Boot to Gecko IRC channel: #b2g

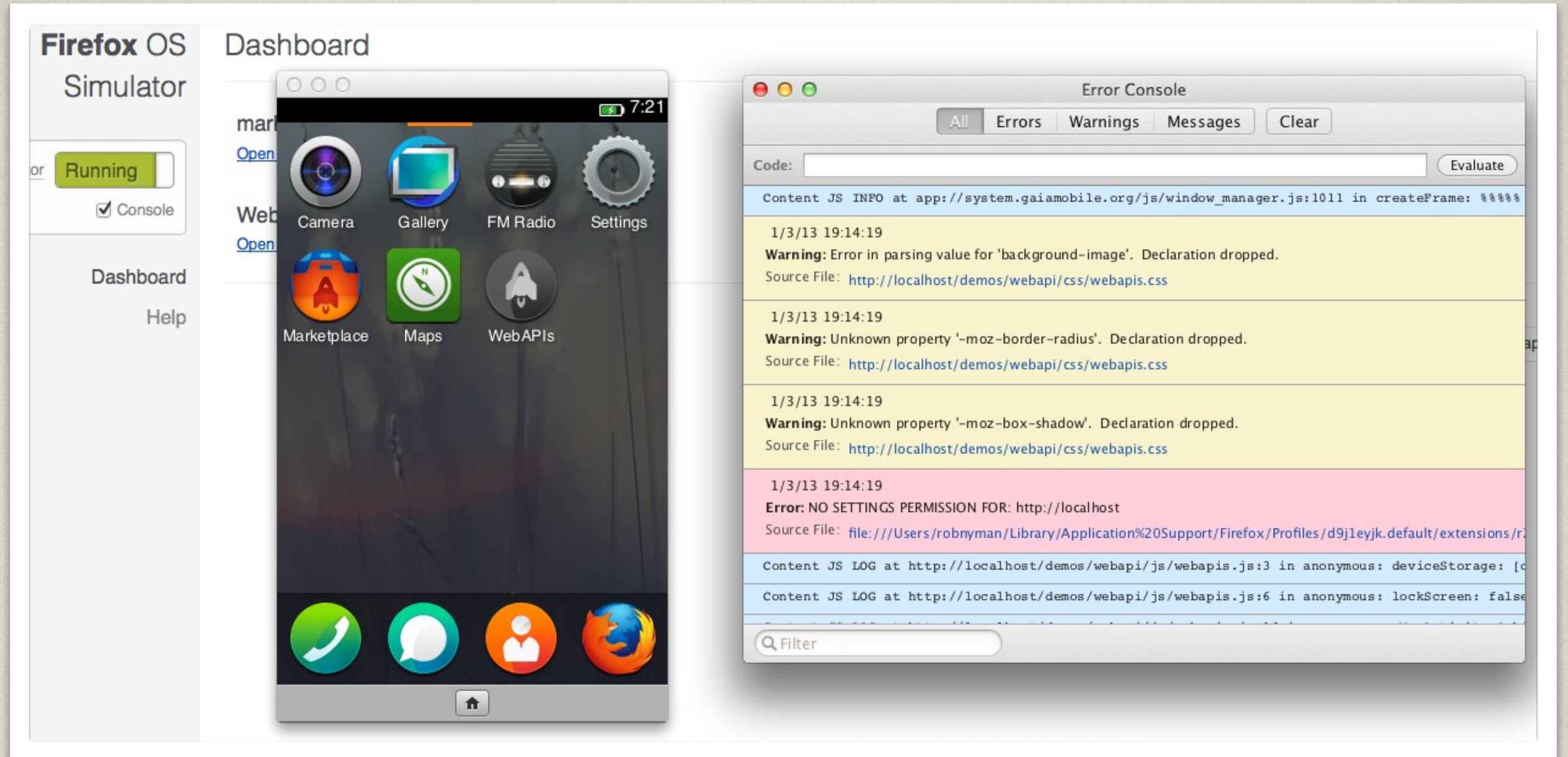
Don't forget about the *netiquette*...

RELATED TOPICS

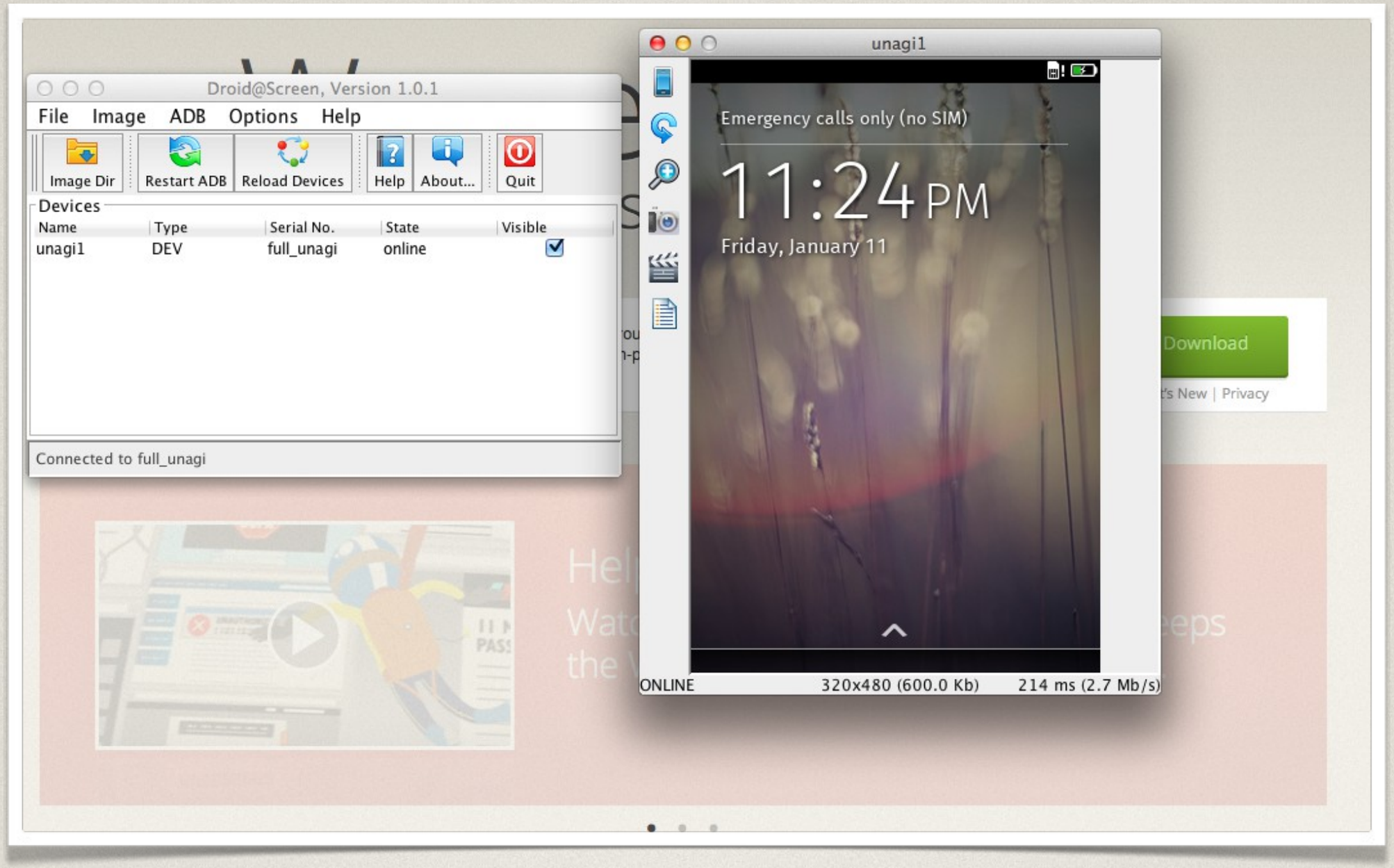
- Mobile
- HTML
- CSS
- JavaScript

RESOURCES

- Mozilla wiki FAQ
- Roadmap
- Feature support chart



https://addons.mozilla.org/firefox/addon/firefox-os-simulator/



Droid@Screen

http://blog.ribomation.com/droid-at-screen/

Open Web Apps

Web apps are apps built using standard Web technologies. They work in any modern Web browser.

The Open Web apps project proposes some small additions to existing sites to turn them into apps.

These apps run on desktop browsers and mobile devices.

We're running the Kuma wiki for MDN now. Please help us improve it by submitting bugs.

Getting started with making apps

Web apps are apps built using standard Web technologies. They work in any modern Web browser, and can be developed using your favorite tools. Some characteristics that distinguish Web apps from websites: Apps are installed by a user, they are self-contained and don't always require the chrome of a browser window, and they can be built to run offline. Gmail, Twitter, and Etherpad are Web apps.

The Open Web apps project proposes some small additions to existing sites to turn them into apps that run in a rich, fun, and powerful computing environment. These apps run on desktop browsers and mobile devices, and are easier for a user to discover and launch than Web sites. They have access to a growing set of novel features, such as synchronizing across all of a user's devices.

Before you start

If you are a first time developer looking to write web apps then you may want to verify the implementation state of the API.

Publishing the app

The only thing you have to do to create a Web app from a Web site is to add an *app manifest*. This is a JSON file that describes your app, including its name, its icons, and a human-readable description.

The manifest must be hosted from the same domain as your website, and must be served with a Content-Type of `application/x-web-app-manifest+json` (Note: this is currently not enforced by Firefox, but it is necessary for the Marketplace). For full details about the manifest refer to the documentation, and to help get started you can try out the online manifest checker.

TABLE OF CONTENTS

TAGS FILES

https://developer.mozilla.org/docs/Apps/Getting_Started

Steps to Take

1. Develop Web App using HTML5, CSS, & Javascript

2. Create an app manifest file

3. Publish/install the app

1.

Develop Web App using HTML5, CSS & JavaScript

Reuse any existing web site/app or develop from scratch with open web standards.

Utilize HTML5 features such as localStorage, offline manifest, IndexedDB and access Web APIs for more options.

Responsive web design for adapting to varying resolutions and screen orientation.

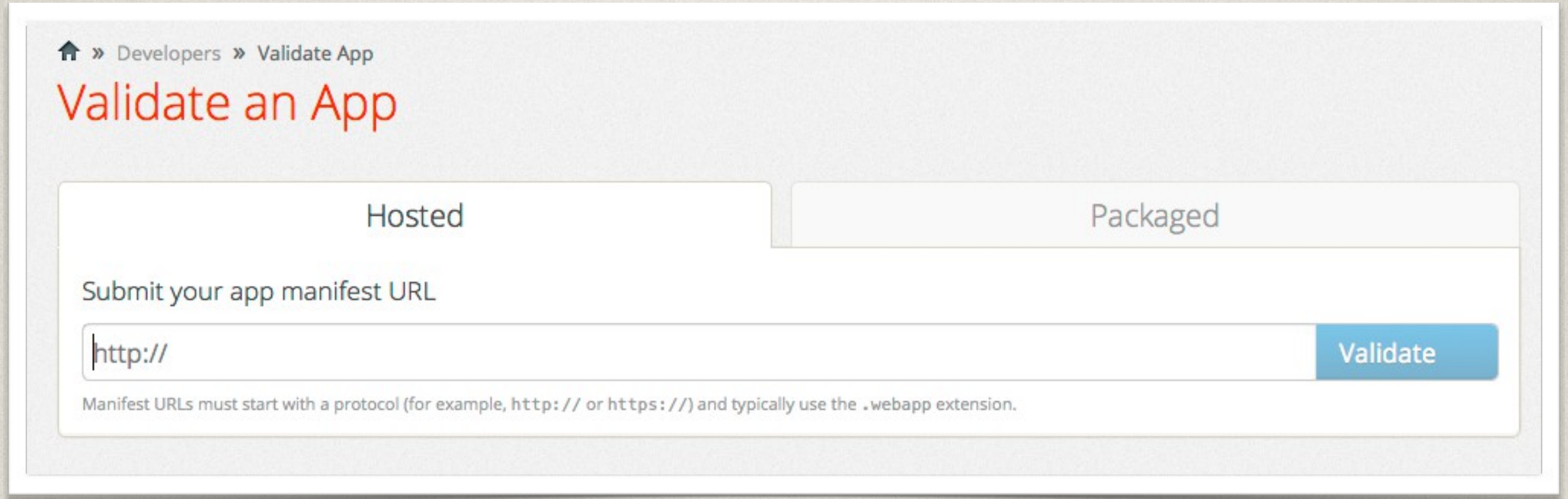
Guitar Hero.

Now on Linux

2. Getting an app manifest file

Getting a file with a `.webapp` extension

```
curl -I http://mozillalabs.com/manifest.webapp
```

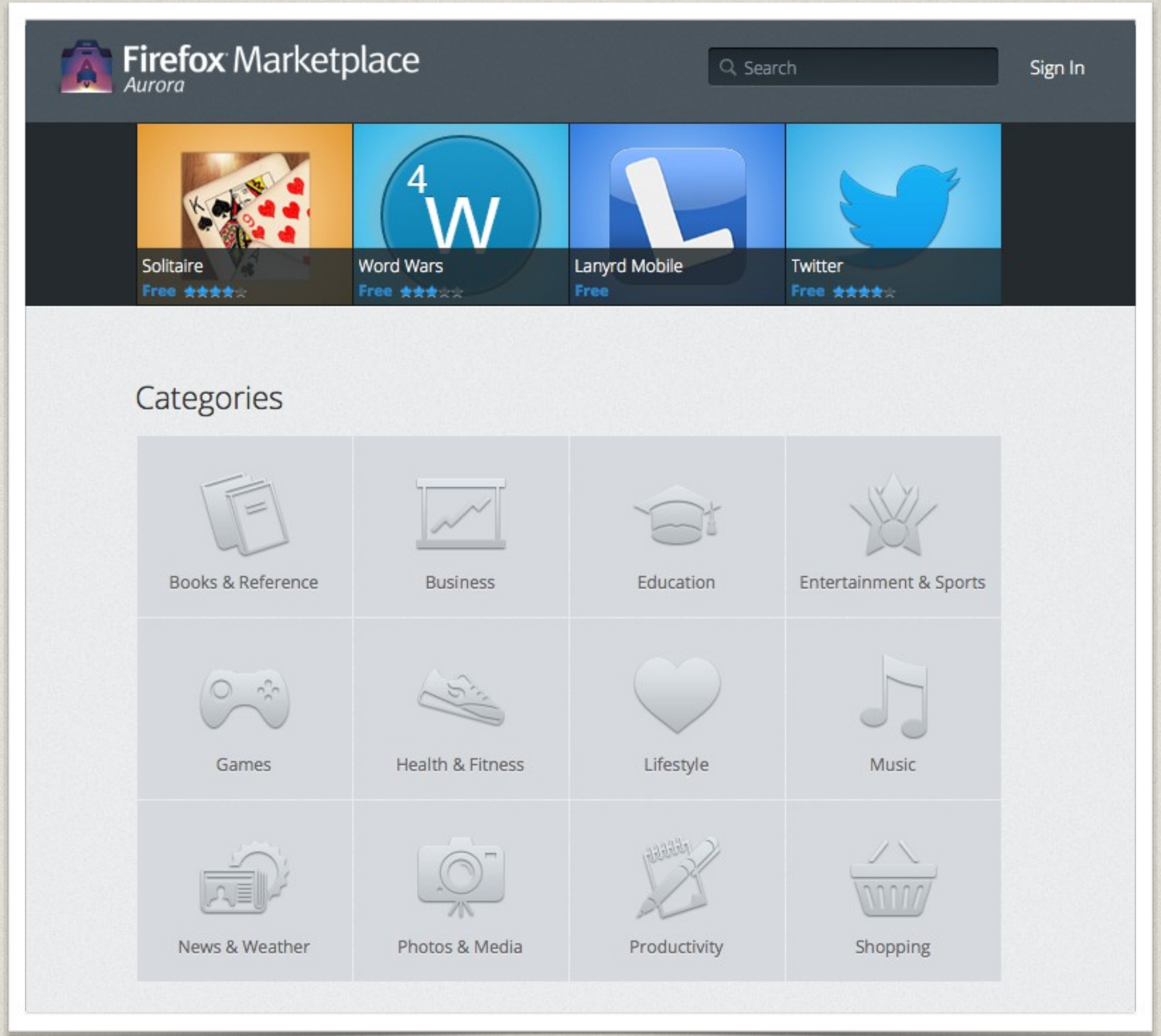


MANIFEST CHECKER

http://appmanifest.org/

3. Publish/install the app

Firefox Marketplace



https://marketplace.firefox.com/

https://marketplace.firefox.com/developers/

Installing/hosting the app

```
var request = navigator.mozApps.install(
    "http://mozillalabs.com/MozillaBall.webapp",
    {
        user_id:
    }
);

request.onsuccess = function() {
  // Success! Notification, launch page etc
}

request.onerror = function() {
  // Failed. this.error.name has details
}
```

```
var request = navigator.mozApps.installPackage(
    "http://mozillalabs.com/manifest.webapp"
);

request.onsuccess = function() {
  // Success!
}

request.onerror = function() {
  // Failed.
}
```

Packaged vs. Hosted Apps

A packaged app is an Open Web App that has all of its resources (HTML, CSS, JavaScript, app manifest, and so on) contained in a zip file, instead of having its resources on a Web server.

A packaged app is simply a zip file with the app manifest in its root directory. The manifest must be named manifest.webapp.

Can be privileged apps with more API access than hosted apps

Special protocol internal to the zip file: app://<uuid>

Manifest file must be named manifest.webapp

Resources are accessed from the zip file, which is stored on the device where the app is installed

Installed with a different mozApps API function: installPackage()

Enforce a specific Content Security Policy for all application content

Can embed remote content in iframes, but that content will not have access to privileged APIs nor will it have the default CSP applied to it

Have an update process for getting new versions of the app to users. Hosted apps do not need this process.

App development for Web developers

Minimum requirements

If you are a Web developer and you have a website or Web application that you would like to make into an installable Open Web app, there is *technically* very little that you need to do. The minimum requirements are few:

1. Create an app manifest.
2. Serve the app manifest in a file with a file extension of `.webapp`. Set the `Content-Type` header to `application/x-web-app-manifest+json`.
3. Publish the app, either on your own site or in an app store (or both). Publishing it yourself requires adding some code to your site to manage installing and updating the app in users' browsers.

TABLE OF CONTENTS

Optional features

Philosophically, the idea of an installable Open Web app is much more than simply adding a manifest to your site. Web standards technologies can be viewed as a full-blown application platform that happens to use a browser engine for rendering user interfaces and interpreting code, and happens to use Web protocols for communicating with a server. Mozilla offers "Web runtime" executables for various platforms so that apps can run in their own window, without a browser window frame.

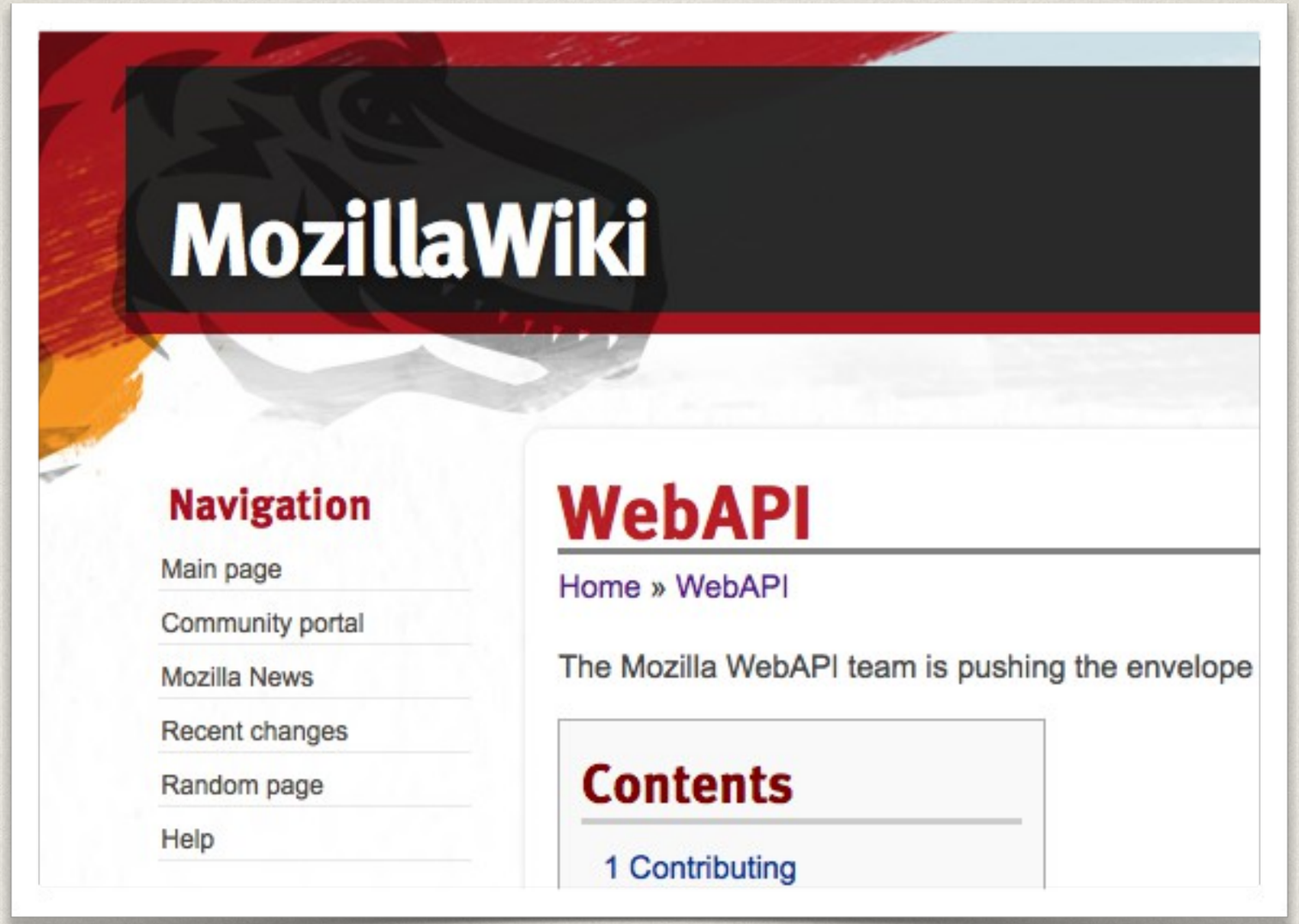
To "appify" a website, there are many application-specific questions to consider:

- Should my app work when not connected to the Web?
- How does my app use data, and how does it need to be stored?
- Can my app's performance benefit from advanced platform features like Web Workers or WebSockets?
- And many more

https://developer.mozilla.org/docs/Apps/For_Web_developers

WebAPIs

The Mozilla WebAPI team is pushing the envelope of the web to include --- and in places exceed --- the capabilities of competing stacks.



https://wiki.mozilla.org/WebAPI

Security Levels

Granted by default

Safe web APIs that don’t expose privacy sensitive data. WebGL, fullscreen, audio, etc.

Granted by user

location, camera, file system access

Granted when installed

No quota for localStorage, IndexedDB, offline cache

Granted by authorized store

Privacy and security sensitive APIs such as Contacts API

Verified by signature

Highly privileged APIs such as radio access (dialer)

Web Content

Regular web content

Installed Web App

A regular web app

Privileged Web App

More access, more responsibility

Certified Web App

Device-critical applications

Planned for initial release of B2G (aka Basecamp)

API	Bugs	Description	Progress	Availability		
WebTelephony	bug 674726	Allow placing and answering phone calls as well as build in-call UI.	Security Design Complete	D	A	B
Vibration API (W3C)	bug 679966	Control device vibration for things like haptic feedback in games. Not intended to solve things like vibration for notification.	Done on B2G and Android. Standard in progress. Security Design Complete	D	A	B
WebSMS	bug 674725	Send/receive SMS messages as well as manage messages stored on device.	Done on Android though might not ship there for security reasons. Done for B2G. Security Design Complete	D	A	B
Idle API	bug 715041	Get notifications when user is idle.	Implemented. Security Design Complete	D	A	B
Screen Orientation	bug 720794 bug 740188 bug 673922	Get notification when screen orientation changes as well as control which screen orientation a page/app wants.	Implemented! Security Design Complete	D	A	B
Settings API	bug 678695	Set system-wide configurations that are saved permanently on the device.	Implementation done for content, chrome in progress. Security Design Complete	D	A	B
Power Management API	bug 708964	Turn on/off screen, cpu, device power, etc. Listen and inspect resource lock events.	API design and implementation in progress. Security Design Complete	D	A	B

https://wiki.mozilla.org/WebAPI#Planned_for_initial_release_of_B2G_.28aka_Basecamp.29

```
"permissions": {
    "contacts": {
        "description": "Required for autocompletion in the share screen",
        "access": "readcreate"
    },
    "alarms": {
        "description": "Required to schedule notifications"
    }
}
```

AlarmAPI

BrowserAPI

Contacts

device-storage:music/device-storage:videos/device-storage:pictures/device-storage:sdcard:

Add, read, or modify files stored at a central location on the device. access property required: one of readonly, readwrite, readcreate, or createonly.

FMRadio

geolocation

systemXHR

TCP Socket API

wake-lock-screen

NEED PERMISSION

Vibration API (W3C)	Web Activities
Screen Orientation	Push Notifications API
Geolocation API	WebFM API
Mouse Lock API (W3C)	WebPayment
Open WebApps	IndexedDB (W3C)
Network Information API (W3C)	Ambient light sensor
Battery Status API (W3C)	Proximity sensor
Alarm API	Notification

REGULAR APIS

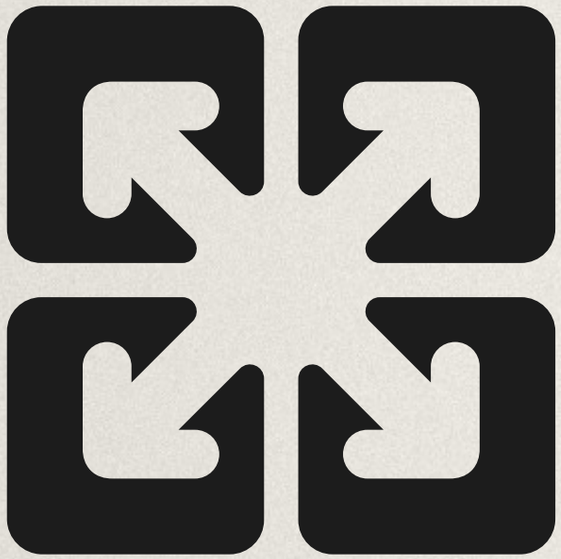
BATTERY STATUS

API

```
var battery = navigator.battery;
if (battery) {
        var batteryLevel = Math.round(battery.level * 100) + "%",
            charging = (battery.charging)? "" : "not ",
            chargingTime = parseInt(battery.chargingTime / 60, 10,
            dischargingTime = parseInt(battery.dischargingTime / 60, 10);

    // Set events
    battery.addEventListener("levelchange", setStatus, false);
    battery.addEventListener("chargingchange", setStatus, false);
    battery.addEventListener("chargingtimechange", setStatus, false);
    battery.addEventListener("dischargingtimechange", setStatus, false);
}
```

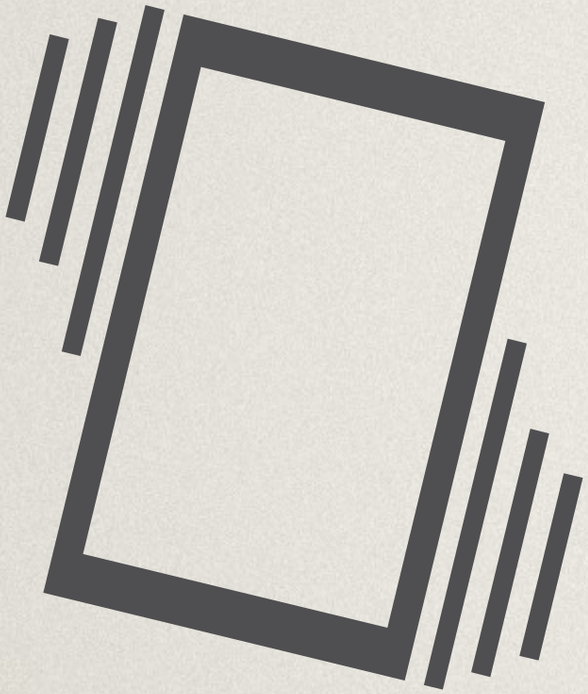
NOTIFICATION

```
var notification = navigator.mozNotification;
notification.createNotification(
    "See this",
    "This is a notification",
    iconURL
);
```



SCREEN ORIENTATION API

```
// Portrait mode:
screen.mozLockOrientation("portrait");

/*
    Possible values:
        "landscape"
        "portrait"
        "landscape-primary"
        "landscape-secondary"
        "portrait-primary"
        "portrait-secondary"
*/
```

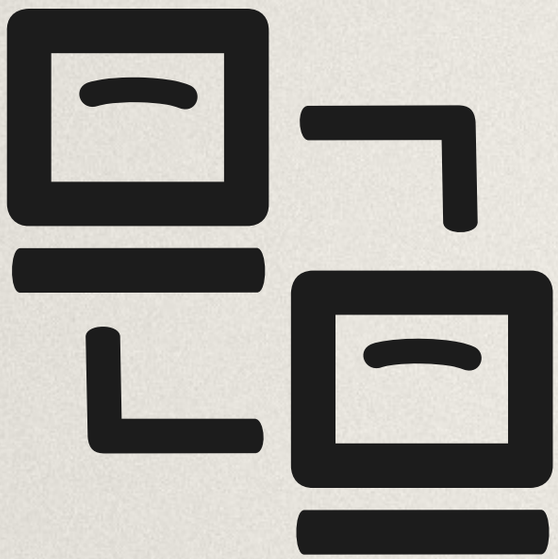


VIBRATION API

```
// Vibrate for one second
navigator.vibrate(1000);

// Vibration pattern [vibrationTime, pause,…]
navigator.vibrate([200, 100, 200, 100]);

// Vibrate for 5 seconds
navigator.vibrate(5000);

// Turn off vibration
navigator.vibrate(0);
```

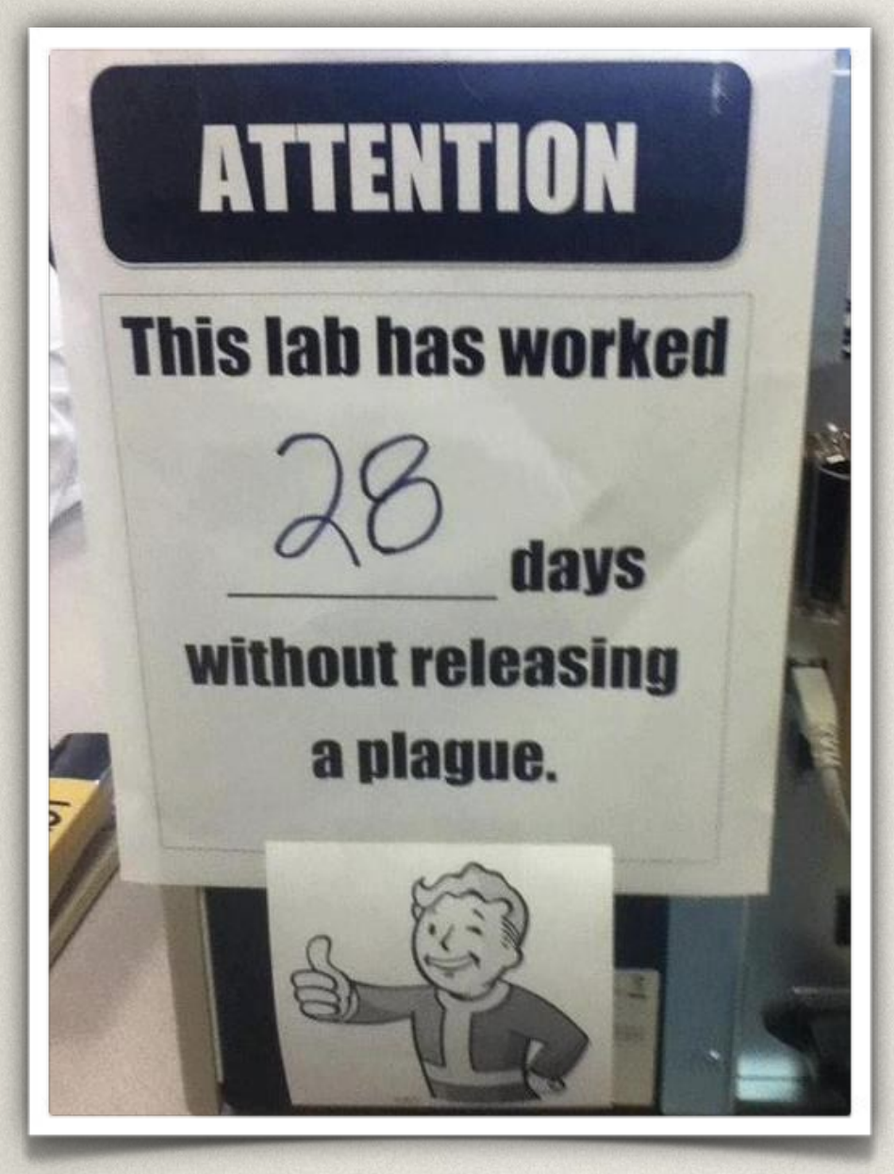
WEB PAYMENTS

```
var pay = navigator.mozPay(paymentToken);
pay.onsuccess = function (event) {
    // Weee! Money!
};
```



NETWORK INFORMATION API

```
var connection = window.navigator.mozConnection,
    online = connection.bandwidth > 0,
    metered = connection.metered;
```


ATTENTION
This lab has worked
28
days
without releasing
a plague.

ALARM API

```
var alarmId1,
    request = navigator.mozAlarms.add(
        new Date("May 15, 2012 16:20:00"),
        "honorTimezone",
        {
            mydata: "my event"
        }
    );

request.onsuccess = function (event) {
    alarmId1 = event.target.result;
};

request.onerror = function (event) {
    console.log(event.target.error.name);
};
```

```
var request = navigator.mozAlarms.getAll();

request.onsuccess = function (event) {
    console.log(JSON.stringify(event.target.result));
};

request.onerror = function (event) {
    console.log(event.target.error.name);
};
```

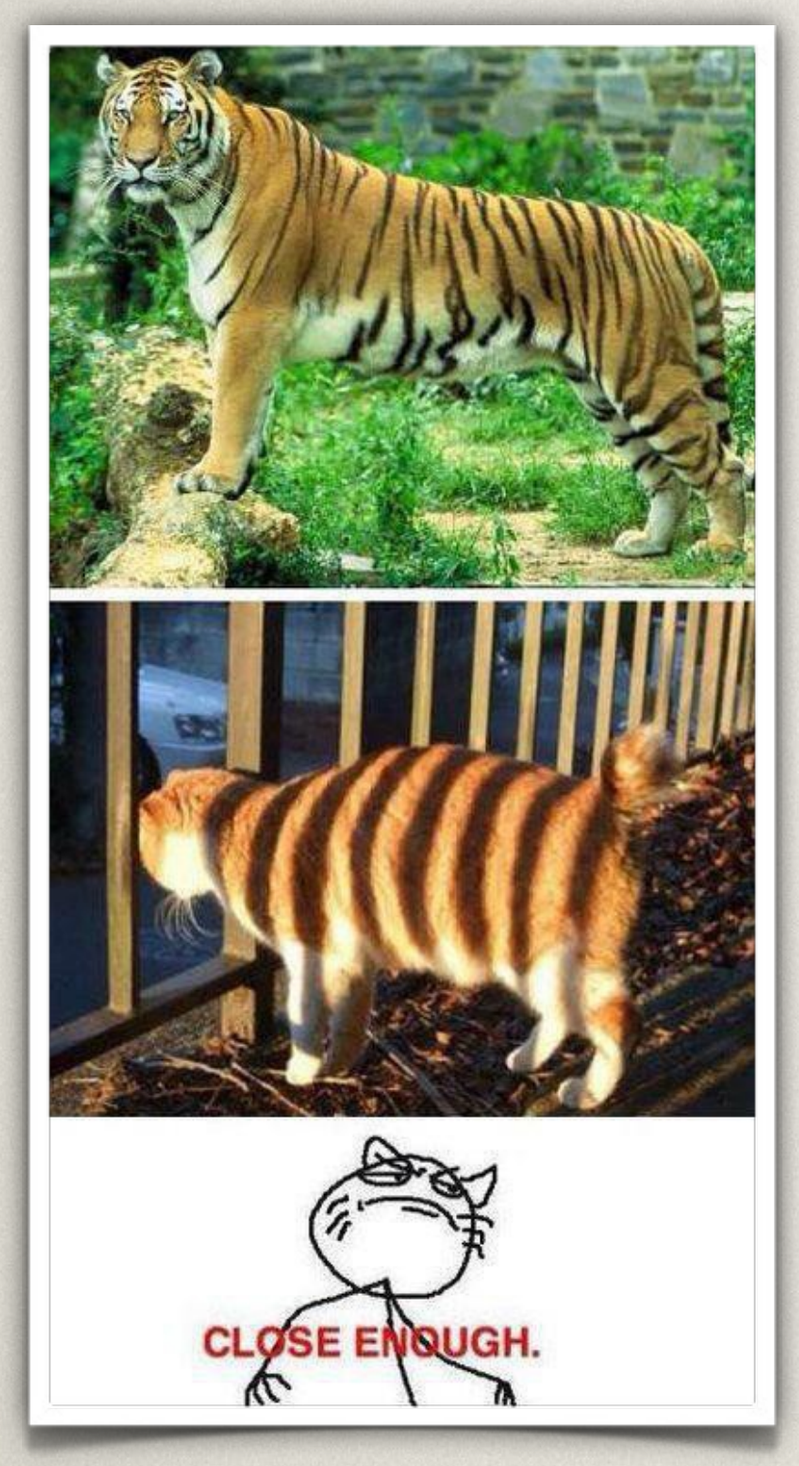
```
navigator.mozAlarms.remove(alarmId1);
```

```
navigator.mozSetMessageHandler(
    "alarm",
    function (message) {
        // Note: message has to be set in the manifest file
        console.log("Alarm fired: " + JSON.stringify(message));
    }
);
```

```
{
    "messages": ["alarm"]
}
```

DEVICEPROXIMITY

```
window.addEventListener("deviceproximity", function (event) {
    // Current device proximity, in centimeters
    console.log(event.value);

    // The maximum sensing distance the sensor is
    // able to report, in centimeters
    console.log(event.max);

    // The minimum sensing distance the sensor is
    // able to report, in centimeters
    console.log(event.min);
});
```



AMBIENT LIGHT EVENTS

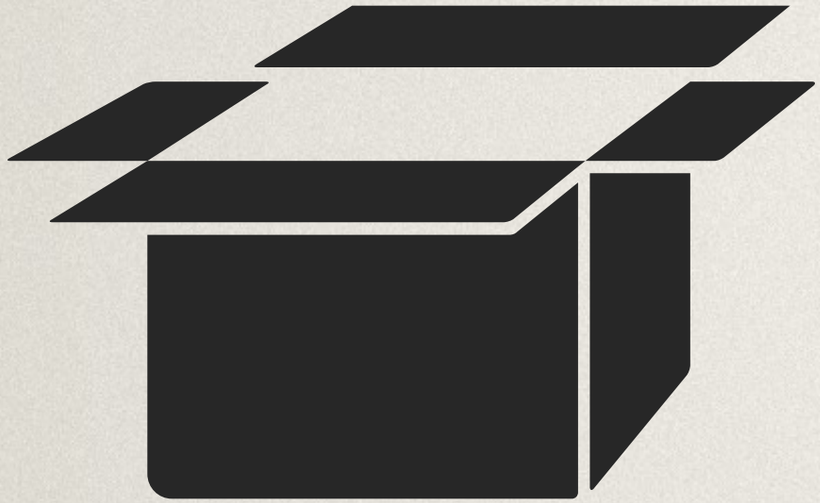
```
window.addEventListener("devicelight", function (event) {
    // The level of the ambient light in lux
    console.log(event.value);
});
```

```
window.addEventListener("lightlevel", function (event) {
    // Possible values: "normal", "bright", "dim"
    console.log(event.value);
});
```

```
window.addEventListener("devicelight", function (event) {
    //  The lux values for "dim" typically begin below 50,
    // and the values for "bright" begin above 10000
    console.log(event.value);
});
```

- Device Storage API
- Browser API
- TCP Socket API
- Contacts API
- systemXHR

PRIVILEGED APIS

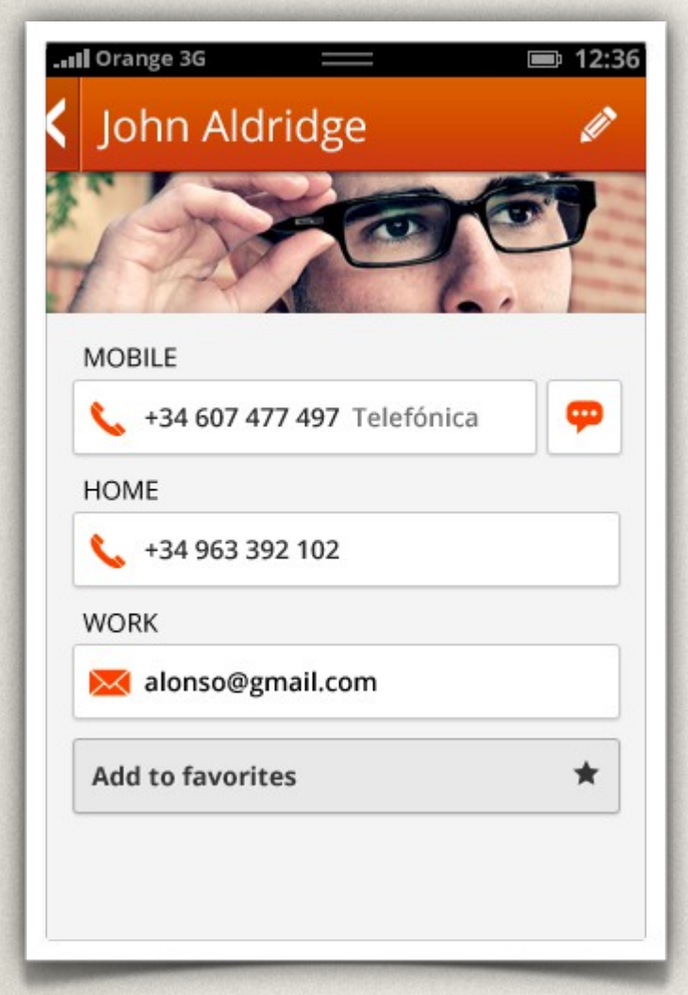


DEVICE STORAGE API

```
var deviceStorage = navigator.getDeviceStorage("videos");
```

```
// "external", "shared", or "default".
deviceStorage.type;

// Add a file - returns DOMRequest with file name
deviceStorage.add(blob);

// Same as .add, with provided name
deviceStorage.addNamed(blob, name);

// Returns DOMRequest/non-editable File object
deviceStorage.get(name);

// Returns editable FileHandle object
deviceStorage.getEditable(name);

// Returns DOMRequest with success or failure
deviceStorage.delete(name);

// Enumerates files
deviceStorage.enumerate([directory]);

// Enumerates files as FileHandles
deviceStorage.enumerateEditable([directory]);
```

```
var storage = navigator.getDeviceStorage("videos"),
    cursor = storage.enumerate();

cursor.onerror = function() {
  console.error("Error in DeviceStorage.enumerate()", cursor.error.name);
};

cursor.onsuccess = function() {
    if (!cursor.result)
        return;

    var file = cursor.result;

    // If this isn't a video, skip it
    if (file.type.substring(0, 6) !== "video/") {
        cursor.continue();
        return;
    }

    // If it isn't playable, skip it
    var testplayer = document.createElement("video");
    if (!testplayer.canPlayType(file.type)) {
        cursor.continue();
        return;
    }
};
```

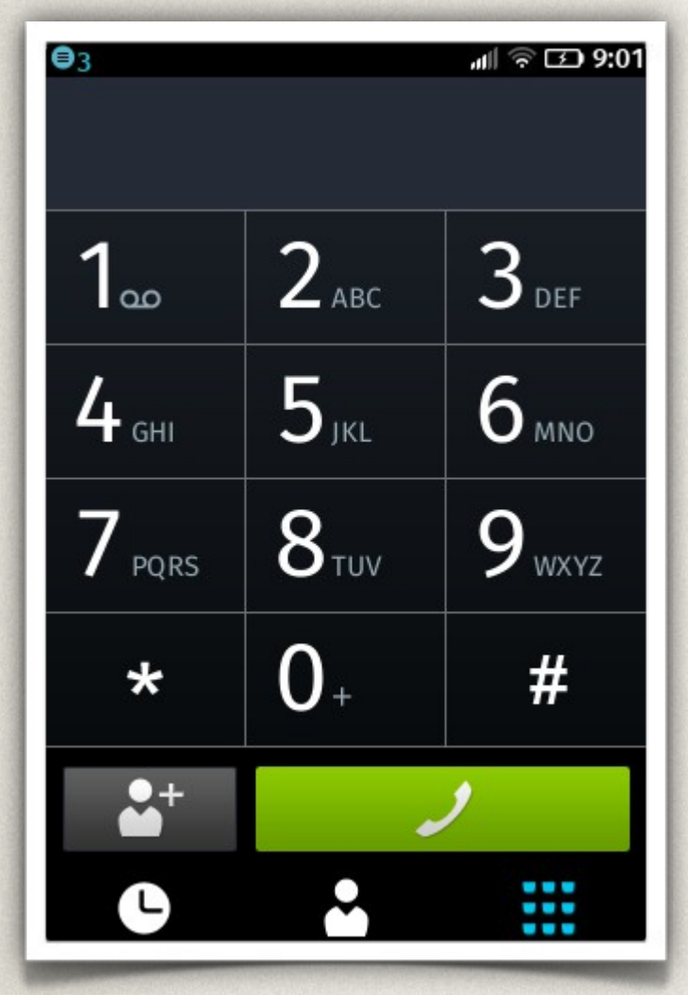


CONTACTS API

```
var contact = new mozContact();
contact.init({name: "Tom"});

var request = navigator.mozContacts.save(contact);
request.onsuccess = function() {
    console.log("Success");
};

request.onerror = function() {
    console.log("Error")
};
```

WebTelephony	WebBluetooth
WebSMS	Permissions API
Idle API	Network Stats API
Settings API	Camera API
Power Management API	Time/Clock API
Mobile Connection API	Attention screen
WiFi Information API	Voicemail

CERTIFIED APIS

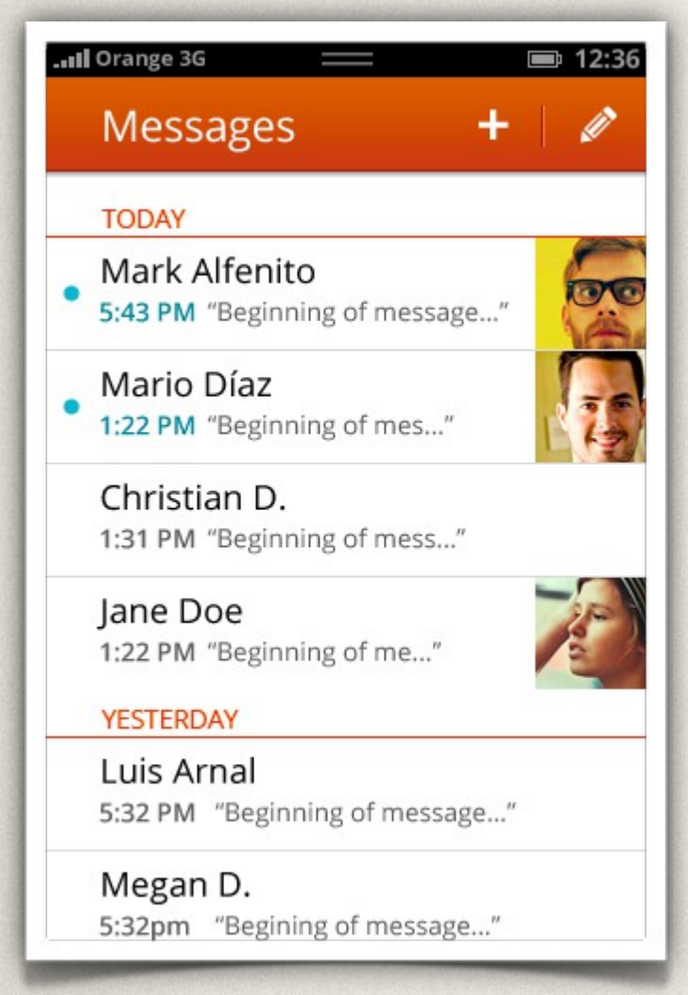


WEBTELEPHONY

```
// Telephony object
var tel = navigator.mozTelephony;

// Check if the phone is muted (read/write property)
console.log(tel.muted);

// Check if the speaker is enabled (read/write property)
console.log(tel.speakerEnabled);
```

```
// Place a call
var cal = tel.dial("123456789");
```

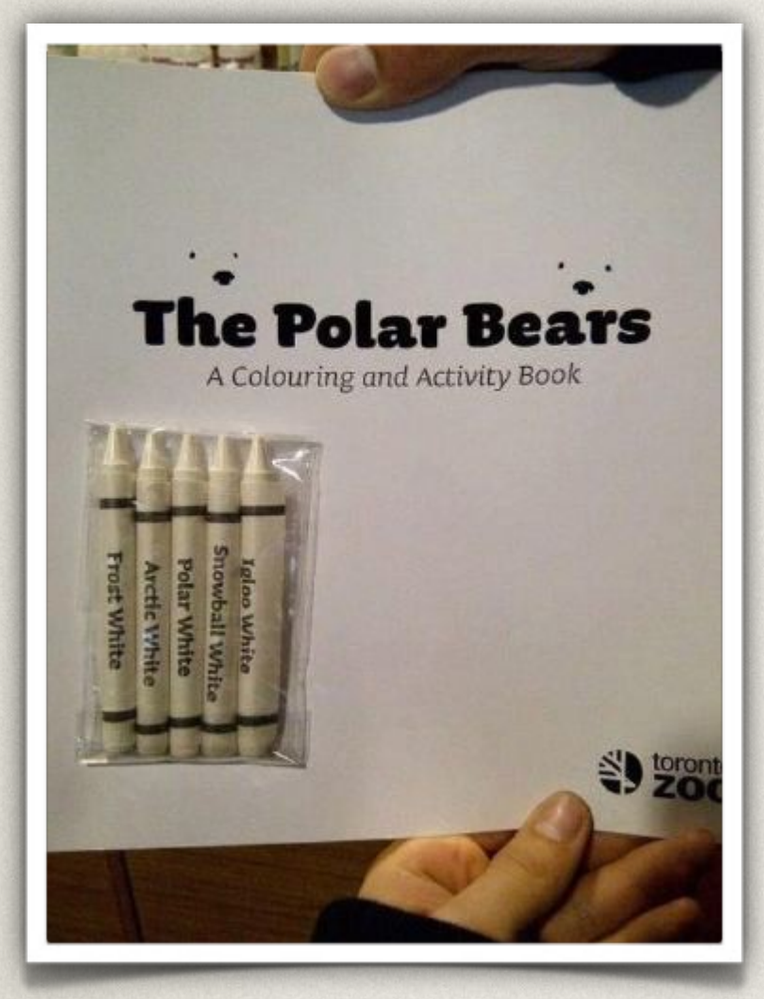
```
// Receiving a call
tel.onincoming = function (event) {
    var incomingCall = event.call;

    // Get the number of the incoming call
    console.log(incomingCall.number);

    // Answer the call
    incomingCall.answer();
};

// Disconnect a call
call.hangUp();


// Iterating over calls, and taking action depending on their
changed status
tel.oncallschanged = function (event) {
    tel.calls.forEach(function (call) {
        // Log the state of each call
        console.log(call.state);
    });
};
```


Orange 3G
12:36
Messages
TODAY
Mark Alfenito
5:43 PM "Beginning of message..."
Mario Díaz
1:22 PM "Beginning of mes..."
Christian D.
1:31 PM "Beginning of mess..."
Jane Doe
1:22 PM "Beginning of me..."
YESTERDAY
Luis Arnal
5:32 PM "Beginning of message..."
Megan D.
5:32pm "Begining of message..."

WEBSMS

```
// SMS object
var sms = navigator.mozSMS;

// Send a message
sms.send("123456789", "Hello world!");
```

```
// Recieve a message
sms.onreceived = function (event) {
    // Read message
    console.log(event.message);
};
```



WEB ACTIVITIES


1:23 PM
Firefox OS Boilerplate
WebActivities
pick
Wallpaper
Gallery
Camera
Cancel

```
{
    "activities": {
        "share": {
            "filters": {
                type: ["image/png", "image/gif"],
            }
            "href": "sharing.html",
            "disposition": "window"
        }
    }
}
```

```
var activity = new MozActivity({
    name: "view",
    data: {
        type: "image/png",
        url: ...
    }
});

activity.onsuccess = function () {
    console.log("Showing the image!");
};

activity.onerror = function () {
    console.log("Can't view the image!");
};
```

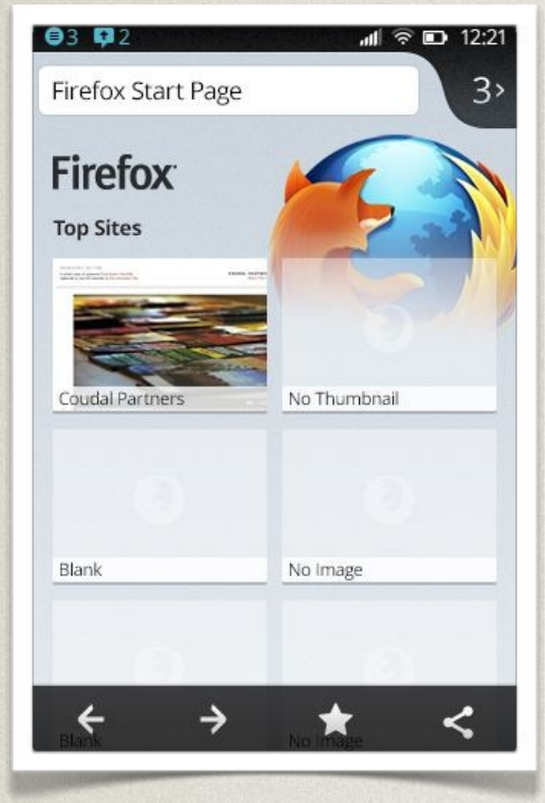
```
var register = navigator.mozRegisterActivityHandler({
    name: "view",
    disposition: "inline",
    filters: {
        type: "image/png"
    }
});

register.onerror = function () {
    console.log("Failed to register activity");
}
```

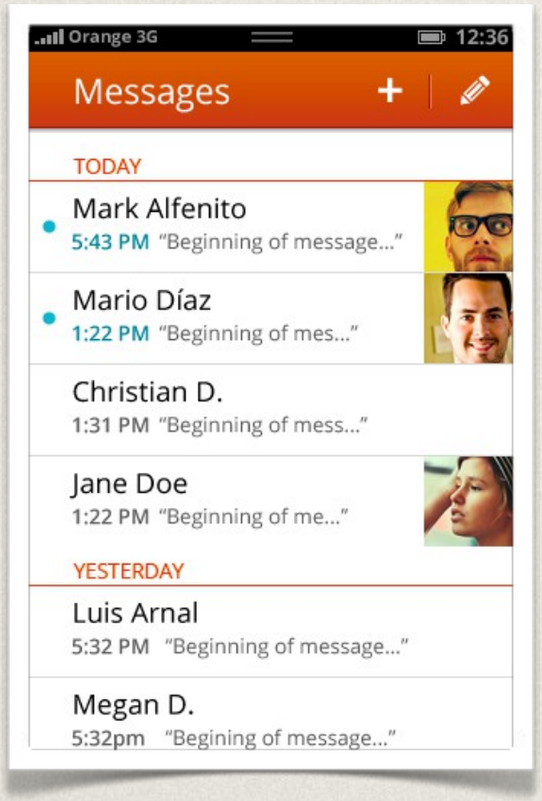
```
navigator.mozSetMessageHandler("activity", function (a) {
    var img = getImageObject();
    img.src = a.source.url;
    // Call a.postResult() or a.postError() if
    // the activity should return a value
});
```

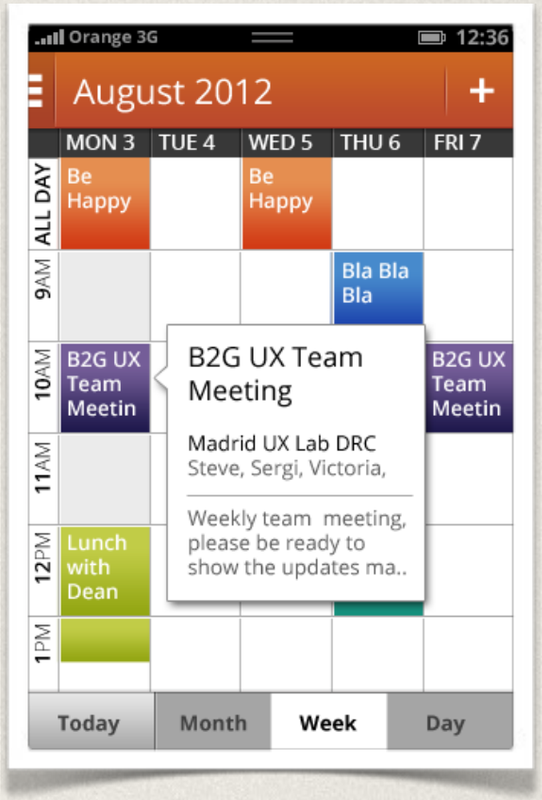
Future APIs

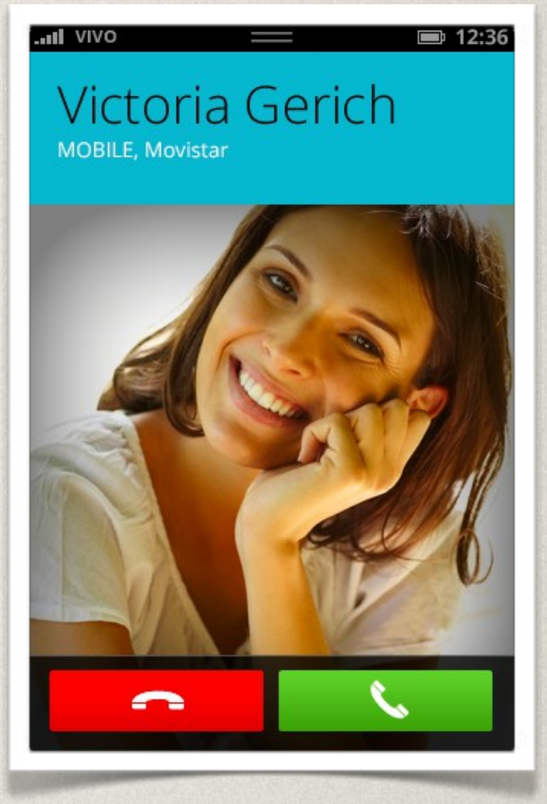
Resource lock API	Spellcheck API
UDP Datagram Socket API	LogAPI
Peer to Peer API	Keyboard/IME API
WebNFC	WebRTC
WebUSB	FileHandle API
HTTP-cache API	Sync API
Calendar API	

Web Apps from Mozilla


3 2
12:21
Firefox Start Page
3
Firefox
Top Sites
Coudal Partners
No Thumbnail
Blank
No Image
Blank
No Image


Orange 3G
12:36
Messages
TODAY
Mark Alfenito
5:43 PM "Beginning of message..."
Mario Díaz
1:22 PM "Beginning of mes..."
Christian D.
1:31 PM "Beginning of mess..."
Jane Doe
1:22 PM "Beginning of me..."
YESTERDAY
Luis Arnal
5:32 PM "Beginning of message..."
Megan D.
5:32pm "Begining of message..."


Orange 3G
12:36
August 2012
MON 3
TUE 4
WED 5
THU 6
FRI 7
ALL DAY
9AM
10AM
11AM
12PM
1PM
Be Happy
Be Happy
Bla Bla Bla
B2G UX Team Meetin
B2G UX Team Meetin
Lunch with Dean
B2G UX Team Meeting
Madrid UX Lab DRC
Steve, Sergi, Victoria,
Weekly team meeting, please be ready to show the updates ma..
Today
Month
Week
Day


VIVO
12:36
Victoria Gerich
MOBILE, Movistar

- Dialer
- Contacts
- Settings
- SMS
- Web browser
- Gallery
- Video Player
- Music Player
- E-mail (POP, IMAP)
- Calendar
- Alarm Clock
- Camera
- Notes
- First Run Experience
- Notifications
- Home Screen
- Mozilla Marketplace
- System Updater
- Localization Support

Search or type a command | Explore | Gist | Blog | Help | robnyman

PUBLIC mozilla-b2g / gaia

Watch | Star 965 | Fork 455

Code | Network | Pull Requests 91 | Issues 333 | Graphs

Gaia is a HTML5-based Phone UI for the Boot 2 Gecko Project — Read more
https://wiki.mozilla.org/B2G

Clone in Mac | ZIP | HTTP | SSH | Git Read-Only | git://github.com/mozilla-b2g/gaia.git | Read-Only access

branch: master | Files | Commits | Branches 4 | Tags 2

gaia / 1000+ commits

Merge pull request #7234 from dominickuo/music-open-received-file ...

dominickuo authored 4 hours ago — latest commit ac6440dc6a

apps	4 hours ago	Merge pull request #7234 from dominickuo/music-open-received-file [dominickuo]
build	3 days ago	Bug 821296 - Allow single files to be packaged as shared resources an… [gabrielesvelto]
dictionaries	4 months ago	Remove en dictionary. We already have en_gb and en_us. [andreasgal]
external-apps	9 days ago	Merge pull request #7153 from dclarke/master [dclarke]
locales	4 months ago	Merged with master [lodr]
media-samples	6 months ago	move sample photos from /sdcard/Pictures back to /sdcard/DCIM [davidflanagan]
shared	2 days ago	Merge pull request #7237 from leibovic/manifest-locales [leibovic]
showcase_apps	3 days ago	Bug 822565 - Remove the calculator app [vingtetun]
test_apps	3 days ago	Merge pull request #6907 from lissyx/fix-geoloc-buttons [vingtetun]
test_media	6 months ago	Add a directory to allow for Test Media to be delivered onto the device [dclarke]
tests	3 days ago	Bug 820188 Email app integration tests a=test-only r=squib [willkg]

Hacking Gaia for Firefox OS, part 1

on January 14, 2013 by Schalk Neethling in Firefox OS 25 comments

So I guess pretty much everyone is aware of this awesome new product being developed at Mozilla that is making a lot of noise in the mobile world. I am of course referring to FirefoxOS (code named Boot2Gecko) and if you have not heard of it yet, then you need to head over to the product page right now and also, the Wiki for a more technical overview.

Not to go into too much detail and restate what is already on the wiki, FirefoxOS is made up of three distinct parts:

- Gonk – The 'OS' i.e. low level Linux kernel and hardware abstraction layer (HAL)
- Gecko – The application runtime, also present in your desktop and mobile Firefox
- Gaia – The user facing front end of the OS

ABOUT THE AUTHOR

Schalk Neethling
Front-End Engineer at Mozilla works on the Web Tools team as well as a recent addition to the Gaia team. An evangelist, writer and developer with a passion for making the web better for everyone.

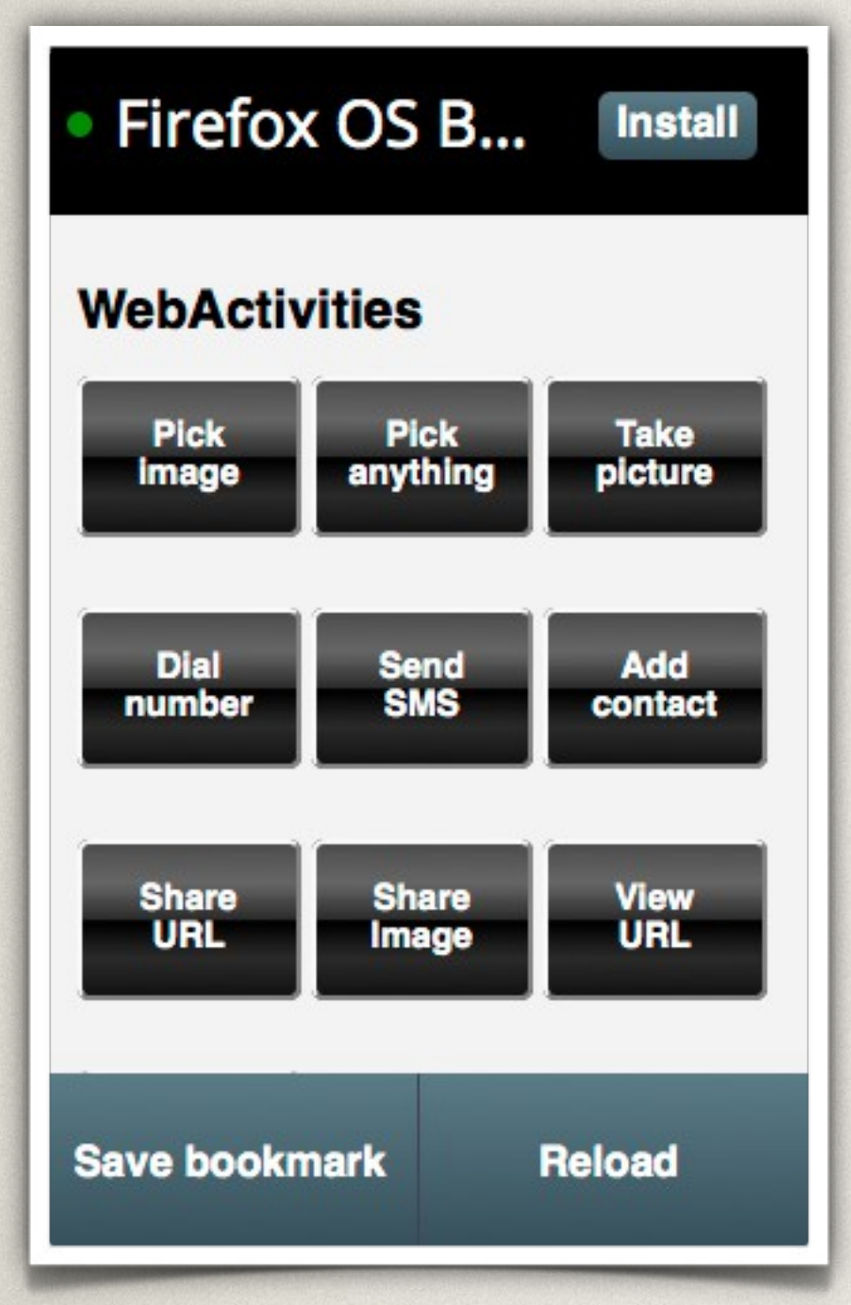
Read more articles by Schalk Neethling…

ARTICLES BY CATEGORY

35 Days (45) Interviews (2)
@font-face (9) JägerMonkey (5)

https://hacks.mozilla.org/2013/01/hacking-gaia-for-firefox-os-part-1/

Installing apps



FIREFOX OS BOILERPLATE APP

https://github.com/robnyman/Firefox-OS-Boilerplate-App

MOZILLA
HACKS.MOZILLA.ORG

Search hacks.mozilla.org

Home | Articles | Demos | About

Home » Articles » « Older Article Newer Article »

fxosstub – a minimalist's working example of the design guide rules for Firefox OS

on December 10, 2012 by Pierre Richard in Firefox OS 12 comments

This post is written by Pierre Richard, Principal, Jaxo, Inc.

I know, by experience and practice, the importance of the first hours—or even the first minutes—spent discovering and learning new concepts. During this period, developers take their go/no-go decision, mostly based on the apparent complexity of what they see.

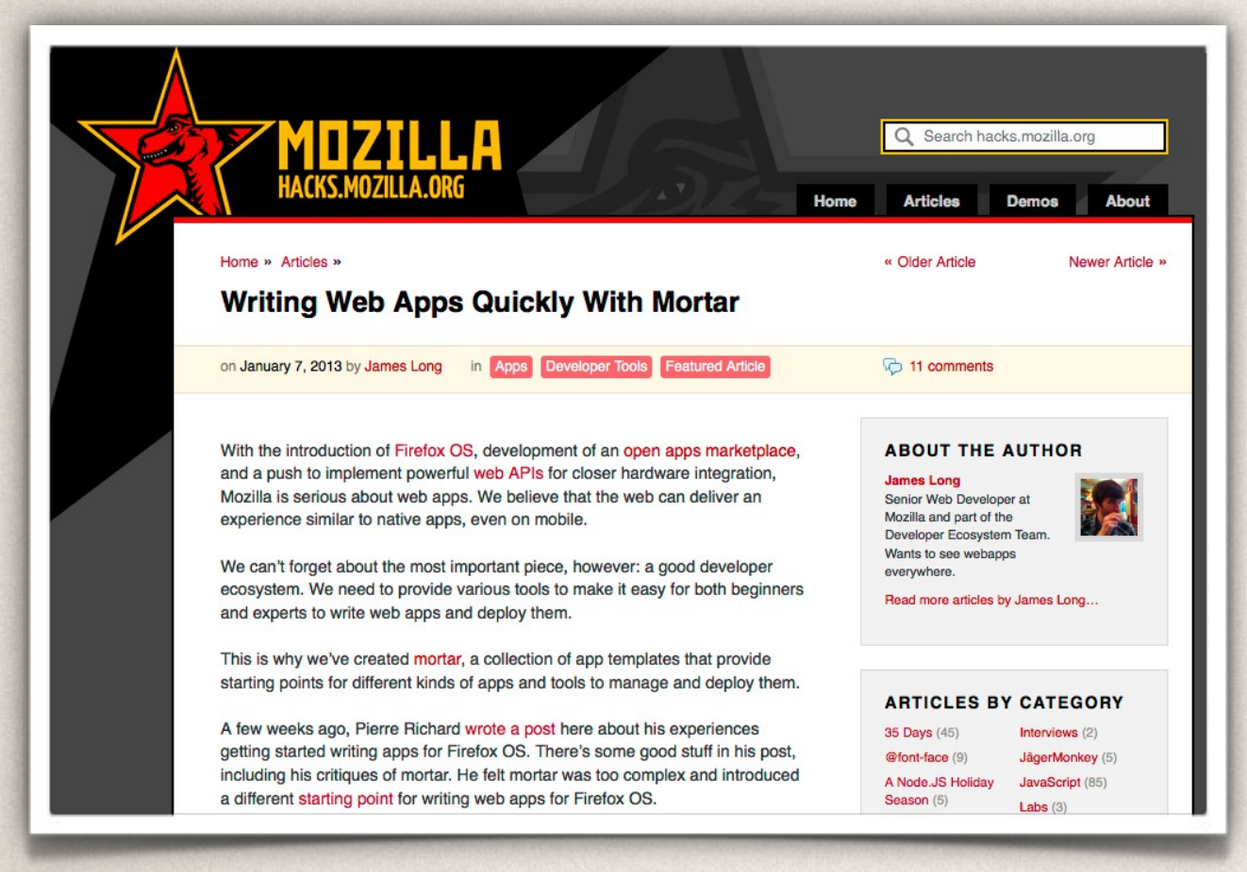
Once you gain expertise in a given domain, you have tendencies to forget the obvious, the basics, which is where the new developers ("newbies") need the most help. Experts are focusing on topics largely irrelevant to newbies wanting to get the essentials.

ABOUT THE AUTHOR

Pierre Richard
As principal and co-founder of Jaxo, Inc., Pierre G. Richard has over 25 years of software engineering and product development experience, having shipped products for IBM and start-ups, generating $10M's in software sales. Pierre is one of the world's leading experts in algorithms and imaging techniques for information processing in barcodes (QR Codes, Data Matrix) and markup languages (XML, XSL), having authored the industry's best software

FXOSSTUB

https://hacks.mozilla.org/2012/12/fxosstub-a-minimalists-working-example-of-the-design-guide-rules-for-firefox-os/



MOZILLA
HACKS.MOZILLA.ORG

Search hacks.mozilla.org

Home | Articles | Demos | About

Home » Articles »

« Older Article Newer Article »

Writing Web Apps Quickly With Mortar

on January 7, 2013 by James Long in Apps Developer Tools Featured Article 11 comments

With the introduction of Firefox OS, development of an open apps marketplace, and a push to implement powerful web APIs for closer hardware integration, Mozilla is serious about web apps. We believe that the web can deliver an experience similar to native apps, even on mobile.

We can't forget about the most important piece, however: a good developer ecosystem. We need to provide various tools to make it easy for both beginners and experts to write web apps and deploy them.

This is why we've created mortar, a collection of app templates that provide starting points for different kinds of apps and tools to manage and deploy them.

A few weeks ago, Pierre Richard wrote a post here about his experiences getting started writing apps for Firefox OS. There's some good stuff in his post, including his critiques of mortar. He felt mortar was too complex and introduced a different starting point for writing web apps for Firefox OS.

ABOUT THE AUTHOR

James Long
Senior Web Developer at Mozilla and part of the Developer Ecosystem Team. Wants to see webapps everywhere.

Read more articles by James Long…

ARTICLES BY CATEGORY

35 Days (45)
@font-face (9)
A Node.JS Holiday Season (5)
Interviews (2)
JägerMonkey (5)
JavaScript (85)
Labs (3)

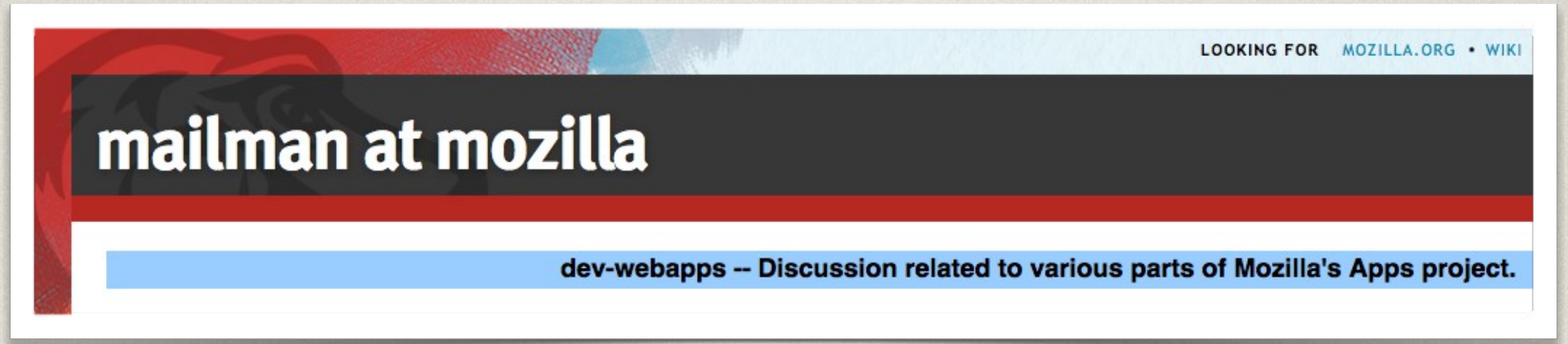
MORTAR

https://hacks.mozilla.org/2013/01/writing-web-apps-quickly-with-mortar/

Getting help

irc://irc.mozilla.org/

#openwebapps



https://lists.mozilla.org/listinfo/dev-webapps

mozilla

Clauber Stipkovic

Comunidade
Mozilla Brasil

@clauberhalic