



# FIREFOX OS

---

概况和开发环境介绍



# 把Open Web带入移动设备的 操作系统





- **Firefox OS**为移动设备建立了一个开放的Web平台。
- **Firefox OS**打破了现存专有平台对产权负担和限制。
- 在Web标准中，增加了和硬件交互的接口，并可以充分发挥出移动设备的硬件性能。



# 开发者的机会

- 使用HTML5和新的Web API，可以实现许多以前不容易实现的效果和功能。
- 开发人员将不再需要针对特定平台的原生API进行学习、开发和发展，将会有更广阔的适应空间。



# OEM和运营商的好处

- OEM和运营商能够将内容和服务整合到设备中，不必再受限于操作系统。
- 能够自定义的用户体验，管理应用程序分配和保留客户的关注度，忠诚度以及计费关系。



# 消费者的自由度

- 能够方便地访问和下载自己的内容。
- 有更多选择的自由。
- 多种设备只需要一次付费购买。





# Firefox OS系统结构



Gaia

Home	Dialer	IM	Browser	Camera	Clock	Notes
Contacts	SMS	Calendar	Media Player	Alarm	FM Radio	Photo Album

Gecko

Browser Engine	Document Parser	Frame Constructor	JavaScript Interpreter	NSS/PSM	XPCOM
Rendering Engine	Style System	Content Model	Necko	Widget	XPCConnect

Libraries

Surface Manager	Media Framework	SQLite	UI	Libc
OpenGL IES	Audio Manager	FreeType	SSL	...

hal

HAL

Graphics	Audio	Camera	Bluetooth	GPS	Radio	Wifi
----------	-------	--------	-----------	-----	-------	------

Linux Kernel

Display Driver	Camera Driver	Bluetooth Driver	GPS Driver	Binder Driver	Shared Memory Driver
USB Driver	Keypad Driver	Wifi Driver	Audio Driver	Radio Driver	Power Management

Gonk



# AlarmAPI

partial interface Navigator

{

    readonly attribute AlarmsManager mozAlarms;

};

interface AlarmsManager

{

    DOMRequest getAll();

    DOMRequest add(in jsval date, in DOMString  
        respectTimezone, [optional] in jsval data);

    void remove(in unsigned long id);

};

<https://wiki.mozilla.org/WebAPI/AlarmAPI>



# AlarmAPI

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

```
request.onsuccess = function (e) {  
    alarmId1 = e.target.result;  
};  
request.onerror = function (e) {  
    alert(e.target.error.name);  
};
```

<https://wiki.mozilla.org/WebAPI/AlarmAPI>



# AlarmAPI

## AlarmsManager.js

```
add: function add(aDate, aRespectTimezone, aData) {  
  this._cpmm.sendAsyncMessage (  
    "AlarmsManager:Add",  
    { requestId: this.getRequestId(request),  
      date: aDate,  
      ignoreTimezone: isIgnore_Timezone,  
      data: aData,  pageURL: this._pageURL,  
      manifestURL: this._manifestURL  
    });  
}
```

[http://mxr.mozilla.org/mozilla-central/  
source/dom/alarm/](http://mxr.mozilla.org/mozilla-central/source/dom/alarm/)



# AlarmAPI

AlarmService.jsm

```
receiveMessage: function receiveMessage(aMessage)
{
    switch (aMessage.name) {
        case "AlarmsManager:Add":
            ...
    }
}

let alarmHalService = this._alarmHalService =
Cc["@mozilla.org/alarmHalService;1"] .getService(Ci.
nsIAlarmHalService);
```

[http://mxr.mozilla.org/mozilla-central/  
source/dom/alarm/](http://mxr.mozilla.org/mozilla-central/source/dom/alarm/)



# AlarmAPI

AlarmHalService.cpp

```
bool status = hal::SetAlarm(aSeconds,  
aNanoseconds);
```

GonkHal.cpp

```
const int result = ioctl(sAlarmData->mFd, 1005  
ANDROID_ALARM_SET(ANDROID_ALARM_RTC_W  
AKEUP), &ts);
```

<http://mxr.mozilla.org/mozilla-central/source/dom/alarm/AlarmHalService.cpp>  
<http://mxr.mozilla.org/mozilla-central/source/hal/gonk/GonkHal.cpp>



# Web App的开发环境



- QEMU Emulator
- Firefox OS Desktop
- Firefox OS Simulator
- Firefox Nightly
- Firefox OS Device





# QEMU EMULATOR



- 需要对Firefox OS源码进行编译。
- 模拟器运行很慢。
- Firefox OS设备模拟器在PC上运行，完全模拟手机设备。
- 在模拟器上调试开发的过程和实际设备上的一致。
- Firefox OS的部分测试代码是在Emulator上运行的，例如：  
测试打电话的功能，测试程序会运行两个Emulator，控制其中一个Emulator拨打电话，如果另外一个Emulator有来电并能接通，就说明程序逻辑OK。



1. Download Source Code

2. Config Environment

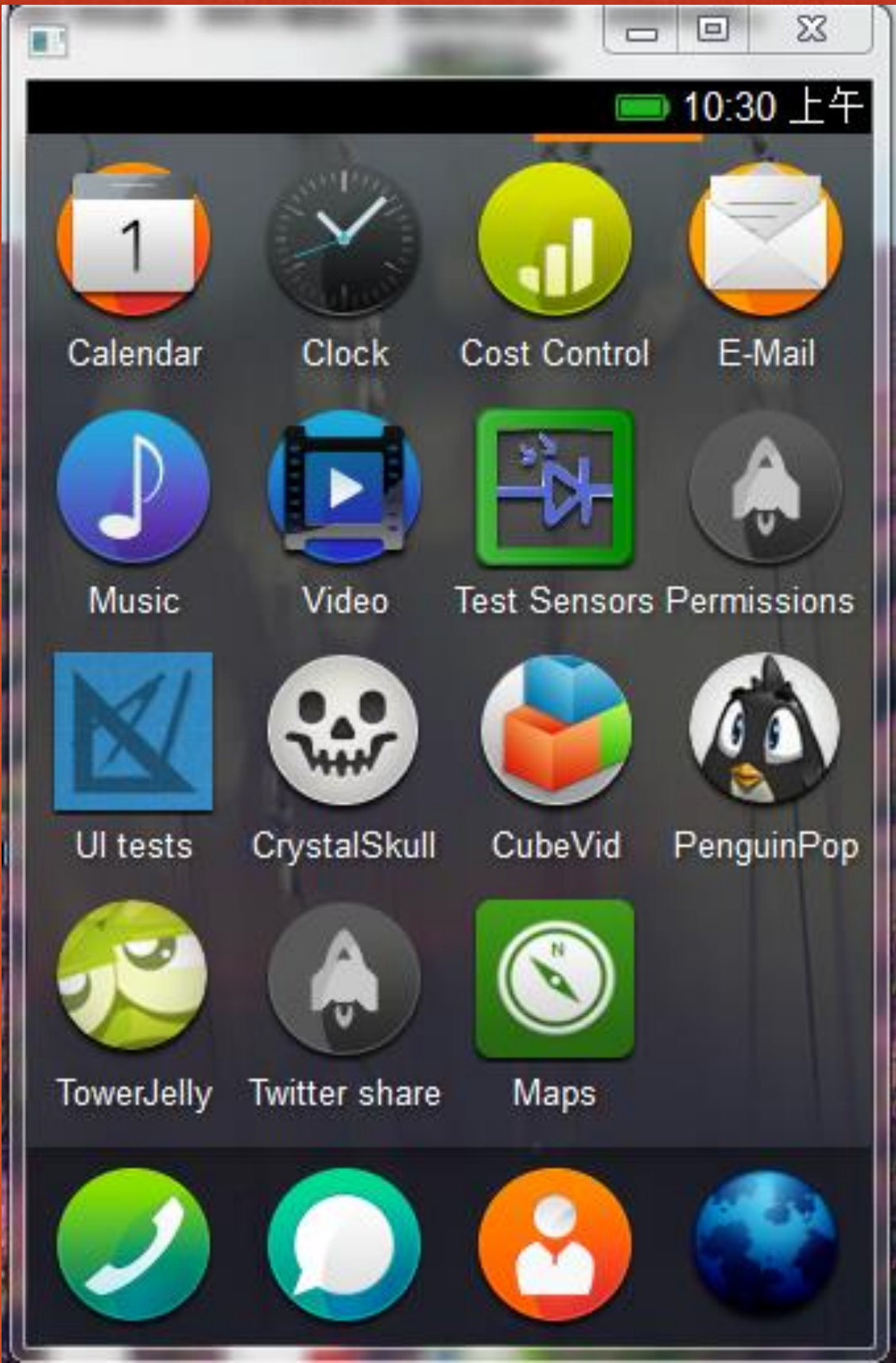
3. Bulid Emulator

4. Run



```
test@ThinkPad:/home$ git clone git://github.com/mozilla-b2g/B2G.git
test@ThinkPad:/home$
test@ThinkPad:/home$
test@ThinkPad:/home$ cd B2G
test@ThinkPad:/home/B2G$ ./config.sh emulator
test@ThinkPad:/home/B2G $
test@ThinkPad:/home/B2G $
test@ThinkPad:/home/B2G $ ./build.sh
test@ThinkPad:/home/B2G $
test@ThinkPad:/home/B2G $
test@ThinkPad:/home/B2G $ ./run-emulator.sh
```





# FIREFOX DESKTOP



- 不需要进行编译。
- 运行在桌面上的Firefox OS模拟器，本质上是一个改造过的浏览器，使之具备Gaia能够正确运行的基本环境，
- Firefox OS Desktop和QEMU Emulator有本质区别，它没有虚拟设备支持，Device API也不存在，例如：电话应用可以打开但是不能打电话。
- 在Firefox OS Desktop上，用户可以体验Firefox OS的运行效果，开发者也可以用来做不需要DeviceAPI支持的Feature开发和调试。
- Firefox OS Desktop虽然是改造过的火狐浏览器，但是火狐自带的开发者工具以及第三方开发工具（Firebug等）没法使用。



1. Download Desktop Builds

2. Checkout Gaia

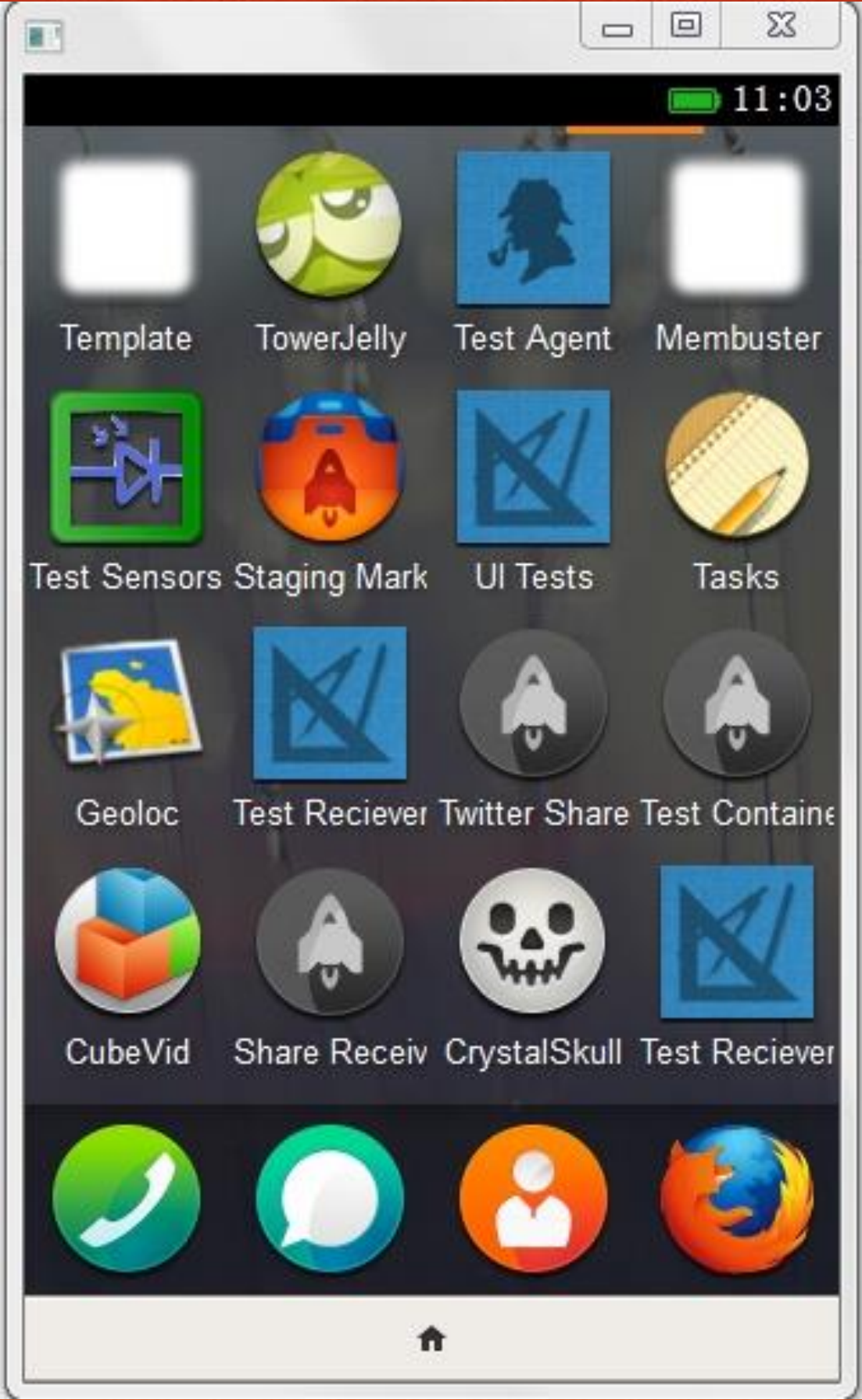
3. Set Environment

4. Run



- 1、 <http://ftp.mozilla.org/pub/mozilla.org/b2g/nightly/latest-mozilla-central/>
- 2、 `git clone git://github.com/mozilla-b2g/gaia`
- 3、 # In gaia directory  
    `make profile`
- 4、 `/path/to/b2g -profile ~/gaia/profile`



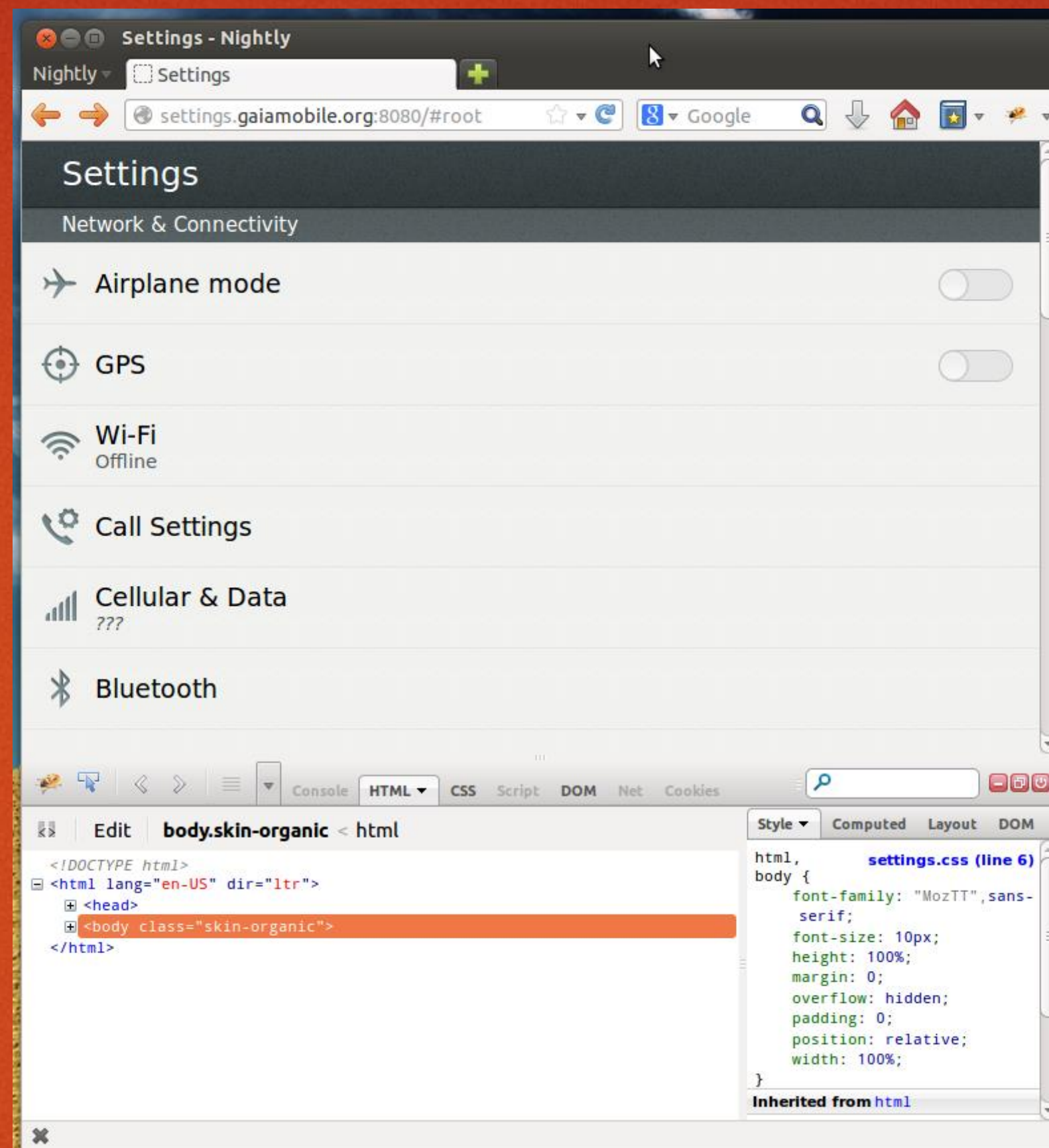


# FIREFOX SIMULATOR



- 不需要进行编译。
- 是运行在Firefox浏览器上的一个Addon。
- 和Firefox OS Desktop类似，它没有虚拟设备支持。
- 用户可以安装Web App，体验Firefox OS的运行效果，开发者也可以用来做不需要DeviceAPI支持的Feature开发和调试，比如：应用窗口管理等。
- 火狐自带的开发者工具以及第三方开发工具（Firebug等）没法使用。





# FIREFOX NIGHTLY



- 不需要进行编译。
- **Firefox Nightly**是一个完整的火狐浏览器。
- 开发者工具以及第三方开发工具都能正常使用，用来开发**Gaia**应用十分方便，对于**HTML5**应用开发者来说，**Nightly**是一个十分理想的开发环境。
- 但是**Gaia**的一些系统应用很多是不能使用的，例如**System**和**Homescreen**只能看到一个黑屏和一个白屏。



1. Download Firefox Nightly
2. Checkout Gaia
3. Set Environment
4. Run



- 1、 <http://ftp.mozilla.org/pub/mozilla.org/firefox/nightly/latest-mozilla-central/>
- 2、 `git clone git://github.com/mozilla-b2g/gaia`
- 3、 # In gaia directory  
    `DEBUG=1 make profile`
- 4、 `/path/to/firefox -profile ~/gaia/profile`





# FIREFOX DEVICE

---



- 需要对Firefox OS源码进行编译。
- Firefox OS 实际运行环境。
- 所有代码最终需要在设备上运行，验证程序是否运行正确。
- Gonk, Gecko 以及部分Gaia程序（与特定硬件相关的API）需要运行在设备上进行调试。



# 参考资料

- <https://wiki.mozilla.org/B2G>
- <https://wiki.mozilla.org/Gecko>
- <https://wiki.mozilla.org/Gaia>
- [http://en.wikipedia.org/wiki/Boot\\_to\\_Gecko](http://en.wikipedia.org/wiki/Boot_to_Gecko)
- [https://developer.mozilla.org/en-US/docs/Mozilla/Firefox\\_OS](https://developer.mozilla.org/en-US/docs/Mozilla/Firefox_OS)
- <https://developer.mozilla.org/en-US/docs/Apps/>
- <http://mozilla.com.cn/>