

Applications mobiles et domotique

Projet - Serveur REST Node.js

Les codes sources indispensables pour la mise en route des différentes parties de ce projet sont disponibles sur GitHub à l'adresse suivante : <http://github.com/yjakiew/projet-ei5>

Dans la suite de ce document, nous ferons l'hypothèse que l'utilisateur a téléchargé et décompressé l'archive du projet dans le dossier de son choix.

(Télécharger l'archive du projet grâce au bouton "Download ZIP" dans la colonne de droite sur la page du projet)

branch: master projet-ei5

Server + Client

BaptisteGauduchon authored 6 days ago latest commit 1f11530e46

Arduino	Arduino : now send ip AND id when registering	13 days ago
client-iPhone	jQueryClient - Deleted tests files	a month ago
server-NodeJS	Server + Client	6 days ago
LICENSE	Initial commit	5 months ago
README.md	Update readme	3 months ago

README.md

Node.JS REST server - Arduino

SSH clone URL
git@github.com:yjac

You can clone with HTTPS, SSH, or Subversion.

Clone in Desktop

Download ZIP

1 - Arduino

Pré-réquis

L'IDE propriétaire est nécessaire pour ouvrir, éditer et téléverser le programme vers l'arduino. il peut être téléchargé à l'adresses suivante : <http://arduino.cc/en/Main/Software>

Utilisation

Le programme Arduino se trouve dans le dossier suivant :

Arduino/arduino/arduino.ino

Il suffit alors d'ouvrir l'IDE Arduino pour charger le programme et le téléverser vers l'arduino de votre choix.

Note :

- Certain paramètres nécessitent une modification manuelle avant de flasher une arduino :
 - L'adresse **IP** (attention au conflit)
 - La **description** (le nom de la carte)
 - L'**id** de la carte (peut être la même chose que la description)
- Certains autres paramètres peuvent être modifiés afin de s'adapter à la configuration d'un serveur REST en particulier (pour faire cohabiter plusieurs serveurs / arduinos sur un même réseau)
 - L'**ip** du serveur d'enregistrement
 - Le **port** d'écoute du serveur d'enregistrement

2 - Serveur REST

Le code relatif au serveur REST se trouve dans le répertoire “**server-Node.JS**”.

Pré-réquis :

Avant de pouvoir utiliser le serveur, il est nécessaire d'installer le framework Node.JS. Les exécutables pour les plateformes Windows® et Mac OS® peuvent être trouvés à l'adresse suivante : <http://nodejs.org/download/>.

La procédure d'installation est standard, il suffit de suivre les instructions données par l'installateur.

Il faudra par ailleurs vérifier que la carte réseau est bien configurée pour faire fonctionner le serveur d'enregistrement pour les arduinos. En fonction de son système d'exploitation, l'utilisateur prendra soin de configurer l'adresse IPV4 de sa carte réseau de la manière suivante:

- **IP** : 192.168.2.1
- **Masque** de réseau : 255.255.255.0

Mise en route :

Le serveur se lance en ligne de commande. Voici les étapes à suivre :

1. Pour Windows : ouvrir l'invite de commande spécifique à l'environnement Node.JS, que l'on trouve dans le menu démarrer ainsi que dans le répertoire spécifié lors de l'installation.
2. Pour Mac OS : l'exécutable “node” est ajouté automatiquement au “Path” du shell utilisateur lors de l'installation.
3. Indépendamment de la distribution, il faut se déplacer dans le dossier qui contient le code du serveur REST :

Windows :	<code>dir /chemin/du/bureau/projet-ei5-master/serveur-NodeJS</code>
Mac :	<code>cd /chemin/du/bureau/projet-ei5-master/serveur-NodeJS</code>

4. Une fois dans le bon dossier, on peut lancer le server :

Windows :	<code>node web.js</code>
Mac :	<code>sudo node web.js</code>

5. Si tout a bien fonctionné, les log du serveur doivent ressembler à ceci :

```

server-NodeJS — node — bash — ttys002 — 132x20
Last login: Tue Feb 25 15:55:36 on ttys001
baptistsmacbook:server-NodeJS Phantom$ sudo node web.js
Password:
25 Feb 15:55:59 - [WEB] Server launched on port 8080
25 Feb 15:55:59 - [DAO] Server listening for new Arduinos on 192.168.2.1:100

```

6. En cas d'erreur de configuration de la carte réseau, ou si le port d'écoute du serveur web est déjà utilisé par un autre service sur la machine, vous obtiendrez un message d'erreur similaire à celui-ci (erreur non capturée volontairement pour forcer une vérification de la part de l'utilisateur) :

```
events.js:72
      throw er; // Unhandled 'error' event
      ^
Error: listen EADDRINUSE
    at errnoException (net.js:901:11)
    at Server._listen2 (net.js:1039:14)
    at listen (net.js:1061:10)
    at Server.listen (net.js:1135:5)
    at Function.app.listen
(/Users/Phantom/GitHub/projet-ei5/server-NodeJS/node_modules/e
xpress/lib/application.js:533:24)
    at Object.<anonymous>
(/Users/Phantom/GitHub/projet-ei5/server-NodeJS/web.js:167:8)
    at Module._compile (module.js:456:26)
    at Object.Module._extensions..js (module.js:474:10)
    at Module.load (module.js:356:32)
    at Function.Module._load (module.js:312:12)
```

Ici, "EADDRINUSE" signifie que le port 8080 à l'adresse 192.168.2.1 est déjà utilisé par un autre service, tel qu'apache ou un autre serveur REST.

Utilisation :

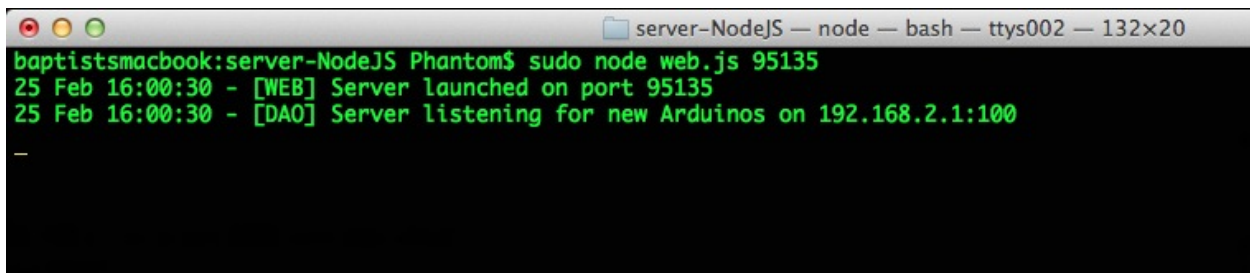
Une fois le serveur lancé, il est disponible à l'adresse configurée dans le code, à savoir :

```
http://ipDeLaMachine:8080/rest/arduinos
```

Notes :

Certains paramètres peuvent être modifiés lors du lancement du serveur ainsi que dans le code :

- le **port** peut être changé en passant le nouveau numéro souhaité en paramètre lors du lancement du serveur :

A terminal window titled "server-NodeJS — node — bash — ttys002 — 132x20" shows the command "sudo node web.js 95135" being executed. The output shows the server launching on port 95135 and listening for new Arduinos on 192.168.2.1:100.

```
baptistsmacbook:server-NodeJS Phantom$ sudo node web.js 95135
25 Feb 16:00:30 - [WEB] Server launched on port 95135
25 Feb 16:00:30 - [DAO] Server listening for new Arduinos on 192.168.2.1:100
—
```

- l'**IP** du serveur d'écoute peut lui être modifier dans le code de la couche [DAO] (partie déclaration des variables)

L'adresse IP du serveur web dépend de l'interface avec laquelle on vient se connecter dessus (celle de la carte Wifi si on utilise un routeur Wifi par exemple).

La console contient tous les logs relatifs au fonctionnement du serveur : on peut ainsi voir, pour chacune des couches, les informations transiter entre le clients et l'Arduino.

Client jQuery Mobile

Le code du client web jQuery se trouve dans le repertoire “**server-Node.JS/public**” du server REST.

Utilisation :

L'utilisation du client jQuery mobile est simple : il suffit, dans un navigateur, de taper l'adresse de base du server. Par exemple, en local :

http://localhost:8080/

Le client est alors chargé dans la navigateur par notre serveur web, prêt à l'utilisation.

Note :

Il est nécessaire de modifier la configuration du client lorsqu'on se connecte à un serveur distant. Le client est en effet configuré par défaut pour chercher le serveur REST en local.

Il faut alors se rendre sur la page “Change Settings” afin de rentrer les paramètres adéquats du serveur (ip, port, url de base).

Menu

Server Settings

Change server settings :

Server address (IPv4): 127.0.0.1

Port: 8080

Url path (after ip & port): /rest

Actions :

Clear ✕ Set ✓

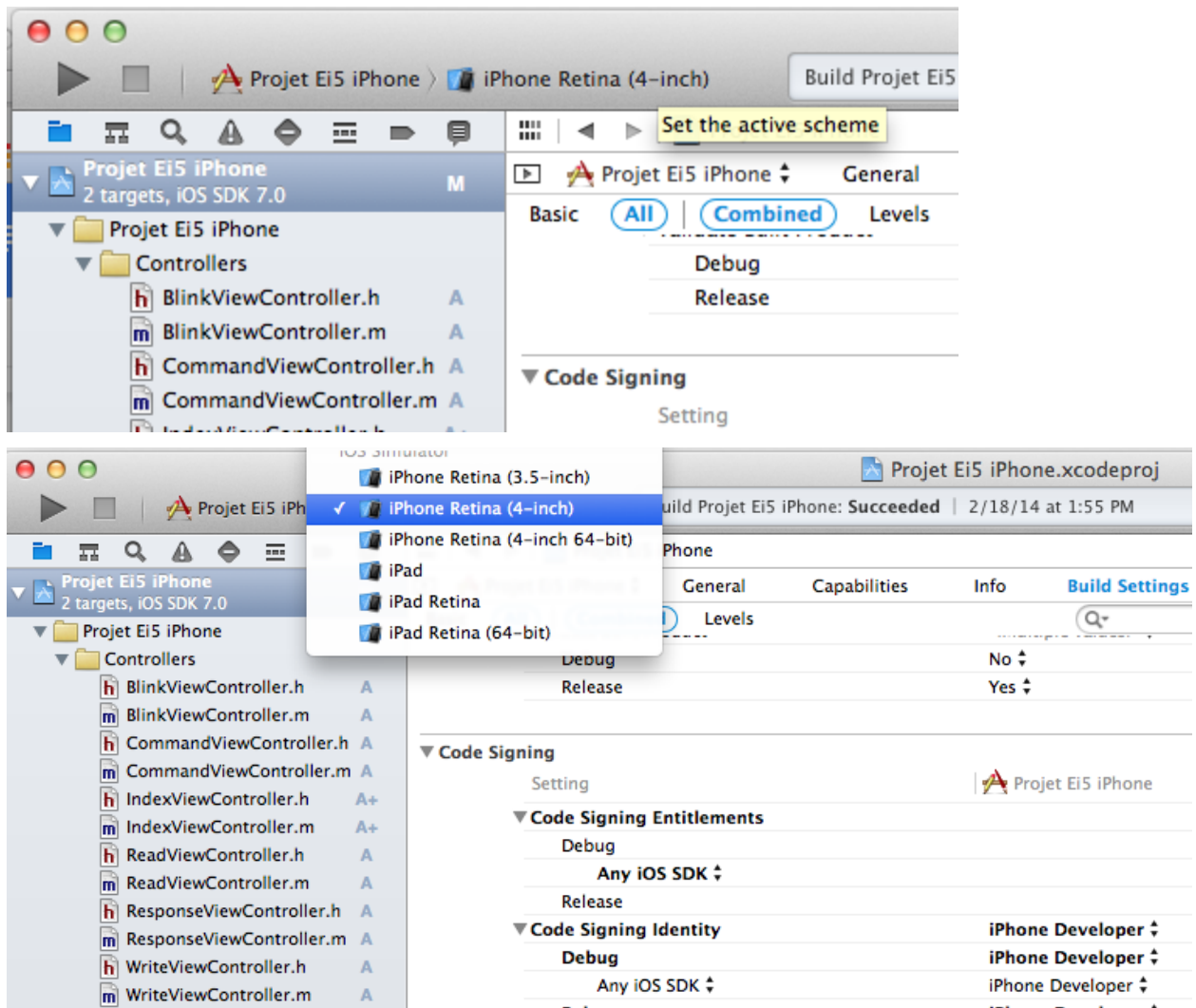
Menu

Client iOS iPhone

L'application a été développée avec l'IDE Xcode 5.0.2 (5A3005) avec le SDK de base iOS 7.0.

Certains éléments du fichier de configuration du projet Xcode ont été modifiés pour pouvoir tester l'application sur un terminal jailbreaké sans avoir besoin de license développeur, malheureusement sans succès. Lors de tests sur un vrai terminal, peut être que les éléments de configuration, situés sous la catégorie "Code Signing", devront être modifiés.

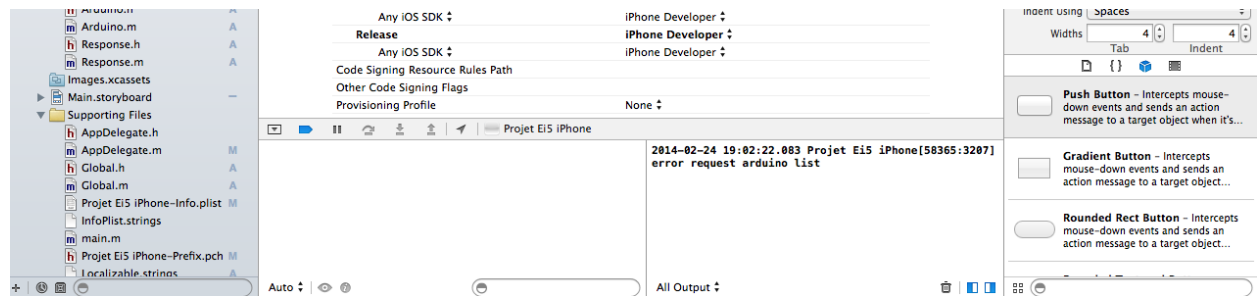
L'application a été conçue et testée pour l'iPhone retina (4-inch). Ainsi pour lancer l'application dans le simulateur de terminal il est nécessaire de le préciser :



Pour lancer le simulateur, utilisez simplement le raccourci Command + R, cela lancera la compilation et le simulateur.

Le simulateur pour la cible iPhone retina, de part la résolution du terminal, est une fenêtre très grande. Selon la résolution de votre écran vous serez peut-être amené à devoir la réduire pour pouvoir afficher l'écran du simulateur entièrement sur le vôtre. Pour cela aller dans le menu "Window > Scale" et sélectionnez 75 ou 50%.

Lors de la simulation, des logs sont parfois affichés au sein de l'IDE et peuvent permettre de mieux comprendre si l'application ne répond pas comme vous le souhaitez :



Le fonctionnement et l'utilisation de l'application est expliqué en détails dans la partie "Client iOS" du rapport de projet disponible à l'adresse :

<http://yjajkiew.github.io/projet-ei5/Rapport.pdf>