webOS Signage

Flight Schedule App Setup Guide

Abstraction

This document provides guide on setting up the environment required to use the Flight Schedule Template app provided by LG.

To use the Flight Schedule Template app, the following two software items are needed to be set up.

- Mongo DB
- Node.JS

This document provides guide only for Windows, but the software listed above has versions for platforms other than Windows.

Related Documents

Guide on using the Flight Schedule App Template
http://developer.lge.com/webOSSignage/develop/signage-web-app-template/fids/

Contact Information

Contact us at signagesupport@lge.com or at the developer site forum. Please feel free to leave any comments, questions, suggestions, or problems you may have.

1 Mongo DB

To install and use Mongo DB, follow the instructions in each section in the order the sections are presented.

1.1 Downloading Mongo DB

Go to http://www.mongodb.org/downloads?_ga=1.33071160.706000627.1414732130#previous to download the 2.6.8 standard version.

Make sure to download the right version for your target system.

- Mongo DB for Windows Server 2008 R2 edition
- Mongo DB for Windows 64-bit
- Mongo DB for Windows 32-bit

NOTE

The latest version of Mongo DB does not support Windows XP. Use Windows released after Windows XP.

1.2 Installing Mongo DB

1. Locate where the downloaded Mongo DB msi file is with your file explorer. The file is most likely to be downloaded in the default Downloads folder.

2. Double click the msi file. A set of screens appears to guide you through the installation process.

1.3 Setting up Mongo DB Environment

Mongo DB requires a data directory in to store the data to present in your app. The default data directory path Mongo DB uses is `\data\db`.

You can opt to use your own data storage directory, but to use the default data directory, create one by following the steps below.

1. Launch a Command Prompt.

2. Create the default data directory path by entering the following command in the Command Prompt window.

```
mkdir \data\db
```

Note

If you opt not to use the default directory, you need to add an option when you launch Mongo DB. For more information, refer to the guide provided in the following section.

1.4 Launching Mongo DB

1. Launch a Command Prompt window.

2. Go to the directory where Mongo DB is installed.

3. Launch Mongo DB by running the `mongod.exe` from the Command Prompt window, using the following command.

```
[root_directory_of_Mongo_DB_installed]>mongod.exe

Example
C:\Program Files\MongoDB\bin>mongod.exe
```

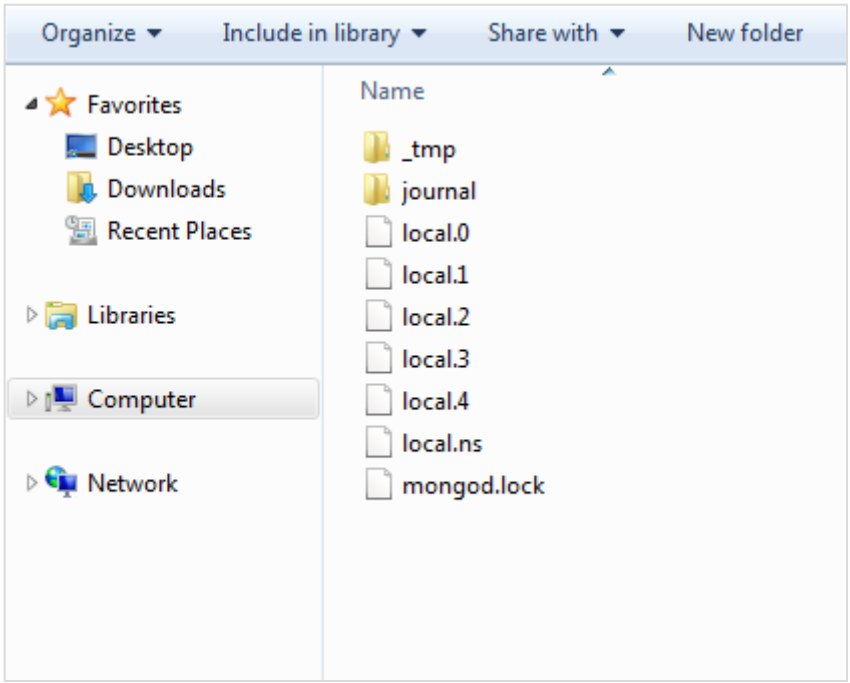
   In case of NOT using the default data directory, add the `--dbpath` option to mongod.exe. For example, to store data in the path `d:\test\mongodb\data`, enter the command as shown below.

```
[root_directory_of_Mongo_DB_installed]> mongod.exe --dbpath
[file_path_to_desired_directory]

Example
c:\Program Files\MongoDB\bin>mongod.exe --dbpath d:\test\mongodb\data
```

Once Mongo DB is launched, you will see something like the following in your Command Prompt window.

```
2015-01-14T14:42:25.026+0530 [initandlisten] allocator: system
2015-01-14T14:42:25.026+0530 [initandlisten] options: { master: true, storage: {
 dbPath: "D:\MongoDbFireWorks\collections" } }
2015-01-14T14:42:25.046+0530 [initandlisten] journal dir=D:\MongoDbFireWorks\col
lections\journal
2015-01-14T14:42:25.046+0530 [initandlisten] recover : no journal files present,
 no recovery needed
2015-01-14T14:42:25.106+0530 [initandlisten] allocating new ns file D:\MongoDbFi
reWorks\collections\local.ns, filling with zeroes...
2015-01-14T14:42:25.306+0530 [FileAllocator] allocating new datafile D:\MongoDbF
ireWorks\collections\local.0, filling with zeroes...
2015-01-14T14:42:25.306+0530 [FileAllocator] creating directory D:\MongoDbFireWo
rks\collections\_tmp
2015-01-14T14:42:25.676+0530 [FileAllocator] done allocating datafile D:\MongoDb
FireWorks\collections\local.0, size: 64MB,  took 0.362 secs
2015-01-14T14:42:25.676+0530 [initandlisten] build index on: local.startup_log p
roperties: { v: 1, key: { _id: 1 }, name: "_id_", ns: "local.startup_log" }
2015-01-14T14:42:25.676+0530 [initandlisten]     added index to empty collection

2015-01-14T14:42:25.686+0530 [initandlisten] command local.$cmd command: create
{ create: "startup_log", size: 10485760, capped: true } ntoreturn:1 keyUpdates:0
 numYields:0  reslen:37 575ms
2015-01-14T14:42:25.686+0530 [initandlisten] ******
2015-01-14T14:42:25.686+0530 [initandlisten] creating replication oplog of size:
 8194MB...
2015-01-14T14:42:25.686+0530 [FileAllocator] allocating new datafile D:\MongoDbF
ireWorks\collections\local.1, filling with zeroes...
2015-01-14T14:42:45.711+0530 [FileAllocator] done allocating datafile D:\MongoDb
FireWorks\collections\local.1, size: 2047MB,  took 20.027 secs
2015-01-14T14:42:45.821+0530 [initandlisten] ExtentManager took 20 seconds to op
en: D:\MongoDbFireWorks\collections\local.1
2015-01-14T14:42:45.831+0530 [FileAllocator] allocating new datafile D:\MongoDbF
ireWorks\collections\local.2, filling with zeroes...
2015-01-14T14:43:09.240+0530 [FileAllocator] done allocating datafile D:\MongoDb
FireWorks\collections\local.2, size: 2047MB,  took 23.383 secs
2015-01-14T14:43:09.380+0530 [initandlisten] ExtentManager took 23 seconds to op
en: D:\MongoDbFireWorks\collections\local.2
2015-01-14T14:43:09.380+0530 [FileAllocator] allocating new datafile D:\MongoDbF
ireWorks\collections\local.3, filling with zeroes...
2015-01-14T14:43:39.001+0530 [FileAllocator] done allocating datafile D:\MongoDb
FireWorks\collections\local.3, size: 2047MB,  took 29.621 secs
2015-01-14T14:43:39.111+0530 [initandlisten] ExtentManager took 29 seconds to op
en: D:\MongoDbFireWorks\collections\local.3
2015-01-14T14:43:39.121+0530 [FileAllocator] allocating new datafile D:\MongoDbF
ireWorks\collections\local.4, filling with zeroes...
2015-01-14T14:44:11.265+0530 [FileAllocator] done allocating datafile D:\MongoDb
FireWorks\collections\local.4, size: 2047MB,  took 32.144 secs
2015-01-14T14:44:11.405+0530 [initandlisten] ExtentManager took 32 seconds to op
en: D:\MongoDbFireWorks\collections\local.4
2015-01-14T14:44:16.036+0530 [initandlisten] ******
2015-01-14T14:44:16.076+0530 [initandlisten] waiting for connections on port 270
17
2015-01-14T14:44:16.116+0530 [initandlisten] connection accepted from 127.0.0.1:
50080 #1 (1 connection now open)
2015-01-14T14:44:16.745+0530 [conn1] end connection 127.0.0.1:50080 (0 connectio
ns now open)
```

4. Check in your data storage directory that folders and files as shown below have been created.



1.5 Checking Running Mongo DB

As explained in the previous section, when you run Mongo DB, you will see the result message in your Command Prompt window and folders and files created in your data directory.

However, if you wish to check if it is running correctly or encounter a problem in running it, try the following instructions.

Note

Mongo DB should be running to check if it is running correctly. Keep the Command Prompt window from which you ran Mongo DB.

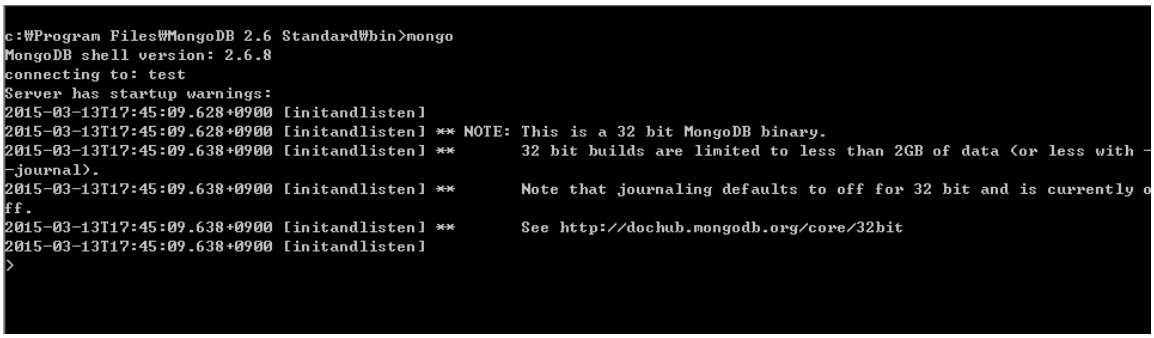
1. Open a new Command Prompt window.

2. Go to the directory where Mongo DB is installed.

3. Enter the command `mongo.exe` in the Command Prompt window.

```
[root_directory_of_Mongo_DB_installed]>mongo.exe

Example
C:\Program Files\MongoDB\bin>mongo.exe
```

Check the messages printed on this Command Prompt window and the one from which you ran Mongo DB.



Note

To find what operations you can perform with `mongo.exe`, enter the `help` command in the Command Prompt window.

2 Node.js

To install and use Node.js server, follow the instructions in each section in the order the sections are presented.

2.1 Downloading Node.js

Go to http://nodejs.org/download/ and download the Node.js installer. Make sure to download the right version for your target system.

2.2 Installing Node.js

1. Locate where the downloaded Node.js installer file is with your file explorer. The file is most likely to be downloaded in the default Downloads folder.

2. Double click the msi or exe file. A set of screens appears to guide you through the installation process.

2.3 Running Server

Running a Node.js server for the Flight Schedule app requires Mongo DB running. Launch Mongo DB before you initiate to run the Node.js server.

1. Open a Command Prompt window.

2. Go to the server directory in your app is located at. Note that `server` directory is provided in the template.

```
e.g. cd c:\webOS Signage Apps\Flight Schedule\server
```

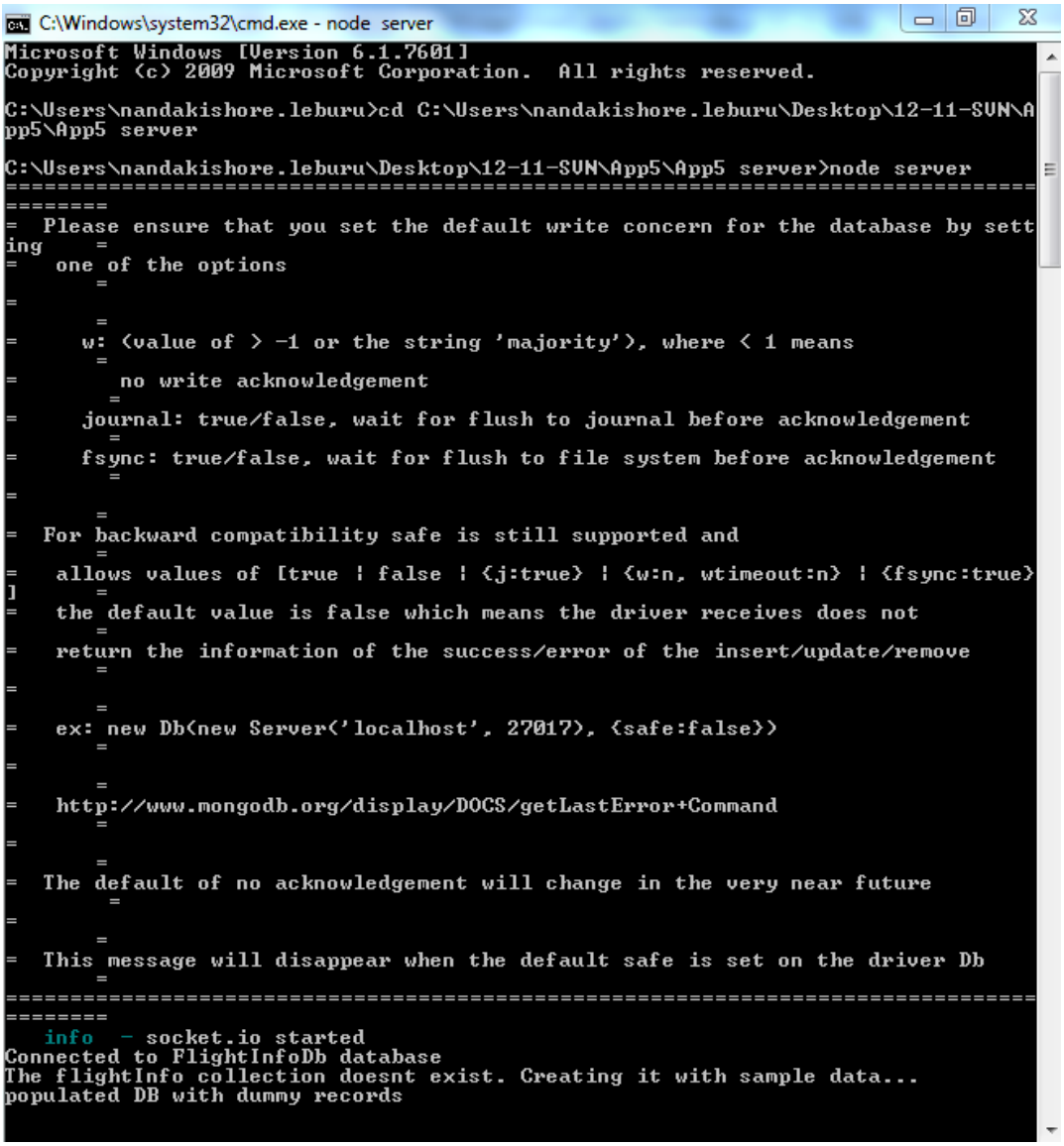
3. Enter the node server command in the Command Prompt window.

```
[App_root_directory\server]>node server

Example
c:\webOS Signage Apps\Flight Schedule\server>node server
```

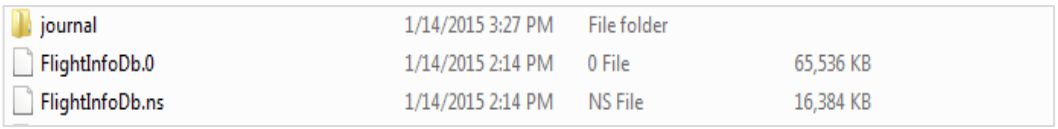
Once the Node.js server is launched a connection is established between your server and Mongo DB, you will see something like the following in your Command Prompt window.

Note

Running a server uses the `server.js` file provided in the template. The `server.js` file contains configurations of port and application to load required modules.



4. Check your data storage directory which you set for Mongo DB. The flight data to display on your app can be seen shown below.



Now everything needed to run the Flight Schedule app is complete. Run the app by launching the index.html file.

IMPORTANT

Before you run your app, check the *3 Running the App* chapter to make modifications in your app.

3 Running the App

When you have set up the environment—Mongo DB and Node.js server—you need to make a few changes in your app.

3.1 Server Side Changes

server.js

To change the port number used in communication between client and server apps, change the port defined in the file `server.js`. The default port number defined in the template is 8989.

flight.js

To change the values of Mongo DB connection port, DB collection name and end points, change the values defined in the `flight.js` file.

3.2 Client Side Changes

Change the default IP values used in the template to your server IP.

1. In the `index.html` change the IP to your server IP.

IMPORTANT

If you opt to use a different port number, make sure to reflect it in the `server.js` file too. The default port number defined is 8989.

```
<script src='http://[your_IP]:[your_port_number]/socket.io/socket.io.js'></script>

Example
<script src='http://10.234.56.789:8989/socket.io/socket.io.js'></script>
```

2. In the `data.json` file, add your server IP address, port number and API method.

```
"ipWithPort" : "http://[your_IP]:[your_port_number]",
"getApiMethod": http://[your_IP]:[your_port_number]/flights

Example
"ipWithPort" : "http://10.234.56.789:8989",
"getApiMethod": "http://10.234.56.789:8989/flights"
```