CALIPSO FOR EARLY ADOPTERS
Something new for developers to test

(1)Dr. Gaurav Kumar Jain

Associate Professor, Department of Computer Science & Information Technology, Regional College For Education Research and Technology, Jaipur, Rajasthan
gaurav.rinkujain.jain@gmail.com

(2) Mr. Ravi Ranjan
Regional College For Education Research and Technology, Jaipur, Rajasthan

(3) Mr. Chetan Kumar Saini
Regional College For Education Research and Technology, Jaipur, Rajasthan

(4)Mr. Mahendra Kumar Sharma
Regional College For Education Research and Technology, Jaipur, Rajasthan

ABSTRACT

In this rapidly growing environment of Information Technology, those developers who can rapidly adopt new technology are required the most. Calipso is a content management system (CMS), based on the NodeJS server and MongoDB to perform incredibly fast and flexible operations. Calipso is a latest project, which is still under development. It has a total new idea than other CMS and hence developed a CMS that didn't suffer from any of the issues that attempting to shoehorn content into a relational database leads to.

KEYWORDS

Content Management System, Node.JS, MongoDB, Relational Database.

1. INTRODUCTION

Content Management System (CMS) is a computer program that allows publishing, editing and modifying content on a web site as well as maintenance from a central interface. Such systems of content management provide procedures to manage workflow in a collaborative environment. These procedures can be manual steps or an automated cascade.

Calipso is a content management system (CMS), based on the NodeJS server. Due to the asynchronous nature of NodeJS, it seemed like a good idea to try to build a CMS made up of modules that could execute asynchronously in a non-blocking way.

The original vision of Calipso was to use NodeJS and MongoDB to create an incredibly fast and flexible CMS that didn't suffer from any of the issues that attempting to shoehorn content into a relational database leads to.

The initial core writing of the project Calipso began in the end of april 2011. Calipso uses a modular approach to delivering the functions you would expect from a CMS. All core features, excluding bootstrapping, theming and forms are provided by modules (and these may well be factored out into modules at some point in the future).

2. MODULES

A module is best though of as a miniature MVC express application that can define its own mongoose (mongodb) models, define the routes it will respond to and match them up to its own controller functions, and then define and render content fragments (unstyled) into blocks in the response. Themes can then take these blocks and assemble them into a working site.

Modules are placed inside the 'modules' folder on your server, in one of three folders:

- **Core** : reserved for core Calipso modules
- **Community** : for downloaded contrib / community modules
- **Site** : for any custom modules for your own site (this can be called anything site is just an example)

The module structure itself is then also quite straight forward and adheres to the following structure inside a module folder:

- **community/template/package.json** : Standard CommonJS Package.json (include all local dependencies).
- **community/template/template.js** : the module file itself.
- **community/template/templates** : any templates for rendering output
- **community/template/templates/template.html** : an EJS template
- **community/template/templates/template.jade** : a Jade template

You can add additional files to include, as well as static folders that allow modules to include additional files (see the richforms example) - apart from the convention above you are free to structure your module however you wish, for example, factoring out a complex model definition out of the module.js and into a model.js (that is then required) is absolutely fine and a good thing in a complex module.

3. IDEAS AND CORE PRINCIPLES

3.1 Core

- The core would do very little, it would provide enough to get a basic server running.
- Everything that can and should be asynchronous must be asynchronous.
- The core will focus on providing the basic framework within which modules (e.g. essentially mini MVC applications) can be run and themed.
- It should be multi-lingual from core outwards.
- Environment configuration will be abstracted out to enable use on large and complex projects.
- A Command Line configuration tool will be created to enable integration with Continuous Integration platforms.

3.2 Storage

- MongoDB will be used as much as possible, the initial versions therefore will not be storage agnostic.
- Apart from configuration, everything should go into MongoDB by default (including sessions - to enable correct operation in a clustered configuration).

3.3 Modules

- Modules can respond to a request, but they will do so in parallel and fully asynchronously.
- Where there are dependencies, these are described within the dependent module (e.g. the Content module contains the fact that it depends on the Content Types module), and managed via events.
- Modules will install, initialise and register themselves.
- Modules expose their own functions and helpers to views.

3.4 Themes

- Themes can be switched at run time.
- At all times the modules expose UI elements (e.g. structure), this structure can be later modified within the theming layer.

4.TRANSLATION LIBRARIES

Calipso now has the bare bones of a translation library, that enables a simple (Drupal like) translation function to enable translation of static content into other languages.

The library works by exposing a function: t('Hello World') in views, and via the request object: req.t('Hello World') within module code. It will look up this english string in a language file, where the language to translate to is determined in the following priority order:

1. The language setting passed in the URL
2. The users language setting if they are logged in
3. The language setting specified in the configuration

There are only one or two translations done at the moment to Spanish (es), Russian (ru) and Japanese (ja) (via the link above as an example), and all are done via Google Translate as I don't speak anything but basic Spanish! So apologies if Google offends anyone.

If you create an account and specify the language, you can see that it will change most of the 'static' text (the admin screens that you can't yet see on this site but you can if you download Calipso) are also translated).

The ability for content to also be created in a specific language will be added as a core feature (hopefully sometime soon!)

5. INSTALLATION

5.1 Pre-Requisites

First things first, you need to get your basic environment right, the pre-requisites are:

- NodeJS
- NPM
- MongoDB
- Libraries: libexpat1-dev and libbsd-dev (need to confirm). Without these you will struggle to install node-expat and bcrypt.
- A git client (to access the source code on Github).

Ok - good to go?

5.2 Grab the source...

Go to where you want to run calipso from (e.g. /var/www).

```
$ git clone
git://github.com/cliftonc/calips
o.git

Initialized empty Git repository
in /var/www/tmp/calipso/.git/
remote: Counting objects: 724,
done.
remote: Compressing objects:
100% (440/440), done.
remote: Total 724 (delta 298),
reused 520 (delta 216)
Receiving objects: 100%
(724/724), 640.07 KiB | 476
KiB/s, done.
Resolving deltas: 100%
(298/298), done.

$ cd calipso/

$ ls
```

```
app-cluster.js  conf       lib
    media     package.json
 README   test     utils
app.js          docs.html
 Makefile  modules  pids
 support  themes
```

5.3 Ok, now grab all the dependencies via NPM

Calipso depends on a number of other libraries that are available through NPM (the Node Package Manager). If all of these dependencies aren't installed, you will get lots of errors when attempting to run Calipso

```
$ npm install
npm info it worked if it ends
with ok
npm info using npm@0.3.15
npm info using node@v0.4.2
npm info link
/var/www/tmp/calipso

...
<>
...

npm ok
```

5.4 Make sure MongoDB is running...

On Ubuntu, if you have installed mongodb (sudo apt-get install mongodb), you can make sure it is running by:

```
$ status mongodb
mongodb start/running, process
2588
```

If mongo is not running and won't start (sudo start mongodb), some times if it doesn't shut down gracefully you can get a lock left open, and you need to delete the lock file (sudo rm /var/lib/mongod/mongodb.lock) before starting it.

5.5 Ok - lets run it!

Now the first test, try to run it!

```
$ node app

            _ _

  ___  __ _| (_)_ __  ___  ___

 / __|/ _` | | | '_ \/ __|/ _ \

| (__| (_| | | | |_) \__ \ (_) |
 \___|\__,_|_|_| .__/|___/\___/
               |_|


Logging enabled: Console
Calipso server listening on
port:  3000
```

Now, if you browse to:

http://localhost:3000/" \t "_new

You should see the home screen, along with a message that the site has been configured with a default Administrative user and password. You should login with this, and then change it immediately (you can do this from the Admin users profile page).

5.6 What next?

Ok - you're now up and running, you should be able to login, and once logged in begin to navigate the content creation and administration screens.

5.7 Update: Install Script

There is now an install script that you can try, simply run:

```
$ ./bin/install.sh
```

This should install the dependencies via NPM, and do a few quick checks to see if you are up and running.

6. CONCLUSION

Calipso is a really advance Content Management System (CMS), which uses one of the most powerful scripting language 'Node.JS'. Due to the asynchronous nature of NodeJS, it seemed like a good idea to try to build a CMS made up of modules that could execute asynchronously in a non-blocking way.

Calipso's full source code is available on

https://github.com/cliftonc/calipso

This is the start of its journey, Calipso is in the early days, so any developer, co-contributor, or feedback is always welcome!

ACKNOWLEDGEMENTS

We would like to thank everyone, especially developers, early adopters, I.T. enthusiast and family members who provided support and followed up with us.

REFERENCES

[1.] http://calip.so/

[2.] http://calip.so/section/blog

[3.] http://calip.so/section/github

[4.] http://calip.so/section/guide

[5.] http://calip.so/section/quickstart

[6.] http://calip.so/an-introduction-to-calipso.html

[7.] https://github.com/cliftonc/calipso/wiki/Guide-Index

[8.] http://calip.so/coding-standards.html

[9.] http://calip.so/coding-standards.html

[10.] http://calip.so/benchmarking.html