

# Mojito

Node.js MVC Framework



# Mojito

- Mojit = Module + Widget.
- Run Javascript on client and server side using affinity.
- Mojito is a multi-device MVC framework.



# History

- Initiated in 2011
  - Intended to create highly reusable modular codebase.
  - Single codebase for multiple devices.
- Official launched on April 2, 2012 at JS Conf.
- Current stable version 0.7.1
- Yahoo! Sites which use Mojit
  - Yahoo! Mail.
  - Yahoo! News search.
  - Yahoo! Small Business.
  - Yahoo! Fantasy Football.



# Design and Develop

MAIN PAGE MOJIT

HEADER MOJIT

MAIN PAGE CONTENT

SIDE BAR MOJIT

FOOTER MOJIT

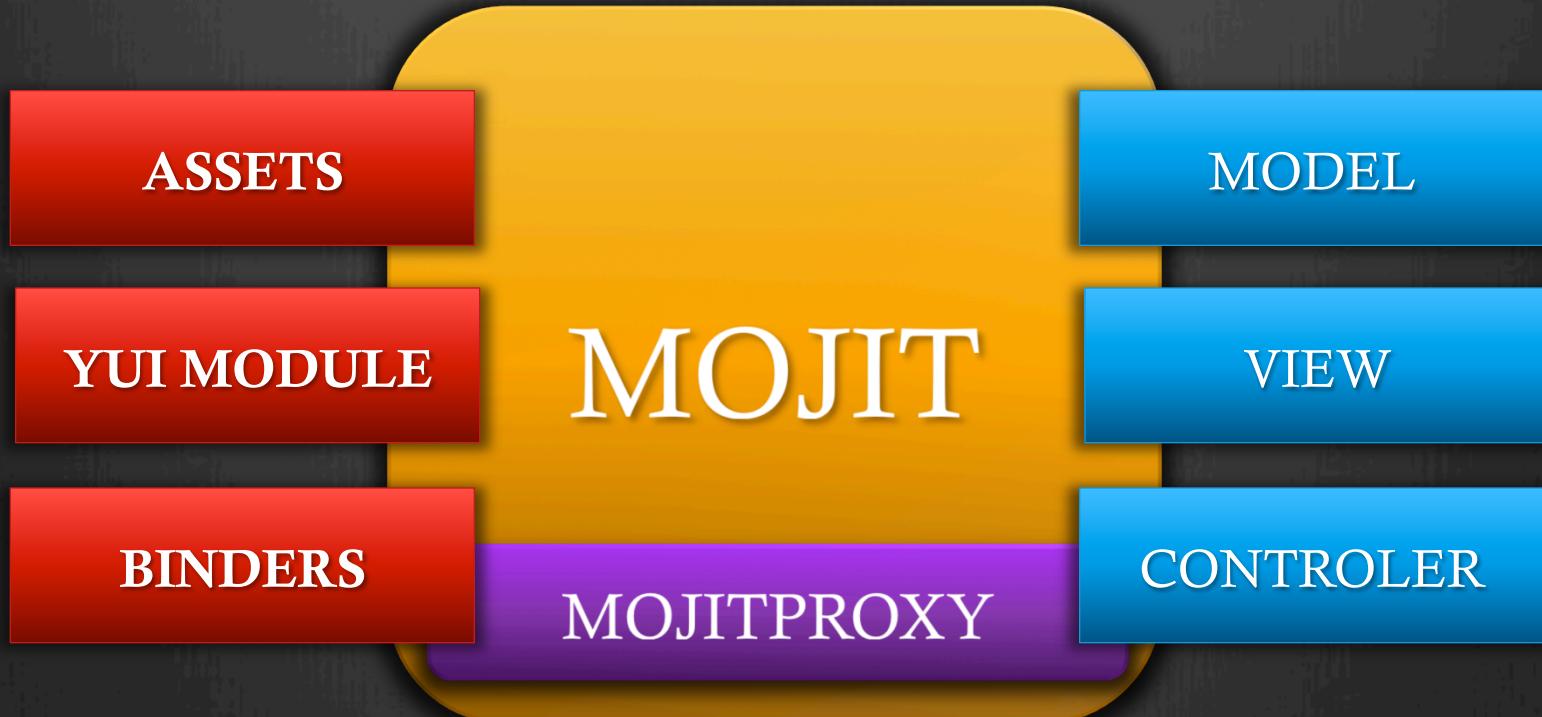


# Architecture

- application.json
- routes.json
- index.js
- MOJITS
  - HEADER
  - FOOTER
  - SIDEBAR
  - NAV
  - ABOUTUS\_PAGE
  - MAIN\_PAGE
    - MODELS
    - VIEWS
  - controler.server.js / controler.client.js /controler.common.js
  - TESTS



# Client & Server



# Controllers

Pass values to **views** using Action Context

```
ac.done({'menu' : 'index'})
```

Passing parameters to **binders** using the Data addon

```
ac.data.set('menu', "index");
```

The Data addon also allows you to share page-level data

```
ac.pageData.set and ac.pageData.get
```

Currently the only way to do this is to pass data to the children  
in either the children config or parameters.



# Where should it run?

controller.`server.js`  
controller.`common.js`  
controller.`client.js`

# Views and Binders

Binder can access data sent by the controller using MojitProxy

```
mojitProxy.data.get('menu');
```

Views can access data using Handler bar expression

```
{{menu}}
```

Binders can call the controller using refreshView or invoke

```
mp.refreshView(); OR mp.invoke();
```



# Context Configuration

```
environment:development  
environment:production  
environment:dev  
environment:test  
environment:stage  
environment:prod  
device:android  
device:blackberry  
device:iemobile  
device:iphone  
device:ipad  
device:kindle  
device:opera-mini  
device:palm  
lang:{BCP 47 language tag}  
runtime:client  
runtime:server
```

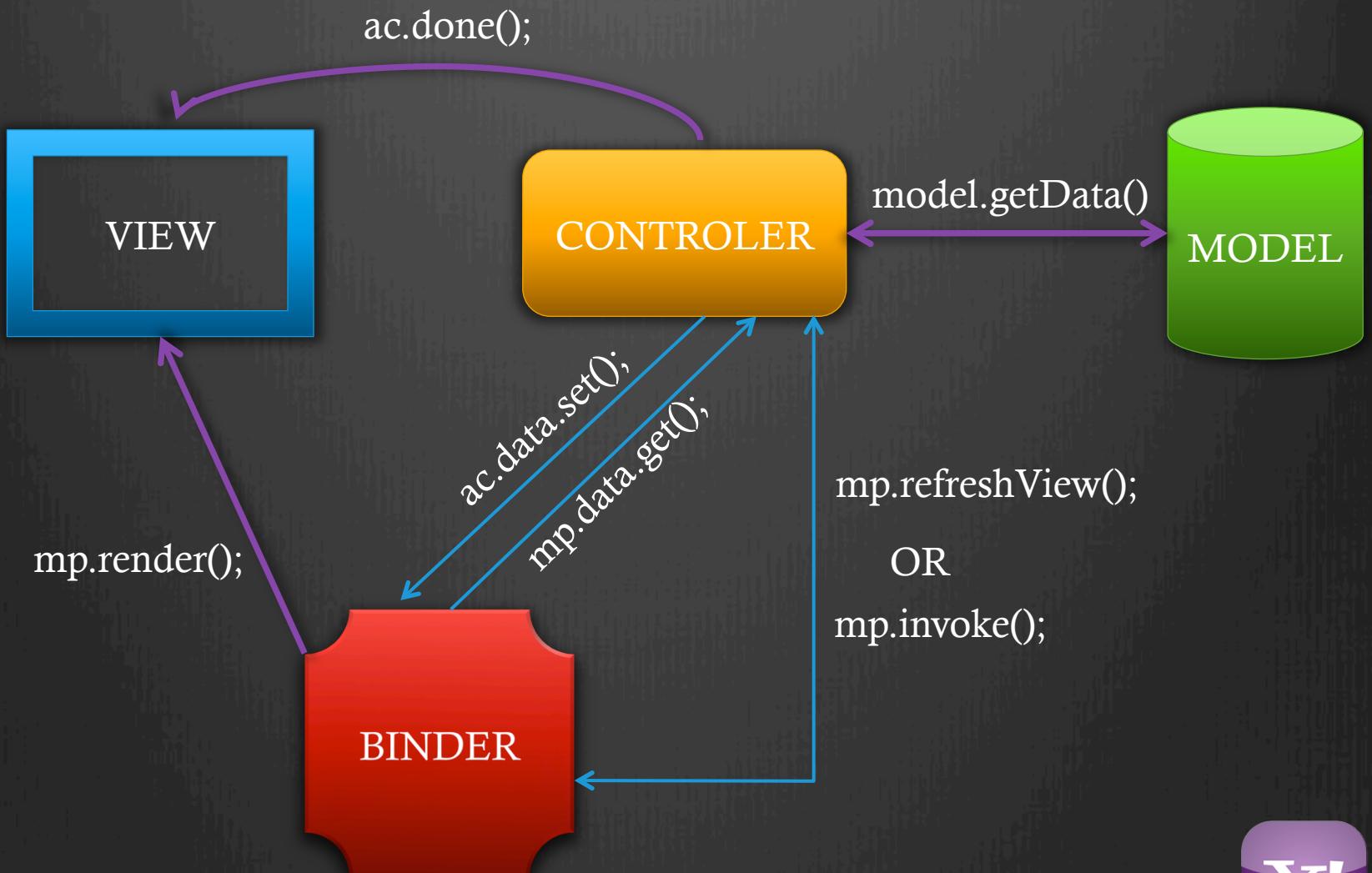
# Models

- Models are pretty conventional. They handle all the REST API calls.
- Mojito provides YQL module which are useful for fetching data.

```
Y.YQL("select * from weather.forecast where woeid=2295420",
    function(data) { callback(null, data.query);});
```



# It all fits together now!



# Get Started

```
$ npm install mojito-cli -g
```

```
$ mojito create app myapp
```

```
$ cd myapp
```

```
$ mojito create mojit header
```

Documentation @ [tinyurl.com/mojitomvc](http://tinyurl.com/mojitomvc)

API Guide @ [tinyurl.com/mojitoapi](http://tinyurl.com/mojitoapi)



# That's all folks!

cavinash@yahoo-inc.com

Ping me @ avin\_fe@yahoo.com

