배우기
# meteor

#meteor

1:

Meteor JavaScript .

(Android / iOS /) .

:

-
- Meteor API
-
-

0.4.0	2012-08-30
0.5.0	2013-10-17
0.6.0	2013-04-04
0.7.0	2013-12-20
0.8.0	2014-04-21
0.9.0	2014-08-26
0.9.1	2014-09-04
0.9.2	2014-09-15
0.9.3	2014-09-25
0.9.4	2014-10-13
1.0.1	2014-12-09
1.0.2	2014-12-19
1.0.3.1	2014-12-09
1.1.0	2015-03-31
1.2.0	2015-09-21
1.3.0	2016-03-27
1.4.0	2016-07-25
1.5.0	2017-05-30

Examples

OS X Linux

Meteor :

```
$ curl https://install.meteor.com/ | sh
```

Windows

Meteor .

Meteor .

```
$ meteor create myapp
```

▪

:

```
$ cd myapp
$ meteor npm install
$ meteor
```

: Meteor http : // localhost : 3000 /

http : // localhost : 3000 Meteor .

- [Meteor Guide] Meteor .
- Meteor Packages at atmosphere - .

Meteor . . Meteor ( ) .

```
meteor create --example <app name>
```

`todos` :

```
meteor create --example todos
```

.

```
meteor create --list
```

Meteor atmospherejs.com .

.

```
meteor add [package-author-name:package-name]
```

:

```
meteor add kadira:flow-router
```

.

```
meteor remove kadira:flow-router
```

.

```
meteor list
```

`./meteor/packages` `./meteor/packages` . .

(:) `packages` .

1.3 Meteor **npm** .

`meteor npm` `meteor npm` `npm` `meteor npm` npm .

. .

```
METEOR_DEBUG_BUILD=1        (logs progress)
METEOR_PROFILE=<n>          (logs time spent)
METEOR_DEBUG_SPRINGBOARD=1  (?)
METEOR_DEBUG_SQL=1          (logs SQLITE calls)
METEOR_PROGRESS_DEBUG=1     (? looks like it might be useful, but seems confusing)
```

`<n>` ms. .

Linux / OSX

```
export METEOR_DEBUG_BUILD=1
export METEOR_PROFILE=100
meteor
```

Windows

```
set METEOR_DEBUG_BUILD=1
set METEOR_PROFILE=100
meteor
```

Meteor Tool & Meteor Project

Meteor Meteor .

```
meteor --version
```

() .

```
meteor show METEOR
```

Meteor .

```
meteor --version
```

`.meteor/release` .

```
cat .meteor/release
```

Meteor , `.meteor/versions` :

```
cat .meteor/versions
```

Meteor Meteor `Meteor.release` .

```
Meteor.release
```

Meteor Tool .

Meteor Meteor .

```
meteor update
```

Meteor Meteor .

```
meteor update --release <release>
```

.

```
meteor update --packages-only
```

meteor update (:

```
meteor update [packageName packageName2 ...]
```

Meteor Cordova . (Apple App Store, Google Play)

1. Meteor :

```
meteor add-platform android
meteor add-platform ios # Only available with Mac OS
```

2. SDK / (iOS , OS).

3. ( ).

```
meteor run android # You may need to configure a default Android emulator first
```

iOS (Mac OS) :

```
meteor run ios # This will auto start an iOS simulator
```

4. App .

```
meteor build <output_folder> --server <url_app_should_connect_to>
```

android / ios .

- android release-unsigned.apk .
- ios Xcode .

Meteor Mobile Apps .
: Meteor Guide> Build> Mobile

: https://riptutorial.com/ko/meteor/topic/439/-

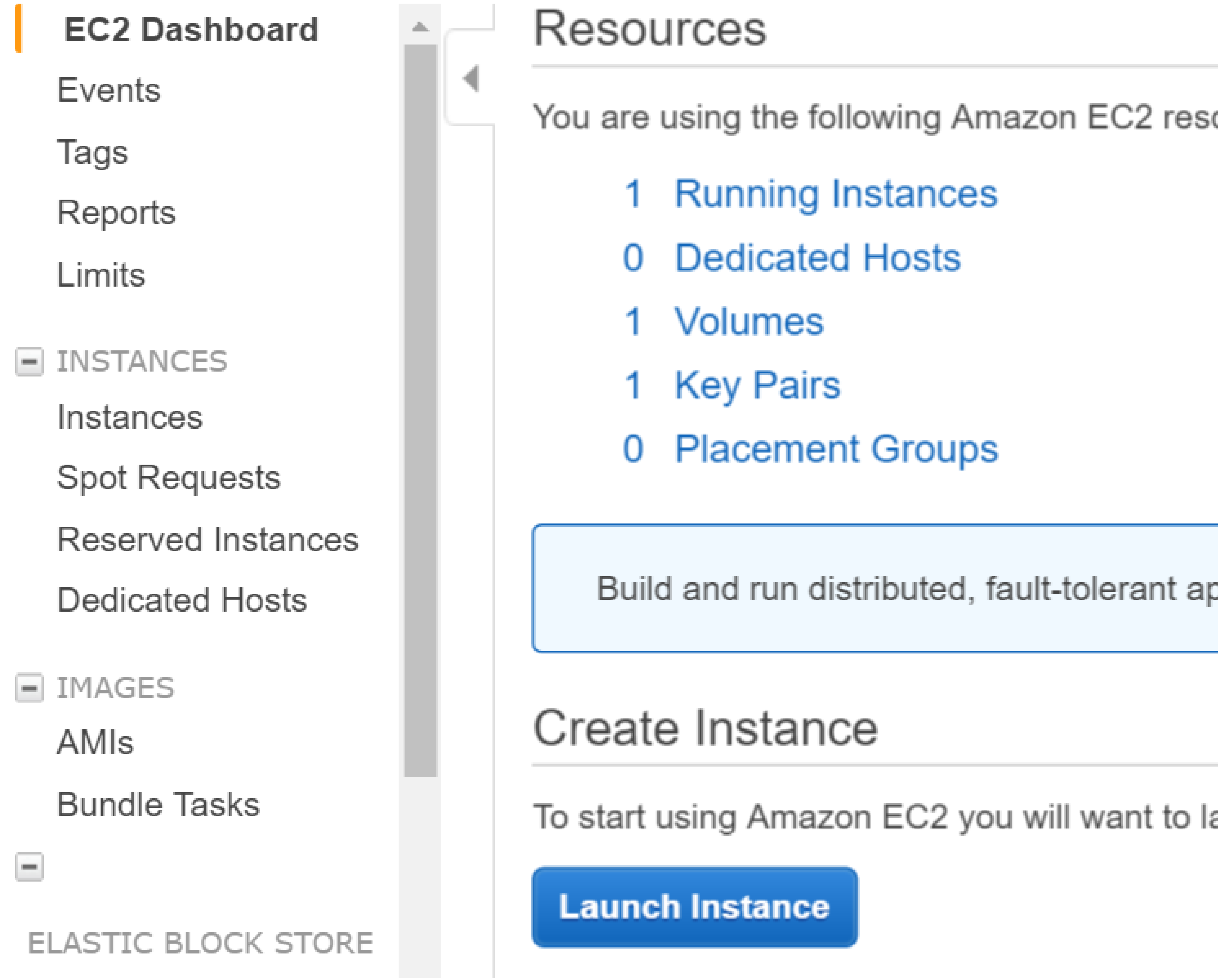
2: AWS EC2 Meteor 1.4

Examples

AWS

. OSS . .

AWS Console . EC2 . EC2 . .



Resources

You are using the following Amazon EC2 reso

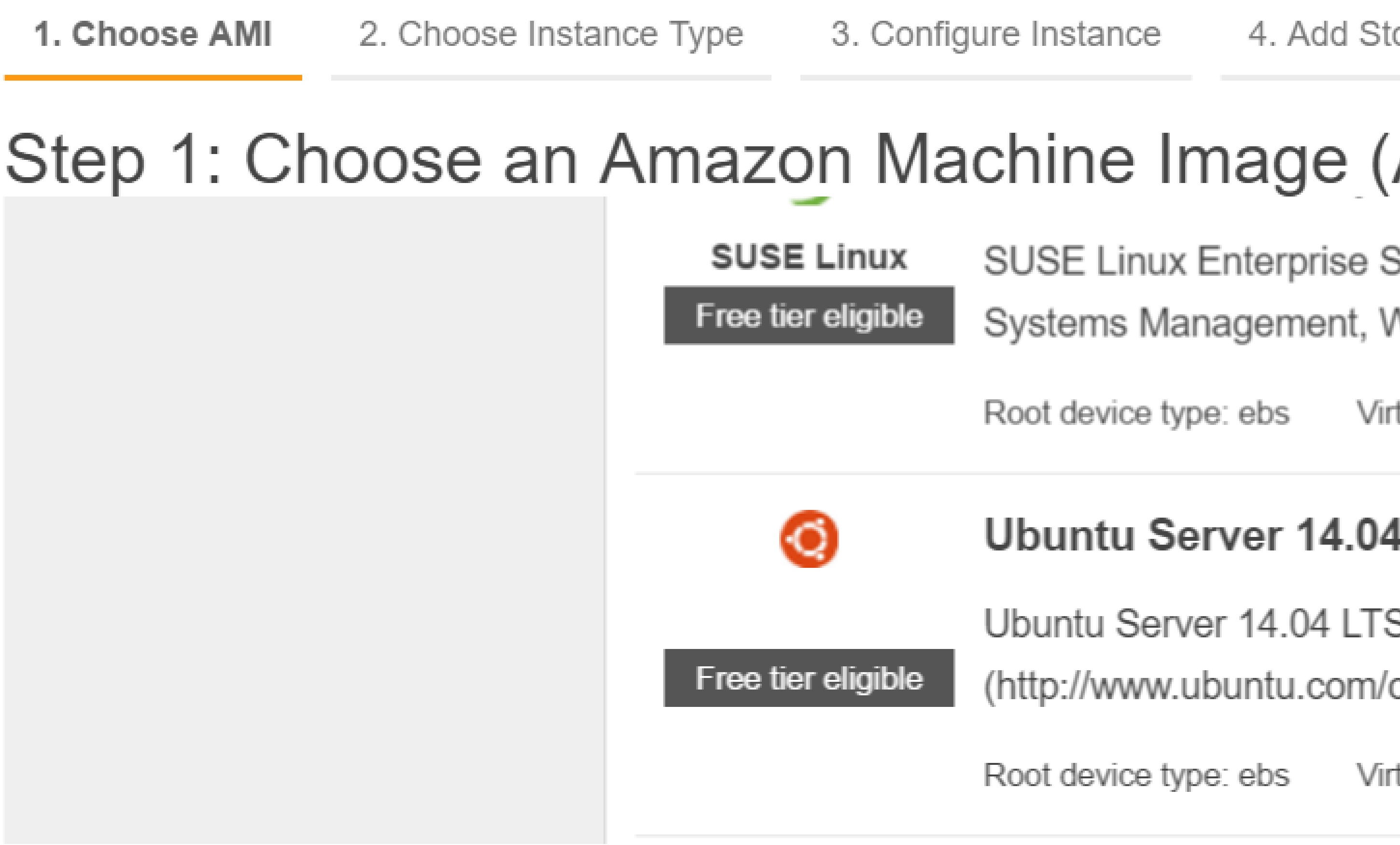
1	Running Instances
0	Dedicated Hosts
1	Volumes
1	Key Pairs
0	Placement Groups

Build and run distributed, fault-tolerant ap

Create Instance

To start using Amazon EC2 you will want to la

.



1. Choose AMI 2. Choose Instance Type 3. Configure Instance 4. Add Sto

Step 1: Choose an Amazon Machine Image (

SUSE Linux
Free tier eligible
SUSE Linux Enterprise S
Systems Management, W
Root device type: ebs Virt

Ubuntu Server 14.04
Free tier eligible
Ubuntu Server 14.04 LTS
(http://www.ubuntu.com/c
Root device type: ebs Virt

.

aws ( )

```
ssh -i "myprivatekey.pem" ubuntu@ec2-xx-xx-xx-xx.ap-south-1.compute.amazonaws.com
```

ec2-xx-xx-xx-xx.ap-south-1.compute.amazonaws.com amazon DNS . . IP .

AWS (mupx)

1. aws server ssh

example `/home/ubuntu/.ssh/myprivatekey.pem`

2. .

```
sudo apt-get update
```

3.

```
 sudo apt-get install python-software-properties
```

4. npm (nvm).

```
sudo apt-get install npm
```

nvm

```
curl https://raw.githubusercontent.com/creationix/nvm/v0.11.1/install.sh | bash
```

```
nvm install 4.4.7

nvm use 4.4.7
```

5. aws cli

```
sudo apt-get install awscli
```

6.

```
 sudo npm install -g mupx

 sudo npm install -g mupx-letsencrypt
```

(1.4 mpux-letsencrypt)

7. mupx   .

```
mupx-letsencrypt init
```

.

```
/usr/bin/env: node: No such file or directory



sudo ln -s /usr/bin/nodejs /usr/bin/node
```

8.

```
curl https://install.meteor.com | /bin/sh
```

9. mup.json (:   1   )

   (vi   )

```
 nano mup.json
```

mup.json

```
    {
  // Server authentication info
  "servers": [
    {
      "host": "ec2-xx-xx-xx-xx.ap-south-1.compute.amazonaws.com",
      "username": "ubuntu",
      //"password": "password",
      // or pem file (ssh based authentication)
      "pem": "~/.ssh/myprivatekey.pem",
      // Also, for non-standard ssh port use this
      //"sshOptions": { "port" : 49154 },
      // server specific environment variables
      "env": {}
    }
  ],

  // Install MongoDB on the server. Does not destroy the local MongoDB on future setups
  "setupMongo": true,

  // WARNING: Node.js is required! Only skip if you already have Node.js installed on server.
  "setupNode": false,

  // WARNING: nodeVersion defaults to 0.10.36 if omitted. Do not use v, just the version
number.
  //"nodeVersion": "4.4.7",

  // Install PhantomJS on the server
  "setupPhantom": true,

  // Show a progress bar during the upload of the bundle to the server.
  // Might cause an error in some rare cases if set to true, for instance in Shippable CI
  "enableUploadProgressBar": true,

  // Application name (no spaces).
  "appName": "my-app",

  // Location of app (local directory). This can reference '~' as the users home directory.
  // i.e., "app": "/Users/ubuntu/my-app",
  // This is the same as the line below.
  "app": "/Users/ubuntu/my-app",

  // Configure environment
  // ROOT_URL must be set to https://YOURDOMAIN.com when using the spiderable package & force
SSL
  // your NGINX proxy or Cloudflare. When using just Meteor on SSL without spiderable this is
not necessary
  "env": {
    "PORT": 80,
    "ROOT_URL": "http://myapp.com",
     // only needed if mongodb is on separate server
    "MONGO_URL": "mongodb://url:port/MyApp",
    "MAIL_URL":   "smtp://postmaster%40myapp.mailgun.org:adj87sjhd7s@smtp.mailgun.org:587/"
  },

  // Meteor Up checks if the app comes online just after the deployment.
  // Before mup checks that, it will wait for the number of seconds configured below.
  "deployCheckWaitTime": 60
}
```

10. mongo Meteor .

    ```
    mupx-letsencrypt setup
    ```

11. mupx

```
mupx-letsencrypt deploy
```

mupx

```
mupx logs -f
```

Docker

```
 docker -D info
```

```
netstat -a
```

CPU

```
 top
```

mongo aws mongo shell acccess .

```
sudo apt-get install mongodb-clients
```

mongodb

```
mongo projectName
```

mongo shell .

```
 db.version()
 db.users.find()
```

. https://github.com/arunoda/meteorupup

mupx-letsencrypt . https://www.npmjs.com/package/mupx-letsencrypt

AWS EC2 Meteor 1.4 : https://riptutorial.com/ko/meteor/topic/4773/aws-ec2-meteor-1-4----

3: Codeship Galaxy

Nate Strausers Metulsor Apps Modulus Galaxy Codeship .

Examples

- deployment_token.json :

  ```
  METEOR_SESSION_FILE=deployment_token.json meteor login
  ```

- Codeship . (https://codeship.com/projects/PROJECT_NUMBER/configure_environment)
  - METEOR_TARGET : your.domain.com
  - METEOR_TOKEN : deployment_token.json / . : `{"sessions": {"www.meteor.com": {"session": "12345 ...`
  - METEOR_SETTING : / .json. : `{"private": {...`
- https://codeship.com/projects/YOUR_PROJECT_NUMBER/deployment_branches/new
  - . : .
- " " .

```
echo $METEOR_TOKEN > deployment_token.json
echo $METEOR_SETTINGS > deployment_settings.json
meteor npm prune --production
DEPLOY_HOSTNAME=galaxy.meteor.com METEOR_SESSION_FILE=deployment_token.json meteor deploy
$METEOR_TARGET --settings deployment_settings.json
```

Codeship Galaxy : https://riptutorial.com/ko/meteor/topic/6743/codeship-galaxy---

4: ES2015 ( )

MDN : https://developer.mozilla.org/en/docs/web/javascript/reference/statements/import MDN : https://developer.mozilla.org/en/docs/web/javascript / reference / statements / export ExploringJS : http://exploringjs.com/es6/ch_modules.html

Examples

```
import url from 'url';
import moment from 'moment';
```

```
import { Meteor } from 'meteor/meteor';
import { SimpleSchema } from 'meteor/aldeed:simple-schema';
```

Meteor

package.js :

```
Npm.depends({
  moment: "2.8.3"
});
```

:

```
import moment from 'moment';
```

```
// Default export
export default {};

// Named export
export const SomeVariable = {};
```

Meteor

mainModule

```
export const SomeVar = {};
```

ES2015 ( ) : https://riptutorial.com/ko/meteor/topic/3763/es2015-------

5: ESLint

Examples

Meteor eslint

`eslint-import-resolver-meteor` `eslint-config-airbnb` Meteor .

/ ES7 lint Meteor `babel-parser` .

```
cd my-project
npm install --save-dev eslint-config-airbnb eslint-plugin-import eslint-plugin-react eslint-plugin-jsx-a11y eslint babel-eslint eslint-import-resolver-meteor
touch .eslintrc.json
```

`.eslintrc.json` `.eslintrc.json` . .

```
{
  "parser": "babel-eslint",
  "settings": {
    "import/resolver": "meteor"
  },
  "extends": "airbnb",
  "rules": {}
}
```

npm lint

`package.json` .

```
{
  "scripts": {
    "lint": "eslint .;exit 0"
  }
}
```

`npm run lint` .

linting `exit 0` . `npm eslint` .

ESLint : https://riptutorial.com/ko/meteor/topic/3772/eslint

6: Meteor + React + ReactRouter

Meteor React ReactRouter . 0.

.

1-

2 - React + ReactRouter

3 -

4-

Examples

1 - https://www.meteor.com/install .

2 . (`--bare`)

```
meteor create --bare MyAwesomeProject
```

(`-p`) :

```
cd MyAwesomeProject

mkdir -p client server imports/api imports/ui/{components,layouts,pages}
imports/startup/{client,server}
```

client / main.html HTML .

```
<head>
   <meta charset="utf-8">
   <title>My Awesome Meteor_React_ReactRouter_Roles App</title>
</head>

<body>
  Welcome to my Meteor_React_ReactRouter_Roles app
</body>
```

5- : (3000 '-p 3000')

```
meteor run -p 3000
```

'localhost : 3000'

:

- . , *imports / startup / {client, server}* index.js .

- https://github.com/rafa-lft/Meteor_React_Base . *Step1_CreateProject* .

React + ReactRouter

```
. cd MyAwesomeProject
```

1-

```
meteor npm install --save react-router@3.0.0 react@15.5.4 react-dom@15.5.4
```

2 - / main.html :

```
<body>
   <div id="react-root"></div>
</body>
```

reactRouter '# react-root' .

3- imports / ui / layouts / App.jsx

```
import React, { Component } from 'react';
import PropTypes from 'prop-types';


class App extends Component {
  constructor(props) {
    super(props);
  }

  render() {
    return (
      <div>
        {this.props.children}
      </div>
    );
  }
}

App.propTypes = {
  children: PropTypes.node
};

export default App;
```

4- imports / startup / client / Routes.jsx Routes

```
import ReactDOM from 'react-dom';
import React, { Component } from 'react';
import { Router, Route, IndexRoute, browserHistory } from 'react-router';

import App from '../../ui/layouts/App.jsx';

import NotFound from '../../ui/pages/NotFound.jsx';
import Index from '../../ui/pages/Index.jsx';
```

```
class Routes extends Component {
  constructor(props) {
    super(props);
  }

  render() {
    return (
      <Router history={ browserHistory }>
        <Route path="/" component={ App }>
          <IndexRoute name="index" component={ Index }/>
          <Route path="*" component={ NotFound }/>
        </Route>
      </Router>
    );
  }
}

Routes.propTypes = {};


Meteor.startup(() =>{
  ReactDOM.render(
    <Routes/>,
    document.getElementById('react-root')
  );
});
```

:

- . imports / ui / pages {Index.jsx, NotFound.jsx} .
- https://github.com/rafa-lft/Meteor_React_Base . . *Step2_ReactRouter*

3 -

. `cd MyAwesomeProject`

1 - : `meteor add accounts-base accounts-password react-meteor-data`

2 - imports / startup / Routes.jsx . *render ()* .

```
  render() {
    return (
      <Router history={ browserHistory }>
        <Route path="/" component={ App }>
          <IndexRoute name="index" component={ Index }/>
          <Route name="login" path="/login" component={ Login }/>
          <Route name="signup" path="/signup" component={ Signup }/>
          <Route name="users" path="/users" component={ Users }/>
          <Route name="editUser" path="/users/:userId" component={ EditUser }/>
          <Route path="*" component={ NotFound }/>
        </Route>
      </Router>
    );
  }
```

:

- . , imports / startup / server / index.js imports / ui / layouts / {App, NavBar} .jsx import / ui / pages / {, , , } .jsx .
- https://github.com/rafa-lft/Meteor_React_Base . *Step3_Accounts* .

1- (https://github.com/alanning/mete-roles)

```
meteor add alanning:roles
```

2 - . / api / accounts / roles.js

```
const ROLES = {
  ROLE1: 'ROLE1',
  ROLE2: 'ROLE2',
  ADMIN: 'ADMIN'
};

export default ROLES;
```

3 - / . `Roles.setUserRoles(user.id, roles);` `Roles.setUserRoles(user.id, roles);` https://github.com/alanning/meteor-roles http://alanning.github.io/meteor-roles/classes/Roles.html .

4- (https://github.com/rafa-lft/Meteor_React_Base .*Tep4_roles* .)) . / / / Routes.jsx

```

class Routes extends Component {
  constructor(props) {
    super(props);
  }

  authenticate(roles, nextState, replace) {
    if (!Meteor.loggingIn() && !Meteor.userId()) {
      replace({
        pathname: '/login',
        state: {nextPathname: nextState.location.pathname}
      });
      return;
    }
    if ('*' === roles) { // allow any logged user
      return;
    }
    let rolesArr = roles;
    if (!_.isArray(roles)) {
      rolesArr = [roles];
    }
    // rolesArr = _.union(rolesArr, [ROLES.ADMIN]);// so ADMIN has access to everything
    if (!Roles.userIsInRole(Meteor.userId(), rolesArr)) {
      replace({
        pathname: '/forbidden',
        state: {nextPathname: nextState.location.pathname}
      });
    }
  }

  render() {
    return (
      <Router history={ browserHistory }>
        <Route path="/" component={ App }>
          <IndexRoute name="index" component={ Index }/>
          <Route name="login" path="/login" component={ Login }/>
          <Route name="signup" path="/signup" component={ Signup }/>

          <Route name="users" path="/users" component={ Users }/>

          <Route name="editUser" path="/users/:userId" component={ EditUser }
                 onEnter={_.partial(this.authenticate, ROLES.ADMIN)} />


          {/* ********************
           Below links are there to show Roles authentication usage.
           Note that you can NOT hide them by
           { Meteor.user() && Roles.userIsInRole(Meteor.user(), ROLES.ROLE1) &&
           <Route name=.....
           }
           as doing so will change the Router component on render(), and ReactRouter will
complain with:
           Warning: [react-router] You cannot change <Router routes>; it will be ignored

           Instead, you can/should hide them on the NavBar.jsx component... don't worry: if
someone tries to access
           them, they will receive the Forbidden.jsx component
           *************/ }
          <Route name="forAnyOne" path="/for_any_one" component={ ForAnyone }/>

          <Route name="forLoggedOnes" path="/for_logged_ones" component={ ForLoggedOnes }
                 onEnter={_.partial(this.authenticate, '*')} />

          <Route name="forAnyRole" path="/for_any_role" component={ ForAnyRole }
                 onEnter={_.partial(this.authenticate, _.keys(ROLES))}/>

          <Route name="forRole1or2" path="/for_role_1_or_2" component={ ForRole1or2 }
                 onEnter={_.partial(this.authenticate, [ROLES.ROLE1, ROLES.ROLE2])} />

          <Route name="forRole1" path="/for_role1" component={ ForRole1 }
                 onEnter={_.partial(this.authenticate, ROLES.ROLE1)}/>

          <Route name="forRole2" path="/for_role2" component={ ForRole2 }
                 onEnter={_.partial(this.authenticate, ROLES.ROLE2)} />


          <Route name="forbidden" path="/forbidden" component={ Forbidden }/>

          <Route path="*" component={ NotFound }/>
        </Route>
      </Router>
    );
  }
}
```

onEnter . . onEnter 2 . (http://underscorejs.org/#partial) () . onEnter *authenticate* .

- . '/ login' .
- === '*' .
- (Roles.userIsInRole) .
- , ADMIN .

(onEnter), , .

ReactRouter (3) Render . Routes.jsx . authenticate / forbidden .

ReactRouter Meteor . . Meteor.user () .

createContainer Meteor.user () . import / ui / layouts / NavBar.jsx .

```
export default createContainer((/* {params}*/) =>{
  Meteor.user(); // so we render again in logout or if any change on our User (ie: new roles)
  const loading = !subscription.ready();
  return {subscriptions: [subscription], loading};
}, NavBar);
```

:

- . , imports / startup / server / index.js imports / ui / layouts / {App, NavBar} .jsx import / ui / pages / {, , , } .jsx .
- https://github.com/rafa-lft/Meteor_React_Base . *Step4_roles* .

Meteor + React + ReactRouter : https://riptutorial.com/ko/meteor/topic/10114/meteor-plus-react-plus-reactrouter

7: Meteor.call

Examples

Meteor.call

```
Meteor.call(name, [arg1, arg2...], [asyncCallback])
```

(1)
(2)
(3) arg1, arg2 ... EJSON []
(4) asyncCallback []

: (**Session** **ReactiveVar**)

```
    var syncCall = Meteor.call("mymethod") // Sync call
```

.

```
    Meteor.methods({
        mymethod: function() {
            let asyncToSync =  Meteor.wrapAsync(asynchronousCall);
            // do something with the result;
            return  asyncToSync;
        }
    });
```

.

:

```
Meteor.call("mymethod", argumentObjectorString, function (error, result) {
    if (error) Session.set("result", error);
    else Session.set("result",result);
}
Session.get("result") -> will contain the result or the error;

//Session variable come with a tracker that trigger whenever a new value is set to the session
variable. \ same behavior using ReactiveVar
```

```
Meteor.methods({
    mymethod: function(ObjectorString) {
        if (true) {
            return true;
        } else {
            throw new Meteor.Error("TitleOfError", "ReasonAndMessageOfError"); // This will
and up in the error parameter of the Meteor.call
        }
    }
});
```

Meteor .

```
Meteor.methods({
  getData() {
    return 'Hello, world!';
  }
});
```

```
<template name="someData">
  {{#if someData}}
    <p>{{someData}}</p>
  {{else}}
    <p>Loading...</p>
  {{/if}}
</template>
```

```
Template.someData.onCreated(function() {
  Meteor.call('getData', function(err, res) {
    Session.set('someData', res);
  });
});

Template.someData.helpers({
  someData: function() {
    return Session.get('someData');
  }
});
```

ReactiveVar

```
Meteor.methods({
  getData() {
    return 'Hello, world!';
  }
});
```

```
<template name="someData">
  {{#if someData}}
    <p>{{someData}}</p>
  {{else}}
    <p>Loading...</p>
  {{/if}}
</template>
```

```
Template.someData.onCreated(function() {

  this.someData = new ReactiveVar();

  Meteor.call('getData', (err, res) => {
    this.someData.set(res);
  });
});

Template.someData.helpers({
  someData: function() {
    return Template.instance().someData.get();
  }
});
```

`reactive-var` . `meteor add reactive-var` run `meteor add reactive-var` `meteor add reactive-var` .

Meteor.call : https://riptutorial.com/ko/meteor/topic/3068/meteor-call---

8: Meteor

Examples

`HTTP [S] _PROXY` env var

Meteor (: ,) .

, Meteor `HTTP_PROXY` `HTTPS_PROXY` ( ) . Meteor :

- Linux Mac OS X

```
export HTTP_PROXY=http://user:password@1.2.3.4:5678
export HTTPS_PROXY=http://user:password@1.2.3.4:5678
meteor update
```

- Windows

```
SET HTTP_PROXY=http://user:password@1.2.3.4:5678
SET HTTPS_PROXY=http://user:password@1.2.3.4:5678
meteor update
```

- Nginx Meteor App Ubuntu .
- 14 Nginx SSL
- Nginx Ubuntu Meteor JS
- SSL
- NameCheap SSL

Meteor : https://riptutorial.com/ko/meteor/topic/517/meteor---

9: Mongo

Ruby Genghis . http://genghisapp.com/

MongoHQ .
http://www.mongohq.com/

Mongo 10Gen Mongo Monitoring Service :
https://mms.mongodb.com/

MongoClient Meteor, Complete Free, Open Source Cross-Platform .

RoboMongo MongoDB

Examples

. variety.js . schema.js. . Mongo .

variety.js

```
mongo test --eval "var collection = 'users'" variety.js
```

schema.js

```
mongo --shell schema.js
```

* .meteorapp.com

--url . 60 / . robomongo .

```
# get the MONGO_URL string for your app
meteor mongo --url $METEOR_APP_URL
```

* .meteor.com

mongodump . / . ! ! ! ! :

```
# get the MONGO_URL string for your app
meteor mongo --url $METEOR_APP_URL

# then quickly copy all the info into the following command
mongodump -u username -p password --port 27017 --db meteor_app_url_com --host production-db-
b1.meteor.io
```

Meteor ?

/ dump BSON blob . .

```
mongodump --db meteor
```

meteordump meteorrestore . . drop .

```
# make sure your app is running
meteor

# then import your data
mongorestore --port 3001 --db meteor /path/to/dump

# a partial import after running > db.comments.drop()
mongorestore --port 3001 --db meteor /path/to/dump -c comments.bson
```

JSON

.

```
mongoexport --db meteor --collection foo --port 3001 --out foo.json
```

Meteor JSON

Meteor . json --jsonArray .

```
mongoimport --db meteor --port 3001 --collection foo --file foo.json
```

Mongo . .

```
// run mongod so we can create a staging database
// note that this is a separate instance from the meteor mongo and minimongo instances
mongod

// import the json data into a staging database
// jsonArray is a useful command, particularly if you're migrating from SQL
mongoimport -d staging -c assets < data.json --jsonArray

// navigate to your application
cd myappdir

// run meteor and initiate it's database
meteor

// connect to the meteor mongodb
meteor mongo --port 3002

// copy collections from staging database into meteor database
db.copyDatabase('staging', 'meteor', 'localhost');
```

. Mongo . . , 2GB . prealloc 2GB. . MongoDB .

reduce-mongodb-database-file-size
mongo-prealloc-files-take-up room

```
// compact the database from within the Mongo shell
db.runCommand( { compact : 'mycollectionname' } )

// repair the database from the command line
mongod --config /usr/local/etc/mongod.conf --repair --repairpath /Volumes/X/mongo_repair --
nojournal

// or dump and re-import from the command line
mongodump -d databasename
echo 'db.dropDatabase()' | mongo databasename
mongorestore dump/databasename
```

. Mongo / dbpath ( ) .

* .meteor.com Mongo

`--url` ? .

```
meteor mongo --url YOURSITE.meteor.com
```

Meteor Mongo

. ' ' tar.gz . .meteor / db mongo . mongodb mongodb Mongo_URL Metoor mongo .

```
MONGO_URL='mongodb://user:password@host:port/databasename'
```

...

```
/var/log/mongodb/server1.log
```

. . ...
mongodb-log-file-growth

...

```
ls /var/log/mongodb/
```

...

```
// put the following in the /etc/logrotate.d/mongod file
/var/log/mongo/*.log {
    daily
    rotate 30
    compress
    dateext
    missingok
    notifempty
    sharedscripts
    copytruncate
    postrotate
```

```
        /bin/kill -SIGUSR1 `cat /var/lib/mongo/mongod.lock 2> /dev/null` 2> /dev/null || true
    endscript
}

// to manually initiate a log file rotation, run from the Mongo shell
use admin
db.runCommand( { logRotate : 1 } )
```

Mongo : https://riptutorial.com/ko/meteor/topic/3707/mongo--

10: Mongo

. . . . . . , " .

" . SQL . Meteor API . , API .

meteor mongo .

```
# run meteor
meteor

# access the database shell in a second terminal window
meteor mongo
```

Examples

```
db.posts.find().forEach(function(doc){
    db.posts.update({_id: doc._id}, {$set:{'version':'v1.0'}}, false, true);
});
```

```
db.posts.find().forEach(function(doc){
    if(doc.arrayOfObjects){
        // the false, true at the end refers to $upsert, and $multi, respectively
        db.accounts.update({_id: doc._id}, {$unset: {'arrayOfObjects': "" }}, false, true);
    }
});
```

```
db.originalName.renameCollection("newName" );
```

....

```
db.posts.find({'text': /.*foo.*|.*bar.*/i})
```

```
db.posts.find().forEach(function(doc){
    if(doc.oldField){
        db.posts.update({_id: doc._id}, {$set:{'newField':doc.oldField}}, false, true);
    }
});
```

```
db.posts.find().forEach(function(doc){
    if(doc.commenters){
        var firstCommenter = db.users.findOne({'_id': doc.commenters[0]._id });
        db.clients.update({_id: doc._id}, {$set:{'firstPost': firstCommenter }}, false, true);

        var firstCommenter = db.users.findOne({'_id': doc.commenters[doc.commenters.length -
1]._id });
        db.clients.update({_id: doc._id}, {$set:{'lastPost': object._id }}, false, true);
    }
});
```

(,   )

```
db.posts.find().forEach(function(doc){
    if(doc.commentsBlobId){
        var commentsBlob = db.comments.findOne({'_id': commentsBlobId });
        db.posts.update({_id: doc._id}, {$set:{'comments': commentsBlob }}, false, true);
    }
});
```

.

```
db.posts.find().forEach(function(doc){
    if(!doc.foo){
        db.posts.update({_id: doc._id}, {$set:{'foo':''}}, false, true);
    }
});
```

.

```
db.posts.find().forEach(function(doc){
    if(!doc.foo){
        db.posts.update({_id: doc._id}, {$set:{'foo':'bar'}}, false, true);
    }
});
```

```
db.posts.find().forEach(function(doc){
    if(doc.foo === 'bar'){
        db.posts.remove({_id: doc._id});
    }
});
```

```
db.posts.find().forEach(function(doc){
    if(doc.foo === 'bar'){
        db.posts.update({_id: doc._id}, {$set:{'foo':'squee'}}, false, true);
    }
});
```

Null

```
db.posts.find().forEach(function(doc){
    if(doc.oldfield){
        // the false, true at the end refers to $upsert, and $multi, respectively
        db.accounts.update({_id: doc._id}, {$unset: {'oldfield': "" }}, false, true);
    }
});
```

ObjectId String .

```
db.posts.find().forEach(function(doc){
     db.accounts.update({_id: doc._id}, {$set: {'_id': doc._id.str }}, false, true);
});
```

```
var newvalue = "";
db.posts.find().forEach(function(doc){
     if(doc.foo){
         newvalue = '"' + doc.foo + '"';
         db.accounts.update({_id: doc._id}, {$set: {'doc.foo': newvalue}});
     }
});
```

```
var newvalue = null;
db.posts.find().forEach(function(doc){
     if(doc.foo){
         newvalue = '"' + doc.foo + '"';
         db.accounts.update({_id: doc._id}, {$set: {'doc.foo': newvalue}});
     }
});
```

_id ObjectID

```
db.posts.find().forEach(function(doc){
    if(doc._id){
        db.posts.update({_id: doc._id}, {$set:{ timestamp: new
Date(parseInt(doc._id.str.slice(0,8), 16) *1000) }}, false, true);
    }
});
```

Date ObjectID

```
var timestamp = Math.floor(new Date(1974, 6, 25).getTime() / 1000);
var hex       = ('00000000' + timestamp.toString(16)).substr(-8); // zero padding
var objectId  = new ObjectId(hex + new ObjectId().str.substring(8));
```

.

```
db.posts.find({"tags.0": {$exists: true }})
```

Mongo   : https://riptutorial.com/ko/meteor/topic/3708/mongo--

11: MongoDB

MongoDB . SQL MongoDB JSON BSON . Meteor MongoDB MongoDB Meteor .

Examples

DB , Meteor Mongo DB

.

1. `mongodump --host some-mongo-host.com:1234 -d DATABASE_NAME -u DATABASE_USER -p DATABASE_PASSWORD` `dump` . `DATABASE_NAME` .
2. `dump` . `mongorestore --db meteor --drop -h localhost --port 3001 DATABASE_NAME`

DB URL

Meteor :

```
meteor mongo --url
```

Mongo DB Meteor

Meteor `MONGO_URL` .

Linux / MacOS :

```
MONGO_URL="mongodb://some-mongo-host.com:1234/mydatabase" meteor
```

```
export MONGO_URL="mongodb://some-mongo-host.com:1234/mydatabase"
meteor
```

Windows

: "

```
set MONGO_URL=mongodb://some-mongo-host.com:1234/mydatabase
meteor
```

NPM

```
//package.json

"scripts": {
    "start": "MONGO_URL=mongodb://some-mongo-host.com:1234/mydatabase meteor"
}

$ npm start
```

MongoDB

URL `MONGO_URL` MongoDB Meteor Meteor (Meteor) .

MongoDB Meteor / . : `Meteor.userId()`

Linux / Mac :

```
MONGO_URL="none" meteor
```

```
export MONGO_URL="none"
meteor
```

Windows :

```
set MONGO_URL=none
meteor
```

Meteor **`mongo`** .

```
meteor mongo
```

: Meteor .

`mongo` .

```
show collections
```

, , MongoDB .

`find()` . :

```
db.collection.find({name: 'Matthias Eckhart'});
```

`name` `Matthias Eckhart` . .

.

```
db.collection.insert({name: 'Matthias Eckhart'});
```

update() . .

```
db.collection.update({name: 'Matthias Eckhart'}, {$set: {name: 'John Doe'}});
```

name John Doe   (Matthias Eckhart).

multi true . .

```
db.collection.update({name: 'Matthias Eckhart'}, {$set: {name: 'John Doe'}}, {multi: true});
```

name Matthias Eckhart   John Doe .

remove()   . .

```
db.collection.remove({name: 'Matthias Eckhart'});
```

name   . .

MongoDB : https://riptutorial.com/ko/meteor/topic/1874/mongodb

12: MongoDB

meteor 0.6 *   "mongodb   ?

(Minimongo)

https://github.com/utunga/pocketmeteor/tree/master/packages/mongowrapper

Examples

Andrew Mao .

```
Meteor.publish("someAggregation", function (args) {
    var sub = this;
    // This works for Meteor 0.6.5
    var db = MongoInternals.defaultRemoteCollectionDriver().mongo.db;

    // Your arguments to Mongo's aggregation. Make these however you want.
    var pipeline = [
        { $match: doSomethingWith(args) },
        { $group: {
            _id: whatWeAreGroupingWith(args),
            count: { $sum: 1 }
        }}
    ];

    db.collection("server_collection_name").aggregate(
        pipeline,
        // Need to wrap the callback so it gets called in a Fiber.
        Meteor.bindEnvironment(
            function(err, result) {
                // Add each of the results to the subscription.
                _.each(result, function(e) {
                    // Generate a random disposable id for aggregated documents
                    sub.added("client_collection_name", Random.id(), {
                        key: e._id.somethingOfInterest,
                        count: e.count
                    });
                });
                sub.ready();
            },
            function(error) {
                Meteor._debug( "Error doing aggregation: " + error);
            }
        )
    );
});
```

`Mongo.Collection#rawCollection()`

.

Meteor 1.3   .

```
Meteor.methods({
   'aggregateUsers'(someId) {
      const collection = MyCollection.rawCollection()
      const aggregate = Meteor.wrapAsync(collection.aggregate, collection)

      const match = { age: { $gte: 25 } }
      const group = { _id:'$age', totalUsers: { $sum: 1 } }

      const results = aggregate([
         { $match: match },
         { $group: group }
      ])

      return results
   }
})
```

MongoDB   : https://riptutorial.com/ko/meteor/topic/4199/mongodb-

13: Nightwatch -

Nightwatch Meteor Acceptance End-to-End v0.5 PHP Spark to Blaze React . Continuous Integration . .

Nightwatch API
Nightwatch.js Google

Examples

Nightwatch . Nightwatch .

.meteor / nightwatch.json

Meteor v1.3 , `default` `phantom` .

```
{
  "nightwatch": {
    "version": "0.9.8"
  },
  "src_folders": [
    "./tests/nightwatch/walkthroughs"
  ],
  "custom_commands_path": [
    "./tests/nightwatch/commands"
  ],
  "custom_assertions_path": [
    "./tests/nightwatch/assertions"
  ],
  "output_folder": "./tests/nightwatch/reports",
  "page_objects_path": "./tests/nightwatch/pages",
  "globals_path": "./tests/nightwatch/globals.json",
  "selenium": {
    "start_process": true,
    "server_path": "./node_modules/starrynight/node_modules/selenium-server-standalone-
jar/jar/selenium-server-standalone-2.45.0.jar",
    "log_path": "tests/nightwatch/logs",
    "host": "127.0.0.1",
    "port": 4444,
    "cli_args": {
      "webdriver.chrome.driver":
"./node_modules/starrynight/node_modules/chromedriver/bin/chromedriver"
    }
  },
  "test_settings": {
    "default": {
      "launch_url": "http://localhost:5000",
      "selenium_host": "127.0.0.1",
      "selenium_port": 4444,
      "pathname": "/wd/hub",
      "silent": true,
      "disable_colors": false,
      "firefox_profile": false,
      "ie_driver": "",
      "screenshots": {
        "enabled": false,
        "path": "./tests/nightwatch/screenshots"
      },
      "desiredCapabilities": {
        "browserName": "chrome",
        "javascriptEnabled": true,
        "acceptSslCerts": true,
        "loggingPrefs": {
          "browser": "ALL"
        }
      },
      "exclude": "./tests/nightwatch/unittests/*",
      "persist_globals": true,
      "detailed_output": false
    },
    "phantom": {
      "desiredCapabilities": {
        "browserName": "phantomjs",
        "javascriptEnabled": true,
        "databaseEnabled": false,
        "locationContextEnabled": false,
        "applicationCacheEnabled": false,
        "browserConnectionEnabled": false,
        "webStorageEnabled": false,
        "acceptSslCerts": true,
        "rotatable": false,
        "nativeEvents": false,
        "phantomjs.binary.path": "./node_modules/starrynight/node_modules/phantomjs-prebuilt/bin/phantomjs"
      }
    },
    "unittests": {
      "selenium": {
        "start_process": false,
        "start_session": false
      },
      "filter": "./tests/nightwatch/unittests/*",
      "exclude": ""
    }
  }
}
```

Nightwatch . **(phantomj)** .

`package.json` devDependencies .

```
{
  "devDependencies": {
    "nightwatch": "0.9.8",
    "selenium-server-standalone-jar": "2.45.0",
    "chromedriver": "2.19.0",
    "phantomjs-prebuilt": "2.1.12"
  }
}
```

depndencies .

```
cd myapp
```

```
meteor npm install
```

Nightwatch .

```
nightwatch -c .meteor/nightwatch.json
nightwatch -c .meteor/nightwatch.json --env phantom
```

.

`package.json` . devDependencies .

```
{
  "name": "myapp",
  "version": "1.0.0",
  "scripts": {
    "start": "meteor --settings settings-development.json",
    "nightwatch": "nightwatch -c .meteor/nightwatch.json",
    "phantom": "nightwatch -c .meteor/nightwatch.json --env phantom",
  }
}
```

.

```
meteor npm run-script nightwatch
meteor npm run-script phantom
```

`nightwatch -c .meteor/nightwatch.json` . , devops .

Meteor Nightwatch .

```
/myapp
/myapp/.meteor/nightwatch.json
/client/main.html
/client/main.js
/client/main.css
/tests
/tests/nightwatch
/tests/nightwatch/assertions
/tests/nightwatch/commands
/tests/nightwatch/data
/tests/nightwatch/logs
/tests/nightwatch/pages
/tests/nightwatch/reports
/tests/nightwatch/screenshots
/tests/nightwatch/walkthroughs
/tests/nightwatch/walkthroughs/critical_path.js
/tests/nightwatch/globals.json
```

Nightwatch `Meteor.settings` `globals.json` .

globals.json

```
{
  "default" : {
    "url" : "http://localhost:3000",
    "user": {
      "name": "Jane Doe",
      "username" : "janedoe",
      "password" : "janedoe123",
      "email" : "janedoe@test.org",
      "userId": null
    }
  },
  "circle" : {
    "url" : "http://localhost:3000",
    "user": {
      "name": "Jane Doe",
      "username" : "janedoe",
      "password" : "janedoe123",
      "email" : "janedoe@test.org"
      "userId": null
    }
  },
  "galaxy" : {
    "url" : "http://myapp.meteorapp.com",
    "user": {
      "name": "Jane Doe",
      "username" : "janedoe",
      "password" : "janedoe123",
      "email" : "janedoe@test.org"
      "userId": null
    }
  }
}
```

, ,   .

```
module.exports = {
  "Login App" : function (client) {
    client
      .url(client.globals.url)
      .login(client.globals.user.email, client.globals.user.password)
      .end();
  }
};
```

Nightwatch -   : https://riptutorial.com/ko/meteor/topic/5901/nightwatch-----

14: Upstart

Examples

Ubuntu Amazon Web Services Rackspace IaaS (Infrastructure as a Service) . Ubuntu . Upstart . Upstart .

Upstart -
Ubuntu Upstart

Intro,
Ubuntu Karmic NodeJS

Git GitHub . GitHub . . Meteor (500MB) .

```
cd /var/www
sudo git clone http://github.com/myaccount/myapp.git
cd /var/www/myapp
meteor build --directory ../myapp-production
sudo service myapp restart
```

.

```
cd myapp
meteor build --directory ../output
cd ..
scp output -r username@destination_host:/var/www/myapp-production
```

`/etc/init/ directory` upstart . `.conf` `/etc/init/myapp.conf` . .

```
## /etc/init/myapp.conf
description "myapp.mydomain.com"
author      "somebody@gmail.com"

# Automatically Run on Startup
start on started mountall
stop on shutdown

# Automatically Respawn:
respawn
respawn limit 99 5

script
    export HOME="/root"
    export MONGO_URL='mongodb://myapp.compose.io:27017/meteor'
    export ROOT_URL='http://myapp.mydomain.com'
    export PORT='80'

    exec /usr/local/bin/node /var/www/myapp/main.js >> /var/log/myapp.log 2>&1
end script
```

,   .

```
# /etc/init/myapp.conf
description "myapp.mydomain.com"
author      "somebody@gmail.com"

# used to be: start on startup
# until we found some mounts weren't ready yet while booting:
start on started mountall
stop on shutdown

# Automatically Respawn:
respawn
respawn limit 99 5

script
    # upstart likes the $HOME variable to be specified
    export HOME="/root"

    # our example assumes you're using a replica set and/or oplog integreation
    export MONGO_URL='mongodb://mongo-a,mongo-b,mongo-c:27017/?replicaSet=meteor'

    # root_url and port are the other two important environment variables to set
    export ROOT_URL='http://myapp.mydomain.com'
    export PORT='80'

    exec /usr/local/bin/node /var/www/production/main.js >> /var/log/node.log 2>&1
end script
```

Upstart   .

```
sudo service myapp start
```

Meteor Apps

https://www.phusionpassenger.com/
https://github.com/phusion/passenger
https://github.com/phusion/passenger/wiki/Phusion-Passenger:-Meteor-tutorial#wiki-installing

Upstart   : https://riptutorial.com/ko/meteor/topic/3377/upstart--

15: Windows Meteor

Mac Linux `meteor` `ssh` . Windows . .

Examples

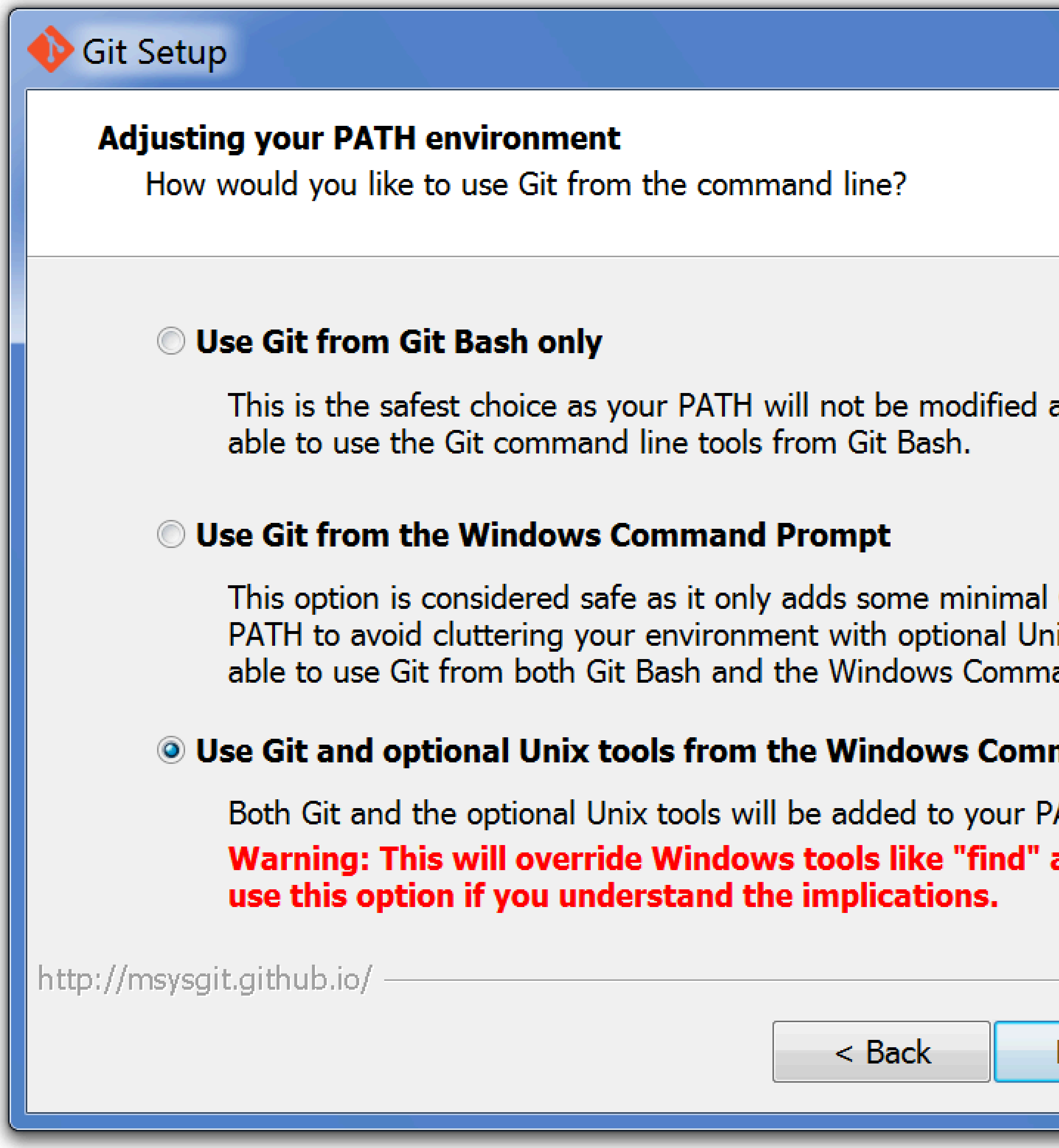
PuTTY ()

Windows PATH Unix PuTTY SSH . PuTTY .

1. `meteor admin get-machine <os-architecture> --json`
2. JSON
3. PuTTY .
4. PuTTY , !

Cygwin (Windows Unix)

Git for Windows "Windows Git Unix " .



Adjusting your PATH environment

How would you like to use Git from the command line?

○ Use Git from Git Bash only

This is the safest choice as your PATH will not be modified a
able to use the Git command line tools from Git Bash.

○ Use Git from the Windows Command Prompt

This option is considered safe as it only adds some minimal
PATH to avoid cluttering your environment with optional Uni
able to use Git from both Git Bash and the Windows Comma

● Use Git and optional Unix tools from the Windows Comm

Both Git and the optional Unix tools will be added to your P

**Warning: This will override Windows tools like "find" a
use this option if you understand the implications.**

`meteor admin get-machine <os-architecture>` Linux Mac . .

Windows Meteor : https://riptutorial.com/ko/meteor/topic/518/windows-meteor---

16:

Examples

. IDE Meteor .

- Atom - Meteor Javascript IDE. .
- Cloud9 - Meteor Cloud Development .
- MeteorDevTools - Blaze, DDP Minimongo Chrome .
- - .
- WebStorm - Meteor IDE.

'Hello World' .

- Robomongo - Mongo . .
- JSON Generator - .
- MacOSX - MacOSX GUI.
- MongoHub - RoboMongo Mongo GUI. MacOSX .
- Mongo3 - . . Ruby .
- Mongo - MMS . MongoDB Atlas .
- Mongo Express - Node.js MongoDB

Meteor . Meteor .

- Google - .
- Zitub.io - GitHub .
- InVision - .
- Meeting Hero - .
- Hackpad - .
- - .
- MadEye - .
- Screenhero - .
- Proto.io - .
- HuBoard - GitHub .
- Zapier - . .
- Teamwork.com - .
- Sprint.ly - GitHub kanban .
- LucidChart - Visio .
- Waffle.io - GitHub Trello / ZenHub .

REST

Meteor API REST . REST API Chrome .

-
- DHC

:

- Hurl.it
- RequestBin

Chrome Safari . 99 % . Firefox .

- Firefox - Firebug
-
- `meteor add meteortoys:allthings`

iOS

Texttastic - iOS .

- Github iPad .

CodeHub - GitHub . .

iOctocat - Github .

iPad iMockups - . .

- iOS . iOS .

JSON Designer - .

: https://riptutorial.com/ko/meteor/topic/4200/-

17: / NPM

Examples

Meteor /

Meteor   /   .

```
meteor node -v
```

/ NPM : https://riptutorial.com/ko/meteor/topic/4599/---npm

18:

Meteor .

Examples

autopublish autopublish . autopublish .

autopublish autopublish .

```
$ meteor remove autopublish
```

. .

```
import { Mongo } from 'meteor/mongo';
import { Meteor } from 'meteor/meteor';

const Todos = new Mongo.Collection('todos');

const TODOS = [
  { title: 'Create documentation' },
  { title: 'Submit to Stack Overflow' }
];

if (Meteor.isServer) {
  Meteor.startup(function () {
    TODOS.forEach(todo => {
      Todos.upsert(
        { title: todo.title },
        { $setOnInsert: todo }
      );
    });
  });

  // first parameter is a name.
  Meteor.publish('todos', function () {
    return Todos.find();
  });
}

if (Meteor.isClient) {
  // subscribe by name to the publication.
  Meteor.startup(function () {
    Meteor.subscribe('todos');
  })
}
```

.

null .

```
Meteor.publish(null, function() {
  return SomeCollection.find();
})
```

.

.

```
Meteor.publish('somePublication', function() {
  return SomeCollection.find()
})
```

.

```
Meteor.subscribe('somePublication')
```

Meteor Spacebars Blaze .

`Meteor` .

.

```
<template name="myTemplate">
  We will use some data from a publication here
</template>
```

.

```
Template.myTemplate.onCreated(function() {
  const templateInstance = this;
  templateInstance.subscribe('somePublication')
})
```

.

: .

▪

.

```
import { Mongo } from 'meteor/mongo';
import { Meteor } from 'meteor/meteor';
import { Random } from 'meteor/random';

if (Meteor.isClient) {
  // established this collection on the client only.
  // a name is required (first parameter) and this is not persisted on the server.
  const Messages = new Mongo.Collection('messages');
  Meteor.startup(function () {
    Meteor.subscribe('messages');
    Messages.find().observe({
      added: function (message) {
        console.log('Received a new message at ' + message.timestamp);
      }
    });
  })
}

if (Meteor.isServer) {
  // this will add a new message every 5 seconds.
  Meteor.publish('messages', function () {
    const interval = Meteor.setInterval(() => {
      this.added('messages', Random.id(), {
        message: '5 seconds have passed',
        timestamp: new Date()
      })
    }, 5000);
    this.added('messages', Random.id(), {
      message: 'First message',
      timestamp: new Date()
    });
    this.onStop(() => Meteor.clearInterval(interval));
  });
}
```

▪

. `this.userId` ID. .

```
import Secrets from '/imports/collections/Secrets';

Meteor.publish('protected_data', function () {
  if (!this.userId) {
    this.error(new Meteor.Error(403, "Not Logged In."));
    this.ready();
  } else {
    return Secrets.find();
  }
});
```

.

```
Meteor.subscribe('protected_data', {
  onError(err) {
    if (err.error === 403) {
      alert("Looks like you're not logged in");
    }
  },
});
```

File / imports / collections / Secrets .

```
const Secrets = new Mongo.Collection('secrets');
```

. . ( ).

`onCreated` .

```
Template.myTemplate.onCreated(function() {
  this.parameter = new ReactiveVar();
  this.autorun(() => {
    this.subscribe('myPublication', this.parameter.get());
  });
});
```

() `.parameter` .

▪

JS

```
Template.templateName.onCreated(function(){
    this.subscribe('subsription1');
    this.subscribe('subscription2');
});
```

HTML

```
<template name="templateName">
    {{#if Template.subscriptionsReady }}
        //your actual view with data. it can be plain HTML or another template
    {{else}}
        //you can use any loader or a simple header
        <h2> Please wait ... </h2>
    {{/if}}
</template>
```

. .

```
import { Recipes } from '../imports/api/recipes.js';
import { Meteor } from 'meteor/meteor';

Meteor.publish('recipes', function() {
  if(this.userId) {
    return Recipe.find({});
  } else {
    this.ready();  // or: return [];
  }
});
```

.

"" .

```
Meteor.publish('USER_THREAD', function(postId) {
  let userId   = this.userId;

  let comments = Comments.find({ userId, postId });
  let replies  = Replies.find({ userId, postId });

  return [comments, replies];
});
```

. Meteor._sleepForMs(ms);

```
Meteor.publish('USER_DATA', function() {
    Meteor._sleepForMs(3000); // Simulate 3 seconds delay
    return Meteor.users.find({});
});
```

.   .

```
// client/subscriptions.js
Meteor.subscribe('usersDirectory');
Meteor.subscribe('userProfile', Meteor.userId());

// server/publications.js
// Publish users directory and user profile

Meteor.publish("usersDirectory", function (userId) {
    return Meteor.users.find({}, {fields: {
        '_id': true,
        'username': true,
        'emails': true,
        'emails[0].address': true,

        // available to everybody
        'profile': true,
        'profile.name': true,
        'profile.avatar': true,
        'profile.role': true
    }});
});
Meteor.publish('userProfile', function (userId) {
    return Meteor.users.find({_id: this.userId}, {fields: {
        '_id': true,
        'username': true,
        'emails': true,
        'emails[0].address': true,

        'profile': true,
        'profile.name': true,
        'profile.avatar': true,
        'profile.role': true,

        // privately accessible items, only availble to the user logged in
        'profile.visibility': true,
        'profile.socialsecurity': true,
        'profile.age': true,
        'profile.dateofbirth': true,
        'profile.zip': true,
        'profile.workphone': true,
        'profile.homephone': true,
        'profile.mobilephone': true,
        'profile.applicantType': true
    }});
});
```

: https://riptutorial.com/ko/meteor/topic/1323/-

19: .

1. Meteor.wrapAsync (func, [context])

func :	(error, result) / .
: ( )	.

(error, result) => {} .

Meteor.wrapAsync Fiber .

Fibers https://www.npmjs.com/package/fibers .

Examples

NPM .

NPM simple-oauth2 oauth2.client.getToken(callback) oauth2.client.getToken(callback) .

```
const oauth2 = require('simple-oauth2')(credentials);

const credentials = {
    clientID: '#####',
    clientSecret: '#####',
    site: "API Endpoint Here."
};

Meteor.startup(() => {
    let token = Meteor.wrapAsync(oauth2.client.getToken)({});
    if (token) {
        let headers = {
            'Content-Type': "application/json",
            'Authorization': `Bearer ${token.access_token}`
        }

        // Make use of requested OAuth2 Token Here (Meteor HTTP.get).
    }
});
```

. : https://riptutorial.com/ko/meteor/topic/2530/-------

20:

Meteor 1.3 Meteor Meteor.js . Meteor . .

.

/
jQuery 3 .

lib
lib Meteor . , .

imports .

packages . `meteor add package:name` `meteor add package:name` , Meteor `package.js` . .

.

. .

. , .

.

Richard Silverton .

.

Examples

Meteor . Meteor bundler "".

```
client/                                   # client application code
client/compatibility/                     # legacy 3rd party javascript libraries
imports/                                  # for lazy loading feature
lib/                                      # any common code for client/server.
packages/                                 # place for all your atmosphere packages
private/                                  # static files that only the server knows about
public/                                   # static files that are available to the client
server/                                   # server code
tests/                                    # unit test files (won't be loaded on client or
server)
```

: Meteor Guide>

. . .

. . .

```
client/                                  # client application code
packages/                                # place for all your atmosphere packages
packages/foo/client                      # client application code
packages/foo/lib                         # any common code for client/server
packages/foo/server                      # server code
packages/foo/tests                       # tests
server/                                  # server code
```

/

Meteor `ecmascript` , ES6 ES2015 . Javascript `import` . `/packages` `/imports` `/packages` .

```
imports                                 #
imports/api                             # isomorphic methods
imports/lib                             # any common code for client/server
imports/client                          # client application code
imports/server                          # server code
```

. : franken-app . / .

```
client/                                  # client application code
client/compatibility/                    # legacy 3rd party javascript libraries
imports                                  #
imports/api                              # isomorphic methods
imports/lib                              # any common code for client/server
imports/client                           # client application code
imports/server                           # server code
lib/                                     # any common code for client/server.
packages/                                # place for all your atmosphere packages
packages/foo/client                      # client application code
packages/foo/lib                         # any common code for client/server
packages/foo/server                      # server code
packages/foo/tests                       # tests
private/                                 # static files that only the server knows about
public/                                  # static files that are available to the client
server/                                  # server code
tests/                                   # unit test files (won't be loaded on client or
server)
```

HTML .

main . .

lib / .

.

.

: Meteor Guide> Application Structure>

: https://riptutorial.com/ko/meteor/topic/3072/-

21:

Examples

Chrome Safari . Chrome ''. Safari Preferences> Advanced ' Develop ' .

Firefox Firebug .

`debugger` .

```
Meteor.methods({
  doSomethingUself: function(){
    debugger;
    niftyFunction();
  }
});
```

Node Inspector

Node Inspector . .

HowToNode -
Strongloop -
Meteor Node Inspector Meteor.js

dr - Meteor Meteor . Meteor . , Robomongo, Nightwatch ... . NodeInspector .

```
# install node-inspector
terminal-a$  npm install -g node-inspector

# start meteor
terminal-a$  NODE_OPTIONS='--debug-brk --debug' mrt run

# alternatively, some people report this syntax being better
terminal-a$  sudo NODE_OPTIONS='--debug' ROOT_URL=http://myapp.com meteor --port 80

# launch node-inspector along side your running app
terminal-b$  node-inspector

# go to the URL given by node-inspector
http://localhost:8080/debug?port=5858
```

npm

`debug` npm .

MeteorHacks - npm Meteor

Meteor 1.0.2 Chrome ! :

```
meteor shell
```

DevTools

: https://riptutorial.com/ko/meteor/topic/3378/

22:

Examples

Iron Router

Iron Router

:

```
meteor add iron:router
```

```
Router.configure({
    //Any template in your routes will render to the {{> yield}} you put inside your layout
template
    layoutTemplate: 'layout',
    loadingTemplate: 'loading'
});
```

```
//this is equal to home page
Router.route('/', function (){
    this.render('home')
});

Router.route('/some-route', function () {
    this.render('template-name');
});
```

```
Router.route('/items/:_id', function () {
    this.render('itemPage', {
        data: function() {
            return Items.findOne({_id: this.params._id})
        }
    });
});
```

2

```
Router.route('/one-route/route', function() {
    //template 'oneTemplate' has {{> yield 'secondary'}} in HTML
    this.render('oneTemplate');

    //this yields to the secondary place
    this.render('anotherTemplate', {
        to: 'secondary'
    });

    //note that you can write a route  for '/one-route'
    //then another for '/one-route/route' which will function exactly like above.
});
```

```
Router.route('/waiting-first', {
    waitOn: function() {
        //subscribes to a publication
        //shows loading template until subscription is ready
        return Meteor.subscribe('somePublication')
    },

    action: function() {
        //render like above examples
    }
});
```

```
Router.route('/waiting-first', {
    waitOn: function() {
        //subscribes to a publication
        //shows loading template until subscription is ready
        return [Meteor.subscribe('somePublication1'),Meteor.subscribe('somePublication2')];
    },

    action: function() {
        //render like above examples
    }
});
```

: http://iron-meteor.github.io/iron-router/

FlowRouter

FlowRouter Iron Router .

FlowRouter

```
meteor add kadira:flow-router
```

.

- : `meteor add kadira:blaze-layout`
- React : `meteor add kadira:react-layout`

(Blaze).

```
<template name="mainLayout">
  {{> Template.dynamic template=area}}
</template>
```

```
FlowRouter.route('/blog/:postId', {
  action: function (params) {
    BlazeLayout.render("mainLayout", {
      area: "blog"
    });
  }
});
```

/

Iron Router .

```
FlowRouter.route("/blog/:catId/:postId", {
  name: "blogPostRoute",
  action: function (params) {
    //...
  }
})
```

.   .

```
// url: /blog/travel/france?showcomments=yes
var catId = FlowRouter.getParam("catId"); // returns "travel"
var postId = FlowRouter.getParam("postId"); // returns "france"

var color = FlowRouter.getQueryParam("showcomments"); // returns "yes"
```

: https://riptutorial.com/ko/meteor/topic/5119/

23:

a) publish-release .json b) . .

Atmosphere . .meteor / versions .

, , . . .

Meteor .
https://forums.meteor.com/t/custom-meteor-release/13736/6

Examples

.

```
meteor publish-release clinical.meteor.rc6.json
```

.

```
meteor run --release clinical:METEOR@1.1.3-rc6
```

Atmosphere NPM `package.json` . .

```
{
  "track":"distroname:METEOR",
  "version":"x.y.z",
  "recommended": false,
  "tool": "distroname:meteor-tool@x.y.z",
  "description": "Description of the Distro",
  "packages": {
    "accounts-base":"1.2.0",
    "accounts-password":"1.1.1",
    ...
  }
}
```

, . Ronen .

http://practicalmeteor.com/using-meteor-publish-release-to-extend-the-mete-command-line-tool/1

helloworld . StarryNight . .

.meteor / versions

StarryNight `.meteor/versions` .

```
npm install -g starrynight
cd myapp
starrynight generate-release-json
```

StarryNight `.meteor/versions` `packages` `.meteor/versions` . JSON .

```
meteor show --ejson METEOR@1.2.1
```

Checkout

```
meteor publish-release --from-checkout
```

`/packages` `/packages` git . `/packages` .

```
git submodule foreach git pull origin master
```

: https://riptutorial.com/ko/meteor/topic/4201/--

24:

Examples

- CSS

ViewPorts . : CSS . ( ) MVC MVVM ViewPorts MVC .

```
// desktop
@media only screen and (min-width: 960px) {
}

// landscape orientation
@media only screen and (min-width: 768px) {
}

// portrait orientation
@media only screen and (min-width: 480px) {
}
```

768px ( ) 1024px ( ) . 3 : 4 iPad . .

. PhoneGap . UI !

```
// create a window of a specific size
var w=window.open('','', 'width=100,height=100');
w.resizeTo(500,500);

// prevent window resize
var size = [window.width,window.height];  //public variable
$(window).resize(function(){
    window.resizeTo(size[0],size[1]);
});
```

.

```
meteor add appcache
meteor add grounddb
```

-

. DOM javascript .

```
// prevent scrolling on the whole page
// this is not meteorish; TODO: translate to meteor-centric code
document.ontouchmove = function(e) {e.preventDefault()};

// prevent scrolling on specific elements
// this is not meteorish; TODO: translate to meteor-centric code
scrollableDiv.ontouchmove = function(e) {e.stopPropagation()};
```

css .

```
#appBody {
  overflow: hidden;
}

#contentContainer {
  .content-scrollable {
    overflow-y: auto;
    -webkit-overflow-scrolling: touch;
  }
}
```

.

```
<div id="appBody">
  <div id="contentContainer">
    <div class="content-scrollable">
      <!-- content -->
    </div>
  </div>
</div>
```

. FastClick Hammer. .

```
meteor add fastclick
meteor add hammer:hammer
```

FastClick Hammer . Todos .

```
Template.appBody.onRendered(function() {
  if (Meteor.isCordova) {
    // set up a swipe left / right handler
    this.hammer = new Hammer(this.find('#appBody'));
    this.hammer.on('swipeleft swiperight', function(event) {
      if (event.gesture.direction === 'right') {
        Session.set(MENU_KEY, true);
      } else if (event.gesture.direction === 'left') {
        Session.set(MENU_KEY, false);
      }
    });
  }
});
```

`mobile-config.js` `mobile-config.js` .

```
App.icons({
  // iOS
  'iphone': 'resources/icons/icon-60x60.png',
  'iphone_2x': 'resources/icons/icon-60x60@2x.png',
  'ipad': 'resources/icons/icon-72x72.png',
  'ipad_2x': 'resources/icons/icon-72x72@2x.png',

  // Android
  'android_ldpi': 'resources/icons/icon-36x36.png',
  'android_mdpi': 'resources/icons/icon-48x48.png',
  'android_hdpi': 'resources/icons/icon-72x72.png',
  'android_xhdpi': 'resources/icons/icon-96x96.png'
});

App.launchScreens({
  // iOS
  'iphone': 'resources/splash/splash-320x480.png',
  'iphone_2x': 'resources/splash/splash-320x480@2x.png',
  'iphone5': 'resources/splash/splash-320x568@2x.png',
  'ipad_portrait': 'resources/splash/splash-768x1024.png',
  'ipad_portrait_2x': 'resources/splash/splash-768x1024@2x.png',
  'ipad_landscape': 'resources/splash/splash-1024x768.png',
  'ipad_landscape_2x': 'resources/splash/splash-1024x768@2x.png',

  // Android
  'android_ldpi_portrait': 'resources/splash/splash-200x320.png',
  'android_ldpi_landscape': 'resources/splash/splash-320x200.png',
  'android_mdpi_portrait': 'resources/splash/splash-320x480.png',
  'android_mdpi_landscape': 'resources/splash/splash-480x320.png',
  'android_hdpi_portrait': 'resources/splash/splash-480x800.png',
  'android_hdpi_landscape': 'resources/splash/splash-800x480.png',
  'android_xhdpi_portrait': 'resources/splash/splash-720x1280.png',
  'android_xhdpi_landscape': 'resources/splash/splash-1280x720.png'
});
```

Meteor Cordova Phonegap Integration .

Xcode Yosemite . Meteor iOS .

- .
- XCode ( )
- Xcode 6.1 .
- .

```
# 5.  clone and rebuild the ios-sim locally
#     (this step will not be needed in future releases)
git clone https://github.com/phonegap/ios-sim.git
cd ios-sim
rake build

# 6.  make sure we can update the .meteor/packages locations
#     (this step will not be needed in future releases)
sudo chmod -R 777 ~/.meteor/packages

# 7.  copy the new build into Meteor locations
#     (this step will not be needed in future releases)
for i in `find ~/.meteor/packages/meteor-tool/ -name ios-sim -type f`; do
  cp -R ./build/Release/ios-sim "$i"
done

# 8.  install the ios platform to your app
cd myapp
meteor list-platforms
meteor add-platform ios
meteor list-platforms

# 9.  and that there aren't dead processes
```

```
ps -ax
kill -9 <pid>
# /Users/abigailwatson/.meteor/packages/meteor-
tool/.1.0.35.wql4jh++os.osx.x86_64+web.browser+web.cordova/meteor-tool-
os.osx.x86_64/dev_bundle/mongodb/bin/mongod
# tail -f /Users/abigailwatson/Code/Medstar/dart/webapp/.meteor/local/cordova-
build/platforms/ios/cordova/console.log

# 10.  make sure there are correct permissions on the application (important!)
sudo chmod -R 777 .meteor/local/

# 11.  run app
meteor run ios

# 12.  if that doesn't work, clear the directory
sudo rm -rf .meteor/local

# 13a.  run meteor again to create the default browser build
meteor

# 13b.  run it a second time so bootstrap and other packages get downloaded into the browser
build
ctrl-x
meteor

# 14.  then run the ios version
ctrl-x
meteor run ios
```

Xcode . " .

IOS

- Apple
- ID
- UUID
- iOS App Development
    - KeychainAccess CertificateSigningRequest
    - CertificateSigningRequest https://developer.apple.com/account/ios/profile/profileCreate.action .
    - .
- Xcode> > Apple .

IOS

- iPhone WiFi . , .
- `sudo meteor run ios-device`
- !

Cordova (config.xml)

Meteor `mobile-config.js` Cordova `config.xml` .

```
Project_folder
 ├── /.meteor
 └── mobile-config.js
```

`mobile-config.js` (, , launchscreens Cordova) .

```
App.info({
  id: 'com.example.matt.uber',
  name: 'über',
  description: 'Get über power in one button click',
  author: 'Matt Development Group',
  email: 'contact@example.com',
  website: 'http://example.com'
});

// Set up resources such as icons and launch screens.
App.icons({
  'iphone': 'icons/icon-60.png',
  'iphone_2x': 'icons/icon-60@2x.png',
  // ... more screen sizes and platforms ...
});

App.launchScreens({
  'iphone': 'splash/Default~iphone.png',
  'iphone_2x': 'splash/Default@2x~iphone.png',
  // ... more screen sizes and platforms ...
});

// Set PhoneGap/Cordova preferences
App.setPreference('BackgroundColor', '0xff0000ff');
App.setPreference('HideKeyboardFormAccessoryBar', true);
App.setPreference('Orientation', 'default');
App.setPreference('Orientation', 'all', 'ios');

// Pass preferences for a particular PhoneGap/Cordova plugin
App.configurePlugin('com.phonegap.plugins.facebookconnect', {
  APP_ID: '1234567890',
  API_KEY: 'supersecretapikey'
});
```

`/.meteor/local/cordova-build/config.xml` . `meteor run ios/android` `meteor build` `meteor run ios/android` .

: Meteor > > >

, . Cordova PhoneGap 'deviceready' .

```
 document.addEventListener('deviceready', function(){
  Session.set('deviceready', true);
 }, false);
```

: https://riptutorial.com/ko/meteor/topic/3705/-

25:

Mongo Who, What, When, Where, Why, How . Mongo .

- GeoJSON
When - ObjectID
Who - Meteor
- JSON

Mongo 'What' .

Examples

idGeneration Mongo .

```
MyCollection = new Meteor.Collection('mycollection', {idGeneration : 'MONGO'});
```

Mongo , , , . .

```
Todos.insert({
  text: "foo",                        // String
  listId: Session.get('list_id'),     // String
  value: parseInt(2),                 // Number
  done: false,                        // Boolean
  createdAt: new Date(),              // Dimestamp
  timestamp: (new Date()).getTime(),  // Time
  tags: []                            // Array
});
```

_id

.

```
var docId = Todos.insert({text: 'foo'});
console.log(docId);
```

:

```
Todos.insert({text: 'foo'}, function(error, docId){
  console.log(docId);
});
```

MongoDB (well-use document) . MongoDB .

MongoDB

, timeseries :

```
DailyStats.insert({
   "date" : moment().format("MM-DD-YYYY"),
   "dateIncrement" : moment().format("YYYYMMDD"),
   "dailyTotal" : 0,
   'bucketA': 0,
   'bucketB': 0,
   'bucketC': 0
   });
```

.   Meteor Method, , REST API   .

```
DailyStats.update({_id: doc._id}, {$inc: {bucketA: 1} });
```

Meteor Clinical Meteor   .

- - Dailystats

Regexes

regexes,   deps autoruns   .

```
// create our collection
WordList =  new Meteor.Collection("wordlist");

// and a default session variable to hold the value we're searching for
Session.setDefault('dictionary_search', '');

Meteor.isClient(function(){
    // we create a reactive context that will rerun items when a Session variable gets updated

    Deps.autorun(function(){
        // and create a subscription that will get re-subscribe to when Session variable gets
updated
        Meteor.subscribe('wordlist', Session.get('dictionary_search'));
    });

    Template.dictionaryIndexTemplate.events({
        'keyup #dictionarySearchInput': function(evt,tmpl){
            // we set the Session variable with the value of our input when it changes
            Session.set('dictionary_search', $('#dictionarySearchInput').val());
        },
        'click #dictionarySearchInput':function(){
            // and clear the session variable when we enter the input
            Session.set('dictionary_search', '');
        },
    });
});
Meteor.isServer(function(){
    Meteor.publish('wordlist', function (word_search) {
        // this query gets rerun whenever the client subscribes to this publication
        return WordList.find({
            // and here we do our regex search
            Word: { $regex: word_search, $options: 'i' }
        },{limit: 100});
    });
});
```

HTML :

```
<input id="dictionarySearchInput" type="text" placeholder="Filter..." value="hello"></input>
```

. regexes .

Javascript

-

Mongo GeoJSON , , `window.navigator.geolocation` , Map API, GeoJSON LatLngs , . . .

- mongodb geoJSON .
- geojson.org
- window.navigator.geolocation
- HTML
- Maps API
- google.maps.LatLng
- Google map.data.loadGeoJson
- - - -
- phonegap-googlemaps-plugin
- LatLng
- maps.documentation
- google.maps.LatLng
- 2dsphere
- 2dsphere
- 2dsphere
-

.

```
Meteor.startup(
  function () {
    var wrappedFind = Meteor.Collection.prototype.find;

    // console.log('[startup] wrapping Collection.find')

    Meteor.Collection.prototype.find = function () {
      // console.log(this._name + '.find', JSON.stringify(arguments))
      return wrappedFind.apply(this, arguments);
    }
  },

  function () {
    var wrappedUpdate = Meteor.Collection.prototype.update;

    // console.log('[startup] wrapping Collection.find')

    Meteor.Collection.prototype.update = function () {
      console.log(this._name + '.update', JSON.stringify(arguments))
      return wrappedUpdate.apply(this, arguments);
    }
  }
);
```

. . . . . .

```
Meteor.startup(function(){
  console.log('starting worker....');

  var dataCursor = Posts.find({viewsCount: {$exists: true}},{limit:20});

  var handle = dataCursor.observeChanges({
    added: function (id, record) {
      if(record.viewsCount > 10){
         // run some statistics
         calculateStatistics();

         // or update a value
         Posts.update({_id: id}, {$set:{
           popular: true
         }});

      }
    },
    removed: function () {
      console.log("Lost one.");
    }
  });
});
```

20 ... ; . 'var' . . opslog Node .

Meteor . .

`percolatestudios:synced-cron` synced `percolatestudios:synced-cron` .

- - cron

: https://riptutorial.com/ko/meteor/topic/5120/-

26: (Vars & Dictionaries)

Examples

:

main.html

```
<template name="test">
     <input type="checkbox" id="checkbox1" name="name" value="data">Check Me
      {{showData}}
</template>
```

Main.js

```
 var check_status='';
 //Reactive Var Initialization
 Template.main.onCreated(function (){
       check_status=new ReactiveVar({});

 });

 Template.main.helpers({
       showData : function(){
           return Collection.find(check_status.get());
       }
 });

 Template.main.events({
      "change #checkbox1" : function(){
              check_status.set({field: 'data'});
       }
 });
```

:

var `check_status` `{}` . `Collection.find(check_status.get())` .

`check_status` `{field: data}` `Template.main.events` . `showData` `Collection.find({field: data})` `field` `'data'` `field` .

`reactive var` (`meteor add reactive-var`).

(Vars & Dictionaries) : https://riptutorial.com/ko/meteor/topic/6535/--vars--amp--dictionaries-

27:

cron-tick . ( ) **percolate : synced-cron** .

Examples

cron

percolate : synced-cron .

:

```
 SyncedCron.add({
  name: 'Find new matches for a saved user filter and send alerts',
  schedule: function(parser) {
    // parser is a later.parse object
    return parser.text('every 10 minutes');
  },
  job: function() {
    user.alerts.map(a => a.findMatchesAndAlert());
  }
});
```

:

```
SyncedCron.start();
```

Galaxy .

: https://riptutorial.com/ko/meteor/topic/4772/-

28:

Examples

Meteor . .

```
meteor
```

std_out std_err .

```
meteor > my_app_log.log 2> my_app_err.log
```

. API . Chrome Safari API . Winston .

. Chrome Safari Firebug Firebug .

Console API . .

Chrome

()

Meteor Chrome DevTools Extension Meteor . ! . .

Chrome DevTools ()

0.5 - 0.7 .

```
Template.landingPage.postsList = function(){
  try{
    return Posts.find();
  }catch(error){
    //color code the error (red)
    console.error(error);
  }
}
```

Chrome Logging API . `.group()` .

.

```
Template.landingPage.getId = function(){
  // using a group block to illustrate function scoping
  console.group('coolFunction');

  // inspect the current data object that landingPage is using
  console.log(this);

  // inspect a specific field of the locally scoped data object
  console.log(JSON.stringify(this._id);
  // close the function scope
  console.groupEnd();
  return this._id;
}
```

Chrome API .

```
Template.landingPage.events({
  'click .selectItemButton':function(){
    // color code and count the user interaction (blue)
    console.count('click .selectItemButton');
  }
});
```

. .

```
var DEBUG = false;
var TRACE = false;
Template.landingPage.events({
  'click .selectItemButton':function(){
    TRACE && console.count('click .selectItemButton');

    Meteor.call('niftyAction', function(errorMessage, result){
        if(errorMessage){
            DEBUG && console.error(errorMessage);
        }
    });
  }
});
```

. ( ). .

```
if (!DEBUG_MODE_ON) {
    console = console || {};
    console.log = function(){};

    console.log = function(){};
    console.error = function(){};
    console.count = function(){};
    console.info = function(){};
}
```

Winston . Winston .

https://atmospherejs.com/?q=winston

Winston .

Loglevel

LogLevel . Meteor .

https://atmospherejs.com/practicalmeteor/loglevel

: https://riptutorial.com/ko/meteor/topic/3376/-

29:

Replica Set . Sharded Database horizintally scalled database , Shard Replica Set . Mongo 2 11 3 . 11, 14, 17, 20, 23 . , 3 2 , 3 2 . 11 2 .

Examples

. (.) .

```
# add the names of each server to the host file of each server
sudo nano /etc/hosts
  10.123.10.101 mongo-a
  10.123.10.102 mongo-b
  10.123.10.103 mongo-c

# install mongodb on the server
sudo apt-key adv --keyserver hkp://keyserver.ubuntu.com:80 --recv 7F0CEB10
echo 'deb http://downloads-distro.mongodb.org/repo/ubuntu-upstart dist 10gen' | sudo tee
/etc/apt/sources.list.d/mongodb.list
sudo apt-get update
sudo apt-get install mongodb-10gen


# create the /data/ directories
sudo mkdir /data
sudo mkdir /data/logs
sudo mkdir /data/db

# make sure the mongodb user and group have access to our custom directories
sudo chown -R mongodb:mongodb /data

# edit the mongo upstart file in /etc/init/mongodb.conf
sudo nano /etc/init/mongodb.conf
  start on started mountall
  stop on shutdown
  respawn
  respawn limit 99 5
  setuid mongodb
  setgid mongodb
  script
    exec /usr/bin/mongod --config /etc/mongodb.conf >> /data/logs/mongo-a.log 2>&1
  end script


# edit mongodb configuration file
sudo nano /etc/mongodb.conf
    dbpath=/data/db
    logpath=/data/logs/mongod.log
    logappend=true
    port=27017
    noauth=true
    replSet=meteor
    fork=true

# add a mongo log-rotation file
sudo nano /etc/logrotate.d/mongod
```

```
  /data/logs/*.log {
    daily
    rotate 30
    compress
    dateext
    missingok
    notifempty
    sharedscripts
    copytruncate
    postrotate
        /bin/kill -SIGUSR1 `cat /data/db/mongod.lock 2> /dev/null` 2> /dev/null || true
    endscript
  }

# make sure mongod service is started and running
sudo service mongodb start
sudo reboot
```

mongo .

```
meteor mongo

  > rs.initiate()
  PRIMARY> rs.add("mongo-a")
  PRIMARY> rs.add("mongo-b")
  PRIMARY> rs.add("mongo-c")
  PRIMARY> rs.setReadPref('secondaryPreferred')
```

: https://riptutorial.com/ko/meteor/topic/4332/---

30:   (, jQuery)

Blaze http://bootsnipp.com/ . HTML CSS   .   , ,   .

Examples

Blaze JQuery Bootstrap .

```
    <nav class="nav navbar-nav">
      <li class="dropdown">
        <a href="#" class="dropdown-toggle" data-toggle="dropdown">{{getSelectedValue}} <span
class="glyphicon glyphicon-user pull-right"></span></a>
        <ul class="fullwidth dropdown-menu">
          <li id="firstOption" class="fullwidth"><a href="#">15 Minutes <span class="glyphicon
glyphicon-cog pull-right"></span></a></li>
          <li class="divider"></li>
          <li id="secondOption"><a href="#">30 Minutes <span class="glyphicon glyphicon-stats
pull-right"></span></a></li>
          <li class="divider"></li>
          <li id="thirdOption"><a href="#">1 Hour <span class="badge pull-right"> 42
</span></a></li>
          <li class="divider"></li>
          <li id="fourthOption"><a href="#">4 Hour <span class="glyphicon glyphicon-heart
pull-right"></span></a></li>
          <li class="divider"></li>
          <li id="fifthOption"><a href="#">8 Hours <span class="glyphicon glyphicon-log-out
pull-right"></span></a></li>
        </ul>
      </li>
    </nav>
```

```
Template.examplePage.helpers({
  getSelectedValue:function(){
    return Session.get('selectedValue');
  }
});
Template.dropDownWidgetName.events({
  'click #firstOption':function(){
    Session.set('selectedValue', 1);
  },
  'click #secondOption':function(){
    Session.set('selectedValue', "blue");
  },
  'click #thirdOption':function(){
    Session.set('selectedValue', $('#thirdOption').innerText);
  },
  'click #fourthOption':function(){
    Session.set('selectedValue', Session.get('otherValue'));
  },
  'click #fifthOption':function(){
    Session.set('selectedValue', Posts.findOne(Session.get('selectedPostId')).title);
  },
});
```

Navbars

/ . .

```
Router.configure({
  layoutTemplate: 'appLayout',
});
Router.route('checklistPage', {
    path: '/lists/:_id',
    onBeforeAction: function() {
      Session.set('selectedListId', this.params._id);
      this.next();
    },
    yieldTemplates: {
      'navbarFooter': {
        to: 'footer'
      }
    }
  });
```

Navbar

```
<template name="navbarFooter">
  <nav id="navbarFooterNav" class="navbar navbar-default navbar-fixed-bottom"
role="navigation">
    <ul class="nav navbar-nav">
      <li><a id="addPostLink"><u>A</u>dd Post</a></li>
      <li><a id="editPostLink"><u>E</u>dit Post</a></li>
      <li><a id="deletePostLink"><u>D</u>elete Post</a></li>
    </ul>
    <ul class="nav navbar-nav navbar-right">
      <li><a id="helpLink"><u>H</u>elp</a></li>
    </ul>
  </nav>
</template>
```

```
<template name="appLayout">
  <div id="appLayout">
    <header id="navbarHeader">
      {{> yield 'header'}}
    </header>

      <div id="mainPanel">
        {{> yield}}
      </div>

    <footer id="navbarFooter" class="{{getTheme}}"">
      {{> yield 'footerActionElements' }}
    </footer>
  </div>
</template>
```

UI . . ( ).

```
<template name="topicsPage">
  <div id="topicsPage" class="container">
```

```
    <div class="panel">
      <div class="panel-heading">
        Nifty Panel
      </div>
      <!-- .... -->
      <div class="panel-footer">
        <!-- step 1.  we click on the button object -->
        <div id="createTopicButton" class="btn {{ getPreferredButtonTheme }}">Create
Topic</div>
      </div>
    </div>

    <!-- step 5 - the handlebars gets activated by the javascript controller -->
    <!-- and toggle the creation of new objects in our model -->
    {{#if creatingNewTopic }}
    <div>
      <label for="topicTextInput"></label>
      <input id="topicTextInput" value="enter some text..."></input>
      <button class="btn btn-warning">Cancel</button>
      <button class="btn btn-success">OK</button>
    </div>
    {{/if}}
  </div>
</template>
```

```
// step 2 - the button object triggers an event in the controller
// which toggles our reactive session variable
Template.topicsPage.events({
  'click #createTopicButton':function(){
    if(Session.get('is_creating_new topic'){
      Session.set('is_creating_new_topic', false);
    }else{
      Session.set('is_creating_new_topic', true);
    }
  }
});

// step 0 - the reactive session variable is set false
Session.setDefault('is_creating_new_topic', false);

// step 4 - the reactive session variable invalidates
// causing the creatNewTopic function to be rerun
Template.topicsPage.creatingNewTopic = function(){
  if(Session.get('is_creating_new_topic')){
    return true;
  }else{
    return false;
  }
}
```

. . , , .

```
// client/subscriptions.js
Meteor.subscribe('posts');

//lib/model.js
Posts  = new Meteor.Collection("posts");
Posts.allow({
    insert: function(){
        return true;
    },
    update: function () {
        return true;
    },
    remove: function(){
        return true;
    }
});


// server.publications.js
Meteor.publish('posts', function () {
  return Posts.find();
});
```

.

```
{
  _id: "3xHCsDexdPHN6vt7P",
  title: "Sample Title",
  text: "Lorem ipsum, solar et...",
  tags: ["foo", "bar", "zkrk", "squee"]
}
```

. , . `selectedPost` , `tagObjects` `tag` blogPost . `title` `text` .

```
<template name="blogPost">
  {{#with selectedPost }}
    <div class="blogPost panel panel-default">
      <div class="panel-heading">
        {{ title }}
      </div>
        {{ text }}
      <div class="panel-footer">
        <ul class="horizontal-tags">
          {{#each tagObjects }}
          <li class="tag removable_tag">
            <div class="name">{{tag}}<i class="fa fa-times"></i></div>
          </li>
          {{/each}}
          <li class="tag edittag">
            <input type="text" id="edittag-input" value="" /><i class="fa fa-plus"></i>
          </li>
        </ul>
      </div>
    </div>
  {{/with}}
</template>
```

.

```
// you will need to set the selectedPostId session variable
// somewhere else in your application
Template.blogPost.selectedPost = function(){
  return Posts.findOne({_id: Session.get('selectedPostId')});
}

// next, we use the _.map() function to read the array from our record
// and convert it into an array of objects that Handlebars/Spacebars can parse
Template.blogPost.tagObjects = function () {
    var post_id = this._id;
    return _.map(this.tags || [], function (tag) {
        return {post_id: post_id, tag: tag};
    });
};

// then we wire up click events
Template.blogPost.events({
    'click .fa-plus': function (evt, tmpl) {
        Posts.update(this._id, {$addToSet: {tags: value}});
    },
    'click .fa-times': function (evt) {
        Posts.update({_id: this._id}, {$pull: {tags: this.tag}});
    }
});
```

, , . UI . Sass Stylus Less .

```
// default desktop view
.fa-plus:hover{
  cursor: pointer;
}
.fa-times:hover{
  cursor: pointer;
}
// landscape orientation view for tablets
@media only screen and (min-width: 768px) {
  .blogPost{
     padding: 20px;
  }
}
// portrait orientation view for tablets
@media only screen and (max-width: 768px) {
  .blogPost{
     padding: 0px;
       border: 0px;
  }
}
// phone view
@media only screen and (max-width: 480px) {
  blogPost{
   .panel-footer{
       display: none;
    }
  }
}
```

Meteor . . FlashAlert Handlebar Session .

LESS Bootstrap-3 . .

```
meteor add less
meteor add ian:bootstrap-3
```

: . div . 2 .

```
<template name="postsPage">
  <div id="postsPage" class="page">
    <div id="postsPageAlert" class="{{alertColor}}">{{alertMessage}}</div>
    <div class="postsList">
      <!-- other code you can ignore in this example -->
    </div>
    <div id="triggerAlertButton" class="btn btn-default">
  </div>
</template>
```

Javascript : Template Helper . 2 2 .

```
Session.setDefault('alertLevel', false);
Session.setDefault('alertMessage', "");

Template.postsPage.alertColor = function(){
 if(Session.get('alertLevel') == "Success"){
  return "alert alert-success";
 }else if(Session.get('alertLevel') == "Info"){
  return "alert alert-info";
 }else if(Session.get('alertLevel') == "Warning"){
  return "alert alert-warning";
 }else if(Session.get('alertLevel') == "Danger"){
  return "alert alert-danger";
 }else{
  return "alert alert-hidden"
 }
}

Template.postsPage.alertMessage = function(){
  return Session.get('alertMessage');
}
```

: DOM Visibility CSS postsPage . . .

```
#postsPage{
  .alert{
    display: block;
  }
  .alert-hidden{
    display: none;
  }
}
```

Javascript :

.

```
Template.postsPage.events({
  'click #triggerAlertButton':function(){
    Session.set('alertLevel', 'Success');
    Session.set('alertMessage', 'You successfully read this important alert message.');
  }
});
```

! , ? `alertLevel` `alertMessage` .   ! :)

...

```
<template name="samplePage">
  <div id="samplePage" class="page">
    <ul class="nav nav-tabs">
      <li id="firstPanelTab"><a href="#firstPanel">First</a></li>
      <li id="secondPanelTab"><a href="#secondPanel">Second</a></li>
    </ul>

    <div id="firstPanel" class="{{firstPanelVisibility}}">
      {{> firstPanel }}
    </div>
    <div id="secondPanel" class="{{secondPanelVisibility}}">
      {{> secondPanel }}
    </div>
  </div>
</template>
```

```
// this variable controls which tab is displayed and associated application state
Session.setDefault('selectedPanel', 1);

Template.name.helpers({
  firstPanelVisibility: function (){
    if(Session.get('selectedPanel') === 1){
      return "visible";
    }else{
      return "hidden";
    }
  },
  secondPanelVisibility: function (){
    if(Session.get('selectedPanel') === 2){
      return "visible";
    }else{
      return "hidden";
    }
  },
  thirdPanelVisibility: function (){
    if(Session.get('selectedPanel') === 3){
      return "visible";
    }else{
      return "hidden";
    }
  },
  firstPanelActive: function (){
    if(Session.get('selectedPanel') === 1){
      return "active panel-tab";
    }else{
      return "panel-tab";
    }
  },
  secondPanelActive: function (){
    if(Session.get('selectedPanel') === 2){
      return "active panel-tab";
    }else{
      return "panel-tab";
    }
  },
  thirdPanelActive: function (){
    if(Session.get('selectedPanel') === 3){
      return "active panel-tab";
    }else{
      return "panel-tab";
    }
  }
});
```

```
.visible {
  display: block;
  visibility: visible;
}
.hidden {
  display: none;
  visibility: hidden;
}
```

.

```
<li id="firstPanelTab" class="{{firstPanelActive}}"><a href="#firstPanel">First</a></li>
<li id="secondPanelTab" class="{{secondPanelActive}}"><a href="#secondPanel">Second</a></li>
```

```
Template.firstPanel.helpers({
  firstPanelActive: function (){
    if(Session.get('selectedPanel') === 1){
      return "active";
    }else{
      return "";
    }
  },
  secondPanelActive: function (){
    if(Session.get('selectedPanel') === 2){
      return "active";
    }else{
      return "";
    }
  },
});
```

(, jQuery) : https://riptutorial.com/ko/meteor/topic/4202/--------jquery--

31:

Blaze HTML . Blaze HTML . Blaze Meteor.js .

Examples

```
<template name="myTemplate">
  {{#each results}}
    <div><span>{{name}}</span><span>{{age}}</span></div>
  {{/each}}
</template>
```

```
Template.myTemplate.onCreated(function() {
  this.results = new ReactiveVar();
  Meteor.call('myMethod', (error, result) => {
    if (error) {
      // do something with the error
    } else {
      // results is an array of {name, age} objects
      this.results.set(result);
    }
  });
});

Template.myTemplate.helpers({
  results() {
    return Template.instance().results.get();
  }
});
```

(: childTemplate parentTemplate,).

```
<template name="parentTemplate">
    {{#with someHelperGettingDataForParentTemplate}}
    <h1>My name is {{firstname}} {{lastname}}</h1>
    //some stuffs here
    {{> childTemplate}}
    {{/with}}
</template>
```

childTemplate . , {{firstname}} {{lastname}} childTemplate .

```
<template name="childTemplate">
<h2>My name is also {{firstname}} {{lastname}}</h2>
</template>
```

childTemplate .

```
<template name="parentTemplate">
    {{#with someHelperGettingDataForParentTemplate}}
    <h1>My name is {{firstname}} {{lastname}}</h1>
    //some stuffs here
    {{> childTemplate childData=someHeplerReturningDataForChild}}
    {{/with}}
</template>
```

someHelperReturningDataForChild {profession : "Meteor Developer", : "stackoverflowing"} childTemplate . .

```
<template name="childTemplate">
    <h2>My profession is {{profession}}</h2>
    <h3>My hobby is {{hobby}}</h3>
</template>
```

Blaze . . . .

1. .

.

```
<template name="welcomeMessage">
  <h1>Welcome back {{fullName}}</h1>
</template>
```

Template . firstName lastName .

```
Template.welcomeMessage.helpers({
  fullName: function() {
    const instance = Template.instance();
    return instance.data.firstName + ' ' + instance.data.lastName
  },
});
```

2. (.)

.

```
Template.registerHelper('equals', function(item1, item2) {
  if (!item1 || !item2) {
    return false;
  }

  return item1 === item2;
});
```

`equals` .

```
<template name="registration">
  {{#if equals currentUser.registrationStatus 'Pending'}}
    <p>Don't forget to complete your registration!<p>
  {{/if}}
</template>
```

: https://riptutorial.com/ko/meteor/topic/2434/-

32:

Meteor Javascript Node.js. . , API JSON . , , Meteor Javascript . . .

25

JavaScript
V8
10

Meteor JS

Examples

. Michael Nygard " Release it! " . . Meteor , , , . .

-
-
-
-
-

-
-
-
-

- AJAX
-
-
- Eutropification

2 , . . .

: https://riptutorial.com/ko/meteor/topic/3363/-

33: (Nightwatch)

Nightwatch Meteor Acceptance End-to-End v0.5 PHP Spark to Blaze React . Continuous Integration . .

Nightwatch API
Nightwatch.js Google

Examples

. , . .

1.
2. (: DOM)
3. /

. . . (URL) . :

```
module.exports = {
  "Hello World" : function (client) {
    client
      // the location of our Meteor app
      .url("http://localhost:3000")

      // the size of the viewport
      .resizeWindow(1024, 768)

      // test app output
      .verify.elementPresent('h1')
      .verify.containsText('h1', "Welcome to Meteor!")
      .verify.containsText('p', "You've pressed the button 0 times")
      .verify.elementPresent('button')

      // simulate user input
      .click('button').pause(500)

      // test app output again, to make sure input worked
      .verify.containsText('p', "button 1 times")

      // saving a copy of our viewport pixel grid
      .saveScreenshot('tests/nightwatch/screenshots/homepage.png')
      .end();
  }
};
```

Nightwatch , . Nightwatch :

```
module.exports = {
  "Login App" : function (client) {
    client
      .url("http://localhost:3000")
      .login("janedoe@somewhere.com", "janedoe123")
      .end();
  }
};
```

./tests/nightwatch/commands/login ./tests/nightwatch/commands/login .

```
exports.command = function(username, password) {

  this
    .verify.elementPresent('#login')

      // we clear the input in case there's any data remaining from previous visits
      .clearValue("#emailInput")
      .clearValue("#passwordInput")

      // we simulate key presses
      .setValue("#emailInput", username)
      .setValue("#passwordInput", password)

    // and we simulate a mouse click
    .click("#signInToAppButton").pause(1000)

  return this; // allows the command to be chained.
};
```

id . .

```
<template name="login">
  <div id="login">
    <input id="emailInput" name="email" type="email" />
    <input id="passwordInput" name="password" type="password" />
    <button id="#signInToAppButton">Sign In</button>
  </div>
</template>
```

Nightwatch .execute() API . Session . ./tests/nightwatch/api/meteor/checkSession . :

```
// syncrhonous version; only works for checking javascript objects on client
exports.command = function(sessionVarName, expectedValue) {
  var client = this;
  this
    .execute(function(data){
      return Session.get(data);
    }, [sessionVarName], function(result){
      client.assert.ok(result.value);
      if(expectedValue){
        client.assert.equal(result.value, expectedValue);
      }
    })
    return this;
};
```

.

```
module.exports = {
  "Check Client Session" : function (client) {
    client
      .url("http://localhost:3000")
      .checkSession("currentUser", "Jane Doe")
      .end();
  }
};
```

/ data .

```
tests/nightwatch/data/IM-0001-1001.dcm
```

. (.)

```
<form id="myform">
    <input type="file" id="fileUpload">
    <input type="text" name="first_name">
    <input type="text" name="last_name">

    <input type="date" name="dob_month">
    <input type="date" name="dob_day">
    <input type="date" name="dob_year">

    <input type="radio" name="gender" value="M">
    <input type="radio" name="gender" value="F">
    <input type="radio" name="gender" value="O">

    <input type="select" name="hs_graduation_year">
    <input type="text" name="city">
    <input type="select" name="state">

    <input type="submit" name="submit" value="Submit">
</form>
```

setValue () .

```
module.exports = {
  "Upload Study" : function (client) {
    console.log(require('path').resolve(__dirname +  '/../data' ));

    var stringArray = "Chicago";

    client
      .url(client.globals.url)
      .verify.elementPresent("form#myform")

      // input[type="file"]
      .verify.elementPresent("input#fileUpload")
      .setValue('input#fileUpload', require('path').resolve(__dirname + '/../data/IM-0001-
1001.dcm'))

      // input[type="text"]
      .setValue('input[name="first_name"]', 'First')
      .setValue('input[name="last_name"]', 'Last')

      // input[type="date"]
      .click('select[name="dob_month"] option[value="3"]')
      .click('select[name="dob_day"] option[value="18"]')
      .click('select[name="dob_year"] option[value="1987"]')

      // input[type="radio"]
      .click('input[name="gender"][value="M"]')

      // input[type="number"]
      .click('select[name="hs_graduation_year"] option[value="2002"]')

      // input[type="text"]
      // sometimes Nightwatch will send text faster than the browser can handle
      // which will cause skipping of letters.  In such cases, we need to slow
      // Nightwatch down; which we do by splitting our input into an array
      // and adding short 50ms pauses between each letter
      for(var i=0; i < userIdArray.length; i++) {
        client.setValue('input[name="city"]', stringArray[i]).pause(50)
      }

      // input[type="select"]
      // after an array input above, we need to resume our method chain...
      client.click('select[name="state"] option[value="CA"]')

      // input[type="number"]
      .setValue('input[name="zip"]', '01234')

      //input [ type="submit" ]
      .click('button[type="submit"]')
      .end();
  }
};
```

Daniel Rinehart .

. UI . React .

```
module.exports = {
  url: 'http://localhost:3000/login',
  commands: [{
  login: function(email, password) {
    return this
      .clearValue('input[name="emailAddress"]')
      .clearValue('input[name="password"]')

      .setValue('input[name="emailAddress"]', email)
      .setValue('input[name="password"]', password)

      .verify.elementPresent('#loginButton')
      .click("#loginButton");
  },
  clear: function() {
    return this
      .waitForElementVisible('@emailInput')
      .clearValue('@emailInput')
      .clearValue('@passInput')
      .waitForElementVisible('@loginButton')
      .click('@loginButton')
  },
  checkElementsRendered: function(){
    return this
```

```
      .verify.elementPresent("#loginPage")
      .verify.elementPresent('input[name="emailAddress"]')
      .verify.elementPresent('input[name="password"]')
  },
  pause: function(time, client) {
    client.pause(time);
    return this;
  },
  saveScreenshot: function(path, client){
    client.saveScreenshot(path);
    return this;
  }
}],
  elements: {
    emailInput: {
      selector: 'input[name=email]'
    },
    passInput: {
      selector: 'input[name=password]'
    },
    loginButton: {
      selector: 'button[type=submit]'
    }
  }
};
```

PageObject Nightwatch `verify.elementPresent` `verify.elementPresent` . . .

```
module.exports = {
  tags: ['accounts', 'passwords', 'users', 'entry'],
  'User can sign up.': function (client) {

    const signupPage = client.page.signupPage();
    const indexPage = client.page.indexPage();

    client.page.signupPage()
      .navigate()
      .checkElementsRendered()
      .signup('Alice', 'Doe', 'alice@test.org', 'alicedoe')
      .pause(1500, client);

    indexPage.expect.element('#indexPage').to.be.present;
    indexPage.expect.element('#authenticatedUsername').text.to.contain('Alice Doe');
  },
}
```

(Nightwatch) : https://riptutorial.com/ko/meteor/topic/6454/---nightwatch--

34:

Examples

(MONGO_URL)

MONGO_URL . . .

```
#make sure you're running the node v0.10.21 or later
npm cache clean -f
npm install -g n
sudo n 0.10.21

# bundle the app
mkdir myapp
cd myapp
git clone http://github.com/myaccount/myapp
meteor bundle --directory ../deployPath
cd ../deployPath

# make sure fibers is installed, as per the README
export MONGO_URL='mongodb://127.0.0.1:27017/mydatabase'
export PORT='3000'
export ROOT_URL='http://myapp.com'

# run the site
node main.js
```

mongo .

```
mongo

> rs.initiate()
PRIMARY> rs.add("mongo-a")
PRIMARY> rs.add("mongo-b")
PRIMARY> rs.add("mongo-c")
PRIMARY> rs.setReadPref('secondaryPreferred')
```

Oplogging

oplog .

```
mongo

PRIMARY> use admin
PRIMARY>
db.addUser({user:"oplogger",pwd:"YOUR_PASSWORD",roles:[],otherDBRoles:{local:["read"]}});
PRIMARY> show users
```

Upstart Script Oplog

IP   .

```
start on started mountall
stop on shutdown

respawn
respawn limit 99 5

script
    # our example assumes you're using a replica set and/or oplog integreation
    export MONGO_URL='mongodb://mongo-a:27017,mongo-b:27017,mongo-c:27017/meteor'

    # here we configure our OPLOG URL
    export MONGO_OPLOG_URL='mongodb://oplogger:YOUR_PASSWORD@mongo-a:27017,mongo-
b:27017,mongo-c:27017/local?authSource=admin'

    # root_url and port are the other two important environment variables to set
    export ROOT_URL='http://myapp.mydomain.com'
    export PORT='80'

    exec /usr/local/bin/node /var/www/production/main.js >> /var/log/node.log 2>&1
end script
```

Sharded Mongo Oplog Tailing

: https://riptutorial.com/ko/meteor/topic/3706/-

35:

Appcache

http://www.html5rocks.com/en/tutorials/indexeddb/todo/
http://grinninggecko.com/2011/04/22/increasing-chromes-offline-application-cache-storage-limit/
http://www.html5rocks.com/en/tutorials/offline/quota-research/
https://developers.google.com/chrome/apps/docs/developers_guide?csw=1#installing
https://developers.google.com/chrome/apps/docs/developers_guide?csw=1#manifest

Examples

Meteor.status ()

Meteor . , :

```
Template.registerHelper('getOnlineStatus', function(){
  return Meteor.status().status;
});

Template.registerHelper('getOnlineColor', function(){
  if(Meteor.status().status === "connected"){
    return "green";
  }else{
    return "orange";
  }
});
```

```
  <div id="onlineStatus" class="{{getOnlineColor}}">
    {{getOnlineStatus}}
  </div>
```

```
.green{
  color: green;
}
.orange{
  color: orange;
}
```

Appcache

appcache . Appcache . mongo , .

```
meteor add appcache
```

GroundDB

.

```
meteor add ground:db
```

```
Lists = new Meteor.Collection("lists");
GroundDB(Lists);

Todos = new Meteor.Collection("todos")
GroundDB(Todos);
```

- Meteor appcache . . ! .
- appcache Chrome .
- GroundDB IronRouter .

: https://riptutorial.com/ko/meteor/topic/3375/-

36: +

React JavaScript . . Meteor React .

:

-
- Meteor + React

Examples

"Hello World"

React :

```
meteor npm install --save react react-dom react-mounter
```

React client/helloworld.jsx .

```
import React, { Component } from 'react';
import { mount } from 'react-mounter';

// This component only renders a paragraph containing "Hello World!"
class HelloWorld extends Component {
  render() {
    return <p>Hello World!</p>;
  }
}

// When the client application starts, display the component by mounting it to the DOM.
Meteor.startup(() => {
  mount(HelloWorld);
});
```

createContainer

Todos autopublish . .

```
import { createContainer } from 'meteor/react-meteor-data';
import React, { Component, PropTypes } from 'react';
import Todos from '/imports/collections/Todos';

export class List extends Component {
  render() {
    const { data } = this.props;
    return (
      <ul className="list">
        {data.map(entry => <li {...entry} />)}
      </ul>
    )
  }
}

List.propTypes = {
  data: PropTypes.array.isRequired
};
```

. .

```
export default createContainer(() => {
  return {
    data: Todos.find().fetch()
  };
}, List);
```

MongoDB

MongoDB React . .

React Meteor , `react-meteor-data` .

```
$ meteor add react-meteor-data
$ meteor npm install react-addons-pure-render-mixin
```

`both/collections.js` . `both` .

```
import { Mongo } from 'meteor/mongo';

// This collection will contain a list of random numbers
export const Numbers = new Mongo.Collection("numbers");
```

. `server/publications.js` .

```
import { Meteor } from 'meteor/meteor';
import { Numbers } from '/both/collections.js';

// This publication synchronizes the entire 'numbers' collection with every subscriber
Meteor.publish("numbers/all", function() {
  return Numbers.find();
});
```

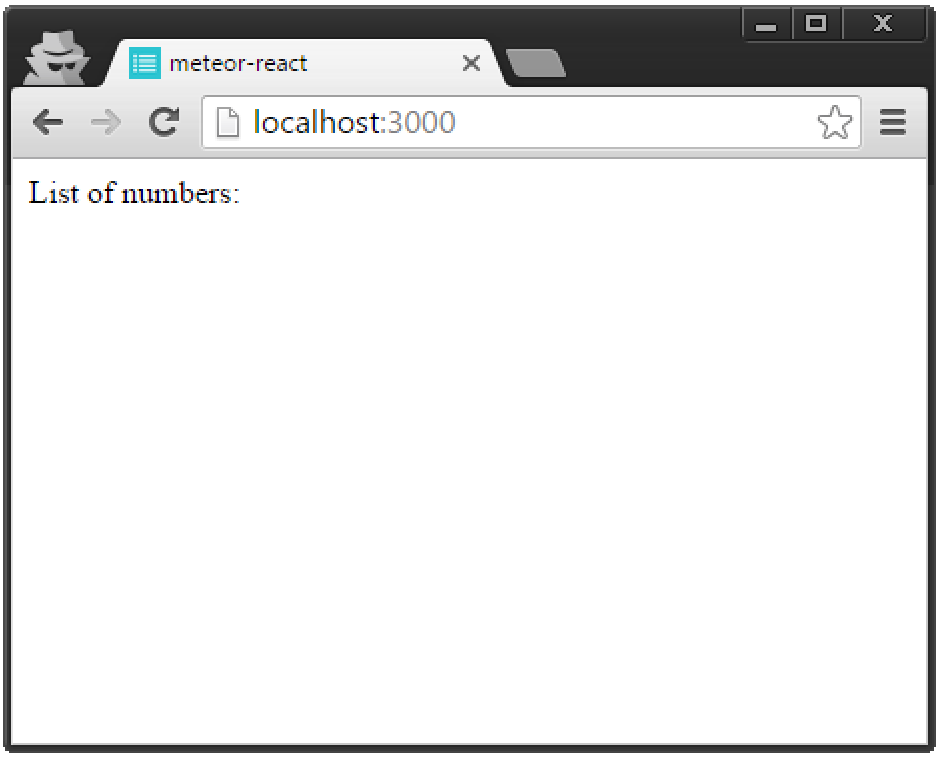
`createComponent` `Numbers` React . `client/shownumbers.jsx` :

```
import React from 'react';
import { createContainer } from 'meteor/react-meteor-data';
import { Numbers } from '/both/collections.js';

// This stateless React component renders its 'numbers' props as a list
function _ShowNumbers({numbers}) {
  return <div>List of numbers:
    <ul>
      // note, that every react element created in this mapping requires
      // a unique key - we're using the _id auto-generated by mongodb here
      {numbers.map(x => <li key={x._id}>{x.number}</li>)}
    </ul>
  </div>;
}

// Creates the 'ShowNumbers' React component. Subscribes to 'numbers/all' publication,
// and passes the contents of 'Numbers' as a React property.
export const ShowNumbers = createContainer(() => {
  Meteor.subscribe('numbers/all');
  return {
    numbers: Numbers.find().fetch(),
  };
}, _ShowNumbers);
```

.



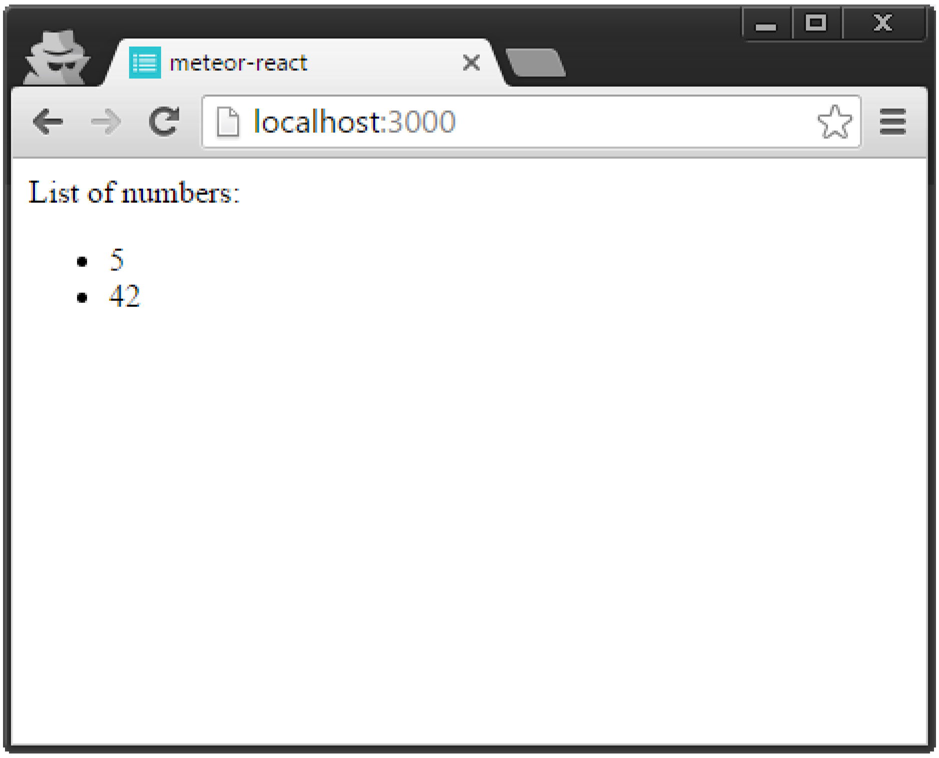
List of numbers:

MongoDB .

```
$ meteor mongo
MongoDB shell version: 3.2.6
connecting to: 127.0.0.1:3001/meteor

meteor:PRIMARY> db.numbers.insert({number: 5});
WriteResult({ "nInserted" : 1 })

meteor:PRIMARY> db.numbers.insert({number: 42});
WriteResult({ "nInserted" : 1 })
```



List of numbers:

- 5
- 42

+ : https://riptutorial.com/ko/meteor/topic/3121/-plus-

37:

Examples

Meteor . Meteor accounts .

-

accounts-password .

meteor add accounts-password .

Accounts.createUser(options, [callback]) .

options .

- username :
- email : .
- password : .
- profile : . . profile .

Meteor.Error 1 .

username email username . .

ID .

.

```
// server side
var id = Accounts.createUser({
    username: "JohnDoe",
    email: "JohnDoe@gmail.com",
    password: "TheRealJohn123",
    profile: {
        firstName: "John",
        lastName: "Doe"
    }
}, function(err) {
    console.log(err.reason);
});
```

.

. Meteor.loginWithPassword(identifier, password, [callback])

Meteor.loginWithPassword(identifier, password, [callback]) .

identifier username username , email userId . password ( ) password .

Meteor.Error .

:

```
// client side
Meteor.loginWithPassword("JohnDoe", "TheRealJohn123", function(err) {
    console.log(err.reason);
});
```

.

`Meteor.userId()` . `Meteor.userId()` `userId` `undefined` `undefined` .

`Meteor.user()` . undefined . . ID, .

`currentUser` . `Meteor.user()` . :

```
{{#if currentUser}}
    <h1>Hello there, {{currentUser.username}}!</h1>
{{else}}
    <h1>Please log in.</h1>
{{/if}}
```

.

`Meteor.logout()` .

.

`profile` . . , , , . **`profile`** . .

. `profile` `isAdmin` Meteor . . .

. .

```
// Deny all client-side updates to user documents
Meteor.users.deny({
  update() { return true; }
});
```

. Meteor diffing .

: https://riptutorial.com/ko/meteor/topic/6219/--

38:

Examples

: vulcanize

Bower (`npm install -g bower`) `bower init` . `bower.json` .

`.bowerrc` . .

```
{
  "directory": "public/bower_components"
}
```

Bower `bower_components` .

Polymer .

.

```
bower install --save PolymerElements/paper-button#^1.0.0 PolymerElements/paper-checkbox#^1.0.0
```

Vulcanize

```
Meteor add differential:vulcanize
```

config.vulcanize . .

```
{
    "polyfill": "/bower_components/webcomponentsjs/webcomponents.min.js",
    "useShadowDom": true, // optional, defaults to shady dom (polymer default)
    "imports": [
        "/bower_components/paper-button/paper-button.html",
        "/bower_components/paper-checkbox/paper-checkbox.html"
    ]
}
```

`"imports"` .

Blaze .

```
<template name="example">
    <div>
        this is a material design button: <paper-button></paper-button>
        this is a material design checkbox: <paper-checkbox></paper-checkbox>
    </div>
</template>
```

: https://riptutorial.com/ko/meteor/topic/4598/---

39:

Examples

- Codeship.com .
-
- Github Bitbucket
- "Configure Your Tests" .
    - " " " ." .
    - .

      ```
      curl -o meteor_install_script.sh https://install.meteor.com/
      chmod +x meteor_install_script.sh
      sed -i "s/type sudo >\/dev\/null 2>&1/\ false /g" meteor_install_script.sh
      ./meteor_install_script.sh
      export PATH=$PATH:~/.meteor/
      meteor --version
      meteor npm install
      ```

    - .

      ```
      npm test
      ```

- Github / Bitbucket .
- .

-
- : mocha-phantomjs :

  ```
  meteor add dispatch:mocha-phantomjs
  ```

- package.json .

  ```
  {
    "name": "awesome meteor package",
    "scripts": {
      "test": "meteor test --driver-package dispatch:mocha-phantomjs --once"
    }
  }
  ```

- `npm test` .

: https://riptutorial.com/ko/meteor/topic/6741/-----

40:

Examples

`private` .

`Assets.getText(assetPath, [asyncCallback])` . JSON `my_text_asset.json` `private` .

```
{
    "title": "Meteor Assets",
    "type": "object",
    "users": [{
        "firstName": "John",
        "lastName": "Doe"
    }, {
        "firstName": "Jane",
        "lastName": "Doe"
    }, {
        "firstName": "Matthias",
        "lastName": "Eckhart"
    }]
}
```

.

```
var myTextAsset = Assets.getText('my_text_asset.json');
var myJSON = JSON.parse(myTextAsset);
console.log(myJSON.title); // prints 'Meteor Assets' in the server's console
```

EJSON **`Assets.getBinary(assetPath, [asyncCallback])`** . `private/img` `my_image.png` .

```
var myBinaryAsset = Assets.getBinary('img/my_image.png');
```

: https://riptutorial.com/ko/meteor/topic/3379/

41: -

Examples

Meteor Electrify

HTML . !

electron , nodejs , npm , git meteor . Meteor .

```
npm install -g electrify
```

- electron ! .
- electrify . .

Electrify Meteor

```
curl https://install.meteor.com/ | sh
```

Meteor . .

- meteor JavaScript . . prototypical . .

NodeJS

```
apt-get install nodejs build-essentials
```

OS . .

- nodejs JavaScript Javascript Node.js . .

npm

npm nodejs 2 . nodejs npm -v .

- npm . . .

Electrify

( ) Meteor Todos .

:

```
apt-get install git-all
```

Git . .

- `git` . (GitHub) (BitBucket) (,) . [] [5].

```
#!/usr/bin/bash

# Change this parameter to choose where to clone the repository to.
TODOSPATH="/home/user/development/meteor-todos"

# Download the repository to the $TODOSPATH location.
git clone https://github.com/meteor/todos.git "$TODOSPATH"

# Change directory (`cd`) into the Todos project folder.
cd "$TODOSPATH"
```

TODOSPATH 'meteor-todos' . (`cd`) Electrify !

```
# It's really this simple.
electrify
```

. . . `electrify` . `sudo electrify` .

. `sudo` (,).

- : https://riptutorial.com/ko/meteor/topic/2526/----------

42:   - Mac OSX

Examples

NPM

Mac OSX Mavericks . NPM .

```
    # install node
    # as of OSX Mavericks, we need the GUI installer (?!)
    # when a good command line alternative is found, we'll post it
    http://nodejs.org/download/

    # install npm
    curl -0 -L https://npmjs.org/install.sh | sh

    # check node is installed correctly
    node --version

    # check npm is installed correctly
    npm -version

    # find your npm path
    which npm

    # make sure npm is in your path
    sudo nano ~/.profile
      export PATH=$PATH:/usr/local/bin
```

Mac OSX Mavericks . NPM .

```
# install meteor
    curl https://install.meteor.com | sh

    # check it's installed correctly
    meteor --version

    # install node
    # as of OSX Mavericks, we need the GUI installer (?!)
    # when a good command line alternative is found, we'll post it
    http://nodejs.org/download/

    # install npm
    curl -0 -L https://npmjs.org/install.sh | sh

    # check node is installed correctly
    node --version

    # check npm is installed correctly
    npm -version

    # find your npm path
    which npm

    # make sure npm is in your path
    sudo nano ~/.profile
      export PATH=$PATH:/usr/local/bin
```

Meteor Mongo, Robomongo, Atom, Linters .

```
# make sure mongo is in your local path
nano ~/.profile
  export PATH=$PATH:/usr/local/mongodb/bin

# or install it to the global path
nano /etc/paths
  /usr/local/mongodb/bin

# create mongo database directory
mkdir /data/
mkdir /data/db
chown -R username:admin /data

# run mongodb server
mongod
ctrl-c

# check that you can connect to your meteor app with stand-alone mongo
terminal-a$ meteor create helloworld
terminal-a$ cd helloworld
terminal-a$ meteor

terminal-b$ mongo -port 3001

# install robomongo database admin tool
http://robomongo.org/

# check you can connect to your mongo instance with robomongo
terminal-a$ meteor create helloworld
terminal-a$ cd helloworld
terminal-a$ meteor

Dock$ Robomongo > Create > localhost:3001
```

```
# install node-inspector
terminal-a$  npm install -g node-inspector

# start meteor
terminal-a$  cd helloworld
terminal-a$  NODE_OPTIONS='--debug-brk --debug' mrt run

# alternatively, some people report this syntax being better
terminal-a$  sudo NODE_OPTIONS='--debug' ROOT_URL=http://helloworld.com meteor --port 80

# launch node-inspector along side your running app
terminal-b$  node-inspector

# go to the URL given by node-inspector and check it's running
http://localhost:8080/debug?port=5858

# install jshint
npm install -g jshint
```

- Mac OSX : https://riptutorial.com/ko/meteor/topic/3294/----mac-osx

43: 3 API

Examples

HTTP

REST API `http` .

```
meteor add http
```

```
HTTP.get('http://foo.net/api/bar/');
```

API

HTTP . . API .

```
Foo = {
  identify: function(input){
    return Http.get('http://foo.net/api/identify/' + input);
  },
  record_action_on_item: function(firstInput, secondInput){
    return Http.put('http://foo.net/api/record_action_on_item/' + firstInput + '&' +
secondInput);
  }
}
```

Meteor Http.get (), Http.post (), Http.put () REST API . http://docs.meteor.com/#http_get

API . . . API 'request'npm . `Npm.require('request')` . https://github.com/mikeal/request

API

API . .

```
packages/foo-api-wrapper/package.js
packages/foo-api-wrapper/readme.md
packages/foo-api-wrapper/foo.api.wrapper.js
```

, `foo-api-wrapper/package.js` :

```
Package.describe({
  summary: "Atmosphere package that impliments the Foo API.",
  name: "myaccount:foo",
  version: '0.0.1'
});

Package.on_use(function (api) {
    api.export('Foo');
    api.addFiles('foo.api.wrapper.js', ["client","server"]);
});
```

`foo-api-wrapper/foo.api.wrapper.js` `Foo` API .

API

.

```
meteor add myaccount:foo
```

:

```
meteor publish myaccount:foo
```

API

.

```
Foo.identify('John');
Foo.record_action_on_item('view', "HackerNews');
```

, , URL API .

3 API : https://riptutorial.com/ko/meteor/topic/3118/-3--api-

44: (Nightwatch)

Nightwatch Meteor Acceptance End-to-End v0.5 PHP Spark to Blaze React . Continuous Integration . .

Nightwatch API
Nightwatch.js Google

Examples

Travis Meteor Continuous Integration . Nightwatch .

.travis.yml

`.travis.yml` `.travis.yml` .

```
# this travis.yml file is for the leaderboard-nightwatch example, when run standalone
language: node_js

node_js:
  - "0.10.38"

services:
  - mongodb

sudo: required

env:
  global:
    - TRAVIS=true
    - CONFIG_PREFIX=`npm config get prefix`
    - DISPLAY=:99.0
    - NODE_ENV=`travis`
  matrix:

cache:
  directories:
    - .meteor/local/build/programs/server/assets/packages
    - .meteor

before_install:
  # set up the node_modules dir, so we know where it is
  - "mkdir -p node_modules &"

  # install nightwatch, selenium, , so we can launch nightwatch and selenium
  - "meteor npm install nightwatch selenium-server-standalone-jar chromedriver"

  # fire up xvfb on port :99.0
  - "sh -e /etc/init.d/xvfb start"

  # set the xvfb screen size to 1280x1024x16
  - "/sbin/start-stop-daemon --start --quiet --pidfile /tmp/custom_xvfb_99.pid --make-pidfile
--background --exec /usr/bin/Xvfb -- :99 -ac -screen 0 1280x1024x16"

  # install meteor
  - "curl https://install.meteor.com | /bin/sh"
  # give meteor a few seconds after installing
  - "sleep 10"

  # setup Meteor app
  - "cd webapp"
  - "meteor &"

  # give Meteor some time to download packages, init data, and to start
  - "sleep 60"

# then run nightwatch using the chromedriver
script: "nightwatch -c .meteor/nightwatch.json"
```

Circle Meteorites   Continuous Integration .   .   .

-
-
-
-
-
-
- npm
-
-

.circle.yml

```
## Customize the test machine
machine:

  # Timezone
  timezone:
    America/Los_Angeles # Set the timezone

  # Add some environment variables
  environment:
    CIRCLE_ENV: test
    CXX: g++-4.8
    DISPLAY: :99.0
    NPM_PREFIX: /home/ubuntu/nvm/v0.10.33
    INITIALIZE: true
    NODE_ENV: circle


## Customize checkout
checkout:
 post:
   #- git submodule sync
   #- git submodule update --init --recursive # use submodules

general:
  build_dir: webapp
  artifacts:
    - "./tests/nightwatch/screenshots" # relative to the build directory

## Customize dependencies
dependencies:
  cache_directories:
    - "~/.meteor" # relative to the user's home directory
    - ~/nvm/v0.10.33/lib/node_modules/starrynight
    - ~/nvm/v0.10.33/bin/starrynight

  pre:
    # Install Starrynight unless it is cached
    - if [ ! -e ~/nvm/v0.10.33/bin/starrynight ]; then npm install -g starrynight; else echo "Starrynight seems to be cached"; fi;
    # Install  Meteor
    - mkdir -p ${HOME}/.meteor
    # If Meteor is already cached, do not need to build it again.
    - if [ ! -e ${HOME}/.meteor/meteor ]; then curl https://install.meteor.com | /bin/sh; else echo "Meteor seems to be cached"; fi;
    # Link the meteor executable into /usr/bin
    - sudo ln -s $HOME/.meteor/meteor /usr/bin/meteor
    # Check if the helloworld directory already exists, if it doesn't, create the helloworld app
    # The following doesn't work, because it should be checking ${HOME}/active-entry/helloworld
    # - if [ ! -e ${HOME}/helloworld ]; then meteor create --release METEOR@1.1.0.3 helloworld; else echo "helloworld app seems to be cached"; fi;

  override:
    #- meteor list

## Customize test commands
test:
  pre:
    #- starrynight fetch
    #- cd packages && rm -rf temp
    #- cd packages && ls -la
    #- starrynight autoconfig
    - meteor update --release METEOR@1.3.3
    - meteor npm install --save jquery bootstrap react react-dom react-router react-bootstrap react-komposer
    - cat .meteor/nightwatch.json
    - meteor:
        background: true
    - sleep 60
  override:
    - meteor npm run-script nightwatch


## Customize deployment commands
#deployment:
#  production:
#    branch: master
#    commands:
#      - printf "<Meteor username>\n<Meteor password>\n" | meteor deploy myapp.meteor.com

## Custom notifications
#notify:
  #webhooks:
    # A list of hashes representing hooks. Only the url field is supported.
    #- url: https://someurl.com/hooks/circle
```

SauceLabs . , . Travis, Circle BrowserStack (hwoever).

```
{
  "selenium" : {
    "start_process" : false,
    "host" : "ondemand.saucelabs.com",
    "port" : 80,
  },
  "test_settings" : {
    "chrome_saucelabs": {
      "selenium_host": "ondemand.saucelabs.com",
      "selenium_port": 80,
      "username": "${SAUCE_USERNAME}",
      "access_key": "${SAUCE_ACCESS_KEY}",
      "use_ssl": false,
      "silent": true,
      "output": true,
      "screenshots": {
        "enabled": false,
        "on_failure": true,
        "path": ""
      },
      "desiredCapabilities": {
        "name": "test-example",
        "browserName": "chrome"
      },
      "globals": {
        "myGlobal": "some_sauce_global"
      }
    },
  }
}
```

BrowserStack

BrowserStack . Selenium .

```
{
  "selenium" : {
    "start_process" : false,
    "host" : "hub.browserstack.com",
    "port" : 80,
  },

  "test_settings" : {
    "default" : {
      "launch_url" : "http://hub.browserstack.com",
      "selenium_port"  : 80,
      "selenium_host"  : "hub.browserstack.com",
      "silent": true,
      "screenshots" : {
        "enabled" : false,
        "path" : "",
      },
      "desiredCapabilities": {
        "browserName": "firefox",
        "javascriptEnabled": true,
        "acceptSslCerts": true,
        "browserstack.user": "USERNAME",
        "browserstack.key": "KEY"
      }
```

```
        }
    }
}
```

(Nightwatch) : https://riptutorial.com/ko/meteor/topic/6550/------nightwatch--

45:

. . David Weldon `PACKAGE_DIRS` .

Examples

MGP

Dispatches great Meteor Github Packages (mgp) :

```
npm install --save mgp
```

`package.json` .

```
"mgp": "mgp"
```

`git-packages.json` . () Meteor Github :

```
{
  "my:yet-another-private-package": {
    "git": "git@github.com:my/private-packages.git",
    "branch": "dev"
  }
}
```

Github repo .

Github

Codeship .

```
meteor npm run mgp
```

. Cipship Github .

- Github . .
- repo . : https://github.com/YOUR_USERNAME/REPO_UNDER_TEST/settings/keys
- SSH . : https://codeship.com/projects/PROJECT_NUMBER/configure
- SSH SSH : https://github.com/settings/keys
-

BitBucket .

: https://riptutorial.com/ko/meteor/topic/6742/-----

46:

CollectionFS . Mongo GridFS Atmosphere Meteor . StackOverflow GridFS .

Filepicker.io

Micha Roon
EventedMind

Examples

/

. . , . , .

/ . .

```
<template name="example">
  <input type=file />
</template>
```

Meteor ``startFileTransfer '' .

```
// client/example.js
Template.example.events({
  'change input': function(ev) {
    _.each(ev.srcElement.files, function(file) {
      Meteor.startFileTransfer(file, file.name);
    });
  }
});

// client/save.js
/**
 * @blob (https://developer.mozilla.org/en-US/docs/DOM/Blob)
 * @name the file's name
 * @type the file's type: binary, text (https://developer.mozilla.org/en-
US/docs/DOM/FileReader#Methods)
 *
 * TODO Support other encodings: https://developer.mozilla.org/en-
US/docs/DOM/FileReader#Methods
 * ArrayBuffer / DataURL (base64)
 */
Meteor.startFileTransfer = function(blob, name, path, type, callback) {
  var fileReader = new FileReader(),
    method, encoding = 'binary', type = type || 'binary';
  switch (type) {
    case 'text':
      // TODO Is this needed? If we're uploading content from file, yes, but if it's from an
input/textarea I think not...
      method = 'readAsText';
      encoding = 'utf8';
      break;
    case 'binary':
      method = 'readAsBinaryString';
      encoding = 'binary';
      break;
    default:
      method = 'readAsBinaryString';
      encoding = 'binary';
      break;
  }
  fileReader.onload = function(file) {
    Meteor.call('saveFileToDisk', file.srcElement.result, name, path, encoding, callback);
  }
  fileReader[method](blob);
}
```

saveFileToDisk . .

```
//
/**
 * TODO support other encodings:
 * http://stackoverflow.com/questions/7329128/how-to-write-binary-data-to-a-file-using-node-js
 */
Meteor.methods({
  saveFileToDisk: function(blob, name, path, encoding) {
    var path = cleanPath(path), fs = __meteor_bootstrap__.require('fs'),
      name = cleanName(name || 'file'), encoding = encoding || 'binary',
      chroot = Meteor.chroot || 'public';
    // Clean up the path. Remove any initial and final '/' -we prefix them-,
    // any sort of attempt to go to the parent directory '..' and any empty directories in
    // between '/////' - which may happen after removing '..'
    path = chroot + (path ? '/' + path + '/' : '/');

    // TODO Add file existance checks, etc...
    fs.writeFile(path + name, blob, encoding, function(err) {
      if (err) {
        throw (new Meteor.Error(500, 'Failed to save file.', err));
      } else {
        console.log('The file ' + name + ' (' + encoding + ') was saved to ' + path);
      }
    });

    function cleanPath(str) {
      if (str) {
        return str.replace(/\.\./g,'').replace(/\/+/g,'').
          replace(/^\/+/,'').replace(/\/+$/,'');
      }
    }
    function cleanName(str) {
      return str.replace(/\.\./g,'').replace(/\//g,'');
    }
  }
});
```

, . CSV .

Dropzone (:)

Dropzone UI REST UI REST .

Iron Router Dropzone .

```
meteor add iron:router
meteor add awatson1978:dropzone
```

dropzone URL .

```
Router.map(function () {
    this.route('uploads', {
      where: 'server',
      action: function () {
        var fs = Npm.require('fs');
        var path = Npm.require('path');
        var self = this;

        ROOT_APP_PATH = fs.realpathSync('.');

        // dropzone.js stores the uploaded file in the /tmp directory, which we access
        fs.readFile(self.request.files.file.path, function (err, data) {

          // and then write the file to the uploads directory
          fs.writeFile(ROOT_APP_PATH + "/assets/app/uploads/" +self.request.files.file.name,
data, 'binary', function (error, result) {
            if(error){
              console.error(error);
            }
            if(result){
              console.log('Success! ', result);
            }
          });
        });
      }
    });
  });
```

! Snazzy UI REST . .

Filepicker.io

. S3, Azure, Rackspace Dropbox Filepicker.io . Filerpicker .

```
meteor add mrt:filepicker
```

Filepicker 3 . Filepicker . Meteor data- * .

```
<input type="filepicker"
  id="filepickerAttachment"
  data-fp-button-class="btn filepickerAttachment"
  data-fp-button-text="Add image"
  data-fp-mimetypes="image/*"
  data-fp-container="modal"
  data-fp-maxsize="5000000"
  data-fp-services="COMPUTER,IMAGE_SEARCH,URL,DROPBOX,GITHUB,GOOGLE_DRIVE,GMAIL">
```

API , , , .

```
if(Meteor.isClient){
  Meteor.startup(function() {
    filepicker.setKey("YourFilepickerApiKey");
  });
  Template.yourTemplate.rendered = function(){
    filepicker.constructWidget($("#filepickerAttachment"));
  }
  Template.yourTemplate.events({
  'change #filepickerAttachment': function (evt) {
    console.log("Event: ", evt, evt.fpfile, "Generated image url:", evt.fpfile.url);
  });
});
```

CollectionFS

Mongo GridFS . CollectionFS .

.

```
meteor add cfs:standard-packages
meteor add cfs:filesystem
```

.

```
<template name="yourTemplate">
    <input class="your-upload-class" type="file">
</template>
```

, .

```
Template.yourTemplate.events({
    'change .your-upload-class': function(event, template) {
        FS.Utility.eachFile(event, function(file) {
            var yourFile = new FS.File(file);
            yourFile.creatorId = Meteor.userId(); // add custom data
            YourFileCollection.insert(yourFile, function (err, fileObj) {
                if (!err) {
                   // do callback stuff
                }
            });
        });
    }
});
```

.

```
YourFileCollection = new FS.Collection("yourFileCollection", {
    stores: [new FS.Store.FileSystem("yourFileCollection", {path: "~/meteor_uploads"})]
});
YourFileCollection.allow({
    insert: function (userId, doc) {
        return !!userId;
    },
    update: function (userId, doc) {
        return doc.creatorId == userId
    },
    download: function (userId, doc) {
        return doc.creatorId == userId
    }
});
```

Raz . CollectionFS CollectionFS .

. .

```
//https://forums.meteor.com/t/read-file-from-the-public-folder/4910/5

// Asynchronous Method.
Meteor.startup(function () {
    console.log('starting up');

    var fs = Npm.require('fs');
    // file originally saved as public/data/taxa.csv
    fs.readFile(process.cwd() + '/../web.browser/app/data/taxa.csv', 'utf8', function (err,
data) {
        if (err) {
            console.log('Error: ' + err);
            return;
        }

        data = JSON.parse(data);
        console.log(data);
    });
});


// Synchronous Method.
Meteor.startup(function () {
    var fs = Npm.require('fs');
    // file originally saved as public/data/taxa.csv
    var data = fs.readFileSync(process.cwd() + '/../web.browser/app/data/taxa.csv', 'utf8');

    if (Icd10.find().count() === 0) {
        Icd10.insert({
            date:  new Date(),
            data:  JSON.parse(data)
        });
    }
});


Meteor.methods({
  parseCsvFile:function (){
    console.log('parseCsvFile');

    var fs = Npm.require('fs');
    // file originally saved as public/data/taxa.csv
    var data = fs.readFileSync(process.cwd() + '/../web.browser/app/data/taxa.csv', 'utf8');
    console.log('data', data);
  }
});
```

: https://riptutorial.com/ko/meteor/topic/3119/--

47:

Examples

"settings.json" .

```
if(Meteor.isServer){
  Meteor.startup(function()){
    // this needs to be run on the server
    var environment, settings;

    environment = process.env.METEOR_ENV || "development";

    settings = {
      development: {
        public: {
          package: {
            name: "jquery-datatables",
            description: "Sort, page, and filter millions of records. Reactively.",
            owner: "LumaPictures",
            repo: "meteor-jquery-datatables"
          }
        },
        private: {}
      },
      staging: {
        public: {},
        private: {}
      },
      production: {
        public: {},
        private: {}
      }
    };

    if (!process.env.METEOR_SETTINGS) {
      console.log("No METEOR_SETTINGS passed in, using locally defined settings.");
      if (environment === "production") {
        Meteor.settings = settings.production;
      } else if (environment === "staging") {
        Meteor.settings = settings.staging;
      } else {
        Meteor.settings = settings.development;
      }
      console.log("Using [ " + environment + " ] Meteor.settings");
    }
  });
}
```

METEOR_SETTINGS ()

METEOR_SETTINGS JSON `Meteor.settings` . `settings.json` .

```
{
  "public":{
    "ga":{
      "account":"UA-XXXXXXX-1"
    }
  }
}
```

.

```
# run your app in local development mode with a settings file
meteor --settings settings.json

# or bundle and prepare it as if you're running in production
# and specify a settings file
meteor bundle --directory /path/to/output
cd /path/to/output
MONGO_URL="mongodb://127.0.0.1:27017" PORT=3000 METEOR_SETTINGS=$(cat /path/to/settings.json)
node main.js
```

Meteor.settings .

```
Meteor.startup(function(){
  if(Meteor.isClient){
    console.log('Google Analytics Account', Meteor.settings.public.ga.account);
  }
});
```

`process.env` .

```
if (Meteor.isServer) {
  Meteor.startup(function () {
    // detect environment by getting the root url of the application
    console.log(JSON.stringify(process.env.ROOT_URL));

    // or by getting the port
    console.log(JSON.stringify(process.env.PORT));

    // alternatively, we can inspect the entire process environment
    console.log(JSON.stringify(process.env));
  });
}
```

Meteor

. .

```
//-------------------------------------------------------------------------------------------
----------
// server/server.js
// we set up a getEnvironment method

Meteor.methods({
  getEnvironment: function(){
    if(process.env.ROOT_URL == "http://localhost:3000"){
        return "development";
    }else{
        return "staging";
    }
  }
 });

//--------------------------------------------------------------------------------------------
----------
// client/main.js
// and then call it from the client

Meteor.call("getEnvironment", function (result) {
  console.log("Your application is running in the " + result + "environment.");
});
```

NODE_ENV

Meteor 1.3 Meteor `NODE_ENV` .

```
if (Meteor.isClient) {
  Meteor.startup(function () {
    if(process.env.NODE_ENV === "testing"){
      console.log("In testing...");
    }
    if(process.env.NODE_ENV === "production"){
      console.log("In production...");
    }
  });
}
```

: https://riptutorial.com/ko/meteor/topic/4198/-

48:

	Meteor .
MONGO_URL	Mongo URL.
ROOT_URL	...
OPLOG_URL	...
MONGO_OPLOG_URL	...
METEOR_ENV	...
NODE_ENV	...
NODE_OPTIONS	...
DISABLE_WEBSOCKETS	...
MAIL_URL	...
DDP_DEFAULT_CONNECTION_URL	...
HTTP_PROXY	...
HTTPS_PROXY	...
METEOR_OFFLINE_CATALOG	...
METEOR_PROFILE	...
METEOR_DEBUG_BUILD	...
TINYTEST_FILTER	...
MOBILE_ROOT_URL	...
NODE_DEBUG	...
BIND_IP	...
PACKAGE_DIRS	...
	...
METEOR_PRINT_CONSTRAINT_SOLVER_INPUT	...
METEOR_CATALOG_COMPRESS_RPCS	...

METEOR_MINIFY_LEGACY	...
METEOR_DEBUG_SQL	...
METEOR_WAREHOUSE_DIR	...
AUTOUPDATE_VERSION	...
USE_GLOBAL_ADK	...
METEOR_AVD	...
DEFAULT_AVD_NAME	...
METEOR_BUILD_FARM_URL	...
METEOR_PACKAGE_SERVER_URL	...
METEOR_PACKAGE_STATS_SERVER_URL	...
DEPLOY_HOSTNAME	...
METEOR_SESSION_FILE	...
METEOR_PROGRESS_DEBUG	...
METEOR_PRETTY_OUTPUT	...
APP_ID	...
AUTOUPDATE_VERSION	...
CONSTRAINT_SOLVER_BENCHMARK	...
DDP_DEFAULT_CONNECTION_URL	...
SERVER_WEBSOCKET_COMPRESSION	...
USE_JSESSIONID	...
METEOR_PKG_SPIDERABLE_PHANTOMJS_ARGS	...
WRITE_RUNNER_JS	...
TINYTEST_FILTER	...
METEOR_PARENT_PID	...
METEOR_TOOL_PATH	...
RUN_ONCE_OUTCOME	...

TREE_HASH_DEBUG	...
METEOR_DEBUG_SPRINGBOARD	...
METEOR_TEST_FAIL_RELEASE_DOWNLOAD	...
METEOR_CATALOG_COMPRESS_RPCS	...
METEOR_TEST_LATEST_RELEASE	...
METEOR_WATCH_POLLING_INTERVAL_MS	...
EMACS	...
METEOR_PACKAGE_STATS_TEST_OUTPUT	...
METEOR_TEST_TMP	...

Examples

Meteor

.

```
PORT=4000 meteor
NODE_ENV="staging" meteor
```

Meteor SMTP

Gmail

```
MAIL_URL=smtp://username%40gmail.com:password@smtp.gmail.com:465/
```

: 2000 . https://support.google.com/a/answer/176600?hl=ko .

: https://riptutorial.com/ko/meteor/topic/3154/-

S. No		Contributors
1		Ankit, Christian Fritz, Community, Gal Dreiman, ghybs, grahan, hwillson, João Rodrigues, levon, Matthias Eckhart, mav, mertyildiran, Ray, reoh, robfallows, Tom Coleman, Zoltan Olah
2	AWS EC2 Meteor 1.4	AGdev
3	Codeship Galaxy	schmidsi
4	ES2015 ( )	reoh
5	ESLint	saimeunt
6	Meteor + React + ReactRouter	rafahoro
7	Meteor.call	Ramil Muratov, Rolljee, Sacha
8	Meteor	AbigailW, Serkan Durusoy, Tom Coleman
9	Mongo	AbigailW, distalx, RamenChef, TechplexEngineer
10	Mongo	AbigailW
11	MongoDB	distalx, Dranithix, hwillson, Matthias Eckhart, robfallows, Thomas Gerot
12	MongoDB	AbigailW, levon
13	Nightwatch -	AbigailW
14	Upstart	AbigailW, ghybs
15	Windows Meteor	Tom Coleman
16		AbigailW, Ankit, Ankit Balyan, Fermuch, Ilya Lyamkin
17	/ NPM	hwillson
18		Abdelrahman Elkady, AbigailW, Chris Pena, corvid, Dair, dangsonbk, Eliezer Steinbock, Faysal Ahmed, ghybs, j6m8, Maciek, RamenChef, Ramil Muratov, robfallows, Serkan Durusoy
19	.	Dranithix

20		AbigailW, anomepani, ghybs, Michael Balmes, Nick Carson, Phe0nix, reoh, Thomas Gerot
21		AbigailW, distalx
22		Ankit, ghybs, Luna, Michael Balmes
23		AbigailW
24		AbigailW, Anis D, Antti Haapala, ghybs
25		AbigailW
26	(Vars & Dictionaries)	Ankit
27		Filipe Névola
28		AbigailW
29		AbigailW, Anis D
30	(, jQuery)	AbigailW, Anis D
31		Dan Cramer, distalx, ghybs, jordanwillis, khem poudel, RamenChef, robfallows, Thomas Gerot
32		AbigailW, RamenChef, reoh
33	(Nightwatch)	AbigailW
34		AbigailW
35		AbigailW
36	+	AbigailW, aedm, corvid, ghybs, RamenChef, Teagan Atwater, zliw
37		Barry Michael Doyle, KrisVos130
38		Thaum Rystra
39		schmidsi
40		Matthias Eckhart
41	-	AbigailW, JuanGesino, Nick Bull, RamenChef
42	- Mac OSX	AbigailW, RamenChef
43	3 API	AbigailW
44	(Nightwatch)	4444, AbigailW

45		schmidsi
46		AbigailW
47		AbigailW, ghybs
48		AbigailW, hcvst