http://jxck.node-ninja.com

Node.js for Beginner

with Live Coding :)

This slide is Beat optimized for

Chrome & FireFox

should use them ;)

Jack

about me

Jxck

- id: **Jxck**
- group: #nodejs_jp
- twitter: Jxck_
- blog: http://d.hatena.ne.jp/Jxck
- tumblr: http://jxck.tumblr.com/
- love: music
- writing node.js book now

Todays Goal

very beginning about node.js

- developing application
- for very begginer
- with express & socket.io
- in **Live & Realtime** coding
- everythigs on Node.js :)

Dive into Node.js

Get Start Noding

`Environment & Package`

Language	**env**	**package**
Ruby	rvm	gem
Python	Virtualenv	PyPI
Perl	Perlbrew	CPAN
Node.js	nvm / nave	npm

```
require('express')
```

What is Express?

Web Application Framework by TJ

- inspired by Sinatra
- very lightweight
- build on Connect
- standard in node

official http://expressjs.com

Install Express & Sckeleton

```
Jxck$ npm install express -g
/* outputs */
Jxck$ express sample
/* outputs */
Jxck$ tree sample
/* outputs */
Jxck$ cd sample
/* outputs */
Jxck$ npm install
/* outputs */
Jxck$ node app.js
/* outputs */
```

```
// Module dependencies.
var express = require('express')
  , routes = require('./routes')

var app = module.exports = express.createServer();

// Configuration
app.configure(function(){
  app.set('views', __dirname + '/views');
  app.set('view engine', 'jade');
  app.use(express.favicon());
  app.use(express.bodyParser());
  app.use(express.methodOverride());
  app.use(app.router);
  app.use(express.static(__dirname + '/public'));
});

app.configure('development', function(){
  app.use(express.errorHandler({ dumpExceptions: true, showStack: true }));
});

app.configure('production', function(){
  app.use(express.errorHandler());
});

// Routes
app.get('/', routes.index);
app.get('/foo', routes.foo);

app.listen(3000);
console.log("Express server listening on port %d in %s mode", app.address().port, app.settings.env);
```

```
// routes
exports.index = function(req, res){
  res.render('index', { title: 'Express' })
};

exports.foo = function(req, res){
  res.render('index', { title: 'Foo' })
};
```

nodefest2011.node-ninja.com

sample routing

resource "Users"

```
// User Data
var users = [{username: 'Jxck', email:
'sample@mail.com'}];

app.get('/users', function(req, res) {
  res.render('users/index', {
    title: 'index',
    users: users
  });
});

app.get('/users/:id', function(req, res) {
  res.render('users/show', {
    title: 'show',
    id: req.param('id'),
    users: users[id]
  });
});

app.get('/users/:id/edit', function(req, res) {
  res.render('users/edit', {
    title: 'edit',
    id: req.param('id'),
    users: users[id]
  });
});
```

```
app.put('/users/:id', function(req, res) {
  users[req.param('id')] = {
    username: req.param('username'),
    email: req.param('email')
  };
  res.redirect('/users/');
});

app.get('/users/new', function(req, res) {
  res.render('users/new', {
    title: 'new'
  });
});

app.post('/users', function(req, res) {
  users.push({
    username: req.param('username'),
    email: req.param('email')
  });
  res.redirect('/users/');
});

app.delete('/users/:id', function(req, res) {
  users.splice(req.param('id'), 1);
  res.redirect('/users/');
});
```

require('socket.io')

What's Socket.IO

Module for Realtime Application by Guillermo

- Yes **WebSocket** but not only.
- support almost all browser
- very easy to use

you can see more from **Author** !!

official http://socket.io translated http://jxck.github.com/socket.io/

layout.jade & index.jade

```
// layout.jade
!!!
html
  head
    title= title
    link(rel="stylesheet",
href="/stylesheets/style.css")
    script(src="http://code.jquery.com/jquery-
1.6.4.js")
    script(src="/socket.io/socket.io.js")
    script(src="/javascripts/client.js")
  body!= body
```

```
// index.jade
h1= title
p Welcome to #{title}
p
  input(id="msg", type="text", name="msg")
  input(id="ok", type="button", value="ok")
ul#display
```

sever.js & client.js

```
// server.js
var app = require('./app'),
    io = require('socket.io');

io.sockets.on('connect', function(socket) {
  socket.on('msg send', function(data) {
    socket.emit('msg push', data);
    socket.broadcast.emit('msg push', data);
  });
});
```

```
// Client
var socket = io.connect();

$(function() {
  var $msg = $('#msg')
  ,   $ok = $('#ok')
  ,   $display = $('#display')
  ;

  $ok.click(function() {
    socket.emit('msg send', $msg.val());
  });

  socket.on('msg push', function(data) {
    var $li = $('<li>').text(data);
    $display.append($li);
  });
});
```

explanation of API http://bit.ly/socketio_API

DataBase ?

Almost of All you want

- MongoDB
- CouchDB
- Redis
- MySQL
- PostgreSQL
- etc

how to test?

`lots of testing framework`

- Jasmine
- NodeUnit
- Expresso
- Vows
- etc

Talk about test http://slidesha.re/test_it_nodejs

Utility

`supports your develop`

- node-inspector
- node-dev
- jake/cake
- etc

In production

`You may use in future`

- forever
- Cluster
- node-http-proxy
- node-redis
- hook.io
- etc

node.js modules http://toolbox.no.de/

Hosting ?

Node on the Cloud

- Joyent
- Heroku
- Cloud Foundary
- DotCloud
- **NEW** Node-Ninja
- etc

RVP-pattern http://bit.ly/nodejitsu_rvp

DIRT architecture http://bit.ly/node_digs_dirt

Node.js Book(japanese)

comming soon ;)

anyone.on

('uncaughtQuestion?');

Question

Node.js と相性のいいストレージ層についてどう考えるか？(How do you think about Strage-Tire of Node.js ?)

Node では、基本的な DB サーバのクライアントは大体あります。しかし、スケーラビリティやデータ形式の関係から MongoDB は相性がいいと思います。(Node.js supports almost of all famous DataBase client library. but I think MongoDB is very get along with Node.js because of Scalability & Data Format)

また、リアルタイムアプリではとにかく速度が求められることが多いので、 Redis のような高速な DB もセッションストレージなどでよく使われています。(And Realtime Application needs "SPEED" anyway, So Redis often used in like SessionStrage Engine.)

May the Node be with you

Jxck.on('end');

```
// thank you ;)
```

review - send the msg

(at conference, everyone wrote `"socket.emit('review', 'something')"` to browser-console and it broadcasts to everyone)

- they
- pushed
- 'hoge'
- or
- 'XXXX'
- like
- this :p