## Intro

Inspired by Sinatra (alternative to Ruby on Rails)

Built on top of connect middleware, which is itself built on top of native node http.

Enhances connect

Request / Response enhancements

Routing

View support

HTML Helpers

Content Negotation

But makes connect exposable

You don’t lose any connect functionality.

Makes for a better development experience than Connect

### Architecture

Request processed by routing mechanism

Routing mechanism determines a function that maps to the request

Routing function processes the request and generates response

Express has no concept of a controller

In traditional MVC, requests map to controllers, which are just a collection of actions.

Express is like a MVR (Model/View/Router)

TODO: Elaborate

As opposed to http module, express does all the work of parsing the url and mapping it to a route.

## Routing

Route - mapping of a app endpoints (url’s) to actions.

The ability to register all your actions in a single file can get ugly real quick.

To approximate the concept of a controller, you can organize all actions related to a certain class of functionality into a separate file, then expose the actions via a module.

The file ends up being similar to a controller, with the routes acting as a dispatcher to actions contained in each “controller” file.

Parameters

Parameters can be represented in several ways

* Part of path
* Query parameter

test.com/foo/bar/1

test.com/foo?bar=1

test.com/foo/bar/100..300

regex

## Configuration

Express makes it easy to configure your application at initialization.

App.set

Store a value under a key

App.get

app.configure(‘dev’, function() {

});

Env param is set based on an environment variable NODE\_ENV

## Enhancing with Middleware

Express wraps connect and exposes functionality as it’s own.