

Aaron Feng

@aaronfeng

4/09/11

Philly Code
Camp



About me

- 📌 Principal Software Engineer at Algorithmics
- 📌 Organizer of: PhillyLambda.org, PhillyAWS.com, and Philly Startup Hackathons
- 📌 Opensource lover



Windows users!
No Visual Studio
.NET not required
Lots of Command Line

source: <http://www.flickr.com/photos/zigazou76/3702501888/>



source: <http://www.flickr.com/photos/53370644@N06/4976494944/>

Get running with Manos

Saturday, April 9, 2011

K.I.S.S.

*Keep
It
Simple
Stupid*

source: <http://www.flickr.com/photos/bjornmeansbear/4773836932/>

Inspired
by

Facebook Tornado

Ways to Handle Requests

- 📌 Synchronous – Simple
- 📌 Process – Harder
- 📌 Threads – WTF

C10K

<http://www.kegel.com/c10k.html>

Single Thread FTW

The Cost of I/O (In Cycles)

L1-Cache	-	3		
L2-Cache	-	14		
RAM	-	250		
Disk	-	41	000	000
Network	-	240	000	000

<http://blog.mixu.net/2011/02/01/understanding-the-node-js-event-loop/>

Streaming



libev

<http://software.schmorp.de/pkg/libev>

libeio

<http://lists.schmorp.de/pipermail/libev/2008q2/000293.html>

Evented I/O for **V8 JavaScript**.

An example of a web server written in Node which responds with "Hello World" for every request.

```
var http = require('http');
http.createServer(function (req, res) {
  res.writeHead(200, {'Content-Type': 'text/plain'});
  res.end('Hello World\n');
}).listen(8124, "127.0.0.1");
console.log('Server running at http://127.0.0.1:8124/');
```

To run the server, put the code into a file `example.js` and execute it with the node program:



GET ASYNC OR GET SUNK

A close-up photograph of a man with a shaved head and glasses, wearing an orange safety vest over a grey shirt. He is drinking from a water bottle, with water spraying out of his mouth. The background is blurred, showing what appears to be a construction site.

IIS
Apache
etc...

source: <http://www.flickr.com/photos/illupino/4692329185/>

Asynchronous application hosting environment



source: <http://www.flickr.com/photos/kwl/4116389731/>

I/O is Evented



source: <http://www.flickr.com/photos/53067724@N00/3434393645/>

Use what you want with Manos

Demo



source: <http://www.flickr.com/photos/acaben/541420967/>

Compile Manos

- Best on: Linux or OSX (right now)
- Install Mono 2.8
- `git clone https://github.com/jacksonh/manos`
- Read [/docs/installation.md](#)

Building Manos App

- 📌 `manos --init foo`
- 📌 `manos --build foo.cs bar.cs`
- 📌 `manos --server foo.dll`
- 📌 `manos --docs`

Routes

```
public HelloWorld () {  
    Route ("/Index", ctx => {  
        ctx.Response.End();  
    });  
}
```

```
[Route ("/", "/Home", "/Index")]  
public void Index (IManosContext ctx) {  
    ctx.Response.End();  
}
```

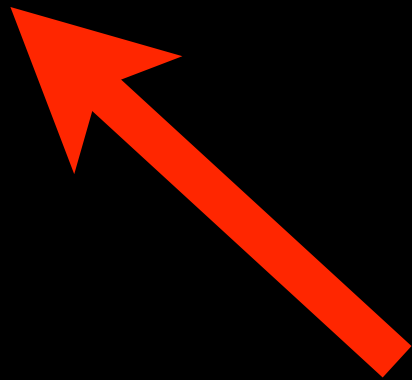
* Implicit routing

Manos Rendering Views

```
ctx.Response.End (“<html>...</html>”);  
  
ctx.Response.SendFile("../index.html");  
  
ctx.Response.Write("...");
```

MVC Rendering Views

```
return this.View (“Index”, “...”);
```



Manos Forms

```
<input type='text' name='link' />  
<input type='checkbox' name='show_info' />
```

```
[Post ("/submit-link")]  
public void SubmitLink (IManosContext ctx, Shorty  
app, string link, bool show_info) {  
}
```

MVC Forms

```
<input type='text' name='link' />  
<input type='checkbox' name='show_info' />
```

```
[AcceptVerbs("HttpVerbs.Post")]  
public ActionResult SubmitLink (string link,  
                                bool show_info) {  
    // return ...;  
}
```

Manos Concepts

 Modules

 Pipes

 Caching

 Timeouts

ManosModule

```
public class StaticContentModule : ManosModule {  
    public StaticContentModule () {  
        Get (".*", Content);  
    }  
}
```

```
public class MyApp : ManosApp {  
    Route ("/Content/",  
        new StaticContentModule ());  
}
```

ManosPipe

- OnPreProcessRequest - before anything (no routing)
- OnPreProcessTarget - target lookup (after routing)
- OnPostProcessTarget - target found
- OnPostProcessRequest - after Manos action
- OnProcessError - used for redirect to error

Caching

```
Cache.Set ("Foo", new Bar ());  
Cache.Set ("Foo",  
    new Blah ("..."),  
    (key, value) => {  
        ctx.Response.Redirect ("..");  
    })  
);  
Cache.Get ("Foo", (key, value) => {  
    // ....  
});
```


Timeouts

```
MyManosApp () {  
  AddTimeout (TimeSpan.FromMinutes (5),  
    app => app.Cache.Clear ());  
}
```

```
MyManosApp () {  
  AddTimeout (TimeSpan.FromMinutes (5),  
    RepeatBehavior.Forever,  
    app => app.Cache.Clear ());  
}
```

Use Cases

- Fast API

- Embedded Daemon

- Streaming via Websockets

Manos in the wild



<http://apch.fr/>



<http://froxy.manosdemono.org/>

getting started is **easy**

sign up for an account

create an application

push your code

```
$ git remote add git@deployfu.com:myapp
```

```
$ git push deployfu master
```

view your page

<http://myapp.deployfu.com>

DeployFu Makes It Easy

Deploy your web apps in the cloud. Scale them w

DeployFu is a hosting service for **node.js**, **Manos**, **AS**

Pylons. Want to get started on DeployFu?

Sign Up



The
ctx.Reponse.
End(“Thanks
for coming”)

Questions?

@aaronfeng

aaron.feng@gmail.com

github.com/aaronfeng