Contact + Source

- Blog: www.raymondcamden.com
- Email: raymondcamden@gmail.com
- Twitter: raymondcamden
- Source (slides + code): https://github.com/cfjedimaster/webtask-serverless-preso

Serverless with Webtask

Section 1 – Defining Serverless

What it is…

- A term very much in flux!
- A simpler, more effective way to build functionality

What it is…

"Serverless computing refers to a model where the existence of servers is **simply hidden from developers**. I.e. that even though servers still exist **developers are relieved** from the need to care about their operation. **They are relieved** from the need to worry about low-level infrastructural and operational details such as scalability, high-availability, infrastructure-security, and so forth."

Credit: https://developer.ibm.com/openwhisk/what-is-serverless-computing/

What it is…

- Node PaaS services
- Database as a Service
- Static site services

“Function as a Service”

- I write a function and it does one thing
- The actual hosting of the function, the making it available, is handled by the service
- I call my function
- “Details”

What you get…

- Small, atomic services
- Composition (typically)
- Cheaper costs (pay when used)

What it isn’t…

- The answer to everything!
- Easier (always)
- Cheaper (always)

Where does this make sense?

- You're building simple, small services that do one thing
- Usage isn't 24/7
- Fine grained details aren't important

Where doesn’t it make sense…

- You’re building a server!
- You have near constant usage
- You end up with huge “functions” thousands of lines long

“It Depends”

An Example (This is what excites me!)

```
var express = require('express');
var app = express();


app.set('port', process.env.PORT || 3000);

app.get('/cats', function(req, res) {
    // talk to some DB or whatevs and get a list of cats
    // then return it, what follows is pseudo-code:

    dbServer.getCats().then(function(cats) {
        res.send(cats);
    });
});

app.listen(app.get('port'), function() {
    console.log('Express running on http://localhost:' +
    app.get('port'));
});
```

An Example (Again – kinda fake!)

```
dbServer.getCats().then(function(cats) {
    send(cats);
});
```

Glenn
@GSto

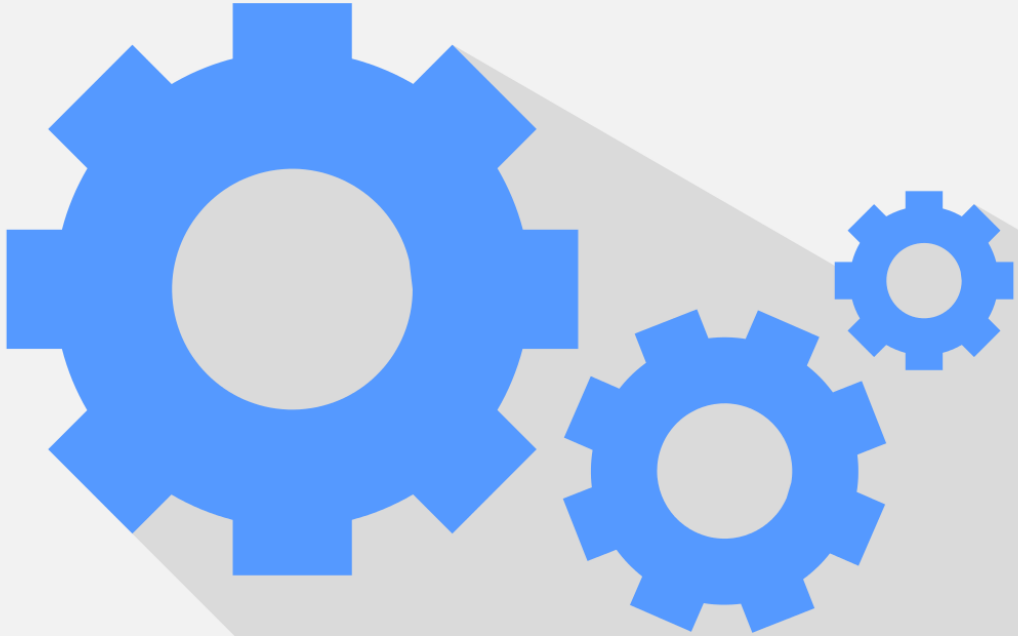
Follow

If Amazon opens a new restaurant where robots automatically bring your food to you, would it be serverless?

8:47 AM - 14 Sep 2018

2

Options…



Apache OpenWhisk

- Open source
- Multi-language platform (Node, Swift, Java, Go, Scala, Python, PHP, Docker)
- Significant supporters (Adobe, IBM, Red Hat, and more…)
- openwhisk.apache.org

IBM Cloud Functions

- Apache OpenWhisk
- UI for editing, monitoring, etc
- Visual composer for building complex serverless apps
- https://console.bluemix.net/openwhisk/
- https://www.raymondcamden.com/tags/openwhisk/

OpenWhisk/IBM Cloud Functions

- Pros:
  - Relatively simple programming interface
  - Easy to get a URL
  - Free
- Cons:
  - Initial setup is a bit of work

Azure Functions

- Part of Azure
- Feature rich
- Multi-language: JavaScript, C#, F#, JavaScript
  - Experimental: Bat, Python, PowerShell, TypeScript
- UI for editing, CLI too (somewhat awkward)
- https://azure.microsoft.com/en-us/services/functions/
- https://www.raymondcamden.com/tags/azure/

Azure Functions

- Pros
  - Good integration with Azure itself
  - Good Visual Studio Code integration
- Cons
  - A bit tricky to setup locally
  - Not 100% free to test

Google Cloud Functions

- Part of Google Cloud
- Node 6, 8, Python
- Online console and CLI
- https://cloud.google.com/functions/

Amazon Lambda

- I bet you've heard of this one…
- Incredibly powerful
- Incredibly complex
- https://aws.amazon.com/lambda/

Serverless Platform

- Not another option!
- A framework on top of serverless called… Serverless
- CLI for scaffolding, deploying, testing
- Yaml file for configuration, deployment, etc
- https://serverless.com

Serverless Platform

- AWS QuickStart
- Guide
- CLI Reference
- Events
- Examples

- Azure QuickStart
- Guide
- CLI Reference
- Events
- Examples

- OpenWhisk QuickStart
- Guide
- CLI Reference
- Events
- Examples

- Google CF QuickStart
- Guide
- CLI Reference
- Events
- Examples

- Kubeless QuickStart
- Guide
- CLI Reference
- Events

- Spotinst QuickStart
- Guide
- CLI Reference
- Events
- Examples

- Webtasks QuickStart
- Guide
- CLI Reference
- Events

- Fn QuickStart
- Guide
- CLI Reference
- Events

- Cloudflare Workers QuickStart
- Guide
- CLI Reference
- Events

More…

- “Targeted” Serverless Features
  - Twilio Functions
  - MongoDB Stitch
- Extend (goextend.io)

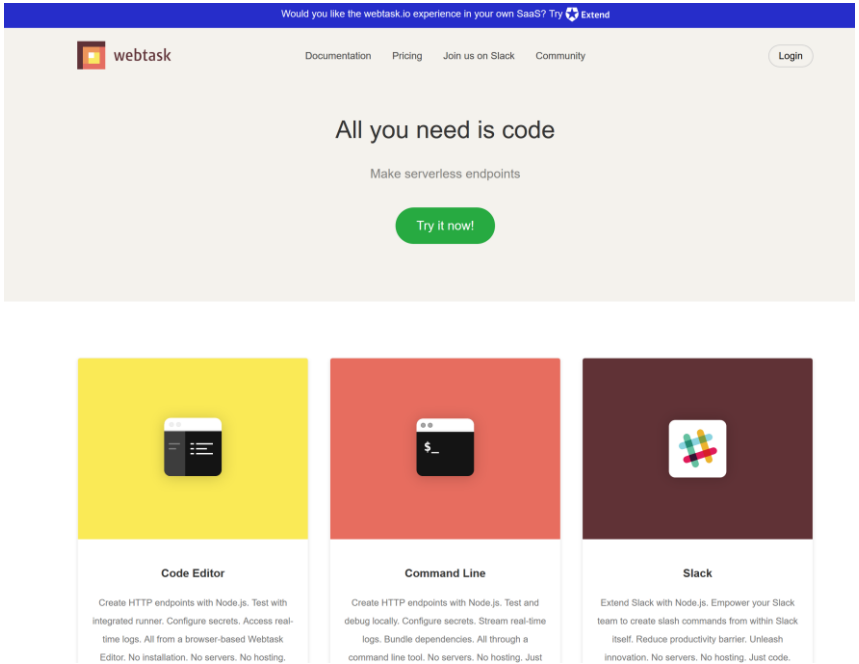
Webtask!

- Next section! ;)

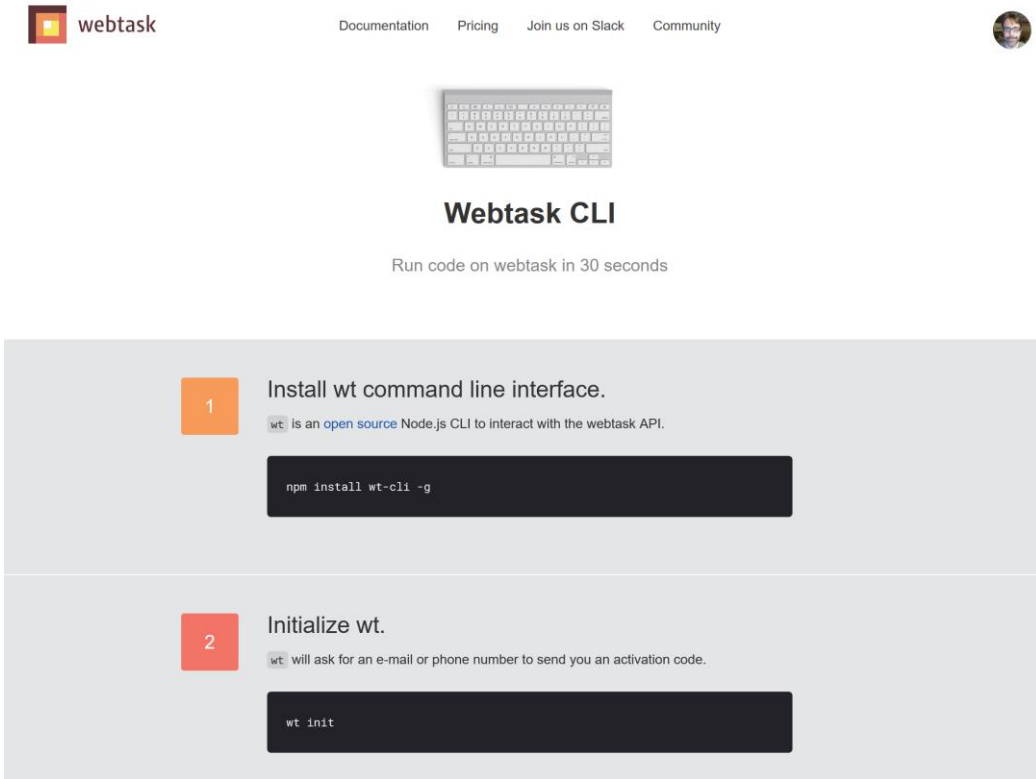
Section 2 – Using Webtask

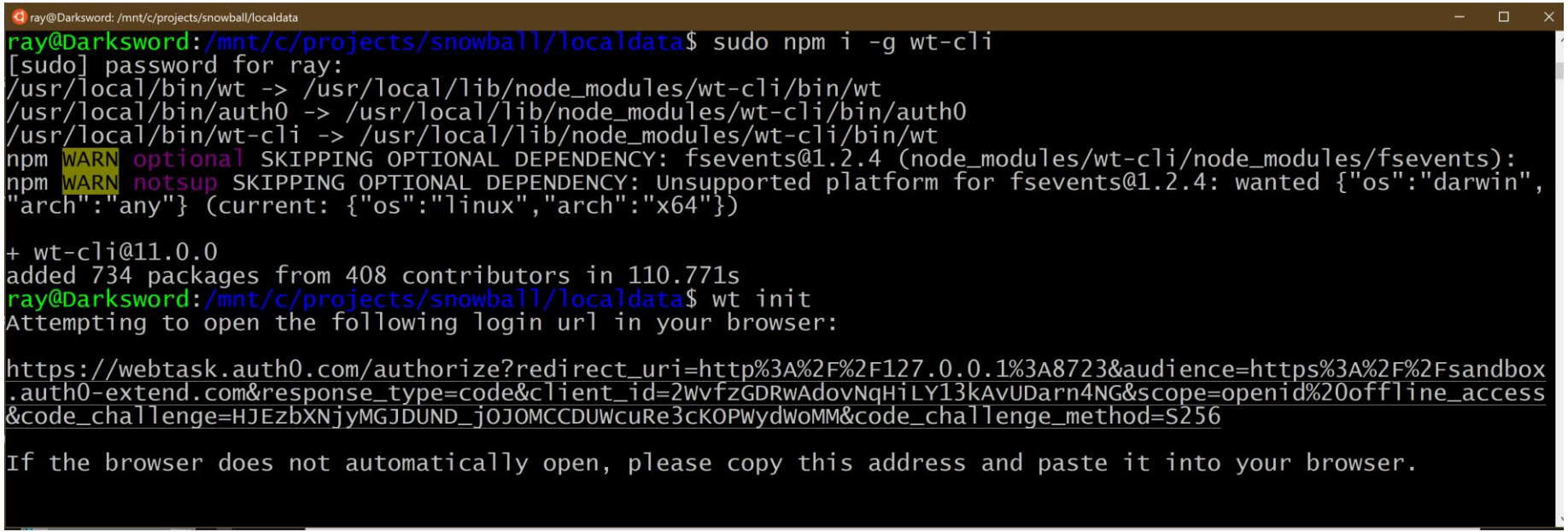
Signing Up

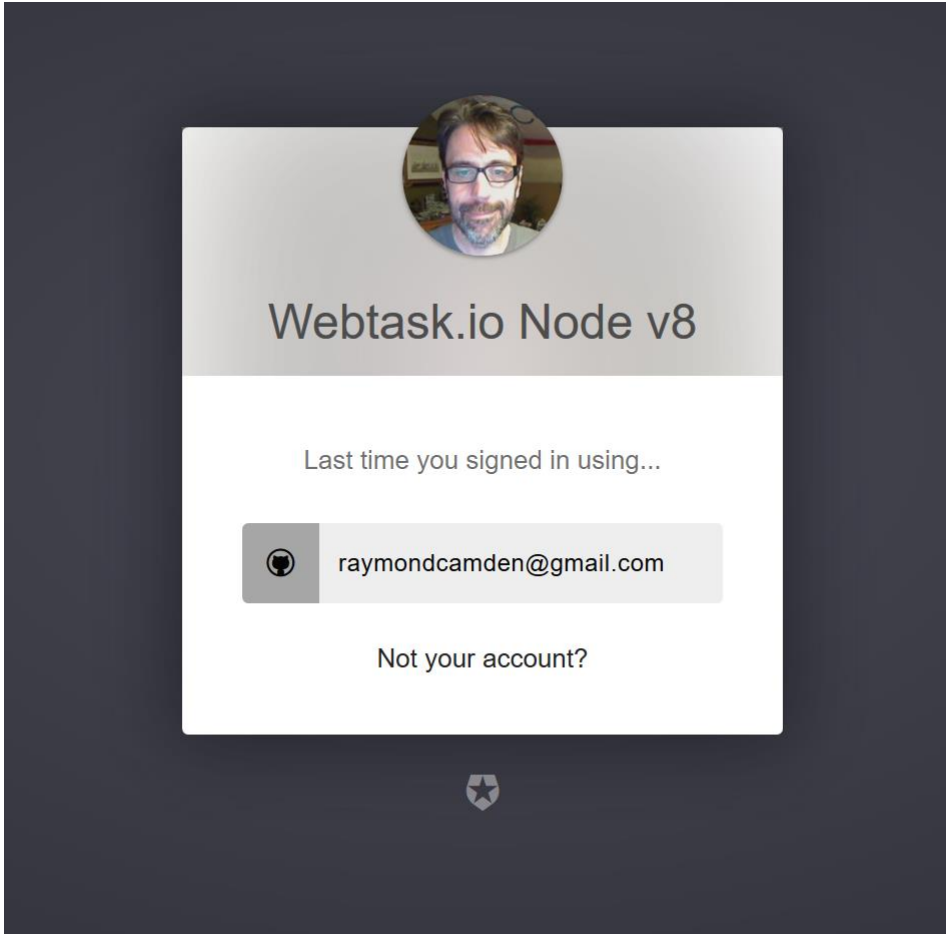
- No credit card!
- Login options: Facebook, Google, GitHub, and Microsoft

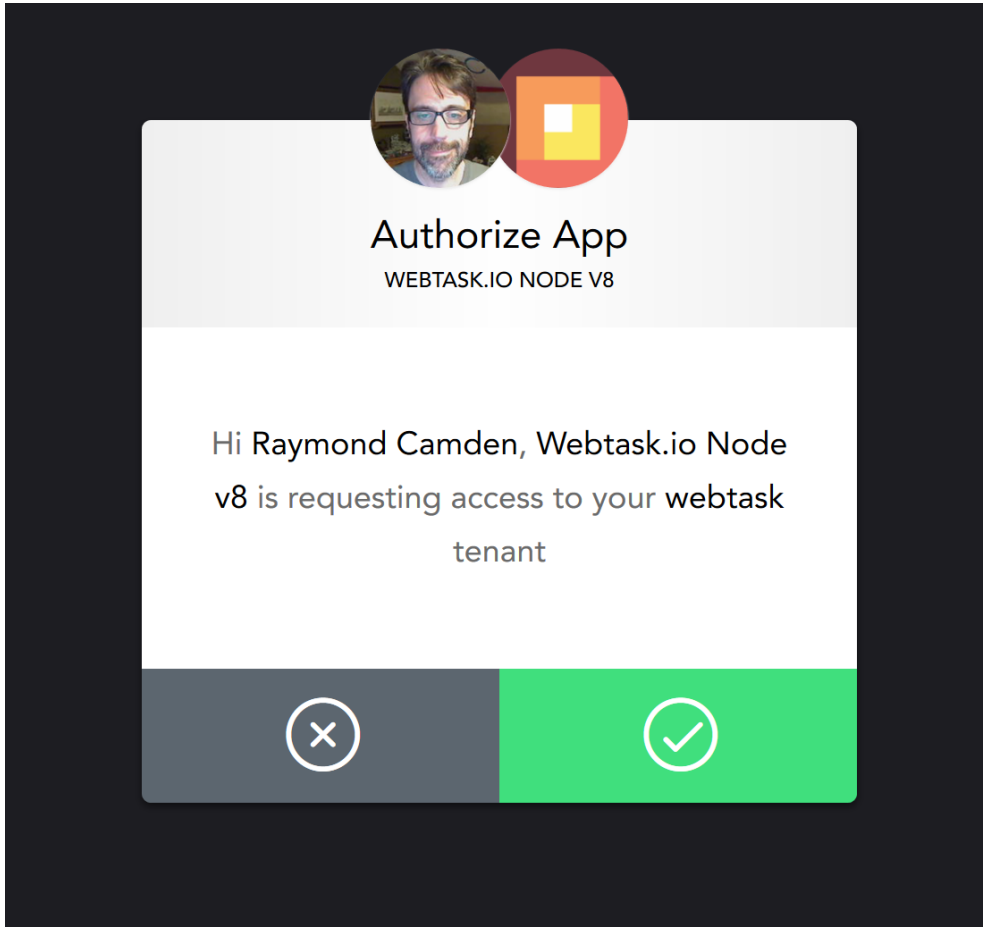


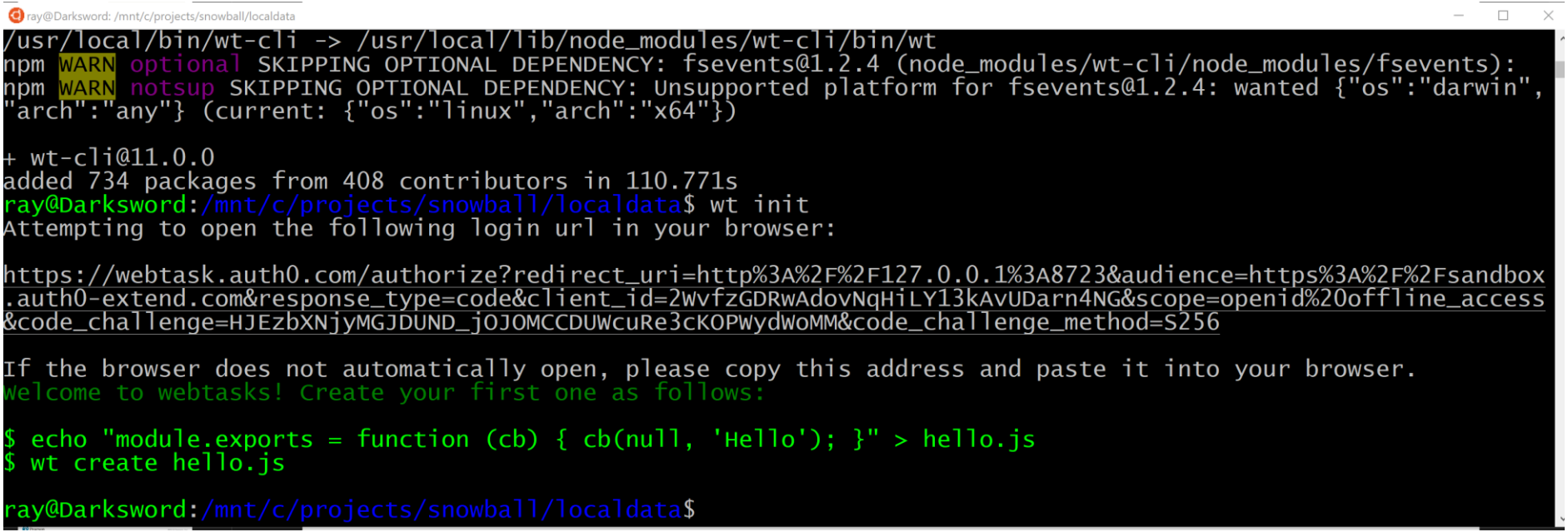
Signing Up











```
ray@Darksword:~$ echo "foo" > hello.js
ray@Darksword:~$ wt create hello.js
Webtask created

You can access your webtask at the following url:

https://wt-c2bde7d7dfc8623f121b0eb5a7102930-0.sandbox.auth0-extend.com/hello
ray@Darksword:~$
```

The Editor!

- https://webtask.io/make
- List your files (and search)
- Make new files (and delete stinky old ones)
- Open files (duh), edit, full screen
- Syntax highlighting, ESLint, Prettier
- Dark theme!
- Oh, and a full testing environment

The Editor!

Writing Webtasks

- Three basic "forms" to follow
- JavaScript
- Simple to Hipster Fancy Stuff

Basic Function Form

```
module.exports = function(cb) {

  cb(null, { hello: 'world' });

};
```

Basic Function Form

```
module.exports = function(cb) {

  try {
    let x = 10;
    let y = a;
    cb(null, { result: x / y });
  } catch(e) {
    cb(e);
  }
};
```

Context Form

```
/**
* @param context {WebtaskContext}
*/
module.exports = function(context, cb) {
  cb(null, { hello: context.query.name || 'Anonymous' });
};
```

Full HTTP Control

```
module.exports = function (context, req, res) {
    res.writeHead(200, { 'Content-Type': 'text/html'});
    let name = context.query.name || 'Anonymous';
    res.end('<h1>Hello, '+name+'</h1>');
}
```

Exercise One – Pig Latin

- Translate an English word to Pig Latin
- Take first consonant, moves it to the end, and adds “ay”. Or if the word begins with a vowel, add “way” to the end
- https://forum.freecodecamp.org/t/freecodecamp-algorithm-challenge-guide-pig-latin/16039
- Create a front end (jQuery, Vue, vanilla JS, anything)

Break

Section 3 – Deeper into Webtask

The Context Object

- Query params
- Body params
- Headers
- Meta
- Secrets

Query params

- Query string parameters
- foo.com/search?name=ray&age=20
- Useful for GET requests
- Parsed by NPM module querystring – not qs.
- context.query = { name: ray, age: 20 }

Query params - Examples

- pearson_c3_qs1
- pearson_c3_qs2

Body params

- Either form fields
- Or application/json POSTed info
- Useful for POST requests
- context.body

Body params - Examples

- pearson_c3_body1
- pearson_c3_body2
- pearson_c3_body2fixed

Header params

- HTTP headers
- Useful for reading HTTP headers (sorry!)

Header params - Examples

- pearson_c3_header1

Meta params

- Metadata
- No real impact on your code (unless you allow it)
- Defined in UI or via cli
- context.meta

Meta params - Examples

- pearson_c3_meta1

Secret params

- Are secret
- Perfect for API keys
- Your code can still expose them!
- context.secrets

Secret params - Examples

- pearson_c3_secret1

Exercise 1 – Super Pig Latin App

- Convert previous example to assume a large body of text
- Take each word and translate
- Return new string of pig latin

Break

Section 4 – Storage

Webtask Storage

- Very, very lightweight persistence system
- One big JSON object
- No query, no filtering, just a blob of data
- Conflict detection (resolution is up to you)
- UI support
- You can still use “real” persistence

Webtask Storage - Writing

```
module.exports = function(context, cb) {

    let data = { name: 'ray', age: 'too old' };

    context.storage.set(data, function (error) {
        if (error) return cb(error);
        // ...
    });

}
```

Webtask Storage - Reading

```
module.exports = function(context, cb) {

    context.storage.get(function (error, data) {
        if (error) return cb(error);
        cb(null, data);
    });

}
```

Webtask Storage - Examples

- pearson_c4_readwrite

Handling Conflicts

- Do “something” and run your set operation again
- Force your updates

Handling Conflicts - Manually

```
module.exports = function(ctx, cb) {
    ctx.storage.get(function (error, data) {
        if (error) return cb(error);
        data = data || { counter: 1 };
        var attempts = 3;
        ctx.storage.set(data, function set_cb(error) {
            if (error) {
                if (error.code === 409 && attempts--) {
                    // resolve conflict and re-attempt set
                    data.counter = Math.max(data.counter, error.conflict.counter) + 1;
                    return ctx.storage.set(data, set_cb);
                }
                return cb(error);
            }
            // ...
        });
    });
}
```

Handling Conflicts - Forced

```
module.exports = function(context, cb) {

    context.storage.set(data, { force:1 }, function (error) {
        if (error) return cb(error);
        // ...
    });

}
```

Exercise 1 – Score Keeper

- API accepts a “score” for a game (1-10)
- Webtask will remember total number of games, total points
- Returns total number of games and average score

Break

Section 5 – NPM Modules

NPM Modules

- Libraries of functionality
- Primary way of sharing code for Node-based applications
- (Heck of a lot more that I'm not covering here…)
- Webtask: Editor and CLI support

NPM Modules via CLI

- Command line automatically detects a package.json
- Also supports --dependency flag
  - --dependency request@2

Editor Support

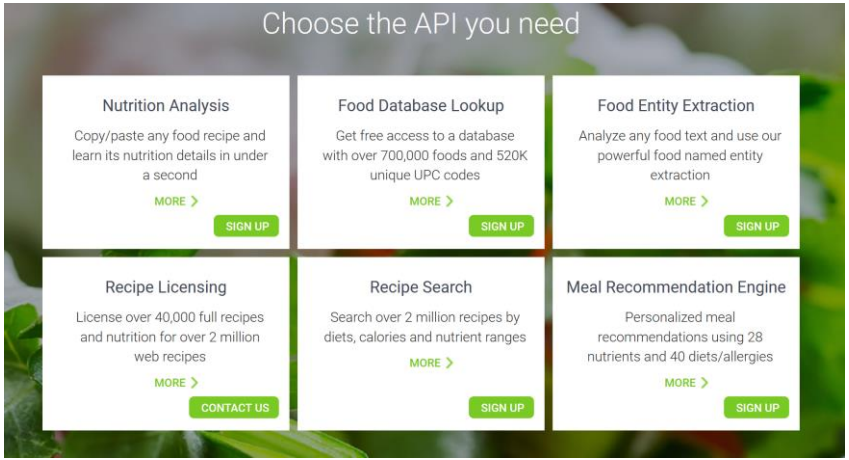
- Tool for managing NPM modules (add, edit version, delete, search)
- Auto-detection in code

Demo

- pearson_c5_npm

Exercise One –Recipe API

- Build an API Wrapper
- A ‘proxy’ to an API to simplify result
- Recipe API (https://developer.edamam.com/)



Exercise One – Recipe API

- Get a key
- Create a proxy to return search results
- Return name, picture, link, and calories
- https://developer.edamam.com/
- https://api.edamam.com/search?q=SEARCH&app_id=X&app_key=Y
- Suggested npm modules: request and request-promise
- Suggestion: Use secrets!

Break

Section 6 –Middleware

Middleware

- If you've used middleware in Node apps – same! (Mostly)
- Logic that runs before your webtask
- Examples:
  - Authentication
  - Modify/prepare data
- Logic that creates new compilers
  - Change the 'function' form
  - Bypass the 'regular' NodeJS compiler

Middleware

- Specified via the editor UI or via metadata
- URL, npm module, or npm module export
- For URLs, result must be text!

Middleware Example

```
module.exports = function() {

    return function (req, res, next) {
        console.log('middleware running');
        return next();
    };

}
```

Middleware Example

```
module.exports = function() {

    return function (req, res, next) {
        console.log('middleware running');
        if(somethingBad) {
            return next(new Errors('Oops'));
        } else {
            return next();
        }
    };

}
```

Middleware Demo

- pearson_c6_middleware1.js

More on Middleware

- You have access to the code of the webtask: req.webtaskContext.compiler.script
- You have access to a JS compiler: req.webtaskContext.nodejsCompiler
- https://webtask.io/docs/middleware
- https://goextend.io/blog/using-alexa-with-webtask

Break

Serverless with Webtask

Section 7 –The Rest!

Everything Else (Mostly)

- CRON (Scheduled) Webtasks
- wt serve / wt debug
  - https://nodejs.org/en/docs/guides/debugging-getting-started/
- wt logs
- Extend (goextend.io)
  - https://goextend.io/docs/getting-started

Why NOT to use Webtask?

- Results
- Chaining
- No upgrade path
  - 1 request per second soft limit
  - 30 second execution limit
  - 10 CRON Schedules
  - 100kb of code per task (NPM modules don’t count)
  - 500kb of storage via Storage API
  - https://webtask.io/docs/limits

Webtask Resources

- Docs: https://webtask.io/docs
- Slack: https://skynet.run.webtask.io/webtask-signup
- Forum: https://community.webtask.io/
- Blog: https://goextend.io/blog/tag/webtasks

Serverless Resources

- Serverless Status
  - https://serverless.email
- Serverless [Cron]icle
  - https://serverless.curated.co/
- My own stuff:
  - https://www.raymondcamden.com/categories/serverless