

THE PROOF IS IN THE PUDDING

HAPI + SENECA



SENECA & HAPI CORE CONTRIBUTOR

Wyatt Preul / @geek



**THE PROOF IS IN THE EATING
OF THE PUDDING**

Pudding



PROOF IS IN THE HOT CHICKEN



The Daily
Business Post



worldline
e-payment services



★macy's

•● OpenTable™

nearForm

longalgae



npm

&yet



 GetHuman

CONDÉ NAST



 Lob

mozilla

 codio



YAHOO!

 MOVING
WORLDS

pebble {code}

 MODULUS
A PROGRESS COMPANY

PayPal™

 Disney

I LOVE GOING OUT AND DOING
NEW THINGS

Bo Jackson #ForeverRoyal

HAPI

- ▶ web framework for node.js
- ▶ enterprise grade & battle tested



HAPI FEATURES

- ▶ caching
- ▶ authentication
- ▶ request lifecycle
- ▶ plugin architecture

SENECA

- ▶ microservices toolkit for node
- ▶ plugin architecture



HAPI + SENECA = ΧΑΪΡΩ (CHAIRO)

- ▶ hapi plugin for enabling seneca support
- ▶ yes, Seneca is a Roman philosopher, and χαῖρω is Greek, but felix is taken

**LETS BUILD A USEFUL
MICROSERVICE**

H O T

-Mattice's

CHICKEN

IS HATTIE B'S OPEN

```
const isOpen = function (time) {  
  var hours = new Date(time).getHours()  
  
  return hours >= 11 && hours < 22  
}
```

HAPI SERVICE

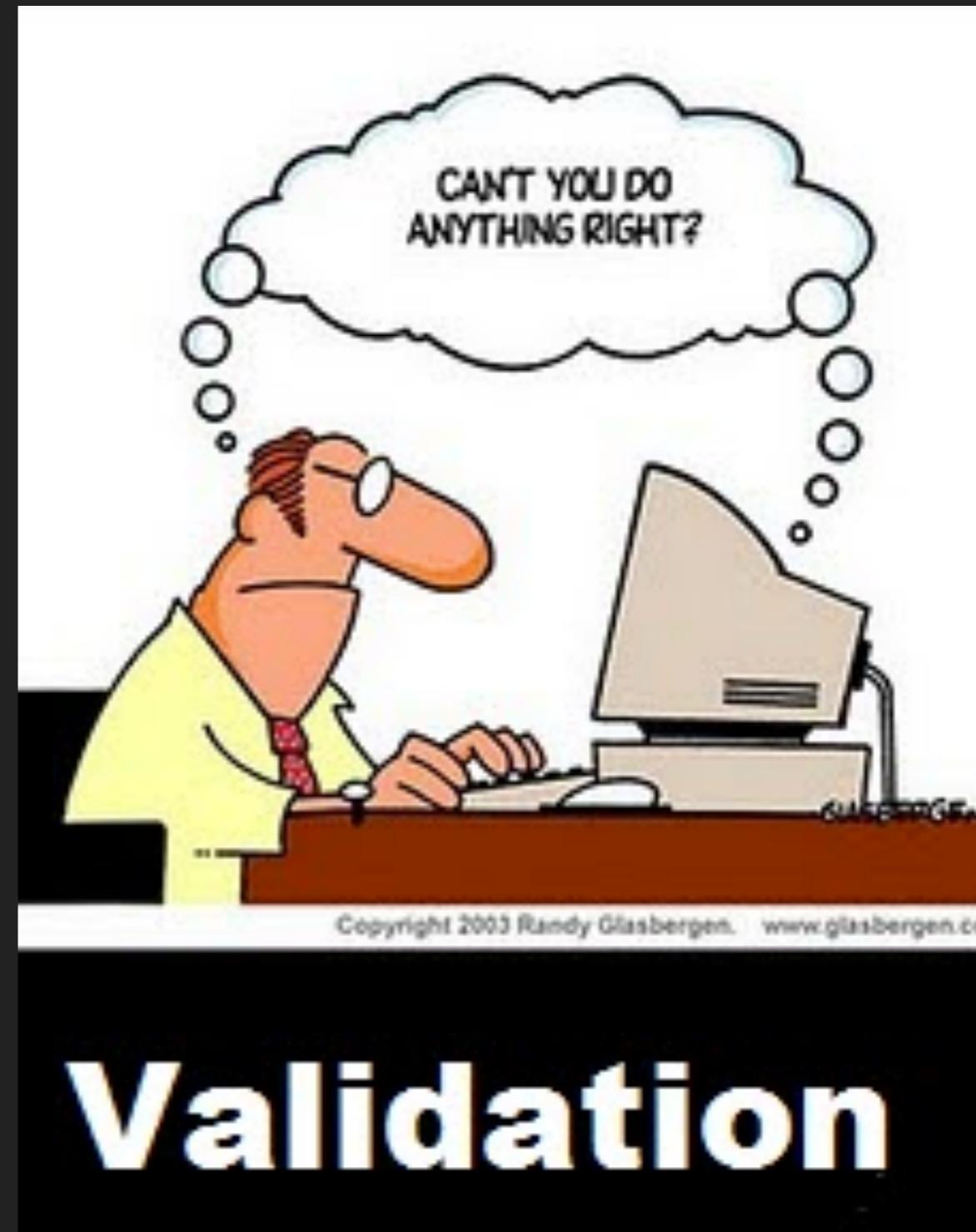
```
const Hapi = require('hapi')
const server = new Hapi.Server()
server.connection({ port: 8080 })

server.route({
  method: 'get',
  path: '/{time}',
  handler: (request, reply) => {
    var hours = new Date(request.params.time).getHours()

    reply(hours >= 11 && hours < 22)
  }
})
```

JOI

- ▶ one of the most powerful modules in the hapi universe
- ▶ can be used outside of hapi!



VALIDATION

```
server.route({
  method: 'get',
  path: '/{time}',
  config: {
    validate: {
      params: {
        time: Joi.date()
      }
    },
    handler: (request, reply) => {
      var hours = request.params.time.getHours()

      reply(hours >= 11 && hours < 22)
    }
  }
})
```

AS A PLUGIN

```
module.exports = function (server, options, next) {
  server.route({
    method: 'get',
    path: '/{time}',
    config: {
      validate: {
        params: {
          time: Joi.date()
        }
      },
      handler: (request, reply) => {
        var hours = request.params.time.getHours()

        reply(hours >= 11 && hours < 22)
      }
    }
  })
  next()
}
```

**IT IS QUALITY RATHER THAN
QUANTITY THAT MATTERS.**

Seneca in Letter XLV to Lucilius

SERVER WITH ROUTE PLUGIN

```
const Hapi = require('hapi')
const Route = require('./route')

const server = new Hapi.Server()
server.connection({ port: 8080 })

server.register(Route, (err) => {
  server.start((err) => {})
})
```



MICROSERVICES

SENECA

IS HATTIE B'S OPEN SENECA MICROSERVICE

```
const Seneca = require('seneca')

const isOpen = function (message, next) {
  var hours = new Date(message.time).getHours()

  next(null, { isOpen: hours >= 11 && hours < 22 })
}

Seneca().add({ role: 'isOpen' }, isOpen).listen()
```

ACTOR MODEL

- ▶ JSON message patterns
- ▶ async by nature



IS HATTIE B'S OPEN SENECA CLIENT

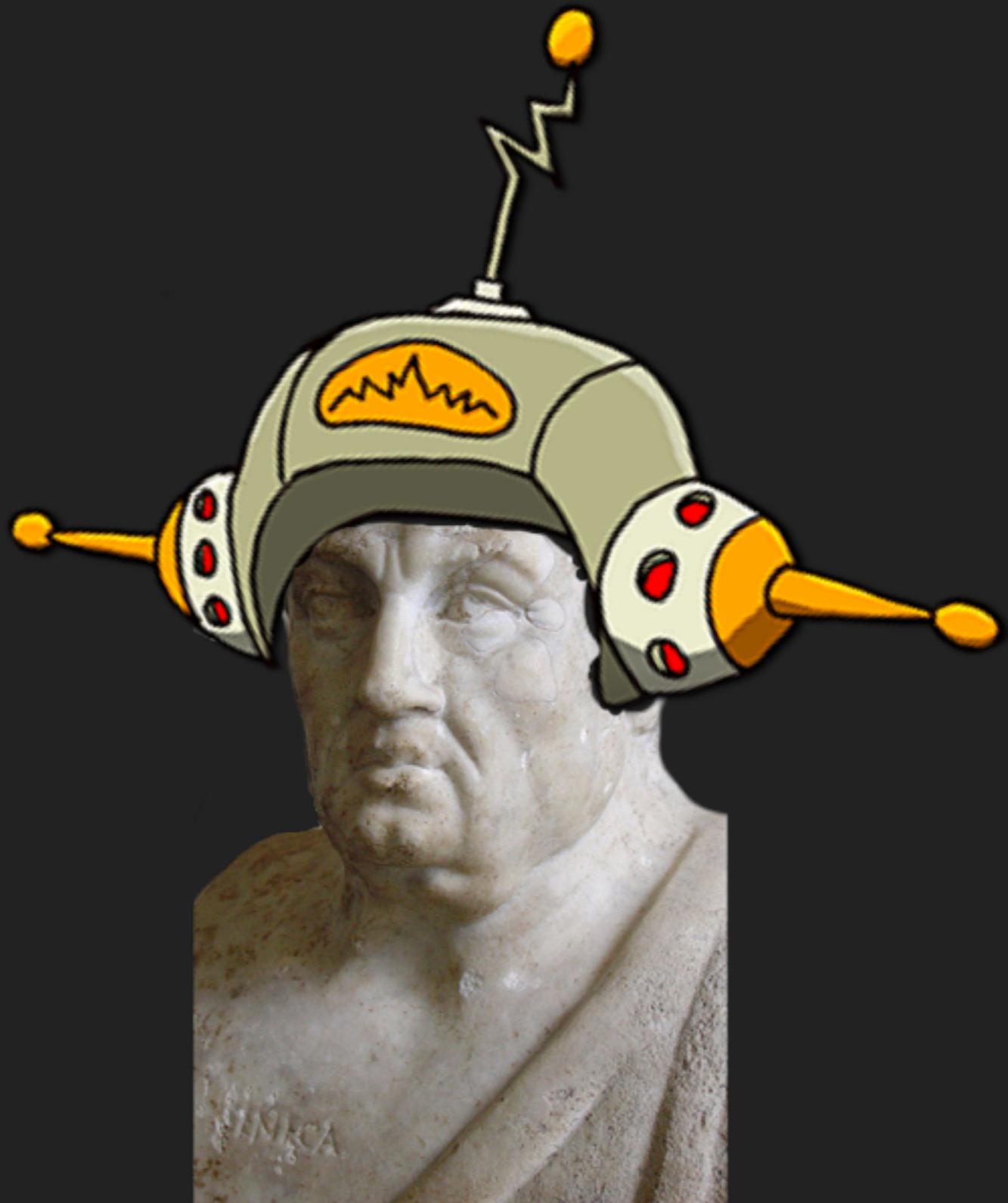
```
const Seneca = require('seneca')

const seneca = Seneca().client()
seneca.act({
  role: 'isOpen',
  time: 'Wed Nov 11 2015 11:06:02'
}, (err, result) => {

  console.log(result.isOpen) // true
})
```

CHAIRO

- ▶ decorate server/request with seneca instance
- ▶ act handlers
- ▶ seneca results get hapi caching support



SENECA SERVICE AS HAPI PLUGIN

```
module.exports = function (server, options, next) {  
  server.dependency('chairo')  
  server.seneca.add({ role: 'isOpen' }, isOpen)  
  next()  
}
```

HAPI HANDLER USING SENECA SERVICE

```
handler: (request, reply) => {

  request.seneca.act({
    role: 'isOpen',
    time: request.params.time
  }, (err, result) => {

    reply(result.isOpen)
  })
}
```

**BUT I AM NOT AT
HATIE B'S**



WHEN I ARRIVE

**WILL HATTIE B'S
OPEN**

GOOGLE DISTANCE TO HATTIE B'S

```
const arrivalTime = function (origin, callback) {
  GoogleDistance.get({
    origin: origin,
    destination: '112 19th Ave S, Nashville, TN 37203',
  }, (err, result) => {

    if (err) {
      return callback(err)
    }

    callback(null, Date.now() + (result.durationValue * 1000))
  })
}
```

AS A ROUTE

```
request.server.methods.arrivalTime({origin: request.params.origin},  
(err, arrival) => {  
  
  if (err) {  
    return reply(Boom.internal(err))  
  }  
  
  request.seneca.act({role: 'isOpen', time: arrival.time},  
(err, message) => {  
  
    if (err) {  
      return reply(Boom.internal(err))  
    }  
  
    reply({  
      arrivalTime: new Date(arrival.time),  
      isOpen: message.isOpen  
    })  
  })  
})
```

EVERYTHING AS A PLUGIN

```
const Chairo = require('chairo')
const Hapi = require('hapi')
const ArrivalTime = require('./arrival_time')
const ArrivalOpenRoute = require('./arrival_open_route')
const IsOpen = require('./is_open')

const plugins = [Chairo, IsOpen, ArrivalTime, ArrivalOpenRoute]
const server = new Hapi.Server()
server.connection({ port: 8080 })

server.register(plugins, () => {
  server.start((err) => {
    console.log(err || 'server started on ' + server.info.port)
  })
})
```

RUN IT

```
$ curl http://localhost:8080/kansas%20city,%20mo
```

```
{  
    arrivalTime: "2015-11-13T09:30:01.525Z",  
    isOpen: false  
}
```

PROJECT MANAGER, UH OH

- ▶ opens JIRA for you to use more servers, since they want to expand to more customers
- ▶ JIRA ticket says to use Docker, since the VP just announced that you all are using Docker



NO PROBLEM, WE ARE USING
PLUGINS AND
MICROSERVICES

your ego

**YEAH, IF YOU COULD GO AHEAD AND READ THIS IN
MY VOICE**

THAT'D BE GREAT

ARRIVAL SERVER

```
const Chairo = require('chairo')
const Hapi = require('hapi')
const ArrivalTime = require('./arrival_time')
const ArrivalOpenRoute = require('./arrival_open_route')

const server = new Hapi.Server()
server.connection({ port: 8080 })

server.register([Chairo, ArrivalTime, ArrivalOpenRoute], () => {
  server.start((err) => {
    console.log(err || 'server started on ' + server.info.port)
  })
})
```

IS OPEN SERVER

```
const Chairo = require('chairo')
const Hapi = require('hapi')
const IsOpen = require('./is_openV2')

const server = new Hapi.Server()
server.connection({ port: 0 })

server.register([Chairo, IsOpen], () => {
  server.start((err) => {
    console.log(err || 'server started on ' + server.info.port)
  })
})
```

CLOSES JIRA TICKET

DONE.

JIRA ISSUE ASSIGNED BACK TO YOU

DOCKER?

DOCKERFILE

```
FROM node

RUN mkdir -p /app
WORKDIR /app
COPY package.json package.json
COPY arrival_time.js arrival_time.js
COPY arrival_open_route.js arrival_open_route.js
COPY arrivalServer.js arrivalServer.js

RUN npm install
```

DOCKER-COMPOSE

```
isopen:  
  build: .  
  dockerfile: Dockerfile-isopen  
  expose:  
    - "8000"  
  entrypoint: node /app/isOpenServer.js  
arrival:  
  build: .  
  dockerfile: Dockerfile-arrival  
  links:  
    - isopen  
  ports:  
    - "8080:8080"  
  entrypoint: node /app/arrivalServer.js
```

A dark, atmospheric illustration depicting a scene of conflict or death. In the center, a knight in full armor, including a helmet with a plume, sits on a horse. He holds a large, ornate sword. To his right, a skull lies on the ground, partially obscured by smoke and fire. Another smaller figure, also in armor, stands behind the skull. The background is filled with swirling smoke, fire, and debris, creating a sense of chaos and destruction.

DOCKER-COMPOSE UP

RECAP

- ▶ hapi + seneca work well together
- ▶ code @ npm install
nodevember-2015
- ▶ the proof is in the hot chicken





TRY IT, YOU MIGHT
LIKE IT

HAPI + SENECA