Groovy Scripting

Why Scripting Languages?

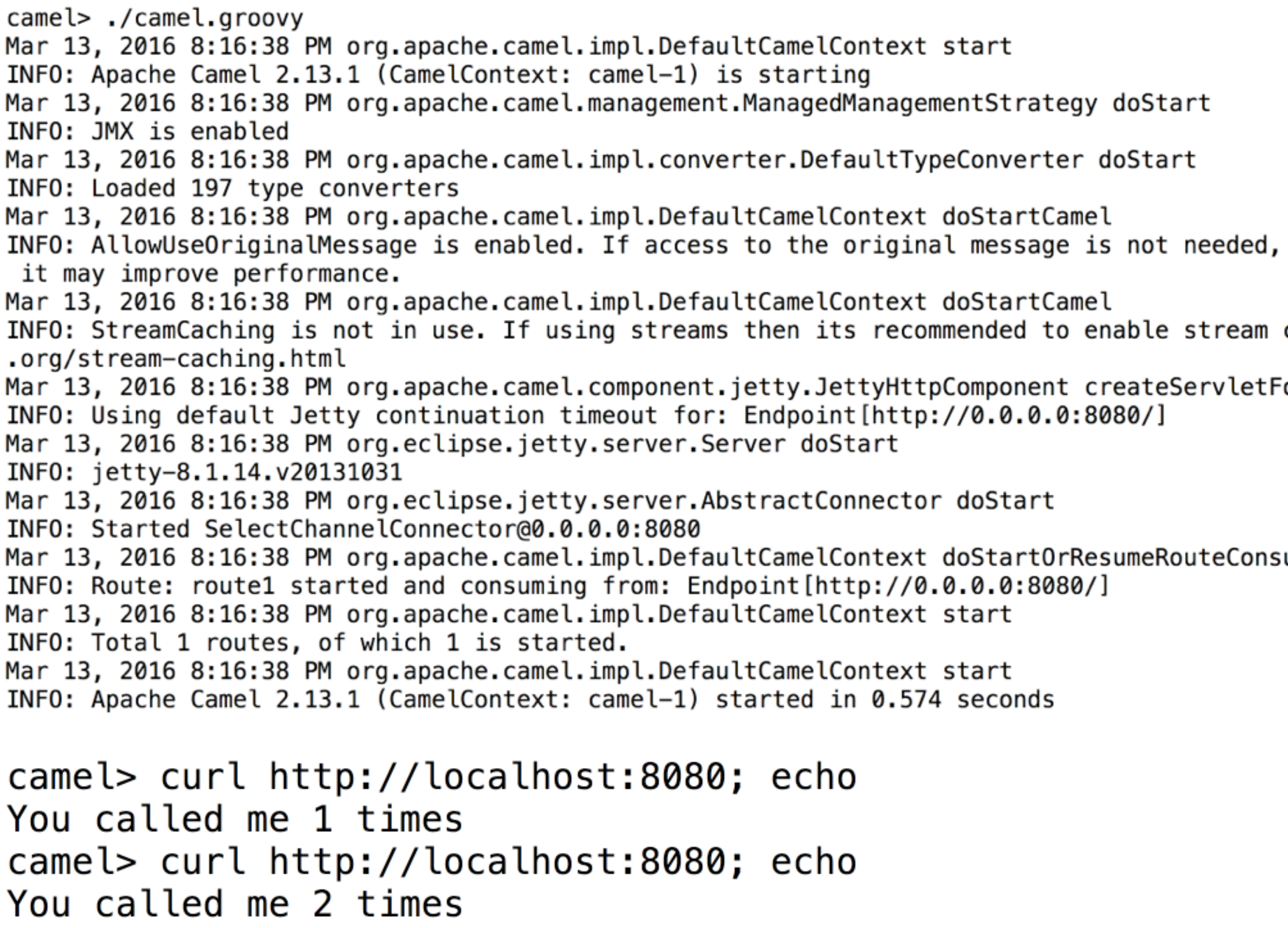
- Some tasks are hard using Bash & friends
- The same tasks might be too small for a full-blown project
- Rapid feedback & bug fixing
- Many scripting languages around
    - Perl, Tcl, Python, Ruby

Groovy

Getting Hooked On Groovy

```groovy
import org.apache.camel.builder.RouteBuilder;
import org.apache.camel.impl.DefaultCamelContext

@Grab(group="org.apache.camel", module="camel-groovy", version="2.13.1")
@Grab(group="org.apache.camel", module="camel-jetty", version="2.13.1")
@Grab(group="org.slf4j", module="slf4j-jdk14", version="1.7.5")

class MyRouteBuilder extends RouteBuilder {
    int num = 0;
    def number = {++num};
    void configure() {
        from("jetty:http://0.0.0.0:8080/")
        .transform({"You called me ${number()} times"})
        .end()
    }
}

def camelContext = new DefaultCamelContext();
camelContext.addRoutes(new MyRouteBuilder())
camelContext.start();
```

http://www.slideshare.net/davsclaus/apache-camel-gr8conf



```
camel> ./camel.groovy
Mar 13, 2016 8:16:38 PM org.apache.camel.impl.DefaultCamelContext start
INFO: Apache Camel 2.13.1 (CamelContext: camel-1) is starting
Mar 13, 2016 8:16:38 PM org.apache.camel.management.ManagedManagementStrategy doStart
INFO: JMX is enabled
Mar 13, 2016 8:16:38 PM org.apache.camel.impl.converter.DefaultTypeConverter doStart
INFO: Loaded 197 type converters
Mar 13, 2016 8:16:38 PM org.apache.camel.impl.DefaultCamelContext doStartCamel
INFO: AllowUseOriginalMessage is enabled. If access to the original message is not needed,
 it may improve performance.
Mar 13, 2016 8:16:38 PM org.apache.camel.impl.DefaultCamelContext doStartCamel
INFO: StreamCaching is not in use. If using streams then its recommended to enable stream c
.org/stream-caching.html
Mar 13, 2016 8:16:38 PM org.apache.camel.component.jetty.JettyHttpComponent createServletFc
INFO: Using default Jetty continuation timeout for: Endpoint[http://0.0.0.0:8080/]
Mar 13, 2016 8:16:38 PM org.eclipse.jetty.server.Server doStart
INFO: jetty-8.1.14.v20131031
Mar 13, 2016 8:16:38 PM org.eclipse.jetty.server.AbstractConnector doStart
INFO: Started SelectChannelConnector@0.0.0.0:8080
Mar 13, 2016 8:16:38 PM org.apache.camel.impl.DefaultCamelContext doStartOrResumeRouteConsu
INFO: Route: route1 started and consuming from: Endpoint[http://0.0.0.0:8080/]
Mar 13, 2016 8:16:38 PM org.apache.camel.impl.DefaultCamelContext start
INFO: Total 1 routes, of which 1 is started.
Mar 13, 2016 8:16:38 PM org.apache.camel.impl.DefaultCamelContext start
INFO: Apache Camel 2.13.1 (CamelContext: camel-1) started in 0.574 seconds

camel> curl http://localhost:8080; echo
You called me 1 times
camel> curl http://localhost:8080; echo
You called me 2 times
```

Behind The Scenes

- Creates an Ivy repository in "~/.groovy"
- Stores transitive dependencies
- Setup a Camel route using embedded Jetty
- Script executed by Groovy interpreter

Good Old Maven Way

- Write a pom.xml declaring dependencies
- Configure the " AppAssembler Maven Plugin" to create a command line application
- Maybe tinker with "Maven Ant Plugin" to create an executable shell script
- Upload/download distributable

THE
SIDE
PROJECT

The Side Project

- Useful but not important things
- Perfect for “bridging days”
- Stretch your skills
- Bring back the fun

Siegfried’s Side Project

- Automate test user documentation
- Set of test users changes every few months
- Each test users has one or more products
  - Account, credit cart, pension ….
- Excel sheet created manually

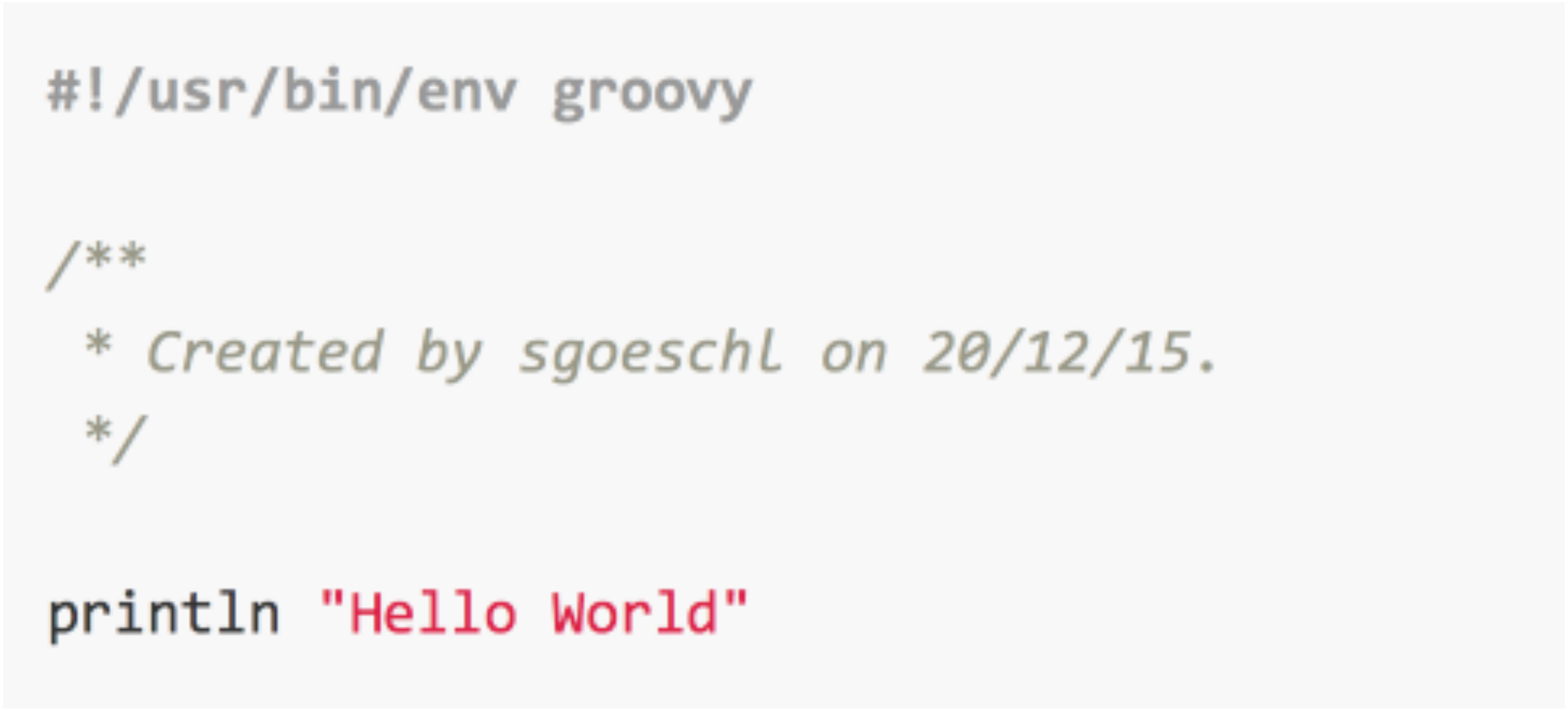
The Problem At Hand

- User id for disposer “9422753404”?
- Any test user with a “Building Saving Plan”?
- When do we get the newest Excel sheet?

Idea Before Christmas

- Let’s make one QA guy happy
- Make a few REST calls to gather data
- Create a HTML document with that data
- Could be done for all test environments
- Let's implement it in Groovy

Groovy First Contact



```
#!/usr/bin/env groovy

/**
 * Created by sgoeschl on 20/12/15.
 */

println "Hello World"
```

Plans are only good intentions unless they immediately degenerate into hard work.

Peter Drucker

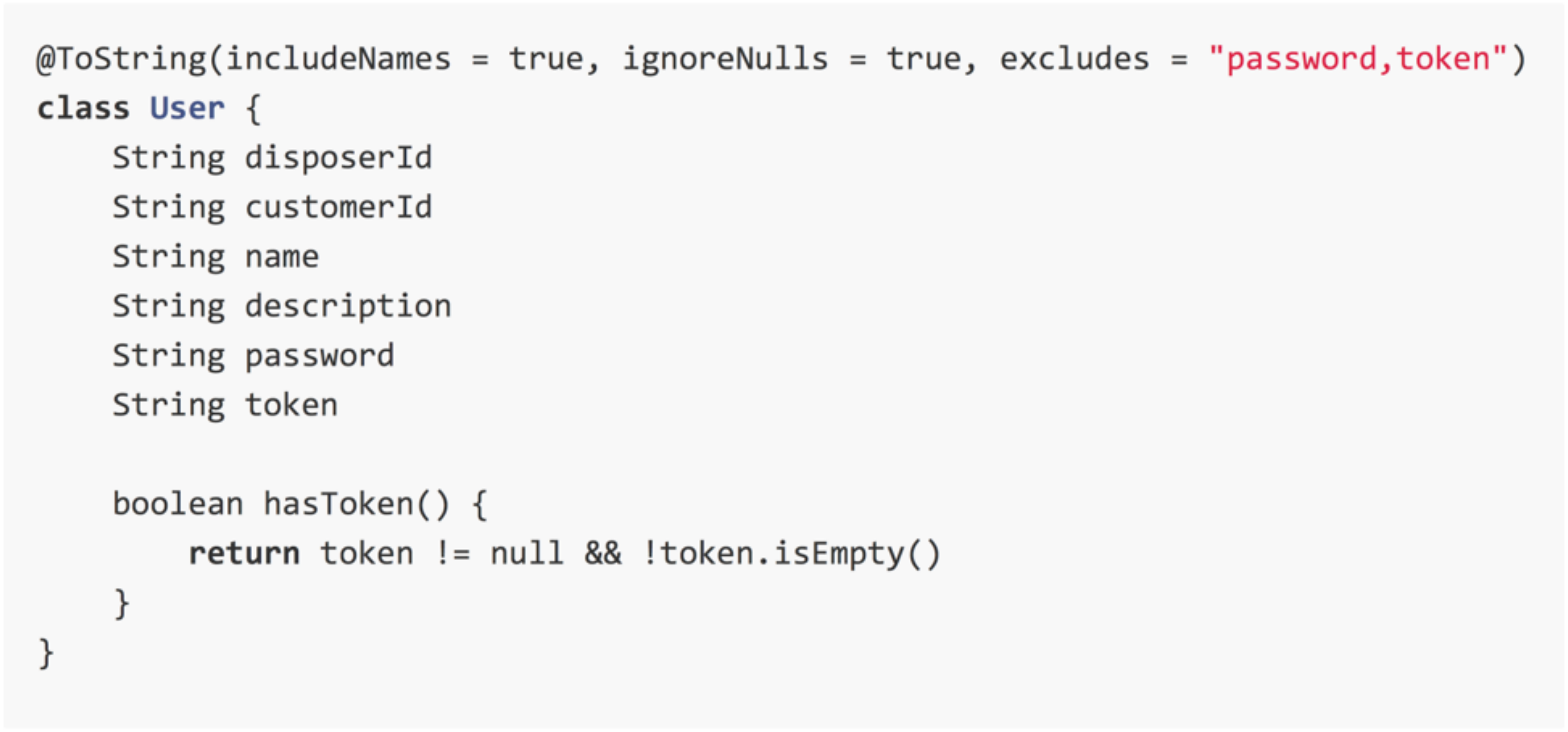
Hard Work Ahead

- Classes to hold a user and its products
- Implement command line parser
- Read user & credentials from a CSV file
- Invoke REST endpoints to get products
- Transform user and their products to documentation

Groovy Classes

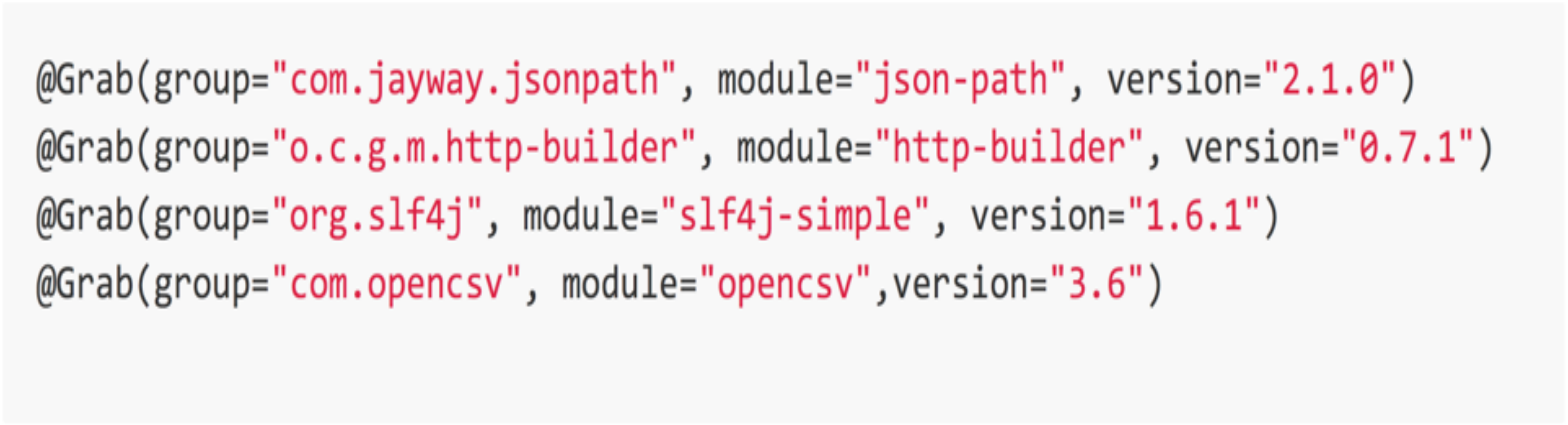
```groovy
@ToString(includeNames = true)
class UserProduct {
    User user
    List<Account> accounts
    List<Card> cards
    List<Building> buildings
    List<Pension> pensions
    List<Insurance> insurances
    List<Security> securities
}
```

Groovy Classes



```
@ToString(includeNames = true, ignoreNulls = true, excludes = "password,token")
class User {
    String disposerId
    String customerId
    String name
    String description
    String password
    String token

    boolean hasToken() {
        return token != null && !token.isEmpty()
    }
}
```

Grape Dependency Manager



```
@Grab(group="com.jayway.jsonpath", module="json-path", version="2.1.0")
@Grab(group="o.c.g.m.http-builder", module="http-builder", version="0.7.1")
@Grab(group="org.slf4j", module="slf4j-simple", version="1.6.1")
@Grab(group="com.opencsv", module="opencsv",version="3.6")
```

Reading CSV

```
CSVReader csvReader = new CSVReader(
    new FileReader(fileName),
    CSVParser.DEFAULT_SEPARATOR,
    CSVParser.DEFAULT_QUOTE_CHARACTER,
    1)

csvReader.readAll().each {
    result.add(
        new User(
            disposerId: it[0],
            password: it[1],
            token: it[2],
            name: it[3],
            description: it[4])
    )
}
```

Groovy HTTPBuilder

```
HTTPBuilder createHTTPBuilder(String resourceUrl) {

    Map defaultHeaders = [
        'Accept'          : 'application/json',
        'Authorization'   : "bearer ${token}"]

    String url = "${baseUrl}/${resourceUrl}"
    HTTPBuilder httpBuilder = new HTTPBuilder(url)
    httpBuilder.setHeaders(defaultHeaders)
    httpBuilder.ignoreSSLIssues()

    def params = httpBuilder.getClient().getParams()
    params.setParameter("http.connection.timeout", new Integer(10 * 1000))
    params.setParameter("http.socket.timeout", new Integer(60 * 1000))

    return httpBuilder
}
```

Groovy HTTPBuilder

```groovy
def getMyAccounts() {
    HTTPBuilder httpBuilder = createHTTPBuilder('netbanking/my/accounts')
    httpBuilder.request(GET, JSON) { req ->
        uri.query = ['_': System.currentTimeMillis()]
        response.success = { resp, json ->
            return json
        }
    }
}
```

Groovy CLIBuilder

```
def cli = new CliBuilder(usage: 'groovy cli.groovy -s[dh] "server"')
cli.h(longOpt: 'help', 'usage information', required: false)
cli.s(longOpt: 'server', 'server to connect to', required: true, args: 1)
cli.d(longOpt: 'debug', 'enable debugging', required: false)

cli.parse(args)
```

```
cli> groovy cli-simple.groovy
error: Missing required option: s
usage: groovy cli.groovy -s[dh] "server"
 -d,--debug            enable debugging
 -h,--help             usage information
 -s,--server <arg>     server to connect to
```

Groovy ConfigSlurper

```
def config = new ConfigSlurper('production').parse('''
website {
 url = "http://default.mycompany.com"
 port = 80
 user = "test"
}

environments {
 development {
  website {
   url = "http://dev.mycompany.com"
   port = 8080
  }
 }
 production {
  website {
   url = "http://www.mycompany.com"
   user = "prodUser"
  }
 }
}''')

assert config.website.url == "http://www.mycompany.com"
assert config.website.port == 80
assert config.website.user == "prodUser"
```

Groovy JSONOutput

```
import groovy.json.JsonOutput

CommandLine cli = new CommandLine(args)
UserProduct userProduct = new UserProduct(user: user, accounts: accounts)
String directoryName = "./out/${cli.getTenant()}/${cli.getSite()}"
File directory = new File(directoryName)
directory.mkdirs()

new File(directory, "products.json").write(JsonOutput.toJson(userProducts))
```

Where Are We Now?

- Holders for users and its products
- Using a command line parser
- Read user & credentials from a CSV file
- REST client to fetch user products
- JSON file containing user products

JSON To HTML

The Problem Ahead

- With XML I would use XSLT
  - Could somehow dump XML
  - XSLT is hard if you don't know it
- Maybe transform JSON to Markdown?
  - JSON & Markdown are everywhere

JSON To Markdown

- The cool kids say ‘node.js'
  - But they always say ‘node.js’
- We could implement it in Groovy
  - Groovy’s MarkupBuilder looks hard
  - Using Apache Velocity & JSONPath?

Groovy Velocity CLI

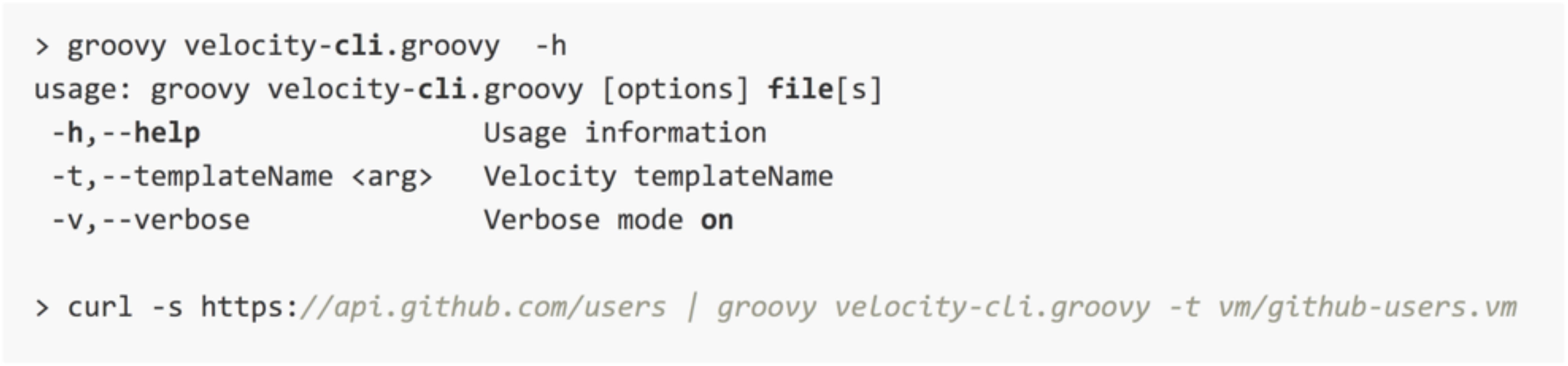
- See https://github.com/sgoeschl/velocity-cli
- See https://github.com/jayway/JsonPath
- Wire Velocity templates with JSON Path
- Supports CSV & JSON transformations

GitHub REST API Example

GitHub REST API

```
[
  {
    "login": "mojombo",
    "id": 1,
    "avatar_url": "https://avatars.githubusercontent.com/u/1?v=3",
    "gravatar_id": "",
    "url": "https://api.github.com/users/mojombo",
    "html_url": "https://github.com/mojombo",
    "followers_url": "https://api.github.com/users/mojombo/followers",
    "following_url": "https://api.github.com/users/mojombo/following{/other_user}",
    "gists_url": "https://api.github.com/users/mojombo/gists{/gist_id}",
    "starred_url": "https://api.github.com/users/mojombo/starred{/owner}{/repo}",
    "subscriptions_url": "https://api.github.com/users/mojombo/subscriptions",
    "organizations_url": "https://api.github.com/users/mojombo/orgs",
    "repos_url": "https://api.github.com/users/mojombo/repos",
    "events_url": "https://api.github.com/users/mojombo/events{/privacy}",
    "received_events_url": "https://api.github.com/users/mojombo/received_events",
    "type": "User",
    "site_admin": false
  }
]
```

Groovy Velocity CLI

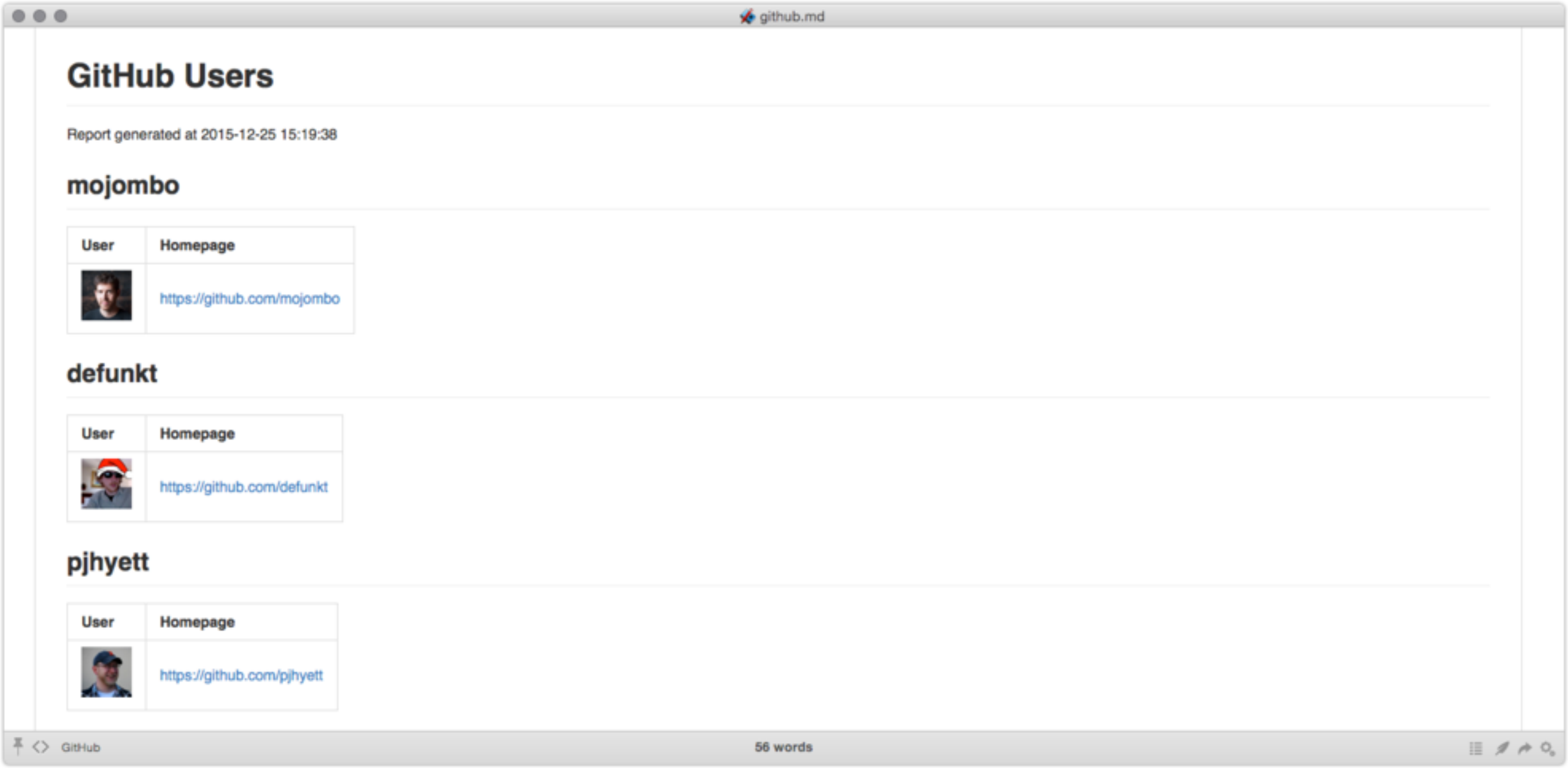


```
> groovy velocity-cli.groovy  -h
usage: groovy velocity-cli.groovy [options] file[s]
 -h,--help                 Usage information
 -t,--templateName <arg>   Velocity templateName
 -v,--verbose              Verbose mode on

> curl -s https://api.github.com/users | groovy velocity-cli.groovy -t vm/github-users.vm
```

Groovy Velocity CLI

```
#set( $json = $documents[0].content )

# GitHub Users

Report generated at $date.get('yyyy-MM-dd HH:mm:ss')

#foreach( $user in $json.read("$[*]") )
  #set( $userAvatarUrl = $user.avatar_url )
  #set( $userHomeUrl = $user.html_url )

#[[##]]# ${user.login}

| User                                                   | Homepage       |
|:------------------------------------------------------|:---------------|
| <img src="$user.avatar_url" width="48" height="48" /> | $userHomeUrl   |
#end
```

Groovy Velocity CLI



github.md

GitHub Users

Report generated at 2015-12-25 15:19:38

mojombo

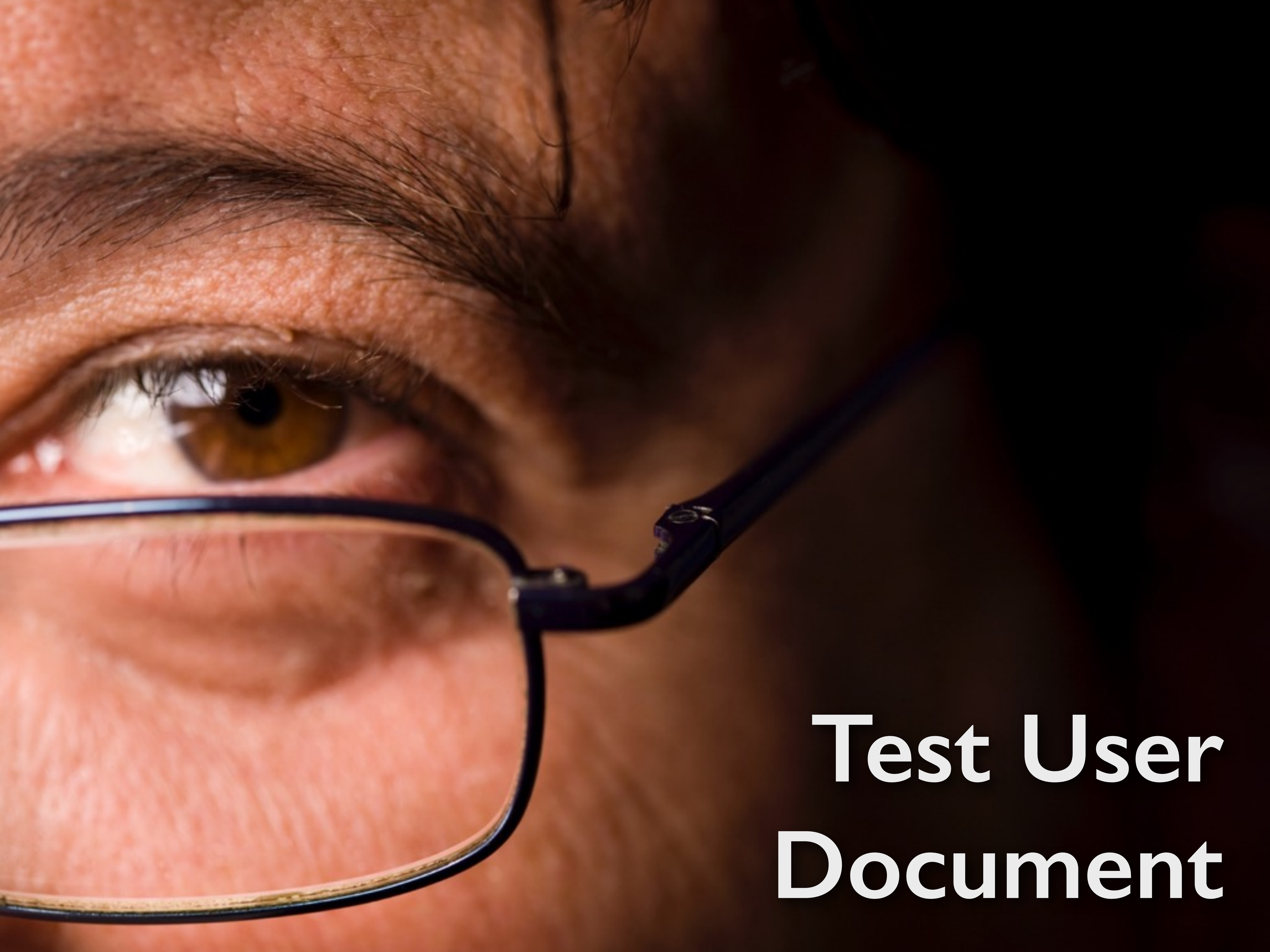
User	Homepage
	https://github.com/mojombo

defunkt

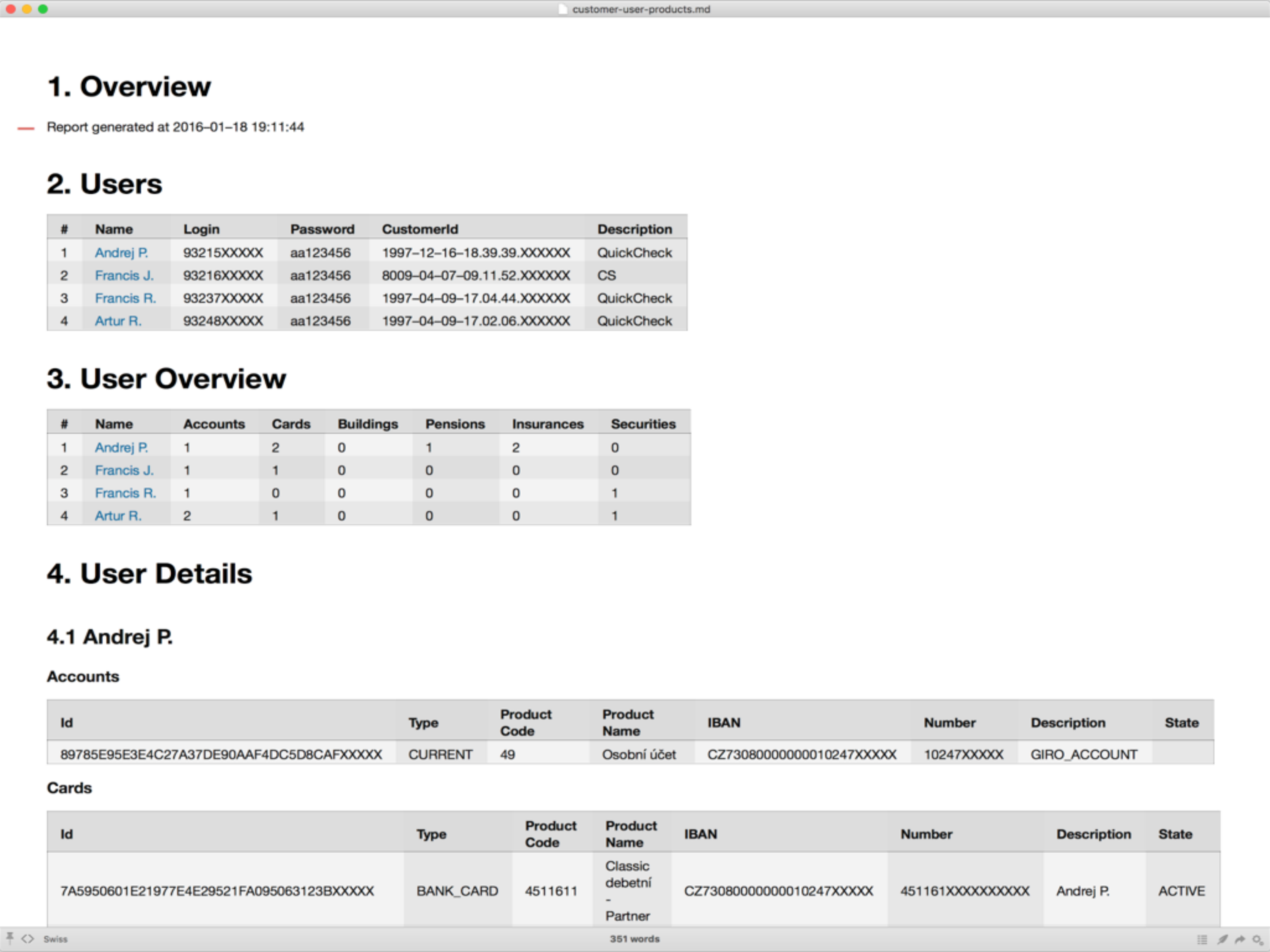
User	Homepage
	https://github.com/defunkt

pjhyett

User	Homepage
	https://github.com/pjhyett



Test User Document



1. Overview

- Report generated at 2016–01–18 19:11:44

2. Users

#	Name	Login	Password	CustomerId	Description
1	Andrej P.	93215XXXXX	aa123456	1997–12–16–18.39.39.XXXXXX	QuickCheck
2	Francis J.	93216XXXXX	aa123456	8009–04–07–09.11.52.XXXXXX	CS
3	Francis R.	93237XXXXX	aa123456	1997–04–09–17.04.44.XXXXXX	QuickCheck
4	Artur R.	93248XXXXX	aa123456	1997–04–09–17.02.06.XXXXXX	QuickCheck

3. User Overview

#	Name	Accounts	Cards	Buildings	Pensions	Insurances	Securities
1	Andrej P.	1	2	0	1	2	0
2	Francis J.	1	1	0	0	0	0
3	Francis R.	1	0	0	0	0	1
4	Artur R.	2	1	0	0	0	1

4. User Details

4.1 Andrej P.

Accounts

Id	Type	Product Code	Product Name	IBAN	Number	Description	State
89785E95E3E4C27A37DE90AAF4DC5D8CAFXXXXX	CURRENT	49	Osobní účet	CZ730800000000010247XXXXX	10247XXXXX	GIRO_ACCOUNT	

Cards

Id	Type	Product Code	Product Name	IBAN	Number	Description	State
7A5950601E21977E4E29521FA095063123BXXXXX	BANK_CARD	4511611	Classic debetní - Partner	CZ730800000000010247XXXXX	451161XXXXXXXXXX	Andrej P.	ACTIVE

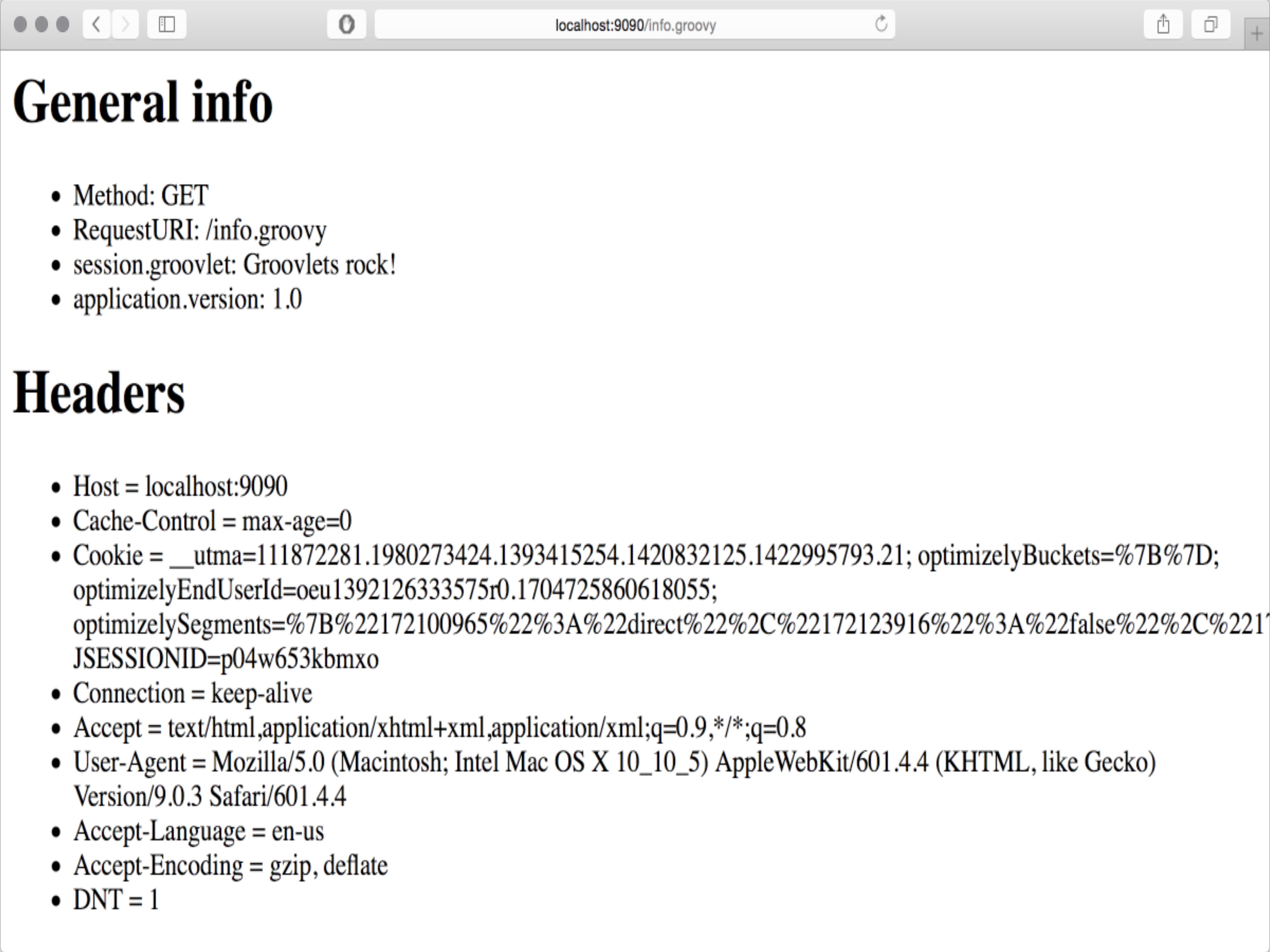
Groovlets

- Implement Java Servlets in Groovy
- Uses GroovyServlet to map servlet request to Groovy scripts
- Usually deployed within servlet container
- But there is a simpler way …

Groovlet Example

```
import groovy.servlet.GroovyServlet
import org.mortbay.jetty.Server
import org.mortbay.jetty.servlet.*

@Grab(group = 'org.mortbay.jetty', module = 'jetty-embedded', version = '6.1.14')

def startJetty() {
    def jetty = new Server(9090)
    def context = new Context(jetty, '/', Context.SESSIONS)
    context.resourceBase = './scripts'
    context.addServlet(GroovyServlet, '*.groovy')
    context.setAttribute('version', '1.0')
    jetty.start()
}

println "Starting Jetty, press Ctrl+C to stop."
startJetty()
```

```
def method = request.method

if (!session) {
    session = request.getSession(true)
}

if (!session.groovlet) {
    session.groovlet = 'Groovlets rock!'
}

html.html {
    head {
        title 'Groovlet info'
    }
    body {
        h1 'General info'
        ul {
            li "Method: ${method}"
            li "RequestURI: ${request.requestURI}"
            li "session.groovlet: ${session.groovlet}"
            li "application.version: ${context.version}"
        }
        h1 'Headers'
        ul {
            headers.each {
                li "${it.key} = ${it.value}"
            }
        }
    }
}
```



General info

- Method: GET
- RequestURI: /info.groovy
- session.groovlet: Groovlets rock!
- application.version: 1.0

Headers

- Host = localhost:9090
- Cache-Control = max-age=0
- Cookie = __utma=111872281.1980273424.1393415254.1420832125.1422995793.21; optimizelyBuckets=%7B%7D; optimizelyEndUserId=oeu1392126333575r0.1704725860618055; optimizelySegments=%7B%22172100965%22%3A%22direct%22%2C%22172123916%22%3A%22false%22%2C%221 JSESSIONID=p04w653kbmxo
- Connection = keep-alive
- Accept = text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
- User-Agent = Mozilla/5.0 (Macintosh; Intel Mac OS X 10_10_5) AppleWebKit/601.4.4 (KHTML, like Gecko) Version/9.0.3 Safari/601.4.4
- Accept-Language = en-us
- Accept-Encoding = gzip, deflate
- DNT = 1

Things To Take Home

Things To Take Home

- Groovy is easy for Java developers
- Groovy is nice for non-trivial scripts
- Groovy works with your favourite libraries
- IDE allows debugging Groovy scripts
- The "Groovy Goodness Notebook"

330
PAGES

Groovy Goodness Notebook

Experience the Groovy programming language through code snippets

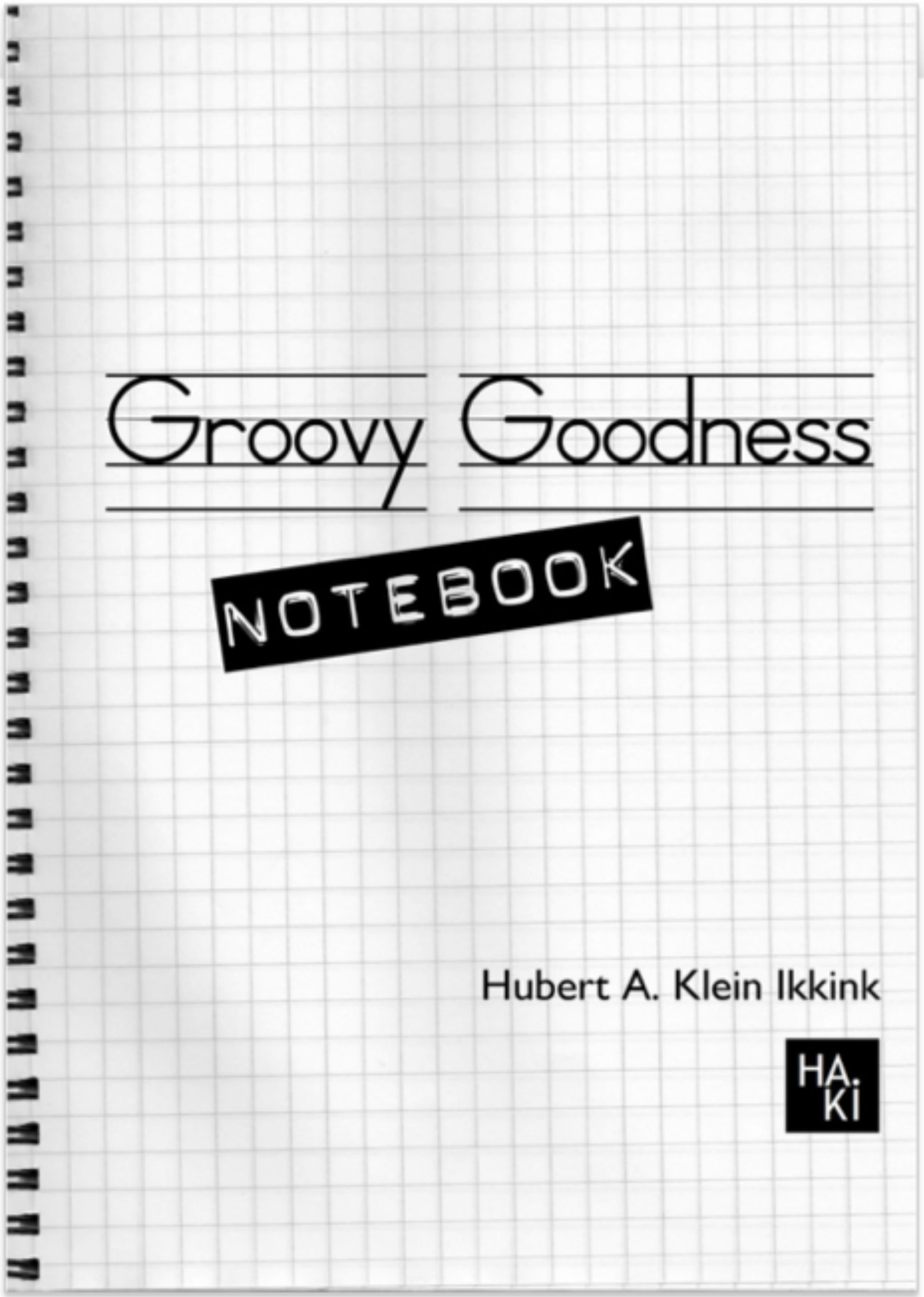
Hubert A. Klein Ikkink (mrhaki)

Learn more about (hidden) Groovy features with code snippets and short articles. The articles and code will get you started quickly and will give more insight in Groovy.

FREE SAMPLE

Read Download

You Own This!



UPDATED 25 DAYS AGO

Things To Take Home

- ASF top-level project
- Embedded in applications
  - SoapUI
- Groovy for integration
  - Spring Boot, Apache Camel
- Grails Web Framework

Questions & Answers

Resources

- http://www.groovy-lang.org
- https://github.com/sgoeschl/groovy-scripting
- https://github.com/sgoeschl/velocity-cli
- https://leanpub.com/groovy-goodness-notebook
- https://github.com/jayway/JsonPath
- http://velocity.apache.org
- https://github.com/jgritman/httpbuilder