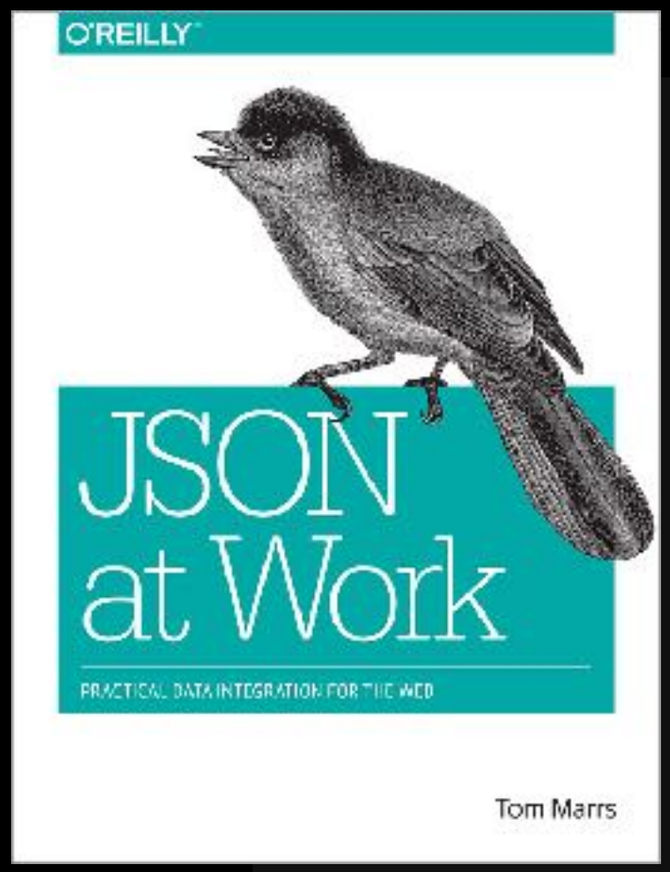
Bitter JSON

(Or When Complex Data Happens to Good Developers)

Tom Marrs

About Me …


O'REILLY
JSON at Work
PRACTICAL DATA INTEGRATION FOR THE WEB
Tom Marrs

O'REILLY
oscon
open source convention

NFJS
No Fluff Just Stuff.com

GREAT INDIAN
DEVELOPER
SUMMIT

dzone
fresh links for developers
REFCARDZ

HTML
5

What's The Point?

JSON Search and Transform

Simplify interaction with RESTful APIs

Your Takeaway

jq + Handlebars Rock!

Agenda

JSON Search & Transform Overview 1

JSON Search 2

JSON Transform 3

We’re Not Covering

REST

Deep JS

Other Languages

Examples and Slides

https://github.com/tmarrs/presentations/tree/master/HTML5/2017/Bitter-JSON

Where Are We?

JSON Search & Transform Overview 1

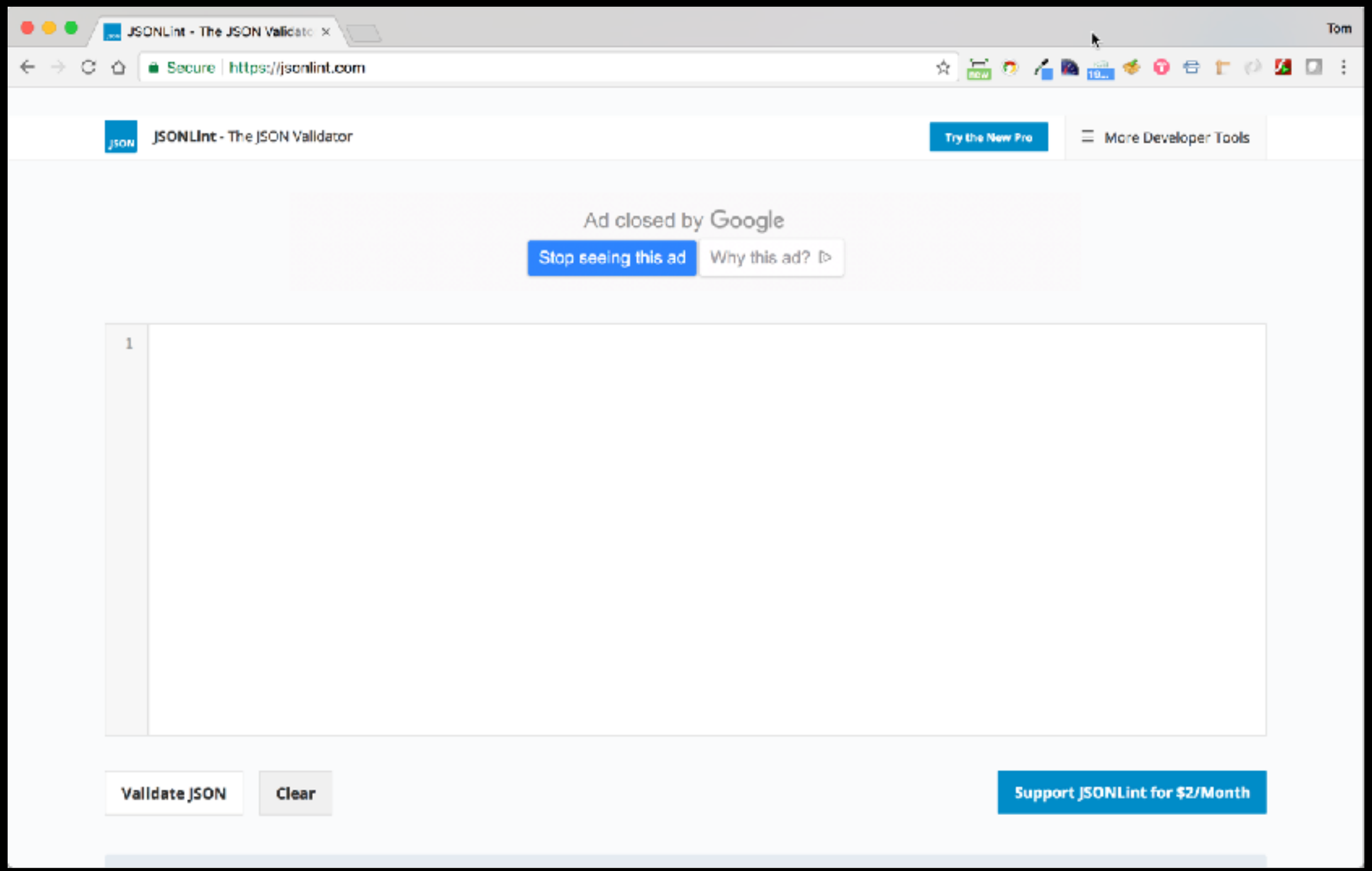
JSON Search 2

JSON Transform 3

The Journey …

Our Client Stack

JSONLint - Validation



JSON Search & Transform Rubric

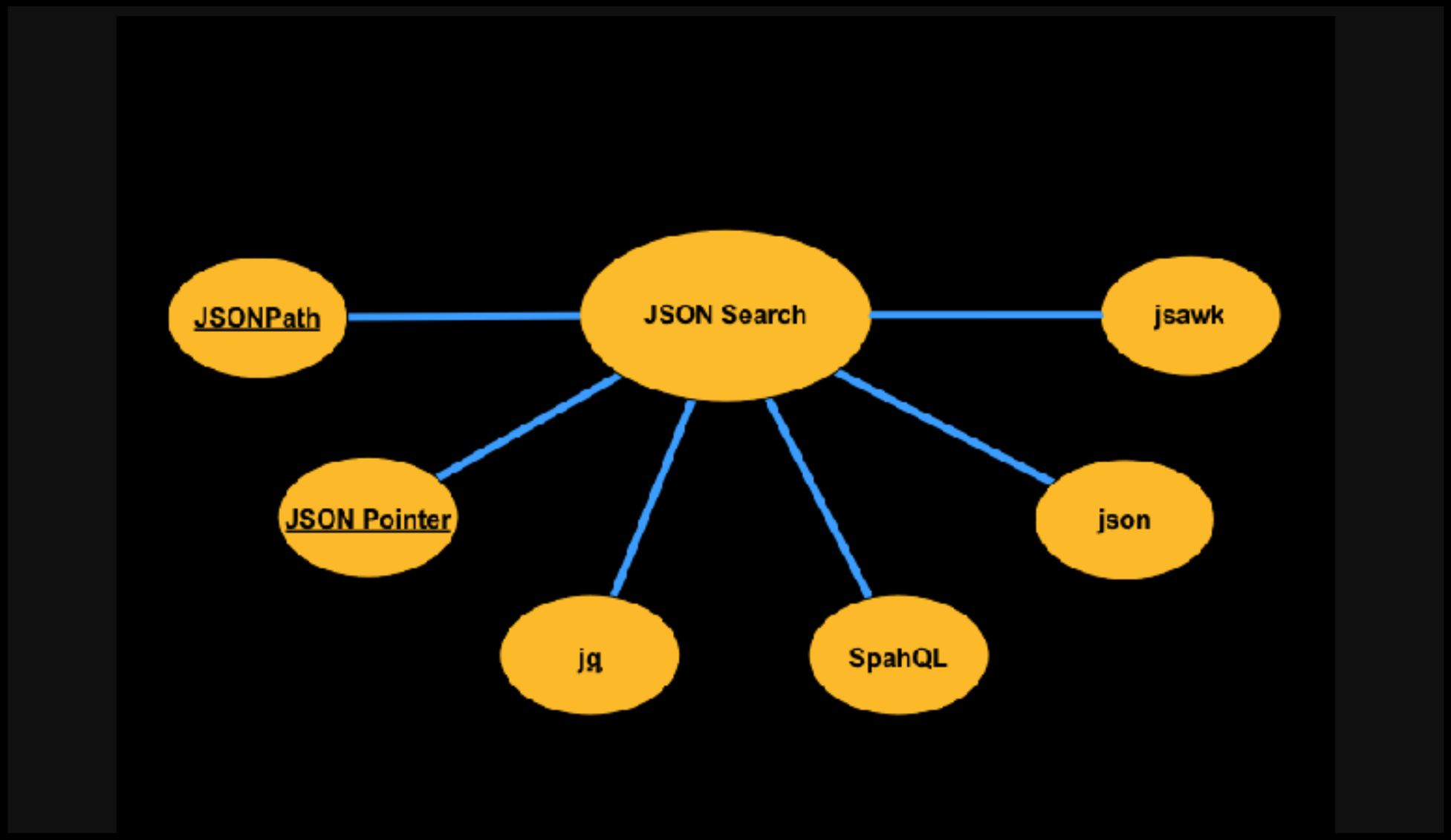
Where Are We?

JSON Search & Transform Overview 1

JSON Search 2

JSON Transform 3

JSON Search Tools

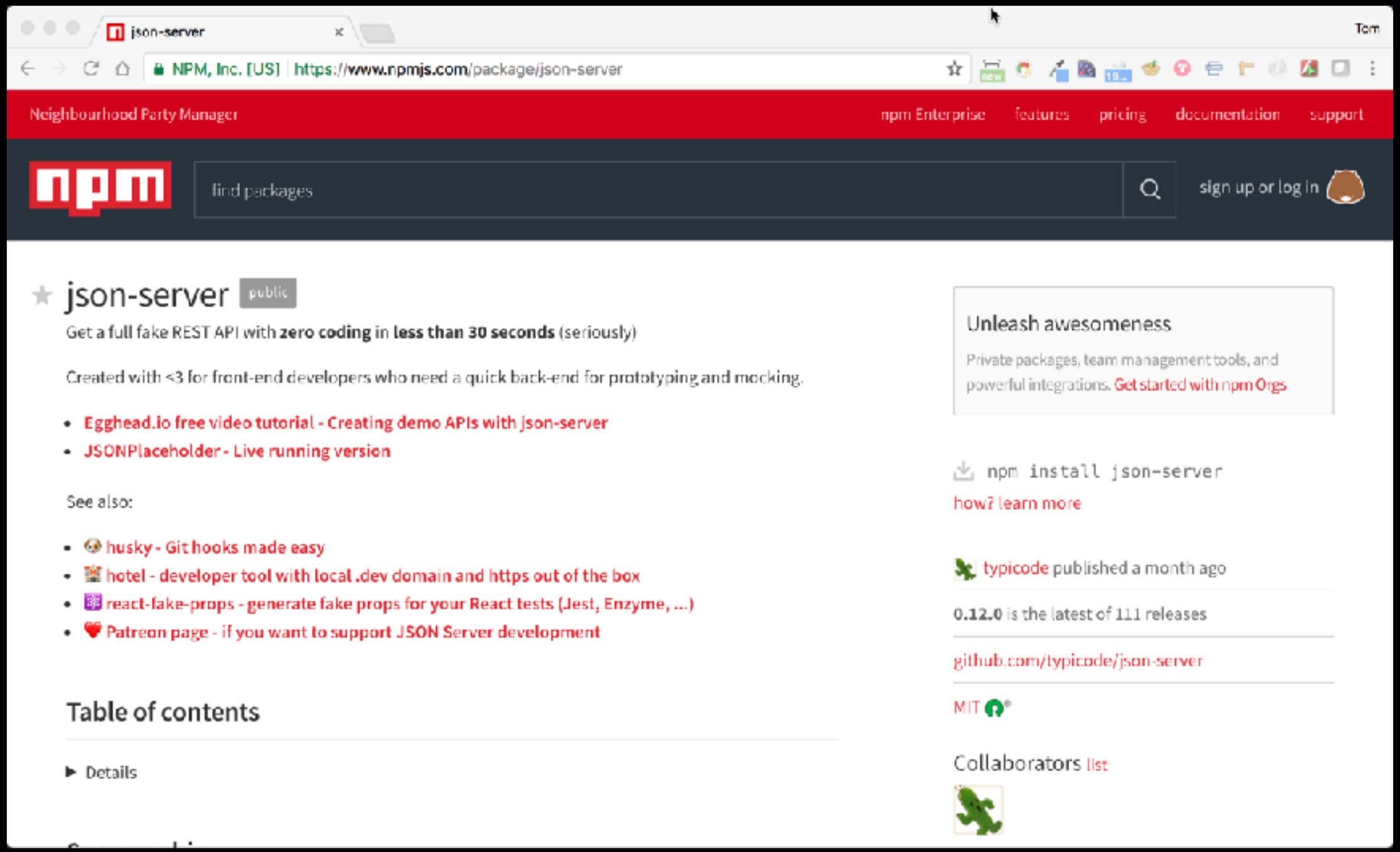


Weather Data API - CLI

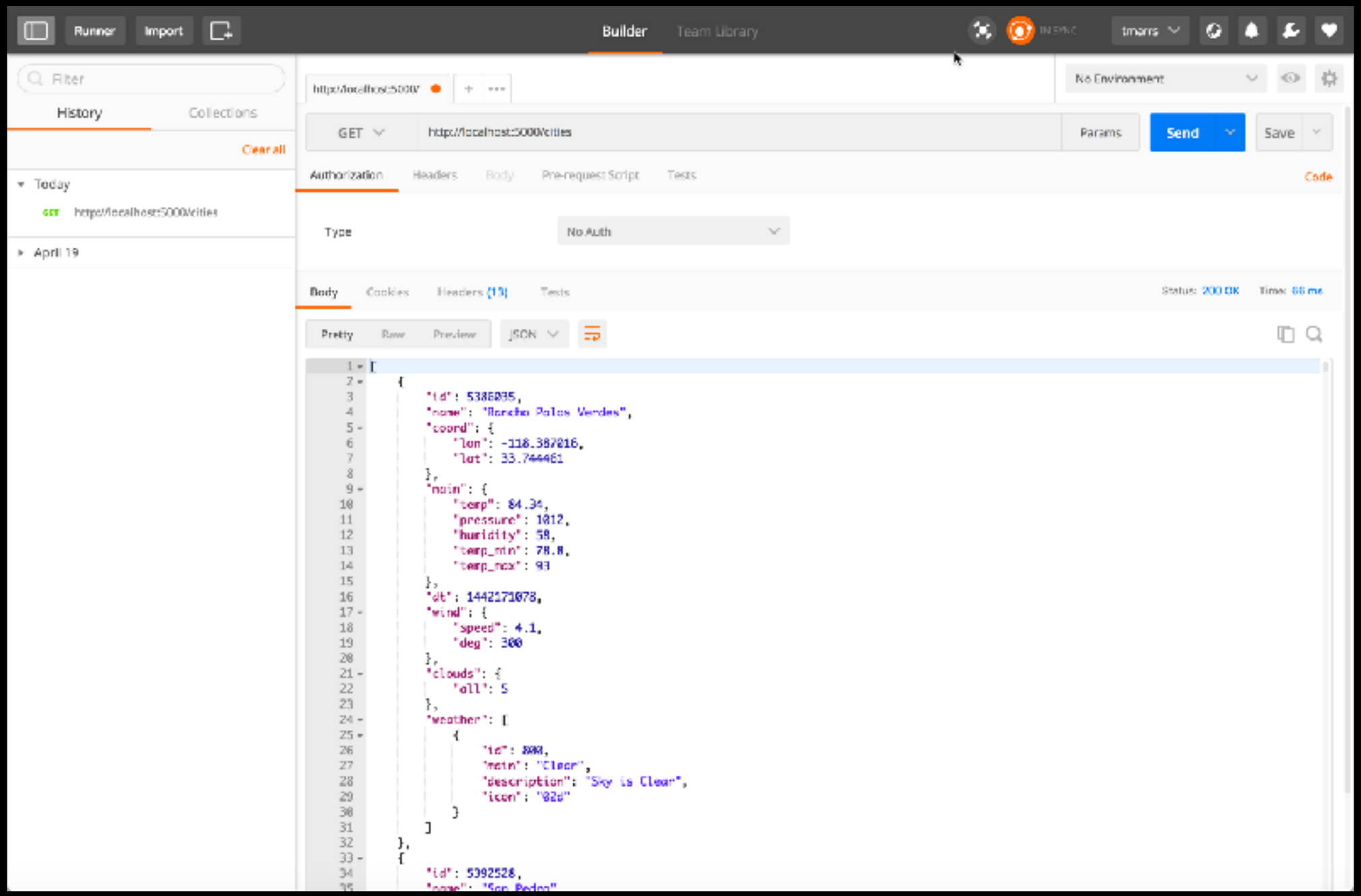
```
json-server -p 5000 \
cities-weather.json
```

http://localhost:5000/cities

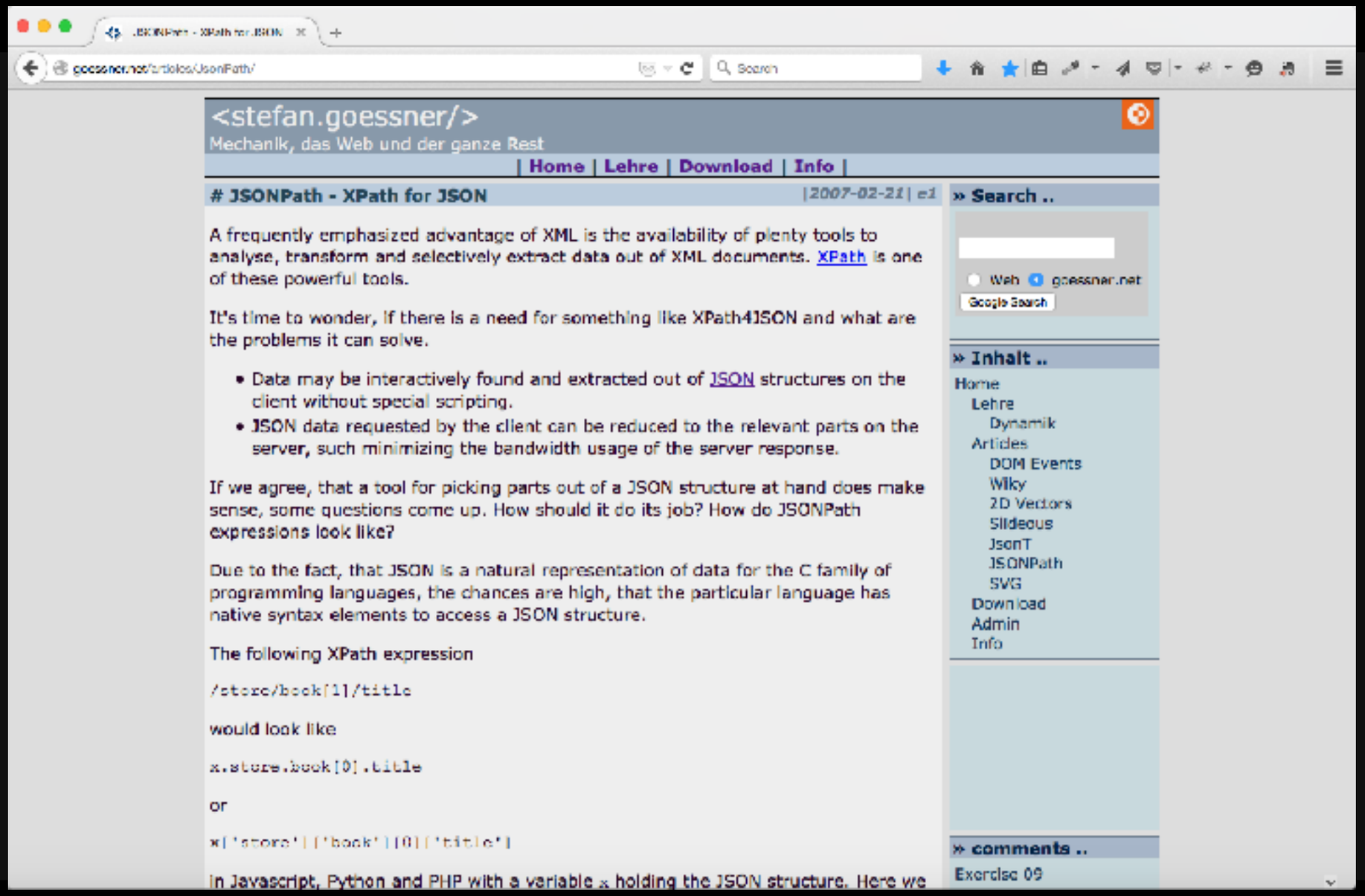
json-server on npm



OpenWeather API - Postman



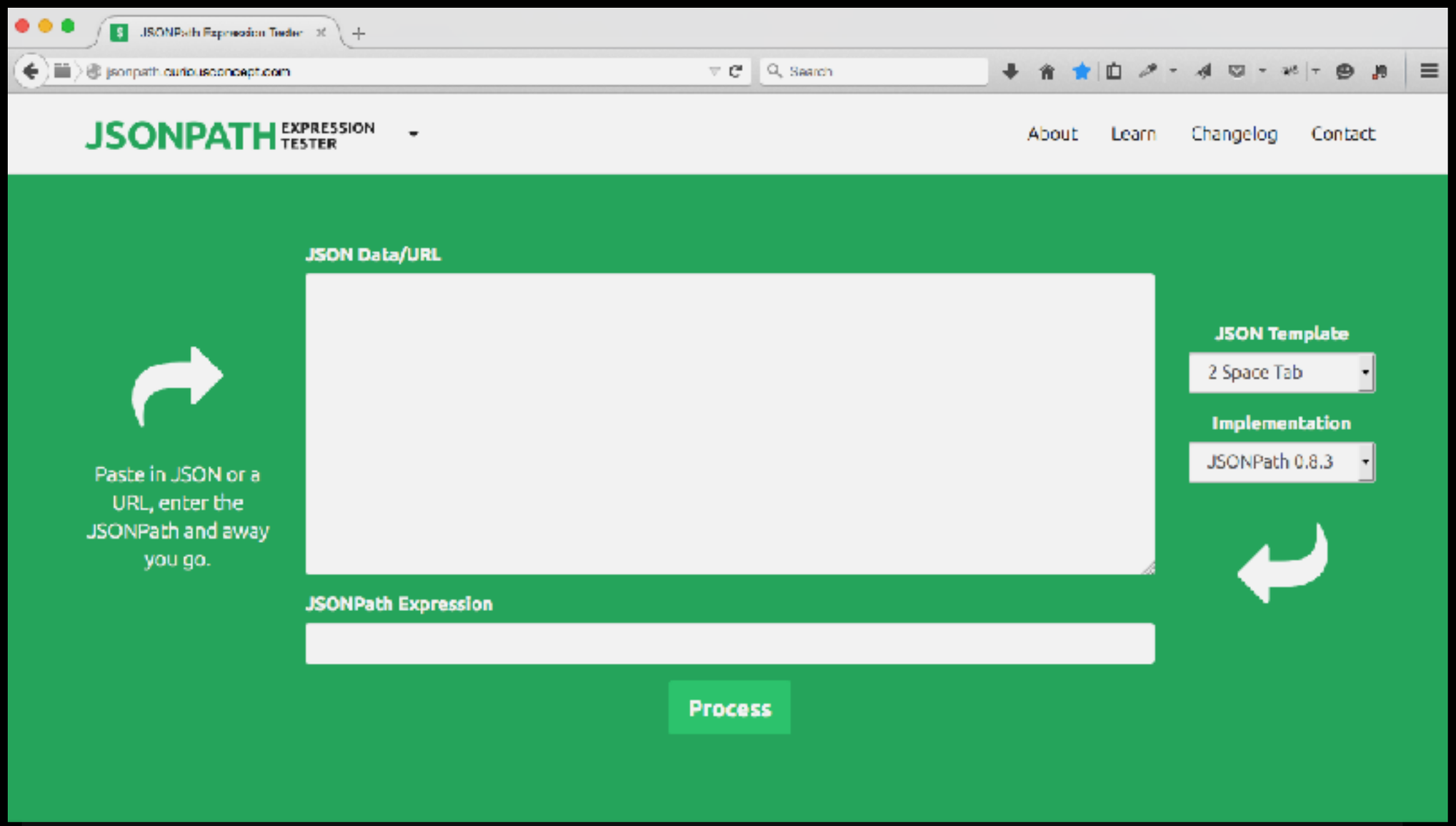
JSONPath


<stefan.goessner/>
Mechanik, das Web und der ganze Rest
| Home | Lehre | Download | Info |
JSONPath - XPath for JSON
|2007-02-21| e1
A frequently emphasized advantage of XML is the availability of plenty tools to analyse, transform and selectively extract data out of XML documents. XPath is one of these powerful tools.
It's time to wonder, if there is a need for something like XPath4JSON and what are the problems it can solve.
• Data may be interactively found and extracted out of JSON structures on the client without special scripting.
• JSON data requested by the client can be reduced to the relevant parts on the server, such minimizing the bandwidth usage of the server response.
If we agree, that a tool for picking parts out of a JSON structure at hand does make sense, some questions come up. How should it do its job? How do JSONPath expressions look like?
Due to the fact, that JSON is a natural representation of data for the C family of programming languages, the chances are high, that the particular language has native syntax elements to access a JSON structure.
The following XPath expression
/store/book[1]/title
would look like
x.store.book[0].title
or
x['store']['book'][0]['title']
» Search ..
Web goessner.net
Google Search
» Inhalt ..
Home
Lehre
Dynamik
Artikes
DOM Events
Wiky
2D Vectors
Slideous
JsonT
JSONPath
SVG
Download
Admin
Info
» comments ..
Exercise 09

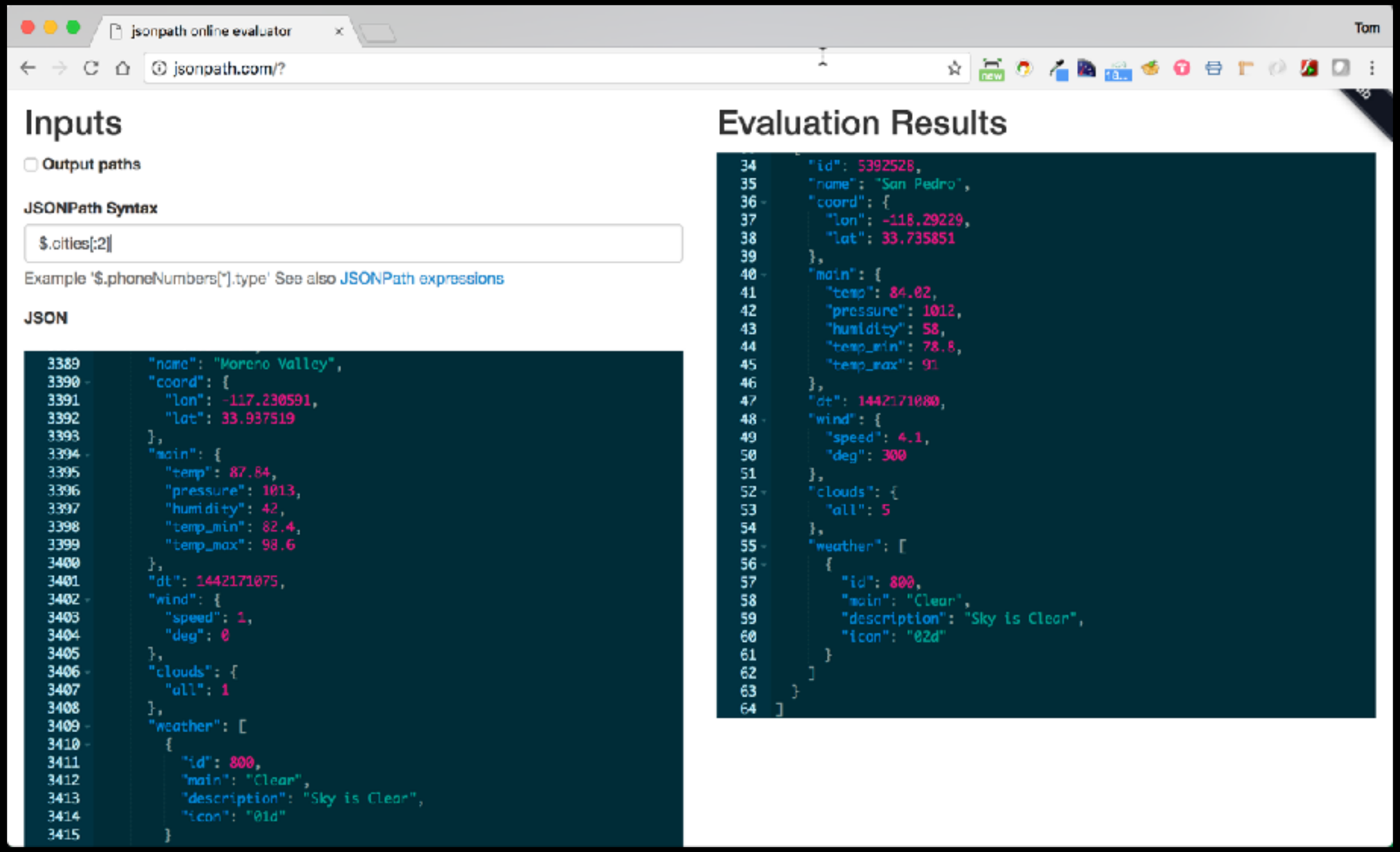
JSONPath Syntax

XPath	JSONPath	Result
/store/book/author	$.store.book[*].author	the authors of all books in the store
//author	$..author	all authors
/store/*	$.store.*	all things in store, which are some books and a red bicycle.
/store//price	$.store..price	the price of everything in the store.
//book[3]	$..book[2]	the third book
//book[last()]	$..book[(@.length-1)] $..book[-1:]	the last book in order.
//book[position()<3]	$..book[0,1] $..book[:2]	the first two books
//book[isbn]	$..book[?(@.isbn)]	filter all books with isbn number
//book[price<10]	$..book[?(@.price<10)]	filter all books cheapier than 10
//*	$..*	all Elements in XML document. All members of JSON structure.

JSONPath Expression Tester



JSONPath Online Evaluator



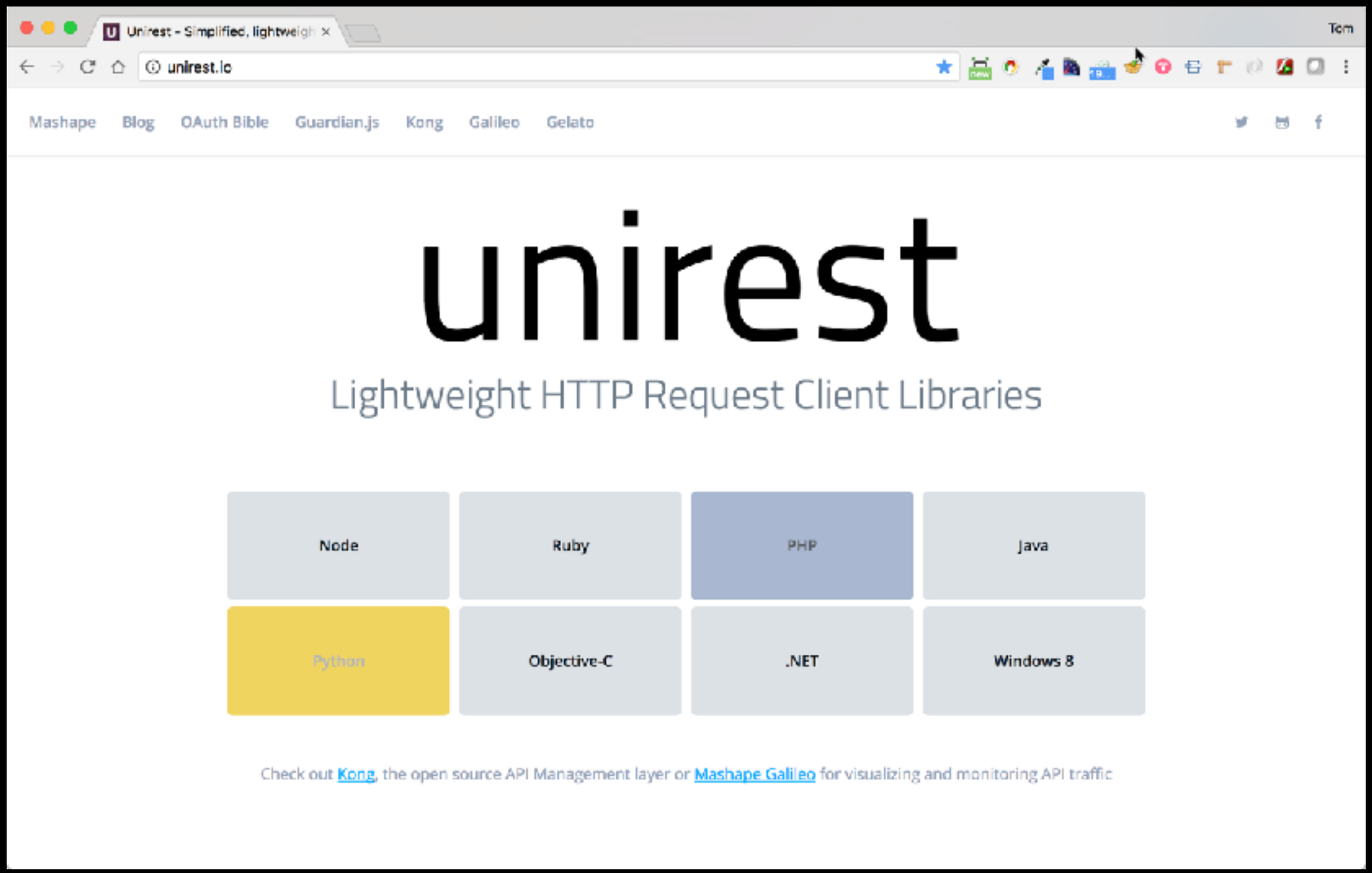

```
3389      "name": "Moreno Valley",
3390      "coord": {
3391        "lon": -117.230591,
3392        "lat": 33.937519
3393      },
3394      "main": {
3395        "temp": 87.84,
3396        "pressure": 1013,
3397        "humidity": 42,
3398        "temp_min": 82.4,
3399        "temp_max": 98.6
3400      },
3401      "dt": 1442171075,
3402      "wind": {
3403        "speed": 1,
3404        "deg": 0
3405      },
3406      "clouds": {
3407        "all": 1
3408      },
3409      "weather": [
3410        {
3411          "id": 800,
3412          "main": "Clear",
3413          "description": "Sky is Clear",
3414          "icon": "01d"
3415        }
```

```
34      "id": 5392528,
35      "name": "San Pedro",
36      "coord": {
37        "lon": -118.29229,
38        "lat": 33.735851
39      },
40      "main": {
41        "temp": 84.02,
42        "pressure": 1012,
43        "humidity": 58,
44        "temp_min": 78.8,
45        "temp_max": 91
46      },
47      "dt": 1442171080,
48      "wind": {
49        "speed": 4.1,
50        "deg": 300
51      },
52      "clouds": {
53        "all": 5
54      },
55      "weather": [
56        {
57          "id": 800,
58          "main": "Clear",
59          "description": "Sky is Clear",
60          "icon": "02d"
61        }
62      ]
63    }
64  ]
```

JSONPath Queries

```
JSONPath query                    Description
-----------------------------------------
$.cities                          Get all elements in the cities Array.
$.cities.length                   Get the number of elements in the cities Array.
$.cities[0::2]                    Get every other element in the cities array.

$.cities[(@.length-1)]            Get the last element in the cities Array.

$..weather                        Get all weather subelements.
$.cities[:3]                      Get the first three elements in cities Array.
$.cities[:3].name                 Get the city name for first three elements in the cities Array.
$.cities[?(@.main.temp > 84)]     Get the cities where the temp > 84.

$.cities[?(@.main.temp >= 84 && @.main.temp <= 85.5)]
                                  Get the cities where the temp is between 84 and 85.5.

$.cities[?(@.weather[0].main == 'Clouds')]
                                  Get the cities with cloudy weather.

$.cities[?(@.weather[0].main.match(/Clo/))]
                                  Get the cities with cloudy weather by using regex.
```

Unirest


Unirest - Simplified, lightweigh
Tom
unirest.io
Mashape
Blog
OAuth Bible
Guardian.js
Kong
Galileo
Gelato
unirest
Lightweight HTTP Request Client Libraries
Node
Ruby
PHP
Java
Python
Objective-C
.NET
Windows 8
Check out Kong, the open source API Management layer or Mashape Galileo for visualizing and monitoring API traffic

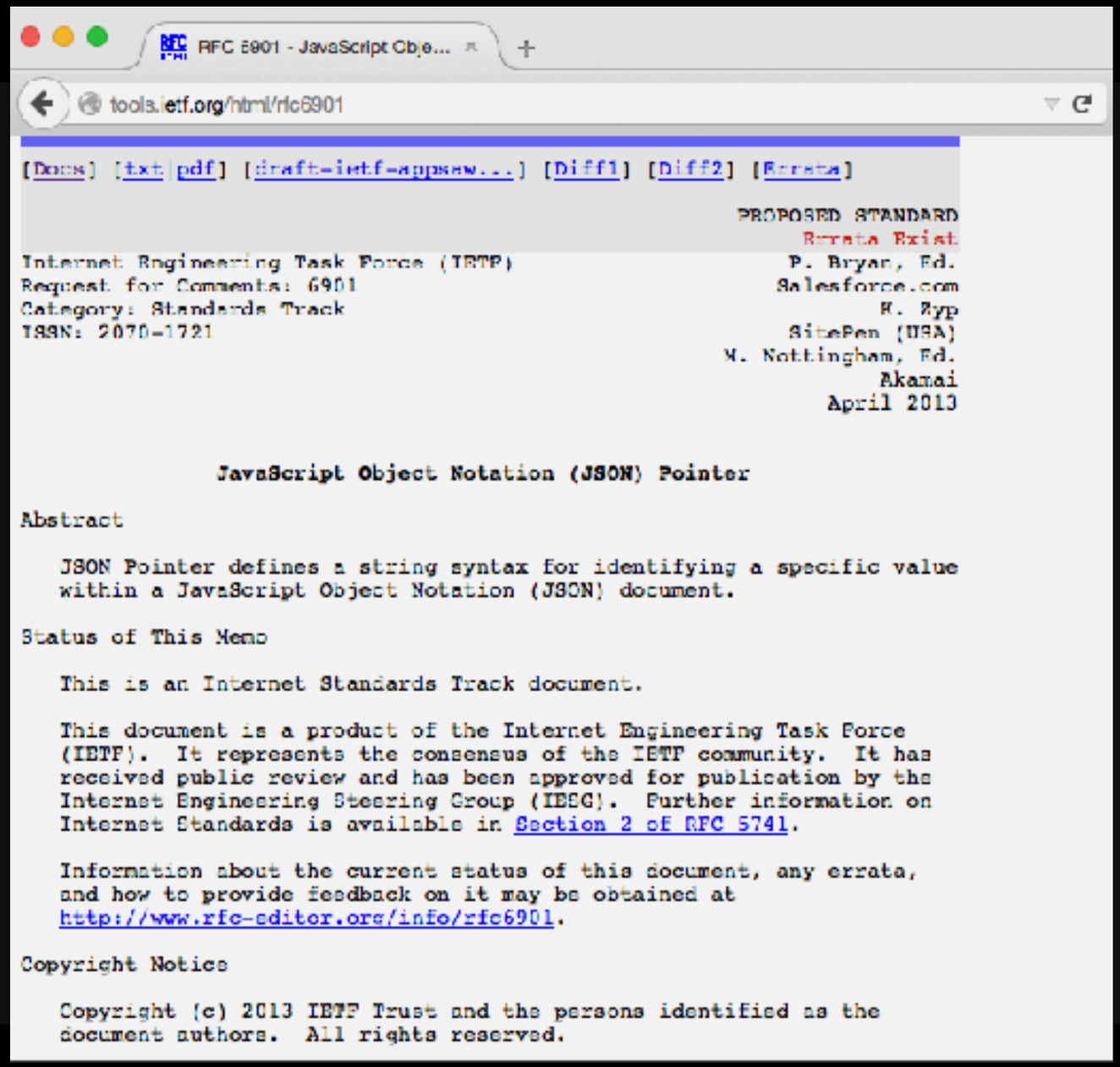
JSONPath Test

```
12  var expect = require('chai').expect;
13  var jp = require('jsonpath');
14  var unirest = require('unirest');
15
16  describe('cities-jsonpath', function() {
17    var req;
18
19    beforeEach(function() {
20      req = unirest.get('http://localhost:5000/cities')
21        .header('Accept', 'application/json');
22    });
23
24    it('should return a 200 response', function(done) {
25      req.end(function(res) {
26        expect(res.statusCode).to.eql(200);
27        expect(res.headers['content-type']).to.eql(
28          'application/json; charset=utf-8');
29        done();
30      });
31    });
32
33    it('should return all cities', function(done) {
34      req.end(function(res) {
35        var cities = res.body;
36
37        //console.log(cities);
38        expect(cities.length).to.eql(110);
39        done();
40      });
41    });
```

JSONPath Scorecard

Mindshare	Y
Dev Community	Y
Platforms	JS, Node.js, Java, RoR
Intuitive	Y
Standard	N

JSON Pointer



RFC 6901 - JavaScript Obje...

tools.ietf.org/html/rfc6901

[Docs] [txt|pdf] [draft-ietf-appsaw...] [Diff1] [Diff2] [Errata]

PROPOSED STANDARD
Errata Exist

Internet Engineering Task Force (IETF)
Request for Comments: 6901
Category: Standards Track
ISSN: 2070-1721

P. Bryan, Ed.
Salesforce.com
K. Zyp
SitePen (USA)
M. Nottingham, Ed.
Akamai
April 2013

JavaScript Object Notation (JSON) Pointer

Abstract

JSON Pointer defines a string syntax for identifying a specific value within a JavaScript Object Notation (JSON) document.

Status of This Memo

This is an Internet Standards Track document.

This document is a product of the Internet Engineering Task Force (IETF). It represents the consensus of the IETF community. It has received public review and has been approved for publication by the Internet Engineering Steering Group (IESG). Further information on Internet Standards is available in Section 2 of RFC 5741.

Information about the current status of this document, any errata, and how to provide feedback on it may be obtained at http://www.rfc-editor.org/info/rfc6901.

Copyright Notice

Copyright (c) 2013 IETF Trust and the persons identified as the document authors. All rights reserved.

JSON Pointer Syntax

For example, given the JSON document

```
{
   "foo": ["bar", "baz"],
   "": 0,
   "a/b": 1,
   "c%d": 2,
   "e^f": 3,
   "g|h": 4,
   "i\\j": 5,
   "k\"l": 6,
   " ": 7,
   "m~n": 8
}
```

The following JSON strings evaluate to the accompanying values:

```
 ""           // the whole document
 "/foo"       ["bar", "baz"]
 "/foo/0"     "bar"
 "/"          0
 "/a~1b"      1
 "/c%d"       2
 "/e^f"       3
 "/g|h"       4
 "/i\\j"      5
 "/k\"l"      6
 "/ "         7
 "/m~0n"      8
```

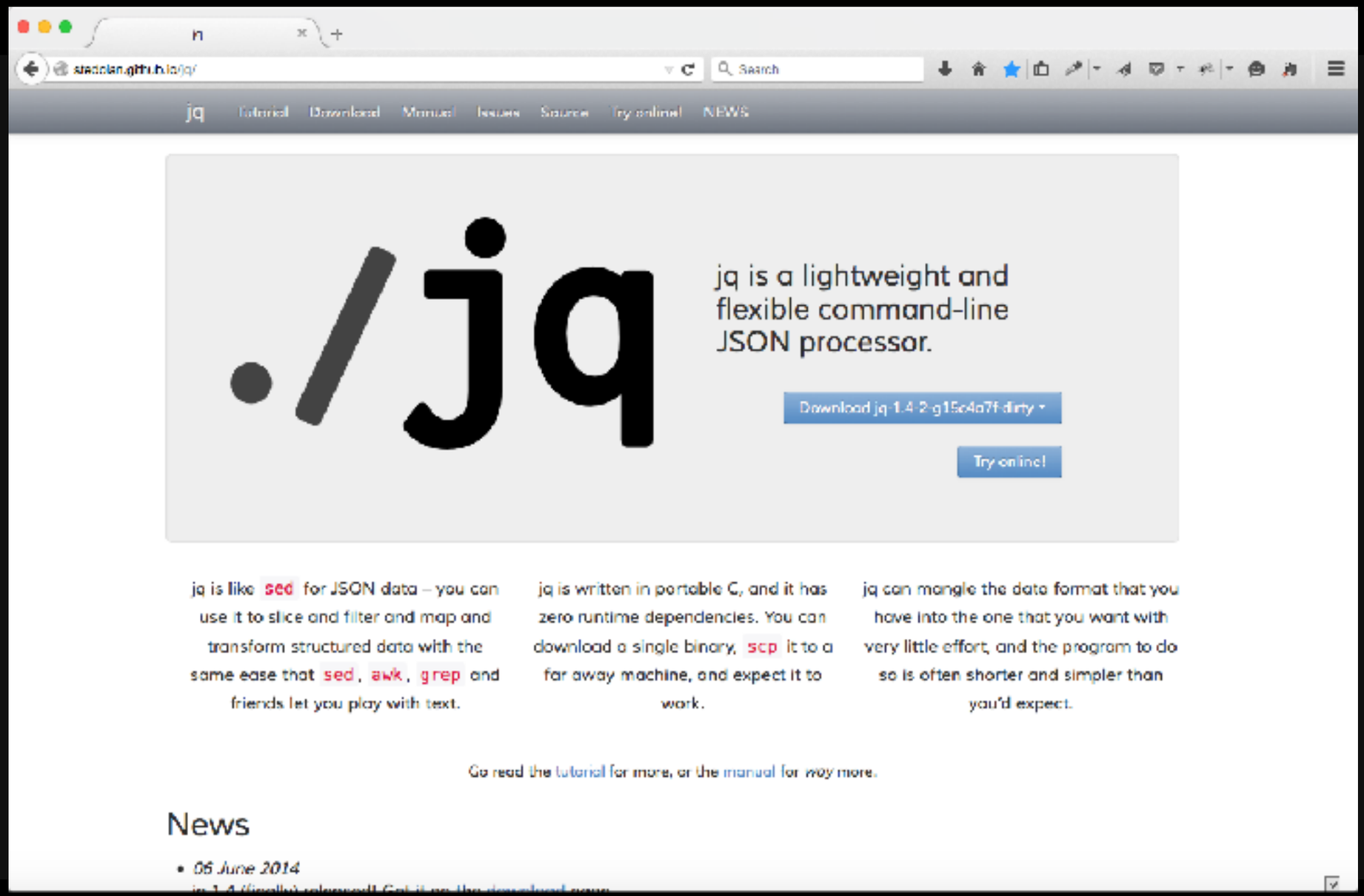
JSON Pointer Test

```
12  var expect = require('chai').expect;
13  var pointer = require('json-pointer');
14  var unirest = require('unirest');
15
16  describe('cities-json-pointer', function() {
17    var req;
18
19    beforeEach(function() {
20      req = unirest.get('http://localhost:5000/cities')
21        .header('Accept', 'application/json');
22    });
23
24    it('should return a 200 response', function(done) {
25      req.end(function(res) {
26        expect(res.statusCode).to.eql(200);
27        expect(res.headers['content-type']).to.eql(
28          'application/json; charset=utf-8');
29        done();
30      });
31    });
32
33    it('should return the 1st city', function(done) {
34      req.end(function(res) {
35        var cities = res.body;
36        var firstCity = null;
37
38        //console.log('\n\n1st Object: ');
39        firstCity = pointer.get(cities, '/0');
40        //console.log(firstCity);
41        expect(firstCity.name).to.eql('Rancho Palos Verdes');
42        expect(firstCity.weather[0].main).to.eql('Clear');
43        done();
44      });
45    });
46
47    it('should return the name of the 2nd city', function(done) {
```

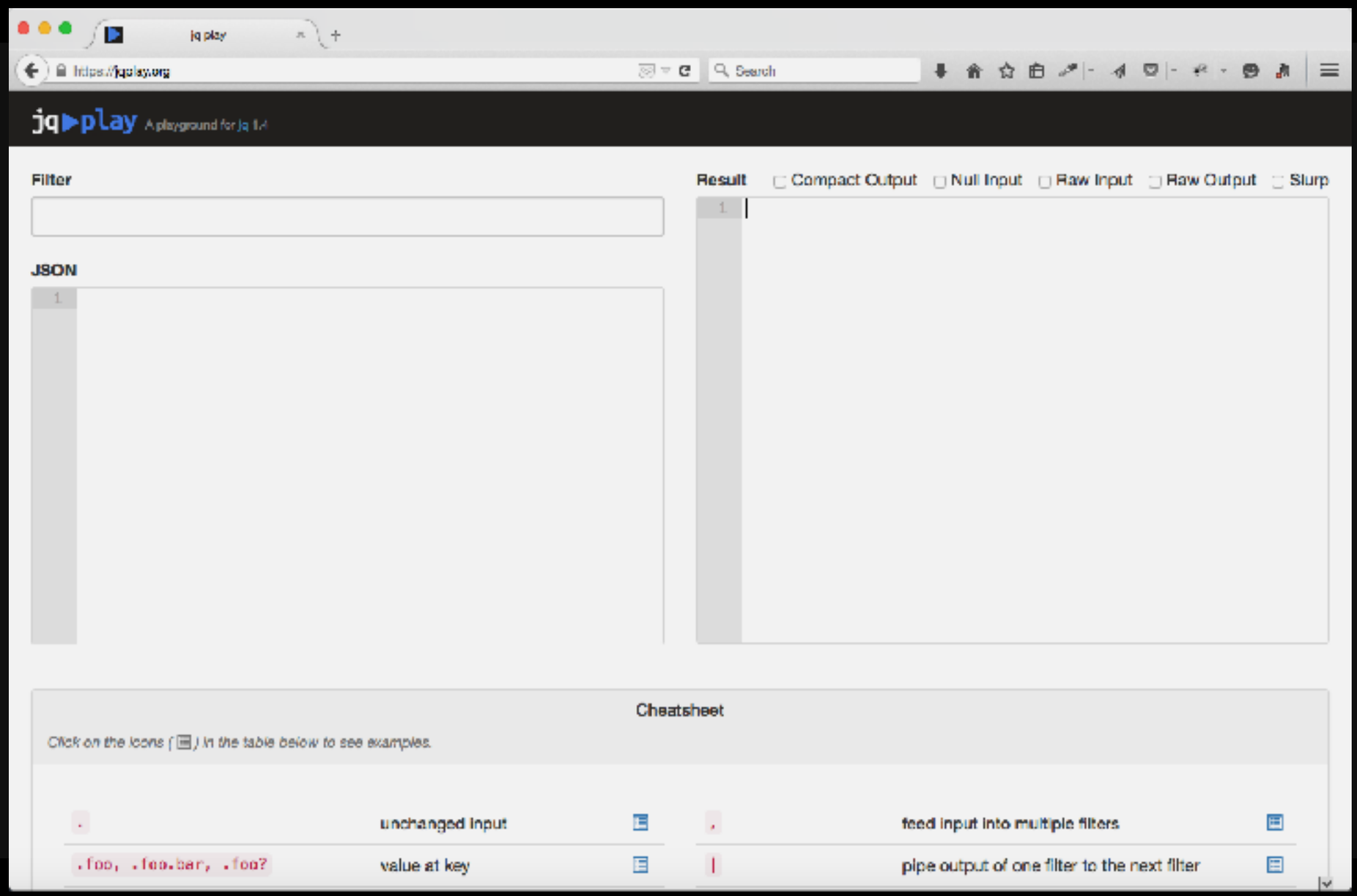
JSON Pointer Scorecard

Mindshare	Y
Dev Community	Y
Platforms	JS, Node.js, Java, RoR
Intuitive	Y
Standard	RFC 6901 - Woot!

jq


jq
Tutorial
Download
Manual
Issues
Source
Try online!
NEWS
./jq
jq is a lightweight and flexible command-line JSON processor.
Download jq-1.4-2-g15c4a7f-dirty
Try online!
jq is like sed for JSON data – you can use it to slice and filter and map and transform structured data with the same ease that sed, awk, grep and friends let you play with text.
jq is written in portable C, and it has zero runtime dependencies. You can download a single binary, scp it to a far away machine, and expect it to work.
jq can mangle the data format that you have into the one that you want with very little effort, and the program to do so is often shorter and simpler than you'd expect.
Go read the tutorial for more, or the manual for way more.
News
06 June 2014

jq play


jq play
https://jqplay.org
Search
jq play A playground for jq 1.4
Filter
JSON
Result
Compact Output
Null Input
Raw Input
Raw Output
Slurp
Cheatsheet
Click on the icons ( ) in the table below to see examples.
. unchanged input
, feed input into multiple filters
.foo, .foo.bar, .foo? value at key
| pipe output of one filter to the next filter

jq Queries

```
jq query Description
--------------------
.cities[0]              Get the first city. jq Array filtering starts at 0.
.cities[-1]             Get the last city. An index of -1 indicates the last element of an Array.
.cities[0:3]            Get the first three cities, where 0 is the start index (inclusive), and 3 is the end
                        index (exclusive).

.cities[:3]             Get the first three cities. This is shorthand, and it omits the start index.

[.cities[:3] | .[] | {name, weather}]
                        Get the name and weather for the first 3 cities and put it in a new array.

.cities[] | select (.main.temp >= 80 and (.main.temp_min >= 79 and .main.temp_max <= 92))
                        Get all cities whose current temperature is >= 80 degrees Fahrenheit and whose
                        min and max temperature ranges between 79 and 92 degrees Fahrenheit
                        (inclusive).
```

jq Test

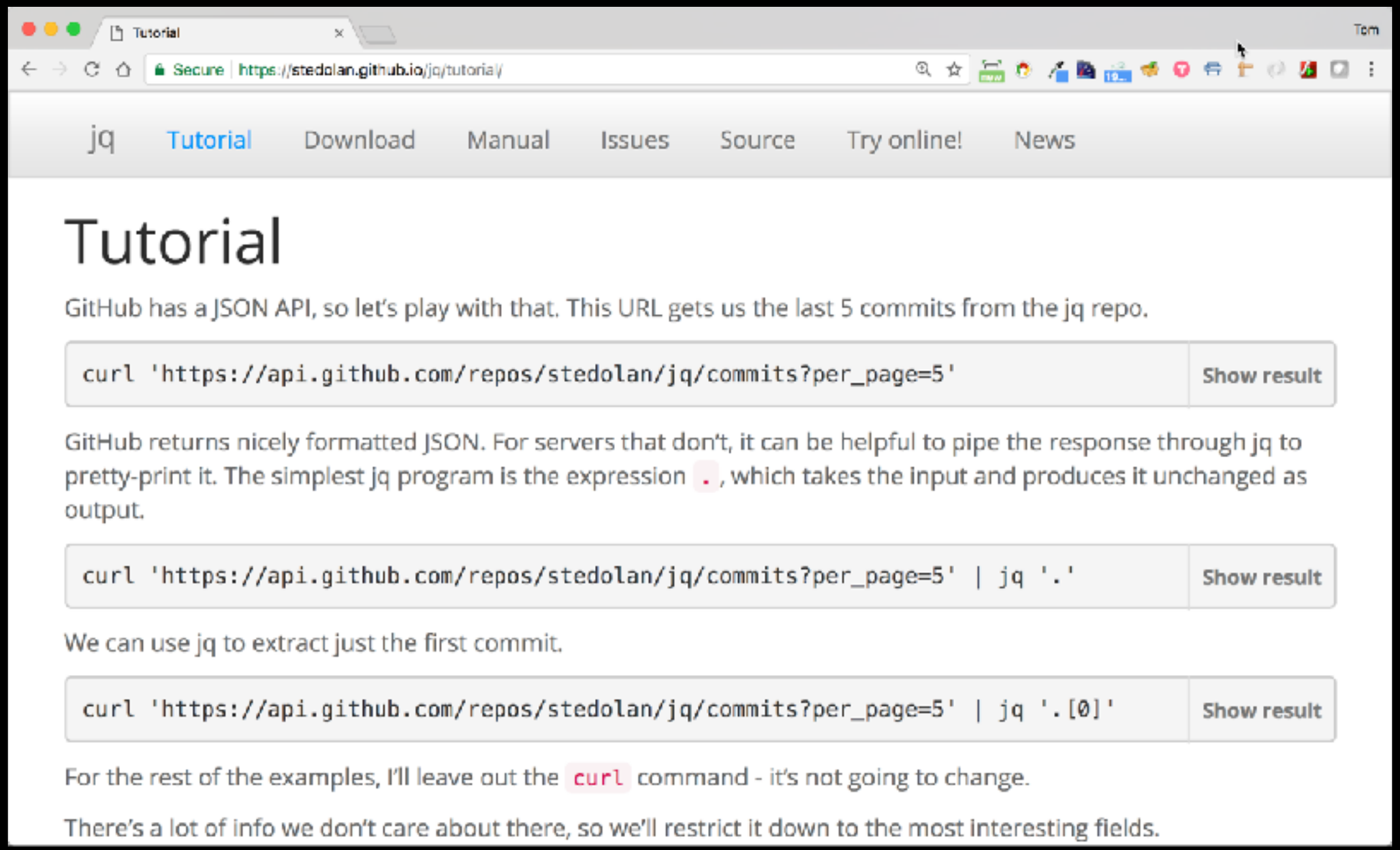
```
12  var expect = require('chai').expect;
13  var jq = require('node-jq');
14  var unirest = require('unirest');
15  var _ = require('underscore');
16
17
18  describe('cities-jq', function() {
19    var req;
20
21    beforeEach(function() {
22      req = unirest.get('http://localhost:5000/cities')
23        .header('Accept', 'application/json');
24    });
25
26    it('should return a 200 response', function(done) {
27      req.end(function(res) {
28        expect(res.statusCode).to.eql(200);
29        expect(res.headers['content-type']).to.eql(
30          'application/json; charset=utf-8');
31        done();
32      });
33    });
34
35    it('should return all cities', function(done) {
36      req.end(function(res) {
37        var cities = res.body;
38
39        //console.log(cities);
40        expect(cities.length).to.eql(110);
41        done();
42      });
43    });
```

jq and cURL

```
curl -X GET 'http://localhost:
5000/cities'
```

```
curl -X GET 'http://localhost:
5000/cities' | jq .[0]
```

jq Tutorial


Tutorial
Tom
Secure
https://stedolan.github.io/jq/tutorial/
jq
Tutorial
Download
Manual
Issues
Source
Try online!
News
Tutorial
GitHub has a JSON API, so let's play with that. This URL gets us the last 5 commits from the jq repo.
curl 'https://api.github.com/repos/stedolan/jq/commits?per_page=5'
Show result
GitHub returns nicely formatted JSON. For servers that don't, it can be helpful to pipe the response through jq to pretty-print it. The simplest jq program is the expression ., which takes the input and produces it unchanged as output.
curl 'https://api.github.com/repos/stedolan/jq/commits?per_page=5' | jq '.'
Show result
We can use jq to extract just the first commit.
curl 'https://api.github.com/repos/stedolan/jq/commits?per_page=5' | jq '.[0]'
Show result
For the rest of the examples, I'll leave out the curl command - it's not going to change.
There's a lot of info we don't care about there, so we'll restrict it down to the most interesting fields.

jq-tutorial

```
json-at-work => jq-tutorial
Run jq-tutorial with one of the following:
  * pick
  * objects
  * mapping
  * filtering
  * output
  * reduce
```

jq Scorecard

Mindshare	Y
Dev Community	Y
Platforms	CLI - Linux/Mac/ Windows, JS, Node.js, Java, RoR
Intuitive	Y
Standard	N

My JSON Search Choices

Tool	Rank
jq	1
JSONPath	2
JSON Pointer	3

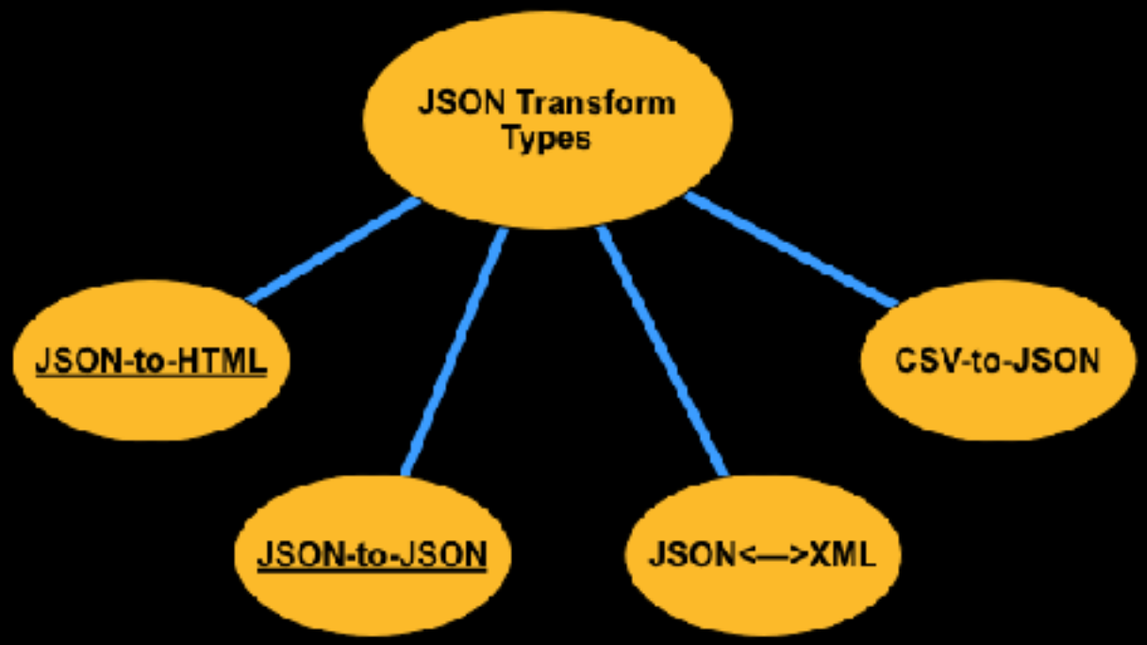
Where Are We?

JSON Search & Transform Overview 1

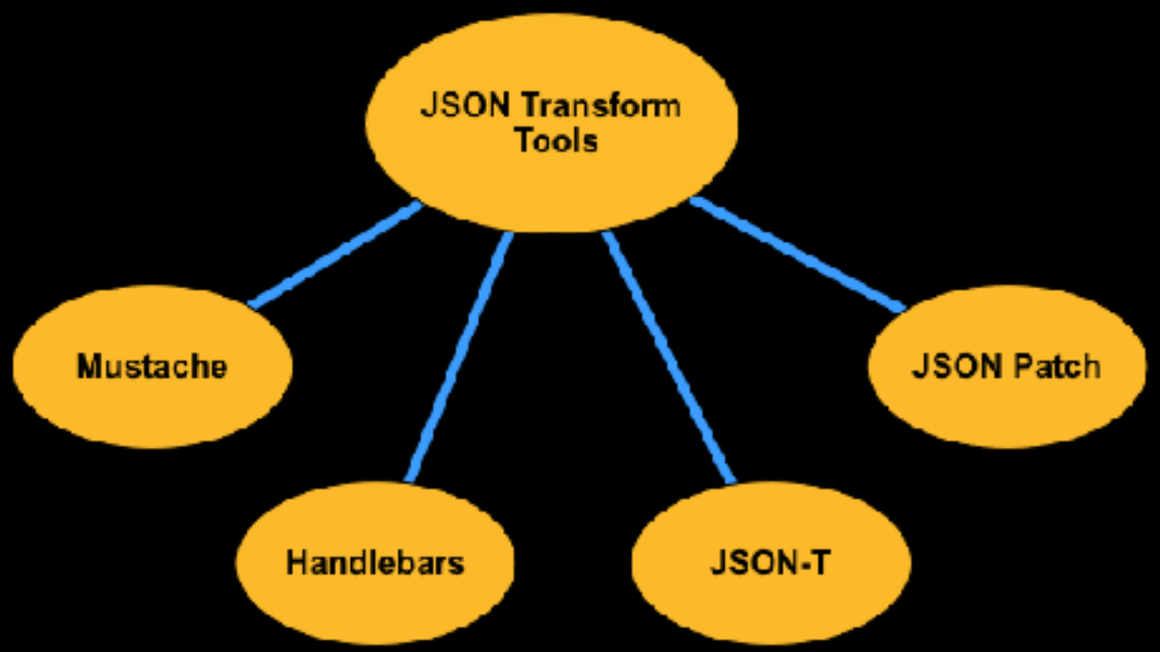
JSON Search 2

JSON Transform 3

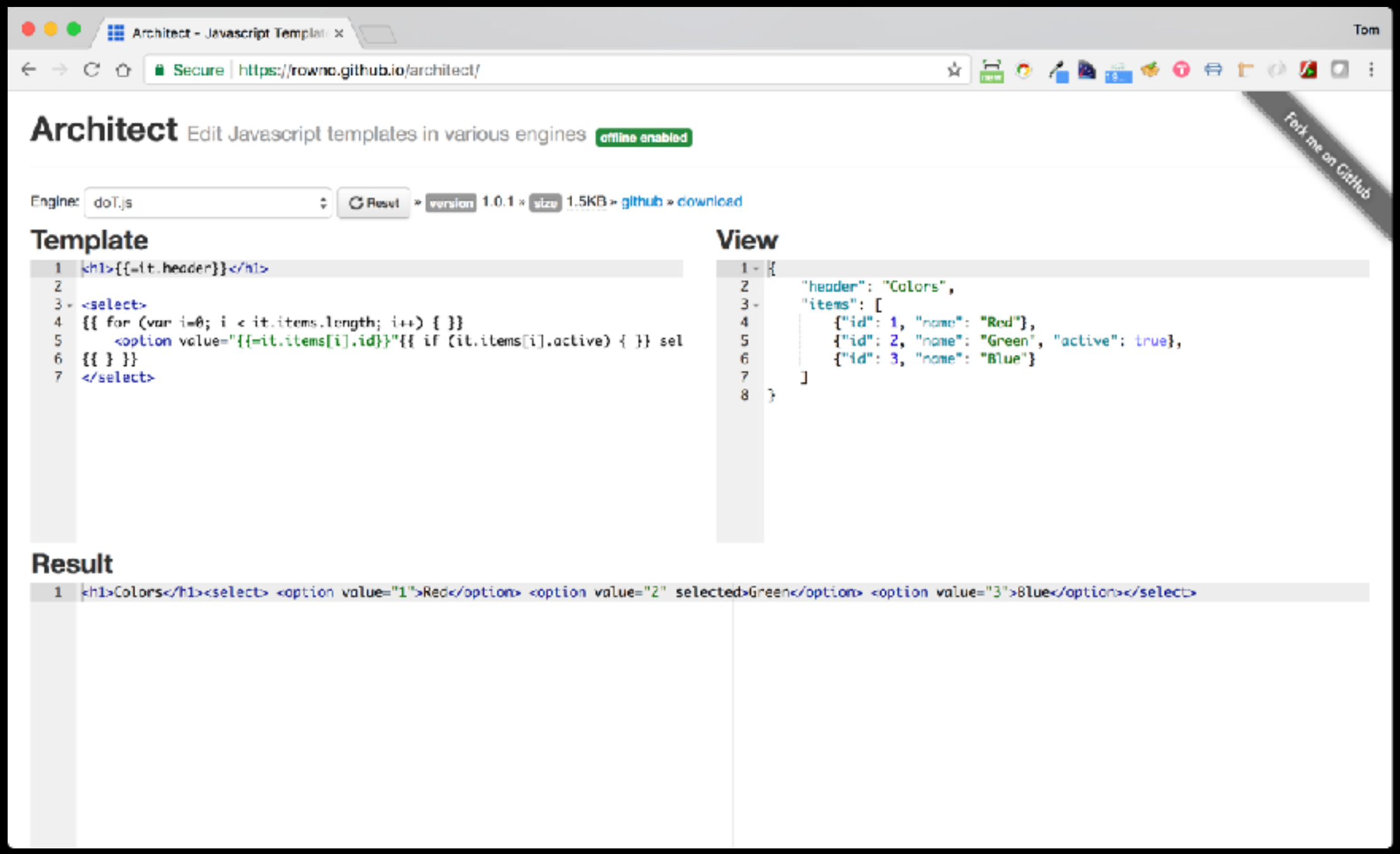
JSON Transform Types



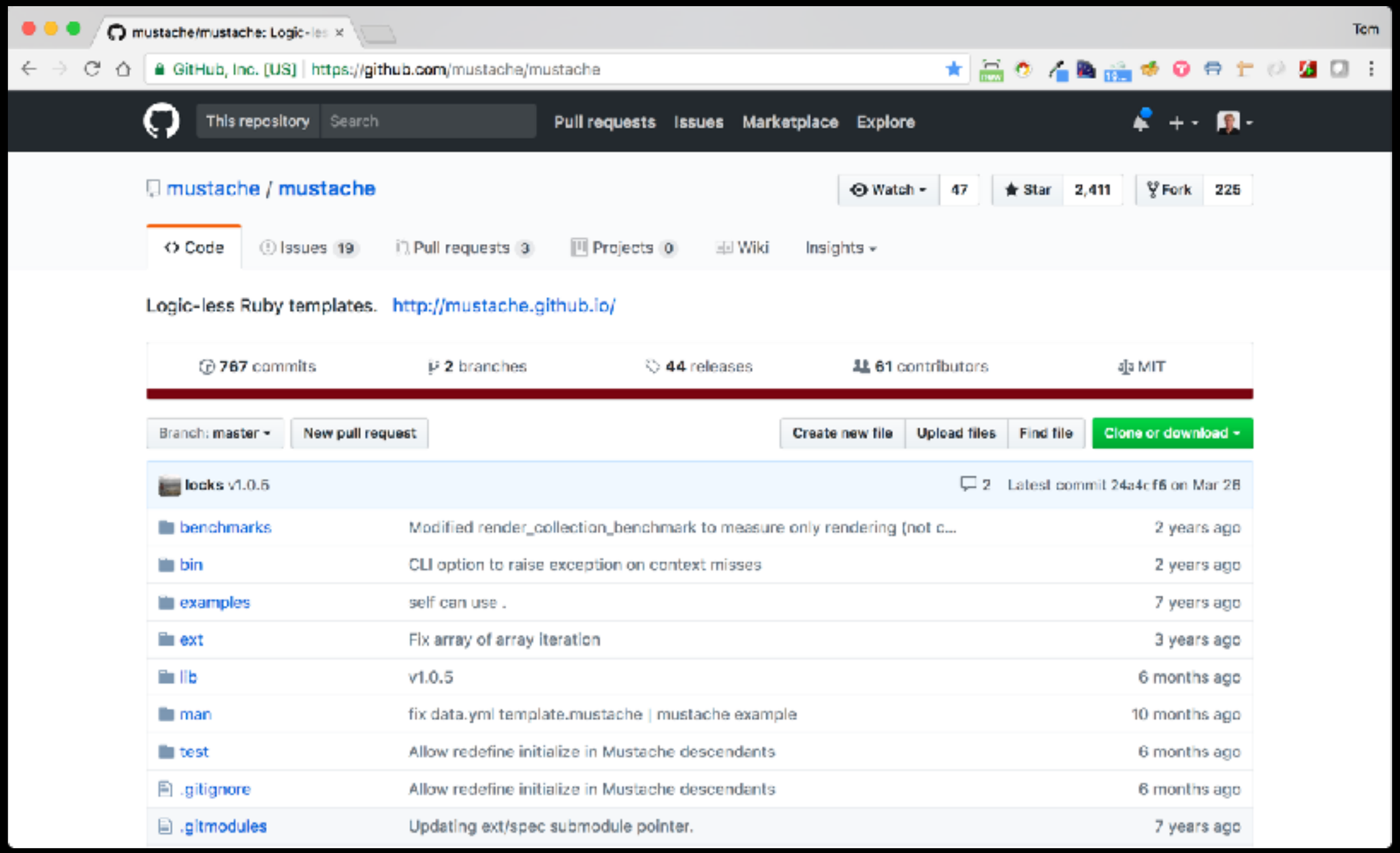
JSON Transform Tools



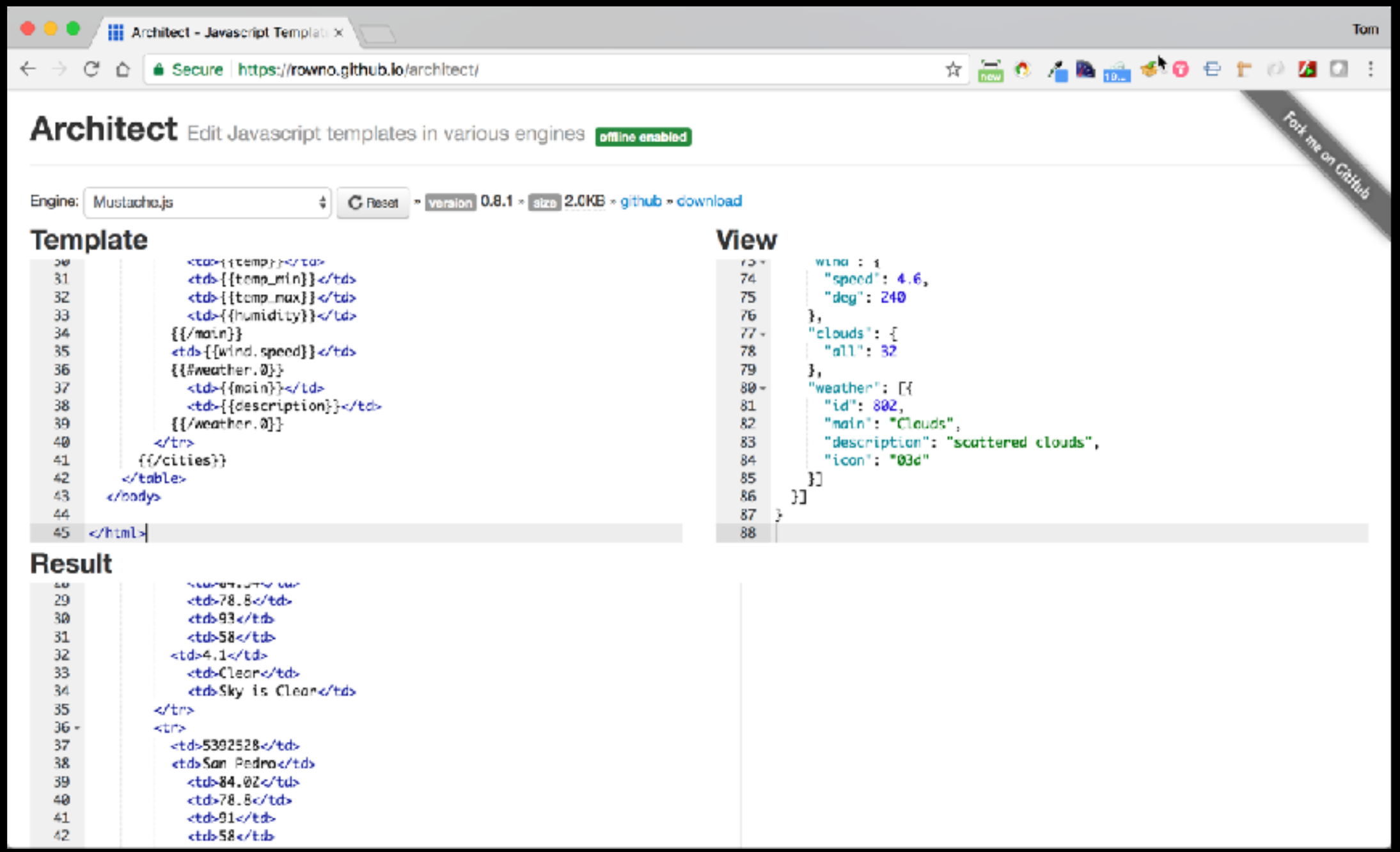
Architect - JS Template Editor



Mustache


mustache/mustache: Logic-
Tom
GitHub, Inc. [US] https://github.com/mustache/mustache
This repository Search
Pull requests Issues Marketplace Explore
mustache / mustache
Watch 47
Star 2,411
Fork 225
Code
Issues 19
Pull requests 3
Projects 0
Wiki
Insights
Logic-less Ruby templates. http://mustache.github.io/
767 commits
2 branches
44 releases
61 contributors
MIT
Branch: master
New pull request
Create new file
Upload files
Find file
Clone or download
locks v1.0.5
2
Latest commit 24a4cf6 on Mar 28
benchmarks
Modified render_collection_benchmark to measure only rendering (not c...
2 years ago
bin
CLI option to raise exception on context misses
2 years ago
examples
self can use .
7 years ago
ext
Fix array of array iteration
3 years ago
lib
v1.0.5
6 months ago
man
fix data.yml template.mustache | mustache example
10 months ago
test
Allow redefine initialize in Mustache descendants
6 months ago
.gitignore
Allow redefine initialize in Mustache descendants
6 months ago
.gitmodules
Updating ext/spec submodule pointer.
7 years ago

Mustache HTML Transform



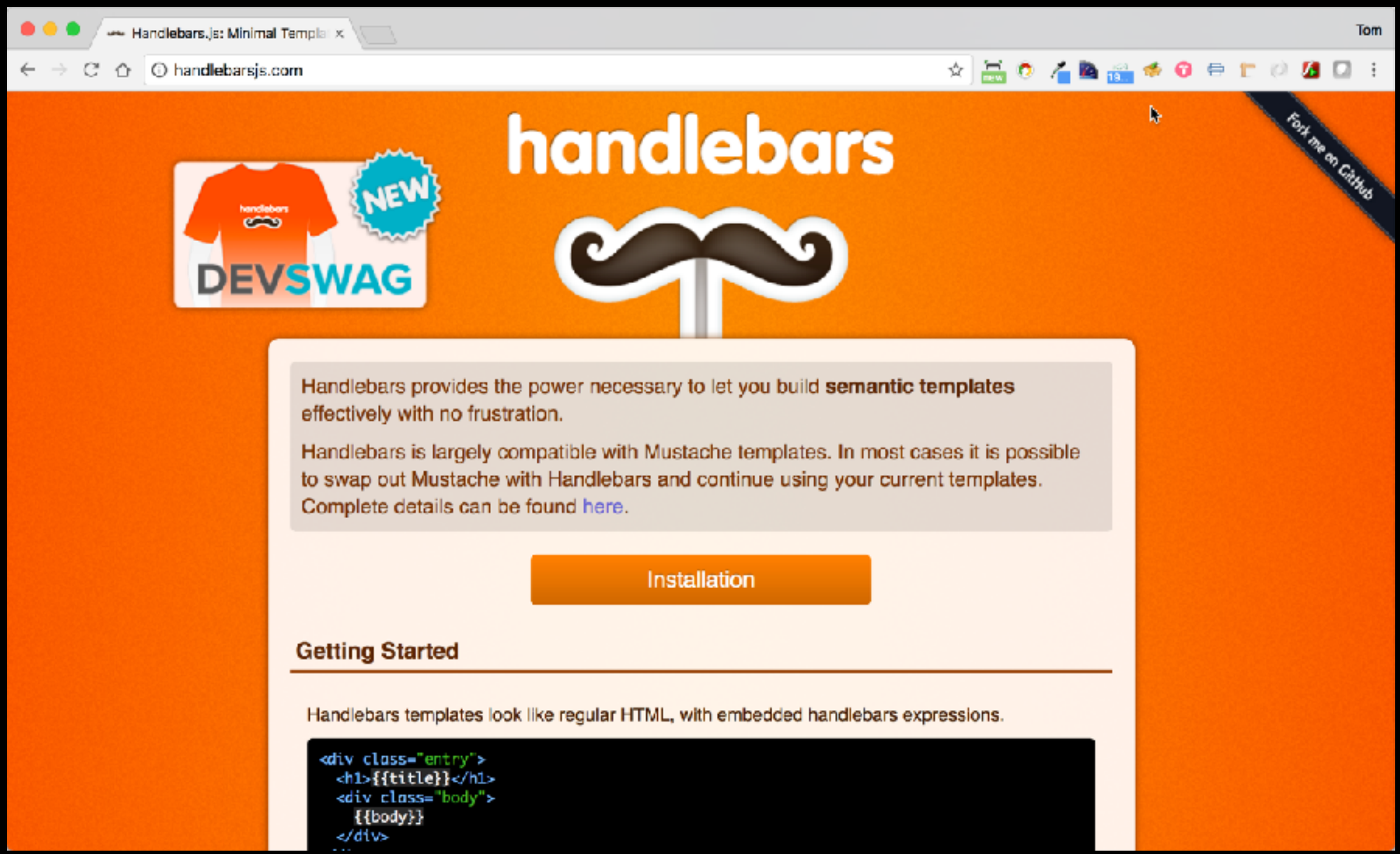
Mustache HTML Template

```
<!DOCTYPE html>
<html>

  <head>
    <meta charset="UTF-8" />
    <title>OpenWeather - California Cities</title>
    <link rel="stylesheet" href="weather.css">
  </head>
  <body>
    <h1>OpenWeather - California Cities</h1>
    <table class="weatherTable">
      <thead>
        <tr>
          <th>ID</th>
          <th>City</th>
          <th>Current Temp</th>
          <th>Low Temp</th>
          <th>High Temp</th>
          <th>Humidity</th>
          <th>Wind Speed</th>
          <th>Summary</th>
          <th>Description</th>
        </tr>
      </thead>
      {{#cities}}
        <tr>
          <td>{{id}}</td>
          <td>{{name}}</td>
          {{#main}}
            <td>{{temp}}</td>
            <td>{{temp_min}}</td>
            <td>{{temp_max}}</td>
            <td>{{humidity}}</td>
          {{/main}}
          <td>{{wind.speed}}</td>
          {{#weather.0}}
            <td>{{main}}</td>
            <td>{{description}}</td>
          {{/weather.0}}
```

Mustache HTML Test

```
12  var expect = require('chai').expect;
13  var jsonfile = require('jsonfile');
14  var fs = require('fs');
15  var mustache = require('mustache');
16
17  describe('cities-mustache', function() {
18    var jsonCitiesFileName = null;
19    var htmlTemplateFileName = null;
20
21
22    beforeEach(function() {
23      var baseDir = __dirname + '/../..';
24
25      jsonCitiesFileName = baseDir + '/data/cities-weather-short.json';
26      htmlTemplateFileName = baseDir +
27        '/templates/transform-html.mustache';
28    });
29
30
31    it('should transform cities JSON data to HTML', function(done) {
32      jsonfile.readFile(jsonCitiesFileName, function(readJsonFileError,
33        jsonObj) {
34        if (!readJsonFileError) {
35          fs.readFile(htmlTemplateFileName, 'utf8', function(
36            readTemplateFileError, templateFileData) {
37            if (!readTemplateFileError) {
38              var template = templateFileData.toString();
39              var html = mustache.render(template, jsonObj);
40
41              console.log('\n\n\nHTML Output:\n' + html);
42              done();
43            } else {
44              done(readTemplateFileError);
45            }
46          });
47        } else {
48          done(readJsonFileError);
49        }
50      });
```

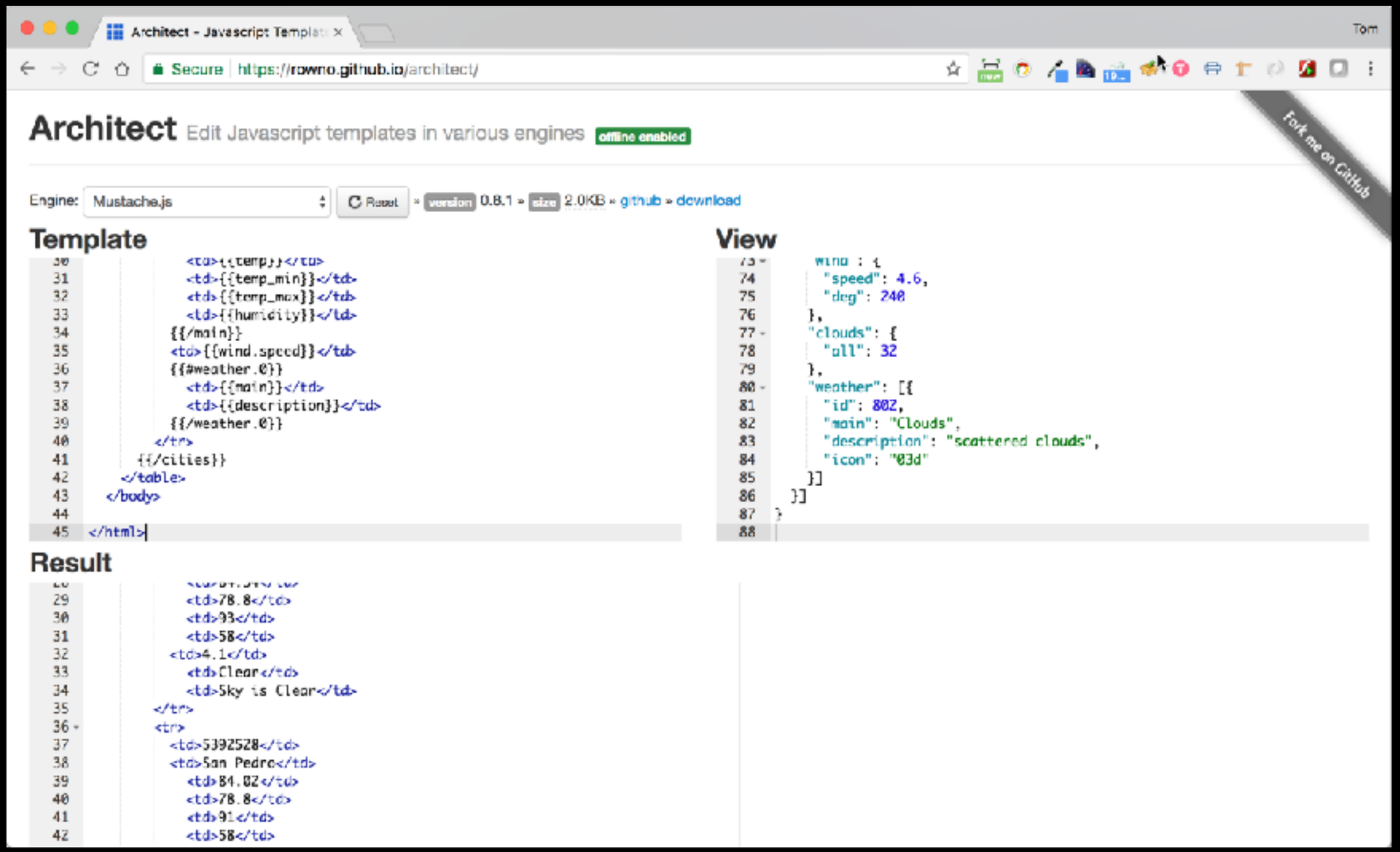
Handlebars


Handlebars.js: Minimal Templa
Tom
handlebarsjs.com
Fork me on GitHub
handlebars
NEW
DEVSWAG
Handlebars provides the power necessary to let you build semantic templates effectively with no frustration.
Handlebars is largely compatible with Mustache templates. In most cases it is possible to swap out Mustache with Handlebars and continue using your current templates. Complete details can be found here.
Installation
Getting Started
Handlebars templates look like regular HTML, with embedded handlebars expressions.

<h1>{{title}}</h1>

{{body}}

Handlebars HTML Transform



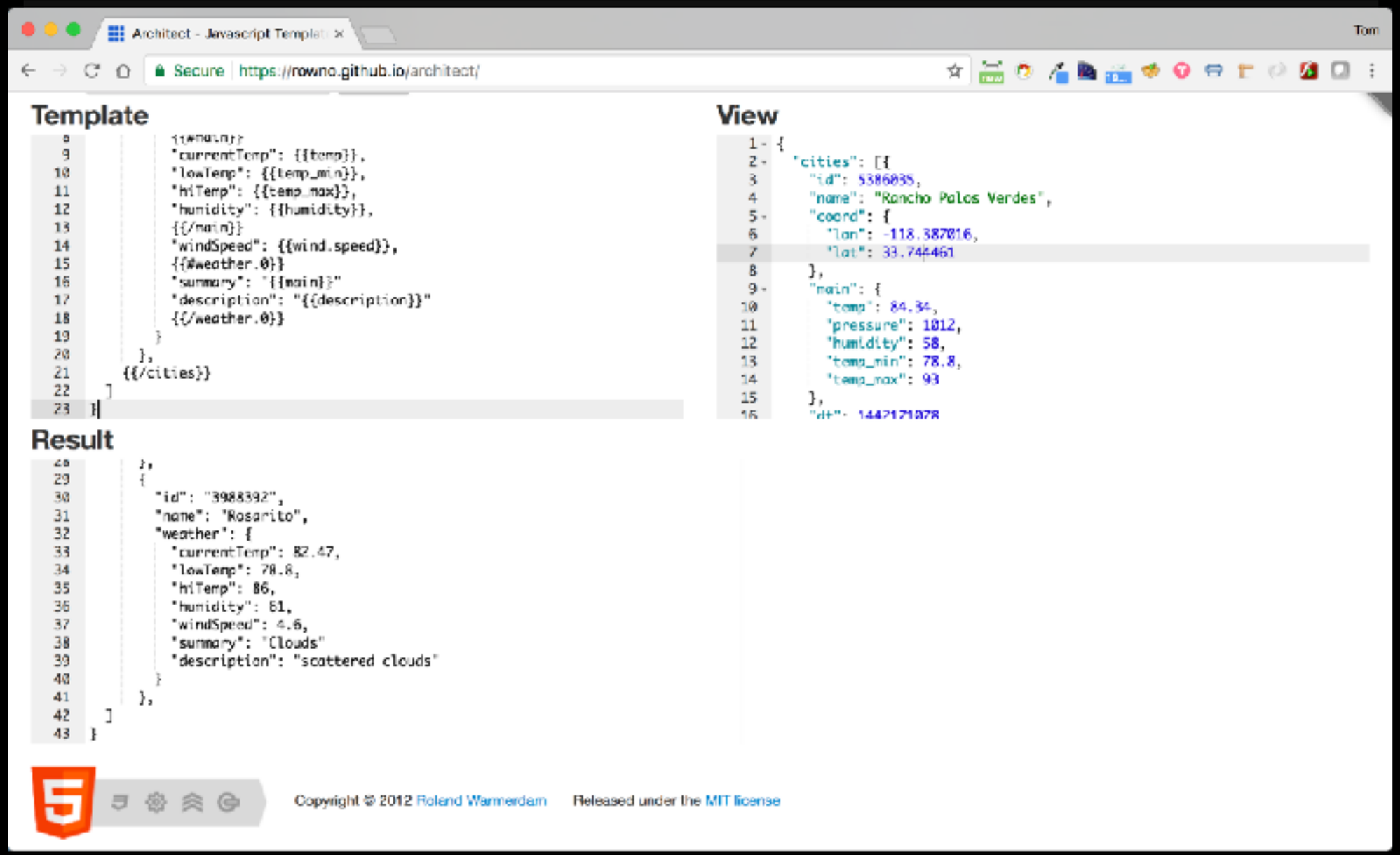
Handlebars HTML Template

```
<!DOCTYPE html>
<html>

  <head>
    <meta charset="UTF-8" />
    <title>OpenWeather - California Cities</title>
    <link rel="stylesheet" href="weather.css">
  </head>
<body>
  <h1>OpenWeather - California Cities</h1>
  <table class="weatherTable">
    <thead>
      <tr>
        <th>ID</th>
        <th>City</th>
        <th>Current Temp</th>
        <th>Low Temp</th>
        <th>High Temp</th>
        <th>Humidity</th>
        <th>Wind Speed</th>
        <th>Summary</th>
        <th>Description</th>
      </tr>
    </thead>
    {{#each cities}}
```

Handlebars HTML Test

```
12  var expect = require('chai').expect;
13  var jsonfile = require('jsonfile');
14  var fs = require('fs');
15  var handlebars = require('handlebars');
16
17  describe('cities-handlebars', function() {
18    var jsonCitiesFileName = null;
19    var htmlTemplateFileName = null;
20
21
22    beforeEach(function() {
23      var baseDir = __dirname + '/../..';
24
25    jsonCitiesFileName = baseDir + '/data/cities-weather-short.json';
26    htmlTemplateFileName = baseDir +
27      '/templates/transform-html.hbs';
28  });
29
30
31  it('should transform cities JSON data to HTML', function(done) {
32    jsonfile.readFile(jsonCitiesFileName, function(readJsonFileError,
33      jsonObj) {
34      if (!readJsonFileError) {
35        fs.readFile(htmlTemplateFileName, 'utf8', function(
36          readTemplateFileError, templateFileData) {
37          if (!readTemplateFileError) {
38            var template = handlebars.compile(tenplateFileData);
39            var html = template(jsonObj);
40
41            console.log('\n\n\nHTML Output:\n' + html);
42            done();
43          } else {
44            done(readTemplateFileError);
45          }
46        });
47      } else {
48        done(readJsonFileError);
49      }
50    });
```

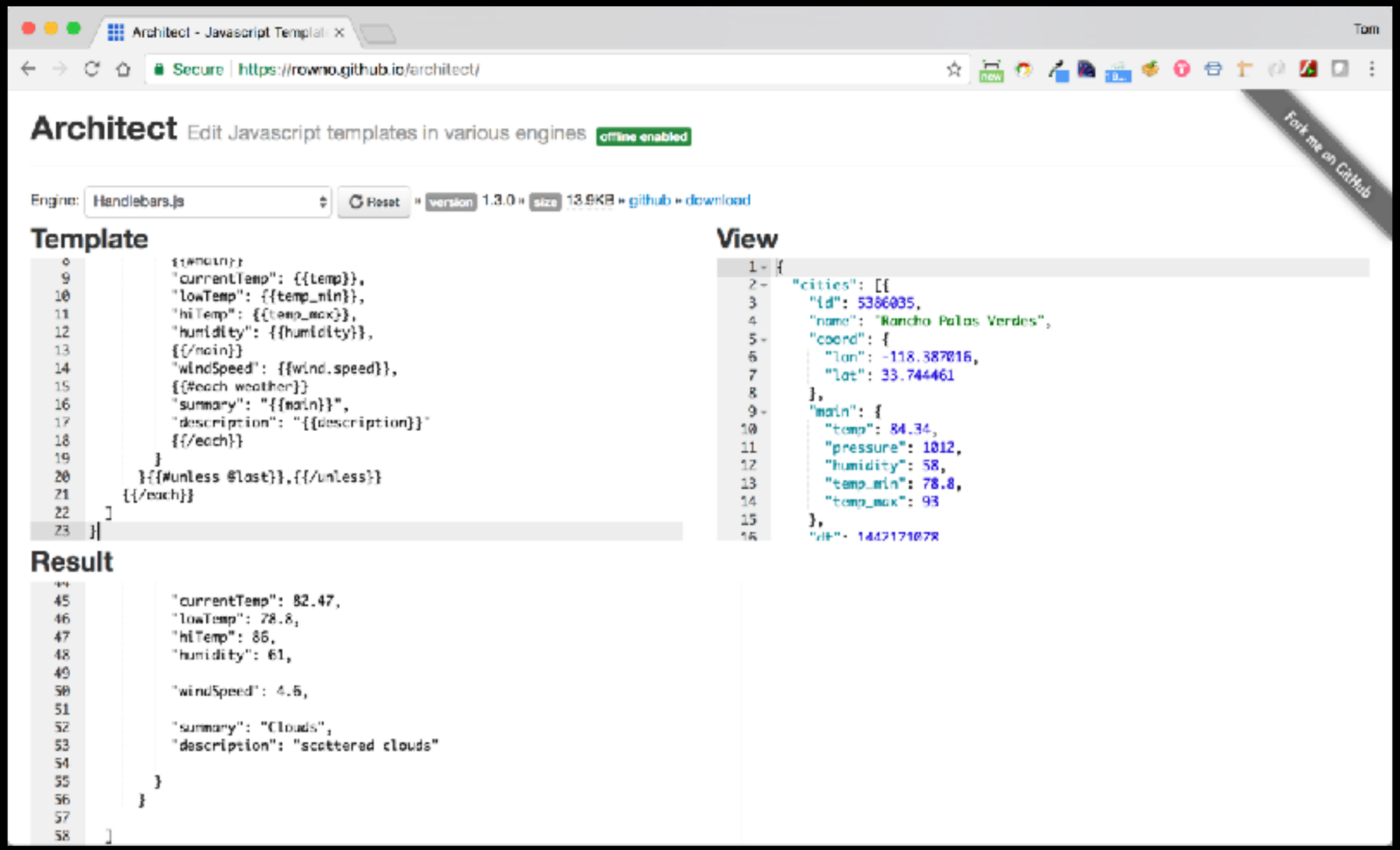
Mustache JSON Transform



Mustache JSON Template

```
{
  "cities": [
    {{#cities}}
      {
        "id": "{{id}}",
        "name": "{{name}}",
        "weather": {
          {{#main}}
          "currentTemp": {{temp}},
          "lowTemp": {{temp_min}},
          "hiTemp": {{temp_max}},
          "humidity": {{humidity}},
          {{/main}}
          "windSpeed": {{wind.speed}},
          {{#weather.0}}
          "summary": "{{main}}"
          "description": "{{description}}"
          {{/weather.0}}
        }
      },
    {{/cities}}
  ]
}
```

Handlebars JSON Transform



Handlebars JSON Template

```
{
  "cities": [
    {{#each cities}}
      {
        "id": "{{id}}",
        "name": "{{name}}",
        "weather": {
          {{#main}}
          "currentTemp": {{temp}},
          "lowTemp": {{temp_min}},
          "hiTemp": {{temp_max}},
          "humidity": {{humidity}},
          {{/main}}
          "windSpeed": {{wind.speed}},
          {{#each weather}}
          "summary": "{{main}}",
          "description": "{{description}}"
          {{/each}}
        }
      }{{#unless @last}},{{/unless}}
    {{/each}}
  ]
}
```

Agenda

JSON Search & Transform Overview 1

JSON Search 2

JSON Transform 3

What's The Point?

JSON Search and Transform

Simplify interaction with RESTful APIs

Your Takeaway

jq + Handlebars Rock!

Questions?

Tom Marrs

thomasamarrs@comcast.net

JSON Resources

JSON Generator - http://www.json-generator.com/

json.org - http://www.json.org

JSON Editor Online - http://jsoneditoronline.org/

JSONPath Resources

http://goessner.net/articles/JsonPath/

https://github.com/jayway/JsonPath

https://rubygems.org/gems/jsonpath/versions/0.5.6

https://www.npmjs.com/package/jsonpath

jq Resources

http://stedolan.github.io/jq/

http://stedolan.github.io/jq/tutorial/

https://github.com/stedolan/jq

https://robots.thoughtbot.com/jq-is-sed-for-json

https://zerokspot.com/weblog/2013/07/18/processing-json-with-jq/

https://jqplay.org/

JSON Groups

Google - http://groups.google.com/group/json-schema

Yahoo! - http://tech.groups.yahoo.com/group/json/