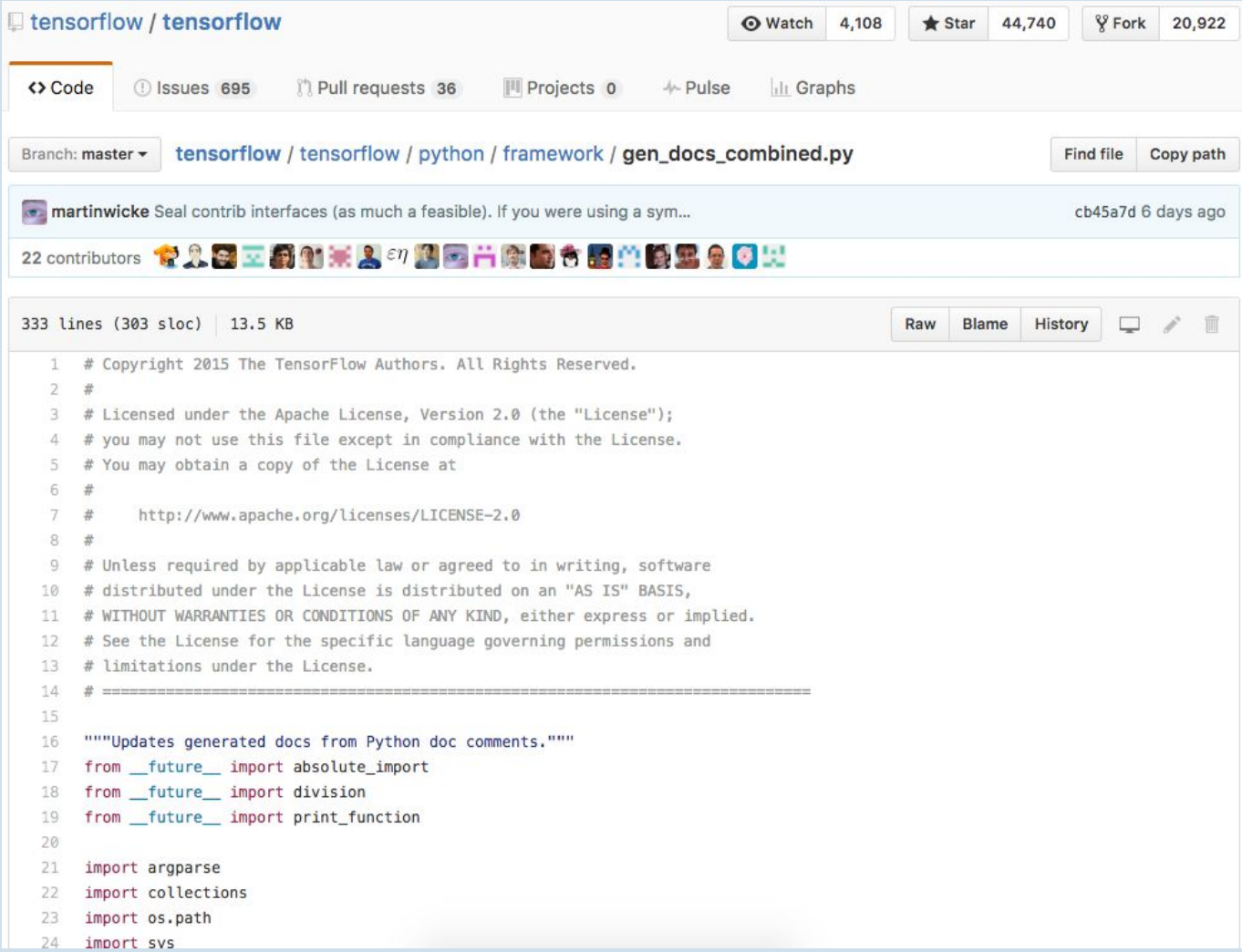
Analyzing 750 billion events and 46 TB of code

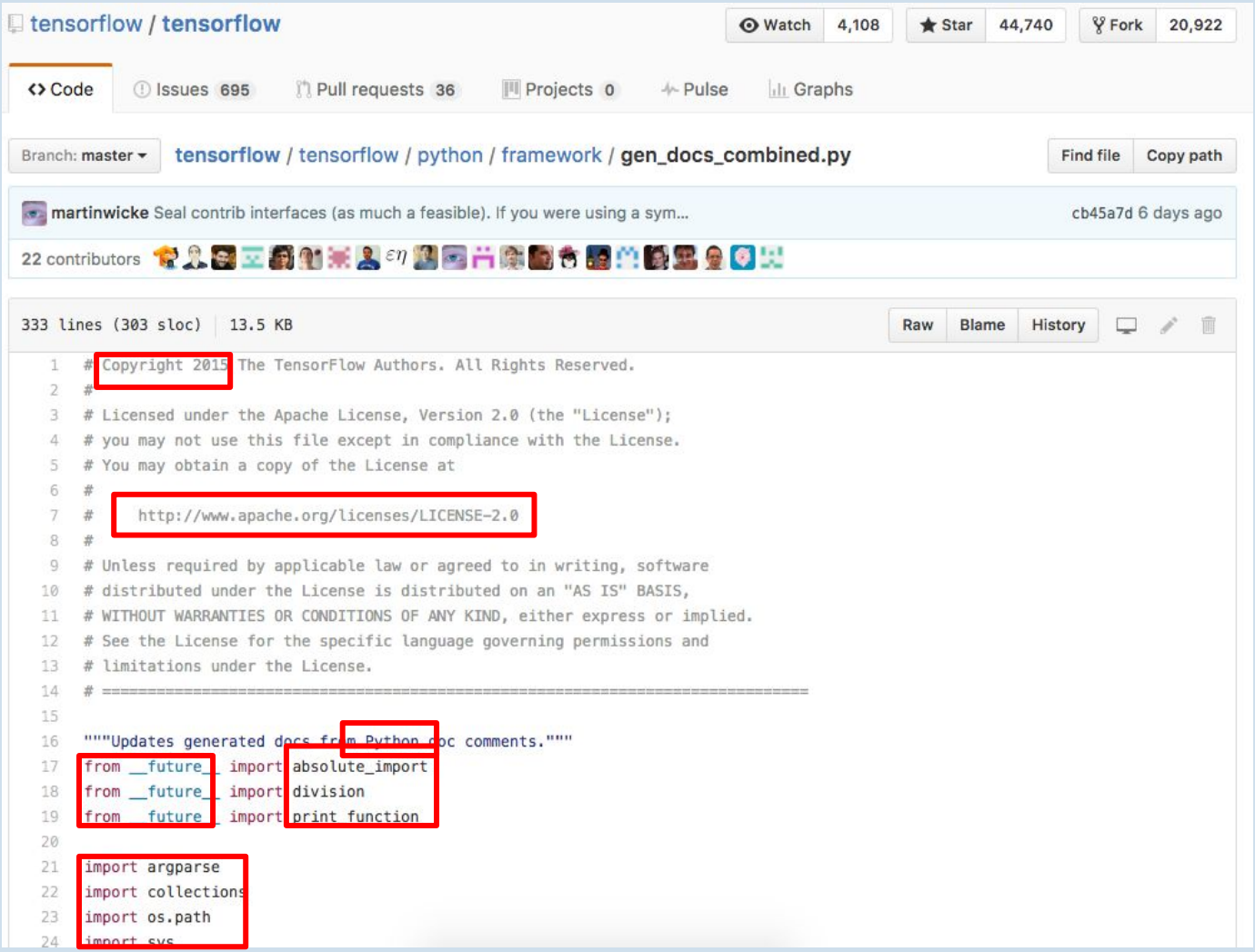
What you can learn from GitHub's shared data on BigQuery

Felipe Hoffa
Developer Advocate

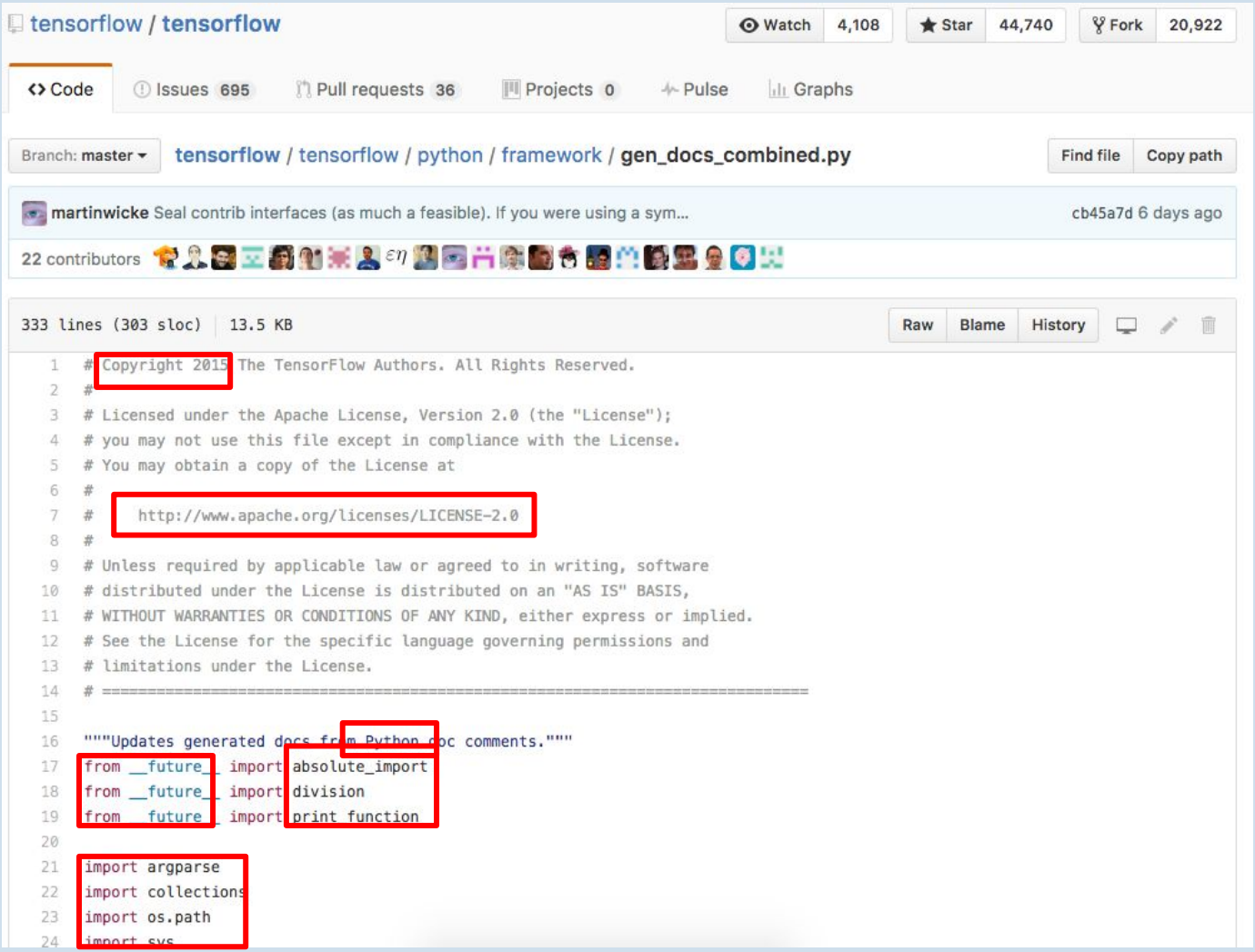
@felipehoffa

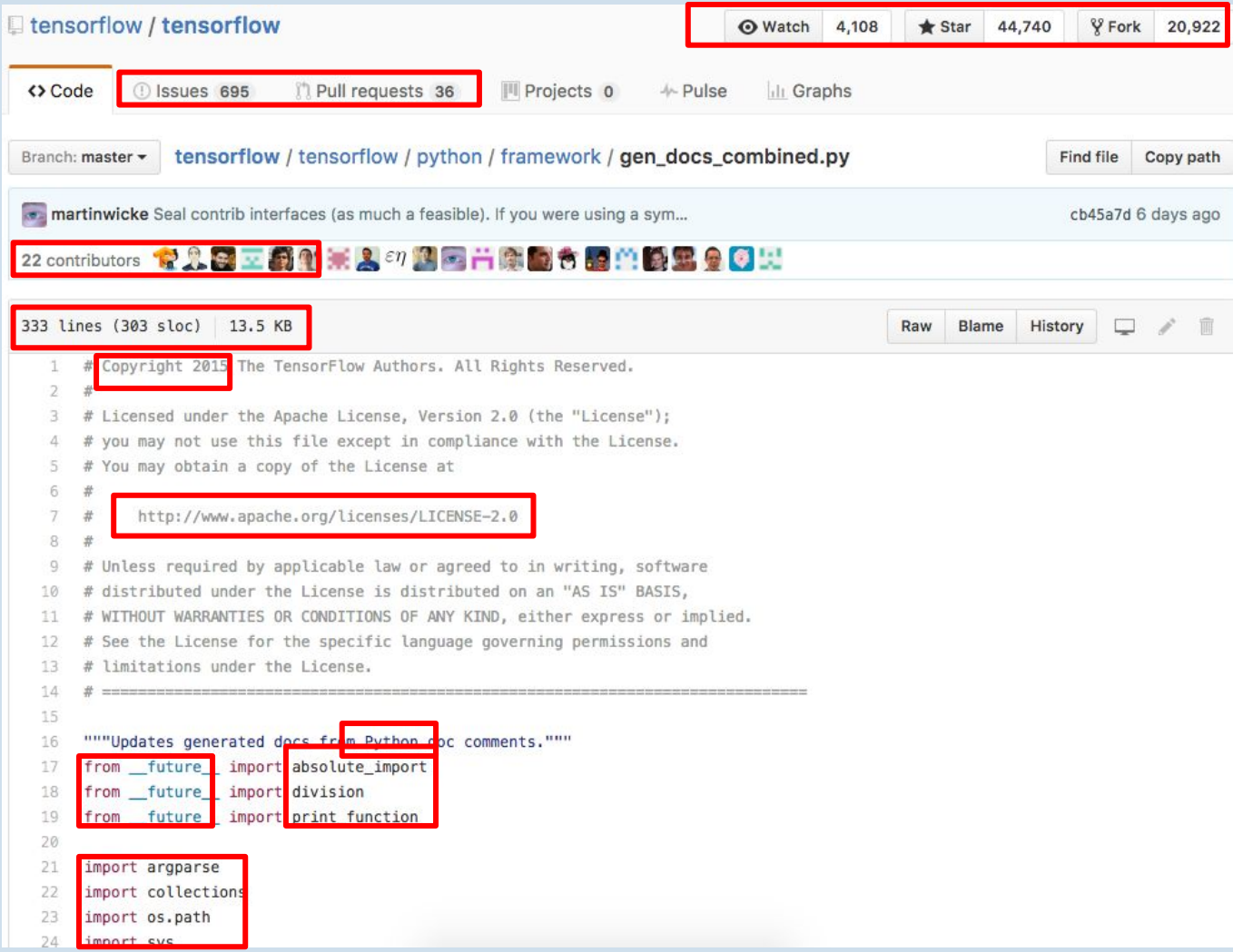
Google Cloud Platform

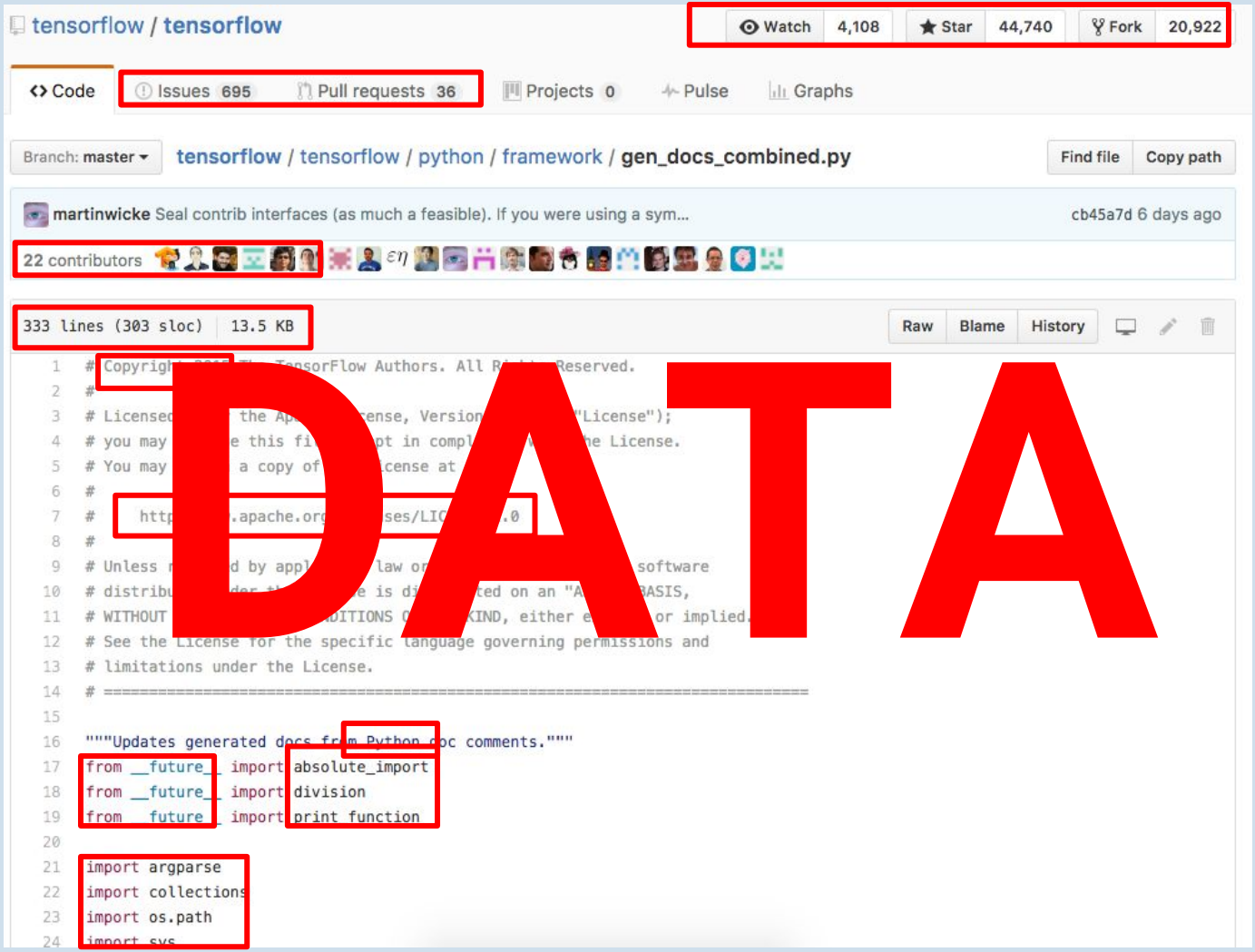

tensorflow / tensorflow
Watch 4,108
Star 44,740
Fork 20,922
Code
Issues 695
Pull requests 36
Projects 0
Pulse
Graphs
Branch: master
tensorflow / tensorflow / python / framework / gen_docs_combined.py
Find file
Copy path
martinwicke Seal contrib interfaces (as much a feasible). If you were using a sym...
cb45a7d 6 days ago
22 contributors
333 lines (303 sloc)
13.5 KB
Raw
Blame
History
Copyright 2015 The TensorFlow Authors. All Rights Reserved.
#
Licensed under the Apache License, Version 2.0 (the "License");
you may not use this file except in compliance with the License.
You may obtain a copy of the License at
#
http://www.apache.org/licenses/LICENSE-2.0
#
Unless required by applicable law or agreed to in writing, software
distributed under the License is distributed on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
See the License for the specific language governing permissions and
limitations under the License.
==
"""Updates generated docs from Python doc comments."""
from __future__ import absolute_import
from __future__ import division
from __future__ import print_function
import argparse
import collections
import os.path
import sys


tensorflow / tensorflow
Watch 4,108
Star 44,740
Fork 20,922
Code
Issues 695
Pull requests 36
Projects 0
Pulse
Graphs
Branch: master
tensorflow / tensorflow / python / framework / gen_docs_combined.py
Find file
Copy path
martinwicke Seal contrib interfaces (as much a feasible). If you were using a sym...
cb45a7d 6 days ago
22 contributors
333 lines (303 sloc)
13.5 KB
Raw
Blame
History
Copyright 2015 The TensorFlow Authors. All Rights Reserved.
#
Licensed under the Apache License, Version 2.0 (the "License");
you may not use this file except in compliance with the License.
You may obtain a copy of the License at
#
#
http://www.apache.org/licenses/LICENSE-2.0
#
Unless required by applicable law or agreed to in writing, software
distributed under the License is distributed on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
See the License for the specific language governing permissions and
limitations under the License.
==
"""Updates generated docs from Python doc comments."""
from __future__ import absolute_import
from __future__ import division
from __future__ import print_function
import argparse
import collections
import os.path
import sys

```
1  # Copyright 2015 The TensorFlow Authors. All Rights Reserved.
2  #
3  # Licensed under the Apache License, Version 2.0 (the "License");
4  # you may not use this file except in compliance with the License.
5  # You may obtain a copy of the License at
6  #
7  #     http://www.apache.org/licenses/LICENSE-2.0
8  #
9  # Unless required by applicable law or agreed to in writing, software
10 # distributed under the License is distributed on an "AS IS" BASIS,
11 # WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
12 # See the License for the specific language governing permissions and
13 # limitations under the License.
14 # ==============================================================================
15
16 """Updates generated docs from Python doc comments."""
17 from __future__ import absolute_import
18 from __future__ import division
19 from __future__ import print_function
20
21 import argparse
22 import collections
23 import os.path
24 import sys
```


tensorflow / tensorflow
Watch 4,108
Star 44,740
Fork 20,922
Code
Issues 695
Pull requests 36
Projects 0
Pulse
Graphs
Branch: master
tensorflow / tensorflow / python / framework / gen_docs_combined.py
Find file
Copy path
martinwicke Seal contrib interfaces (as much a feasible). If you were using a sym...
cb45a7d 6 days ago
22 contributors
333 lines (303 sloc)
13.5 KB
Raw
Blame
History
Copyright 2015 The TensorFlow Authors. All Rights Reserved.
#
Licensed under the Apache License, Version 2.0 (the "License");
you may not use this file except in compliance with the License.
You may obtain a copy of the License at
#
#
http://www.apache.org/licenses/LICENSE-2.0
#
Unless required by applicable law or agreed to in writing, software
distributed under the License is distributed on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
See the License for the specific language governing permissions and
limitations under the License.
==
"""Updates generated docs from Python doc comments."""
from __future__ import absolute_import
from __future__ import division
from __future__ import print_function
import argparse
import collections
import os.path
import sys


tensorflow / tensorflow
Watch 4,108
Star 44,740
Fork 20,922
Code
Issues 695
Pull requests 36
Projects 0
Pulse
Graphs
Branch: master
tensorflow / tensorflow / python / framework / gen_docs_combined.py
Find file
Copy path
martinwicke Seal contrib interfaces (as much a feasible). If you were using a sym...
cb45a7d 6 days ago
22 contributors
333 lines (303 sloc)
13.5 KB
Raw
Blame
History
Copyright 2015 The TensorFlow Authors. All Rights Reserved.
#
Licensed under the Apache License, Version 2.0 (the "License");
you may not use this file except in compliance with the License.
You may obtain a copy of the License at
#
#
http://www.apache.org/licenses/LICENSE-2.0
#
Unless required by applicable law or agreed to in writing, software
distributed under the License is distributed on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
See the License for the specific language governing permissions and
limitations under the License.
==
"""Updates generated docs from Python doc comments."""
from __future__ import absolute_import
from __future__ import division
from __future__ import print_function
import argparse
import collections
import os.path
import sys


tensorflow / tensorflow
Watch 4,108
Star 44,740
Fork 20,922
Code
Issues 695
Pull requests 36
Projects 0
Pulse
Graphs
Branch: master
tensorflow / tensorflow / python / framework / gen_docs_combined.py
Find file
Copy path
martinwicke Seal contrib interfaces (as much a feasible). If you were using a sym...
cb45a7d 6 days ago
22 contributors
333 lines (303 sloc)
13.5 KB
Raw
Blame
History
DATA
"""Updates generated docs from Python doc comments."""
from __future__ import absolute_import
from __future__ import division
from __future__ import print_function
import argparse
import collections
import os.path
import sys

Who wants to analyze GitHub?

Project maintainers

- Popularity
- Who and how?
- Change management:
  - New APIs?
  - Breaking changes?
- Is my project healthy?
  - Issues closed on time?
  - Community participation?

Project users

- What other projects to follow?
- Requesting features
  - Data based requests
- Effective phrasing

Project choosers

- Is this project popular?
- Is this project healthy?
- Is this project well adopted?
- Related projects?

Data lovers

- Data integrators
- You
- Me :)

3 main datasets:

- GitHub Archive
  - 8.7 billion events
  - Hourly updates
- GHTorrent
  - These events annotated
  - Real-time updates
- GitHub repos on BigQuery
  - 46 TB of code

Google Cloud Platform

Google BigQuery

Google BigQuery

- Fast: terabytes in seconds
- Simple: SQL
- Scaleable: From bytes to petabytes
- No CAPEX: Always on
- Interoperable: Tableau, R, Python...
- **Instant sharing**
- **Free monthly quota**

Top projects by stars 2016?

```sql
#standardSQL
SELECT repo.name, COUNT(*) c
FROM `githubarchive.month.2016*`
WHERE type='WatchEvent'
GROUP BY 1
ORDER BY c DESC
LIMIT 20
```

RUN QUERY | Save Query

Row	name	c
1	FreeCodeCamp/FreeCodeCamp	185399
2	jwasham/google-interview-university	31496
3	vuejs/vue	29184
4	vhf/free-programming-books	29060
5	tensorflow/tensorflow	28634
6	facebook/react	26422
7	getify/You-Dont-Know-JS	25349
8	sindresorhus/awesome	25236
9	chrislgarry/Apollo-11	23633
10	yarnpkg/yarn	21487

Really?

```
#standardSQL
SELECT repo.name, COUNT(*) c, COUNT(DISTINCT actor.id) real_c
FROM `githubarchive.month.2016*`
WHERE type='WatchEvent'
GROUP BY 1
ORDER BY real_c DESC
LIMIT 20
```

Ctrl +

RUN QUERY | Save Query | Save View | Format Query | Show Options | Query complete (14.4s elapsed, 13.9 GB proces

Results | Explanation | Job Information | Download as CSV | Download as JSON

Row	name	c	real_c
1	FreeCodeCamp/FreeCodeCamp	185399	174984
2	jwasham/google-interview-university	31496	30973
3	vhf/free-programming-books	29060	27937
4	vuejs/vue	29184	27533
5	tensorflow/tensorflow	28634	27373
6	facebook/react	26422	24976
7	getify/You-Dont-Know-JS	25349	24466
8	sindresorhus/awesome	25236	24345
9	chrislgarry/Apollo-11	23633	23282
10	yarnpkg/yarn	21487	21234

I got stars!
What else did they star?

```sql
SELECT repo, COUNT(*) c FROM (
    SELECT actor.login, ARRAY_AGG(DISTINCT repo.name) repos
    FROM `githubarchive.month.2016*`
    WHERE type="WatchEvent" GROUP BY 1
    HAVING ARRAY_LENGTH(repos) BETWEEN 3 AND 30000  # if you star
    AND 'tensorflow/tensorflow' IN UNNEST(repos)
    AND 'FreeCodeCamp/FreeCodeCamp' NOT IN UNNEST(repos)
), UNNEST(repos) repo
GROUP BY 1 ORDER BY 2 DESC
LIMIT 30
```

Ctrl + Enter: r

RUN QUERY | Save Query | Save View | Format Query | Show Options | Query complete (17.1s elapsed, 14.9 GB processed)

Results | Explanation | Job Information | Download as CSV | Download as JSON | Sav

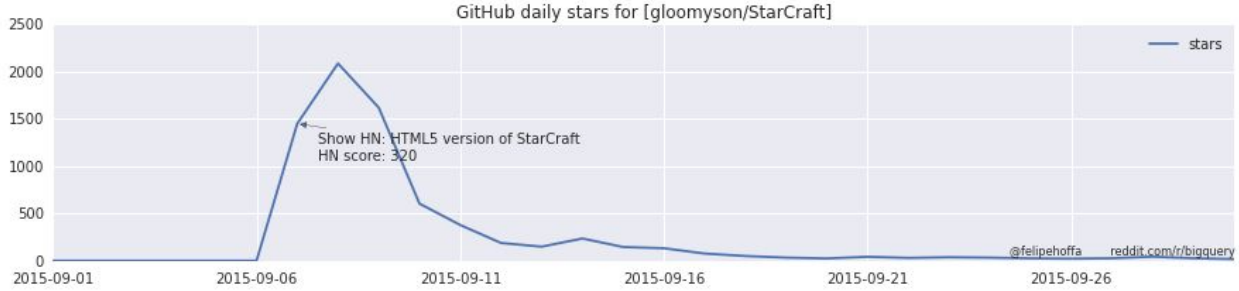
Row	repo	c
1	tensorflow/tensorflow	22613
2	tensorflow/models	2603
3	BVLC/caffe	2397
4	fchollet/keras	2222
5	jwasham/google-interview-university	2198
6	scikit-learn/scikit-learn	2051
7	Microsoft/CNTK	2007

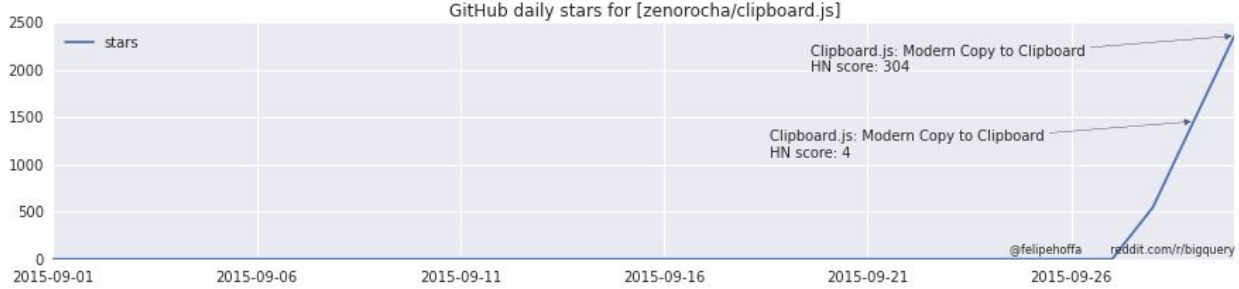
How did they find me?

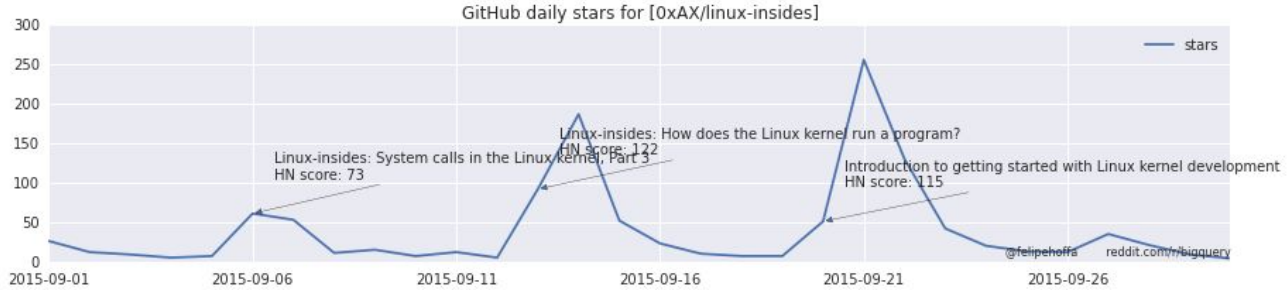
Hacker News?

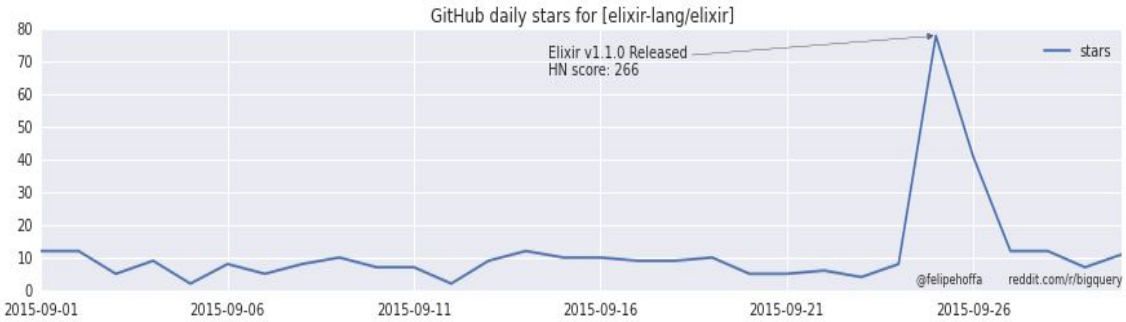
The Hacker News frontpage effect on a project's GitHub stars

To produce this dataset I used the GitHub Archive dataset and the Hacker News dataset - both loaded into Google BigQuery
The question is: How many times does a project get starred after being featured on the frontpage of Hacker News? --@felipehoffa









With BigQuery:

```
SELECT repo_name, created_at date, COUNT(*) c, GROUP_CONCAT_UNQUOTED(UNIQUE(hndate+':'+STRING(hnscore))) hndates, SUM(UNIQUE(hns
core)) hnscore, SUM(c) OVER(PARTITION BY repo_name) monthstars
FROM (
  SELECT repo_name,  actor_login, DATE(MAX(created_at)) created_at, date hndate, score hnscore
  FROM [githubarchive:month.201509] a
  JOIN (
    SELECT REGEXP_EXTRACT(url, r'github.com/([a-zA-Z0-9\-\.]+.[a-zA-Z0-9\-\.]*)') mention, DATE(time_ts) date, score
    FROM [fh-bigquery:hackernews.stories]
    WHERE REGEXP_MATCH(url, r'github.com/[a-zA-Z0-9\-\.]+')
    AND score>10
    AND YEAR(time_ts)=2015 AND MONTH(time_ts)=9
    HAVING NOT (mention CONTAINS '.com/search?' OR mention CONTAINS '.com/blog/')
  ) b
  ON a.repo_name=b.mention
  WHERE type="WatchEvent"
  GROUP BY 1,2, hndate, hnscore
)
GROUP BY 1,2
HAVING hnscore>300
ORDER BY 1,2,4
LIMIT 1000
```

Project health

- Projects with most issues
- Projects with most people filing issues
- Projects with most engagement
- Best projects at closing issues
- Best phrasing for issue closing

```
SELECT repo.name, COUNT(*) c
FROM [githubarchive:month.201606]
WHERE type IN ( 'IssueCommentEvent')
GROUP BY repo.name
ORDER BY c DESC
LIMIT 10
```

Row	repo_name	c
1	kubernetes/kubernetes	17883
2	apache/spark	8620
3	openshift/origin	5753
4	sauron-demo/sauron-demo	4922
5	docker/docker	4661
6	tgstation/tgstation	4443
7	cms-sw/cmssw	4352
8	d3athrow/vgstation13	4134
9	servo/servo	3668
10	rust-lang/rust	3450

GitHub repos with the most comments on issues June 2016

```
SELECT repo.name,
       ROUND(COUNT(*)/EXACT_COUNT_DISTINCT(actor.login),2)
comments_per_author,
       EXACT_COUNT_DISTINCT(actor.login ) authors,
       COUNT(*) comments
FROM [githubarchive:month.201606]
WHERE type IN ('IssueCommentEvent')
AND actor.login NOT IN ( SELECT actor.login FROM (
  SELECT actor.login, COUNT(*) c
  FROM [githubarchive:month.201606]
  WHERE type IN ('IssueCommentEvent')
  GROUP BY 1
  HAVING c>1000
  ORDER BY 2 DESC
))
```

Row	repo_name	comments_per_author	authors	comments
1	kubernetes/kubernetes	18.03	499	8997
2	tensorflow/tensorflow	6.11	405	2475
3	docker/docker	5.94	785	4660
4	Microsoft/vscode	5.71	601	3433
5	facebook/react-native	5.15	608	3133
6	angular/angular	4.89	532	2604
7	magento/magento2	4.17	468	1952
8	npm/npm	2.24	439	984
9	FortAwesome/Font-Awesome	1.6	643	1026

GitHub projects with more comments per author within top commented projects (removing bots)

Even text analysis?

```
SELECT REGEXP_EXTRACT(JS
r'^([A-Za-z]+ [A-Za-z]+
       COUNT(DISTINCT ac
       ROUND(100*SUM(JSO
'$.action')='closed')/SU
'$.action')='opened'),1)
         percentage_clos
FROM [githubarchive:mont
WHERE type='IssuesEvent'
GROUP BY 1
HAVING start IS NOT null
ORDER BY 2 DESC
LIMIT 20
```

Row	start	authors	percentage_closed
1	It would be nice	1292	56.5
2	Is it possible to	1122	73.8
3	I am trying to	1060	79.8
4	I would like to	823	66.4
5	Is there a way	749	74.2
6	It would be great	663	57.6
7	So that the humans	632	59.6
8	When I try to	579	77.4
9	Would it be possible	377	67.5
10	Is there any way	282	65.3
11	Not sure if this	282	81.8
12	I get the following	242	86.6
13	I was trying to	229	69.5
14	I want to use	228	64.2
15	I have the following	225	78.7

Most common ways to open an issue on GitHub, and % that get closed

So where's the code?

Felipe Hoffa Follow

Developer Advocate @Google. Originally from Chile, now in San Francisco and around the world.

Jun 29, 2016 · 4 min read

All the open source code in GitHub now shared within BigQuery: Analyze all the code!

Google BigQuery

COMPOSE QUERY

Query History

Job History

- contents_java_2014
- contents_java_2015
- contents_java_2016
- contents_js
- contents_php
- contents_py
- contents_rb
- file_ages
- hackernews
- io16
- jacobwgillespie

New Query

Query Editor | UDF Editor

```
1 SELECT line, COUNT(*) c
2 FROM (
3   SELECT SPLIT(content, '\n') line
4   FROM [fh-bigquery:github_extracts.contents_java_2016]
5   HAVING REGEXP_MATCH(line, '^import')
6 )
7 GROUP BY 1
8 ORDER BY 2 DESC
9 LIMIT 300
```

RUN QUERY | Save Query | Save View | Format Query | Show Options | Query complete (18.8s elapsed, 12.2 GB processed)

Results | Explanation | Download as CSV | Download as JSON | Save as Table | Save to Google Sheets

Row	line	c
1	import java.util.List;	428404
2	import java.util.ArrayList;	249648
3	import java.util.Map;	221261

Google BigQuery

COMPOSE QUERY

Query History

Job History

- bigquery-public-data
  - baseball
  - bls
  - cloud_storage_geo_index
  - common_eu
  - common_us
  - fec
  - ghcn_d
  - ghcn_m
  - github_repos
    - commits
    - **contents**
    - files
    - languages
    - licenses
    - sample_commits
    - sample_contents
    - sample_files
    - sample_repos
  - hacker_news

Table Details: contents

Schema | Details | Preview

Description

Unique file contents of text files under 1 MiB on the HEAD branch.

Can be joined to [bigquery-public-data:github_repos.files] table using the id columns to identify the repository and file path.

Table Info

Table ID	bigquery-public-data:github_repos.contents
Table Size	1.79 TB
Number of Rows	213,849,274
Creation Time	Mar 11, 2016, 8:18:08 PM
Last Modified	Jan 26, 2017, 1:33:15 PM
Data Location	US
Labels	None Edit

Table Details: contents

Schema	Details	Preview

id	STRING	NULLABLE	Describe this field...
size	INTEGER	NULLABLE	Describe this field...
content	STRING	NULLABLE	Describe this field...
binary	BOOLEAN	NULLABLE	Describe this field...
copies	INTEGER	NULLABLE	Describe this field...

```
1  SELECT SUM(size*copies) total_bytes
2  FROM [bigquery-public-data:github_repos.contents]
```

RUN QUERY | Save Query | Save View | Format Query | Show Options | Query complete (3.8s elapsed, 3.19 GB processed)

Results | Explanation | Job Information

Row	total_bytes
1	46108359148175

Table | JSON

Rules to analyze [bigquery-public-data:github_repos.contents]

- Text files <1MB
- One copy of each unique file
- JOIN with [github_repos.files] for paths
- Don't JOIN with [github_repos.files] to get contents*path.
- Extract first, analyze later
- [github_repos.sample_contents]
    -> 10% of contents, top projects, 1 sample path.

- Only open source projects - https://developer.github.com/v3/licenses/
- Some projects missing - why?

Top java imports growth 2013-16

Row	line	imports_2013	imports_2016	ratio_2013	ratio_2016	win
1	import javax.inject.Inject;	5784	25060	0.17	0.42	41.54
2	import com.google.common.collect.ImmutableList;	4619	19245	0.14	0.32	39.91
3	import android.os.Build;	3625	14133	0.11	0.23	37.08
4	import android.text.TextUtils;	3646	12899	0.11	0.21	32.82
5	import javax.annotation.Nullable;	5083	16487	0.15	0.27	28.89
6	import java.util.UUID;	7110	22440	0.21	0.37	27.63
7	import android.view.LayoutInflater;	9090	28453	0.27	0.47	27.25
8	import org.springframework.beans.factory.annotation.Autowired;	12280	38003	0.36	0.63	26.72
9	import android.view.MenuItem;	4929	15187	0.15	0.25	26.52
10	import android.view.ViewGroup;	10722	32863	0.32	0.55	26.28
11	import static org.mockito.Mockito.verify;	3243	9910	0.1	0.16	26.14
12	import static org.mockito.Mockito.mock;	3736	11195	0.11	0.19	25.22
13	import static org.mockito.Mockito.when;	3960	11736	0.12	0.19	24.7
14	import android.widget.ImageView;	5625	16630	0.17	0.28	24.59

Biggest raise in imports from 2013 to 2016

Requesting a feature for Go

Proposal: Add time.Until() to the time package #14595

Closed **SamWhited** opened this issue on Mar 1, 2016 · 13 comments

SamWhited commented on Mar 1, 2016 — Contributor

I'd like to propose that a `time.Until(t time.Time) time.Duration` function be added to the time package to compliment the existing `Since()` shortcut. This would make writing expressions with an expiration time a bit more readable:

```
<-After(time.Until(expirationTime))
```

vs.

```
<-After(expirationTime.Sub(time.Now()))
```

While it's still fairly obvious what the second one does, it takes a little longer to recognize "sub" as subtraction than just seeing the symbol. Also keeping time expressions more or less readable as english is a nice benefit of having the until shortcut (as you can do with the existing since function).

If this accepted, I've got a CL here for review.

15 2

campoy commented on Jul 27, 2016 Member

There's in total around 2000 repos (counting forks only once) that could benefit from this feature.

```
SELECT REGEXP_EXTRACT(repo_name, r'.*/(.*)') as project, FLOOR(COUNT(*) / COUNT(DISTINCT re
FROM (
  SELECT id, split(content, '\n') as line
  FROM [campoy-github:go_files.contents]
  HAVING line CONTAINS '.Sub(time.Now())'
) as contents JOIN [campoy-github:go_files.files] as files
ON contents.id = files.id
GROUP BY project
ORDER BY n DESC
```

The 10 projects that would benefit the most are

```
Row project                    n           sample
1   robin                      62.0        time.Sleep(expectedExp.Sub(time.Now()) - 500*ti
2   contrib                    26.0        deadlineTimeout := dialer.Deadline.Sub(time.Now
3   j2                         26.0        time.Sleep(expectedExp.Sub(time.Now()) - 500*ti
4   gitarchive                 20.0        c.traceInfo.firstLine.deadline = deadline.Sub(t
5   megacfs                    20.0        sleep := nextRun.Sub(time.Now())
6   microcosm                  20.0        d := deadline.Sub(time.Now())
7   concourse-pipeline-resource 20.0       req.TimeoutSeconds = proto.Float64(cn.readDeadl
8   etcd2-bootstrapper         20.0        Timeout: d.Sub(time.Now()),
9   zypper-docker              18.0        timeout = deadline.Sub(time.Now())
10  doit                       16.0        Timeout: d.Sub(time.Now()),
```

In my opinion there's some projects that could benefit of this, but it's clearly not a high priority addition to the stdlib.

 52

Beyond regex

Static code analysis with UDFs

Felipe Hoffa Follow
Developer Advocate @Google. Originally from Chile, now in San Francisco and around the world.
Jun 29, 2016 · 2 min read

Static JavaScript code analysis inside a SQL query: JSHint+GitHub+BigQuery

Can we run a static code analysis tool for JavaScript inside BigQuery? Yes we can.

```
#set UDF Source URI option to "gs://fh-bigquery/js/jshint-2.5.11.js"

SELECT x error, COUNT(*) files_affected
FROM js(
(
  SELECT content, sample_path, sample_repo_name
  FROM [fh-bigquery:github_extracts.contents_js]
  WHERE LENGTH(content) BETWEEN 1000 AND 1800
  AND ABS(HASH(id))%1000=0  # sampling
),
content, sample_path, sample_repo_name,
"[
  {name: 'x', type:'string'},
  {name: 'sample_path', type:'string'},
  {name: 'sample_repo_name', type:'string'},
  {name: 'content', type:'string'}]",
  "function(r, emit) {
    JSHINT(r.content, {'maxdepth':2});
    // data = JSHINT.data();
    errors = JSHINT.errors;
    set_errors=new Set(errors.map(
      function(x) {
        if(x && 'raw' in x) {return x.raw}}));
    set_errors.forEach(function(x) {
      if(!x) {return;}
      emit({
        x: x,
        sample_repo_name: r.sample_repo_name,
        sample_path: r.sample_path,
      });
    });
  }")
GROUP BY 1
ORDER BY 2 DESC
LIMIT 100


7.4s elapsed, 103 GB processed
```

```
#set UDF Source URI option to "gs://fh-bigquery/js/jshint-2.5.11.js"

SELECT x error, COUNT(*) files_affected
FROM js(
(
  SELECT content, sample_path, sample_repo_name
  FROM [fh-bigquery:github_extracts.contents_js]
  WHERE LENGTH(content) BETWEEN 1000 AND 1800
  AND ABS(HASH(id))%1000=0  # sampling
),
content, sample_path, sam
"[
  {name: 'x', type:'strir
  {name: 'sample_path', t
  {name: 'sample_repo_nam
  {name: 'content', type:
  "function(r, emit) {
    JSHINT(r.content, {'m
    // data = JSHINT.data
    errors = JSHINT.error
    set_errors=new Set(er
      function(x) {
        if(x && 'raw' in
    set_errors.forEach(fu
      if(!x) {return;}
      emit({
        x: x,
        sample_repo_name:
        sample_path: r.sa
      });
    });
  }"
GROUP BY 1
ORDER BY 2 DESC
LIMIT 100


7.4s elapsed, 103 GB proc
```

Row	error	files_affected
1	Missing semicolon.	377
2	Use the function form of "use strict".	199
3	'{a}' is not defined.	192
4	'{a}' is only available in ES6 (use esnext option).	120
5	Expected an assignment or function call and instead saw an expression.	118
6	'{a}' is available in ES6 (use esnext option) or Mozilla JS extensions (use moz).	117
7	Use '{a}' to compare with '{b}'.	65
8	Expected '{a}' and instead saw '{b}'.	53
9	Blocks are nested too deeply. ({a})	42
10	Expected an identifier and instead saw '{a}'.	38
11	Bad line breaking before '{a}'.	33
12	'{a}' is already defined.	27
13	Unnecessary semicolon.	26
14	This character may get silently deleted by one or more browsers.	26

Felipe Hoffa Follow
Developer Advocate @Google. Originally from Chile, now in San Francisco and around the world.
Aug 30, 2016 · 3 min read

400,000 GitHub repositories, 1 billion files, 14 terabytes of code: Spaces or Tabs?

Tabs or spaces. We are going to parse a billion files among 14 programming languages to decide which one is on top.

Spaces vs Tabs - GitHub on BigQuery edition

The rules:

- **Data source**: GitHub files stored in BigQuery.
- **Stars matter**: We'll only consider the top 400,000 repositories—by number of stars they got on GitHub during the period Jan-May 2016.
- **No small files**: Files need to have at least 10 lines that start with a space or a tab.
- **No duplicates**: Duplicate files only have one vote, regardless of how many repos they live in.
- **One vote per file**: Some files use a mix of spaces or tabs. We'll count on which side depending on which method they use more.
- **Top languages**: We'll look into files with the extensions (.java, .h, .js, .c, .php, .html, .cs, .json, .py, .cpp, .xml, .rb, .cc, .go).

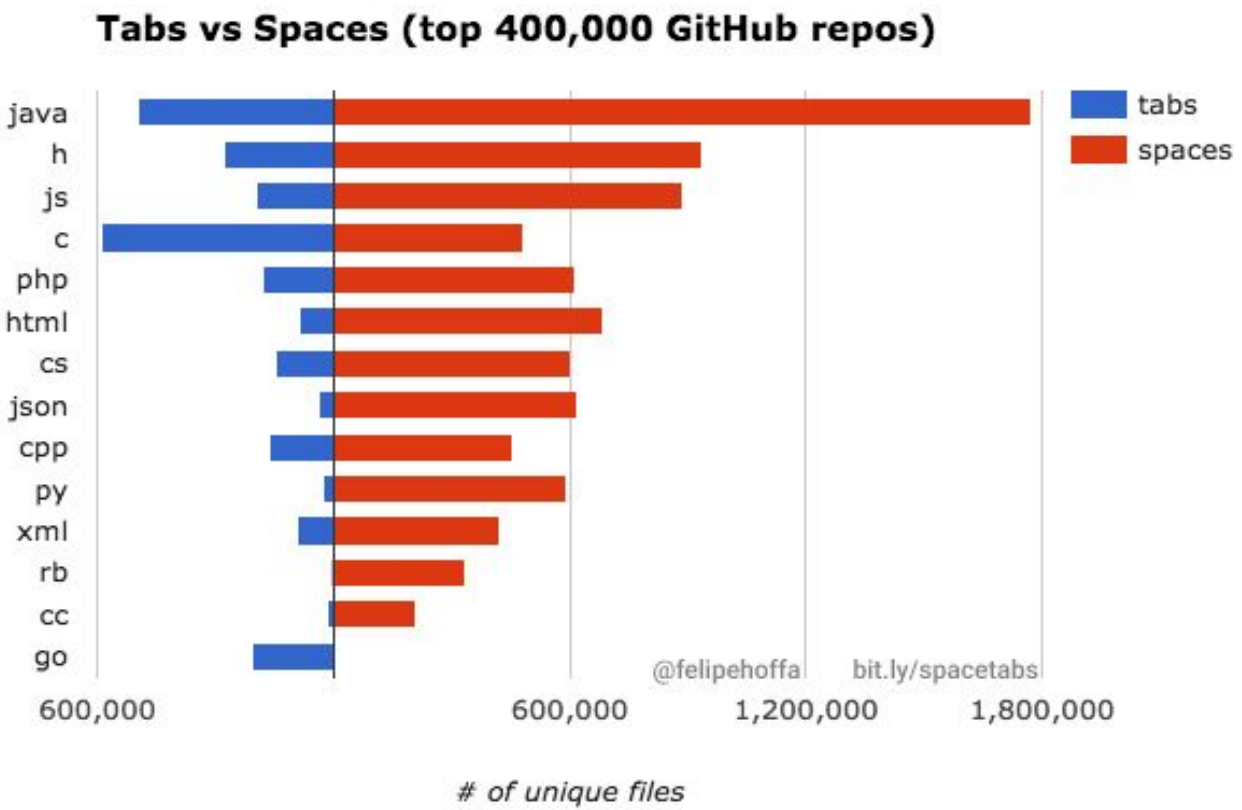
Spaces vs Tabs - Extract

```
SELECT a.id id, size, content, binary, copies, sample_repo_name ,
sample_path
FROM (
  SELECT id, FIRST(path) sample_path, FIRST(repo_name)
sample_repo_name
  FROM [bigquery-public-data:github_repos.sample_files]
  WHERE REGEXP_EXTRACT(path, r'\.([^\.]*)$') IN
('java','h','js','c','php','html','cs','json','py','cpp','xml','rb',
'cc','go')
  GROUP BY id
) a
JOIN [bigquery-public-data:github_repos.contents] b
ON a.id=b.id

864.6s elapsed, 1.60 TB processed
```

Spaces vs Tabs - Apply the rules

```
SELECT ext, tabs, spaces, countext, LOG((spaces+1)/(tabs+1)) lratio
FROM (
  SELECT REGEXP_EXTRACT(sample_path, r'\.([^\.]*)$') ext,
         SUM(best='tab') tabs, SUM(best='space') spaces,
         COUNT(*) countext
  FROM (
    SELECT sample_path, sample_repo_name, IF(SUM(line='
')>SUM(line='\t'), 'space', 'tab') WITHIN RECORD best,
           COUNT(line) WITHIN RECORD c
    FROM (
      SELECT LEFT(SPLIT(content, '\n'), 1) line, sample_path,
sample_repo_name
      FROM [fh-
bigquery:github_extracts.contents_top_repos_top_langs]
      HAVING REGEXP_MATCH(line, r'[ \t]')
    )
    HAVING c>10 # at least 10 lines that start with space or tab
  )
  GROUP BY ext
)
ORDER BY countext DESC
LIMIT 100

16.0s elapsed, 133 GB processed
```

Spaces vs Tabs - Results



Who wants to analyze GitHub?

Project maintainers
Project users
Project choosers
Data lovers
YOU!

GitHub


GitHub
treehouse
1:02 / 19:08
Exploring GitHub with BigQuery at GitHub

Way more:

- 1 hour after the dataset announcement @thomasdarimont was able to find all the java projects that declare certain dependency.
- Lakshmanan V "Popular Java projects on GitHub that could use some help" (analyzed using BigQuery and Dataflow).
- Guillaume Laforge "What can we learn from million lines of Groovy code on Github?".
- Filippo Valsorda "Analyzing Go Vendoring with BigQuery".
- Go project uses BigQuery stats to guide design decisions, more than once.
- David Gageot analyzes 281,212 Docker projects.
- Kan Nishida uses R to cluster R packages.
- Aja Hammerly compares most popular gems according to Rubygems.org download data vs GitHub gem calls.
- Sergey Abakumoff looks at the most popular npm packages and trending keywords. Justin Beckwith performs a similar analysis. Sergey follows up with a deeper assessment on why almost empty packages duplicate all over GitHub. Sergey Abakumoff also analyzes Angular vs React messages.
- Brent Shaffer analyzes PHP code and libraries—also test coverage for different languages.
- A full run down by Egor Zhuk, "Yet another analysis of Github data with Google BigQuery".
- John-David Dalton informs the travis-ci team on the counts for Node versions tested.
- Alex Zhitnitsky reviews 779,236 Java Logging Statements, 1,313 GitHub Repositories to determine "ERROR, WARN or FATAL"?
- Florin Badita "Naming conventions in Python import statements". Then "Naming conventions in Python def function()".
- @anvaka "analyzed ~2TB of code to build an index of the most common words in programming languages". Cool visualizations, full code on GitHub, and a lot of comments on reddit.
- Sergey Abakumoff comes back, linking code to StackOverflow.
- Gareth Rushgrove finds all kind of metrics for Puppet.

A series of posts by Robert Kozikowski:

- Advanced GitHub search with BigQuery.
- Top emacs packages used in GitHub repos.
- Visualizing relationships between python packages.

Vic Iglesias and 1 other Retweeted

TravisTorrent @TravisTorrent · Jan 25

Thanks to @Google **#BigQuery**, you can now enjoy 1 TB of lightening-fast SQL queries to **@TravisTorrent** for free: travistorrent.testroots.org/page_access/

8 12

Moritz Beller Retweeted

/r/datasets @reddit_datasets · Jan 27

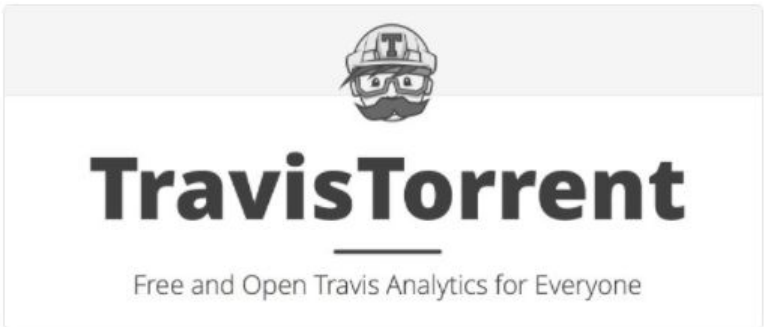
TravisTorrent now in **BigQuery**: 60 facts for each of 3.7 million Travis CI jobs (2.87GB) dlvr.it/ND1zJk

1 1

Georgios Gousios Retweeted

Travis CI @travisci · Jan 16

Become a @travisci log miner (with **TravisTorrent**) and participate in the MSR Mining **Challenge** 2017! blog.travis-ci.com/2017-01-16-tra…



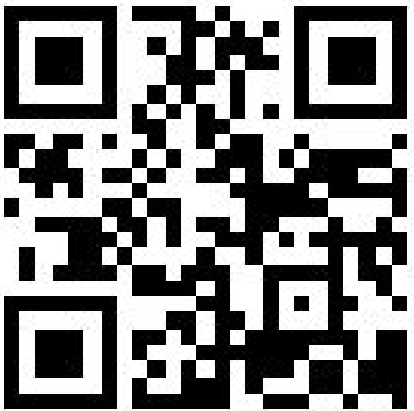
9 19

Questions?

News: reddit.com/r/bigquery
Ask: stackoverflow.com

Felipe Hoffa
@felipehoffa

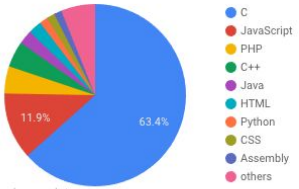
Rate me?



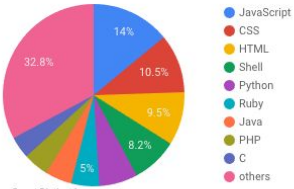
bit.ly/bqfeedback

languages table

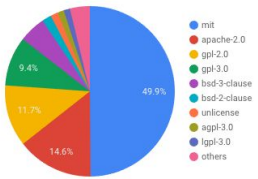
SUM(bytes)



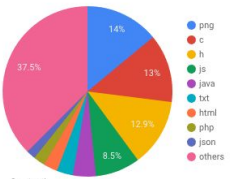
% of repos with language



licenses table

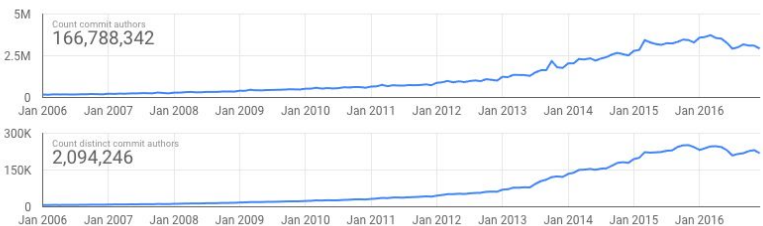


files table

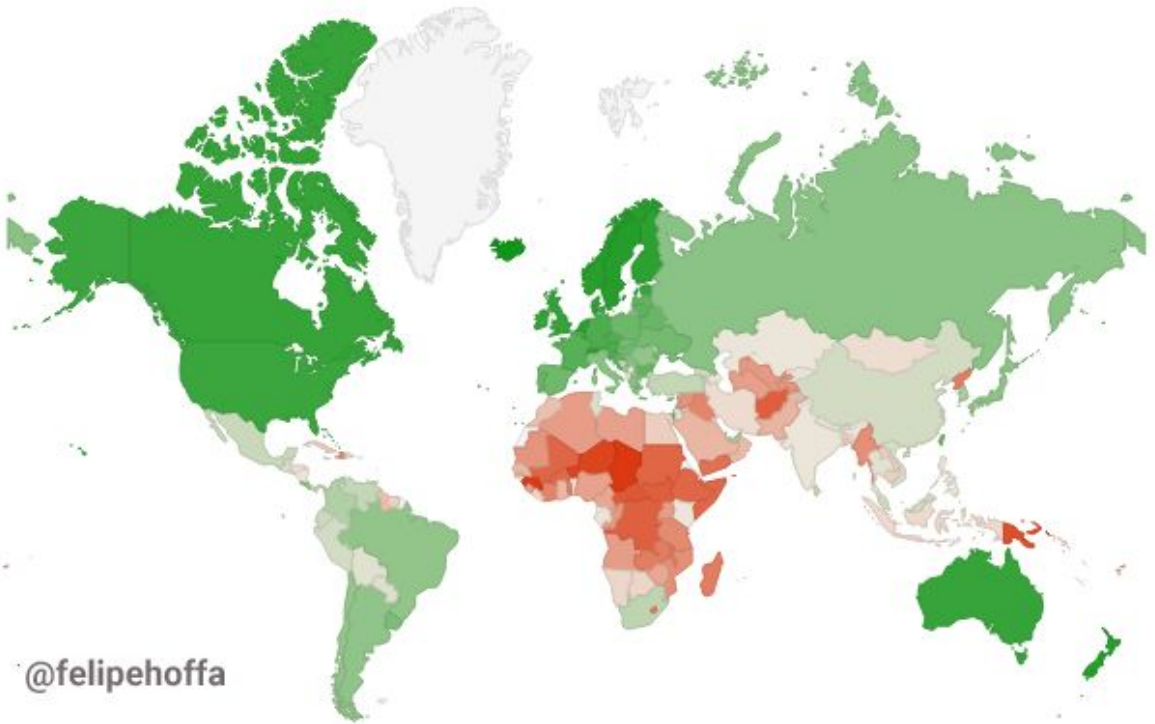


commits table

Jan 1, 2006 - Dec 31, 2016



Top countries by number of unique ids / population



Felipe Hoffa Follow
Developer Advocate @Google. Originally from Chile, now in San Francisco and around the world.
Jun 29, 2016 · 3 min read

Imports in Java from 2013 to 2016: Winners and losers

With all of GitHub open source contents inside BigQuery, we can now go into the actual contents on each file. For example, let's find out the most popular Java imports in 2016:

```
SELECT line, COUNT(*) c
FROM (
  SELECT SPLIT(content, '\n') line
  FROM [fh-bigquery:github_extracts.contents_java_2016]
  HAVING REGEXP_MATCH(line, '^import')
)
GROUP BY 1
ORDER BY 2 DESC
LIMIT 300
```

Row	line	c
1	import java.util.List;	428404
2	import java.util.ArrayList;	249648
3	import java.util.Map;	221261
4	import java.io.IOException;	218662
5	import org.junit.Test;	157233
6	import java.util.HashMap;	128605
7	import java.util.Set;	101870
8	import java.io.File;	100194
9	import org.slf4j.Logger;	86523
10	import org.slf4j.LoggerFactory;	84086
11	import java.util.Arrays;	82459
12	import android.content.Context;	80615
13	import java.util.Collections;	78121
14	import java.util.Collection;	73953

2016 top imports vs 2010 top imports

Row	line	c
1	import java.util.List;	428404
2	import java.util.ArrayList;	249648
3	import java.util.Map;	221261
4	import java.io.IOException;	218662
5	import org.junit.Test;	157233
6	import java.util.HashMap;	128605
7	import java.util.Set;	101870
8	import java.io.File;	100194
9	import org.slf4j.Logger;	86523
10	import org.slf4j.LoggerFactory;	84086
11	import java.util.Arrays;	82459
12	import android.content.Context;	80615
13	import java.util.Collections;	78121
14	import java.util.Collection;	73953

Row	line	c
1	import java.io.IOException;	36503
2	import java.util.List;	35465
3	import java.util.ArrayList;	24212
4	import java.util.Map;	20491
5	import java.io.File;	14455
6	import java.util.HashMap;	13444
7	import java.util.Set;	11242
8	import java.util.Iterator;	10944
9	import org.junit.Test;	10797
10	import java.util.*;	10612
11	import java.util.Collection;	8705
12	import java.io.InputStream;	8650
13	import junit.framework.TestCase;	7721
14	import java.util.Arrays;	7697

GHTorrent on the Google cloud

GHTorrent can be accessed over Google Cloud services. To access the data requires you to have a Google Cloud account. Reasonable use is free of charge and, in the case of BigQuery, it should no longer require a credit card. (Pub/Sub still requires a credit card). You can check what Google considers reasonable at any given moment here.

- Google BigQuery contains an up to date import of the latest GHTorrent MySQL dump.
- Google Pub/Sub exposes real-time streams of GitHub activity.

Both services can be accessed through the Web, the command line (after installing the Google Cloud command line utils) or though various programming languages.

BigQuery

With BigQuery, you can query GHTorrent's MySQL dataset using an SQL-like language (lately, BigQuery also supports vanilla SQL); more importantly, you can join the dataset with other open datasets (e.g. GitHub's own project data, Reddit, TravisTorrent etc) hosted on BigQuery.

To get the most popular programming languages by number of bytes written, run the following:

```
select pl3.lang, sum(pl3.size) as total_bytes
from (
  select pl2.bytes as size, pl2.language as lang
  from (
```

Pub/Sub

Pub/Sub allows subscribers to get events of what is happening on GitHub (or at least GHTorrent's interpretation of what is happening on GitHub) in almost real time. To do so, one needs to *subscribe* to one of the available *topics* with a client in order to start receiving *events*.

The service is complimentary, even though less fine-grained, to GHTorrent's own streaming interface. As is also the case with GHTorrent streaming, the contents of the streams are generated by following the live MongoDB server replication stream. See the code here.

To subscribe to a topic, e.g. `commits`, run the following:

```
gcloud beta pubsub subscriptions create my_commits_subscription --topic projects/ghtorrent-bq/topics/commits
```

To start receiving events, you can try the command line

```
gcloud beta pubsub subscriptions pull --auto-ack --max-messages 5 -- my_commits_subscription
```

The available topics are the following:

```
projects/ghtorrent-bq/topics/commits
projects/ghtorrent-bq/topics/events
projects/ghtorrent-bq/topics/followers
projects/ghtorrent-bq/topics/forks
projects/ghtorrent-bq/topics/issue_comments
projects/ghtorrent-bq/topics/issue_events
projects/ghtorrent-bq/topics/issues
projects/ghtorrent-bq/topics/org_members
projects/ghtorrent-bq/topics/pull_request_comments
projects/ghtorrent-bq/topics/pull_requests
projects/ghtorrent-bq/topics/repo_collaborators
projects/ghtorrent-bq/topics/repo_labels
projects/ghtorrent-bq/topics/repos
projects/ghtorrent-bq/topics/users
projects/ghtorrent-bq/topics/watchers
```