

You can do open source!



/jameslamb



@_jameslamb

Section 1

wtf is open source?

Open source software (OSS) is more than just a bunch of files on a website!

- It's **a software development methodology** that favors discussing and developing in the open.
- It's **a philosophy of software distribution** built on the idea that technologies whose inner workings can be viewed and altered by many people develop more quickly than those that are closely guarded.
- It's **a philosophy of software creation** that acknowledges the fact that software creation is a social activity, not an autonomous task.

Before hosted source code management tools like Github, open source developers still collaborated...by mailing list!

PGSQL-Performance mailing list.

From: Josh Berkus <josh(at)agliodbs(dot)com>
To: pgsql-sql(at)postgresql(dot)org, pgsql-php(at)postgresql(dot)org, pgsql-novice(at)postgresql(dot)org
Subject: PGSQL-Performance mailing list.
Date: 2002-09-27 19:05:49
Message-ID: 200209271205.49756.josh@agliodbs.com
Views: [Raw Message](#) | [Whole Thread](#) | [Download mbox](#)
Thread: 2002-09-27 19:05:49 from Josh Berkus <josh(at)agliodbs(dot)com> 
Lists: [pgsql-novicepgsql-phppgsql-sql](#)

Hey, folks!

I'm mailing to remind everyone about the new PGSQL-PERFORMANCE mailing list. This list was added about 2 weeks ago to provide a forum for the following topics:

1. Postgres server hardware tuning.
2. Postgresql.conf tweaking
3. Index management
4. VACUUMing, ANALYZE and STATISTICS
5. query parsing for efficiency
6. etc.

Please join me there! To subscribe, send an e-mail to:
pgsql-performance-request(at)postgresql(dot)org
... with only the word "subscribe" in the body of the message.

--
-Josh Berkus
Aglio Database Solutions
San Francisco

Browse pgsql-novice by date

	From	Date	Subject
Next Message	D. A.	2002-09-28 01:36:01	"order by" days of the week
Previous Message	Bernard Reißberg	2002-09-27 15:13:46	unsubscribe

Today, OSS is commonly developed in the open, with public source code repositories, Slack groups, JIRA boards, and more

The screenshot shows the GitHub profile for RStudio. The header includes a search bar, navigation links for Pull requests, Issues, Marketplace, and Explore, and a user avator. The main section displays the RStudio logo and repository statistics: 160 Repositories, 13 People, and 0 Projects. Below this, a 'Pinned repositories' section lists six projects: shiny, rstudio, rmarkdown, sparklyr, reticulate, and cheatsheets, each with a brief description and star/fork counts.

Repository	Description	Stars	Forks
shiny	Easy interactive web applications with R	2.9k	1.4k
rstudio	RStudio is an integrated development environment (IDE) for R	2.3k	580
rmarkdown	Dynamic Documents for R	1.2k	542
sparklyr	R Interface to Apache Spark	546	196
reticulate	R Interface to Python	622	94
cheatsheets	RStudio Cheat Sheets	346	101

The screenshot shows the Apache Software Foundation 'Browse Projects' page. It features a navigation bar with 'Dashboards', 'Projects', and 'Issues'. The main content area is divided into 'PROJECT TYPES' (All Project Types, Software, Business) and 'CATEGORIES' (All Categories, MetaModel, Quickstep, Abdera, Accumulo). A table titled 'All Project Types - All Categories' lists projects with columns for Project, Key, and Project Type.

Project	Key	Project Type
Agila	AGILA	Software
aardvark	AAR	Software
Abdera	ABDERA	Software
Accumulo	ACCUMULO	Software
ACE	ACE	Software

The screenshot shows a Slack conversation in the #general channel. The channel name is 'Julia' and it has 1,196 members. The conversation is dated Wednesday, June 20th. A message from Harrison Grodin discusses a new iteration protocol and includes a code snippet for a Julia function. Eric Forgy and Stefan Karpinski respond to the message.

#general
☆ 1,196 | 1 | For general conversation about Julia. Please direct off-topic conversations to #off-topic

Harrison Grodin 10:17 PM
Was just playing with the new iteration protocol, and realized that a potentially-reasonable default implementation of `iterate` could look like this:

```
julia> Base.iterate(iter, state = firstindex(iter)) =  
state > lastindex(iter) ? nothing : (iter[state], nextind(iter, state))
```

Has this already been considered/discussed? Could this have any unintended side effects?
(Side note: is there a reason behind `nextind`'s shortened naming, compared to `firstindex` and `lastindex`?)

Eric Forgy 10:25 PM
nextindex would be so great 😊

Stefan Karpinski 10:27 PM
`nextindex` would probably be better; I'm so tired of making changes though

OSS is distributed with less restrictive licenses than proprietary software

This is a good blog:

<https://writing.kemitchell.com/>

Apache License 2.0

- Grants rights that are worldwide and irrevocable
- If you redistribute code, you must give proper credit

BSD 3-Clause

- Very permissive, allows free copying, distribution, and alteration
- Prohibits use of contributor' names as an endorsement of a redistribution or derivative product

GNU General Public License (GPL)

- A so-called “copyleft” license. If you make modifications, you must release them (also under GPL)

Many popular open source tools were created by for-profit companies, then later released to the world.

Airflow (Airbnb)



Cassandra (Facebook)



Kafka (LinkedIn)




Section 3

How to contribute to projects on Github

In this section, we'll walk through how to contribute to projects on Github. Are you excited???





If you don't have an account yet, visit github.com/join to get started


 [Features](#) [Business](#) [Explore](#) [Marketplace](#) [Pricing](#) [Sign in](#) or [Sign up](#)

Join GitHub

The best way to design, build, and ship software.

 **Step 1:**
Create personal account

 **Step 2:**
Choose your plan

 **Step 3:**
Tailor your experience

Create your personal account

Username

This will be your username. You can add the name of your organization later.

Email address

We'll occasionally send updates about your account to this inbox. We'll never share your email address with anyone.

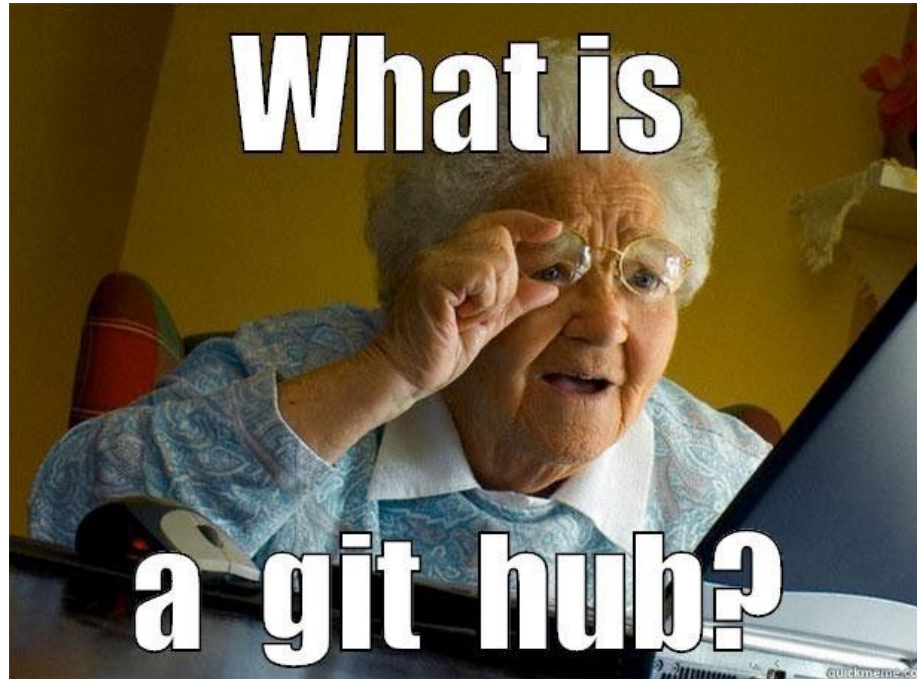
Password

Use at least one lowercase letter, one numeral, and seven characters.

You'll love GitHub

- Unlimited collaborators
- Unlimited public repositories
- ✓ Great communication
- ✓ Frictionless development
- ✓ Open source community

Ok, from this point forward, I'm going to assume you have a Github account.



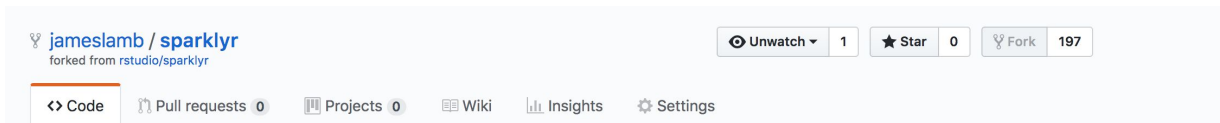
This also assumes that you already have git. If you don't, go to <https://git-scm.com/book/en/v2/Getting-Started-Installing-Git>



(thanks Jenny Bryan, this is bomb)

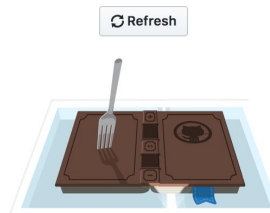
Find a project that you're interested in and fork it

“Forking” a project means making an exact copy of a repository under your personal Github account. For this tutorial, we're going to use “sparklyr”. Go to <https://github.com/rstudio/sparklyr> and click “fork” in the top right corner



Forking rstudio/sparklyr

It should only take a few seconds.



Next, clone that project down to your laptop so we can start making changes!

Clone your fork

`git clone https://github.com/jameslamb/sparklyr`

Add a new remote for the root project (rstudio/sparklyr)

`cd sparklyr`

`git remote add upstream https://github.com/rstudio/sparklyr`

Create a branch to do feature work on

`git checkout -b bugfix/misc`



```
ch-nb-1073:sparklyr jlamb$ git remote -v
origin  https://github.com/jameslamb/sparklyr (fetch)
origin  https://github.com/jameslamb/sparklyr (push)
upstream https://github.com/rstudio/sparklyr (fetch)
upstream https://github.com/rstudio/sparklyr (push)
ch-nb-1073:sparklyr jlamb$
```

Next, make some changes. In this case, I'll take advantage of R's built in spell checker to look for a quick fix in the documentation

```
Console ~/repos/sparklyr/R/
> devtools::spell_check()
WORD FOUND IN
al ml_kmeans.Rd:28,71, ml_lda.Rd:47,49,125,140,150,162
ALS ml_als.Rd:7,59,65,72,74,94
antitonic ml_isotonic_regression.Rd:19
apache ft_quantile_discretizer.Rd:35,82, ml_evaluator.Rd:51, description:2
api ml_evaluator.Rd:51, spark_read_csv.Rd:62, spark_read_json.Rd:45, spark_read_parquet.Rd:47,
spark_read_text.Rd:40
approxQuantile ft_quantile_discretizer.Rd:35,82
aws spark_read_csv.Rd:59, spark_read_json.Rd:42, spark_read_parquet.Rd:44, spark_read_text.Rd:
37
AWS spark_read_csv.Rd:60, spark_read_json.Rd:43, spark_read_parquet.Rd:45, spark_read_text.Rd:
38
Bahmani ml_kmeans.Rd:28,71
bfgs ml_multilayer_perceptron_classifier.Rd:38
bigram ft_ngram.Rd:17
binarize ft_binarizer.Rd:17
binarized ft_r_formula.Rd:63
Binarizer ft_binarizer.Rd:5
Blei ml_lda.Rd:108,140
Bucketizer ft_bucketizer.Rd:5
bucketLength ft_lsh.Rd:26
categoricalCols ft_feature_hasher.Rd:60
cbind sdf_fast_bind_cols.Rd:5,10
checkpointed ml_als.Rd:53, ml_decision_tree.Rd:70, ml_gradient_boosted_trees.Rd:75, ml_lda.Rd:43, ml_ra
ndom_forest.Rd:75
checkpointing ml_lda.Rd:45
ChisqSelector ft_chisq_selector.Rd:21
ChiSqSelector ft_chisq_selector.Rd:5
classpath spark-api.Rd:54
```


Look, I found a bad typo that got copy-pasted through the repo!

```
Results for "tranformation" in ~/repos/sparklyr/R
~/repos/sparklyr/R/ml_feature_pca.R
  1: #' Feature Tranformation -- PCA (Estimator)
~/repos/sparklyr/R/ml_feature_max_abs_scaler.R
  1: #' Feature Tranformation -- MaxAbsScaler (Estimator)
~/repos/sparklyr/R/ml_feature_tokenizer.R
  1: #' Feature Tranformation -- Tokenizer (Transformer)
~/repos/sparklyr/R/ml_feature_count_vectorizer.R
  1: #' Feature Tranformation -- CountVectorizer (Estimator)
~/repos/sparklyr/R/ml_feature_min_max_scaler.R
  1: #' Feature Tranformation -- MinMaxScaler (Estimator)
~/repos/sparklyr/R/ml_feature_string_indexer.R
  1: #' Feature Tranformation -- StringIndexer (Estimator)
~/repos/sparklyr/R/ml_feature_idf.R
  1: #' Feature Tranformation -- IDF (Estimator)
~/repos/sparklyr/R/ml_feature_r_formula.R
  1: #' Feature Tranformation -- RFormula (Estimator)
~/repos/sparklyr/R/ml_feature_standard_scaler.R
  1: #' Feature Tranformation -- StandardScaler (Estimator)
~/repos/sparklyr/R/ml_feature_chisq_selector.R
  1: #' Feature Tranformation -- ChiSqSelector (Estimator)
~/repos/sparklyr/R/ml_feature_stop_words_remover.R
  28: #' Feature Tranformation -- StopWordsRemover (Transformer)
~/repos/sparklyr/R/ml_feature_ngram.R
  1: #' Feature Tranformation -- NGram (Transformer)
```


Once you're done making changes, commit them and push them back to YOUR fork of the repo

Add and commit your changes

```
git add R/*.R
```

```
git commit -m "fixed minor typos in documentation"
```

Push to your fork

```
git push origin bugfix/misc
```

```
ch-nb-1073:sparklyr jlamb$ git status
On branch master
Your branch is up-to-date with 'origin/master'.

Changes not staged for commit:
  (use "git add <file>.." to update what will be committed)
  (use "git checkout -- <file>.." to discard changes in working directory)

        modified:   R/ml_feature_bucketed_random_projection_lsh.R
        modified:   R/ml_feature_chisq_selector.R
        modified:   R/ml_feature_count_vectorizer.R
        modified:   R/ml_feature_idf.R
        modified:   R/ml_feature_max_abs_scaler.R
        modified:   R/ml_feature_min_max_scaler.R
        modified:   R/ml_feature_ngram.R
        modified:   R/ml_feature_pca.R
        modified:   R/ml_feature_r_formula.R
```

Go back to Github in your browser...you should now see a prompt to submit a PR!

The screenshot shows the GitHub interface for the repository **jameslamb / sparklyr**, which is forked from **rstudio/sparklyr**. The repository has 1 pull request, 0 stars, and 197 forks. The navigation bar includes links for Code, Pull requests (0), Projects (0), Wiki, Insights, and Settings. The repository description is "R interface for Apache Spark" with a link to <https://spark.rstudio.com>. The repository statistics show 4,582 commits, 309 branches, 18 releases, 38 contributors, and the Apache-2.0 license. A section titled "Your recently pushed branches:" lists a branch named **bugfix/misc** pushed less than a minute ago, with a button to "Compare & pull request".

Search or jump to... / Pull requests Issues Marketplace Explore

jameslamb / sparklyr
forked from rstudio/sparklyr

Unwatch 1 Star 0 Fork 197

<> Code Pull requests 0 Projects 0 Wiki Insights Settings

R interface for Apache Spark <https://spark.rstudio.com> Edit

Add topics

4,582 commits 309 branches 18 releases 38 contributors Apache-2.0

Your recently pushed branches:

bugfix/misc (less than a minute ago) Compare & pull request

Quick digression...let's talk about PRs

A **pull request** (“PR”) is how you can get your code added to someone else's project. When you submit a pull request, you're telling the maintainers of the destination package

“hey, I did some stuff on my copy of your code. Will you consider adding my stuff to the main code base?”


In other words, you're literally requesting that they pull your code into their project.



Ok back to work...click “compare and pull request” and let’s see what’s in there

Open a pull request

Create a new pull request by comparing changes across two branches. If you need to, you can also [compare across forks](#).

 base fork: `rstudio/sparklyr` ▾ base: `master` ▾ ← head fork: `jameslamb/sparklyr` ▾ compare: `bugfix/misc` ▾

✓ **Able to merge.** These branches can be automatically merged.

You’ll see this stuff at the top of the page.


The “base” branch is the branch on the target repo that you want your code added on top of.

The “compare” branch is the branch (in your repo) with the new code you want added.

Add a short description of what you did in the PR, then click submit!

Open a pull request

Create a new pull request by comparing changes across two branches. If you need to, you can also [compare across forks](#).


 base fork: **rstudio/sparklyr** ▾

base: **master** ▾

← head fork: **jameslamb/sparklyr** ▾

compare: **bugfix/misc** ▾

✓ **Able to merge.** These branches can be automatically merged.



fix minor typos in documentation

Write

Preview

AA B i “ <> 🔗 ☰ ☷ ☹ @ 📌 ↶

Thank you for this excellent package. Please consider this small PR to fix minor misspellings in the documentation.

Attach files by dragging & dropping, [selecting them](#), or pasting from the clipboard.

☒ Allow edits from maintainers. [Learn more](#)

Create pull request

Once you've submitted your PR, now you'll need to hope it builds!

Most projects use one or more “continuous integration” (CI) tools. These run automated tests whenever someone submits a PR.

See that yellow text about “some checks” on the right? That's CI!

Appveyor → testing on Windows

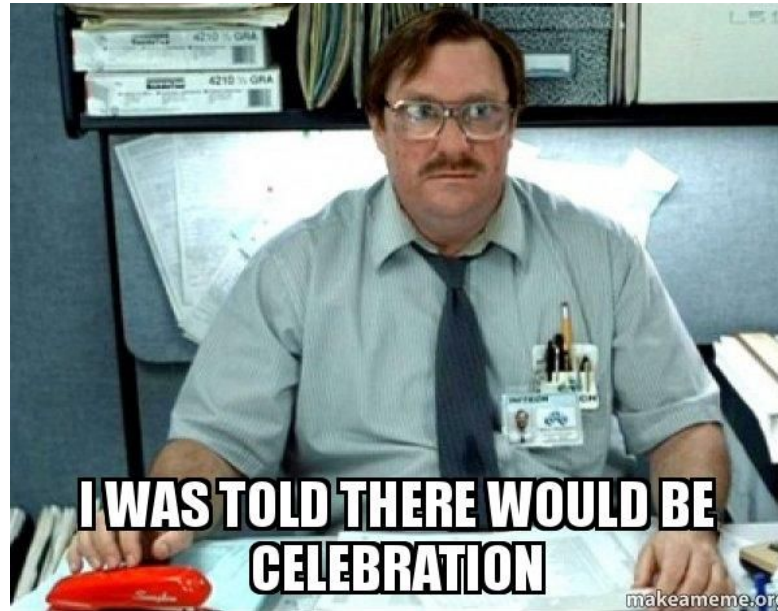
Travis → testing on Linux



A screenshot of a GitHub Pull Request (PR) page. The title is "fix minor typos in documentation #1571". The PR is from "jameslamb:bugfix/misc" to "rstudio:master". It shows 1 commit, 0 checks, and 32 files changed. A comment from "jameslamb" says: "Thank you for this excellent package. Please consider this small PR to fix minor misspellings in the documentation." Below the comment, it says "fixed minor typos in documentation" with a commit hash "86e4cc4". A yellow box highlights the status: "Some checks haven't completed yet" with "2 pending checks". The pending checks are: "continuous-integration/appveyor/pr" (Pending — Waiting for AppVeyor build ...) and "continuous-integration/travis-ci/pr" (Pending — The Travis CI build is in prog...). Below this, a green checkmark indicates "This branch has no conflicts with the base branch". On the right sidebar, there are sections for "Reviewers" (No reviews), "Assignees" (No one assigned), "Labels" (None yet), "Projects" (None yet), "Milestone" (No milestone), and "Notifications" (Unsubscribe button).

If everything goes well, eventually the maintainers will merge your PR!

Getting “merged” means that the maintainers are happy with the changes you’ve proposed and agree to make your code a formal part of the main code base. When that happens, it’s time to celebrate!



So far we've talked about **PRs**, but I also want to talk about **issues**

An issue is a note requesting some action from the maintainers of a project. The name is a little weird, because it's not always something "wrong".

Issues include:

- Bug reports
- Feature requests
- General questions
- Documentation of possible enhancement
- Memes



Github's issue feature is really cool, because it will pick up references in the texts of other issues and PRs (even across projects!) so you follow the discussion around a problem

The screenshot shows a GitHub issue page for the repository 'uptasticsearch'. The issue title is 'Integration tests are very minimal right now #36', which is in a closed state. The issue was opened by 'jameslamb' on September 11, 2017, and has one comment. The issue description states: 'integration tests need a lot of love. We need to test all of the functions in the package on live data in Elasticsearch'. The issue has a 'help wanted' label. A comment from 'jameslamb' 16 days ago states: 'closing this now that #66 was merged'. This comment references issue #66, which is a merged pull request titled 'Add integration tests and make tests easier to develop'. The right sidebar shows settings for the issue, including assignees (none), labels ('help wanted'), projects (none), milestones (none), and notifications (unsubscribe).

<> Code ① Issues 7 Pull requests 1 Projects 0 Wiki Insights Settings

Integration tests are very minimal right now #36

Edit New issue

Closed jameslamb opened this issue on Sep 11, 2017 · 1 comment

jameslamb commented on Sep 11, 2017 Member

uptasticsearch integration tests need a lot of love. We need to test all of the functions in the package on live data in Elasticsearch

jameslamb added the **help wanted** label on Sep 11, 2017

jameslamb referenced this issue on May 21

Add integration tests and make tests easier to develop #66 Merged

jameslamb commented 16 days ago Member

closing this now that #66 was merged

jameslamb closed this 16 days ago

Assignees

No one—assign yourself

Labels

help wanted

Projects

None yet

Milestone

No milestone

Notifications

Unsubscribe

You're receiving notifications because you modified the open/close state.

Thanks for your time!