Arvados - Bug #12194

[API] [Workbench] Colons and ampersands in search should not produce errors

08/30/2017 01:46 PM - Tom Clegg

Status:	Resolved	**Start date:**	08/30/2017
Priority:	Normal	**Due date:**	
Assigned To:	Tom Clegg	**% Done:**	100%
Category:	Workbench	**Estimated time:**	0.00 hour
Target version:	2017-09-13 Sprint		

Description

Currently, typing "bcbio:latest" in the Workbench search box results in a filter like ["any","@@","bcbio:latest"], which produces an API error:

```
{"errors":["#<PG::SyntaxError: ERROR:  syntax error in tsquery: \"bcbio:latest:*\"\n>"],"
error_token":"1504037967+aa140687"}
```

Proposed fix:

On the API side, use plainto_tsquery() instead of to_tsquery(). In Workbench, stop adding ":*" to each search -- that will be dropped by plainto_tsquery() anyway.

This will mean no prefix-matching -- will that be seen as an improvement or a regression?

Subtasks:

Task # 12203: Review 12194-search-always-valid	**Resolved**

Related issues:

Related to Arvados - Bug #12251: [API] Strict checks for "@@" filter operand	**New**	**09/13/2017**

Associated revisions

Revision a6877341 - 09/13/2017 07:37 PM - Tom Clegg

Merge branch '12194-search-always-valid'

refs #12194

Arvados-DCO-1.1-Signed-off-by: Tom Clegg <tclegg@veritasgenetics.com>

Revision da3b350b - 09/13/2017 09:07 PM - Tom Clegg

Merge branch '12194-search-always-valid'

closes #12194

Arvados-DCO-1.1-Signed-off-by: Tom Clegg <tclegg@veritasgenetics.com>

History

#1 - 08/30/2017 07:38 PM - Tom Morris

- Assigned To set to Tom Clegg

- Target version set to 2017-09-13 Sprint

#2 - 08/30/2017 08:41 PM - Tom Clegg

- Status changed from New to In Progress

#3 - 08/31/2017 07:55 PM - Tom Clegg

12194-fulltext-plain-query @ 9e433e06ab2a11fbaaf73c5082c2a64eff596856

#4 - 08/31/2017 08:43 PM - Peter Amstutz

I suspect losing prefix matching would be a regression. One of the use cases for full text search is searching for strings like "Sample123" embedded in filenames. We should probably talk to users and find out if their normal uses will be impacted.

#5 - 09/02/2017 06:33 PM - Tom Clegg

- Subject changed from [API] [Workbench] Fulltext search should use a "plain" query to [API] [Workbench] Colons and ampersands in search should not produce errors

Investigation results:

- Yes, users want to be able to find files called "example_abc123-FOO.bar" by searching "example_abc123"
- to_tsquery('example_abc123:*') matches that example. plainto_tsquery('example_abc123:*') does not match.
- Both to_tsquery('example_abc123-FOO.bar:*') and plainto_tsquery('example_abc123-FOO.bar') succeed.
- Neither form matches with just 'abc123' or 'abc123:*'.
- select ... where column @@ to_tsquery('and:*') returns no rows, and emits lots of logs (one per row?), "query contains only stop words or doesn't contain lexemes". Likewise to_tsquery('').

Here are a few strings that cause server-side errors or surprising results with the current code:

- type "foo:bar" → workbench sends "foo:bar:*" &arr; postgresql syntax error
- type "'foo:bar'" → workbench sends "'foo:bar':*" → interpreted by postgresql as 'foo':* & 'bar':*
- type "and" → workbench sends "and:*" → postgresql matches nothing because "and" is a stop word (ditto "a")
- type "foo & bar" → workbench sends "foo & bar:*" → apiserver translates to "foo & & & bar:*" → postgresql syntax error (ditto "foo | bar")
- type "'foo'" → ok
- type "'foo'" → workbench sends "'foo':*" → postgresql syntax error
- type "'foo & bar'" → workbench sends "'foo & bar':*" → apiserver translates to "'foo & & & bar':*" → matches same results as typing "foo bar"
- type "foo|bar" → workbench sends "foo|bar:*" → postgres searches "foo|bar:*" (with the "or" operator)
- "(foo)" → "(foo):*" → error
- "(foo)a" → "(foo)a:*" → error
- "(foo)&a" → "(foo)&a:*" → OK
- "(foo) a" → "(foo) & a:*" → OK

So, API clients can currently do arbitrary queries by avoiding spaces (delimiting words with &), and users can do this from Workbench by following the same rule and appending "&a" to avoid the implicit prefix-matching and (if the query ends with a parenthesis) syntax errors caused by ":*".

Ideas

- Avoid "only stop words" notices: call to_tsquery(search_string), make sure the result isn't empty, and use the result as the @ operand. If the result is empty, we should ignore/skip the @ filter entirely. This would be consistent with "foo & and" being equivalent to "foo".
- Avoid parse errors: on the API side, instead of blindly replacing whitespace with "&" and expecting the result to be a valid input to to_tsquery(), the API server should either:
  - pass the caller's search string to to_tsquery() directly, or
  - rearrange the query more carefully, e.g.,
    - use something reliable like plainto_tsquery(), which reliably interprets "('((foo |)&| bar:*" as "search for all lexemes and ignore punctuation", i.e., returns "'foo' & 'bar'".
    - add back the prefix-matching operator if given (this is a bit tricky, because plainto_tsquery("foo bar and")="'foo' & 'bar'" -- e.g., if someone types "foo bar and" in workbench, do we want "'foo' & 'bar'", or "'foo':* & 'bar':*"?.
  - ...or auto-detect: accept either a well formed query -- which must start with one of the three chars !'(-- or plain text to pass to plainto_tsquery() to match whole words.
- Meanwhile, Workbench should munge (or not munge) the query according to API expectations: Replace every non-word char with a space, trim trailing space, and (if the resulting query isn't empty) wrap tokens in single quotes, join with &, and append :*.
  - With all three of the API options, and the current API behavior, this would produce the same searches as the current implementation -- except that it can't (even secretly) do boolean "or", and it doesn't produce errors, even if API server hasn't been fixed.
  - The { "and" → "'and':*" → "only stop words" → match-nothing } case would still be a bit odd until/unless we update API to make a special case for that, but at least it wouldn't be a server error.

#6 - 09/05/2017 07:20 PM - Tom Clegg

From discussion offline:

- API should support queries like "foo bar", "foo & bar", "foo*&bar*", "foo* bar", "foo|bar" where "*" enables prefix matching, & means and, and | means or. At least for now, queries with non-alphanumeric characters other than |&* and space should return 422 errors; likewise malformed queries like "&foo", "bar&", "foo&&bar", "foo& |bar".
  - Except: for compatibility, start by silently replacing :* with *.
- Workbench should **not** wrap words in single-quotes as proposed earlier, because that would break the new API.

#7 - 09/08/2017 01:51 AM - Tom Clegg

12194-search-always-valid @ 43eb8f415a1a28bfb721892d51b5ba002ac113ea

(from note-5) "munge (or not munge) the query according to API expectations: Replace every non-word char with a space, trim trailing space, and (if the resulting query isn't empty) ~~wrap tokens in single quotes,~~ join with &, and append :*"

Plus one change: append :* to each term, not just the last term. Seems more predictable.

The resulting query looks like "foo:*&bar:*". This behaves well with the current API, and will still behave well for all user input when the API starts rejecting poorly formed queries.

#8 - 09/11/2017 11:05 PM - Lucas Di Pentima

This LGTM, except for the fact that "or" queries are not included, as described on note-6. Just mentioning this in case it's not on purpose.

#9 - 09/13/2017 08:00 PM - Tom Clegg

- Category changed from API to Workbench

- Status changed from In Progress to Resolved

Split API improvement to #12251