

Item 82

git_comments:

git_commits:

1. **summary:** JENA-1459: highlighting support for jena-text
message: JENA-1459: highlighting support for jena-text This closes #339.

github_issues:

github_issues_comments:

github_pulls:

1. **title:** resolve JENA-1459 add jena text highlighting
body: This PR adds the highlighting feature describe in JENA-1459. This PR includes unit tests for the highlighting feature. Once the PR is approved, I'll submit an update to the jena-text documentation.

github_pulls_comments:

1. There are some conflicts (due to PR #335, I think). Is this simple to resolve? Is that because this PR includes some of the other one?
2. Good grief!. Yes the conflicts are with PR [#335](https://github.com/apache/jena/pull/335). I apparently merged too fast when I saw the approval for 335 and proceeded with the development of the highlight. When I issued this PR no conflicts were showing so I thought all was ok. I had prepared a squashed branch per prior requests. What is the procedure to remedy?
3. It's no big deal-- happens all the time! Do you know how to rebase and resolve conflicts in Git?
<https://help.github.com/articles/resolving-merge-conflicts-after-a-git-rebase/>
<https://help.github.com/articles/resolving-a-merge-conflict-using-the-command-line/>
4. They were simple enough to fix inline via the github tool.
5. Cool. That's essentially an in-line version of the same workflow, minus the rebase. That means you end up with an extra no-real-content commit
(<https://github.com/apache/jena/pull/339/commits/c9453b4c835ac7ab3eb9b0c9bcf8857d11b1a8b8>) but you get where you need to be.
6. I'm glad it was comparatively simple. I was not looking forward to experimenting with rebasing in public ;-)
7. Relax-- it's not nearly as bad as all that. :grin: I often say that having to rebase PRs a lot is good-- it means the project is lively and the code is evolving. For me, the trick (as almost always with git) was to remember that you are making/managing/corralling `_deltas_` (aka commits), not versions. A branch can be thought of as a chain of deltas that starts somewhere and ends with a specific commit. Rebasing means taking a chain of commits that ends one branch and swapping what comes before it to be another branch. So in this case, you would have swapped the prefix of your branch (which has all the commits in master `_except_` those that were in #335) for master itself. You would have had to do exactly the same adjustments, but you would have ended up with a series of commits that appeared (from the POV of seeing how deltas add up to change) as though you had begun work after #335 merged, which makes for a cleaner public history. Of course your commits always keep their metadata, so we will always actually know when they occurred in clock time.
8. Thanks for the help. I appreciate it.
9. Done! (One trivial syntax error fixed.)
10. What was the error? I puzzled that I was not seeing it.
11. `TS_Text.java`, line 54 had a leading backtick character. I thought I could safely fix it :-)+` , `TestTextHighlighting.class`
12. Thanks for fixing. Maybe that got introduced when I "corrected" the conflicts a couple of days ago since I did that inline - I suppose I won't used that approach again, otherwise the tools should have detected it at my end.
13. That is one prob with using the Github on-line editors. But they `_are_` convenient! I use them for corrections to human-centric docs like READMEs, etc.

github_pulls_reviews:

jira_issues:

1. **summary:** add highlighting support to jena-text

description: This issue proposes an improvement to jena-text to include optional highlighting of results via: `{{org.apache.lucene.search.highlight.Highlighter}}` and `{{org.apache.lucene.search.highlight.SimpleHTMLFormatter}}` The improvement will add an optional input argument to `{{TextQueryPF}}` that signals that highlighting should be performed on the Lucene search results; optionally indicates the `_start_` and `_end_` char sequences of a highlighted term; optionally indicates the maximum number of fragments to highlight; and optionally indicates a fragment separator. The highlighted results are bound to the `{{?literal}}` output argument of `{{TextQueryPF}}`. Inclusion of this improvement will introduce a simple extraction of the `_highlight_` option string and a single test for the presence of the option string so that the inclusion of the improvement will be of minimal impact when highlighting is not used. The `_highlight_` option string is passed directly to `{{TextIndex.query(...)}}` and so can be used from code other than `{{TextQueryPF}}`. The simplest use of highlighting is like: `{code} select ?s ?lit where { (?s ?sc ?lit) text:query (skos:prefLabel "one" 100 "lang:en" "highlight:") . }` which will produce results such as: `{code} "another →one← abc"@en {code}` the right-arrow (`\u21a6`) and left-arrow (`\u21a4`) are the default `_start_` and `_end_` highlighting character sequences. These are chosen to be very unlikely to occur in literals. These can be changed easily via `{{"s:"}}` and `{{"e:"}}` in the highlight options, for example: `{code} select ?s ?lit where { (?s ?sc ?lit) text:query (skos:prefLabel "one" 100 "lang:en" "highlight: s:<em class='hilite'> | e:") . }` which will produce results such as: `{code} "another <em class='hilite'>one abc"@en {code}` Coding of this improvement is complete and a PR can be issued if there is agreement that this improvement should be included in jena-text.

jira_issues_comments:

1. Would this be generic for the Lucene module and the Elasticsearch module or limited to Lucene?
2. It is currently limited to `{{TextIndexLucene}}`. The `{{TextIndex}}` i/f includes a call to pass the highlight opts to any implementation of the i/f including the `{{TextIndexES}}` which currently just ignores the highlight opts. I am totally unfamiliar with how highlighting might be controlled in the `{{TextIndexES}}`. As it stands, highlight opts will be silently ignored when passed to Elasticsearch.
3. GitHub user xristy opened a pull request: <https://github.com/apache/jena/pull/339> resolve JENA-1459 add jena text highlighting This PR adds the highlighting feature describe in JENA-1459. This PR includes unit tests for the highlighting feature. Once the PR is approved, I'll submit an update to the jena-text documentation. You can merge this pull request into a Git repository by running: `$ git pull https://github.com/BuddhistDigitalResourceCenter/jena JENA-1459-Highlighting` Alternatively you can review and apply these changes as the patch at: <https://github.com/apache/jena/pull/339.patch> To close this pull request, make a commit to your master/trunk branch with (at least) the following in the commit message: This closes #339 ---- commit 1299d653d295f3dfebaeb25984d8e5d55c3f5245 Author: Chris Tomlinson <ct@...> Date: 2018-01-06T17:53:50Z Merged Highlighting ----
4. Github user afs commented on the issue: <https://github.com/apache/jena/pull/339> There are some conflicts (due to PR #335, I think). Is this simple to resolve? Is that because this PR includes some of the other one?
5. Github user xristy commented on the issue: <https://github.com/apache/jena/pull/339> Good grief!. Yes the conflicts are with PR [#335](<https://github.com/apache/jena/pull/335>). I apparently merged too fast when I saw the approval for 335 and proceeded with the development of the highlight. When I issued this PR no conflicts were showing so I thought all was ok. I had prepared a squashed branch per prior requests. What is the procedure to remedy?
6. Github user ajs6f commented on the issue: <https://github.com/apache/jena/pull/339> It's no big deal-- happens all the time! Do you know how to rebase and resolve conflicts in Git? <https://help.github.com/articles/resolving-merge-conflicts-after-a-git-rebase/> <https://help.github.com/articles/resolving-a-merge-conflict-using-the-command-line/>
7. Github user xristy commented on the issue: <https://github.com/apache/jena/pull/339> They were simple enough to fix inline via the github tool.
8. Github user ajs6f commented on the issue: <https://github.com/apache/jena/pull/339> Cool. That's essentially an in-line version of the same workflow, minus the rebase. That means you end up with an extra no-real-content commit (<https://github.com/apache/jena/pull/339/commits/c9453b4c835ac7ab3eb9b0c9bcf8857d11b1a8b8>) but you get where you need to be.

9. Github user xristy commented on the issue: <https://github.com/apache/jena/pull/339> I'm glad it was comparatively simple. I was not looking forward to experimenting with rebasing in public -;
10. Github user ajs6f commented on the issue: <https://github.com/apache/jena/pull/339> Relax-- it's not nearly as bad as all that. :grin: I often say that having to rebase PRs a lot is good-- it means the project is lively and the code is evolving. For me, the trick (as almost always with git) was to remember that you are making/managing/corralling `_deltas_` (aka commits), not versions. A branch can be thought of as a chain of deltas that starts somewhere and ends with a specific commit. Rebasing means taking a chain of commits that ends one branch and swapping what comes before it to be another branch. So in this case, you would have swapped the prefix of your branch (which has all the commits in master `_except_` those that were in #335) for master itself. You would have had to do exactly the same adjustments, but you would have ended up with a series of commits that appeared (from the POV of seeing how deltas add up to change) as though you had begun work after #335 merged, which makes for a cleaner public history. Of course your commits always keep their metadata, so we will always actually know when they occurred in clock time.
11. Github user xristy commented on the issue: <https://github.com/apache/jena/pull/339> Thanks for the help. I appreciate it.
12. Commit `c9453b4c835ac7ab3eb9b0c9bcf8857d11b1a8b8` in jena's branch `refs/heads/master` from [`~code-ferret`] [<https://git-wip-us.apache.org/repos/asf?p=jena.git;h=c9453b4>] Merge branch 'master' into JENA-1459-Highlighting
13. Commit `40e78407e3e48b9ea3e2195fd833a2c56f497937` in jena's branch `refs/heads/master` from [`~andy.seaborne`] [<https://git-wip-us.apache.org/repos/asf?p=jena.git;h=40e7840>] JENA-1459: highlighting support for jena-text This closes #339.
14. Github user asfgit closed the pull request at: <https://github.com/apache/jena/pull/339>
15. Github user afs commented on the issue: <https://github.com/apache/jena/pull/339> Done! (One trivial syntax error fixed.)
16. Github user xristy commented on the issue: <https://github.com/apache/jena/pull/339> What was the error? I puzzled that I was not seeing it.
17. Github user afs commented on the issue: <https://github.com/apache/jena/pull/339> `TS_Text.java`, line 54 had a leading backtick character. I thought I could safely fix it :-) + ` , `TestTextHighlighting.class`
18. Github user xristy commented on the issue: <https://github.com/apache/jena/pull/339> Thanks for fixing. Maybe that got introduced when I "corrected" the conflicts a couple of days ago since I did that inline - I suppose I won't used that approach again, otherwise the tools should have detected it at my end.
19. Github user ajs6f commented on the issue: <https://github.com/apache/jena/pull/339> That is one prob with using the Github on-line editors. But they `_are_` convenient! I use them for corrections to human-centric docs like READMEs, etc.
20. I've added documentation for the highlighting feature to the Jena Text documentation and committed via "Improve this page".
21. Commit `1821786` from [`~andy.seaborne`] in branch 'site/trunk' [<https://svn.apache.org/r1821786>] JENA-1459: Documentation updates