Item 311

**git_comments:**

1. * Randomizes the ResponseParser to test that both javabin and xml responses parse correctly. See SOLR-15070

**git_commits:**

1. **summary:** SOLR-15070: Remove HashMap usage in SuggestComponent rsp (#2183)
**message:** SOLR-15070: Remove HashMap usage in SuggestComponent rsp (#2183) Prior to this commit, SuggestComponent used a HashMap as part of the response it built on the server side. This class is serialized/ deserialized differently depending on the SolrJ ResponseParser used: a LinkedHashMap when javabin was used, and a SimpleOrderedMap when XML was used. This discrepancy led to ClassCastException's in downstream SolrJ code. This commit fixes the issue by changing SuggestComponent to avoid these types that are serialized differently. "suggest" response sections now deserialize as a NamedList in SolrJ, and the SuggesterResponse POJO has been updated accordingly.

**github_issues:**

**github_issues_comments:**

**github_pulls:**

1. **title:** SOLR-15070: Remove HashMap usage in SuggestComponent rsp
**body:** # Description The HashMap added to the response by SuggestComponent is handled differently by each output format/wt/ResponseParser that Solr supports. This discrepancy causes SolrJ produce different NamedList response structures when different wt's are used. This leads to ClassCastException's elsewhere in SolrJ, particularly when QueryResponse attempts to parse responses which include "suggest" info. # Solution This PR replaces the HashMap in SuggestComponent with a NamedList - ensuring that the NamedList SolrJ produces on the client side is the same regardless of the wire format in use. # Tests Manual testing via SolrJ snippets, as well as a revamping of TestSuggesterResponse to use more than just BinaryResponseParser. # Checklist Please review the following and check all that apply: - [x] I have reviewed the guidelines for [How to Contribute](https://wiki.apache.org/solr/HowToContribute) and my code conforms to the standards described there to the best of my ability. - [x] I have created a Jira issue and added the issue ID to my pull request title. - [x] I have given Solr maintainers [access](https://help.github.com/en/articles/allowing-changes-to-a-pull-request-branch-created-from-a-fork) to contribute to my PR branch. (optional but recommended) - [x] I have developed this patch against the `master` branch. - [x] I have run `./gradlew check`. - [x] I have added tests for my changes. - [ ] I have added documentation for the [Ref Guide](https://github.com/apache/lucene-solr/tree/master/solr/solr-ref-guide) (for Solr changes only).

**github_pulls_comments:**

1. would using another SimpleOrderedMap (which extends NamedList) be better than a NamedList directly?
2. > would using another SimpleOrderedMap (which extends NamedList) be better than a NamedList directly? Maybe? I'll admit I'm not as familiar as I should be with the differences between the two and when each one is preferable. I'll look into it!
3. It looks like SimpleOrderedMap is preferred in the sort of serialization/deserialization usecase involved here, so I've updated the PR to use it instead as Mike suggested. This changes the object used on the server side - the client side already had SimpleOrderedMap objects by virtue of the deserialization implementations themselves (which this PR doesn't touch).
4. If the client side doesn't change, then does that mean this is safe to back port to 8x? Also, looking at the code, CluteringInfo might be similarly affected, would you have any interest in validating that?
5. > If the client side doesn't change, then does that mean this is safe to back port to 8x? Sorry, I think a comment I made above was misleading. This PR as a whole will change the deserialized NamedList response structures that SolrJ users get back. Which makes this approach tough to backport as-is. I do intend to commit a branch_8x fix to workaround the ClassCastException on the client side, but that'll be a different approach. My comment above that starts "This changes the object used on the server side [only]

..." was intended to address in advance a question I thought you might ask following the switch to SimpleOrderedMap. ("Hey I still see the `NamedList` type in QueryResponse, why didn't the Suggest code change there as well?"). But in hindsight I see why that comment reads like I'm saying this is backwards-compatible. Sorry for the confusion. To your other question, I might have interest in looking into ClusterInfo. But if so, I'll probably punt it to a separate PR so that I can wait for feedback on a [mailing list thread](http://mail-archives.apache.org/mod_mbox/lucene-dev/202101.mbox/%3CCAPCX2-%2BuGBgagBWPeD8wT%3Dud9p3_cbWxW77vH1RPZqaNVpZXGA%40mail.gmail.com%3E) that relates to all this (without holding up the Suggest code here).

**github_pulls_reviews:**

**jira_issues:**

**jira_issues_comments:**