

Item 200

git_comments:

git_commits:

1. **summary:** ARROW-2491: [Python] raise NotImplementedError on from_buffers with nested types
message: ARROW-2491: [Python] raise NotImplementedError on from_buffers with nested types
Author: Korn, Uwe <Uwe.Korn@blue-yonder.com> Closes #1927 from xhochy/ARROW-2491 and squashes the following commits: f2300dbf <Korn, Uwe> Test for NotImplementedError 0dbcc34f <Korn, Uwe> ARROW-2491: raise NotImplementedError on from_buffers with nested types

github_issues:

github_issues_comments:

github_pulls:

1. **title:** ARROW-2491: [Python] raise NotImplementedError on from_buffers with nested types
body:
label: test
2. **title:** ARROW-2491: [Python] raise NotImplementedError on from_buffers with nested types
body:
3. **title:** ARROW-2491: [Python] raise NotImplementedError on from_buffers with nested types
body:
4. **title:** ARROW-2491: [Python] raise NotImplementedError on from_buffers with nested types
body:
5. **title:** ARROW-2491: [Python] raise NotImplementedError on from_buffers with nested types
body:
6. **title:** ARROW-2491: [Python] raise NotImplementedError on from_buffers with nested types
body:
label: test
7. **title:** ARROW-2491: [Python] raise NotImplementedError on from_buffers with nested types
body:
8. **title:** ARROW-2491: [Python] raise NotImplementedError on from_buffers with nested types
body:

github_pulls_comments:

1. **body:** Note you've got a flake8 failure.
label: test
2. Failure is due to Plasma

github_pulls_reviews:

1. Can't this be a DataType method or property instead?
2. Yes, this could also be a DataType but as the comment suggests, I have just added this definition here so I can call the C++ function from Python. The user-facing `is_primitive` accepts a DataType.
3. **body:** This raise should probably be tested as well.
label: test
4. Done

jira_issues:

1. **summary:** [Python] Array.from_buffers does not work for ListArray
description: When you get the buffers from a ListArray and feed it back into `{{pyarrow.Array.from_buffers}}` the code fails with a DCHECK: `{code} ./src/arrow/array.cc:247 Check failed: (data->buffers.size()) == (2))` `{code}`
2. **summary:** [Python] Array.from_buffers does not work for ListArray
description: When you get the buffers from a ListArray and feed it back into `{{pyarrow.Array.from_buffers}}` the code fails with a DCHECK: `{code} ./src/arrow/array.cc:247 Check`

failed: (data->buffers.size()) == (2))) {code}

3. **summary:** [Python] Array.from_buffers does not work for ListArray

description: When you get the buffers from a ListArray and feed it back into

{{pyarrow.Array.from_buffers}} the code fails with a DCHECK: {code} ./src/arrow/array.cc:247 Check failed: (data->buffers.size()) == (2))) {code}

4. **summary:** [Python] Array.from_buffers does not work for ListArray

description: When you get the buffers from a ListArray and feed it back into

{{pyarrow.Array.from_buffers}} the code fails with a DCHECK: {code} ./src/arrow/array.cc:247 Check failed: (data->buffers.size()) == (2))) {code}

jira_issues_comments:

1. The `from_buffers` method currently does not provide the necessary feature set for nested data and only works for primitive types. I will add a {{NotImplementedError}} so that at least the segfaults are gone. I'm working on the pickling at the moment and I'll have to get something similar for this working. Once the {{pickle}} roundtrip works, we can also discuss a possible alternative API for {{from_buffers}} for nested types.
2. xhochy opened a new pull request #1927: ARROW-2491: raise NotImplementedError on from_buffers with nested types URL: <https://github.com/apache/arrow/pull/1927> -----
----- This is an automated message from the Apache Git Service. To respond to the message, please log on GitHub and use the URL above to go to the specific comment. For queries about this service, please contact Infrastructure at: users@infra.apache.org
3. pitrou commented on a change in pull request #1927: ARROW-2491: [Python] raise NotImplementedError on from_buffers with nested types URL: https://github.com/apache/arrow/pull/1927#discussion_r183306910 ##### File path: python/pyarrow/types.pxi ##### @@ -75,6 +75,11 @@ cdef bytes _datatype_to_pep3118(CDataType* type): return char +def _is_primitive(Type type): Review comment: Can't this be a DataType method or property instead? -----
- This is an automated message from the Apache Git Service. To respond to the message, please log on GitHub and use the URL above to go to the specific comment. For queries about this service, please contact Infrastructure at: users@infra.apache.org
4. Issue resolved by pull request 1927 [<https://github.com/apache/arrow/pull/1927>]