

Iceberg Python Library Sync Agenda and Meeting Notes

- This doc includes agendas for upcoming syncs and meeting notes from previous discussions
- Please add agenda items that you'd like to discuss
- Everyone is welcome to attend syncs. Join the [iceberg-python-sync](#) list on Google Groups to receive an invitation.
- Join the #python channel in apache-iceberg slack.
- Additional details on [structure](#).

Upcoming Syncs

09/30/2025

Attendees:

-

Agenda [please put the name in the front of the item]:



- [Kevin] Special Edition! Community brainstorm session!

08/26/2025

Attendees:

- Kevin Liu, Alex Stephen, Fokko Driesprong, Sung Yun, Gabriel Igliozi, Yingjian Wu

Agenda [please put the name in the front of the item]:

- [Kevin] 0.10.0rc1 out! (please verify and vote, [devlist thread](#))
- [Kevin] Set timeline for release cycle, at least once every 2 months
 - Could be sooner if we choose
 - Next release, 0.11, target around beginning of November
- [Kevin] Schedule a community brainstorm session, next pyiceberg community sync
 - Send out email on devlist
 - [\[announce\] pyiceberg community brainstorm session 9/30](#)
- [Sung]  RFC: Pylceberg Public API Stabilization and Cleanup for 1.0.0 Release
 -  Pylceberg existing Public APIs
- [Alex] Catalog integration tests
- [Yingjian] Commit retry in pyiceberg
 - Prereq: Delete files, <https://github.com/apache/iceberg-python/pull/2255>
- [Kevin] avro memory leak (?)
 - <https://github.com/apache/iceberg-python/issues/2325>

- Not a memory leak. Issue with how caching manifest works
- TODO:
 - Make cache configurable
 - Explore using cache with memory size limit
 - Explore bringing iceberg-rust memory limit cache over to pyiceberg

Previous Sync Notes

07/29/2025

Attendees: Kevin Liu, Alex Stephen, Drew Gallardo, Ehsan totoni, Talat Uyarer, Tom McCormick, Tushar Choudhary, Yingjian Wu, Gabriel Igliozi, Sung Yun

Agenda [please put the name in the front of the item]:

- [Kevin] 0.10.0 release
 - [0.10 milestone](#), very close!!
 - Pyarrow library dep upperbound [#2258](#)
 - Add support for Pyarrow filesystem specific properties [#2251](#)
- [Tom] ORC support <https://github.com/apache/iceberg-python/pull/2236>
- [Yingjian] Deletion Vector Rust or Pylceberg
 - Add delete file index to pyiceberg and support equality delete reads [#2255](#)
- [Yingjian] Staging snapshot commit
 - <https://test.pypi.org/project/pyiceberg/>
- [Alex] PR Reviews for Table Metadata Update (1, 2) + [Views](#)
- [Kevin] TODO add issue for checking feature parity between fsspec vs pyarrow [#2259](#)
- [Kevin] TODO schedule a community brainstorm session! [#2260](#)

06/24/2025

Attendees: Kevin, Jun, Alex, Isaac, Ehsan, Yingjian, Fokko

Agenda [please put the name in the front of the item]:

- [Kevin] 0.10.0!
 - Align on Table maintenance API, <https://github.com/apache/iceberg-python/pull/2143>
 - Kevin: Send out [DISCUSS] thread
- [Kevin] Iceberg Dev Calendar, <https://iceberg.apache.org/community/#iceberg-community-events>
 -
- [Ehsan] Pandas 3 Iceberg support - partition spec and sort order issues <https://github.com/pandas-dev/pandas/blob/main/pandas/io/iceberg.py>
- [Ehsan] Adding another engine in Pylceberg
- [Yingjian] Progress on <https://github.com/apache/iceberg-python/issues/1678>

- Similar to <https://github.com/apache/iceberg-python/issues/819>
- [Yingjian] Progress on Write support [Equality Delete/ Positional Delete/Deletion Vector] Read Support on Equality Delete
 - <https://github.com/apache/iceberg-python/issues/1818>
- [Alex] Highest priority to-dos
 - <https://github.com/apache/iceberg-python/issues/1856>

05/27/2025

Attendees: Kevin, Fokko, Sreesh, Talat, Jun, Sung

Agenda [please put the name in the front of the item]:

- [Sreesh] Incremental Reads (<https://github.com/apache/iceberg-python/pull/2031> / <https://github.com/apache/iceberg-python/issues/240>)
- [Kevin] pyiceberg-core as part of pyarrow required dep
- [Kevin] Github Copilot for PR reviews
- [Kevin] iceberg dev event calendar
- [Fokko] [Expose Avro reader to PyIceberg](#)
 - Depends on [this one](#)
- [Kevin] Auth API

04/29/2025

Attendees: Kevin, Fokko, Sung, Jun

Agenda [please put the name in the front of the item]:

- [Kevin] 0.9.1 release ([devlist discussion](#))
- [Kevin] redundant ci? <https://github.com/apache/iceberg-python/issues/1942>
- [Kevin] 0.10.0 release
 - Include Pyiceberg-core 0.5.0
- [Sung] AuthManager API
- [Sung] Isolation Levels for Optimistic Concurrency Support

03/25/2025

Attendees: Fokko Driesprong, Kevin Liu, Srinivas Lade, Sung Yun, Talat Uyarer, Tushar Choudhary, Jun He

Agenda [please put the name in the front of the item]:

- V3 support: <https://github.com/apache/iceberg-python/issues/1818>
- 0.9.1?
 - Fix snapshot summary: <https://github.com/apache/iceberg-python/pull/1840>
 - Fix recursion error: <https://github.com/apache/iceberg-python/pull/1830>
- [Srinivas] Status of PyArrowFileIO ADLS Support ([Fokko's attempt](#))

- [Srinivas] Integration of External Dataframe Libraries for Reads / Writes
 - <https://github.com/pola-rs/polars/pull/15018>
- [Kevin] Update <https://iceberg.apache.org/status/>
- [Kevin] April 8/9 <https://www.icebergsummit2025.com/>
- [Kevin] pyiceberg <> pyiceberg-core
 - Possibly avro reader

02/25/2025

Attendees: Fokko Driesprong, Kevin Liu, Srinivas Lade, Jun He

Agenda [please put the name in the front of the item]:

- [kevin] [0.9.0rc2](#), help [verify the release!](#)
- [kevin] [pyiceberg nightly build on testpypi](#)
- [\[kevin\] pyiceberg-core nightly build on testpypi](#)

1/28/2025

Attendees: Matt Martin, Kevin Liu, Om Kenge, Sung Yun, Morfa Morales, Eric Fokko Driesprong, Tushar Choudhary, Sreesh, Jun He

Agenda [please put the name in the front of the item]:

- [Kevin] [`MERGE INTO`](#) implementation
 - Implementation: [Feature: MERGE/Upsert Support #1534](#)
 - `Overwrite_filter`: Overwrite with Filter Conditions Example - Large Amount of Filter Conditions [#1571](#)
 - [Matt] here is how polars extends delta lakes merge command: https://docs.pola.rs/api/python/dev/reference/api/polars.DataFrame.write_delta.html
 - TODO: define specific features of MERGE INTO support
- [Kevin] Datafusion <> Iceberg integration
 - Expression API, parser replacement
 - Runtime, pyarrow compute replacement
- [Kevin] 0.9.0 release?
- [Kevin] [Nightly build](#)
- [Sung] [Nested parsing support](#): [Reference](#) - string or tuple?

12/31/2024

Cancelled due to no items in the agenda.

11/26/2024

Attendees: Andre Anastacio, Fokko Driesprong, Honah, Kevin Liu, Nate Lema, Om Kenge, Raul Cumplido, Sung Yun, Tushar Choudhary, Jun He

Agenda [please put the name in the front of the item]:

- [André] [Proposal PyIceberg Public API definition](#)
 - <https://github.com/apache/iceberg-python/issues/1099>
- [Sung] [Assigning fresh ids in create_table API](#)
 - Instead of changing the behavior of the higher level API, we will contain fresh ID assignment in the higher level create_table API and use a lower level _create_table API that respects the provided IDs, and leverage the lower level API for a separate migrate_table API
- [Kevin] [0.8.1 patch release](#)
 - <https://github.com/apache/iceberg-python/pull/1369>
 - <https://github.com/apache/iceberg-python/milestone/9>
 - <https://github.com/apache/iceberg-python/releases>
- [Tushar] [Partition row count](#)
 - Duck db https://github.com/duckdb/duckdb_iceberg/pull/72

10/29/2024

Attendees: Fokko Driesprong, Marcin Rudolf, Daniel Brenner, Adrian Brudaru, Honah, Jorrit, Kevin Liu, Nate Lema, Raj Ravindra Phadke, Tobias Caouette, Jun He, Sung Yun

Agenda:

- [Kevin] [\[WIP\] PyIceberg Roadmap](#)
- [Kevin] Next release 0.7.x
 - <https://github.com/apache/iceberg-python/pull/1246>
- [Marcin] What are the most important features to implement?
- [Kevin] just FYI, [iceberg devlist](#)

09/24/2024

Attendees: Leonid Makarovsky, Sung Yun, Andre Anastacio, Kevin Liu, Jun He

Agenda [please put the name in the front of the item]:

- [Jun] Continue the discussion on Incremental processing support for pyiceberg
 - <https://netflixtechblog.com/incremental-processing-using-netflix-maestro-and-apache-iceberg-b8ba072ddeeb>
 - <https://py.iceberg.apache.org/api/#add-files>
 - Follow up items
 - Cookbooks

- Add this support implementation to the cookbook
- New low level primitives to pyiceberg
 - Create a table like another table
 - Get data file references between two given snapshot ids or timestamps
- [Sung] 0.8.0 Release
 - 0.8.0 Release Manager - Sung
 - Next Version Release Manager - Jun He
- [Kevin] Feature priorities
- [Kevin] Production use case survey
- [Kevin] V1 table read/write [#1194](#)
- [Kevin] Python <> Rust integration
- [Sung] [1.0.0 Release](#)

08/27/2024

Attendees: Sung, Fokko, Jun, Honah

Agenda [please put the name in the front of the item];

- [Sung] [Pylceberg 0.7.1](#) is out!
- [Sung] [Pylceberg 1.0.0](#) - when and what?
 - Focus on API stability
- [Jun] Incremental processing support to pyiceberg
 - <https://netflixtechblog.com/incremental-processing-using-netflix-maestro-and-apache-iceberg-b8ba072ddeeb>
 - <https://py.iceberg.apache.org/api/#add-files>

07/30/2024

Attendees: Sung Yun, Fokko Driesprong, Kevin, Tushar Choudhary, Abhishek Kapatkar, Jun He

Agenda [please put the name in the front of the item];

- [Sung] RecordBatchReader [Memory Benchmarks](#)
- [Sung] Flat names and issues ([NameMapping](#), [SQL Expression parsing](#))
 - Suggestion to support nested fields by default, and then have a quoted syntax for specifying fields with dot in the name
- [Kevin] Ideas to make package release process easier
 - https://docs.google.com/document/d/10_TffIU9wGhtbp6RP54wdt-i0U1CYK-Zb6RpWmLkRrU/edit?usp=sharing
- [Kevin] Pyiceberg site update cadence, sync with release
 - Add to release process
- [Kevin] Release cadence, 0.7.1

- [Kevin] 0.7.1 release note and highlights

06/25/2024

Attendees: Sung Yun, Fokko Driesprong, Honah, Mehul Batra, Tushar Choudhary, Chinmay Bhat, Jun He

Agenda [please put the name in the front of the item];

- [sung] RecordBatch vs [Table](#), or both?
- [sung] the standards for introducing a new API
 - Send out a dev list mail about the deprecation of Table level API in favor of Transaction level API
 - Python channel announcement
 - Deprecate Table level APIs for operations that have a Transaction equivalent
- [sung] 0.7.0 release
- [kevinliu] Nightly build [#734](#)
- [kevinliu] Schema correctness issue [#751](#)

Python

```
from pyiceberg.types import FixedType
from copy import deepcopy
```

```
deepcopy([FixedType(8), FixedType(16)])
```

```
[FixedType(length=16), FixedType(length=16)]
```

- [Mehul] Pyiceberg 3.12 support pending on ray [#254](#) (Ray: The experimental version of Python 3.12 is now available)
- [kevinliu] Add metadata tables [#511](#)
 - 3 PRs away from all implementations complete ([#614](#), [#828](#), [#594](#))

05/28/2024

Attendees: Fokko Driesprong, Chinmay bhat, Kevin Liu, Abhishek Kapatkar, Jay Chia, Mehul Batra Tushar Choudhary, Jun He

Agenda:

- [kevinliu] Support `version-hint.txt` [#763](#)
 - Let's do it in python side first
- [kevinliu] Announcement: Seattle Apache Iceberg meetup
 - [Join #meetup-seattle](#) on Iceberg slack channel

- [kevinliu] PyIceberg project update
 - Can follow similar idea of <https://github.com/apache/iceberg-rust/issues/348>
 - <https://github.com/apache/iceberg-python/issues/736>
- [Mehul] Improve pyiceberg readme <https://github.com/apache/iceberg-python/issues/774>
 - [Tushar] code reference is missing in mkdocs
- [Chinmay] Implement caching of manifest files (and list)
 - <https://github.com/apache/iceberg-python/issues/595>

04/30/2024

Attendees: Fokko, Tushar, Sung, Drew, Yufei, Adrian, Kevin, Hendrik, Mehul, Abhishek, Honah, Jay

Agenda:

- [Tushar] Merge into / Upsert [#402](#)
 - Partitioned writes [#353](#)
 - Support for partial deletes [#569](#)
- [Tushar] Performance testing
 - Similar to [#428](#)
 - Follow up item to create an issue to track it
 - [Mehul] <https://pola.rs/posts/benchmarks/>
 - we can take inspiration like this for Pyiceberg using tpch benchmark for the library like polars did or any database or compute engine does
- [kevinliu] Consolidating Catalog behavior [#488](#)
 - <https://github.com/apache/iceberg-python/issues/507#issuecomment-2084257385>
- [kevinliu] HadoopFileSystem [#310](#)
 - [Jay] FYI PyArrow filesystem also supports fsspec: <https://arrow.apache.org/docs/python/generated/pyarrow.fs.FSSpecHandler.html>
 - You can wrap an fsspec fs in a pyarrow fs. So for any filesystems that pyarrow doesn't support you can "fallback" on fsspec.
 - [Sung] This is the from_uri method I mentioned earlier that returns FileSystem and str tuple as the result: https://arrow.apache.org/docs/python/generated/pyarrow.fs.FileSystem.html#pyarrow.fs.FileSystem.from_uri
 - [Kevin] Followup: Start a thread in the dev list
- [kevinliu] Iceberg Schema
 - Create a type mapping on the website
 - <https://github.com/recap-build/recap>
 - Support for categorical types: <https://github.com/apache/iceberg-python/issues/450>
- [kevinliu] pyiceberg project
- [Sung] [0.7.0 Release](#) (Sung Yun)

03/26/2024

Attendees: Abhishek Kapatkar, Adrian Qin, Drew Gallardo, Fokko Driesprong, Honah, Jay Chia, Kevin Liu, Mehul Batra, Sung Yun, Jun He

Agenda:

- (Sung Yun) - beginning of May
- 0.6.1 Release (Honah)
 - [V1 table: Correctly convert partition_spec and sort_order \(#544\)](#)
 - [Cast None current-snapshot-id as -1 for Backwards Compatibility](#)
 - [Fix dead links in docs](#)
 - [Fix race condition on `Table.scan` with `limit`](#)
 - ... (feel free to add here)
-

02/27/2024

Agenda:

- No item and cancel it

01/30/2024

Attendees: Adrian Qin, Drew Gallardo, Fokko Driesprong, Hendrik Makait, Honah, Kevin Liu, Sung Yun, Jun He

Agenda:

- Pylceberg 0.6.0 release (Fokko)
 - ETA: this week
 - Release manager: Sung Yun
 - No pending PRs for this release
- [Partitioned Writes using PyArrow](#) (Sung, Adrian)
 - Will submit the PR soon

12/26/2023

Agenda:

- No item and cancel it

11/28/2023

Agenda:

- No item and cancel it

10/31/2023

Agenda:

- No item and cancel it

09/26/2023

Attendees: Abhishek Kapatkar, Jun He

Agenda:

- No item and cancel it

08/29/2023

Attendees: Fokko Driesprong, Max De Bayser, Abhishek Kapatkar, Dhruv Pratap, Jun He

Agenda:

- Tentative release time frame for write data support
- [Abhishek] API to serialize data into concrete schema and partition sec types
- [Pylceberg 0.5.0 release](#)

07/25/2023

Attendees: Fokko Driesprong, Max De Bayser, Abhishek Kapatkar, Jun He

Agenda:

- Any plan for Python CDC support

06/27/2023

Attendees: Fokko Driesprong, Max de Bayser, Abhishek Kapatkar, Jun He

Agenda:

- Pylceberg 0.4.0 release
 - Vote
- Pylceberg 0.5.0 milestones
 - Write support
 - Table column update
 - etc.

05/30/2023

Attendees: Jonas, Murilo, Jun

Agenda:

- No item and cancel it

04/25/2023

Attendees: Fokko, Jonas, Jun

Agenda:

- [Pylceberg 0.4.0 release](#)
- <https://github.com/apache/iceberg/pull/6997>:
 - [Handling non-utc timezone and other precisions in visitor?](#)
- [Write support is next](#)
- Pending issues:
 -

02/28/2023

Attendees: Fokko Driesprong, Rushan Jiang, Abhishek Kapatkar, Jun He

Agenda:

- Upcoming changes
 - Remove python-legacy
 - Might create a branch to keep the python-legacy
- Pylceberg 0.4.0 release: <https://github.com/apache/iceberg/milestone/27>

01/31/2023

Attendees: Samuel Redai Dhruv Pratap, Rushan Jiang, Srinivas Lade, Abhishek Kapatkar Loc Nguyen, Jun He

Agenda: no items for today's meeting and end the meeting early.

11/29/2022

Attendees: Ryan Blue, Abhishek Kapatkar, Dhruv Pratap, Fokko Driesprong, Samuel Redai, Jun He, Martin Durant

Agenda:

- Highlights
 - Progresses in Pylceberg
 - Plan scan, row filter, partition filter
 - PyArrow dataset
 - Support Glue
 - Pending items
 - Stats filter for data file
 - Column projection support ([GH Issue](#))
 - Compatibility test
 - Features at Netflix
 -
- Pylceberg release (0.2.0)
 - Jun He will be the release manager with Fokko's help ([how-to-release](#))
 - Targeted release date: by the end of this week
 - Conversion to pyarrow is experimental
 - Will start a slack thread to continue the release discussion
 - 0.2.0 milestone: <https://github.com/apache/iceberg/milestone/23>

10/25/2022

Attendees: Srinivas Lade, Dhruv Pratap, Jae Sim, Abhishek Kapatkar, Hendrik Makait, Rushan Jiang

Agenda:

- Highlights:
 - Released Apache Pylceberg 0.1.0! Thanks Fokko!
 - <https://pypi.org/project/pyiceberg/>
- Discussion topics
 - Check on progress on [Table Scan \(Plan Files\) API](#) [Dhruv Pratap]
 - Netflix's preliminary use-case integration is dependent on this path.
 - Is it possible to get a quick draft PR out?
 - Follow-up on Read Data ([0.2.0](#)) release [Dhruv Pratap]
 - What's a realistic timeline for this?
 - Communication between Python and Java Client Libraries [Srinivas Lade]
 - APIs for translating common types (Schemas, Types, ...)
 - Testing between libraries

09/27/2022

Attendees: Fokko, Sam, Abhishek, Srinivas, Jun


Agenda:

- Highlights:
 - Release Apache Pylceberg 0.1.0
 - This month, there is again a lot of progress!
 - <https://github.com/apache/iceberg/issues?page=1&q=label%3Apython>
 - Thanks to all the contributors and reviewers.
 - Fokko, Joshua, Dhruv, Sam, Ryan, ...
- Discussion topics
 - Release
 - Apache Pylceberg 0.1.0
 - +1 on the mailing list
 - Manifest
 - Scale out
 - Multi-processing
 - Async io
 - Docs
 - <https://py.iceberg.apache.org/>

08/30/2022

Attendees:

Agenda:

- Highlights:
 - This month, there is again a lot of progress!
 - <https://github.com/apache/iceberg/issues?page=1&q=label%3Apython>
 - Thanks to all the contributors and reviewers.
- Self introduction of new python community members
 -
- Discussion topics
 - Using of parameterized in tests
 - Follow-up on an action Item on Netflix to document its expected Iceberg Python write patterns. [Abhishek Kapatkar / Dhruv Pratap]
 -  Netflix python iceberg data write patterns
 - Pylceberg OSS community is trying to avoid taking on a Pyarrow dependency.
 - Pyarrow versions have a nuisance of not being backwards compatible. So chances of Pylceberg's pyarrow version matching with client's pyarrow version is pretty slim and thus we want to avoid that dependency.
 - Two possible paths for Pylceberg to provide dataframe
 - Long-term path. Pyarrow dependency has plans to natively support deriving dataframe from Iceberg.
 - How does this align with the Pylceberg timeline?

- Pyarrow Iceberg read support will follow Pylceberg Read Data release, and likewise write.
 - Benefits
 - Predicate pushdown to Iceberg metadata layer
 - What does the end-user journey look like?
 - Read/Write data likely through Pyarrow. Pending discussions with Pyarrow OSS community.
 - User loads the Pyarrow dataset and Pylceberg works in the background.
 - Ex: <https://arrow.apache.org/docs/python/dataset.html>
 - Iceberg metadata interaction would still be through Pylceberg.
- Short-term path. Once we have Pylceberg read path, kragle can get the plan files and do the Pyarrow/Panda plumbing until Pyarrow is ready with the native iceberg support.
- Follow-up on Pylceberg 0.1.0 release whereabouts (Read Metadata)
 - Last sync we spoke it was expected sometime in Aug.
 - Fokko will be the release manager for this.
 - Will include CLI support and inspect tables.
 - Working on the release docs.
 - Include REST and Hive Catalog support.
 - Tentative timeline? - RC out this week.
 - Spec and testing left, Fokko is working with Sam on this, and will publish announcements over Slack #python.
 - Read Metadata, load namespaces, load table, Rename table
-

07/26/2022

Attendees: Fokko, Kyle, Sam, Ryan, Daniel, Eduard, Jun

Agenda:

- Highlights:
 - This month, there is again a lot of progress!
 - <https://github.com/apache/iceberg/issues?page=1&q=label%3Apython>
 - Thanks to all the contributors and reviewers.
- Self introduction of new python community members
 -
- Discussion topics
 - 0.1.0 release goals and timeline
 - CLI
 - Show table metadata (schema, partition spec, write order)
 - Cat metadata files?

- Legacy use case: Inspect metadata
 - Hive catalog
- Refactoring base.py -> __init__.py
- S3FS FileIO
 - Performance testing with NVidia, Apple

06/28/2022

Attendees: Ryan Blue, Sam Redai, Daniel weeks, Fokko Driesprong, Dhruv Pratap, Abhishek Kapatkar, Jun He

Agenda:

- Highlights:
 - This month, there are a lot of progress!
 - <https://github.com/apache/iceberg/issues?page=1&q=label%3Apython>
 - Thanks to all the contributors and reviewers: Fokko, chulucninh09, Sam, Ryan, Dan, Jun, etc.
- Self introduction of new python community members
 -
- Discussion topics
 - Renaming both *import name* and eventual *pypi namespace* to `pyiceberg`
 - <https://pypi.org/project/pyiceberg/>
 - Docs
 - Sam will add a docs folder with a page of doc
 - List accessor
 -

05/31/2022

Attendees: Ryan, Sam Redai, Daniel weeks, Shrub Pratap, Fokko Driesprong, Joshua Robinson, Steve, Chu Luc Ninh, Abhishek Kapatkar, Kyle Bendickson, Jun He

Agenda:

- Highlights:
 - We have made a lot of progress in the past month!
 - <https://github.com/apache/iceberg/issues?page=1&q=label%3Apython>
 - Thanks all the contributors and reviewers: Fokko, Dhruv, CircArgs, Sam, Pucheng Yang, Ryan, Kyle Bendickson
- Self introduction of new python community members
 -
- Discussion topics
 - Dropping support for python 3.7
 - YES
 - Switching from tox to pre-commit (PR#[4811](#))

- YES
- Using Poetry (PR #[4844](#))
 - Will ship pre-commit first
- Proposal to use MinIO as part of CI to run S3 based tests (part of PR #[4792](#))
 -
- Using classes generated from OpenAPI spec vs. Custom “FromDict/ToDict” serializers”
 - Have a POC PR first

04/27/2022

Attendees: Nancy Lin, Abhishek Kapatkar, Chu Ninh, Sam Redai, Jun He, Dhruv Pratap, Nick Ouellet, Micah Kornfield, Ryan Blue, Daniel Weeks, Bijan Houle

Agenda:

- Highlights:
 - We have made a lot of progress! Here is the list of merged PRs:
 - PR [#4262](#) [Python] support on Literals (Thanks Hongyue/Steve Zhang)
 - PR [#4468](#) [Python] reorganize expressions module structure (Thanks Hongyue/Steve Zhang)
 - PR [#4590](#) Add PartitionField (Thanks Hongyue/Steve Zhang)
 - PR [#4263](#) [Python] Serialization/Deserialization for types and partition values (Thanks Sam)
 - PR [#4504](#) [Python] Add Schema.find_column_name method (Thanks Sam)
 - PR [#4619](#) Add BoundReference class (Thanks Sam)
 - PR [#4609](#) Add a generic Accessor class (Thanks Sam)
 - PR [#4598](#) Adding file related enums and StructProtocol (Thanks Sam)
 - PR [#4285](#) [Python] Add doctest to tox (Thanks Fokko)
 - PR [#4318](#) [Python] Add Schema class (Thanks Abhishek and Sam)
 - PR [#4416](#) [Python] Python: Add bucket transform (Thanks Jun)
 - PR [#4466](#) [Python] Expression base operations (Thanks Nick)
 - PR [#4436](#) [Python] Updates to literals (Thanks Ryan)
 - PR [#4480](#) [Python] Refactor to use common decimal and datetime util (Thanks Ryan)
 - Self introduction of new python community members
- Discussion topics
 - Predicate design (subclass vs. `op` method)
 - Go with subclassing (and keep open the option to switch to `op` ENUM if a reason becomes clear)
 - Builder pattern
 - <https://github.com/apache/iceberg/issues/3228>
 - <https://github.com/apache/iceberg/issues/4631>
 - Progress on breaking transforms PR

- Terms
- First Catalog implementation
 - RESTCatalog and InMemoryCatalog as the first catalog implementations

03/15/2022

Attendees: Nancy Lin, Abhishek Kapatkar, David Berg, Chu Ninh, Anjali Norwood, Sam Redai, Ryan Blue, Bijan Houle, Jun He, Patrick Ames, Daniel Weeks, Kyle Bendickson

Agenda:

- Highlights:
 - Currently Open PRs (all reviewers welcome!)
 - PR [#3450](#) [Python] support iceberg transforms in python
 - PR [#4262](#) [Python] support on Literals (Thanks Hongyue/Steve Zhang)
 - PR [#4263](#) [Python] Serialization/Deserialization for types and partition values (Thanks Sam)
 - PR [#4285](#) [Python] Add doctest to tox (Thanks Fokko)
 - PR [#4318](#) [Python] Add Schema class (Thanks Abhishek and Sam)
 - Merged PRs
 - PyArrowFileIO PR [#4081](#) (Thanks Sam)
 - Types improvement PR [#4016](#) (Thanks Nick)
- Iceberg class [structure](#) doc (Thanks Sam)
- Discussion topics
 - David Berg would like to discuss his use case for Iceberg Python and discuss his requirements to see if/how the ongoing work satisfies it.
 - Slides:
 - Ryan
 - Largely taken care of those concerns with minimal requirements
 - You can choose which library to open files
 - PyArrow is an optional dependency. If you import that package, you import it.
 - Daniel
 - Metadata operation
 - David
 - Security issues
 - can help iceberg python library tests in the scale
 - Ryan
 - python people doing review and opening PRs
 - e.g. decimal handling
 - David
 - willing to use OSS iceberg python library metadata write path as well

02/07/2022

Attendees: Abhishek Kapatkar, Bijan Houle, Chu Luc Ninh, Daniel Weeks, Kyle Bendickson, Nancy Lin, Mayur Srivastava, Micah Kornfield, Sam Redai, Patrick Ames, Ryan Blue, Jun He

- Highlights
 - Python library changes on the old one
 - <https://github.com/apache/iceberg/pull/3940> (Thanks Eric)
 - <https://github.com/apache/iceberg/pull/3927> (Thanks Eric)
 - <https://github.com/apache/iceberg/pull/3875> (Thanks Pucheng)
 - Expanding primitive types: <https://github.com/apache/iceberg/pull/3839> (Thanks Nick)
 - FileIO ABC: <https://github.com/apache/iceberg/pull/3691> (Thanks Samuel)
 - Under review
 - Transforms PR: <https://github.com/apache/iceberg/pull/3450>
 - PartitionSpec PR: <https://github.com/apache/iceberg/pull/3407>
 - Add __eq__ for types: <https://github.com/apache/iceberg/pull/3965> (merged)
 - Adding TableMetadata object <https://github.com/apache/iceberg/pull/3677>
 - Adding FileIO protocols: <https://github.com/apache/iceberg/pull/4021>
 - Please check <https://github.com/apache/iceberg/projects/7> for all open issues
 - Free feel to take issues
 - In case if someone is assigned but not working on it, feel free to ping and take it.
 - Feel free to open threads in slack python channel about any iceberg specs or Java/python legacy implementations
- Discussions
 - Python legacy vs new one
 - [Ryan] they do not compete each other
 - [Jun] new one is a major version upgrade
 - Types PRs
 - [Ryan] for non-primitive types, those created type objects will never be GC-ed due to memorization, specially struct types. May not use it for struct or nested types.
 - [Micah] Seems to be fine for Fixed types
 - [Daniel] For distributed computing system, it might accumulate a large number of objects never GCed
 - [Kyle] there is a weak ref (<https://docs.python.org/3/library/weakref.html>) module in python
 - [Patrick] there is a core driver process staying in memory
 - Ryan will follow up with Nick in the PR.
 - FileIO for read & write
 - [Sam] possible pyarrow perf issue
 - [Ryan] might not worry about it for now
 - [Micah] be careful if python library is OK to write bytes

- [Ryan] can create adapters to do that
 - [Sam] first file io implementation based on pyarrow
- Read and write and process data locally
 - [Chu] write capacity is performed by other engines
 - [Ryan] will have the capacity to process data locally, e.g.
 - Metadata read, write
 - Data read to pyarrow and write to parquet
 - [Daniel] In addition to the distributed engine, very likely to offer native path to read and write data locally in python
 - [Chu] How about transformation part
 - [Ryan] not necessary to provide processing capacity locally. Iceberg is not a distributed process engine.
 - [Sam] Feel like a narrow use case
 - [Daniel] It requires a lot of work with a lot of complexity. Might be better to let compute engine to handle complicated things
 - [Chu] It's just about simplification for local development
 - [Ryan] able to read and write arrow is the primitive iceberg will provide
 - [Patrick] For merging in Ray, our current merge impl would just need APIs to read the file(s) of the delta(s) to merge from some Data Catalog source and then APIs to write the result back to a target Data Catalog destination
- Cross language (Java <-> Python) integration tests
 - [Micah] is there one?
 - [Ryan] No, but we have correctness tests in Java
 - [Jun] How will we deal with old format written by old library (Java) and do we need to implement the fix in python as well
 - [Ryan] Recommend to read the data correctly
 - [Micah] Have docs to describe those
 - [Ryan] Great idea to document them
 - This is a judgment call
- Tool for integration tests
 - [Ryan] CLI tool seems to be a good way to do that
 - [Micah] might be two types of CLIs

12/15/2021

Attendees: TBD


- Progress check
 - types and literals in one w/ generics: <https://github.com/apache/iceberg/pull/3601>
 - transforms PR: <https://github.com/apache/iceberg/pull/3450>
 - PartitionSpec PR: <https://github.com/apache/iceberg/pull/3407>
 - Please check <https://github.com/apache/iceberg/projects/7> for all open issues
 - Free feel to take issues

- In case if someone is assigned but not working on it, feel free to ping and take it.
 - Feel free to open threads in slack python channel about any iceberg specs or Java/python legacy implementations
- Discussions
 - types & literals and its impacts on transforms and other classes
 - Unification vs performance
 - [Ryan] In Java, we don't care a lot about performance except that it is called in a tight loop, e.g. literal and transform. In the other hand, the goal is not to expect very performant for most paths. Maintainability is more important for those parts.
 - [Patrick] Agree with Ryan
 - [Dan] functions to identity files
 - [Ted] high cardinality table might have a lot of manifest files
 - [Ryan] no need to over-optimize but don't want if statement in the tight loop
 - [Dan] Process engine does care about it and the performance element comes through
 - [Florian] sticks to builtin if we really need it very fast. Otherwise, don't worry about that. Make it maintainable.
 - [Patrick] Do we have any performance tests? I can put some test data.
 - [Ryan] we really want query planning time is not the bottleneck. The bottleneck is reading data and deserializing it from avro, not running filters.
 - [Dan] it is also related to the query pattern and data
 - [Conclusion] no need to optimize performance in most places. In tight loop, we should avoid having if else check etc.
 - Implementation complexity and maintainability
 - [Nick] about maintenance cost, it is in a static part of the code
 - [Ryan] it is a bit hard to review and it would be better to talk about that per individual decisions.
 - [Nick] will try split it into incremental things
 - [Ryan] then we can come with a decision
 - [Sam] if some python engineers can take a look, it then will help us to understand maintainability
 - [Florian] hesitate to use meta programming. Based on his experience, very very few cases might be useful. It is used rarely and might have hard time to understand and maintain it
 - [Xinbin] If it is static, probably fine. Otherwise, better off it
 - [Next step] Nick will further break down the PR and then put some details and examples as well for the community to review.
 - File IO pluggability
 - Optional dependency sets? e.g. ``pip3 install py-iceberg[s3]``

- [Sam] the optional dependency helps users use it locally. But we don't need to publish different package
- [Florian] this is only working for pip. For example, Conda does not support.
- [Ryan] That sounds reasonable. If package manager does not support it, import and try block will continue to work
- [Chu] Two patterns: we can select components to install or we assume and go to code with a try block.
- [Ted] some docs will be helpful
- [Patrick] bring your own implementation and should support inject own implementation
- [Ryan] This is a good point and that's why we bring file io. File IO API is the minimal API you need to support
- [Dan] Bringing customized implementation is very valuable.
- [Ryan] This also brings up a big open question. What returns from the input stream and output stream. What do we return there in Python?
- [Patrick] Useful to look at arrow
 - Specifically, <https://arrow.apache.org/docs/python/generated/pyarrow.NativeFile.html> for IO Streams.
- [Ryan] There are lots of complexity to be file system
- [Florian] IO stream interface
 - <https://docs.python.org/3/library/io.html>
 - Pyarrow <https://github.com/apache/arrow/blob/master/python/pyarrow/fs.py>
 - Focused around <https://filesystem-spec.readthedocs.io/en/latest/index.html>
 - <https://arrow.apache.org/docs/python/generated/pyarrow.NativeFile.html>
- [Dan] there is still possible to ignore file IO
- [Ted] take a look at legacy implementation and will try to come up with some doc and share it with the community

11/10/2021

Attendees: Abhishek Kapatkar, Bijan Houle, Dan Katz, Jason Reid, Kyle Bendickson, Nancy Lin, Neelesh Salian, Sam Redai, Satish Varma Dandu, Chu Luc Ninh, Jun He

- Progress check
 - API update
 - [Abhishek]  Iceberg Python API Design
 - Continue to collect feedback on it offline
 - [Florian] will add some code snippets, OK examples, bad examples
 - Github issue: <https://github.com/apache/iceberg/issues/3283>
 - [Sam] iceberg seems not to be an engine

- [Ted] iceberg file io only needs to provide how we resolve the schema evolution
 - [Chu] separate the responsibility of io engine and manifest
 - [Done] layout PR: <https://github.com/apache/iceberg/pull/3184>
 - [Done] types PR: <https://github.com/apache/iceberg/pull/3234>
 - Should check pyarrow types, expressions, etc.
 - Code formatting, docs, typing
 - pre-commit + black + isort + mypy
 - black formatter
 - transforms PR: <https://github.com/apache/iceberg/pull/3450>
 - PartitionSpec PR: <https://github.com/apache/iceberg/pull/3407>
 - operations PR: <https://github.com/apache/iceberg/pull/3399>
 - expression PR:
 - Please check <https://github.com/apache/iceberg/projects/7> for all open issues
 - Some conventions in the iceberg python implementation
 - [Sam] Bar to use builder pattern is higher in Python
 - [Neelesh] We should focus on what users see and better that users do not see builder
 - [Jun] we will update python doc to list those conventions
- Iceberg development plan discussion
 - catalog plugins
 - [Jun] which ones to build first and also dependency management
 - [Sam] JDBC is an important one
 - [Abhishek] what would be the good default for users
 - [Jason] For Java side, no default. Developers configure them.
 - [Abhishek] Most of end users might not make those decisions
 - [Jason] may pre-install all dependencies to make it easy out of box
 - [Chu] containize their application and want to install only my dependencies. Default pip install to install everything. Users can choose different ones
 - [Chu] no standardized way to do catalog. Do we need to follow the Java way? Or we can standardize it in python
 - [Jason] Java has a clear API but the implementation is not very pythonic. We should focus on the spec instead of following Java.
 - [Ted] Can bring this topic to the general iceberg community sync
 - io plugins
 - [Jason] we should consider both read and write in the first release
 - [Chu] parquet support will satisfy our needs
 - [Sam] pyarrow implementation is quite extensive and might be the first to integrate
 - [Ted] +1 on pyarrow
 - [Jun] <https://github.com/apache/iceberg/issues/3464> about arrow
 - [Jason] for type system, iceberg needs to have its own.
 - [Chu] agree that we should follow iceberg spec

- [Sam] it would be great if we can consolidate literals and types
- Strengthen our collective understanding on use cases
 - Cloudtrail logs: Insert-only small files, randomly written into S3 bucket. When receiving S3 notification about new written files, lambda function kicks in and do json read, then transform data to iceberg table and append file into target S3 bucket in parquet format. Doing this using Spark/Flink seems to be overkill because cloudtrail raw logs are relatively small (only 20-50kb per file), native Python lambda can handle this well with small cost.
 - General CDC processing with event-sourcing style: AWS DMS can deliver CDC files into S3 bucket, using the same technique with Cloudtrails, trigger lambda function using S3 events. CDC files are about 1-5mb in size, lambda function should be able to run simple parquet read (using pandas or pyarrow) then convert to iceberg table and do upsert. Expected RAM to run those lambda function: 128MB (lowest RAM of lambda).
 - Scheduled compaction: Lambda function to compact & repartition small iceberg files in batch job style to optimal size (128MB). Iceberg python lib will automatically calculate the splits based on split size parameter.
- Integration with distributed compute engine (dask, ray): For larger-than-local dataset, dask/ray will come to play. However, dask/ray can introduce multiple conflict write due to current iceberg design on write commit. Python lib should be able to integrate with dask/ray engine with some configurable orchestration capability to avoid repeated conflict commits.
- Iceberg read & write proposal discussion
 - Design doc link (TBA)

10/12/2021

Attendees: Abhishek Kapatkar, Bijan Houle, Chen Song, Dan Katz, Daniel Weeks, Florian Jetter, Jarod Maupin, Jason Reid, Kyle Bendickson, Nancy Lin, Neelesh Salian, Sam Redai, Satish Varma Dandu, Jun He

- Progress check
 - Continue the high level API discussion
 - Revisit design doc: [Iceberg Python Library Design](#)
 - [Jason] structure looks good
 - [Dan] may discuss APIs in future meeting once we have more concrete implementation
 - [Florian] will have an isolation layer between iceberg and others
 - [Dan] current priority to enable engine developers. Also do want a user facing API. But we want to have a robust developer API first.
 - [Florian] primary API integration with dask requires table APIs, etc. For example, query planning API might be important for engine developers.
 - [Dan] Seems similar to what Java developers need as well
 - [Florian] camelcase, etc, small items. More critically, how dask commits
 - [Jason] Metadata exploration, SQL support?

- [Dan] Adding SQL layer is engine specific
- [Jason] any comments from Dask or Ray developers
- [Florian] SQL is not needed. If there is table.describe, should be enough
- [Dan] there are metadata tables and data tables.
- [Jason] Seems very SQL centric. Feel not very pythonic, table object, attributes to access the table instead referencing dot table version etc.
- [Dan] separate metadata and data for distributed processing. For higher level metadata, it might not need distributed processing
- [Jun] it might be better to be consistent across Java, Python, SQL
- [Jason] python can have different APIs to help users exploration
- [Florian] python is easier to set up for exploration
- [Dan] It is hard to set up Java iceberg library for exploration. Don't think we need to worry about the transition from Java to Python. Further question is Native python versus engine python.
- [Jason] nice to explore metadata over native python way
- [Dan] basic exploration api
- [Florian] easy to use, explore with very small data and don't want to set up dask, ray, etc.
- [Dan] iceberg features, schema evolution etc. keep these things in mind and continue to revisit it once we have concrete implementations
- [Dan, Jason] can come later but no
- Gists update?
 - [Abhishek] [Iceberg Python API Design](#)
 - Please review it offline
 - [Florian] will add some code snippets, OK examples, bad examples
 - Github issue: <https://github.com/apache/iceberg/issues/3283>
- [Done] layout PR: <https://github.com/apache/iceberg/pull/3184>
- types PR: <https://github.com/apache/iceberg/pull/3234>
 - Should check pyarrow types, expressions, etc.
 - Code formatting, docs, typing
 - pre-commit + black + isort + mypy
 - black formatter
- expression PR:
- transforms PR:
- Please check <https://github.com/apache/iceberg/projects/7> for all open issues
- Iceberg development plan discussion
 - catalog plugins
 - io plugins
- Iceberg read & write proposal discussion
 - Design doc link (TBA)

09/27/2021

Attendees: Daniel Weeks, Jason Reid, Ryan Blue, Bijan Houle, Florian Jetter, Abhishek Kapatkar, Nancy Lin, Lyle Bendicks, Jack Ye, Ted Gooch, Arthur Wiedmer, Dan Katz, Jarod Moupin, Jun He

- Progress check
 - Iceberg python library structure design discussion
 - Design doc: [Iceberg Python Library Design](#)
 - [Jack] Module plugin
 - [Dan] may be too early to discuss it
 - [Florian] code locality, import less is better
 - [Florian] high level public api, keep public API small, prefer `from iceberg import table``, etc.
 - [Abhishek] define those public apis in `__init__.py`
 - [Florian] not use duck typing for important classes, not use meta class, start with a little stricter
 - [Abhishek] strict typing is great
 - [Dan] strict type public and internally use dynamic typing
 - [Florian] starting with stricter will help later
 - [Abhishek] have some typing checker, e.g. type check library, mypi, typechecked
 - [Ryan] no need
 - [Bijan] type hinting is good for developers, but runtime type check seems no need
 - [Florian] runtime type check might have perf issue, start with static linting first, can use type annotation to statically check the type
 - [Bijan] start with a py file first, and if too big and then create a module
 - [Ryan] keep things flat and not group things as module if unnecessary
 - [Jason] code locality
 - [Florian] with io as module is helpful when importing, can let people to install iceberg without extra dependencies
 - [Ryan] +1 for exposing catalog constructors
 - Layout PR: <https://github.com/apache/iceberg/pull/3184>

09/08/2021

Attendees: Ryan Blue, Daniel Weeks, Chinmay Chandak, Bijan Houle, Jason Reid, Satish Varma Dandu, Ted Gooch, Abhishek Kapatkar, Jack Ye, Alexander Wu, Rohi Kulkarni, Jun He, Patrick Ames


- Iceberg python library brainstorm session
 - Follow up PR

- <https://github.com/apache/iceberg/pull/3074>
- Will refactor and separate it into 2 PRs
- Project structure
 - Dependency management
 - [Jason] very slim core without arrow etc.
 - [Ted & Chinmay] arrow release sometimes breaks
 - [Ted] package functions
 - [Daniel] how many modules we want
 - Should we separate api, core, etc. like Java?
 - [Jason] seems not pythonic
 - [Ryan] base classes can be actual classes to use
 - E.g. SPI classes
 - [Jason] duck typing, no strong contract
 - [Jun] will python abc (Abstract Base Classes) help?
 - [Daniel] iceberg can use the available python io
 - [Ryan] table operations class, we can swap one with another. How to tell devs how to implement that
 - [Daniel] do we need lots of pluggable implementations in Python?
 - May not do it everywhere similar to Java
 - [Ryan] ABC classes make senses in some cases
 - [Danie] Catalog and table implementations seem good fit
 - [Ted] file io
 - [Alexander] Ray node provider interface:
 https://github.com/ray-project/ray/blob/master/python/ray/autoscaler/node_provider.py
 - [Jason] Here is Dask's "abstract" Engine class that Iceberg would likely want to provide an implementation for to provide data to Dask:
 <https://github.com/dask/dask/blob/474f479e1b5e77e831da02232a84a9f49508889b/dask/dataframe/io/parquet/utils.py#L9>
 - Should keep a few high level ones, table, catalog, scan with interface
 - How to break it down
 - [Ryan] maybe follow java package
 - [Jun & Daniel] Approach from bottom up, low layer primitive first, e.g. Types
 - [Jason] top down design and bottom up implementation
 - High level design for layout
 - [Daniel] a commit with the layout
 - [Bijan] goal of this work?
 - [Jason] to be user friendly
 - [Daniel] More consumable for python users and help python devs to contribute
 - [Bijan] will re-packaging help?

- Pretty clean break is what we are looking for
- Lots of circular dependency
- [all] top down design and bottom up implementation
 - 1 PR for the package structure first
 - In addition to docs for engine developers, 1 doc or PR to describe user facing interface looks like
 - [Abishek] Design doc for the design of each interface, classes?
 - [Daniel] Seems unnecessary
 - [Ryan] Current high level design doc is written for engine developers. May reconsider high level API
 - [Daniel] What the user facing APIs look like
 - [Ryan] e.g. immutable scan in Java might be not pythonic
 - [Daniel] may follow pyspark convention for user facing
 - [Chinmay] will start some gists sharing with the community
 - [Patrick] will be happy to provide some info
 - Ray node provider interface:
https://github.com/ray-project/ray/blob/master/python/ray/autoscaler/node_provider.py
 - [Ted & Abishek] will also share some thoughts
- Python 3 support only
 - py37,py38,py39 in the current tox setting
- PyPI package name candidates
 - apache-iceberg
 - iceberg-py
 - **py-iceberg [+6]**
 - piceberg
 - pyceberg
 - icePyrg
- Common guideline for devs
 - Unit tests are based on Pytest
 - Type hints are highly recommended
 - Google-style Python docstrings:
https://sphinxcontrib-napoleon.readthedocs.io/en/latest/example_google.html
 - Guidelines when to use dict, when to use class, etc.
- [will be moved to the next sync meeting] Iceberg read & write proposal discussion
 - Design doc link (TBA)

08/18/2021

Attendees: Ryan Blue, Chen Song, Carl Steinbach, Daniel Weeks, Eduard Tudenhoefner, Jason Reid, Leonardo Nascimento, Ryan Murray, SzeHon Ho, Ted Gooch, Jun He, Alexander Wu, Jack Ye, Patrick Ames, Zhe Zhang, Satish Dandu

- Logistics
 - Set up [iceberg-python-sync](#) google group
 - Set up #python slack channel
 - Sync frequency, monthly, bi-monthly, etc.
 - Monthly first
- Go over the feature plan doc
 -  Iceberg Python Library Feature Plan
 - Currently, it is a translation of Java
 - [Jun] Current implementation is a mirror of Java and sometimes, not very optimal in Python
 - [Ryan] should we write it in a python way?
 - [Dan] implement it in Pythonic way
 - [Ryan] +1, Pythonic
 - [Jack] consider to expose query planning to the Ray
 - [Ryan] focus on iceberg library python
 - [Dan] lots of fundamental change and may be easier to start a new python library
 - [Ted] Need a top down redesign
 - [Jun] maybe create a new folder to keep the new implementation
 - [Jason] piceberg
 - [Dan] either new folder (rename the current one) or branching
 - Let's rename the existing one to python_legacy
 - Create a new one (python, piceberg, IcePyrg)
 - Release
 - Apache release is a source release and go over the apache release process
 - Aware of licencing
 - If taking code from other places, note where the code came from
 - Not pushing anything to pypi
 - Cannot push anything voted by PMC
 - Versioning between Java and Python
 - Will not sync
 - Assume separate
 - Should be spec implementation, no need to have feature parity
 - Keep feature metrics in website
 - Capability metrics in Python
 - Rename, create a new one, pull the changes from old to this one,
 - Features in 0.1.0 release
 - a round trip with some limits first
 - read a table, write a table,
 - 0.1 read, 0.2 write?
 - Release often
 - Avro

- Read, write parquet can be separated

- Existing function is already substantial
 - Limited round trip would be helpful
 - Release cadence
 - Initial may take longer
 - Focus on V1 implementation first
 - Write support
 - V2 only?
 - CDC for python?
 - Compaction
 - Another brainstorm session after re-structuring the folder
 - To finalize the structure
- Roadmap discussion
 - Discuss refactoring project
 - Same versus separated repository
 - Same repo
 - (In)dependent of Java release
 - Independent release
 - Complete Iceberg Table V1 Support
 - First V0.1 release
 - Add Iceberg Table V2 Support
 - Catalog support
 - Actions support
 - Python CLI for operation and maintenance
 - Can also be used for tutorial
 - Improve documentation with examples
 - Generate javadoc similar website
- Release plan
 - First python library release
 - Milestone: <https://github.com/apache/iceberg/milestone/3>