

splunk-add-on-for-microsoft-cloud-services

Created with  FOSSA

3rd-Party Software Report for **splunk-add-on-for-microsoft-cloud-services**

This document lists the open source software and other third-party components used in splunk-add-on-for-microsoft-cloud-services.

Any information contained in this report is subject to the terms of your FOSSA agreement and any applicable export control laws.

Generated
July 2, 2026

Revision
c9aa2ba077bea93143d69f8281f748ab6bb07438

Dependencies Summary

This section shows the summary of all packages in splunk-add-on-for-microsoft-cloud-services.

Package	Declared License(s)	Discovered License(s)
@splunk/react-icons 4.14.0	Apache-2.0, proprietary-license	None
@splunk/react-ui 4.47.1	Apache-2.0, proprietary-license	MIT
@splunk/splunk-utils 2.3.4	Apache-2.0, proprietary-license	None
@splunk/themes 0.19.0	Apache-2.0, proprietary-license	None
@splunk/ui-utils 1.12.0	Apache-2.0, proprietary-	None

	license	
azure-core 1.39.0	MIT	None
azure-data-tables 12.7.0	MIT	None
azure-eventhub 5.15.1	MIT	None
azure-eventhub-checkpointstoreblob 1.2.0	MIT	None
azure-identity 1.25.3	MIT	None
azure-mgmt-compute 38.0.0	MIT	None
azure-mgmt-resource 25.0.0	MIT	None
azure-storage-blob 12.29.0	MIT	None
cattrs 23.1.2	MIT	None
certifi 2025.8.3	MPL-2.0	None
cryptography 48.0.0	Apache-2.0	None
exceptiongroup 1.3.1	Python-2.0, MIT	None
grpcio 1.74.0	MPL-2.0, BSD-3-Clause, Apache-2.0	apache-2.0 WITH llvm-exception, BSD-2-Clause, Info-ZIP, ISC, MIT, mit-old-style, mit-old-style-no-advert, Zlib
idna 3.16	BSD-3-Clause	None
isodate 0.7.2	BSD-3-Clause	None
opentelemetry-api 1.39.1	Apache-2.0	None
opentelemetry-exporter-otlp-proto-grpc 1.39.1	Apache-2.0	None
opentelemetry-sdk 1.39.1	Apache-2.0	None
raw-loader 4.0.2	MIT	None

react 16.14.0	MIT	None
react-dom 16.14.0	MIT	None
requests 2.32.5	Apache-2.0	None
solnlib 8.1.0	Apache-2.0	None
sortedcontainers 2.4.0	Apache-2.0	None
splunk-sdk 2.1.0	Apache-2.0	None
splunktalib 3.0.6	Apache-2.0	None
splunktaucclib 6.6.0	Apache-2.0	None
ssh://git@github.com/splunk/splunksdcsdk.git v1.2.1	None	None
styled-components 5.3.11	MIT	None
u-msgpack-python 2.8.0	MIT	None
underscore 1.13.8	MIT	None
urllib3 1.26.19	MIT	None
websocket-client 1.8.0	Apache-2.0	None
zipp 3.23.0	MIT	None

Dependencies Details

This section shows the details for each package in splunk-add-on-for-microsoft-cloud-services.

@splunk/react-icons 4.14.0

npm+@splunk/react-icons\$4.14.0

[Direct](#)

Description

Library of various React icons

Package Info

Package Author(s)

devinfo@splunk.com, vtsvetkov@splunk.com, webplatform@splunk.com

Package Manager

NPM

Project URL

<https://www.npmjs.com/package/@splunk/react-icons>

Package Download URL

<https://registry.npmjs.org/@splunk/react-icons/-/react-icons-4.14.0.tgz>

Dependency Path

@splunk/react-ui > @splunk/react-icons, @splunk/react-icons

Declared License(s)

Apache-2.0

Copyright(s)

None

proprietary-license

Copyright(s)

Splunk Inc.

Discovered License(s)

None

Notice File(s)

None

@splunk/react-ui 4.47.1

npm+@splunk/react-ui\$4.47.1

Direct

Description

Library of React components that implement the Splunk design language

Package Info

Package Author(s)

devinfo@splunk.com, vtsvetkov@splunk.com, webplatform@splunk.com

Package Manager

NPM

Project URL

<https://www.npmjs.com/package/@splunk/react-ui>

Package Download URL

<https://registry.npmjs.org/@splunk/react-ui/-/react-ui-4.47.1.tgz>

Dependency Path

@splunk/react-ui

Declared License(s)

Apache-2.0

Copyright(s)

None

proprietary-license

Copyright(s)
Splunk Inc.

Discovered License(s)

MIT

Copyright(s)
2014 Call-Em-All

Notice File(s)

None

@splunk/splunk-utils 2.3.4

npm+@splunk/splunk-utils\$2.3.4

Direct

Description

Library of common Splunk Enterprise utilities

Package Info

Package Author(s)
devinfo@splunk.com, vtsvetkov@splunk.com, webplatform@splunk.com

Package Manager
NPM

Project URL
<https://www.npmjs.com/package/@splunk/splunk-utils>

Package Download URL

<https://registry.npmjs.org/@splunk/splunk-utils/-/splunk-utils-2.3.4.tgz>

Dependency Path

@splunk/splunk-utils

Declared License(s)

Apache-2.0

Copyright(s)

None

proprietary-license

Copyright(s)

Splunk Inc.

Discovered License(s)

None

Notice File(s)

None

@splunk/themes 0.19.0

npm+@splunk/themes\$0.19.0

Direct

Description

Theme variables and mixins for the Splunk design language

Package Info

Package Author(s)

devinfo@splunk.com, vtsvetkov@splunk.com, webplatform@splunk.com

Package Manager

NPM

Project URL

<https://www.npmjs.com/package/@splunk/themes>

Package Download URL

<https://registry.npmjs.org/@splunk/themes/-/themes-0.19.0.tgz>

Dependency Path

@splunk/themes

Declared License(s)

Apache-2.0

Copyright(s)

None

proprietary-license

Copyright(s)

Splunk Inc.

Discovered License(s)

None

Notice File(s)

None

@splunk/ui-utils 1.12.0

npm+@splunk/ui-utils\$1.12.0

[Direct](#)

Description

Library of common UI utilities

Package Info

Package Author(s)

devinfo@splunk.com, webplatform@splunk.com

Package Manager

NPM

Project URL

<https://www.npmjs.com/package/@splunk/ui-utils>

Package Download URL

<https://registry.npmjs.org/@splunk/ui-utils/-/ui-utils-1.12.0.tgz>

Dependency Path

@splunk/splunk-utils > @splunk/ui-utils, @splunk/react-ui > @splunk/ui-utils,
@splunk/react-ui > @splunk/react-icons > @splunk/ui-utils, @splunk/react-
icons > @splunk/ui-utils, @splunk/ui-utils

Declared License(s)

Apache-2.0

Copyright(s)

None

proprietary-license

Copyright(s)

Splunk Inc.

Discovered License(s)

None

Notice File(s)

None

azure-core 1.39.0

pip+azure-core\$1.39.0

Direct

Description

Microsoft Azure Core Library for Python

Package Info

Package Author(s)

Microsoft Corporation <azpysdkhelp@microsoft.com> License-Expression: MIT

Package Manager

Pip

Project URL

<https://pypi.org/project/azure-core/>

Package Download URL

https://files.pythonhosted.org/packages/34/83/bbde3faa84ddcb8eb0eca4b3ffb3221252281db4ce351300fe248c5c70b1/azure_core-1.39.0.tar.gz

Dependency Path

azure-storage-blob > azure-core, azure-mgmt-resource > azure-mgmt-core > azure-core, azure-mgmt-compute > azure-mgmt-core > azure-core, azure-identity > azure-core, azure-eventhub-checkpointstoreblob > azure-core, azure-eventhub-checkpointstoreblob > azure-eventhub > azure-core, azure-eventhub > azure-core, azure-data-tables > azure-core, azure-core

Declared License(s)

MIT

Copyright(s)
Microsoft Corporation.

Discovered License(s)

None

Notice File(s)

None

azure-data-tables 12.7.0

pip+azure-data-tables\$12.7.0

Direct

Description

Microsoft Azure Azure Data Tables Client Library for Python

Package Info

Package Author(s)
ascl@microsoft.com

Package Manager
Pip

Project URL
<https://github.com/Azure/azure-sdk-for-python/tree/main/sdk/tables/azure-data-tables>

Package Download URL
https://files.pythonhosted.org/packages/ad/b8/e76cfa7f5f847722664c6f8108c391deadcb9d9aad579650c128affd4669/azure_data_tables-12.7.0.tar.gz

Dependency Path

azure-data-tables

Declared License(s)

MIT

Copyright(s)

Microsoft Corporation.

Discovered License(s)

None

Notice File(s)

None

azure-eventhub 5.15.1

pip+azure-eventhub\$5.15.1

Direct

Description

Azure Event Hubs client library for Python Azure Event Hubs is a highly scalable publish-subscribe service that can ingest millions of events per second and stream them to multiple consumers. This lets you process and analyze the massive amounts of data produced by your connected devices and applications. Once Event Hubs has collected the data, you can retrieve, transform, and store it by using any real-time analytics provider or with batching/storage adapters. If you would like to know more about Azure Event Hubs, you may wish to review: [What is Event Hubs] (<https://docs.microsoft.com/azure/event-hubs/event-hubs-about>)? The Azure Event Hubs client library allows for publishing and consuming of Azure Event Hubs events and may be used to:

- Emit telemetry about your application for business intelligence and diagnostic purposes.
- Publish facts about the state of your application which interested parties may observe and use as a trigger for taking action.
- Observe interesting operations and interactions happening within your business or other ecosystem, allowing loosely coupled systems to interact without the need to bind them together.

- Receive events from one or more publishers, transform them to better meet the needs of your ecosystem, then publish the transformed events to a new stream for consumers to observe. [Source code]

(<https://github.com/Azure/azure-sdk-for-python/blob/main/sdk/eventhub/azure-eventhub/>) | [Package (PyPi)] (<https://pypi.org/project/azure-eventhub/>) | [Package (Conda)] (<https://anaconda.org/microsoft/azure-eventhub/>) | [API reference documentation][api_reference] | [Product documentation] (<https://docs.microsoft.com/azure/event-hubs/>) | [Samples] (<https://github.com/Azure/azure-sdk-for-python/tree/main/sdk/eventhub/azure-eventhub/samples>)

Getting started

Prerequisites - Python 3.8 or later. - **Microsoft Azure Subscription:** To use Azure services, including Azure Event Hubs, you'll need a subscription. If you do not have an existing Azure account, you may sign up for a free trial or use your MSDN subscriber benefits when you [create an account](<https://azure.microsoft.com/free/>). - **Event Hubs namespace with an Event Hub:** To interact with Azure Event Hubs, you'll also need to have a namespace and Event Hub available. If you are not familiar with creating Azure resources, you may wish to follow the step-by-step guide for [creating an Event Hub using the Azure portal] (<https://docs.microsoft.com/azure/event-hubs/event-hubs-create>). There, you can also find detailed instructions for using the Azure CLI, Azure PowerShell, or Azure Resource Manager (ARM) templates to create an Event Hub.

Install the package

Install the Azure Event Hubs client library for Python with pip: `$ pip install azure-eventhub`

Authenticate the client

Interaction with Event Hubs starts with an instance of `EventHubConsumerClient` or `EventHubProducerClient` class. You need either the host name, SAS/AAD credential and event hub name or a connection string to instantiate the client object. **[Create client from connection string:]** (https://github.com/Azure/azure-sdk-for-python/blob/main/sdk/eventhub/azure-eventhub/samples/sync_samples/connection_string_authentication.py)

For the Event Hubs client library to interact with an Event Hub, the easiest means is to use a connection string, which is created automatically when creating an Event Hubs namespace. If you aren't familiar with shared access policies in Azure, you may wish to follow the step-by-step guide to [get an Event Hubs connection string](<https://docs.microsoft.com/azure/event-hubs/event-hubs-get-connection-string>). - The `from_connection_string` method takes the connection string of the form

`Endpoint=sb://<yournamespace>.servicebus.windows.net/;SharedAccessKeyName=<yoursharedaccesskeyname>;SharedAccessKey=<yoursharedaccesskey>`

and entity name to your Event Hub instance. You can get the connection string from the [Azure portal](<https://docs.microsoft.com/azure/event-hubs/event-hubs-get-connection-string#get-connection-string-from-the>

portal). ****[Create client using the azure-identity library:]**

(https://github.com/Azure/azure-sdk-for-python/blob/main/sdk/eventhub/azure-eventhub/samples/sync_samples/client_identity_authentication.py)******

Alternately, one can use a Credential object to authenticate via AAD with the azure-identity package. - This constructor demonstrated in the sample linked above takes the host name and entity name of your Event Hub instance and credential that implements the [TokenCredential]

(<https://github.com/Azure/azure-sdk-for-python/blob/main/sdk/core/azure-core/azure/core/credentials.py>) protocol. There are implementations of the TokenCredential`protocol available in the [azure-identity package]

(<https://pypi.org/project/azure-identity/>). The host name is of the format <yournamespace.servicebus.windows.net>. - To use the credential types provided by `azure-identity`, please install the package: `pip install azure-identity`- Additionally, to use the async API, you must first install an async transport, such as [aiohttp](<https://pypi.org/project/aiohttp/>): `pip install aiohttp`- When using Azure Active Directory, your principal must be assigned a role which allows access to Event Hubs, such as the Azure Event Hubs Data Owner role. For more information about using Azure Active Directory authorization with Event Hubs, please refer to [the associated documentation](<https://docs.microsoft.com/azure/event-hubs/authorize-access-azure-active-directory>). **## Key concepts - An**

****EventHubProducerClient**** is a source of telemetry data, diagnostics information, usage logs, or other log data, as part of an embedded device solution, a mobile device application, a game title running on a console or other device, some client or server based business solution, or a web site. - An ****EventHubConsumerClient**** picks up such information from the Event Hub and processes it. Processing may involve aggregation, complex computation, and filtering. Processing may also involve distribution or storage of the information in a raw or transformed fashion. Event Hub consumers are often robust and high-scale platform infrastructure parts with built-in analytics capabilities, like Azure Stream Analytics, Apache Spark, or Apache Storm. - A ****partition**** is an ordered sequence of events that is held in an Event Hub. Azure Event Hubs provides message streaming through a partitioned consumer pattern in which each consumer only reads a specific subset, or partition, of the message stream. As newer events arrive, they are added to the end of this sequence. The number of partitions is specified at the time an Event Hub is created and cannot be changed. - A ****consumer group**** is a view of an entire Event Hub. Consumer groups enable multiple consuming applications to each have a separate view of the event stream, and to read the stream independently at their own pace and from their own position. There can be at most 5 concurrent readers on a partition per consumer group; however it is recommended that there is only one active consumer for a given partition and consumer group pairing. Each active

reader receives all of the events from its partition; if there are multiple readers on the same partition, then they will receive duplicate events. For more concepts and deeper discussion, see: [Event Hubs Features] (<https://docs.microsoft.com/azure/event-hubs/event-hubs-features>). Also, the concepts for AMQP are well documented in [OASIS Advanced Messaging Queuing Protocol (AMQP) Version 1.0] (<https://docs.oasis-open.org/amqp/core/v1.0/os/amqp-core-overview-v1.0-os.html>).

Thread safety

We do not guarantee that the `EventHubProducerClient` or `EventHubConsumerClient` are thread-safe. We do not recommend reusing these instances across threads. It is up to the running application to use these classes in a thread-safe manner. The data model type, `EventDataBatch` is not thread-safe. It should not be shared across threads nor used concurrently with client methods.

Examples

The following sections provide several code snippets covering some of the most common Event Hubs tasks, including:

- [Inspect an Event Hub](#inspect-an-event-hub)
- [Publish events to an Event Hub](#publish-events-to-an-event-hub)
- [Consume events from an Event Hub](#consume-events-from-an-event-hub)
- [Consume events from an Event Hub in batches](#consume-events-from-an-event-hub-in-batches)
- [Publish events to an Event Hub asynchronously](#publish-events-to-an-event-hub-asynchronously)
- [Consume events from an Event Hub asynchronously](#consume-events-from-an-event-hub-asynchronously)
- [Consume events from an Event Hub in batches asynchronously](#consume-events-from-an-event-hub-in-batches-asynchronously)
- [Consume events and save checkpoints using a checkpoint store](#consume-events-and-save-checkpoints-using-a-checkpoint-store)
- [Use `EventHubConsumerClient` to work with IoT Hub](#use-eventhubconsumerclient-to-work-with-iot-hub)

Inspect an Event Hub

Get the partition ids of an Event Hub.

```
python import os from
azure.eventhub import EventHubConsumerClient from azure.identity import
DefaultAzureCredential FULLY_QUALIFIED_NAMESPACE =
os.environ["EVENT_HUB_HOSTNAME"] EVENTHUB_NAME =
os.environ['EVENT_HUB_NAME'] consumer_client = EventHubConsumerClient(
fully_qualified_namespace=FULLY_QUALIFIED_NAMESPACE,
consumer_group='$Default', eventhub_name=EVENTHUB_NAME,
credential=DefaultAzureCredential(), ) with consumer_client: pass #
consumer_client is now ready to be used.
```

Publish events to an Event Hub

Use the `create_batch` method on `EventHubProducerClient` to create an `EventDataBatch` object which can then be sent using the `send_batch` method. Events may be added to the `EventDataBatch` using the `add` method until the maximum batch size limit in bytes has been reached.

```
<!--
SNIPPET:send.send_event_data_batch --> python def
send_event_data_batch(producer): # Without specifying partition_id or
partition_key # the events will be distributed to available partitions via round-
robin. event_data_batch = producer.create_batch()
```

```

event_data_batch.add(EventData("Single message"))
producer.send_batch(event_data_batch) `<!-- END SNIPPET --> ### Consume
events from an Event Hub There are multiple ways to consume events from
an EventHub. To simply trigger a callback when an event is received, the
EventHubConsumerClient.receive`method will be of use as follows: python
import logging from azure.eventhub import EventHubConsumerClient from
azure.identity import DefaultAzureCredential fully_qualified_namespace = '<<
EVENT HUBS FULLY QUALIFIED NAMESPACE >>' consumer_group = '<<
CONSUMER GROUP >>' eventhub_name = '<< NAME OF THE EVENT HUB >>'
client = EventHubConsumerClient(
fully_qualified_namespace=fully_qualified_namespace,
eventhub_name=eventhub_name, consumer_group=consumer_group,
credential=DefaultAzureCredential(), ) logger =
logging.getLogger("azure.eventhub") logging.basicConfig(level=logging.INFO)
def on_event(partition_context, event): logger.info("Received event from
partition {}".format(partition_context.partition_id))
partition_context.update_checkpoint(event) with client: client.receive(
on_event=on_event, starting_position="-1", # "-1" is from the beginning of the
partition. ) # receive events from specified partition: #
client.receive(on_event=on_event, partition_id='0') `### Consume events
from an Event Hub in batches Whereas the above sample triggers the
callback for each message as it is received, the following sample triggers
the callback on a batch of events, attempting to receive a number at a time.
python import logging from azure.eventhub import EventHubConsumerClient
from azure.identity import DefaultAzureCredential fully_qualified_namespace
= '<< EVENT HUBS FULLY QUALIFIED NAMESPACE >>' consumer_group = '<<
CONSUMER GROUP >>' eventhub_name = '<< NAME OF THE EVENT HUB >>'
client = EventHubConsumerClient(
fully_qualified_namespace=fully_qualified_namespace,
eventhub_name=eventhub_name, consumer_group=consumer_group,
credential=DefaultAzureCredential(), ) logger =
logging.getLogger("azure.eventhub") logging.basicConfig(level=logging.INFO)
def on_event_batch(partition_context, events): logger.info("Received event
from partition {}".format(partition_context.partition_id))
partition_context.update_checkpoint() with client: client.receive_batch(
on_event_batch=on_event_batch, starting_position="-1", # "-1" is from the
beginning of the partition. ) # receive events from specified partition: #
client.receive_batch(on_event_batch=on_event_batch, partition_id='0') `###
Publish events to an Event Hub asynchronously Use the `create_batch`
method on EventHubProducer`to create an EventDataBatch`object which can
then be sent using the `send_batch`method. Events may be added to the
EventDataBatch`using the `add`method until the maximum batch size limit in
bytes has been reached. python import asyncio from azure.eventhub.ai
import EventHubProducerClient # The package name suffixed with ".ai" for

```

```

async from azure.eventhub import EventData from azure.identity.aio import
DefaultAzureCredential fully_qualified_namespace = '<< EVENT HUBS FULLY
QUALIFIED NAMESPACE >>' eventhub_name = '<< NAME OF THE EVENT HUB
>>' async def create_batch(client): event_data_batch = await
client.create_batch() can_add = True while can_add: try:
event_data_batch.add(EventData('Message inside EventBatchData')) except
ValueError: can_add = False # EventDataBatch object reaches max_size.
return event_data_batch async def send(): client = EventHubProducerClient(
fully_qualified_namespace=fully_qualified_namespace,
eventhub_name=eventhub_name, credential=DefaultAzureCredential(), )
batch_data = await create_batch(client) async with client: await
client.send_batch(batch_data) if __name__ == '__main__': loop =
asyncio.get_event_loop() loop.run_until_complete(send()) `### Consume
events from an Event Hub asynchronously This SDK supports both
synchronous and asyncio based code. To receive as demonstrated in the
samples above, but within aio, one would need the following: python import
logging import asyncio from azure.eventhub.aio import
EventHubConsumerClient from azure.identity.aio import
DefaultAzureCredential fully_qualified_namespace = '<< EVENT HUBS FULLY
QUALIFIED NAMESPACE >>' consumer_group = '<< CONSUMER GROUP >>'
eventhub_name = '<< NAME OF THE EVENT HUB >>' logger =
logging.getLogger("azure.eventhub") logging.basicConfig(level=logging.INFO)
async def on_event(partition_context, event): logger.info("Received event
from partition {}".format(partition_context.partition_id)) await
partition_context.update_checkpoint(event) async def receive(): client =
EventHubConsumerClient(
fully_qualified_namespace=fully_qualified_namespace,
eventhub_name=eventhub_name, consumer_group=consumer_group,
credential=DefaultAzureCredential(), ) async with client: await client.receive(
on_event=on_event, starting_position="-1", # "-1" is from the beginning of the
partition. ) # receive events from specified partition: # await
client.receive(on_event=on_event, partition_id='0') if __name__ == '__main__':
loop = asyncio.get_event_loop() loop.run_until_complete(receive()) `###
Consume events from an Event Hub in batches asynchronously All
synchronous functions are supported in aio as well. As demonstrated above
for synchronous batch receipt, one can accomplish the same within asyncio
as follows: python import logging import asyncio from azure.eventhub.aio
import EventHubConsumerClient from azure.identity.aio import
DefaultAzureCredential fully_qualified_namespace = '<< EVENT HUBS FULLY
QUALIFIED NAMESPACE >>' consumer_group = '<< CONSUMER GROUP >>'
eventhub_name = '<< NAME OF THE EVENT HUB >>' logger =
logging.getLogger("azure.eventhub") logging.basicConfig(level=logging.INFO)
async def on_event_batch(partition_context, events): logger.info("Received
event from partition {}".format(partition_context.partition_id)) await

```

```

partition_context.update_checkpoint() async def receive_batch(): client =
EventHubConsumerClient(
fully_qualified_namespace=fully_qualified_namespace,
eventhub_name=eventhub_name, consumer_group=consumer_group,
credential=DefaultAzureCredential(), ) async with client: await
client.receive_batch( on_event_batch=on_event_batch, starting_position="-1",
# "-1" is from the beginning of the partition. ) # receive events from specified
partition: # await client.receive_batch(on_event_batch=on_event_batch,
partition_id='0') if __name__ == '__main__': loop = asyncio.get_event_loop()
loop.run_until_complete(receive_batch()) `### Consume events and save
checkpoints using a checkpoint store EventHubConsumerClient`is a high
level construct which allows you to receive events from multiple partitions at
once and load balance with other consumers using the same Event Hub and
consumer group. This also allows the user to track progress when events are
processed using checkpoints. A checkpoint is meant to represent the last
successfully processed event by the user from a particular partition of a
consumer group in an Event Hub instance. The EventHubConsumerClient`
uses an instance of CheckpointStore`to update checkpoints and to store the
relevant information required by the load balancing algorithm. Search pypi
with the prefix `azure-eventhub-checkpointstore`to find packages that
support this and use the CheckpointStore`implementation from one such
package. Please note that both sync and async libraries are provided. In the
below example, we create an instance of EventHubConsumerClient`and use
a BlobCheckpointStore`. You need to [create an Azure Storage account]
(https://docs.microsoft.com/azure/storage/common/storage-quickstart-create-account?tabs=azure-portal) and a [Blob Container]
(https://docs.microsoft.com/azure/storage/blobs/storage-quickstart-blobs-portal#create-a-container) to run the code. [Azure Blob Storage Checkpoint
Store Async](https://github.com/Azure/azure-sdk-for-
python/blob/main/sdk/eventhub/azure-eventhub-checkpointstoreblob-ai)
and [Azure Blob Storage Checkpoint Store Sync]
(https://github.com/Azure/azure-sdk-for-
python/blob/main/sdk/eventhub/azure-eventhub-checkpointstoreblob) are
one of the CheckpointStore`implementations we provide that applies Azure
Blob Storage as the persistent store.
python import asyncio from
azure.eventhub.ai import EventHubConsumerClient from
azure.eventhub.extensions.checkpointstoreblobai import
BlobCheckpointStore from azure.identity.ai import DefaultAzureCredential
fully_qualified_namespace = '<< EVENT HUBS FULLY QUALIFIED NAMESPACE
>>' consumer_group = '<< CONSUMER GROUP >>' eventhub_name = '<<
NAME OF THE EVENT HUB >>' blob_account_url = '<< STORAGE ACCOUNT
URL >>' container_name = '<<NAME OF THE BLOB CONTAINER>>' async def
on_event(partition_context, event): # do something await
partition_context.update_checkpoint(event) # Or update_checkpoint every N

```

```

events for better performance.
async def receive(client):
    await client.receive(
        on_event=on_event,
        starting_position="-1", # "-1" is from the beginning of the
        partition. )
    async def main():
        checkpoint_store = BlobCheckpointStore(
            blob_account_url=blob_account_url,
            container_name=container_name,
            credential=DefaultAzureCredential() )
        client = EventHubConsumerClient(
            fully_qualified_namespace=fully_qualified_namespace,
            eventhub_name=eventhub_name,
            credential=DefaultAzureCredential(),
            consumer_group=consumer_group,
            checkpoint_store=checkpoint_store, #
            # For load balancing and checkpoint. Leave None for no load balancing )
        async with client:
            await receive(client)
    if __name__ == '__main__':
        loop = asyncio.get_event_loop()
        loop.run_until_complete(main())
`### Use EventHubConsumerClient to work with IoT Hub
You can use EventHubConsumerClient to work with IoT Hub as well. This is useful for
receiving telemetry data of IoT Hub from the linked EventHub. The associated
connection string will not have send claims, hence sending events is not
possible. Please notice that the connection string needs to be for an [Event
Hub-compatible endpoint](https://docs.microsoft.com/azure/iot-hub/iot-hub-devguide-messages-read-builtin),
e.g. "Endpoint=sb://my-iothub-namespace-[uid].servicebus.windows.net/;SharedAccessKeyName=my-SA-
name;SharedAccessKey=my-SA-key;EntityPath=my-iot-hub-name"
There are two ways to get the Event Hubs compatible endpoint:
- Manually get the "Built-in endpoints" of the IoT Hub in Azure Portal and receive from it.
python
from azure.eventhub import EventHubConsumerClient
connection_str = 'Endpoint=sb://my-iothub-namespace-[uid].servicebus.windows.net/;SharedAccessKeyName=my-SA-
name;SharedAccessKey=my-SA-key;EntityPath=my-iot-hub-name'
consumer_group = '<< CONSUMER GROUP >>'
client = EventHubConsumerClient.from_connection_string(connection_str,
consumer_group)
partition_ids = client.get_partition_ids()
`- Programmatically retrieve the built-in Event Hubs compatible endpoint. Refer to [IoT Hub
Connection String Sample](https://github.com/Azure/azure-sdk-for-python/blob/main/sdk/eventhub/azure-eventhub/samples/async_samples/iot_hub_connection_string_receive_async.py).
## Troubleshooting
See the [azure-eventhub troubleshooting guide](https://github.com/Azure/azure-sdk-for-python/blob/main/sdk/eventhub/azure-eventhub/TROUBLESHOOTING.md)
for details on how to diagnose various failure scenarios.
### Logging
- Enable [azure.eventhub.logger](https://github.com/Azure/azure-sdk-for-python/blob/main/sdk/eventhub/azure-eventhub/TROUBLESHOOTING.md) to collect traces from the library.
- Enable AMQP frame level trace by setting logging_enable=True when creating the client.
- Refer to [this guide](https://learn.microsoft.com/azure/developer/python/sdk/azure-sdk-logging) on configuring logging for Azure libraries for Python for additional information.
python
import logging
import sys
handler = logging.StreamHandler(stream=sys.stdout)
log_fmt =

```

```

logging.Formatter(fmt="%asctime)s | %(threadName)s | %(levelname)s | %(
(name)s | %(message)s") handler.setFormatter(log_fmt) logger =
logging.getLogger('azure.eventhub') logger.setLevel(logging.DEBUG)
logger.addHandler(handler) ... from azure.eventhub import
EventHubProducerClient, EventHubConsumerClient producer =
EventHubProducerClient(..., logging_enable=True) consumer =
EventHubConsumerClient(..., logging_enable=True) `### Next steps ### More
sample code Please take a look at the [samples]
(https://github.com/Azure/azure-sdk-for-
python/blob/main/sdk/eventhub/azure-eventhub/samples) directory for
detailed examples of how to use this library to send and receive events
to/from Event Hubs. ### Documentation Reference documentation is
available [here]
(https://azuresdkdocs.blob.core.windows.net/\$web/python/azure-
eventhub/latest/azure.eventhub.html). ### Schema Registry and Avro
Encoder The EventHubs SDK integrates nicely with the [Schema Registry]
[schemaregistry_service] service and [Avro][avro]. For more information,
please refer to [Schema Registry SDK][schemaregistry_repo] and [Schema
Registry Avro Encoder SDK][schemaregistry_avroencoder_repo]. ### Pure
Python AMQP Transport and Backward Compatibility Support The Azure
Event Hubs client library is now based on a pure Python AMQP
implementation. `uAMQP` has been removed as required dependency. To use
`uAMQP` as the underlying transport: 1. Install `uamqp` with pip. `$ pip install
uamqp` 2. Pass `uamqp_transport=True` during client construction.
python
from azure.eventhub import EventHubProducerClient,
EventHubConsumerClient from azure.identity import DefaultAzureCredential
fully_qualified_namespace = '<< EVENT HUBS FULLY QUALIFIED NAMESPACE
>>' consumer_group = '<< CONSUMER GROUP >>' eventhub_name = '<<
NAME OF THE EVENT HUB >>' client = EventHubProducerClient(
fully_qualified_namespace=fully_qualified_namespace,
eventhub_name=eventhub_name, credential=DefaultAzureCredential(),
uamqp_transport=True ) client = EventHubConsumerClient(
fully_qualified_namespace=fully_qualified_namespace,
eventhub_name=eventhub_name, credential=DefaultAzureCredential(),
consumer_group=consumer_group, uamqp_transport=True ) `Note: The
`message` attribute on EventData/EventDataBatch, which previously exposed
the `uamqp.Message`, has been deprecated. The "Legacy" objects returned
by EventData.message/EventDataBatch.message have been introduced to
help facilitate the transition. ### Building uAMQP wheel from source If
[uAMQP](https://pypi.org/project/uamqp/) is intended to be used as the
underlying AMQP protocol implementation for `azure-eventhub`, uAMQP
wheels can be found for most major operating systems. If you intend to use
`uAMQP` and you're running on a platform for which uAMQP wheels are not
provided, please follow the [uAMQP Installation]

```

(<https://github.com/Azure/azure-uamqp-python#installation>) guidance to install from source. ### Provide Feedback If you encounter any bugs or have suggestions, please file an issue in the [Issues] (<https://github.com/Azure/azure-sdk-for-python/issues>) section of the project. ## Contributing This project welcomes contributions and suggestions. Most contributions require you to agree to a Contributor License Agreement (CLA) declaring that you have the right to, and actually do, grant us the rights to use your contribution. For details, visit <https://cla.microsoft.com>. When you submit a pull request, a CLA-bot will automatically determine whether you need to provide a CLA and decorate the PR appropriately (e.g., label, comment). Simply follow the instructions provided by the bot. You will only need to do this once across all repos using our CLA. This project has adopted the [Microsoft Open Source Code of Conduct](<https://opensource.microsoft.com/codeofconduct/>). For more information see the [Code of Conduct FAQ] (<https://opensource.microsoft.com/codeofconduct/faq/>) or contact [opencode@microsoft.com](mailto:opencode@microsoft.com) with any additional questions or comments. [avro]: <https://avro.apache.org/> [api_reference]: [file:///Users/silambarasanr/Downloads/project-attribution-custom+8617_splunk-add-on-for-microsoft-cloud-services\\$c9aa2ba077bea93143d69f8281f748ab6b...](https://docs.microsoft.com/python/api/overview/azure/eventhub-readme[schemaregistry_service]: https://aka.ms/schemaregistry[schemaregistry_repo]: https://github.com/Azure/azure-sdk-for-python/tree/main/sdk/schemaregistry/azure-schemaregistry[schemaregistry_avroencoder_repo]: https://github.com/Azure/azure-sdk-for-python/tree/main/sdk/schemaregistry/azure-schemaregistry-avroencoder ! [Impressions](https://azure-sdk-impressions.azurewebsites.net/api/impressions/azure-sdk-for-python/sdk/eventhub/azure-eventhub/README.png) # Release History ## 5.13.0 (2024-11-12) ### Features Added - Added `ssl_context` parameter to the clients to allow users to pass in the SSL context, in which case, `connection_verify` will be ignored if specified. ### Other Changes - Added debug logging to track received messages. ## 5.12.2 (2024-10-02) ### Bugs Fixed - Implemented backpressure for async consumer to address a memory leak issue. ([#36398](https://github.com/Azure/azure-sdk-for-python/issues/36398)) ## 5.12.1 (2024-06-11) ### Bugs Fixed - Fixed a bug where the correct URI was not being used for consumer authentication, causing issues when assigning roles at the consumer group level. ([#35337](https://github.com/Azure/azure-sdk-for-python/issues/35337)) ## 5.12.0 (2024-05-16) ### Features Added - Added support for non-tls connections ([#34272]) ## 5.11.7 (2024-04-10) ### Bugs Fixed - Fixed a bug where using EventHubProducerClient in buffered mode could potentially drop a buffered message without actually sending it. ([#34712](https://github.com/Azure/azure-sdk-for-python/pull/34712)) ## Other Changes - Updated network trace logging to replace None values in AMQP</p>
</div>
<div data-bbox=)

connection info with empty strings as per the OpenTelemetry specification.

5.11.6 (2024-02-12) This version and all future versions will require Python 3.8+. Python 3.7 is no longer supported. ### Features Added - Added `keep_alive` functionality on `EventHubProducerClient` to allow for long-living producers. [#33726](https://github.com/Azure/azure-sdk-for-python/issues/33726) ### Other Changes - Added support for Python 3.12.

5.11.5 (2023-11-13) ### Bugs Fixed - Fixed a `pyAMQP` error where events split across multiple `TransferFrames` were depleting the link credit by more than 1 credit per message. ([#32767](https://github.com/Azure/azure-sdk-for-python/pull/32767))

5.11.4 (2023-08-08) ### Features Added - A new float keyword argument `socket_timeout` has been added to the sync and async `EventHubConsumerClient` and `EventHubProducerClient`. ### Bugs Fixed - Fixed bug [#31258](https://github.com/Azure/azure-sdk-for-python/issues/31258) where async `BlobCheckpointStore` was reprocessing old events after an error.

5.11.3 (2023-07-12) ### Bugs Fixed - Fixed the error `end frame received on invalid channel` which was raised when a disconnect was sent by the service ([#30860](https://github.com/Azure/azure-sdk-for-python/pull/30860)) - Fixed the error `link already closed` which was raised when the client was closing and disconnecting from the service ([#30836](https://github.com/Azure/azure-sdk-for-python/pull/30836))

Other Changes - Updated tracing ([#29934](https://github.com/Azure/azure-sdk-for-python/pull/29934)): - Span names renamed: - `Azure.EventHubs.send` to `EventHubs.send` - `Azure.EventHubs.message` to `EventHubs.message` - `Azure.EventHubs.process` to `EventHubs.process` - An `EventHubs.receive` span will be created upon receiving events. - Additional attributes added to spans: - `messaging.system` - messaging system (i.e., `eventhubs`) - `messaging.operation` - type of operation (i.e., `publish`, `receive`, or `process`) - `messaging.batch.message_count` - number of messages sent, received, or processed (if more than one) - The `component` attribute was removed from all spans. - All `send` spans now contain links to `message` spans. Now, `message` spans will no longer contain a link to the `send` span. - Message application properties will now contain values for `traceparent` (and `tracestate` if applicable) - Process spans will now be a direct children of message span contexts in when event handling on a per-message basis. ([#30537](https://github.com/Azure/azure-sdk-for-python/pull/30537))

5.11.2 (2023-03-20) ### Bugs Fixed - Fixed a bug that would prevent reconnect after a long idle period, network drop (issue #28996)

5.11.1 (2023-01-25) ### Bugs Fixed - Fixed a bug where, when `websocket-client` was not installed, the error was not caught/raised properly (issue #28453).

5.11.0 (2023-01-19) Version 5.11.0 is our first stable release of the Azure Event Hubs client library based on a pure Python implemented AMQP stack. ### Features Added - A new boolean keyword argument `uamqp_transport` has been added to sync and async

EventHubProducerClient/EventHubConsumerClient constructors which indicates whether to use the `uamqp` library or the default pure Python AMQP library as the underlying transport.

Bugs Fixed - Fixed a bug that caused an error when sending batches with tracing enabled (issue #27986). - Fixed a bug where `EventHubSharedKeyCredential` returned an `AccessToken.token` of type `bytes` and not `str`, now matching the documentation.

Other Changes - The `message` attribute on `EventData/EventDataBatch`, which previously exposed the `uamqp.Message`, has been deprecated. - `LegacyMessage/LegacyBatchMessage` objects returned by the `message` attribute on `EventData/EventDataBatch` have been introduced to help facilitate the transition. - Removed `uAMQP` from required dependencies. - Adding `uamqp >= 1.6.3` as an optional dependency for use with the `uamqp_transport` keyword. - Added support for Python 3.11.

5.8.0b2 (2022-10-11)
Features Added - Updated the optional dependency for async transport using AMQP over WebSocket from `websocket-client` to `aiortc` (Issue #24315, thanks @hansmbakker for the suggestion).

5.8.0b1 (2022-09-22) This version and all future versions will require Python 3.7+. Python 3.6 is no longer supported.

Other Changes - Added the `uamqp_transport` optional parameter to the clients, to allow switching to the `uamqp` library as the transport.

5.10.1 (2022-08-22) This version and all future versions will require Python 3.7+, Python 3.6 is no longer supported.

Bugs Fixed - Fixed a bug in `async BufferedProducer` that would block when flushing the queue causing the client to freeze up (issue #23510). - Fixed a bug in the `async EventHubProducerClient` and `EventHubConsumerClient` that set the default value of the `transport_type` parameter in the `from_connection_string` methods to `None` rather than `TransportType.Amqp`.

Other Changes - Internal refactoring to support upcoming Pure Python AMQP-based release. - Updated `uAMQP` dependency to 1.6.0.

5.8.0a5 (2022-07-19)
Bugs Fixed - Fixed bug that prevented token refresh at regular intervals. - Fixed bug that was improperly passing the `debug` keyword argument, so that network trace debug logs are output when requested.

Other Changes - Added logging added in to track proper token refreshes & fetches, output exception reason for producer init failure.

5.10.0 (2022-06-08)
Features Added - Includes the following features related to buffered sending of events:

- A new method `send_event` to `EventHubProducerClient` which allows sending single `EventData` or `AmqpAnnotatedMessage`.
- Buffered mode sending to `EventHubProducerClient` which is intended to allow for efficient publishing of events without having to explicitly manage batches in the application.
- The constructor of `EventHubProducerClient` and `from_connection_string` method takes the following new keyword arguments for configuration:
 - `buffered_mode`: The flag to enable/disable buffered mode sending.
 - `on_success`: The callback to be called once events have been successfully published.
 - `on_error`: The callback to be called once events have failed to be

published. - `max_buffer_length`: The total number of events per partition that can be buffered before a flush will be triggered. - `max_wait_time`: The amount of time to wait for a batch to be built with events in the buffer before publishing. - A new method `EventHubProducerClient.flush` which flushes events in the buffer to be sent immediately. - A new method `EventHubProducerClient.get_buffered_event_count` which returns the number of events that are buffered and waiting to be published for a given partition. - A new property `EventHubProducerClient.total_buffered_event_count` which returns the total number of events that are currently buffered and waiting to be published, across all partitions. - A new boolean keyword argument `flush` to `EventHubProducerClient.close` which indicates whether to flush the buffer or not while closing. ## 5.8.0a4 (2022-06-07) ### Features Added - Added support for connection using websocket and http proxy. - Added support for custom endpoint connection over websocket. ## 5.9.0 (2022-05-10) ### Features Added - The classmethod `from_message_content` has been added to `EventData` for interoperability with the Schema Registry Avro Encoder library, and takes `content` and `content_type` as positional parameters. ### Other Changes - Features related to buffered sending of events are still in beta and will not be included in this release. ## 5.9.0b3 (2022-04-20) ### Features Added - Introduced new method `send_event` to `EventHubProducerClient` which allows sending single `EventData` or `AmqpAnnotatedMessage`. - Introduced buffered mode sending to `EventHubProducerClient` which is intended to allow for efficient publishing of events without having to explicitly manage batches in the application. - The constructor of `EventHubProducerClient` and `from_connection_string` method now takes the following new keyword arguments for configuration: - `buffered_mode`: The flag to enable/disable buffered mode sending. - `on_success`: The callback to be called once events have been successfully published. - `on_error`: The callback to be called once events have failed to be published. - `max_buffer_length`: The total number of events per partition that can be buffered before a flush will be triggered. - `max_wait_time`: The amount of time to wait for a batch to be built with events in the buffer before publishing. - Introduced new method `EventHubProducerClient.flush` which flushes events in the buffer to be sent immediately. - Introduced new method `EventHubProducerClient.get_buffered_event_count` which returns the number of events that are buffered and waiting to be published for a given partition. - Introduced new property `EventHubProducerClient.total_buffered_event_count` which returns the total number of events that are currently buffered and waiting to be published, across all partitions. - Introduced new boolean keyword argument `flush` to `EventHubProducerClient.close` which indicates whether to flush the buffer or not while closing. ### Other Changes - Updated `EventData` internals for interoperability with the Schema Registry Avro Encoder library. ## 5.9.0b2 (2022-03-09) ### Breaking Changes - `from_message_data` on `EventData` has been renamed `from_message_content`

for interoperability with the Schema Registry Avro Encoder library. The `data` parameter has been renamed to `content`. ## 5.8.0a3 (2022-03-08) ### Other Changes - Improved the performance of async sending and receiving. ## 5.9.0b1 (2022-02-09) - The following features have been temporarily pulled out of `async EventHubProducerClient` and `EventHubConsumerClient` which will be added back in future previews as we work towards a stable release: - Passing the following keyword arguments to the constructors and `from_connection_string` methods of the `EventHubProducerClient` and `EventHubConsumerClient` is not supported: `transport_type`, `http_proxy`, `custom_endpoint_address`, and `connection_verify`. ## 5.8.0a2 (2022-02-09) ### Features Added - Added support for `async EventHubProducerClient` and `EventHubConsumerClient`. ## 5.8.0a1 (2022-01-13) Version 5.8.0a1 is our first efforts to build an Azure Event Hubs client library based on pure python implemented AMQP stack. ### Breaking changes - The following features have been temporarily pulled out which will be added back in future previews as we work towards a stable release: - Async is not supported. - Passing the following keyword arguments to the constructors and `from_connection_string` methods of the `EventHubProducerClient` and `EventHubConsumerClient` is not supported: `transport_type`, `http_proxy`, `custom_endpoint_address`, and `connection_verify`. ### Other Changes - uAMQP dependency is removed. ## 5.7.0 (2022-01-12) This version and all future versions will require Python 3.6+. Python 2.7 is no longer supported. ### Features Added - Added support for fixed (linear) retry backoff: - `Sync/async EventHubProducerClient` and `EventHubConsumerClient` constructors and `from_connection_string` take `retry_mode` as a keyword argument. ### Bugs Fixed - Fixed a bug that `EventHubProducerClient` could be reopened for sending events instead of encountering with `KeyError` when the client is previously closed (issue #21849). ### Other Changes - Improved token refresh timing to prevent potentially blocking main flow when the token is about to get expired soon. - Updated uAMQP dependency to 1.5.1. ## 5.6.1 (2021-10-06) ### Bugs Fixed - Fixed a bug for checking that `azure.eventhub.amqp.AmqpMessageHeader` and `azure.eventhub.amqp.AmqpMessageProperties` contain specific properties using the `in` keyword. ### Other Changes - Updated uAMQP dependency to 1.4.3. - Added support for Python 3.10. - Fixed memory leak in win32 socketio and tlsio (issue #19777). - Fixed memory leak in the process of converting `AMQPValue` into string (issue #19777). ## 5.6.0 (2021-07-07) ### Features Added - Added support for sending AMQP annotated message which allows full access to the AMQP message fields. - Introduced new namespace `azure.eventhub.amqp`. - Added new enum class `azure.eventhub.amqp.AmqpMessageBodyType` to represent the body type of the message which includes: - `DATA`: The body of message consists of one or more data sections and each section contains opaque binary data. - `SEQUENCE`: The body of message consists of one or more sequence

sections and each section contains an arbitrary number of structured data elements. - VALUE: The body of message consists of one amqp-value section and the section contains a single AMQP value. - Introduced new class `azure.eventhub.amqp.AmqpAnnotatedMessage` for accessing low-level amqp message sections which can be instantiated for sending. - Introduced new classes `azure.eventhub.amqp.AmqpMessageHeader` and `azure.eventhub.amqp.AmqpMessageProperties` for accessing amqp header and properties. - Added new property `body_type` on `azure.eventhub.EventData` which returns `azure.eventhub.amqp.AmqpMessageBodyType`. - Added new read-only property `raw_amqp_message` on `azure.eventhub.EventData` which returns `azure.eventhub.amqp.AmqpAnnotatedMessage`. ### Fixed - Updated uAMQP dependency to 1.4.1. - Fixed a bug that attributes `creation_time`, `absolute_expiry_time` and `group_sequence` on `MessageProperties` should be compatible with integer types on Python 2.7. ## 5.5.0 (2021-05-13) **New Features** - Added support for using `azure.core.credentials.AzureNamedKeyCredential` as credential for authenticating producer and consumer clients. **Bug Fixes** - Fixed bug that custom user agent string should be put in front of the built-in user agent string instead of being appended. - Updated uAMQP dependency to 1.4.0. - Fixed memory leaks in the process of link attach where source and target cython objects are not properly deallocated (#15747). - Improved management operation callback not to parse description value of non `AMQP_TYPE_STRING` type as string (#18361). **Notes** - Updated azure-core dependency to 1.14.0. ## 5.4.0 (2021-04-07) This version follows from version 5.3.1, rather than 5.4.0b1 so that the preview idempotent producer feature is not included. **New Features** - Added support for using `azure.core.credentials.AzureSasCredential` as credential for authenticating producer and consumer clients. - Updated `list_ownership`, `claim_ownership`, `update_checkpoint`, `list_checkpoints` on sync and async `CheckpointStore` to support taking `**kwargs`. - WARNING: Implementing a custom checkpointstore that does not support taking `**kwargs` in the methods listed previously will result in the following pylint error: W0221: Parameters differ from overridden _____ method (arguments-differ). - Updated `update_checkpoint` on sync and async `PartitionContext` to support taking `**kwargs`. **Bug Fixes** - Updated uAMQP dependency to 1.3.0. - Fixed bug that sending message of large size triggering segmentation fault when the underlying socket connection is lost (#13739, #14543). - Fixed bug in link flow control where link credit and delivery count should be calculated based on per message instead of per transfer frame (#16934). **Notes** - Updated azure-core dependency to 1.13.0. ## 5.4.0b1 (2021-03-09) This version and all future versions will require Python 2.7 or Python 3.6+, Python 3.5 is no longer supported. **New Features** - Added support for idempotent publishing which is supported by the service to endeavor to reduce the

number of duplicate events that are published. - `EventHubProducerClient` constructor accepts two new parameters for idempotent publishing: - `enable_idempotent_partitions`: A boolean value to tell the `EventHubProducerClient` whether to enable idempotency. - `partition_config`: The set of configurations that can be specified to influence publishing behavior specific to the configured Event Hub partition. - Introduced a new method `get_partition_publishing_properties` on `EventHubProducerClient` to inspect the information about the state of publishing for a partition. - Introduced a new property `published_sequence_number` on `EventData` to get the publishing sequence number assigned to the event at the time it was successfully published. - Introduced a new property `starting_published_sequence_number` on `EventDataBatch` to get the publishing sequence number assigned to the first event in the batch at the time the batch was successfully published. - Introduced a new class `azure.eventhub.PartitionPublishingConfiguration` which is a set of configurations that can be specified to influence the behavior when publishing directly to an Event Hub partition.

****Notes**** - Updated `uAMQP` dependency to 1.2.15. **## 5.3.1 (2021-03-09)** This version will be the last version to officially support Python 3.5, future versions will require Python 2.7 or Python 3.6+. ****Bug fixes**** - Sending empty `event_data_batch` will be a no-op now instead of raising error. **## 5.3.0 (2021-02-08)** ****New Features**** - Added a `parse_connection_string` method which parses a connection string into a properties bag, `EventHubConnectionStringProperties`, containing its component parts. - The constructor and `from_connection_string` method of `EventHubConsumerClient` and `EventHubProducerClient` now accept two new optional arguments: - `custom_endpoint_address` which allows for specifying a custom endpoint to use when communicating with the Event Hubs service, and is useful when your network does not allow communicating to the standard Event Hubs endpoint. - `connection_verify` which allows for specifying the path to the custom CA_BUNDLE file of the SSL certificate which is used to authenticate the identity of the connection endpoint.

****Notes**** - Updated `uAMQP` dependency to 1.2.14. **## 5.2.1 (2021-01-11)** ****Bug fixes**** - Updated `azure.eventhub.extension.__init__.py` to be compatible with `pkgutil`-style namespace (PR #13210, thanks @pjachowi). - Updated `uAMQP` dependency to 1.2.13 - Added support for Python 3.9. - Fixed bug that macOS was unable to detect network error (#15473). - Fixed bug that `uamqp.ReceiveClient` and `uamqp.ReceiveClientAsync` receive messages during connection establishment (#15555). - Fixed bug where connection establishment on macOS with Clang 12 triggering unrecognized selector exception (#15567). - Fixed bug in accessing message properties triggering segmentation fault when the underlying C bytes are NULL (#15568). **## 5.2.0 (2020-09-08)** ****New Features**** - Connection strings used with `from_connection_string` methods now supports using the `SharedAccessSignature` key in lieu of `sharedaccesskey` and

sharedaccesskeyname, taking the string of the properly constructed token as value. ## 5.2.0b1 (2020-07-06) **New Features** -

EventHubConsumerClient constructor accepts two new parameters for the load balancer. - load_balancing_strategy, which can be "greedy" or "balanced". With greedy strategy, one execution of load balancing will claim as many partitions as required to balance the load whereas with balanced strategy one execution of load balancing will claim at most 1 partition. - partition_ownership_expiration_interval, which allows you to customize the partition ownership expiration for load balancing. A consumer client may lose its owned partitions more often with a smaller expiration interval. But a larger interval may result in idle partitions not being claimed for longer time. - Added enum class azure.eventhub.LoadBalancingStrategy for load_balancing_strategy. ## 5.1.0 (2020-05-04) **New Features** -

EventHubProducerClient.send_batch accepts either an EventDataBatch or a finite list of EventData. #9181 - Added enqueueTime to span links of distributed tracing. #9599 **Bug fixes** - Fixed a bug that turned azure.eventhub.EventHubConsumerClient into an exclusive receiver when it has no checkpoint store. #11181 - Updated uAMQP dependency to 1.2.7. - Fixed bug in setting certificate of tlsio on MacOS. #7201 - Fixed bug that caused segmentation fault in network tracing on MacOS when setting logging_enable to True in EventHubConsumerClient and EventHubProducerClient. ## 5.1.0b1 (2020-04-06) **New Features** - Added EventHubConsumerClient.receive_batch() to receive and process events in batches instead of one by one. #9184 - EventHubConsumerClient.receive() has a new param max_wait_time. on_event is called every max_wait_time when no events are received and max_wait_time is not None or 0. - Param event of PartitionContext.update_checkpoint is now optional. The last received event is used when param event is not passed in. - EventData.system_properties has added missing properties when consuming messages from IoT Hub. #10408 ## 5.0.1 (2020-03-09) **Bug fixes** - Fixed a bug that swallowed errors when receiving events with azure.eventhub.EventHubConsumerClient #9660 - Fixed a bug that caused get_eventhub_properties, get_partition_ids, and get_partition_properties to raise an error on Azure Stack #9920 ## 5.0.0 (2020-01-13) **Breaking changes** - EventData - Removed deprecated property application_properties and deprecated method encode_message(). - EventHubConsumerClient - on_error would be called when EventHubConsumerClient failed to claim ownership of partitions. - on_partition_close and on_partition_initialize would be called in the case of exceptions raised by on_event callback. - EventHubConsumerClient would close and re-open the internal partition receiver in this case. - Default starting position from where EventHubConsumerClient should resume receiving after recovering from an error has been re-prioritized. - If there is checkpoint, it will resume from the checkpoint. - If there is no checkpoint but starting_position

is provided, it will resume from `starting_position`. - If there is no checkpoint or `starting_position`, it will resume from the latest position. - `PartitionContext` - `update_checkpoint` would do in-memory checkpoint instead of doing nothing when checkpoint store is not explicitly provided. - The in-memory checkpoints would be used for `EventHubConsumerClient` receiving recovering. - `get_partition_ids`, `get_partition_properties`, `get_eventhub_properties` would raise error in the case of service returning an error status code. - `AuthenticationError` would be raised when service returning error code 401. - `ConnectError` would be raised when service returning error code 404. - `EventHubError` would be raised when service returning other error codes. ## 5.0.0b6 (2019-12-03) **Breaking changes** - All exceptions should now be imported from `azure.eventhub.exceptions`. - Introduced separate `EventHubSharedKeyCredential` objects for synchronous and asynchronous operations. For async, import the credentials object from the `azure.eventhub.aio` namespace. - `EventData` - Renamed property `application_properties` to `properties`. - `EventData` no longer has attribute `last_enqueued_event_properties` - use this on `PartitionContext` instead. - `EventDataBatch` - `EventDataBatch.try_add` has been renamed to `EventDataBatch.add`. - Renamed property `size` to `size_in_bytes`. - Renamed attribute `max_size` to `max_size_in_bytes`. - `EventHubConsumerClient` and `EventHubProducerClient` - Renamed method `get_properties` to `get_eventhub_properties`. - Renamed parameters in constructor: `host` to `fully_qualified_namespace`, `event_hub_path` to `eventhub_name`. - Renamed parameters in `get_partition_properties`: `partition` to `partition_id`. - Renamed parameter `consumer_group_name` to `consumer_group` and moved that parameter from `receive` method to the constructor of `EventHubConsumerClient`. - Renamed parameter `initial_event_position` to `starting_position` on the `receive` method of `EventHubConsumerClient`. - Renamed parameter `event_hub_path` to `eventhub_name` in constructor and `from_connection_string` method of the client object. - `EventHubProducerClient.send` has been renamed to `send_batch` which will only accept `EventDataBatch` object as input. - `EventHubProducerClient.create_batch` now also takes the `partition_id` and `partition_key` as optional parameters (which are no longer specified at send). - Renamed module `PartitionManager` to `CheckpointStore`. - Receive event callback parameter has been renamed to `on_event` and now operates on a single event rather than a list of events. - Removed class `EventPostition`. - The `starting_position` parameter of the `receive` method accepts `offset(str)`, `sequence number(int)`, `datetime(datetime.datetime)` or `dict` of these types. - The `starting_position_inclusive` parameter of the `receive` method accepts `bool` or `dict` indicating whether the given event position is inclusive or not. - `PartitionContext` no longer has attribute `owner_id`. - `PartitionContext` now has attribute `last_enqueued_event_properties` which is populated if `track_last_enqueued_event_properties` is set to `True` in the `receive` method.

****New features**** - Added new parameter `idle_timeout` in `construct` and `from_connection_string` to `EventHubConsumerClient` and `EventHubProducerClient` after which the underlying connection will close if there is no further activity. **## 5.0.0b5 (2019-11-04) **Breaking changes**** - `EventHubClient`, `EventHubConsumer` and `EventHubProducer` has been removed. Use `EventHubProducerClient` and `EventHubConsumerClient` instead. - Construction of both objects is the same as it was for the previous client. - Introduced `EventHubProducerClient` as substitution for `EventHubProducer`. - `EventHubProducerClient` supports sending events to different partitions. - Introduced `EventHubConsumerClient` as substitution for `EventHubConsumer`. - `EventHubConsumerClient` supports receiving events from single/all partitions. - There are no longer methods which directly return `EventData`, all receiving is done via callback method: `on_events`. - `EventHubConsumerClient` has taken on the responsibility of `EventProcessor`. - `EventHubConsumerClient` now accepts `PartitionManager` to do load-balancing and checkpoint. - Replaced `PartitionProcessor` by four independent callback methods accepted by the `receive` method on `EventHubConsumerClient`: `on_events(partition_context, events)` called when events are received. `on_error(partition_context, exception)` called when errors occur. `on_partition_initialize(partition_context)` called when a partition consumer is opened. `on_partition_close(partition_context, reason)` called when a partition consumer is closed. - Some modules and classes that were importable from several different places have been removed: - `azure.eventhub.common` has been removed. Import from `azure.eventhub` instead. - `azure.eventhub.client_abstract` has been removed. Use `azure.eventhub.EventHubProducerClient` or `azure.eventhub.EventHubConsumerClient` instead. - `azure.eventhub.client` has been removed. Use `azure.eventhub.EventHubProducerClient` or `azure.eventhub.EventHubConsumerClient` instead. - `azure.eventhub.producer` has been removed. Use `azure.eventhub.EventHubProducerClient` instead. - `azure.eventhub.consumer` has been removed. Use `azure.eventhub.EventHubConsumerClient` instead. - `azure.eventhub.aio.client_async` has been removed. Use `azure.eventhub.aio.EventHubProducerClient` or `azure.eventhub.aio.EventHubConsumerClient` instead. - `azure.eventhub.aio.producer_async` has been removed. Use `azure.eventhub.aio.EventHubProducerClient` instead. - `azure.eventhub.aio.consumer_async` has been removed. Use `azure.eventhub.aio.EventHubConsumerClient` instead. - `azure.eventhub.aio.event_processor.event_processor` has been removed. Use `azure.eventhub.aio.EventHubConsumerClient` instead. - `azure.eventhub.aio.event_processor.partition_processor` has been removed. Use callback methods instead. - `azure.eventhub.aio.event_processor.partition_manager` has been removed.

Import from `azure.eventhub.aio` instead. -
`azure.eventhub.aio.event_processor.partition_context` has been removed.
 Import from `azure.eventhub.aio` instead. -
`azure.eventhub.aio.event_processor.sample_partition_manager` has been removed. ****Bug fixes**** - Fixed bug in user-agent string not being parsed. **## 5.0.0b4 (2019-10-08) **New features**** - Added support for tracing (issue #7153). - Added the capability of tracking last enqueued event properties of the partition to `EventHubConsumer`. - Added new boolean type parameter `track_last_enqueued_event_properties` in method `EventHubClient.create_consumer()`. - Added new property `last_enqueued_event_properties` of `EventHubConsumer` which contains `sequence_number`, `offset`, `enqueued_time` and `retrieval_time` information. - By default the capability is disabled as it will cost extra bandwidth for transferring more information if turned on. ****Breaking changes**** - Removed support for IoT Hub direct connection. - [EventHubs compatible connection string](https://docs.microsoft.com/azure/iot-hub/iot-hub-devguide-messages-read-builtin) of an IoT Hub can be used to create `EventHubClient` and read properties or events from an IoT Hub. - Removed support for sending `EventData` to IoT Hub. - Removed parameter `exception` in method `close()` of `EventHubConsumer` and `EventHubProducer`. - Updated uAMQP dependency to 1.2.3. **## 5.0.0b3 (2019-09-10) **New features**** - Added support for automatic load balancing among multiple `EventProcessor`. - Added `BlobPartitionManager` which implements `PartitionManager`. - Azure Blob Storage is applied for storing data used by `EventProcessor`. - Packaged separately as a plug-in to `EventProcessor`. - For details, please refer to [Azure Blob Storage Partition Manager](https://github.com/Azure/azure-sdk-for-python/tree/main/sdk/eventhub/azure-eventhub-checkpointstoreblob-aio). - Added property `system_properties` on `EventData`. ****Breaking changes**** - Removed constructor method of `PartitionProcessor`. For initialization please implement the method `initialize`. - Replaced `CheckpointManager` by `PartitionContext`. - `PartitionContext` has partition context information and method `update_checkpoint`. - Updated all methods of `PartitionProcessor` to include `PartitionContext` as part of the arguments. - Updated accessibility of class members in `EventHub/EventHubConsumer/EventHubProducer` to be private. - Moved `azure.eventhub.eventprocessor` under `aio` package, which now becomes `azure.eventhub.aio.eventprocessor`. **## 5.0.0b2 (2019-08-06) **New features**** - Added method `create_batch` on the `EventHubProducer` to create an `EventDataBatch` that can then be used to add events until the maximum size is reached. - This batch object can then be used in the `send()` method to send all the added events to Event Hubs. - This allows publishers to build batches without the possibility of encountering the error around the message size exceeding the supported limit when sending events. - It also allows publishers with bandwidth concerns to control the size of each batch published. - Added new configuration parameters for exponential delay

between retry operations. - `retry_total`: The total number of attempts to redo the failed operation. - `backoff_factor`: The delay time factor. - `backoff_max`: The maximum delay time in total. - Added support for context manager on `EventHubClient`. - Added new error type `OperationTimeoutError` for send operation. - Introduced a new class `EventProcessor` which replaces the older concept of `[Event Processor Host]` (<https://docs.microsoft.com/azure/event-hubs/event-hubs-event-processor-host>). This early preview is intended to allow users to test the new design using a single instance of `EventProcessor`. The ability to checkpoints to a durable store will be added in future updates. - `EventProcessor`: `EventProcessor` creates and runs consumers for all partitions of the eventhub. - `PartitionManager`: `PartitionManager` defines the interface for getting/claiming ownerships of partitions and updating checkpoints. - `PartitionProcessor`: `PartitionProcessor` defines the interface for processing events. - `CheckpointManager`: `CheckpointManager` takes responsibility for updating checkpoints during events processing. **Breaking changes** - `EventProcessorHost` was replaced by `EventProcessor`, please read the new features for details. - Replaced `max_retries` configuration parameter of the `EventHubClient` with `retry_total`. ## 5.0.0b1 (2019-06-25) Version 5.0.0b1 is a preview of our efforts to create a client library that is user friendly and idiomatic to the Python ecosystem. The reasons for most of the changes in this update can be found in the [Azure SDK Design Guidelines for Python] (<https://azuresdkspecs.z5.web.core.windows.net/PythonSpec.html>). For more information, please visit <https://aka.ms/azure-sdk-preview1-python>. **New features** - Added new configuration parameters for creating `EventHubClient`. - `credential`: The credential object used for authentication which implements `TokenCredential` interface of getting tokens. - `transport_type`: The type of transport protocol that will be used for communicating with the Event Hubs service. - `max_retries`: The max number of attempts to redo the failed operation when an error happened. - for detailed information about the configuration parameters, please read the reference documentation. - Added new methods `get_partition_properties` and `get_partition_ids` to `EventHubClient`. - Added support for http proxy. - Added support for authentication using azure-identity credential. - Added support for transport using AMQP over WebSocket. **Breaking changes** - New error hierarchy - `azure.error.EventHubError` - `azure.error.ConnectionLostError` - `azure.error.ConnectError` - `azure.error.AuthenticationError` - `azure.error.EventDataError` - `azure.error.EventDataSendError` - Renamed `Sender/Receiver` to `EventHubProducer/EventHubConsumer`. - Renamed `add_sender` to `create_producer` and `add_receiver` to `create_consumer` in `EventHubClient`. - `EventHubConsumer` is now iterable. - Rename class `azure.eventhub.Offset` to `azure.eventhub.EventPosition`. - Rename method `get_eventhub_info` to `get_properties` of `EventHubClient`. - Reorganized connection management, `EventHubClient` is no longer responsible for

opening/closing EventHubProducer/EventHubConsumer. - Each EventHubProducer/EventHubConsumer is responsible for its own connection management. - Added support for context manager on EventHubProducer and EventHubConsumer. - Reorganized async APIs into "azure.eventhub.aio" namespace and rename to drop the "_async" suffix. - Updated uAMQP dependency to 1.2. ## 1.3.1 (2019-02-28) ****BugFixes**** - Fixed bug where datetime offset filter was using a local timestamp rather than UTC. - Fixed stackoverflow error in continuous connection reconnect attempts. ## 1.3.0 (2019-01-29) ****BugFixes**** - Added support for auto reconnect on token expiration and other auth errors (issue #89). ****Features**** - Added ability to create ServiceBusClient from an existing SAS auth token, including providing a function to auto-renew that token on expiry. - Added support for storing a custom EPH context value in checkpoint (PR #84, thanks @konstantinmiller) ## 1.2.0 (2018-11-29) - Support for Python 2.7 in azure.eventhub module (azure.eventprocessorhost will not support Python 2.7). - Parse EventData.queued_time as a UTC timestamp (issue #72, thanks @vjrantal) ## 1.1.1 (2018-10-03) - Fixed bug in Azure namespace package. ## 1.1.0 (2018-09-21) - Changes to AzureStorageCheckpointLeaseManager` parameters to support other connection options (issue #61): - The `storage_account_name`, `storage_account_key` and `lease_container_name` arguments are now optional keyword arguments. - Added a `sas_token` argument that must be specified with `storage_account_name` in place of `storage_account_key`. - Added an `endpoint_suffix` argument to support storage endpoints in National Clouds. - Added a `connection_string` argument that, if specified, overrides all other endpoint arguments. - The `lease_container_name` argument now defaults to "eph-leases" if not specified. - Fix for clients failing to start if run called multiplied times (issue #64). - Added convenience methods `body_as_str` and `body_as_json` to EventData object for easier processing of message data. ## 1.0.0 (2018-08-22) - API stable. - Renamed internal `_async` module to `async_ops` for docs generation. - Added optional `auth_timeout` parameter to EventHubClient` and EventHubClientAsync` to configure how long to allow for token negotiation to complete. Default is 60 seconds. - Added optional `send_timeout` parameter to EventHubClient.add\_sender` and EventHubClientAsync.add_async_sender` to determine the timeout for Events to be successfully sent. Default value is 60 seconds. - Reformatted logging for performance. ## 0.2.0 (2018-08-06) - Stability improvements for EPH. - Updated uAMQP version. - Added new configuration options for Sender and Receiver; `keep_alive` and `auto_reconnect`. These flags have been added to the following: - EventHubClient.add\_receiver` - EventHubClient.add_sender` - EventHubClientAsync.add\_async\_receiver` - EventHubClientAsync.add_async_sender` - EPHOptions.keep\_alive\_interval` - EPHOptions.auto_reconnect_on_error` ## 0.2.0rc2 (2018-07-29) - **\*\*Breaking change\*\*** EventData.offset` will now return an object of type

`~uamqp.common.Offset`` rather than `str`. The original string value can be retrieved from `~uamqp.common.Offset.value``. - Each sender/receiver will now run in its own independent connection. - Updated uAMQP dependency to 0.2.0 - Fixed issue with IoT Hub clients not being able to retrieve partition information. - Added support for HTTP proxy settings to both `EventHubClient` and `EPH`. - Added error handling policy to automatically reconnect on retryable error. - Added keep-alive thread for maintaining an unused connection. ## 0.2.0rc1 (2018-07-06) - **Breaking change** Restructured library to support Python 3.7. Submodule `~async`` has been renamed and all classes from this module can now be imported from `azure.eventhub` directly. - **Breaking change** Removed optional `callback`` argument from `Receiver.receive`` and `AsyncReceiver.receive``. - **Breaking change** `EventData.properties`` has been renamed to `EventData.application_properties``. This removes the potential for messages to be processed via callback for not yet returned in the batch. - Updated uAMQP dependency to v0.1.0 - Added support for constructing IoT Hub connections. - Fixed memory leak in receive operations. - Dropped Python 2.7 wheel support. ## 0.2.0b2 (2018-05-29) - Added `namespace_suffix`` to `EventHubConfig()` to support national clouds. - Added `device_id`` attribute to `EventData` to support IoT Hub use cases. - Added message header to workaround service bug for `PartitionKey` support. - Updated uAMQP dependency to vRC1. ## 0.2.0b1 (2018-04-20) - Updated uAMQP to latest version. - Further testing and minor bug fixes. ## 0.2.0a2 (2018-04-02) - Updated uAQMP dependency.

Package Info

Package Author(s)

azpysdkhelp@microsoft.com

Package Manager

Pip

Project URL

<https://github.com/Azure/azure-sdk-for-python/tree/main/sdk/eventhub/azure-eventhub>

Package Download URL

https://files.pythonhosted.org/packages/20/bf/96eec5aa57cb08f1e84e1dfa3888e52dbba53e218596cdab5b61d795a1ae/azure_eventhub-5.15.1.tar.gz

Dependency Path

azure-eventhub-checkpointstoreblob > azure-eventhub, azure-eventhub

Declared License(s)

MIT

Copyright(s)

Microsoft Corporation.

Discovered License(s)

None

Notice File(s)

None

azure-eventhub-checkpointstoreblob 1.2.0

pip+azure-eventhub-checkpointstoreblob\$1.2.0

Direct

Description

Azure EventHubs Checkpoint Store using Storage Blobs client library for Python Azure EventHubs Checkpoint Store is used for storing checkpoints while processing events from Azure Event Hubs. This Checkpoint Store package works as a plug-in package to EventHubConsumerClient. It uses Azure Storage Blob as the persistent store for maintaining checkpoints and partition ownership information. Please note that this is a sync library, for async version of the Azure EventHubs Checkpoint Store client library, please refer to [azure-eventhub-checkpointstoreblob-ai] (<https://github.com/Azure/azure-sdk-for-python/blob/main/sdk/eventhub/azure-eventhub-checkpointstoreblob-ai>). [Source code](<https://github.com/Azure/azure-sdk-for-python/blob/main/sdk/eventhub/azure-eventhub-checkpointstoreblob/>) | [Package (PyPi)](<https://pypi.org/project/azure-eventhub-checkpointstoreblob>) | [Package (Conda)](<https://anaconda.org/microsoft/azure-eventhub/>) | [API reference documentation] (<https://azuresdkdocs.z19.web.core.windows.net/python/azure->

eventhub/latest/azure.eventhub.html#azure.eventhub.CheckpointStore) | [Azure Eventhubs documentation](https://learn.microsoft.com/azure/event-hubs/) | [Azure Storage documentation](https://learn.microsoft.com/azure/storage/) ## _Disclaimer_ Azure SDK Python packages support for Python 2.7 has ended 01 January 2022. For more information and questions, please refer to https://github.com/Azure/azure-sdk-for-python/issues/20691_ ## Getting started ### Prerequisites - Python 3.8 or later. - **Microsoft Azure Subscription:** To use Azure services, including Azure Event Hubs, you'll need a subscription. If you do not have an existing Azure account, you may sign up for a free trial or use your MSDN subscriber benefits when you [create an account](https://azure.microsoft.com/). - **Event Hubs namespace with an Event Hub:** To interact with Azure Event Hubs, you'll also need to have a namespace and Event Hub available. If you are not familiar with creating Azure resources, you may wish to follow the step-by-step guide for [creating an Event Hub using the Azure portal](https://learn.microsoft.com/azure/event-hubs/event-hubs-create). There, you can also find detailed instructions for using the Azure CLI, Azure PowerShell, or Azure Resource Manager (ARM) templates to create an Event Hub. - **Azure Storage Account:** You'll need to have an Azure Storage Account and create a Azure Blob Storage Block Container to store the checkpoint data with blobs. You may follow the guide [creating an Azure Block Blob Storage Account](https://learn.microsoft.com/azure/storage/blobs/storage-blob-create-account-block-blob). ### Install the package `pip install azure-eventhub-checkpointstoreblob` ## Key concepts ### Checkpointing Checkpointing is a process by which readers mark or commit their position within a partition event sequence. Checkpointing is the responsibility of the consumer and occurs on a per-partition basis within a consumer group. This responsibility means that for each consumer group, each partition reader must keep track of its current position in the event stream, and can inform the service when it considers the data stream complete. If a reader disconnects from a partition, when it reconnects it begins reading at the checkpoint that was previously submitted by the last reader of that partition in that consumer group. When the reader connects, it passes the offset to the event hub to specify the location at which to start reading. In this way, you can use checkpointing to both mark events as "complete" by downstream applications, and to provide resiliency if a failover between readers running on different machines occurs. It is possible to return to older data by specifying a lower offset from this checkpointing process. Through this mechanism, checkpointing enables both failover resiliency and event stream replay. ### Offsets & sequence numbers Both offset & sequence number refer to the position of an event within a partition. You can think of them as a client-side cursor. The offset is a byte numbering of the event. The offset/sequence number enables an event

consumer (reader) to specify a point in the event stream from which they want to begin reading events. You can specify a timestamp such that you receive events enqueued only after the given timestamp. Consumers are responsible for storing their own offset values outside of the Event Hubs service. Within a partition, each event includes an offset, sequence number and the timestamp of when it was enqueued. ## Examples - [Create an Azure EventHubs EventHubConsumerClient](#create-an-eventhubconsumerclient) - [Consume events using a BlobCheckpointStore](#consume-events-using-a-blobcheckpointstore-to-do-checkpoint) ###

Create an EventHubConsumerClient`The easiest way to create a

EventHubConsumerClient`is to use a connection string. python from

azure.eventhub import EventHubConsumerClient eventhub_client =

EventHubConsumerClient.from_connection_string("my_eventhub_namespace_connection_string",

"my_consumer_group", eventhub_name="my_eventhub")`For other ways of

creating a EventHubConsumerClient, refer to [EventHubs library]

([https://github.com/Azure/azure-sdk-for-](https://github.com/Azure/azure-sdk-for-python/blob/main/sdk/eventhub/azure-eventhub)

[python/blob/main/sdk/eventhub/azure-eventhub](https://github.com/Azure/azure-sdk-for-python/blob/main/sdk/eventhub/azure-eventhub)) for more details. ###

Consume events using a BlobCheckpointStore`to do checkpoint python from

azure.eventhub import EventHubConsumerClient from

azure.eventhub.extensions.checkpointstoreblob import BlobCheckpointStore

connection_str = '<< CONNECTION STRING FOR THE EVENT HUBS

NAMESPACE >>' consumer_group = '<< CONSUMER GROUP >>'

eventhub_name = '<< NAME OF THE EVENT HUB >>' storage_connection_str =

'<< CONNECTION STRING OF THE STORAGE >>' container_name = '<<

STORAGE CONTAINER NAME>>' def on_event(partition_context, event): # Put

your code here. partition_context.update_checkpoint(event) # Or

update_checkpoint every N events for better performance. def main():

checkpoint_store = BlobCheckpointStore.from_connection_string(

storage_connection_str, container_name) client =

EventHubConsumerClient.from_connection_string(connection_str,

consumer_group, eventhub_name=eventhub_name,

checkpoint_store=checkpoint_store,) with client: client.receive(on_event) if

__name__ == '__main__': main() `#### Use BlobCheckpointStore`with a

different version of Azure Storage Service API Some environments have

different versions of Azure Storage Service API. BlobCheckpointStore`by

default uses the Storage Service API version 2019-07-07. To use it against a

different version, specify `api\_version`when you create the

BlobCheckpointStore`object. ## Troubleshooting ### General Enabling

logging will be helpful to do trouble shooting. #### Logging - Enable

`azure.eventhub.extensions.checkpointstoreblob`logger to collect traces

from the library. - Enable `azure.eventhub`logger to collect traces from the

main azure-eventhub library. - Enable

`azure.eventhub.extensions.checkpointstoreblob.\_vendor.storage`logger to

collect traces from azure storage blob library. - Enable `uamqp`logger to

collect traces from the underlying uAMQP library. - Enable AMQP frame level trace by setting `logging_enable=True` when creating the client. ## Next steps ### More sample code Get started with our [EventHubs Checkpoint Store samples](https://github.com/Azure/azure-sdk-for-python/blob/main/sdk/eventhub/azure-eventhub-checkpointstoreblob/samples). - [receive_events_using_checkpoint_store.py](https://github.com/Azure/azure-sdk-for-python/blob/main/sdk/eventhub/azure-eventhub-checkpointstoreblob/samples/receive_events_using_checkpoint_store.py) - EventHubConsumerClient with blob checkpoint store example - [receive_events_using_checkpoint_store_storage_api_version.py](https://github.com/Azure/azure-sdk-for-python/blob/main/sdk/eventhub/azure-eventhub-checkpointstoreblob/samples/receive_events_using_checkpoint_store_storage_api_version.py) - EventHubConsumerClient with blob checkpoint store and storage version example ### Documentation Reference documentation is available [here](https://azuresdkdocs.z19.web.core.windows.net/python/azure-eventhub/latest/azure.eventhub.html#azure.eventhub.CheckpointStore). ### Provide Feedback If you encounter any bugs or have suggestions, please file an issue in the [Issues](https://github.com/Azure/azure-sdk-for-python/issues) section of the project. ## Contributing This project welcomes contributions and suggestions. Most contributions require you to agree to a Contributor License Agreement (CLA) declaring that you have the right to, and actually do, grant us the rights to use your contribution. For details, visit <https://cla.microsoft.com>. When you submit a pull request, a CLA-bot will automatically determine whether you need to provide a CLA and decorate the PR appropriately (e.g., label, comment). Simply follow the instructions provided by the bot. You will only need to do this once across all repos using our CLA. This project has adopted the [Microsoft Open Source Code of Conduct](https://opensource.microsoft.com/codeofconduct/). For more information see the [Code of Conduct FAQ](https://opensource.microsoft.com/codeofconduct/faq/) or contact opencode@microsoft.com with any additional questions or comments. # Release History ## 1.2.0 (2025-02-13) This version and all future versions will require Python 3.8+. Python 2.7, 3.6, and 3.7 are no longer supported. ### Bugs Fixed - Fixed a bug with `BlobCheckpointStore.claim_ownership` mutating the `ownership_list` argument to no longer mutate the argument. - Updated `azure-core` dependency to 1.20.1 to fix `chardet` ImportError. ### Other Changes - Updated vendor `azure-storage-blob` dependency to v12.24.0. - Fixed typing/linting issues and other bugs. See `azure-storage-blob CHANGELOG.md` for more info. ## 1.1.4 (2021-04-07) This version and all future versions will require Python 2.7 or Python 3.6+, Python 3.5 is no longer supported. **New features** - Updated `list_ownership`, `claim_ownership`, `update_checkpoint`, `list_checkpoints` on

BlobCheckpointStore`to support taking `**kwargs`` ## 1.1.3 (2021-03-09) This version will be the last version to officially support Python 3.5, future versions will require Python 2.7 or Python 3.6+. ****Bug fixes**** - Updated vendor azure-storage-blob dependency to v12.7.1. - Fixed storage blob authentication failure due to request date header too old (#16192). ## 1.1.2 (2021-01-11) ****Bug fixes**** - Fixed a bug that BlobCheckpointStore.list_ownership`and BlobCheckpointStore.list\_checkpoints`triggering KeyError`due to reading empty metadata of parent node when working with Data Lake enabled Blob Storage. ## 1.1.1 (2020-09-08) ****Bug fixes**** - Fixed a bug that may gradually slow down retrieving checkpoint data from the storage blob if the storage account "File share soft delete" is enabled. #12836 ## 1.1.0 (2020-03-09) ****New features**** - Param `api_version`` of BlobCheckpointStore`now supports older versions of Azure Storage Service API. ## 1.0.0 (2020-01-13) Stable release. No new features or API changes. ## 1.0.0b6 (2019-12-04) **\*\*Breaking changes\*\*** - Renamed BlobPartitionManager`to BlobCheckpointStore`. - Constructor of BlobCheckpointStore`has been updated to take the storage container details directly rather than an instance of ContainerClient`. - A `from_connection_string`` constructor has been added for Blob Storage connection strings. - Module blobstoragepm`is now internal, all imports should be directly from `azure.eventhub.extensions.checkpointstoreblob``. - BlobCheckpointStore`now has a `close()`` function for shutting down an HTTP connection pool, additionally the object can be used in a context manager to manage the connection. ## 1.0.0b5 (2019-11-04) ****New features**** - BlobPartitionManager`that uses Azure Blob Storage Block Blob to store EventProcessor checkpoint data ![Impressions](https://azure-sdk-impressions.azurewebsites.net/api/impressions/azure-sdk-for-python/sdk/eventhub/azure-eventhub-checkpointstoreblob/HISTORY.png)

Package Info

Package Author(s)

azpysdkhelp@microsoft.com

Package Manager

Pip

Project URL

<https://github.com/Azure/azure-sdk-for-python/tree/main/sdk/eventhub/azure-eventhub-checkpointstoreblob>

Package Download URL

https://files.pythonhosted.org/packages/0d/f1/b683ef6e5e053401023453ef0ba77b6e029e5efbc5fa1468a36b0affac25/azure_eventhub_checkpointstoreblob-1.2.0.tar.gz

Dependency Path

azure-eventhub-checkpointstoreblob

Declared License(s)

MIT

Copyright(s)

Microsoft Corporation.

Discovered License(s)

None

Notice File(s)

None

azure-identity 1.25.3

pip+azure-identity\$1.25.3

Direct

Description

Microsoft Azure Identity Library for Python

Package Info

Package Author(s)

Microsoft Corporation <azpysdkhelp@microsoft.com> License-Expression: MIT

Package Manager

Pip

Project URL

<https://pypi.org/project/azure-identity/>

Package Download URL

https://files.pythonhosted.org/packages/c5/0e/3a63efb48aa4a5ae2cfca61ee152fbc6b668092134d3eb8bfda472dd5c617/azure_identity-1.25.3.tar.gz

Dependency Path

azure-identity

Declared License(s)

MIT

Copyright(s)

Microsoft Corporation.

Discovered License(s)

None

Notice File(s)

None

azure-mgmt-compute 38.0.0

pip+azure-mgmt-compute\$38.0.0

Direct

Description

Microsoft Azure Compute Management Client Library for Python

Package Info

Package Author(s)

Microsoft Corporation <azpysdkhelp@microsoft.com>

Package Manager

Pip

Project URL

<https://pypi.org/project/azure-mgmt-compute/>

Package Download URL

https://files.pythonhosted.org/packages/4f/ee/a143c8395267aea8d735c5336519ef94eeaa45f235c89c06f417d1d2ce60/azure_mgmt_compute-38.0.0.tar.gz

Dependency Path

azure-mgmt-compute

Declared License(s)

MIT

Copyright(s)

Microsoft Corporation.

Discovered License(s)

None

Notice File(s)

None

azure-mgmt-resource 25.0.0

pip+azure-mgmt-resource\$25.0.0

Direct

Description

Microsoft Azure Resource Management Client Library for Python

Package Info

Package Author(s)

Microsoft Corporation <azpysdkhelp@microsoft.com>

Package Manager

Pip

Project URL

<https://pypi.org/project/azure-mgmt-resource/>

Package Download URL

https://files.pythonhosted.org/packages/d7/7e/b3f9d544a94782be9c5ab8123d2fa7fb20cbdf3f5a16b883f308865e1406/azure_mgmt_resource-25.0.0.tar.gz

Dependency Path

azure-mgmt-resource

Declared License(s)

MIT

Copyright(s)

Microsoft Corporation.

Discovered License(s)

None

Notice File(s)

None

azure-storage-blob 12.29.0

pip+azure-storage-blob\$12.29.0

Direct

Description

Microsoft Azure Blob Storage Client Library for Python

Package Info

Package Author(s)

ascl@microsoft.com

Package Manager

Pip

Project URL

<https://github.com/Azure/azure-sdk-for-python/tree/main/sdk/storage/azure-storage-blob>

Package Download URL

https://files.pythonhosted.org/packages/59/25/fdcf1e381922dbab8ba23d6fd78d397fe6cbac6b480310218834b7bc91fe/azure_storage_blob-12.29.0.tar.gz

Dependency Path

azure-storage-blob

Declared License(s)

MIT

Copyright(s)

Microsoft Corporation.

Discovered License(s)

None

Notice File(s)

None

cattr 23.1.2

pip+cattr\$23.1.2

Direct

Description

Composable complex class support for attr and dataclasses.

Package Info

Package Author(s)

Tin Tvrtkovic <tinchester@gmail.com>

Package Manager

Pip

Project URL

<https://pypi.org/project/cattr/>

Package Download URL

<https://files.pythonhosted.org/packages/68/d4/27f9fd840e74d51b6d6a024d39ff495b56ffde71d28eb82758b7b85d0617/cattr-23.1.2.tar.gz>

Dependency Path

cattr

Declared License(s)

MIT

Copyright(s)

2016 Tin Tvrtković

Discovered License(s)

None

Notice File(s)

None

certifi 2025.8.3

pip+certifi\$2025.8.3

Direct

Description

Python package for providing Mozilla's CA Bundle.

Package Info

Package Author(s)
me@kennethreitz.com

Package Manager
Pip

Project URL
<https://github.com/certifi/python-certifi>

Package Download URL
<https://files.pythonhosted.org/packages/dc/67/960ebe6bf230a96cda2e0abcf73af550ec4f090005363542f0765df162e0/certifi-2025.8.3.tar.gz>

Dependency Path
splunktaucclib > requests > certifi, splunktaucclib > splunktalib > requests > certifi, splunktalib > requests > certifi, azure-identity > msal-extensions > msal > requests > certifi, azure-identity > msal > requests > certifi, azure-data-tables > azure-core > requests > certifi, azure-core > requests > certifi, ssh://git@github.com/splunk/splunksdcs.git > requests > certifi, requests > certifi, certifi

Declared License(s)

MPL-2.0

Copyright(s)

None

Discovered License(s)

None

Notice File(s)

None

cryptography 48.0.0

pip+cryptography\$48.0.0

Direct

Description

pyca/cryptography ===== .. image::
https://img.shields.io/pypi/v/cryptography.svg :target:
https://pypi.org/project/cryptography/ :alt: Latest Version .. image::
https://readthedocs.org/projects/cryptography/badge/?version=latest
:target: https://cryptography.io :alt: Latest Docs .. image::
https://github.com/pyca/cryptography/workflows/CI/badge.svg?
branch=main :target: https://github.com/pyca/cryptography/actions?
query=workflow%3ACI+branch%3Amain `cryptography` is a package which
provides cryptographic recipes and primitives to Python developers. Our
goal is for it to be your "cryptographic standard library". It supports Python
3.7+ and PyPy3 7.3.11+. `cryptography` includes both high level recipes and
low level interfaces to common cryptographic algorithms such as symmetric
ciphers, message digests, and key derivation functions. For example, to
encrypt something with `cryptography`'s high level symmetric encryption
recipe: .. code-block:: pycon >>> from cryptography.fernet import Fernet >>>
Put this somewhere safe! >>> key = Fernet.generate_key() >>> f =
Fernet(key) >>> token = f.encrypt(b"A really secret message. Not for prying
eyes.") >>> token b'...' >>> f.decrypt(token) b'A really secret message. Not for
prying eyes.' You can find more information in the `documentation`_. You can
install `cryptography` with: .. code-block:: console \$ pip install cryptography
For full details see the installation documentation_. Discussion ~~~~~ If
you run into bugs, you can file them in our issue tracker_. We maintain a
`cryptography-dev`_ mailing list for development discussion. You can also join
#pyca on irc.libera.chat to ask questions or get involved. Security ~~~~~
Need to report a security issue? Please consult our `security reporting`_

documentation. .._documentation: <https://cryptography.io/> .._the installation documentation: <https://cryptography.io/en/latest/installation/> .._issue tracker: <https://github.com/pyca/cryptography/issues> .._cryptography-dev: <https://mail.python.org/mailman/listinfo/cryptography-dev> .._security reporting: <https://cryptography.io/en/latest/security/>

Package Info

Package Author(s)

The Python Cryptographic Authority and individual contributors
<cryptography-dev@python.org>

Package Manager

Pip

Project URL

<https://pypi.org/project/cryptography/>

Package Download URL

<https://files.pythonhosted.org/packages/9f/a9/db8f313fdcd85d767d4973515e1db101f9c71f95fced83233de224673757/cryptography-48.0.0.tar.gz>

Dependency Path

azure-identity > msal-extensions > msal > pyjwt > cryptography, azure-identity > msal > pyjwt > cryptography, azure-identity > msal-extensions > msal > cryptography, azure-identity > msal > cryptography, azure-storage-blob > cryptography, azure-identity > cryptography, azure-eventhub-checkpointstoreblob > cryptography, cryptography

Declared License(s)

Apache-2.0

Copyright(s)

None

Discovered License(s)

None

Notice File(s)

None

exceptiongroup 1.3.1

pip+exceptiongroup\$1.3.1

Direct

Description

Backport of PEP 654 (exception groups)

Package Info

Package Author(s)

Alex Grönholm <alex.gronholm@nextday.fi>

Package Manager

Pip

Project URL

<https://pypi.org/project/exceptiongroup/>

Package Download URL

<https://files.pythonhosted.org/packages/50/79/66800aadf48771f6b62f7eb014e352e5d06856655206165d775e675a02c9/exceptiongroup-1.3.1.tar.gz>

Dependency Path

cattr > exceptiongroup, exceptiongroup

Declared License(s)

Python-2.0

Copyright(s)

2022 Alex Grönholm

MIT

Copyright(s)

2022 Alex Grönholm

Discovered License(s)

None

Notice File(s)

None

grpcio 1.74.0

pip+grpcio\$1.74.0

Direct

Description

HTTP/2-based RPC framework

Package Info

Package Author(s)

The gRPC Authors <grpc-io@googlegroups.com>

Package Manager

Pip

Project URL

<https://pypi.org/project/grpcio/>

Package Download URL

<https://files.pythonhosted.org/packages/38/b4/35feb8f7cab7239c5b94bd2db71abb3d6adb5f335ad8f131abb6060840b6/grpcio-1.74.0.tar.gz>

Dependency Path

splunktaucclib > solnlib > grpcio, solnlib > grpcio, splunktaucclib > solnlib > opentelemetry-exporter-otlp-proto-grpc > grpcio, solnlib > opentelemetry-exporter-otlp-proto-grpc > grpcio, opentelemetry-exporter-otlp-proto-grpc > grpcio, grpcio

Declared License(s)**MPL-2.0****Copyright(s)**

2016 Google Inc.

BSD-3-Clause**Copyright(s)**

2016 Google Inc.

Apache-2.0**Copyright(s)**

2016 Google Inc.

Discovered License(s)**apache-2.0 WITH llvm-exception****Copyright(s)**

None

BSD-2-Clause**Copyright(s)**

2005 Google Inc.

Info-ZIP

Copyright(s)

1998-2010 Gilles Vollant (minizip) (
<http://www.winimage.com/zLibDll/minizip.html>

ISC

Copyright(s)

2004 by Internet Systems Consortium, Inc. ("ISC")

MIT

Copyright(s)

2022 Google LLC

mit-old-style

Copyright(s)

1997

mit-old-style-no-advert

Copyright(s)

2009 - 2013 by Daniel Stenberg et al

Zlib

Copyright(s)

2003, 2012, 2013 Mark Adler

Notice File(s)

None

idna 3.16

pip+idna\$3.16

Direct

Description

Internationalized Domain Names in Applications (IDNA)

Package Info

Package Author(s)

Kim Davies <kim+pypi@gumleaf.org>

Package Manager

Pip

Project URL

<https://pypi.org/project/idna/>

Package Download URL

<https://files.pythonhosted.org/packages/1a/88/bcf9709822fe69d02c2a6a77956c98ce6ea8ca8767a9aadcedc7eb6a2390/idna-3.16.tar.gz>

Dependency Path

azure-data-tables > yarl > idna, splunktaucclib > requests > idna,
splunktaucclib > splunktalib > requests > idna, splunktalib > requests > idna,
azure-identity > msal-extensions > msal > requests > idna, azure-identity >
msal > requests > idna, azure-core > requests > idna,
ssh://git@github.com/splunk/splunksdcs.git > requests > idna, requests >
idna, idna

Declared License(s)

BSD-3-Clause

Copyright(s)

2013-2026 Kim Davies and contributors.

Discovered License(s)

None

Notice File(s)

None

isodate 0.7.2

pip+isodate\$0.7.2

Direct

Description

An ISO 8601 date/time/duration parser and formatter

Package Info

Package Author(s)

Gerhard Weis

Package Manager

Pip

Project URL

<https://pypi.org/project/isodate/>

Package Download URL

<https://files.pythonhosted.org/packages/54/4d/e940025e2ce31a8ce1202635910747e5a87cc3a6a6bb2d00973375014749/isodate-0.7.2.tar.gz>

Dependency Path

azure-storage-blob > isodate, azure-mgmt-resource > isodate, azure-mgmt-compute > isodate, azure-eventhub-checkpointstoreblob > isodate, azure-data-tables > isodate, isodate

Declared License(s)

BSD-3-Clause

Copyright(s)

2021 Hugo van Kemenade and contributors

Discovered License(s)

None

Notice File(s)

None

opentelemetry-api 1.39.1

pip+opentelemetry-api\$1.39.1

Direct

Description

OpenTelemetry Python API

Package Info

Package Author(s)

OpenTelemetry Authors <cncf-opentelemetry-contributors@lists.cncf.io>

Package Manager

Pip

Project URL

<https://pypi.org/project/opentelemetry-api/>

Package Download URL

https://files.pythonhosted.org/packages/97/b9/3161be15bb8e3ad01be8be5a968a9237c3027c5be504362ff800fca3e442/opentelemetry_api-1.39.1.tar.gz

Dependency Path

splunktaucclib > solnlib > opentelemetry-api, solnlib > opentelemetry-api, splunktaucclib > solnlib > opentelemetry-sdk > opentelemetry-semantic-conventions > opentelemetry-api, solnlib > opentelemetry-sdk > opentelemetry-semantic-conventions > opentelemetry-api, solnlib > opentelemetry-exporter-otlp-proto-grpc > opentelemetry-sdk > opentelemetry-semantic-conventions > opentelemetry-api, opentelemetry-exporter-otlp-proto-grpc > opentelemetry-sdk > opentelemetry-semantic-conventions > opentelemetry-api, opentelemetry-sdk > opentelemetry-semantic-conventions > opentelemetry-api, opentelemetry-sdk > opentelemetry-api, opentelemetry-exporter-otlp-proto-grpc > opentelemetry-api, opentelemetry-api

Declared License(s)

Apache-2.0

Copyright(s)

The OpenTelemetry Authors

Discovered License(s)

None

Notice File(s)

None

opentelemetry-exporter-otlp-proto-grpc 1.39.1

pip+opentelemetry-exporter-otlp-proto-grpc\$1.39.1

Direct

Description

OpenTelemetry Collector Protobuf over gRPC Exporter

Package Info

Package Author(s)

OpenTelemetry Authors <cncf-opentelemetry-contributors@lists.cncf.io>

Package Manager

Pip

Project URL

<https://pypi.org/project/opentelemetry-exporter-otlp-proto-grpc/>

Package Download URL

https://files.pythonhosted.org/packages/53/48/b329fed2c610c2c32c9366d9dc597202c9d1e58e631c137ba15248d8850f/opentelemetry_exporter_otlp_proto_grpc-1.39.1.tar.gz

Dependency Path

splunktaucclib > solnlib > opentelemetry-exporter-otlp-proto-grpc, solnlib > opentelemetry-exporter-otlp-proto-grpc, opentelemetry-exporter-otlp-proto-grpc

Declared License(s)

Apache-2.0

Copyright(s)

The OpenTelemetry Authors

Discovered License(s)

None

Notice File(s)

None

opentelemetry-sdk 1.39.1

pip+opentelemetry-sdk\$1.39.1

Direct

Description

OpenTelemetry Python SDK

Package Info

Package Author(s)

OpenTelemetry Authors <cncf-opentelemetry-contributors@lists.cncf.io>

Package Manager

Pip

Project URL

<https://pypi.org/project/opentelemetry-sdk/>

Package Download URL

https://files.pythonhosted.org/packages/eb/fb/c76080c9ba07e1e8235d24cdcc4d125ef7aa3edf23eb4e497c2e50889adc/opentelemetry_sdk-1.39.1.tar.gz

Dependency Path

splunktaucclib > solnlib > opentelemetry-sdk, solnlib > opentelemetry-sdk,
splunktaucclib > solnlib > opentelemetry-exporter-otlp-proto-grpc >
opentelemetry-sdk, solnlib > opentelemetry-exporter-otlp-proto-grpc >
opentelemetry-sdk, opentelemetry-exporter-otlp-proto-grpc >
opentelemetry-sdk, opentelemetry-sdk

Declared License(s)

Apache-2.0

Copyright(s)

The OpenTelemetry Authors

Discovered License(s)

None

Notice File(s)

None

raw-loader 4.0.2

npm+raw-loader\$4.0.2

Direct

Description

A loader for webpack that allows importing files as a String

Package Info**Package Author(s)**

bebraw@gmail.com, eric@smarterspam.com, mail@johannesewald.de,
michael.ciniawsky@gmail.com, sean.larkin@cuw.edu,
sheo13666q@gmail.com, tobias.koppers@googlemail.com,
wiens.joshua@gmail.com

Package Manager

NPM

Project URL<https://www.npmjs.com/package/raw-loader>**Package Download URL**<https://registry.npmjs.org/raw-loader/-/raw-loader-4.0.2.tgz>**Dependency Path**

raw-loader

Declared License(s)

MIT

Copyright(s)

JS Foundation and other contributors

Discovered License(s)

None

Notice File(s)

None

react 16.14.0

npm+react\$16.14.0

Direct

Description

React is a JavaScript library for building user interfaces.

Package Info

Package Author(s)

opensource+npm@fb.com, react-core@meta.com

Package Manager

NPM

Project URL

<https://react.dev/>

Package Download URL

<https://registry.npmjs.org/react/-/react-16.14.0.tgz>

Dependency Path

react

Declared License(s)

MIT

Copyright(s)

Facebook, Inc. and its affiliates.

Discovered License(s)

None

Notice File(s)

None

react-dom 16.14.0
npm+react-dom\$16.14.0

Direct

Description

React package for working with the DOM.

Package Info

Package Author(s)
opensource+npm@fb.com, react-core@meta.com

Package Manager
NPM

Project URL
<https://react.dev/>

Package Download URL
<https://registry.npmjs.org/react-dom/-/react-dom-16.14.0.tgz>

Dependency Path
react-dom

Declared License(s)

MIT

Copyright(s)
Facebook, Inc. and its affiliates.

Discovered License(s)

None

Notice File(s)

None

requests 2.32.5

pip+requests\$2.32.5

Direct

Description

Python HTTP for Humans.

Package Info

Package Author(s)

Kenneth Reitz <me@kennethreitz.org>

Package Manager

Pip

Project URL

<https://pypi.org/project/requests/>

Package Download URL

<https://files.pythonhosted.org/packages/c9/74/b3ff8e6c8446842c3f5c837e9c3dfcfe2018ea6ecef224c710c85ef728f4/requests-2.32.5.tar.gz>

Dependency Path

splunktaucclib > requests, splunktaucclib > splunktalib > requests,
splunktalib > requests, azure-identity > msal-extensions > msal > requests,
azure-identity > msal > requests, azure-eventhub > azure-core > requests,
azure-data-tables > azure-core > requests, azure-core > requests,
ssh://git@github.com/splunk/splunksdsc.git > requests, requests

Declared License(s)

Apache-2.0

Copyright(s)

None

Discovered License(s)

None

Notice File(s)

NOTICE

File Path

NOTICE

Notice Text

Requests
Copyright 2019 Kenneth Reitz

Copyright(s)

Copyright © 2019 Kenneth Reitz

solnlib 8.1.0

pip+solnlib\$8.1.0

Direct

Description

The Splunk Software Development Kit for Splunk Solutions

Package Info

Package Author(s)

addonfactory@splunk.com

Package Manager

Pip

Project URL

<https://github.com/splunk/addonfactory-solutions-library-python>

Package Download URL

<https://files.pythonhosted.org/packages/21/3c/5501f1d50fae9c6ce9150d711e6d0120cbef6856275132e16b911ebddd9c/solnlib-8.1.0.tar.gz>

Dependency Path

splunktaucclib > solnlib, solnlib

Declared License(s)

Apache-2.0

Copyright(s)

2021 Splunk Inc.

Discovered License(s)

None

Notice File(s)

None

sortedcontainers 2.4.0

pip+sortedcontainers\$2.4.0

Direct

Description

Sorted Containers -- Sorted List, Sorted Dict, Sorted Set

Package Info

Package Author(s)

contact@grantjenks.com

Package Manager

Pip

Project URL

<http://www.grantjenks.com/docs/sortedcontainers/>

Package Download URL

<https://files.pythonhosted.org/packages/e8/c4/ba2f8066cceb6f23394729afe52f3bf7adec04bf9ed2c820b39e19299111/sortedcontainers-2.4.0.tar.gz>

Dependency Path

splunktaucclib > splunktalib > sortedcontainers, splunktalib >
sortedcontainers, splunktaucclib > solnlib > sortedcontainers, solnlib >
sortedcontainers, ssh://git@github.com/splunk/splunksdc.git >
sortedcontainers, sortedcontainers

Declared License(s)

Apache-2.0

Copyright(s)

2014-2019 Grant Jenks

Discovered License(s)

None

Notice File(s)

None

splunk-sdk 2.1.0

pip+splunk-sdk\$2.1.0

Direct

Description

None

Package Info

Package Author(s)

devinfo@splunk.com

Package Manager

Pip

Project URL

<http://github.com/splunk/splunk-sdk-python>

Package Download URL

<https://files.pythonhosted.org/packages/2c/89/31974e00319b750a5a126717d286e8addb683554454fed1ae6c86977cbcf/splunk-sdk-2.1.0.tar.gz>

Dependency Path

splunktaucclib > splunk-sdk, splunktaucclib > solnlib > splunk-sdk, solnlib > splunk-sdk, ssh://git@github.com/splunk/splunksdc.git > splunk-sdk, splunk-sdk

Declared License(s)

Apache-2.0

Copyright(s)

2011-2015 Splunk, Inc.

Discovered License(s)

None

Notice File(s)

None

splunktalib 3.0.6

pip+splunktalib\$3.0.6

Direct

Description

Supporting library for Splunk Add-ons

Package Info

Package Author(s)
rfaircloth@splunk.com

Package Manager
Pip

Project URL
<https://github.com/splunk/addonfactory-ta-library-python>

Package Download URL
<https://files.pythonhosted.org/packages/fa/e6/c2010c0b1b29b90d83b1a4eb8be59a9773f9fd316724024411c34555b5cc/splunktalib-3.0.6.tar.gz>

Dependency Path
splunktaucclib > splunktalib, splunktalib

Declared License(s)

Apache-2.0

Copyright(s)
2021 Splunk Inc.

Discovered License(s)

None

Notice File(s)

None

splunktaucclib 6.6.0

pip+splunktaucclib\$6.6.0

Direct

Description

None

Package Info

Package Author(s)
addonfactory@splunk.com

Package Manager
Pip

Project URL
<https://pypi.org/project/splunktaucclib/>

Package Download URL
<https://files.pythonhosted.org/packages/04/44/3ff499ad18a8407bc82a5676493edd1509943e75bbf0703f6ba28635f72f/splunktaucclib-6.6.0.tar.gz>

Dependency Path
splunktaucclib

Declared License(s)

Apache-2.0

Copyright(s)
2021 Splunk Inc.

Discovered License(s)

None

Notice File(s)

None

styled-components 5.3.11

npm+styled-components\$5.3.11

Direct

Description

Fast, expressive styling for React.

Package Info

Package Author(s)

contact@mxstbr.com, ej@quantizor.dev, phil@kitten.sh

Package Manager

NPM

Project URL

<https://styled-components.com/>

Package Download URL

<https://registry.npmjs.org/styled-components/-/styled-components-5.3.11.tgz>

Dependency Path

styled-components

Declared License(s)

MIT

Copyright(s)

None

Discovered License(s)

None

Notice File(s)

None

u-msgpack-python 2.8.0

pip+u-msgpack-python\$2.8.0

Direct

Description

A portable, lightweight MessagePack serializer and deserializer written in pure Python.

Package Info

Package Author(s)

v@sergeev.io

Package Manager

Pip

Project URL

<https://github.com/vsergeev/u-msgpack-python>

Package Download URL

<https://files.pythonhosted.org/packages/36/9d/a40411a475e7d4838994b7f6bcc6bfca9acc5b119ce3a7503608c4428b49/u-msgpack-python-2.8.0.tar.gz>

Dependency Path

ssh://git@github.com/splunk/splunksd.git > u-msgpack-python, u-msgpack-python

Declared License(s)

MIT

Copyright(s)

2013-2023 vsergeev / Ivan (Vanya) A. Sergeev

Discovered License(s)

None

Notice File(s)

None

underscore 1.13.8

npm+underscore\$1.13.8

Direct

Description

JavaScript's functional programming helper library.

Package Info

Package Author(s)

dev@juliangonggrijp.com, jashkenas@gmail.com

Package Manager

NPM

Project URL

<https://underscorejs.org/>

Package Download URL

<https://registry.npmjs.org/underscore/-/underscore-1.13.8.tgz>

Dependency Path

underscore

Declared License(s)

MIT

Copyright(s)

2009-2022 Jeremy Ashkenas, Julian Gonggrijp, and DocumentCloud and Investigative Reporters & Editors

Discovered License(s)

None

Notice File(s)

None

urllib3 1.26.19

pip+urllib3\$1.26.19

Direct

Description

```
<h1 align="center"> ![urllib3]
(https://github.com/urllib3/urllib3/raw/main/docs/_static/banner_github.svg)
</h1> <p align="center"> <a href="https://pypi.org/project/urllib3"></a> <a href="https://pypi.org/project/urllib3">
</a> <a href="https://discord.gg/urllib3"></a> <a
href="https://github.com/urllib3/urllib3/actions?query=workflow%3ACI">
</a> <a href="https://github.com/urllib3/urllib3/actions?
query=workflow%3ACI"></a> <a
href="https://urllib3.readthedocs.io"></a>
<br> <a href="https://deps.dev/pypi/urllib3"></a> <a href="https://slsa.dev"></a> <a
href="https://bestpractices.coreinfrastructure.org/projects/6227">
</a> </p>
urllib3 is a powerful, *user-friendly* HTTP client for Python. Much
of the Python ecosystem already uses urllib3 and you should too. urllib3
brings many critical features that are missing from the Python standard
libraries: - Thread safety. - Connection pooling. - Client-side SSL/TLS
verification. - File uploads with multipart encoding. - Helpers for retrying
requests and dealing with HTTP redirects. - Support for gzip, deflate, brotli,
and zstd encoding. - Proxy support for HTTP and SOCKS. - 100% test
coverage. urllib3 is powerful and easy to use:
python3 >>> import urllib3 >>>
resp = urllib3.request("GET", "http://httpbin.org/robots.txt") >>> resp.status
200 >>> resp.data
b'User-agent: *\nDisallow: /deny\n'
`## Installing urllib3
can be installed with [pip](https://pip.pypa.io):
bash $ python -m pip install urllib3
Alternatively, you can grab the latest source code from [GitHub]
(https://github.com/urllib3/urllib3):
bash $ git clone https://github.com/urllib3/urllib3.git
$ cd urllib3 $ pip install . `##
Documentation
urllib3 has usage and reference documentation at
[urllib3.readthedocs.io](https://urllib3.readthedocs.io).
## Community
urllib3 has a [community Discord channel](https://discord.gg/urllib3) for asking
questions and collaborating with other contributors. Drop by and say hello 🙌
## Contributing
urllib3 happily accepts contributions. Please see our
[contributing documentation]
(https://urllib3.readthedocs.io/en/latest/contributing.html) for some tips on
getting started.
## Security Disclosures
To report a security vulnerability, please use the
[Tidelift security contact](https://tidelift.com/security).
Tidelift will coordinate the fix and disclosure with maintainers.
## Maintainers
- [@sethmlarson](https://github.com/sethmlarson) (Seth M. Larson) -
[@pquentin](https://github.com/pquentin) (Quentin Pradet) -
[@illia-v](https://github.com/illia-v) (Illia Volochii) -
[@theacodes](https://github.com/theacodes) (Thea Flowers) -
[@haikuginger](https://github.com/haikuginger) (Jess Shapiro) -
[@lukasa](https://github.com/lukasa) (Cory Benfield) -
[@sigmavirus24](https://github.com/sigmavirus24) (Ian Stapleton Cordasco) -
[@shazow](https://github.com/shazow) (Andrey Petrov) 🙌
## Sponsorship
If your company benefits from this library, please consider
[sponsoring its development](https://urllib3.readthedocs.io/en/latest/sponsors.html).
## For Enterprise
Professional support for urllib3 is available as part of the
[Tidelift Subscription][1]. Tidelift gives software development teams a single
source for purchasing and maintaining their software, with professional grade
assurances from the experts who know it best, while seamlessly integrating

```

with existing tools. [1]: https://tidelift.com/subscription/pkg/pypi-urllib3?utm_source=pypi-urllib3&utm_medium=referral&utm_campaign=readme

Package Info

Package Author(s)

Andrey Petrov <andrey.petrov@shazow.net>

Package Manager

Pip

Project URL

<https://pypi.org/project/urllib3/>

Package Download URL

<https://files.pythonhosted.org/packages/c8/93/65e479b023bbc46dab3e092bda6b0005424ea3217d711964ccdede3f9b1b/urllib3-1.26.19.tar.gz>

Dependency Path

splunktaucclib > urllib3, splunktaucclib > splunktalib > urllib3, splunktalib > urllib3, splunktaucclib > requests > urllib3, splunktalib > requests > urllib3, azure-identity > msal > requests > urllib3, azure-core > requests > urllib3, ssh://git@github.com/splunk/splunksdsc.git > requests > urllib3, requests > urllib3, urllib3

Declared License(s)

MIT

Copyright(s)

2008-2020 Andrey Petrov and contributors (see CONTRIBUTORS.txt)

Discovered License(s)

None

Notice File(s)

None

websocket-client 1.8.0

pip+websocket-client\$1.8.0

Direct

Description

WebSocket client for Python with low level API options

Package Info

Package Author(s)

liris.pp@gmail.com

Package Manager

Pip

Project URL

<https://github.com/websocket-client/websocket-client.git>

Package Download URL

https://files.pythonhosted.org/packages/e6/30/fba0d96b4b5bf5948ed3f4681f7da2f9f64512e1d303f94b4cc174c24a5/websocket_client-1.8.0.tar.gz

Dependency Path

websocket-client

Declared License(s)

Apache-2.0

Copyright(s)

2024 engn33r

Discovered License(s)

None

Notice File(s)

None

zipp 3.23.0

pip+zipp\$3.23.0

Direct

Description

Backport of pathlib-compatible object wrapper for zip files

Package Info**Package Author(s)**

"Jason R. Coombs" <jaraco@jaraco.com>

Package Manager

Pip

Project URL

<https://pypi.org/project/zipp/>

Package Download URL

<https://files.pythonhosted.org/packages/e3/02/0f2892c661036d50ede074e376733dca2ae7c6eb617489437771209d4180/zipp-3.23.0.tar.gz>

Dependency Path

splunktaucclib > solnlib > opentelemetry-api > importlib-metadata > zipp,
solnlib > opentelemetry-api > importlib-metadata > zipp, splunktaucclib >
solnlib > opentelemetry-sdk > opentelemetry-semantic-conventions >
opentelemetry-api > importlib-metadata > zipp, solnlib > opentelemetry-sdk >
opentelemetry-semantic-conventions > opentelemetry-api > importlib-
metadata > zipp, opentelemetry-exporter-otlp-proto-grpc > opentelemetry-
sdk > opentelemetry-semantic-conventions > opentelemetry-api > importlib-
metadata > zipp, opentelemetry-sdk > opentelemetry-semantic-conventions
> opentelemetry-api > importlib-metadata > zipp, opentelemetry-sdk >
opentelemetry-api > importlib-metadata > zipp, opentelemetry-exporter-otlp-
proto-grpc > opentelemetry-api > importlib-metadata > zipp, opentelemetry-
api > importlib-metadata > zipp, zipp

Declared License(s)

MIT

Copyright(s)

2025 copyright holders>

Discovered License(s)

None

Notice File(s)

None

ssh://git@github.com/splunk/splunksdc.git v1.2.1

git+ssh://git@github.com/splunk/splunksdc.git\$ v1.2.1

Direct

Description

None

Package Info

Package Author(s)

None

Package Manager

Git

Project URL

None

Package Download URL

None

Dependency Path

ssh://git@github.com/splunk/splunksdc.git

Declared License(s)

None

Discovered License(s)

None

Notice File(s)

None