SMQTK

Release 0.18.0

Kitware, Inc.

Jul 03, 2023

CONTENTS

GitHub

Python toolkit for pluggable algorithms and data structures for multimedia-based machine learning.

CHAPTER

ONE

INSTALLATION

Please reference the SMQTK-Core installation documentation as such documentation for this package is nearly identical. Of course, replace uses of *smqtk-core* with *smqtk-dataprovider*.

CHAPTER

TWO

RELEASE PROCESS AND NOTES

2.1 Steps of the SMQTK Release Process

Please reference the SMQTK-Core release process documentation as that same process is applicable here, of course replacing uses of *smqtk-core* with *smqtk-dataprovider*.

2.2 Release Notes

2.2.1 Pending Release Notes

Updates / New Features

Fixes

2.2.2 SMQTK v0.15.0 Release Notes

This is the initial release of `smqtk-dataprovider`, spinning off from v0.14.0 of the monolithic SMQTK library.

Fixes

CI

- Fix issues with typechecking caused by using more strict checks.
- Add CI for github using actions.

Misc.

- Minor fixes to package metadata files.
- Fixed issue with packages specifier in `setup.py` where it was only excluding the top-level `tests` module but including the rest. Fixed to only explicitly include the `smqtk_descriptors` package and submodules.

2.2.3 v0.16.0

This minor release updates the build system used to Poetry, updates the `smqtk-core` package dependency to a version >= 0.18.0 (the current release) and makes use of its importlib metadata pass-through.

Updates / New Features

Dependencies

- Remove dependency on `setuptool`'s `pkg_resources` module. Taking the stance of bullet number 5 in from Python's Packaging User-guide with regards to getting this package's version. The "needs to be installed" requirement from before is maintained.
- Added `ipython` (and appropriately supporting version of `jedi`) as development dependencies. Minimum versioning is set to support python 3.6 (current versions follow NEP 29 and thus require python 3.7+).

Misc.

- Now standardize to using Poetry for environment/build/publish management.
  - Collapsed pytest configuration into the `pyproject.toml` file.

Testing

- Added terminal-output coverage report in the standard pytest config in the `pyproject.toml` file.

Fixes

CI

- Remove a debug command in a GitHub CI workflow job.
- Fix some LGTM warnings.
- Update CI configurations to use Poetry.

Docs

- Fix for use with poetry where appropriate.

2.2.4 v0.17.0

This minor release removes support for python version 3.6 which has since reached EoL.

Updates / New Features

CI

- Updated CI unittests workflow to include codecov reporting. Reduced CodeCov report submission by skipping this step on scheduled runs.
- Update GitHub actions workflows with pinned python versions to use 3.7.
- Update code-cov action usage to use v3.
- Added properties file for use with SonarQube and SonarCloud.
- Added script and workflow to support release process as described in smqtk-core shared document.
- Added explicit provision of codecov repository token to github action.

- Add testing for py3.11.
- Use modern numpy for python 3.8 and beyond.

Data Elements

- Memory
  - Removed assertion that given data was specifically a bytes instance via superfluous `memoryview` construction.
- PostgreSQL
  - Removed outdated defaults for host and port.
- URL
  - Removed injection of `http` on construction to the beginning of a given URL if any schema was missing.

Dependencies

- Updated minimum required python version to 3.7 to follow python end of life.
- Updated development abstract dep versions to "*" since we do not currently require any specific versions.

Documentation

- Updated CONTRIBUTING.md to reference smqtk-core's CONTRIBUTING.md file.

Fixes

CI

- Modified CI unittests workflow to run for PRs targeting branches that match the *release** glob.
- Fixed new issues raised by updated version of `mypy`.

Dependency Versions

- Updated the locked version of urllib3 to address a security vulnerability.
- Updated the developer dependency and locked version of ipython to address a security vulnerability.
- Removed *jedi = "^0.17.2"* requirement since recent *ipython = "^7.17.3"* update appropriately addresses the dependency.

2.2.5 v0.18.0

This minor release updates the mimumum supported python to *python = "^3.8"*, addresses dependency vulnerabilities, and updates typing to conform with current mypy and pytest standards.

Updates / New Features

Python

- New minimum supported python changed to *python = "^3.8"*.

CI

- Updated CI unittests to reflect new minimum support *python = "^3.8"*.

Fixes

Dependency Versions

- Updated the locked versions of dependencies to reflect new minimum support `python = “^3.8”.

CHAPTER

THREE

DATAPROVIDER

An important part of any algorithm is the data it's working over and the data that it produces. An important part of working with large scales of data is where the data is stored and how it's accessed. The `smqtk_dataprovider` module contains interfaces and plugins for various core data structures, allowing plugin implementations to decide where and how the underlying raw data should be stored and accessed. This potentially allows algorithms to handle more data that would otherwise be feasible on a single machine.

3.1 DataProvider Structures

The following are the core data representation interfaces included in this package.

Note:

It is required that implementations have a common serialization format so that they may be stored or transported by other structures in a general way without caring what the specific implementation is. For this we require that all implementations be serializable via the `pickle` module functions.

3.1.1 DataElement

`class smqtk_dataprovider.DataElement`

Abstract interface for a byte data container.

The primary "value" of a `DataElement` is the byte content wrapped. Since this can technically change due to external forces, we cannot guarantee that an element is immutable. Thus `DataElement` instances are not considered generally hashable. Specific implementations may define a `__hash__` method if that implementation reflects a data source that guarantees immutability.

UUIDs should be cast-able to a string and maintain unique-ness after conversion.

`clean_temp()` → None

Clean any temporary files created by this element. This does nothing if no temporary files have been generated for this element yet.

`abstract content_type()` → str | None

Returns

Standard type/subtype string for this data element, or None if the content type is unknown.

Return type

str or None

classmethod from_uri(*uri: str*) → *DataElement*

Construct a new instance based on the given URI.

This function may not be implemented for all DataElement types.

Parameters
`uri` (*str*) – URI string to resolve into an element instance

Raises

- **NoUriResolutionError** – This element type does not implement URI resolution.
- **InvalidUriError** – This element type could not resolve the provided URI string.

Returns
New element instance of our type.

Return type
DataElement

abstract get_bytes() → bytes

Returns
Get the bytes for this data element.

Return type
bytes

abstract is_empty() → bool

Check if this element contains no bytes.

The intend of this method is to quickly check if there is any data behind this element, ideally without having to read all/any of the underlying data.

Returns
If this element contains 0 bytes.

Return type
bool

is_read_only() → bool

Returns
If this element can only be read from.

Return type
bool

md5() → str

Get the MD5 checksum of this element's binary content.

Returns
MD5 hex checksum of the data content.

Return type
str

abstract set_bytes(*b: bytes*) → None

Set bytes to this data element.

Not all implementations may support setting bytes (check `writable` method return).

This base abstract method should be called by sub-class implementations first. We check for mutability based on `writable()` method return.

Parameters
`b` (*bytes*) – bytes to set.

Raises
`ReadOnlyError` – This data element can only be read from / does not support writing.

`sha1()` → str

Get the SHA1 checksum of this element's binary content.

Returns
SHA1 hex checksum of the data content.

Return type
str

`sha512()` → str

Get the SHA512 checksum of this element's binary content.

Returns
SHA512 hex checksum of the data content.

Return type
str

`to_buffered_reader()` → BytesIO

Wrap this element's bytes in a `io.BufferedReader` instance for use as file-like object for reading.

As we use the `get_bytes` function, this element's bytes must safely fit in memory for this method to be usable.

Returns
New BufferedReader instance

Return type
io.BufferedReader

`uuid()` → Hashable

UUID for this data element.

This many take different forms from integers to strings to a uuid.UUID instance. This must return a hashable data type.

By default, this ends up being the hex stringification of the SHA1 hash of this data's bytes. Specific implementations may provide other UUIDs, however.

Returns
UUID value for this data element. This return value should be hashable.

Return type
collections.abc.Hashable

`abstract writable()` → bool

Returns
if this instance supports setting bytes.

Return type
bool

`write_temp`(*temp_dir: str | None = None*) → str

Write this data's bytes to a temporary file on disk, returning the path to the written file, whose extension is guessed based on this data's content type.

It is not guaranteed that the returned file path does not point to the original data, i.e. writing to the returned filepath may modify the original data.

NOTE:

The file path returned should not be explicitly removed by the user. Instead, the `clean_temp()` method should be called on this object.

Parameters

`temp_dir` (*`None or str`*) – Optional directory to write temporary file in, otherwise we use the platform default temporary files directory. If this is an empty string, we count it the same as having provided None.

Returns

Path to the temporary file

Return type

str

3.1.2 DataSet

`class smqtk_dataprovider.`**`DataSet`**

Abstract interface for data sets, that contain an arbitrary number of `DataElement` instances of arbitrary implementation type, keyed on `DataElement` UUID values.

This should only be used with DataElements whose byte content is expected not to change. If they do, then UUID keys may no longer represent the elements associated with them.

`abstract` **`add_data`**(**elems:* DataElement) → None

Add the given data element(s) instance to this data set.

NOTE: Implementing methods should check that input elements are in fact DataElement instances.

Parameters

`elems` (*`smqtk.representation.DataElement`*) – Data element(s) to add

`abstract` **`count`**() → int

Returns

The number of data elements in this set.

Return type

int

`abstract` **`get_data`**(*uuid: Hashable*) → *DataElement*

Get the data element the given uuid references, or raise an exception if the uuid does not reference any element in this set.

Raises

`KeyError` – If the given uuid does not refer to an element in this data set.

Parameters

`uuid` (*`collections.abc.Hashable`*) – The uuid of the element to retrieve.

Returns

The data element instance for the given uuid.

Return type

smqtk.representation.DataElement

abstract has_uuid(*uuid: Hashable*) → bool

Test if the given uuid refers to an element in this data set.

Parameters

uuid (*collections.abc.Hashable*) – Unique ID to test for inclusion. This should match the type that the set implementation expects or cares about.

Returns

True if the given uuid matches an element in this set, or False if it does not.

Return type

bool

abstract uuids() → Set[Hashable]

Returns

A new set of uuids represented in this data set.

Return type

set

3.1.3 KeyValueStore

class smqtk_dataprovider.**KeyValueStore**

Interface for general key/value storage.

Implementations may impose restrictions on what types keys or values may be due to backend used.

Data access and manipulation should be thread-safe.

abstract add(*key: Hashable*, *value: Any*) → *KeyValueStore*

Add a key-value pair to this store.

NOTE: **Implementing sub-classes should call this super-method. This super method should not be considered a critical section for thread safety unless ``is_read_only`` is not thread-safe.**

Parameters

- **key** (*Hashable*) – Key for the value. Must be hashable.
- **value** (*object*) – Python object to store.

Raises

ReadOnlyError – If this instance is marked as read-only.

Returns

Self.

Return type

KeyValueStore

abstract add_many(*d: Mapping[Hashable, Any]*) → *KeyValueStore*

Add multiple key-value pairs at a time into this store as represented in the provided dictionary *d*.

Parameters

d (*dict[Hashable, object]*) – Dictionary of key-value pairs to add to this store.

Raises

ReadOnlyError – If this instance is marked as read-only.

Returns

Self.

Return type
KeyValueStore

`abstract clear()` → *KeyValueStore*

Clear this key-value store.

NOTE: **Implementing sub-classes should call this super-method. This super method should not be considered a critical section for thread safety.**

Raises
`ReadOnlyError` – If this instance is marked as read-only.

Returns
Self.

Return type
KeyValueStore

`abstract count()` → int

Returns
The number of key-value relationships in this store.

Return type
int | long

`abstract get(`*key: ~typing.Hashable*, *default: ~typing.Any = <smqtk_dataprovider.interfaces.key_value_store.KeyValueStoreNoDefaultValueType object>*`)` → Any

Get the value for the given key.

NOTE: **Implementing sub-classes are responsible for raising a ``KeyError`` where appropriate.**

Parameters

- **`key`** – Key to get the value of.
- **`default`** – Optional default value if the given key is not present in this store. This may be any value except for the `NO_DEFAULT_VALUE` constant (custom anonymous class instance).

Raises
`KeyError` – The given key is not present in this store and no default value given.

Returns
Deserialized python object stored for the given key.

`get_many(`*keys: ~typing.Iterable[~typing.Hashable], default: ~typing.Any = <smqtk_dataprovider.interfaces.key_value_store.KeyValueStoreNoDefaultValueType object>*`)` → Iterable[Any]

Get the values for the given keys.

NOTE: **Implementing sub-classes are responsible for raising a ``KeyError`` where appropriate.**

Parameters

- **`keys`** (*`collections.abc.Iterable[Hashable]`*) – The keys for which associated values are requested.
- **`default`** (*`object`*) – Optional default value if a given key is not present in this store. This may be any value except for the `NO_DEFAULT_VALUE` constant (custom anonymous class instance).

Raises

`KeyError` – A given key is not present in this store and no default value given.

Returns

Iterable of deserialized python objects stored for the given keys in the order that the corresponding keys were provided.

Return type

collections.abc.Iterable

`abstract has`(*key: Hashable*) → bool

Check if this store has a value for the given key.

Parameters

`key` (*`Hashable`*) – Key to check for a value for.

Returns

If this store has a value for the given key.

Return type

bool

`abstract is_read_only`() → bool

Returns

True if this instance is read-only and False if it is not.

Return type

bool

`abstract keys`() → Iterator[Hashable]

Returns

Iterator over keys in this store.

Return type

collections.abc.Iterator[Hashable]

`abstract remove`(*key: Hashable*) → *KeyValueStore*

Remove a single key-value entry.

Parameters

`key` (*`Hashable`*) – Key to remove.

Raises

- `ReadOnlyError` – If this instance is marked as read-only.
- `KeyError` – The given key is not present in this store and no default value given.

Returns

Self.

Return type

KeyValueStore

`abstract remove_many`(*keys: Iterable[Hashable]*) → *KeyValueStore*

Remove multiple keys and associated values.

Parameters

`keys` (*`collections.abc.Iterable[Hashable]`*) – Iterable of keys to remove. If this is empty this method does nothing.

Raises

- **ReadOnlyError** – If this instance is marked as read-only.
- **KeyError** – The given key is not present in this store and no default value given. The store is not modified if any key is invalid.

Returns
Self.

Return type
KeyValueStore

values() → Iterator

Returns
Iterator over values in this store. Values are not guaranteed to be in any particular order.

Return type
collections.abc.Iterator

CHAPTER FOUR

INDICES AND TABLES

- genindex
- modindex
- search

INDEX