
opaque_keys Documentation

Release

Author

September 21, 2016

1	opaque_keys package	3
1.1	Subpackages	3
1.1.1	opaque_keys.edx package	3
	Submodules	3
	opaque_keys.edx.keys module	3
	opaque_keys.edx.locations module	5
	opaque_keys.edx.locator module	7
	Module contents	14
1.2	Module contents	14
2	Indices and tables	17
	Python Module Index	19

Contents:

opaque_keys package

1.1 Subpackages

1.1.1 opaque_keys.edx package

Submodules

opaque_keys.edx.keys module

; OpaqueKey abstract classes for edx-platform object types (courses, definitions, usages, and assets).

class `opaque_keys.edx.keys.AsideDefinitionKey` (*args, **kwargs)

Bases: `opaque_keys.edx.keys.DefinitionKey`

A definition key for an aside.

aside_type

Return the type of this aside.

definition_key

Return the DefinitionKey that this aside is decorating.

class `opaque_keys.edx.keys.AsideUsageKey` (*args, **kwargs)

Bases: `opaque_keys.edx.keys.UsageKey`

A usage key for an aside.

aside_type

Return the type of this aside.

usage_key

Return the UsageKey that this aside is decorating.

class `opaque_keys.edx.keys.AssetKey` (*args, **kwargs)

Bases: `opaque_keys.edx.keys.CourseObjectMixin`, `opaque_keys.OpaqueKey`

An `opaque_keys.OpaqueKey` identifying a course asset.

KEY_TYPE = 'asset_key'

asset_type

Return what type of asset this is.

path

Return the path for this asset.

class `opaque_keys.edx.keys.BlockTypeKey` (*args, **kwargs)

Bases: `opaque_keys.OpaqueKey`

A key class that encodes XBlock-family block types, including which family the block was loaded from.

KEY_TYPE = 'block_type'

block_family

Return the block-family identifier (the entry-point used to load that block family).

block_type

Return the block_type of this block (the key in the entry-point to load the block with).

class `opaque_keys.edx.keys.CourseKey` (*args, **kwargs)

Bases: `opaque_keys.OpaqueKey`

An `opaque_keys.OpaqueKey` identifying a particular Course object.

KEY_TYPE = 'course_key'

course

The name for this course.

In old-style IDs, it's the "course" in org/course/run

make_asset_key (asset_type, path)

Return an asset key, given the given the specified path.

This function should not actually create any new ids, but should simply return one that already exists.

make_usage_key (block_type, block_id)

Return a usage key, given the given the specified block_type and block_id.

This function should not actually create any new ids, but should simply return one that already exists.

org

The organization that this course belongs to.

run

The run for this course.

In old-style IDs, it's the "run" in org/course/run

class `opaque_keys.edx.keys.CourseObjectMixin`

Bases: `object`

An abstract `opaque_keys.OpaqueKey` mixin for keys that belong to courses.

course_key

Return the `CourseKey` for the course containing this usage.

map_into_course (course_key)

Return a new `UsageKey` or `AssetKey` representing this usage inside the course identified by the supplied `CourseKey`. It returns the same type as `self`

Args: `course_key` (`CourseKey`): The course to map this object into.

Returns: A new `CourseObjectMixin` instance.

class `opaque_keys.edx.keys.DefinitionKey` (*args, **kwargs)

Bases: `opaque_keys.OpaqueKey`

An `opaque_keys.OpaqueKey` identifying an XBlock definition.

KEY_TYPE = 'definition_key'

block_type

The XBlock type of this definition.

class `opaque_keys.edx.keys.UsageKey` (**args, **kwargs*)

Bases: `opaque_keys.edx.keys.CourseObjectMixin`, `opaque_keys.OpaqueKey`

An `opaque_keys.OpaqueKey` identifying an XBlock usage.

KEY_TYPE = 'usage_key'

block_id

The name of this usage.

block_type

The XBlock type of this usage.

definition_key

Return the `DefinitionKey` for the XBlock containing this usage.

class `opaque_keys.edx.keys.i4xEncoder` (*skipkeys=False, ensure_ascii=True, check_circular=True, allow_nan=True, sort_keys=False, indent=None, separators=None, encoding='utf-8', default=None*)

Bases: `json.encoder.JSONEncoder`

If provided as the cls to `json.dumps`, will serialize and Locations as i4x strings and other keys using the unicode strings.

default (*key*)

opaque_keys.edx.locations module

Deprecated OpaqueKey implementations used by XML and Mongo modulestores

class `opaque_keys.edx.locations.AssetLocation` (*org, course, run, category, name, revision=None, **kwargs*)

Bases: `opaque_keys.edx.locations.LocationBase`, `opaque_keys.edx.locator.AssetLocator`

Deprecated. Use `locator.AssetLocator`

DEPRECATED_TAG = 'c4x'

classmethod `from_deprecated_string` (*serialized*)

replace (***kwargs*)

Return: a new `AssetLocation` with specific `kwargs` replacing their corresponding values.

Using AssetLocator's replace function results in a mismatch of `__init__` args and `kwargs`. Replace tries to instantiate an `AssetLocation` object with `AssetLocator`'s args and `kwargs`.

class `opaque_keys.edx.locations.DeprecatedLocation` (*course_key, block_type, block_id, **kwargs*)

Bases: `opaque_keys.edx.locator.BlockUsageLocator`

The short-lived location:org+course+run+block_type+block_id syntax

CANONICAL_NAMESPACE = 'location'

URL_RE = `<_sre.SRE_Pattern object at 0x12a7ff0>`

URL_RE_SOURCE = `'\n (?P<org>[\w\-\~:]+)\n+(?P<course>[\w\-\~:]+)\n+(?P<run>[\w\-\~:]+)\n+(?P<block_type>[\w\-\~:]+)\n'`

class opaque_keys.edx.locations.**Location**(org, course, run, category, name, revision=None, **kwargs)

Bases: *opaque_keys.edx.locations.LocationBase, opaque_keys.edx.locator.BlockUsageLocator*

Deprecated. Use locator.BlockUsageLocator

DEPRECATED_TAG = 'i4x'

replace (**kwargs)

Return: a new *Location* with specific **kwargs** replacing their corresponding values.

Using BlockUsageLocator's replace function results in a mismatch of __init__ args and kwargs.

Replace tries to instantiate a Location object with BlockUsageLocator's args and kwargs.

class opaque_keys.edx.locations.**LocationBase**(org, course, run, category, name, revision=None, **kwargs)

Bases: object

Deprecated. Base class for *Location* and *AssetLocation*

DEPRECATED_TAG = None

classmethod clean (value)

Deprecated. See BlockUsageLocator.clean

classmethod clean_for_html (value)

Deprecated. See BlockUsageLocator.clean_for_html

classmethod clean_for_url_name (value)

Deprecated. See BlockUsageLocator.clean_for_url_name

classmethod clean_keeping_underscores (value)

Deprecated. See BlockUsageLocator.clean_keeping_underscores

classmethod from_deprecated_string (serialized)

Deprecated. Use locator.BlockUsageLocator.from_string().

classmethod from_string (serialized)

Deprecated. Use locator.BlockUsageLocator.from_string().

tag

Deprecated. Returns the deprecated tag for this Location.

class opaque_keys.edx.locations.**SlashSeparatedCourseKey**(org, course, run, **kwargs)

Bases: *opaque_keys.edx.locator.CourseLocator*

Deprecated. Use locator.CourseLocator

classmethod from_deprecated_string (serialized)

Deprecated. Use locator.CourseLocator.from_string

classmethod from_string (serialized)

Deprecated. Use locator.CourseLocator.from_string().

replace (**kwargs)

Return: a new *SlashSeparatedCourseKey* with specific **kwargs** replacing their corresponding values.

Using CourseLocator's replace function results in a mismatch of __init__ args and kwargs.

Replace tries to instantiate a SlashSeparatedCourseKey object with CourseLocator args and kwargs.

class `opaque_keys.edx.locations.i4xEncoder` (*args, **kwargs)
 Bases: `opaque_keys.edx.keys.i4xEncoder`
 Deprecated. Use `keys.i4xEncoder`

opaque_keys.edx.locator module

Identifier for course resources.

class `opaque_keys.edx.locator.AssetLocator` (course_key, block_type, block_id, **kwargs)
 Bases: `opaque_keys.edx.locator.BlockUsageLocator`, `opaque_keys.edx.keys.AssetKey`

An AssetKey implementation class.

ALLOWED_ID_RE = <_sre.SRE_Pattern object>

ASSET_URL_RE = <_sre.SRE_Pattern object>

CANONICAL_NAMESPACE = 'asset-v1'

DEPRECATED_ALLOWED_ID_RE = <_sre.SRE_Pattern object>

DEPRECATED_TAG = 'c4x'

asset_type

block_id

block_type

course_key

path

replace (**kwargs)

tag

Returns the deprecated tag for this Location.

to_deprecated_list_repr ()

Thumbnail locations are stored as lists [c4x, org, course, thumbnail, path, None] in contentstore.mongo. That should be the only use of this method, but the method is general enough to provide the pre-opaque Location fields as an array in the old order with the tag.

to_deprecated_string ()

Deprecated. Use `unicode(key)` instead.

class `opaque_keys.edx.locator.BlockLocatorBase` (*args, **kwargs)
 Bases: `opaque_keys.edx.locator.Locator`

Abstract base class for XBlock locators.

See subclasses for more detail, particularly `CourseLocator` and `BlockUsageLocator`.

ALLOWED_ID_RE = <_sre.SRE_Pattern object>

BLOCK_ALLOWED_ID_CHARS = '[\\w\\-\\.:%]'

BLOCK_PREFIX = 'block'

BRANCH_PREFIX = 'branch'

DEPRECATED_ALLOWED_ID_RE = <_sre.SRE_Pattern object>

URL_RE = <_sre.SRE_Pattern object at 0x118f490>

URL_RE_SOURCE = '\\n ((?P<org>[\\w\\-\\.:%]+)\\+(?P<course>[\\w\\-\\.:%]+)(\\+(?P<run>[\\w\\-\\.:%]+)?(\\+(?=.)\\|Z))?)?\\n (bran

classmethod `parse_url` (*string*)

If it can be parsed as a `version_guid` with no preceding `org + offering`, returns a dict with key `'version_guid'` and the value,

If it can be parsed as a `org + offering`, returns a dict with key `'id'` and optional keys `'branch'` and `'version_guid'`.

Raises: `InvalidKeyError`: if string cannot be parsed -or- string ends with a newline.

class `opaque_keys.edx.locator.BlockUsageLocator` (*course_key*, *block_type*, *block_id*,
***kwargs*)

Bases: `opaque_keys.edx.locator.BlockLocatorBase`, `opaque_keys.edx.keys.UsageKey`

Encodes a location.

Locations address modules (aka blocks) which are definitions situated in a course instance. Thus, a Location must identify the course and the occurrence of the defined element in the course. Courses can be a version of an offering, the current draft head, or the current production version.

Locators can contain both a version and a `org + course + run w/ branch`. The split mongo functions may raise errors if these conflict w/ the current db state (i.e., the course's `branch != the version_guid`)

Locations can express as urls as well as dictionaries. They consist of `package_identifier: course_guid | version_guid block : guid branch : string`

`BlockUsageLocators` also support deprecated `Location`-style formatting with the following mapping: `Location(org, course, run, category, name, revision)` is represented as a `BlockUsageLocator` with:

- `course_key` = a `CourseKey` comprised of (`org`, `course`, `run`, `branch=revision`)
- `block_type` = `category`
- `block_id` = `name`

CANONICAL_NAMESPACE = `'block-v1'`

CHECKED_INIT = `False`

DEPRECATED_INVALID_CHARS = `<_sre.SRE_Pattern object>`

DEPRECATED_INVALID_CHARS_NAME = `<_sre.SRE_Pattern object>`

DEPRECATED_INVALID_HTML_CHARS = `<_sre.SRE_Pattern object>`

DEPRECATED_TAG = `'i4x'`

DEPRECATED_URL_RE = `<_sre.SRE_Pattern object at 0x11d1f70>`

KEY_FIELDS = (`'course_key'`, `'block_type'`, `'block_id'`)

block_id = `None`

block_type = `None`

branch

Returns the branch for this object's `course_key`.

category

Deprecated. The ambiguously named field from `Location` which code expects to find. Equivalent to `block_type`.

classmethod `clean` (*value*)

Should only be called on deprecated-style values

Return value, made into a form legal for locations

classmethod `clean_for_html` (*value*)

Should only be called on deprecated-style values

Convert a string into a form that's safe for use in html ids, classes, urls, etc. Replaces all INVALID_HTML_CHARS with '_', collapses multiple '_' chars

classmethod `clean_for_url_name` (*value*)

Should only be called on deprecated-style values

Convert value into a format valid for location names (allows colons).

classmethod `clean_keeping_underscores` (*value*)

Should only be called on deprecated-style values

Return value, replacing INVALID_CHARS, but not collapsing multiple '_' chars. This for cleaning asset names, as the YouTube ID's may have underscores in them, and we need the transcript asset name to match. In the future we may want to change the behavior of `_clean`.

course

Returns the course for this object's `course_key`.

course_agnostic ()

We only care about the locator's version not its course. Returns a copy of itself without any course info.

Raises: ValueError if the block locator has no `version_guid`

course_key = None**definition_key**

Returns the definition key for this object. Undefined for Locators.

for_branch (*branch*)

Return a UsageLocator for the same block in a different branch of the course.

for_version (*version_guid*)

Return a UsageLocator for the same block in a different branch of the course.

html_id ()

Return an id which can be used on an html page as an id attr of an html element. It is currently also persisted by some clients to identify blocks.

To make compatible with old Location object functionality. I don't believe this behavior fits at this place, but I have no way to override. We should clearly define the purpose and restrictions of this (e.g., I'm assuming periods are fine).

is_fully_specified ()

Returns boolean; whether or not this object's `course_key` is fully specified.

classmethod `make_relative` (*course_locator*, *block_type*, *block_id*)

Return a new instance which has the given `block_id` in the given course :param `course_locator`: may be a BlockUsageLocator in the same snapshot

map_into_course (*course_key*)

Return a new instance which has the this `block_id` in the given course :param `course_key`: a CourseKey object representing the new course to map into

name

Deprecated. The ambiguously named field from Location which code expects to find. Equivalent to `block_id`.

offering

Deprecated. Use `course` and `run` independently.

org

Returns the org for this object's course_key.

replace (**kwargs)**revision**

Deprecated. The ambiguously named field from Location which code expects to find. Equivalent to branch.

run

Returns the run for this object's course_key.

to_deprecated_son (prefix='', tag='i4x')

Returns a SON object that represents this location

to_deprecated_string ()

Deprecated. Use unicode(key) instead.

version

Deprecated. The ambiguously named field from CourseLocation which code expects to find. Equivalent to version_guid.

version_agnostic ()

We don't care if the locator's version is not the current head; so, avoid version conflict by reducing info. Returns a copy of itself without any version info.

Raises: ValueError: if the block locator has no org, course, and run

version_guid

Returns the version guid for this object.

class opaque_keys.edx.locator.**CourseLocator** (org=None, course=None, run=None, branch=None, version_guid=None, deprecated=False, **kwargs)

Bases: *opaque_keys.edx.locator.BlockLocatorBase*, *opaque_keys.edx.keys.CourseKey*

Examples of valid CourseLocator specifications: CourseLocator(version_guid=ObjectId('519665f6223ebd6980884f2b'))

CourseLocator(org='mit.eecs', course='6.002x', run='T2_2014') CourseLocator(org='mit.eecs', course='6002x', run='fall_2014' branch = 'published')

CourseLocator.from_string('course-v1:version@519665f6223ebd6980884f2b')

CourseLocator.from_string('course-v1:mit.eecs+6002x') CourseLocator.from_string('course-v1:mit.eecs+6002x+branch@published')

CourseLocator.from_string('course-v1:mit.eecs+6002x+branch@published+version@519665f6223ebd6980884f2b')

Should have at least a specific org, course, and run with optional 'branch', or version_guid (which points to a specific version). Can contain both in which case the persistence layer may raise exceptions if the given version != the current such version of the course.

CANONICAL_NAMESPACE = 'course-v1'

CHECKED_INIT = False

INVALID_CHARS_DEPRECATED = <_sre.SRE_Pattern object>

KEY_FIELDS = ('org', 'course', 'run', 'branch', 'version_guid')

branch**course****course_agnostic** ()

We only care about the locator's version not its course. Returns a copy of itself without any course info.

Raises: ValueError: if the block locator has no version_guid

for_branch (*branch*)

Return a new CourseLocator for another branch of the same course (also version agnostic)

for_version (*version_guid*)

Return a new CourseLocator for another version of the same course and branch. Usually used when the head is updated (and thus the course x branch now points to this version)

html_id ()

Return an id which can be used on an html page as an id attr of an html element.

To make compatible with old Location object functionality. I don't believe this behavior fits at this place, but I have no way to override. We should clearly define the purpose and restrictions of this (e.g., I'm assuming periods are fine).

make_asset_key (*asset_type, path*)**make_usage_key** (*block_type, block_id*)**make_usage_key_from_deprecated_string** (*location_url*)

Deprecated mechanism for creating a UsageKey given a CourseKey and a serialized Location.

NOTE: this prejudicially takes the tag, org, and course from the url not self.

Raises: InvalidKeyError: if the url does not parse

offering

Deprecated. Use course and run independently.

org**run****to_deprecated_string** ()

Deprecated. Use unicode(key) instead.

version

Deprecated. The ambiguously named field from CourseLocation which code expects to find. Equivalent to version_guid.

version_agnostic ()

We don't care if the locator's version is not the current head; so, avoid version conflict by reducing info. Returns a copy of itself without any version info.

Raises: ValueError: if the block locator has no org & course, run

version_guid

class opaque_keys.edx.locator.**DefinitionLocator** (*block_type, definition_id, deprecated=False*)

Bases: *opaque_keys.edx.locator.Locator, opaque_keys.edx.keys.DefinitionKey*

Container for how to locate a description (the course-independent content).

CANONICAL_NAMESPACE = 'def-v1'

CHECKED_INIT = False

KEY_FIELDS = ('definition_id', 'block_type')

URL_RE = <_sre.SRE_Pattern object>

block_type = None

definition_id = None

version ()

Returns the ObjectId referencing this specific location.

class `opaque_keys.edx.locator.LibraryLocator` (*org=None, library=None, branch=None, version_guid=None, **kwargs*)
Bases: `opaque_keys.edx.locator.BlockLocatorBase`, `opaque_keys.edx.keys.CourseKey`

Locates a library. Libraries are XBlock structures with a 'library' block at their root.

Libraries are treated analogously to courses for now. Once opaque keys are better supported, they will no longer have the 'run' property, and may no longer conform to CourseKey but rather some more general key type.

Examples of valid LibraryLocator specifications: `LibraryLocator(version_guid=ObjectId('519665f6223ebd6980884f2b'))`
`LibraryLocator(org='UniX', library='PhysicsProbs')` `LibraryLocator.from_string('library-v1:UniX+PhysicsProbs')`

`version_guid` is optional.

The constructor accepts 'course' as a deprecated alias for the 'library' attribute.

`branch` is optional.

CANONICAL_NAMESPACE = 'library-v1'

CHECKED_INIT = False

KEY_FIELDS = ('org', 'library', 'branch', 'version_guid')

RUN = 'library'

branch

course

Deprecated. Return a 'course' for compatibility with CourseLocator.

course_agnostic()

We only care about the locator's version not its library. Returns a copy of itself without any library info.

Raises: ValueError: if the block locator has no `version_guid`

for_branch (*branch*)

Return a new CourseLocator for another branch of the same library (also version agnostic)

for_version (*version_guid*)

Return a new LibraryLocator for another version of the same library and branch. Usually used when the head is updated (and thus the library x branch now points to this version)

html_id()

Return an id which can be used on an html page as an id attr of an html element.

library

make_asset_key (*asset_type, path*)

make_usage_key (*block_type, block_id*)

org

run

Deprecated. Return a 'run' for compatibility with CourseLocator.

version

Deprecated. The ambiguously named field from CourseLocation which code expects to find. Equivalent to `version_guid`.

version_agnostic()

We don't care if the locator's version is not the current head; so, avoid version conflict by reducing info. Returns a copy of itself without any version info.

Raises: ValueError: if the block locator has no `org` & `course`, `run`

version_guid

class opaque_keys.edx.locator.**LibraryUsageLocator** (*library_key*, *block_type*, *block_id*,
**kwargs)

Bases: *opaque_keys.edx.locator.BlockUsageLocator*

Just like BlockUsageLocator, but this points to a block stored in a library, not a course.

CANONICAL_NAMESPACE = 'lib-block-v1'

KEY_FIELDS = ('library_key', 'block_type', 'block_id')

block_id = None

block_type = None

course_key

To enable compatibility with BlockUsageLocator, we provide a read-only course_key property.

for_branch (*branch*)

Return a UsageLocator for the same block in a different branch of the library.

for_version (*version_guid*)

Return a UsageLocator for the same block in a different version of the library.

library_key = None

replace (**kwargs)

run

Returns the run for this object's library_key.

to_deprecated_bson (*prefix*=' ', *tag*='i4x')

Disable some deprecated methods of our parent class.

version_agnostic ()

We don't care if the locator's version is not the current head; so, avoid version conflict by reducing info. Returns a copy of itself without any version info.

Raises: ValueError: if the block locator has no org, course, and run

class opaque_keys.edx.locator.**LocalId** (*block_id=None*)

Bases: object

Class for local ids for non-persisted xblocks (which can have hardcoded block_ids if necessary)

class opaque_keys.edx.locator.**Locator** (*args, **kwargs)

Bases: *opaque_keys.OpaqueKey*

A locator identifies a course resource.

Locator is an abstract base class: do not instantiate

ALLOWED_ID_CHARS = '[\\w\\-\\.:]'

BLOCK_TYPE_PREFIX = 'type'

DEPRECATED_ALLOWED_ID_CHARS = '[\\w\\-\\.:%]'

VERSION_PREFIX = 'version'

classmethod **as_object_id** (*value*)

Attempts to cast value as a bson.objectid.ObjectId.

Raises: ValueError: if casting fails

version

Returns the ObjectId referencing this specific location.

Raises: `InvalidKeyError`: if the instance doesn't have a complete enough specification.

class `opaque_keys.edx.locator.VersionTree` (*locator, tree_dict=None*)
Bases: `object`

Holds trees of Locators to represent version histories.

Module contents

1.2 Module contents

Defines the `OpaqueKey` class, to be used as the base-class for implementing pluggable `OpaqueKeys`.

These keys are designed to provide a limited, forward-evolveable interface to an application, while concealing the particulars of the serialization formats, and allowing new serialization formats to be installed transparently.

exception `opaque_keys.InvalidKeyError` (*key_class, serialized*)
Bases: `exceptions.Exception`

Raised to indicated that a serialized key isn't valid (wasn't able to be parsed by any available providers).

class `opaque_keys.OpaqueKey` (**args, **kwargs*)
Bases: `object`

A base-class for implementing pluggable opaque keys. Individual key subclasses identify particular types of resources, without specifying the actual form of the key (or its serialization).

There are two levels of expected subclasses: Key type definitions, and key implementations

```
OpaqueKey
 |
Key type
 |
Key implementation
```

The key type base class must define the class property `KEY_TYPE`, which identifies which `entry_point` namespace the keys implementations should be registered with.

The `KeyImplementation` classes must define the following:

CANONICAL_NAMESPACE Identifies the key namespace for the particular key implementation (when serializing). Key implementations must be registered using the `CANONICAL_NAMESPACE` as their `entry_point` name, but can also be registered with other names for backwards compatibility.

KEY_FIELDS A list of attribute names that will be used to establish object identity. Key implementation instances will compare equal iff all of their `KEY_FIELDS` match, and will not compare equal to instances of different `KeyImplementation` classes (even if the `KEY_FIELDS` match). These fields must be hashable.

`_to_string` Serialize the key into a unicode object. This should not include the namespace prefix (`CANONICAL_NAMESPACE`).

`_from_string` Construct an instance of this `OpaqueKey` from a unicode object. The namespace will already have been parsed.

`OpaqueKeys` will not have optional constructor parameters (due to the implementation of `KEY_FIELDS`), by default. However, an implementation class can provide a default, as long as it passes that default to a call to `super().__init__`. If the `KeyImplementation` sets the class attribute `CHECKED_INIT` to `False`, then the `OpaqueKey` base class constructor will not validate any of the `KEY_FIELDS` arguments, and will instead just expect all `KEY_FIELDS` to be passed as `kwargs`.

`OpaqueKey` objects are immutable.

Serialization of an *OpaqueKey* is performed by using the `unicode()` builtin. Deserialization is performed by the `from_string()` method.

CANONICAL_NAMESPACE = None

CHECKED_INIT = True

KEY_FIELDS = []

LOADED_DRIVERS = defaultdict(None, {})

NAMESPACE_SEPARATOR = u':'

deprecated

classmethod `from_string` (*serialized*)

Return a *OpaqueKey* object deserialized from the *serialized* argument. This object will be an instance of a subclass of the *cls* argument.

Args: *serialized*: A stringified form of a *OpaqueKey*

classmethod `get_namespace_plugin` (*namespace*)

Return the registered *OpaqueKey* subclass of *cls* for the supplied namespace

replace (***kwargs*)

Return: a new *OpaqueKey* with **KEY_FIELDS** specified in **kwargs** replaced their corresponding values. Deprecation value is also preserved.

Subclasses should override this if they have required properties that aren't included in their **KEY_FIELDS**.

classmethod `set_deprecated_fallback` (*fallback*)

Register a deprecated fallback class for this class to revert to.

class `opaque_keys.OpaqueKeyMetaclass`

Bases: `abc.ABCMeta`

Metaclass for *OpaqueKey*. Sets the default value for the values in **KEY_FIELDS** to None.

Indices and tables

- `genindex`
- `modindex`
- `search`

O

`opaque_keys`, 14
`opaque_keys.edx`, 14
`opaque_keys.edx.keys`, 3
`opaque_keys.edx.locations`, 5
`opaque_keys.edx.locator`, 7

A

ALLOWED_ID_CHARS
(opaque_keys.edx.locator.Locator attribute), 13

ALLOWED_ID_RE (opaque_keys.edx.locator.AssetLocator attribute), 7

ALLOWED_ID_RE (opaque_keys.edx.locator.BlockLocatorBase attribute), 7

as_object_id() (opaque_keys.edx.locator.Locator class method), 13

aside_type (opaque_keys.edx.keys.AsideDefinitionKey attribute), 3

aside_type (opaque_keys.edx.keys.AsideUsageKey attribute), 3

AsideDefinitionKey (class in opaque_keys.edx.keys), 3

AsideUsageKey (class in opaque_keys.edx.keys), 3

asset_type (opaque_keys.edx.keys.AssetIdKey attribute), 3

asset_type (opaque_keys.edx.locator.AssetLocator attribute), 7

ASSET_URL_RE (opaque_keys.edx.locator.AssetLocator attribute), 7

AssetKey (class in opaque_keys.edx.keys), 3

AssetLocation (class in opaque_keys.edx.locations), 5

AssetLocator (class in opaque_keys.edx.locator), 7

B

BLOCK_ALLOWED_ID_CHARS
(opaque_keys.edx.locator.BlockLocatorBase attribute), 7

block_family (opaque_keys.edx.keys.BlockTypeKey attribute), 4

block_id (opaque_keys.edx.keys.UsageKey attribute), 5

block_id (opaque_keys.edx.locator.AssetLocator attribute), 7

block_id (opaque_keys.edx.locator.BlockUsageLocator attribute), 8

block_id (opaque_keys.edx.locator.LibraryUsageLocator attribute), 13

BLOCK_PREFIX (opaque_keys.edx.locator.BlockLocatorBase attribute), 7

block_type (opaque_keys.edx.keys.BlockTypeKey

attribute), 4

block_type (opaque_keys.edx.keys.DefinitionKey attribute), 4

block_type (opaque_keys.edx.keys.UsageKey attribute), 5

block_type (opaque_keys.edx.locator.AssetLocator attribute), 7

block_type (opaque_keys.edx.locator.BlockUsageLocator attribute), 8

block_type (opaque_keys.edx.locator.DefinitionLocator attribute), 11

block_type (opaque_keys.edx.locator.LibraryUsageLocator attribute), 13

BLOCK_TYPE_PREFIX
(opaque_keys.edx.locator.Locator attribute), 13

BlockLocatorBase (class in opaque_keys.edx.locator), 7

BlockTypeKey (class in opaque_keys.edx.keys), 3

BlockUsageLocator (class in opaque_keys.edx.locator), 8

branch (opaque_keys.edx.locator.BlockUsageLocator attribute), 8

branch (opaque_keys.edx.locator.CourseLocator attribute), 10

branch (opaque_keys.edx.locator.LibraryLocator attribute), 12

BRANCH_PREFIX (opaque_keys.edx.locator.BlockLocatorBase attribute), 7

C

CANONICAL_NAMESPACE
(opaque_keys.edx.locations.DeprecatedLocation attribute), 5

CANONICAL_NAMESPACE
(opaque_keys.edx.locator.AssetLocator attribute), 7

CANONICAL_NAMESPACE
(opaque_keys.edx.locator.BlockUsageLocator attribute), 8

CANONICAL_NAMESPACE
(opaque_keys.edx.locator.CourseLocator attribute), 10

CANONICAL_NAMESPACE (opaque_keys.edx.locator.DefinitionLocator attribute), 11

CANONICAL_NAMESPACE (opaque_keys.edx.locator.LibraryLocator attribute), 12

CANONICAL_NAMESPACE (opaque_keys.edx.locator.LibraryUsageLocator attribute), 13

CANONICAL_NAMESPACE (opaque_keys.OpaqueKey attribute), 15

category (opaque_keys.edx.locator.BlockUsageLocator attribute), 8

CHECKED_INIT (opaque_keys.edx.locator.BlockUsageLocator attribute), 8

CHECKED_INIT (opaque_keys.edx.locator.CourseLocator attribute), 10

CHECKED_INIT (opaque_keys.edx.locator.DefinitionLocator attribute), 11

CHECKED_INIT (opaque_keys.edx.locator.LibraryLocator attribute), 12

CHECKED_INIT (opaque_keys.OpaqueKey attribute), 15

clean() (opaque_keys.edx.locations.LocationBase class method), 6

clean() (opaque_keys.edx.locator.BlockUsageLocator class method), 8

clean_for_html() (opaque_keys.edx.locations.LocationBase class method), 6

clean_for_html() (opaque_keys.edx.locator.BlockUsageLocator class method), 8

clean_for_url_name() (opaque_keys.edx.locations.LocationBase class method), 6

clean_for_url_name() (opaque_keys.edx.locator.BlockUsageLocator class method), 9

clean_keeping_underscores() (opaque_keys.edx.locations.LocationBase class method), 6

clean_keeping_underscores() (opaque_keys.edx.locator.BlockUsageLocator class method), 9

course (opaque_keys.edx.keys.CourseKey attribute), 4

course (opaque_keys.edx.locator.BlockUsageLocator attribute), 9

course (opaque_keys.edx.locator.CourseLocator attribute), 10

course (opaque_keys.edx.locator.LibraryLocator attribute), 12

course_agnostic() (opaque_keys.edx.locator.BlockUsageLocator method), 9

course_agnostic() (opaque_keys.edx.locator.CourseLocator method), 10

course_agnostic() (opaque_keys.edx.locator.LibraryLocator method), 12

course_key (opaque_keys.edx.keys.CourseObjectMixin attribute), 4

course_key (opaque_keys.edx.locator.AssetLocator attribute), 7

course_key (opaque_keys.edx.locator.BlockUsageLocator attribute), 9

course_key (opaque_keys.edx.locator.LibraryUsageLocator attribute), 13

CourseKey (class in opaque_keys.edx.keys), 4

CourseLocator (class in opaque_keys.edx.locator), 10

CourseObjectMixin (class in opaque_keys.edx.keys), 4

D

default() (opaque_keys.edx.keys.i4xEncoder method), 5

definition_id (opaque_keys.edx.locator.DefinitionLocator attribute), 11

definition_key (opaque_keys.edx.keys.AsideDefinitionKey attribute), 3

definition_key (opaque_keys.edx.keys.UsageKey attribute), 5

definition_key (opaque_keys.edx.locator.BlockUsageLocator attribute), 9

DefinitionKey (class in opaque_keys.edx.keys), 4

DefinitionLocator (class in opaque_keys.edx.locator), 11

deprecated (opaque_keys.OpaqueKey attribute), 15

DEPRECATED_ALLOWED_ID_CHARS (opaque_keys.edx.locator.Locator attribute), 13

DEPRECATED_ALLOWED_ID_RE (opaque_keys.edx.locator.AssetLocator attribute), 7

DEPRECATED_ALLOWED_ID_RE (opaque_keys.edx.locator.BlockLocatorBase attribute), 7

DEPRECATED_INVALID_CHARS (opaque_keys.edx.locator.BlockUsageLocator attribute), 8

DEPRECATED_INVALID_CHARS_NAME (opaque_keys.edx.locator.BlockUsageLocator attribute), 8

DEPRECATED_INVALID_HTML_CHARS (opaque_keys.edx.locator.BlockUsageLocator attribute), 8

DEPRECATED_TAG (opaque_keys.edx.locations.AssetLocation attribute), 5

DEPRECATED_TAG (opaque_keys.edx.locations.Location attribute), 6

DEPRECATED_TAG (opaque_keys.edx.locations.LocationBase attribute), 6

DEPRECATED_TAG (opaque_keys.edx.locator.AssetLocator attribute), 7

DEPRECATED_TAG (opaque_keys.edx.locator.BlockUsageLocator attribute), 8

DEPRECATED_URL_RE (opaque_keys.edx.locator.BlockUsageLocator attribute), 7

- attribute), 8
- DeprecatedLocation (class in opaque_keys.edx.locations), 5
- ## F
- for_branch() (opaque_keys.edx.locator.BlockUsageLocator method), 9
- for_branch() (opaque_keys.edx.locator.CourseLocator method), 10
- for_branch() (opaque_keys.edx.locator.LibraryLocator method), 12
- for_branch() (opaque_keys.edx.locator.LibraryUsageLocator method), 13
- for_version() (opaque_keys.edx.locator.BlockUsageLocator method), 9
- for_version() (opaque_keys.edx.locator.CourseLocator method), 11
- for_version() (opaque_keys.edx.locator.LibraryLocator method), 12
- for_version() (opaque_keys.edx.locator.LibraryUsageLocator method), 13
- from_deprecated_string() (opaque_keys.edx.locations.AssetLocation class method), 5
- from_deprecated_string() (opaque_keys.edx.locations.LocationBase class method), 6
- from_deprecated_string() (opaque_keys.edx.locations.SlashSeparatedCourseKey class method), 6
- from_string() (opaque_keys.edx.locations.LocationBase class method), 6
- from_string() (opaque_keys.edx.locations.SlashSeparatedCourseKey class method), 6
- from_string() (opaque_keys.OpaqueKey class method), 15
- ## G
- get_namespace_plugin() (opaque_keys.OpaqueKey class method), 15
- ## H
- html_id() (opaque_keys.edx.locator.BlockUsageLocator method), 9
- html_id() (opaque_keys.edx.locator.CourseLocator method), 11
- html_id() (opaque_keys.edx.locator.LibraryLocator method), 12
- ## I
- i4xEncoder (class in opaque_keys.edx.keys), 5
- i4xEncoder (class in opaque_keys.edx.locations), 6
- INVALID_CHARS_DEPRECATED (opaque_keys.edx.locator.CourseLocator attribute), 10
- InvalidKeyError, 14
- is_fully_specified() (opaque_keys.edx.locator.BlockUsageLocator method), 9
- ## K
- KEY_FIELDS (opaque_keys.edx.locator.BlockUsageLocator attribute), 8
- KEY_FIELDS (opaque_keys.edx.locator.CourseLocator attribute), 10
- KEY_FIELDS (opaque_keys.edx.locator.DefinitionLocator attribute), 11
- KEY_FIELDS (opaque_keys.edx.locator.LibraryLocator attribute), 12
- KEY_FIELDS (opaque_keys.edx.locator.LibraryUsageLocator attribute), 13
- KEY_FIELDS (opaque_keys.OpaqueKey attribute), 15
- KEY_TYPE (opaque_keys.edx.keys.AssetKey attribute), 3
- KEY_TYPE (opaque_keys.edx.keys.BlockTypeKey attribute), 4
- KEY_TYPE (opaque_keys.edx.keys.CourseKey attribute), 4
- KEY_TYPE (opaque_keys.edx.keys.DefinitionKey attribute), 4
- KEY_TYPE (opaque_keys.edx.keys.UsageKey attribute), 5
- ## L
- library (opaque_keys.edx.locator.LibraryLocator attribute), 12
- library_key (opaque_keys.edx.locator.LibraryUsageLocator attribute), 13
- LibraryLocator (class in opaque_keys.edx.locator), 12
- LibraryUsageLocator (class in opaque_keys.edx.locator), 13
- LOADED_DRIVERS (opaque_keys.OpaqueKey attribute), 15
- LocalId (class in opaque_keys.edx.locator), 13
- Location (class in opaque_keys.edx.locations), 5
- LocationBase (class in opaque_keys.edx.locations), 6
- Locator (class in opaque_keys.edx.locator), 13
- ## M
- make_asset_key() (opaque_keys.edx.keys.CourseKey method), 4
- make_asset_key() (opaque_keys.edx.locator.CourseLocator method), 11
- make_asset_key() (opaque_keys.edx.locator.LibraryLocator method), 12
- make_relative() (opaque_keys.edx.locator.BlockUsageLocator class method), 9

make_usage_key() (opaque_keys.edx.keys.CourseKey replace() (opaque_keys.edx.locator.AssetLocator method), 4 method), 7
 make_usage_key() (opaque_keys.edx.locator.CourseLocator replace() (opaque_keys.edx.locator.BlockUsageLocator method), 11 method), 10
 make_usage_key() (opaque_keys.edx.locator.LibraryLocator replace() (opaque_keys.edx.locator.LibraryUsageLocator method), 12 method), 13
 make_usage_key_from_deprecated_string() (opaque_keys.edx.locator.CourseLocator replace() (opaque_keys.OpaqueKey method), 15 method), 11 revision (opaque_keys.edx.locator.BlockUsageLocator attribute), 10
 map_into_course() (opaque_keys.edx.keys.CourseObjectMixin (opaque_keys.edx.keys.CourseKey attribute), 4 method), 4 run (opaque_keys.edx.locator.BlockUsageLocator attribute), 10
 map_into_course() (opaque_keys.edx.locator.BlockUsageLocator (opaque_keys.edx.locator.CourseLocator attribute), 11 method), 9 RUN (opaque_keys.edx.locator.LibraryLocator attribute), 12
 run (opaque_keys.edx.locator.LibraryLocator attribute), 12
 run (opaque_keys.edx.locator.LibraryUsageLocator attribute), 13

N

name (opaque_keys.edx.locator.BlockUsageLocator attribute), 9
 NAMESPACE_SEPARATOR (opaque_keys.OpaqueKey attribute), 15

O

offering (opaque_keys.edx.locator.BlockUsageLocator attribute), 9
 offering (opaque_keys.edx.locator.CourseLocator attribute), 11
 opaque_keys (module), 14
 opaque_keys.edx (module), 14
 opaque_keys.edx.keys (module), 3
 opaque_keys.edx.locations (module), 5
 opaque_keys.edx.locator (module), 7
 OpaqueKey (class in opaque_keys), 14
 OpaqueKeyMetaClass (class in opaque_keys), 15
 org (opaque_keys.edx.keys.CourseKey attribute), 4
 org (opaque_keys.edx.locator.BlockUsageLocator attribute), 9
 org (opaque_keys.edx.locator.CourseLocator attribute), 11
 org (opaque_keys.edx.locator.LibraryLocator attribute), 12

P

parse_url() (opaque_keys.edx.locator.BlockLocatorBase class method), 7
 path (opaque_keys.edx.keys.AssetKey attribute), 3
 path (opaque_keys.edx.locator.AssetLocator attribute), 7

R

replace() (opaque_keys.edx.locations.AssetPosition method), 5
 replace() (opaque_keys.edx.locations.Location method), 6
 replace() (opaque_keys.edx.locations.SlashSeparatedCourseKey method), 6

S

set_deprecated_fallback() (opaque_keys.OpaqueKey class method), 15
 SlashSeparatedCourseKey (class in opaque_keys.edx.locations), 6

T

tag (opaque_keys.edx.locations.LocationBase attribute), 6
 tag (opaque_keys.edx.locator.AssetLocator attribute), 7
 to_deprecated_list_repr() (opaque_keys.edx.locator.AssetLocator method), 7
 to_deprecated_son() (opaque_keys.edx.locator.BlockUsageLocator method), 10
 to_deprecated_son() (opaque_keys.edx.locator.LibraryUsageLocator method), 13
 to_deprecated_string() (opaque_keys.edx.locator.AssetLocator method), 7
 to_deprecated_string() (opaque_keys.edx.locator.BlockUsageLocator method), 10
 to_deprecated_string() (opaque_keys.edx.locator.CourseLocator method), 11

U

URL_RE (opaque_keys.edx.locations.DeprecatedLocation attribute), 5
 URL_RE (opaque_keys.edx.locator.BlockLocatorBase attribute), 7
 URL_RE (opaque_keys.edx.locator.DefinitionLocator attribute), 11
 URL_RE_SOURCE (opaque_keys.edx.locations.DeprecatedLocation attribute), 5

URL_RE_SOURCE (opaque_keys.edx.locator.BlockLocatorBase attribute), 7

usage_key (opaque_keys.edx.keys.AsideUsageKey attribute), 3

UsageKey (class in opaque_keys.edx.keys), 5

V

version (opaque_keys.edx.locator.BlockUsageLocator attribute), 10

version (opaque_keys.edx.locator.CourseLocator attribute), 11

version (opaque_keys.edx.locator.LibraryLocator attribute), 12

version (opaque_keys.edx.locator.Locator attribute), 13

version() (opaque_keys.edx.locator.DefinitionLocator method), 11

version_agnostic() (opaque_keys.edx.locator.BlockUsageLocator method), 10

version_agnostic() (opaque_keys.edx.locator.CourseLocator method), 11

version_agnostic() (opaque_keys.edx.locator.LibraryLocator method), 12

version_agnostic() (opaque_keys.edx.locator.LibraryUsageLocator method), 13

version_guid (opaque_keys.edx.locator.BlockUsageLocator attribute), 10

version_guid (opaque_keys.edx.locator.CourseLocator attribute), 11

version_guid (opaque_keys.edx.locator.LibraryLocator attribute), 13

VERSION_PREFIX (opaque_keys.edx.locator.Locator attribute), 13

VersionTree (class in opaque_keys.edx.locator), 14