progress-interface

Release 0.1

Jared Lumpe

Nov 06, 2021

CONTENTS:

CHAPTER

ONE

API

1.1 progress_interface.base

Core functionality.

class progress_interface.base.**AbstractProgressMonitor**

Bases: abc.ABC

Abstract base class for an object which tracks the progress of a long-running task and possibly displays it to the user.

Concrete subclasses must implement the *moveto()* and *create()* methods along with the *n*, *total*, and *closed* attributes. They may also optionally override *increment()* and *close()*.

n

Number of completed iterations. Do not modify directly, use the *increment()* and *moveto()* methods instead.

Type int

total

Expected total number of iterations.

Type int

closed

Whether the monitor has been closed/completed.

Type int

close()

Stop tracking/displaying progress and perform whatever cleanup is necessary.

classmethod config(***kw*)

Get a ProgressConfig which creates instances of the class with the given default settings..

Keyword arguments are passed on to *create()*.

Return type *progress_interface.base.ProgressConfig*

abstract classmethod create(*total*, ***, *initial=0*, *desc=None*, *file=None*, ***kw*)

Factory function with standardized signature to create instances of the class.

Parameters

- **total** (*int*) – Total number of iterations to completion.
- **initial** (*int*) – Initial value of *n*.
- **desc** (*Optional[str]*) – Description to display to the user, if applicable.

- **file** (*Optional[TextIO]*) – File-like object to write text output to, if applicable. Defaults to `sys.stderr`.
- ****kw** – Additional options depending on the subclass.

Return type *progress_interface.base.AbstractProgressMonitor*

increment(*delta=1*)
Increment the position of the monitor by the given value.

Parameters **delta** (*int*) –

abstract moveto(*n*)
Set the monitor's position to the given value.

Parameters **n** (*int*) –

class progress_interface.base.**NullProgressMonitor**
Bases: *progress_interface.base.AbstractProgressMonitor*

Progress monitor which does nothing.

close()
Stop tracking/displaying progress and perform whatever cleanup is necessary.

increment(*delta=1*)
Increment the position of the monitor by the given value.

Parameters **delta** (*int*) –

moveto(*n*)
Set the monitor's position to the given value.

Parameters **n** (*int*) –

class progress_interface.base.**ProgressConfig**
Bases: `object`

Configuration settings used to create new progress monitor instances.

This allows callers to pass the desired progress monitor type and other settings to a function without needing to know the total length and other details about the task, which can be determined within the function body.

factory
The *AbstractProgressMonitor.create()* method of a concrete progress monitor type, or another factory with the same signature which returns a progress monitor instance.

Type Callable[[int], *progress_interface.base.AbstractProgressMonitor*]

kw
Keyword arguments to pass to `factory`.

Type Dict[str, Any]

__init__(*factory*, *kw*)

Parameters

- **factory** (*Callable[[int], progress_interface.base.AbstractProgressMonitor]*) –
- **kw** (*Dict[str, Any]*) –

create(*total*, ***kw*)
Create a progress monitor instance by calling the factory function with the stored keyword arguments.

The signature of this function is identical to *AbstractProgressMonitor.create()*.

Parameters **total** (*int*) –

Return type *progress_interface.base.AbstractProgressMonitor*

update(**args*, ***kw*)
Update keyword arguments and return a new instance.

Parameters **args** (*Mapping[str, Any]*) –

Return type *progress_interface.base.ProgressConfig*

class progress_interface.base.**ProgressIterator**
Bases: Iterator

__init__(*iterable*, *monitor*)

Parameters

- **iterable** (*Iterable*) –
- **monitor** (progress_interface.base.AbstractProgressMonitor) –

progress_interface.base.**default_config**()
Get the default *ProgressConfig* instance to use.

Currently attempts to use *TqdmProgressMonitor*, if tqdm is not importable prints a warning and uses *NullProgressMonitor*.

Return type *progress_interface.base.ProgressConfig*

progress_interface.base.**get_progress**(*arg*, *total*, *initial=0*, ***kw*)
Create a progress monitor instance.

See *progress_config()* for description of allowed types/values for the argument.

Parameters

- **arg** (*Optional[Union[*progress_interface.base.ProgressConfig, *str, bool, type, Callable[[int],* progress_interface.base.AbstractProgressMonitor*]]]*) –
- **total** (*int*) – Number of expected iterations.
- **initial** (*int*) – Initial position of progress monitor.
- ****kw** – Additional keyword arguments to pass to progress monitor class or factory function defined by arg..

Return type *progress_interface.base.AbstractProgressMonitor*

progress_interface.base.**iter_progress**(*iterable*, *progress=True*, *total=None*, ***kw*)
Display a progress monitor while iterating over an object.

The returned iterator object can also be used as a context manager to ensure that the progress monitor is closed properly even if iteration does not finish.

Parameters

- **itarable** – Iterable object.
- **progress** (*Optional[Union[*progress_interface.base.ProgressConfig, *str, bool, type, Callable[[int],* progress_interface.base.AbstractProgressMonitor*]]]*) – Passed to *get_progress()*.

- **total** (*Optional[int]*) – Total number of expected iterations. Defaults to `len(iterable)`.
- ****kw** – Additional keyword arguments to pass to progress monitor factory.
- **iterable** (*Iterable*) –

Returns Iterator over values in `iterable` which advances a progress monitor.

Return type *ProgressIterator*

`progress_interface.base.`**progress_config**(*arg*, ***kw*)

Get a `ProgressConfig` instance from a variety argument types.

Accepts the following types/values for the argument:

- *ProgressConfig*
- `None` - uses `NullProgressBar`.
- `True` - uses value returned by *default_config()*.
- `False` - same as `None`.
- `str` key - Looks up progress bar class/factory function in *REGISTRY*.
- *AbstractProgressMonitor* subclass
- `Callable` - factory function. Must have same signature as *AbstractProgressMonitor.create()*.

Parameters

- **arg** (*Optional[Union[progress_interface.base.ProgressConfig, str, bool, type, Callable[[int], progress_interface.base.AbstractProgressMonitor]]]*) – See above.
- ****kw** – Additional keyword arguments to add to the returned config object.

Return type *progress_interface.base.ProgressConfig*

`progress_interface.base.`**register**(*key*, *arg=None*, ***, *overwrite=False*)

Register a progress monitor class or factory function under the given key.

If `arg` is not None, it is converted to a `ProgressConfig` instance and registered immediately. Otherwise a decorator function is returned which registers its argument under the given key.

Parameters

- **key** (*str*) – Key to register under.
- **arg** (*Optional[Union[progress_interface.base.ProgressConfig, str, bool, type, Callable[[int], progress_interface.base.AbstractProgressMonitor]]]*) – None or any value that can be passed to *progress_config()*.
- **overwrite** (*bool*) – Whether to allow overwriting of existing keys.

Returns The `ProgressConfig` instance registered if `arg` is not None, otherwise a decorator function which registers its argument and returns it unchanged.

Return type Union[*ProgressConfig*, Callable]

`progress_interface.base.`**ProgressArg**

Type alias for argument to `get_config()` and *get_progress()*.

alias of Optional[Union[*progress_interface.base.ProgressConfig*, str, bool, type, Callable[[int], *progress_interface.base.AbstractProgressMonitor*]]]

progress_interface.base.**ProgressFactoryFunc**
Type alias for a factory function with signature `(total: int, **kw) -> AbstractProgressMonitor`.

alias of Callable[[int], *progress_interface.base.AbstractProgressMonitor*]

progress_interface.base.**REGISTRY = {'click': <progress_interface.base.ProgressConfig object>, 'tqdm': <progress_interface.base.ProgressConfig object>, 'tqdm-notebook': <progress_interface.base.ProgressConfig object>, 'tqdm-std': <progress_interface.base.ProgressConfig object>}**
Registry of string keys to *ProgressConfig* instances.

1.2 progress_interface.monitors

Progress monitor implementations.

class progress_interface.monitors.**ClickProgressMonitor**
Bases: *progress_interface.base.AbstractProgressMonitor*

Wrapper around a progress bar from the `click` library, using `click.progressbar()`.

__init__(*pbar*)

Parameters **pbar** – Progress bar object returned by `click.progressbar`.

close()
Stop tracking/displaying progress and perform whatever cleanup is necessary.

classmethod create(*total*, *, *initial=0*, *desc=None*, *file=None*, ***kw*)

Parameters

- ****kw** – Passed to `click.progressbar`.
- **total** (*int*) –
- **initial** (*int*) –
- **desc** (*Optional[str]*) –
- **file** (*Optional[TextIO]*) –

increment(*delta=1*)
Increment the position of the monitor by the given value.

Parameters **delta** (*int*) –

moveto(*n*)
Set the monitor's position to the given value.

Parameters **n** (*int*) –

class progress_interface.monitors.**TqdmProgressMonitor**
Bases: *progress_interface.base.AbstractProgressMonitor*

Wrapper around a progress bar from the `tqdm` library.

__init__(*pbar*)

Parameters **pbar** – tqdm instance.

close()
Stop tracking/displaying progress and perform whatever cleanup is necessary.

classmethod create(*total*, *, *initial=0*, *desc=None*, *file=None*, *tqdm='tqdm.auto:tqdm'*, ***kw*)

Parameters

- **tqdm** (*Union[type, str]*) – tqdm class to use. Can be a string formatted like 'tqdm.std:tqdm'.
- ****kw** – Passed to tqdm constructor.
- **total** (*int*) –
- **initial** (*int*) –
- **desc** (*Optional[str]*) –
- **file** (*Optional[TextIO]*) –

increment(*delta=1*)
Increment the position of the monitor by the given value.

Parameters **delta** (*int*) –

moveto(*n*)
Set the monitor's position to the given value.

Parameters **n** (*int*) –

1.3 progress_interface.test

Utilities for testing.

class progress_interface.test.**TestProgressMonitor**
Bases: *progress_interface.base.AbstractProgressMonitor*

Progress monitor which displays no user information but does track progress information.

To be used for testing.

__init__(*total*, *initial=0*, *allow_decrement=True*, ***kw*)

Parameters

- **total** (*int*) –
- **initial** (*int*) –
- **allow_decrement** (*bool*) –

close()
Stop tracking/displaying progress and perform whatever cleanup is necessary.

increment(*delta=1*)
Increment the position of the monitor by the given value.

Parameters **delta** (*int*) –

moveto(*n*)
Set the monitor's position to the given value.

Parameters **n** (*int*) –

progress_interface.test.**capture_progress**(*config*)

Creates a ProgressConfig which captures references to the progress monitor instances created with it.

This is intended to be used for testing functions which create progress monitor instances internally that normally would not be accessible by the caller. The captured instance can be checked to ensure it has the correct attributes and went through the full range of iterations, for example.

Returns The first item is a modified ProgessConfig instance which can be passed to the function to be tested. The second is a list which is initially empty, and is populated with progress monitor instances as they are created by it.

Return type Tuple[*ProgressConfig*, List[*AbstractProgressMonitor*]]

Parameters **config** (progress_interface.base.ProgressConfig) –

CHAPTER

TWO

INDICES AND TABLES

- genindex
- modindex
- search

PYTHON MODULE INDEX

p

INDEX

Symbols

A

C

D

F

G

I

K

M

N

P

R

T

U