libcobblersignatures Documentation

Release 0.0.1

Enno Gotthold

Dec 13, 2022

Contents:

This library supplements the data for the `cobbler import` workflow. Cobbler is a server which configures your machine to act as a bootserver, thus managing TFTP, DHCP and optionally DNS. More information about Cobbler can be found here: Github

For more Information about what this library is for, see this documentation. I recommend starting with the Quickstart document from the TOC below.

CHAPTER 1

Quickstart

Installing from PyPi: `pip install libcobblersignatures`

Running the CLI when installed: `cobbler-manage-signatures`

1.1 What is this library for?

The purpose of this library is to manage and abstract the access on how we manage the data for our operating system detection. Currently this is a JSON file which is saved to the hard drive.

To understand what this library does we need to go into detail about what Cobbler does with the provided data. Cobbler has a functionality called `import`, which means that you give the CLI or API an ISO image which then gets searched through to save information about it in Cobbler. This means that Cobbler somehow needs to detect which operating system and which version of it is inside the ISO image. This library provides the data to distinguish between these ISOs from each other.

1.2 How to use it?

In the CLI Section you can see the most basic usage of this library. To get the most of it, it's not recommended to install this library directly but rather use it together with Cobbler who has a dependency on it. If you just want to manipulate the `signatures.json` file, it's recommended to use a built package from your package manager (if available).

CHAPTER 2

Command Line Interface

The CLI is reachable via the command `cobbler-manage-signatures`. After that you are in an interactive session. This session does not save anything except you explicitly tell the CLI to do so.

2.1 The top-level-menu

- Import: See *The Import menu*
- Export: See *The Export menu*
- Edit: See *The Edit menu*
- Exit: This button has no confirmation and will discard everything it has in memory.

2.2 The Import menu

Except for the last case, this menu always asks for the source in the next step. The source will always be a single line of text. Currently if you have more then a single line the CLI may break.

- URL: This may be any URL which is reachable by the requests package. The file then is downloaded and parsed.
- String: This should be a valid one line JSON string which is then parsed by the application.
- File: This is a relative or absolute path to the JSON file. The relative path is relative to your current location, not the location of the script.
- Go Back: Go one menu level up again.

2.3 The Export menu

Except for the last case, this menu will persist what you have achieved so far with the CLI. In case there is nothing to be exported you will receive a message for that. Files are overwritten without question, so use the second option with great care.

- String: Hand back a single line of JSON in the stdout stream.

- File: Write the JSON well formatted to the given location.
- Go Back: Go one menu level up again.

2.4 The Edit menu

Except for the last case, this menu will give you the ability to edit your JSON file. Since there is no full Cobbler instance behind this tool, the logical validations are very loose. However we try to check as much as reasonably possible.

- Add Operating System Breed: This option is followed by a question for the name. Please only use alphanumeric characters, dashes and underscores.
- Remove Operating System Breed: This option is followed by a multiple choice menu where you can select a single Breed.
- Edit the name of an Operating System Breed: This option presents you with a list of Breeds and then gives you an input to set the new name.
- Add Operating System Version: If you have a breed then this menu lets you select it and then presents you an input where you can choose the name of the Version.
- Remove Operating System Version: If you have a breed, this menu lets you select it and then presents you with a menu which does the same for the version.
- Edit the information of an Operating System Version: See *Remove Operating System Version*
- Start from scratch: Discards everything you have done so far and creates an empty object for you.
- Go Back: Go one menu level up again.

2.4.1 Remove Operating System Version

- `signatures`
- `version_file`
- `version_file_regex`
- `kernel_arch`
- `kernel_arch_regex`
- `supported_arches`
- `supported_repo_breeds`
- `kernel_file`
- `initrd_file`
- `isolinux_ok`
- `default_autoinstall`
- `kernel_options`
- `kernel_options_post`
- `template_files`
- `boot_files`
- `boot_loaders`

CHAPTER 3

Specification of `signatures.json`

3.1 Goal

This file should contain the data which is used by a program to recognize with a general algorithm which operating system an ISO image passes to the program. This specification should not be limited itself to one operating system. If that this is not possible, we shall focus on Linux and its distributions.

3.2 File structure

The file should be a single JSON object which has a single key with the name *breeds*. This key should contain a single sub-object with key-value pairs for each operating system group.

Currently the following operating system groups are existing:

- redhat
- debian
- ubuntu
- suse
- vmware
- freebsd
- xen
- unix
- windows
- powerkvm
- generic

This list should be modified when new groups are added, removed or changed when changed in the JSON.

Each of these operating system groups have a key for each version of it. The key should be in lowercase and should have a version suffix. The name of the key should be unique across the distribution section and contains an object.

The object has the following keys (type included):

Name	Type	Description
`signatures`	Array of Strings	
`version_file`	String	
`version_file_regex`	String	
`kernel_arch`	String	
`kernel_arch_regex`	String	
`supported_arches`	Array of Strings	
`supported_repo_breeds`	Array of Strings	
`kernel_file`	String	
`initrd_file`	String	
`isolinux_ok`	Boolean	
`default_autoinstall`	String	
`kernel_options`	String	
`kernel_options_post`	String	
`template_files`	String	
`boot_files`	Array of Strings	
`boot_loaders`	Dictionary	

3.3 Example files

Please have a look at https://github.com/cobbler/cobbler/blob/master/config/cobbler/distro_signatures.json

The most simple valid (but useless) `signatures.json` file will be:

```
{
  "breeds": {}
}
```

CHAPTER 4

libcobblersignatures

4.1 libcobblersignatures package

4.1.1 Subpackages

libcobblersignatures.models package

Submodules

libcobblersignatures.models.osbreed module

class `libcobblersignatures.models.osbreed.`**`OsBreed`**(*name: str*)

Bases: `object`

On operating system breed like SUSE or Redhat. The specification of the attributes and what values are valid are described in the JSON specification.

`decode`(*data: dict*)

Decodes the received data. Decoding of each single version is done by the corresponding decode method in `Osversion`.

Parameters **`data`** – The data to decode.

`encode`() → dict

Encodes the current OsBreed and nested Osversions.

Returns The dictionary with all its versions. The name of the OsBreed is not included as this is normally the name of the returned key.

`name`

This property represents the name of the operating system breed.

Setter Invalid values will be discarded.

Deleter Always raises a TypeError since this is not allowed.

Getter The last successfully validated name of the operating system breed.

`osversion_add`(*name: str*, *version: libcobblersignatures.models.osversion.Osversion*)

Add an Osversion to this OsBreed.

Parameters

- **`name`** – The name of the version to add.
- **`version`** – The Osversion to add to the OsBreed.

Raises `ValueError` – If the Name is not a `str` and/or the Version is not an `Osversion`.

`osversion_remove`(*key: str*)
Remove an Osversion with its key.

Parameters `key` – The key of the dictionary to remove.

`osversions`
An ordered dictionary which contains all versions of the operating system breed.

Setter Raises a TypeError in case the value is not an OrderedDict.

Getter The ordered dictionary.

Deleter Will reset the number of versions to an empty list.

libcobblersignatures.models.osversion module

Module for the datastructure of an operating system version. An operating system version needs to be grouped under an operating system breed.

class `libcobblersignatures.models.osversion.`**`Osversion`**
Bases: `object`

A version of an operating system breed like `openSUSE Leap 15.2`. The specification of the attributes and what values are valid are described in the JSON specification.

`boot_files`
Unknown field currently. There for compatibility reasons for now. Used by xenserver

Getter The last successfully validated value of this field.

Setter Will set this if the validation succeeds, otherwise will raise an exception (TypeError).

Deleter Resets this to an empty list.

Type set

`boot_loaders`
Defines the supported well known boot loaders inside Cobbler.

Getter The last successfully validated value of this field.

Setter If validation is successful the value will be set, otherwise raises an exception (TypeError).

Deleter Resets this property to an empty dict.

Type dict

`decode`(*data: dict*)
Decodes the received data. This means parsing each attribute from the JSON into the property of an object.

Parameters `data` – The data to decode.

`default_autoinstall`
The filename for the default autoinstall template in Cobbler.

Getter The last successfully validated value of this field.

Setter Will set this if the validation succeeds, otherwise will raise an exception (TypeError).

Deleter Resets this to an empty str.

Type str

encode() → dict

Encodes a single *Osversion*. This means that the properties of an object is transferred into a JSON.

Returns The dictionary with the data.

initrd_file

The regular expression to match to find the initrd.

Getter The last successfully validated value of this field.

Setter Will set this if the validation succeeds, otherwise will raise an exception (TypeError).

Deleter Resets this to an empty str.

Type str

isolinux_ok

Unknown field currently. There for compatibility reasons for now.

Getter The last successfully validated value of this field.

Setter Will set this if the validation succeeds, otherwise will raise an exception (TypeError).

Deleter Resets this to `False`.

Type bool

kernel_arch

The regular expression which tells Cobbler where to look for the architecture of the operating system. In some cases this may also be a path to the file where Cobbler should look for the architecture.

Getter The value of the last successfully validated kernel_arch.

Setter May raise a TypeError in case the value was not of type str.

Deleter Resets the value instead of deleting the attribute.

Type str

kernel_arch_regex

In case `kernel_arch` does not point to the architecture directly, this is the regex where Cobbler looks for in the file specified by `kernel_arch`.

Getter The last successfully validated value of this field.

Setter Will set this if the validation succeeds, otherwise will raise an exception (TypeError).

Deleter Resets this to an empty str.

Type str

kernel_file

The regular expression to match to find the kernel.

Getter The last successfully validated value of this field.

Setter Will set this if the validation succeeds, otherwise will raise an exception (TypeError).

Deleter Resets this to an empty str.

Type str

kernel_options

Default kernel options to apply to the imported ISO.

Getter The last successfully validated value of this field.

Setter Will set this if the validation succeeds, otherwise will raise an exception (TypeError).

Deleter Resets this to an empty str.

Type str

kernel_options_post

Default kernel post options to apply to the imported ISO.

Getter The last successfully validated value of this field.

Setter Will set this if the validation succeeds, otherwise will raise an exception (TypeError).

Deleter Resets this to an empty str.

Type str

signatures

This is a list of strings with currently an unknown functionality.

Setter May raise a TypeError in case the value was not of type list.

Getter Returns the last correctly validated str of the property.

Deleter Resets this to an empty list.

Type set

supported_arches

Unused field currently. There for compatibility reasons for now.

Getter The last successfully validated value of this field.

Setter Will set this if the validation succeeds, otherwise will raise an exception (TypeError).

Deleter Resets this to an empty list.

Type set

supported_repo_breeds

Unused field currently. There for compatibility reasons for now.

Getter The last successfully validated value of this field.

Setter Will set this if the validation succeeds, otherwise will raise an exception (TypeError).

Deleter Resets this to an empty list.

Type set

template_files

Currently only used in ESXi. Needs more investigation what this is for.

Getter The last successfully validated value of this field.

Setter Will set this if the validation succeeds, otherwise will raise an exception (TypeError).

Deleter Resets this to an empty str.

Type str

version_file

The regular expression which points to the file with the os-version info.

Setter May raise a TypeError in case the value was not of type str.

Getter Returns the last correctly validated str of the property.

Deleter Resets this to an empty string.

Type str

version_file_regex

The regular expression which tells Cobbler to pick this version if it matches.

Getter The str with the regex.

Setter Validates the regex and raises in case an error was detected (TypeError).

Deleter Resets the attribute to an empty str instead of deleting it.

Type str

Module contents

This module contains the data structures of the library. You may use them without using the library, however they may not work as expected when using them on its own.

4.1.2 Submodules

4.1.3 libcobblersignatures.cli module

4.1.4 libcobblersignatures.enums module

This module contains all enums which are used in the library.

class `libcobblersignatures.enums.`**`ExportTypes`**

Bases: `enum.Enum`

An enumeration which defines the possible export targets for the JSON file.

`FILE = 0`

This value shall be given when the content of the manipulated content shall be exported to a file.

`STRING = 1`

This value shall be given when the content of the manipulated content shall be exported to a String which is outputted to the shell.

class `libcobblersignatures.enums.`**`ImportTypes`**

Bases: `enum.Enum`

An enumeration which defines the possible sources for importing the JSON file.

`FILE = 0`

This value shall be given when the content shall be imported from a file on a file system which is locally accessible.

`STRING = 2`

This value shall be given when the content shall be imported from a string. This string shall not contain any linebreaks.

`URL = 1`

This value shall be given when the content shall be imported from a URL.

class `libcobblersignatures.enums.`**`OsArchitectures`**

Bases: `enum.Enum`

An enumeration which defines the in Cobbler available architectures.

`amd64 = 5`

Synonym for `x86_64`.

`i386 = 1`

32-bit architecture which is also called `IA-32` or `80x86` by some people.

`ppc = 3`

32-bit big-endian PowerPC architecture.

`ppc64 = 4`

64-bit big-endian PowerPC architecture.

`x86_64 = 2`

64-bit architecture which is also called `x64`, `x86-64`, `AMD64` or `amd64`.

class `libcobblersignatures.enums.`**`RepositoryBreeds`**

Bases: `enum.Enum`

An enumeration which defines the in Cobbler available repository breeds.

`apt = 4`
A repository which is managed by apt.

`rhn = 2`
A repository type from Red Hat which can be used by yum.

`rsync = 1`
A repository which is synced by rsync.

`yum = 3`
A repository which is manged by yum.

4.1.5 Module contents

This library contains additionally a CLI which enables you to have a feature equal way to manipulate the objects, as well as to im- and export them.

class `libcobblersignatures.`**`Signatures`**

Bases: `object`

This is the entry point of the library. You may create one or more objects of this class. Each instance of it is independent of the other. I recommend only to have on object at a time of this class. This library doesn't persist anything except you explicitly use the export method of the instance.

`addosbreed`(*name: str*)
Add a new `OsBreed`.

Parameters **`name`** – The name of the new breed. Must not exist in the currently loaded models.

`addosversion`(*breedindex: int*, *versionname: str*, *versiondata*)
Add a new `Osversion`.

Parameters

- **`breedindex`** – The index of the operating system breed. This can be found by using `get_breed_index_by_name()`.
- **`versionname`** – The name of the new version.
- **`versiondata`** – The object with the data of the version to add. If this is `None` then an empty version will be created.

`exportsignatures`(*export_type: libcobblersignatures.enums.ExportTypes*, *target: str = ''*, *sort_keys: bool = False*, *indent: Union[None, int] = None*)
This is the main export function.

Parameters

- **`export_type`** – One of the values from the `ExportTypes`.
- **`target`** – This is only required when using this for a file based export. Otherwise this can be skipped.
- **`sort_keys`** – If the keys of the dictionary should be sorted to be more human readable.
- **`indent`** – If this is something other then `None` then the JSON will be pretty printed.

Raises

- **`ValueError`** – When the `ExportTypes` is not implemented or not known.
- **`TypeError`** – When one of the arguments has the wrong type.

`get_breed_index_by_name`(*name: str*) → int
Searches with the name of the `OsBreed` for the index.

Parameters **`name`** – The name of the `OsBreed` to look for.

Returns The number of the index or `-1`.

`importsignatures`(*import_type: libcobblersignatures.enums.ImportTypes*, *source: str*)
This is the main import function.

Parameters

- **`import_type`** – This is one of the four options from the ImportTypes Enum
- **`source`** – A string containing the data to be parsed into a json in the end

`jsontomodels`()
Convert the loaded JSON to the internal modules. Without calling this the loaded data will not be available for manipulation.

`osbreeds`
This property represents the currently manipulated data structures.

Returns The list with the content which is currently manipulated.

`removeosbreed`(*index: int*)
Remove an operating system breed via its index. All nested content will be removed.

Parameters **`index`** – The index which will be removed.

`removeosversion`(*breedindex: int*, *versionname: str*)
Remove a single operating system version.

Parameters

- **`breedindex`** – The index of the operating system breed index.
- **`versionname`** – The name of the version to remove.

`signaturesjson`
This property represents the json which was im- or exported.

Raises **`JSONDecodeError`** – If invalid json is given

Returns The last valid content of the json which the library knows about.

CHAPTER 5

Indices and tables

- genindex
- modindex
- search

Python Module Index

Index