OmrDatasetTools Documentation

Release 1.0

Alexander Pacha

Apr 01, 2024

CONTENTS:

1 Example usage **3**
- 1.1 `Downloader` Module 3
- 1.2 Image Generators 8

2 Indices and tables **15**

Python Module Index **17**

Index **19**

This is a collection of tools for working with the datasets referenced from this repository, including common function like:

- Downloading the datasets and extracting them
- Processing specific datasets, e.g. generating images from the raw HOMUS dataset.

A pip-package is available so you can include the tools conveniently into your projects by using

```
pip install omrdatasettools
```

CHAPTER

ONE

EXAMPLE USAGE

Consider you want to work with the HOMUS dataset, you can use the following script to download the dataset:

```
from omrdatasettools import Downloader, OmrDataset

downloader = Downloader()
downloader.download_and_extract_dataset(OmrDataset.HOMUS_V2, "data")
```

Once the download has completed, you may want to work with images, instead of textual descriptions, so we can generate them.

```
from omrdatasettools import HomusImageGenerator

HomusImageGenerator.create_images(raw_data_directory="data",
                                  destination_directory="homus_data",
                                  stroke_thicknesses=[3],
                                  canvas_width=96,
                                  canvas_height=192,
                                  staff_line_spacing=14,
                                  staff_line_vertical_offsets=[24])
```

Note that the image generator has a lot of options that define how the images will be generated, but to stay with this example, it will create image of the size 192x96, draw the symbols with a line-thickness of three pixels, superimpose five staff-lines with 14 pixel space inbetween, beginning at 24 pixels from the top.

1.1 Downloader Module

class omrdatasettools.Downloader.**Downloader**

The class for downloading OMR datasets. It downloads the selected dataset from Github and extracts it to a specified directory.

Downloader.**download_and_extract_dataset**(*dataset: OmrDataset*, *destination_directory: str | Path*, *tmp_directory: Path | None = None*)

Starts the download of the dataset and extracts it into the specified directory.

Parameters

- **dataset** – The dataset that should be downloaded
- **destination_directory** – The target directory, where the dataset should be extracted into
- **tmp_directory** – The optional directory where the compressed dataset will be downloaded to

1.1.1 Examples

```
>>> from omrdatasettools import Downloader, OmrDataset
>>> downloader = Downloader()
>>> downloader.download_and_extract_dataset(OmrDataset.Homus_V2,
↪"data")
```

Downloader.**download_images_from_mei_annotation**(*dataset: OmrDataset*, *dataset_directory: str*, *base_url: str*)

Crawls the images of an Edirom dataset, if provided with the respective URL. To avoid repetitive crawling, this URL has to be provided manually. If you are interested in these datasets, please contact the authors.

1.1.2 Examples

```
>>> from omrdatasettools import Downloader, OmrDataset
>>> downloader = Downloader()
>>> downloader.download_and_extract_dataset(OmrDataset.Edirom_
↪Bargheer, "data/Bargheer")
>>> downloader.download_images_from_mei_annotation(OmrDataset.
↪Edirom_Bargheer, "data/Bargheer",
>>>     "INSERT_DATASET_URL_HERE")
```

or

```
>>> downloader.download_and_extract_dataset(OmrDataset.Edirom_
↪FreischuetzDigital, "data/Freischuetz")
>>> downloader.download_images_from_mei_annotation(OmrDataset.
↪Edirom_FreischuetzDigital, "data/Freischuetz",
>>>     "INSERT_DATASET_URL_HERE")
```

1.1.3 OmrDataset Module

class omrdatasettools.OmrDataset.**OmrDataset**(*value*, *names=None*, *, *module=None*, *qualname=None*, *type=None*, *start=1*, *boundary=None*)

The available OMR datasets that can be automatically downloaded with Downloader.py

AudioLabs_v1 = 24

The AudioLabs v1 dataset (aka. Measure Bounding Box Annotation) from https://www.audiolabs-erlangen.de/resources/MIR/2019-ISMIR-LBD-Measures, Copyright 2019 by Frank Zalkow, Angel Villar Corrales, TJ Tsai, Vlora Arifi-Müller, and Meinard Müller under CC BY-NC-SA 4.0 license.

AudioLabs_v2 = 25

The AudioLabs v2 dataset, enhanced with staves, staff measures and the original system measures. The annotations are available in csv, JSON and COCO format.

Audiveris = 1

The Audiveris OMR dataset from https://github.com/Audiveris/omr-dataset-tools, Copyright 2017 by Hervé Bitteur under AGPL-3.0 license

Baro = 2

The Baro Single Stave dataset from http://www.cvc.uab.es/people/abaro/datasets.html, Copyright 2019 Arnau Baró, Pau Riba, Jorge Calvo-Zaragoza, and Alicia Fornés under CC-BY-NC-SA 4.0 license

Capitan = 3

The Capitan dataset from http://grfia.dlsi.ua.es/, License unspecified, free for research purposes

ChoiAccidentals = 26

The Accidentals detection dataset by Kwon-Young Choi from https://www-intuidoc.irisa.fr/en/choi_accidentals/, License unspecified.

CvcMuscima_MultiConditionAligned = 4

Custom version of the CVC-MUSCIMA dataset that contains all images in grayscale, binary and with the following staff-line augmentations: interrupted, kanungo, thickness-variation-v1/2, y-variation-v1/2 typeset-emulation and white-speckles. (all data augmentations that could be aligned automatically). The grayscale images are different from the WriterIdentification dataset, in such a way, that they were aligned to the images from the Staff-Removal dataset. This is the recommended dataset for object detection, as the MUSCIMA++ annotations can be used with a variety of underlying images. See https://github.com/apacha/CVC-MUSCIMA to learn more.

CvcMuscima_StaffRemoval = 5

The larger version of the CVC-MUSCIMA dataset for staff removal in black and white with augmentations from http://www.cvc.uab.es/cvcmuscima/index_

database.html, Copyright 2012 Alicia Fornés, Anjan Dutta, Albert Gordo and Josep Lladós under CC-BY-NC-SA 4.0 license

CvcMuscima_WriterIdentification = 6

The smaller version of the CVC-MUSCIMA dataset for writer identification in grayscale from http://www.cvc.uab.es/cvcmuscima/index_database.html, Copyright 2012 Alicia Fornés, Anjan Dutta, Albert Gordo and Josep Lladós under CC-BY-NC-SA 4.0 license

DeepScores_V1_Extended = 21

The DeepScore dataset (version 1) with extended vocabulary from https://tuggeluk.github.io/downloads/, License unspecified.

DeepScores_V1_Extended_100_Pages = 20

Subselection of 100 pages from the DeepScore dataset (version 1) with extended vocabulary from https://tuggeluk.github.io/downloads/, License unspecified.

DeepScores_V2_Complete = 23

The complete DeepScore dataset (version 2) from https://zenodo.org/records/4012193, under CC BY 4.0 license.

WARNING: The size of this dataset is over 80GB!

DeepScores_V2_Dense = 22

Subselection of 1714 pages from the DeepScore dataset (version 2) with extended vocabulary from https://zenodo.org/records/4012193, under CC BY 4.0 license.

DoReMi = 27

DoReMi dataset from https://github.com/steinbergmedia/DoReMi/, License unspecified.

Edirom_Bargheer = 7

Edirom dataset. All rights reserved

Edirom_FreischuetzDigital = 8

Edirom datasets on Freischuetz from https://freischuetz-digital.de/edition.html. All rights reserved.

Fornes = 9

The Fornes Music Symbols dataset from http://www.cvc.uab.es/~afornes/, License unspecified - citation requested

Homus_V1 = 10

The official HOMUS dataset from http://grfia.dlsi.ua.es/homus/, License unspecified.

Homus_V2 = 11

The improved version of the HOMUS dataset with several bugs-fixed from https://github.com/apacha/Homus.

MScoreLib_All = 30

The full MScoreLib corpus from http://mscorelib.com/, manually inputed scores by humans, License unspecified.

MScoreLib_Prokofiev = 32

MScoreLib corpus of Prokofiev music, converted with SharpEye and PhotoScore, from http://mscorelib.com/, License unspecified.

MScoreLib_Scriabin = 31

MScoreLib corpus of Scriabin music, converted with SharpEye and PhotoScore, from http://mscorelib.com/, License unspecified.

MuscimaPlusPlus_Images = 14

The subset of 140 images from the CVC-MUSCIMA dataset that were used for the MUSCIMA++ dataset.

MuscimaPlusPlus_MeasureAnnotations = 15

A sub-set of the MUSCIMA++ annotations that contains bounding-box annotations for staves, staff measures and system measures. It was semi-automatically constructed from existing annotations and manually verified for correctness. The annotations are available in a plain JSON format as well as in the COCO format.

MuscimaPlusPlus_V1 = 12

The MUSCIMA++ dataset from https://ufal.mff.cuni.cz/muscima, Copyright 2017 Jan Hajic jr. under CC-BY-NC-SA 4.0 license.

MuscimaPlusPlus_V2 = 13

The second version of the MUSCIMA++ dataset from https://github.com/OMR-Research/muscima-pp.

OpenOmr = 16

The OpenOMR Symbols dataset from https://sourceforge.net/projects/openomr/, Copyright 2013 by Arnaud F. Desaedeleer under GPL license.

OpenScoreLieder = 28

OpenScore Lieder corpus from https://github.com/OpenScore/Lieder, CC-0 license.

OpenScoreStringQuartets = 29

OpenScore StringQuartet corpus from https://github.com/OpenScore/StringQuartets, CC-0 license.

Printed = 17

The Printed Music Symbols dataset from https://github.com/apacha/PrintedMusicSymbolsDataset, Copyright 2017 by Alexander Pacha under MIT license.

Rebelo1 = 18

The Rebelo dataset (part 1) with music symbols from http://www.inescporto.pt/~arebelo/index.php, Copyright 2017 by Ana Rebelo under CC BY-SA 4.0 license

Rebelo2 = 19

The Rebelo dataset (part 2) with music symbols from http://www.inescporto.pt/~arebelo/index.php, Copyright 2017 by Ana Rebelo under CC BY-SA 4.0 license

dataset_download_urls() → Dict[str, str]

Returns a mapping with all URLs, mapped from their enum keys

get_dataset_download_url() → str

Returns the url of the selected dataset. Example usage: OmrDataset.Fornes.get_dataset_download_url()

get_dataset_filename() → str

Returns the name of the downloaded zip file of a dataset. Example usage: OmrDataset.Fornes.get_dataset_filename()

1.2 Image Generators

1.2.1 AudiverisOmrImageGenerator Module

class omrdatasettools.AudiverisOmrImageGenerator.**AudiverisOmrImageGenerator**

AudiverisOmrImageGenerator.**extract_symbols**(*raw_data_directory: str*, *destination_directory: str*)

Extracts the symbols from the raw XML documents and matching images of the Audiveris OMR dataset into individual symbols

Parameters

- **raw_data_directory** – The directory, that contains the xml-files and matching images
- **destination_directory** – The directory, in which the symbols should be generated into. One sub-folder per symbol category will be generated automatically

1.2.2 CapitanImageGenerator Module

class omrdatasettools.CapitanImageGenerator.**CapitanImageGenerator**

CapitanImageGenerator.**create_capitan_images**(*raw_data_directory: str*, *destination_directory: str*, *stroke_thicknesses: List[int]*) → None

Creates a visual representation of the Capitan strokes by parsing all text-files and the symbols as specified by the parameters by drawing lines that connect the points from each stroke of each symbol.

Parameters

- **raw_data_directory** – The directory, that contains the raw capitan dataset

- **destination_directory** – The directory, in which the symbols should be generated into. One sub-folder per symbol category will be generated automatically
- **stroke_thicknesses** – The thickness of the pen, used for drawing the lines in pixels. If multiple are specified, multiple images will be generated that have a different suffix, e.g. 1-16-3.png for the 3-px version and 1-16-2.png for the 2-px version of the image 1-16

1.2.3 HomusImageGenerator Module

class omrdatasettools.HomusImageGenerator.**HomusImageGenerator**

static HomusImageGenerator.**create_images**(*raw_data_directory: str*, *destination_directory: str*, *stroke_thicknesses: List[int]*, *canvas_width: int = None*, *canvas_height: int = None*, *staff_line_spacing: int = 14*, *staff_line_vertical_offsets: List[int] = None*, *random_position_on_canvas: bool = False*) → dict

Creates a visual representation of the Homus Dataset by parsing all text-files and the symbols as specified by the parameters by drawing lines that connect the points from each stroke of each symbol.

Each symbol will be drawn in the center of a fixed canvas, specified by width and height.

Parameters

- **raw_data_directory** – The directory, that contains the text-files that contain the textual representation of the music symbols
- **destination_directory** – The directory, in which the symbols should be generated into. One sub-folder per symbol category will be generated automatically
- **stroke_thicknesses** – The thickness of the pen, used for drawing the lines in pixels. If multiple are specified, multiple images will be generated that have a different suffix, e.g. 1-16-3.png for the 3-px version and 1-16-2.png for the 2-px version of the image 1-16
- **canvas_width** – The width of the canvas, that each image will be drawn upon, regardless of the original size of the symbol. Larger symbols will be cropped. If the original size of the symbol should be used, provided None here.
- **canvas_height** – The height of the canvas, that each image will be drawn upon, regardless of the original size of the symbol. Larger symbols will be cropped. If the original size of the symbol should be used, provided None here

- **staff_line_spacing** – Number of pixels spacing between each of the five staff-lines
- **staff_line_vertical_offsets** – List of vertical offsets, where the staff-lines will be superimposed over the drawn images. If None is provided, no staff-lines will be superimposed. If multiple values are provided, multiple versions of each symbol will be generated with the appropriate staff-lines, e.g. 1-5_3_offset_70.png and 1-5_3_offset_77.png for two versions of the symbol 1-5 with stroke thickness 3 and staff-line offsets 70 and 77 pixels from the top.
- **random_position_on_canvas** – True, if the symbols should be randomly placed on the fixed canvas. False, if the symbols should be centered in the fixed canvas. Note that this flag only has an effect, if fixed canvas sizes are used.

Returns
A dictionary that contains the file-names of all generated symbols and the respective bounding-boxes of each symbol.

static HomusImageGenerator.**add_arguments_for_homus_image_generator**(*parser: ArgumentParser*)

1.2.4 HomusSymbol Module

class omrdatasettools.HomusImageGenerator.**HomusSymbol**(*content: str*, *strokes: List[List[Point2D]]*, *symbol_class: str*, *dimensions: Rectangle*)

static HomusSymbol.**initialize_from_string**(*content: str*) → *HomusSymbol*

Create and initializes a new symbol from a string

Parameters
content – The content of a symbol as read from the text-file

Returns
The initialized symbol

Return type
HomusSymbol

HomusSymbol.**draw_into_bitmap**(*export_path: ExportPath*, *stroke_thickness: int*, *margin: int = 0*) → None

Draws the symbol in the original size that it has plus an optional margin

Parameters

- **export_path** – The path, where the symbols should be created on disk
- **stroke_thickness** – Pen-thickness for drawing the symbol in pixels
- **margin** – An optional margin for each symbol

HomusSymbol.**draw_onto_canvas**(*export_path: ExportPath*, *stroke_thickness: int*, *margin: int*, *destination_width: int*, *destination_height: int*, *staff_line_spacing: int = 14*, *staff_line_vertical_offsets: List[int] = None*, *bounding_boxes: dict = None*, *random_position_on_canvas: bool = False*) → None

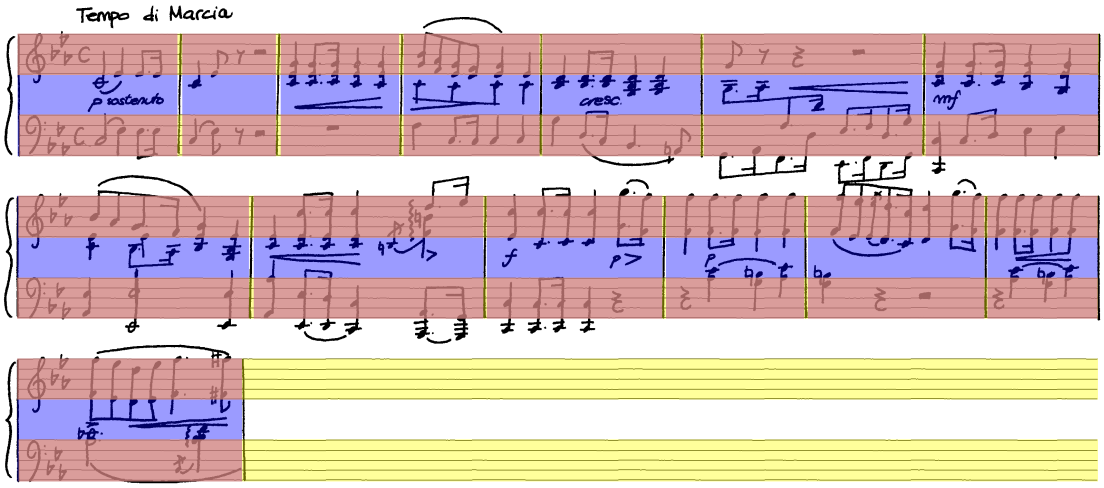
Draws the symbol onto a canvas with a fixed size

Parameters

- **bounding_boxes** – The dictionary into which the bounding-boxes will be added of each generated image
- **export_path** – The path, where the symbols should be created on disk
- **stroke_thickness** –
- **margin** –
- **destination_width** –
- **destination_height** –
- **staff_line_spacing** –
- **staff_line_vertical_offsets** – Offsets used for drawing staff-lines. If None provided, no staff-lines will be drawn if multiple integers are provided, multiple images will be generated

1.2.5 MeasureVisualizer Module

This class can be used to generate visualizations of measure annotations, such as this one for the Muscima++ dataset:



class omrdatasettools.MeasureVisualizer.**MeasureVisualizer**(*draw_system_measures: bool*, *draw_stave_measures: bool*, *draw_staves: bool*)

Class that can be used to visualize the measure annotations that are provided as json files. Allows to enable/disable whether to draw system-measures (one measure for all instruments), stave-measures (one measure, single stave of one instrument) and staves (the whole line of a single instrument). System-measures contain all stave-measures that are being played simulatenously.

Bounding boxes will be drawn with semitransparent colors.

MeasureVisualizer.**draw_bounding_boxes_for_all_images_in_directory**(*image_directory*, *json_annotation_direct*

MeasureVisualizer.**draw_bounding_boxes_into_image**(*image_path: str*, *ground_truth_annotations_path: str*)

1.2.6 MuscimaPlusPlusMaskImageGenerator Module

class omrdatasettools.MuscimaPlusPlusMaskImageGenerator.**MaskType**(*value*, *names=None*, ***, *module=None*, *qualname=None*, *type=None*, *start=1*, *boundary=None*)

The type of masks that should be generated

NODES_SEMANTIC_SEGMENTATION = 1

creates mask images, where each type of node gets the same color mask (for semantic segmentation). The classes staffLine, staff and staffSpace are ignored.

STAFF_BLOBS_INSTANCE_SEGMENTATION = 3

creates mask images, where each staff will receive one big blob (filling the staff space regions) per staff line for instance segmentation. So each staff will have a different color.

STAFF_LINES_INSTANCE_SEGMENTATION = 2

creates mask images, where the masks of the staff lines are contained for instance segmentation. All five lines that form a staff will have the same color.

class omrdatasettools.MuscimaPlusPlusMaskImageGenerator.
MuscimaPlusPlusMaskImageGenerator

MuscimaPlusPlusMaskImageGenerator.**render_node_masks**(*raw_data_directory: str*, *destination_directory: str*, *mask_type:* MaskType)

Extracts all symbols from the raw XML documents and generates individual symbols from the masks

Parameters

- **raw_data_directory** – The directory, that contains the xml-files and matching images
- **destination_directory** – The directory, in which the symbols should be generated into. Per file, one mask will be generated.
- **mask_type** – The type of masks that you want to generate, e.g., masks for each node or staff lines only.

1.2.7 MuscimaPlusPlusSymbolImageGenerator Module

class omrdatasettools.MuscimaPlusPlusSymbolImageGenerator.
MuscimaPlusPlusSymbolImageGenerator

MuscimaPlusPlusSymbolImageGenerator.**extract_and_render_all_symbol_masks**(*raw_data_direc*
str,
des-
ti-
na-
tion_directory:
str)

Extracts all symbols from the raw XML documents and generates individual symbols from the masks

Parameters

- **raw_data_directory** – The directory, that contains the xml-files and matching images
- **destination_directory** – The directory, in which the symbols should be generated into. One sub-folder per symbol category will be generated automatically

CHAPTER

TWO

INDICES AND TABLES

- genindex
- modindex
- search

PYTHON MODULE INDEX

O

omrdatasettools, 8
omrdatasettools.Downloader, 3

INDEX

A

add_arguments_for_homus_image_generator() (*omrdataset-tools.HomusImageGenerator.HomusImageGenerator static method*), 10
AudioLabs_v1 (*omrdataset-tools.OmrDataset.OmrDataset attribute*), 5
AudioLabs_v2 (*omrdataset-tools.OmrDataset.OmrDataset attribute*), 5
Audiveris (*omrdataset-tools.OmrDataset.OmrDataset attribute*), 5
AudiverisOmrImageGenerator (*class in omrdataset-tools.AudiverisOmrImageGenerator*), 8

B

Baro (*omrdataset-tools.OmrDataset.OmrDataset attribute*), 5

C

Capitan (*omrdataset-tools.OmrDataset.OmrDataset attribute*), 5
CapitanImageGenerator (*class in omrdataset-tools.CapitanImageGenerator*), 8
ChoiAccidentals (*omrdataset-tools.OmrDataset.OmrDataset attribute*), 5
create_capitan_images() (*omrdataset-tools.CapitanImageGenerator.CapitanImageGenerator method*), 8
create_images() (*omrdataset-tools.HomusImageGenerator.HomusImageGenerator static method*), 9
CvcMuscima_MultiConditionAligned (*omrdataset-tools.OmrDataset.OmrDataset attribute*), 5
CvcMuscima_StaffRemoval (*omrdataset-tools.OmrDataset.OmrDataset attribute*), 5
CvcMuscima_WriterIdentification (*omrdataset-tools.OmrDataset.OmrDataset attribute*), 6

D

dataset_download_urls() (*omrdataset-tools.OmrDataset.OmrDataset method*), 7
DeepScores_V1_Extended (*omrdataset-tools.OmrDataset.OmrDataset attribute*), 6
DeepScores_V1_Extended_100_Pages (*omrdataset-tools.OmrDataset.OmrDataset attribute*), 6
DeepScores_V2_Complete (*omrdataset-tools.OmrDataset.OmrDataset attribute*), 6
DeepScores_V2_Dense (*omrdataset-tools.OmrDataset.OmrDataset attribute*), 6
DoReMi (*omrdataset-tools.OmrDataset.OmrDataset attribute*), 6
download_and_extract_dataset() (*omrdataset-tools.Downloader.Downloader*

method), 3
download_images_from_mei_annotation() (*omrdataset-tools.Downloader.Downloader method*), 4
Downloader (*class in omrdataset-tools.Downloader*), 3
draw_bounding_boxes_for_all_images_in_directory() (*omrdataset-tools.MeasureVisualizer.MeasureVisualizer method*), 12
draw_bounding_boxes_into_image() (*omrdataset-tools.MeasureVisualizer.MeasureVisualizer method*), 12
draw_into_bitmap() (*omrdataset-tools.HomusImageGenerator.HomusSymbol method*), 10
draw_onto_canvas() (*omrdataset-tools.HomusImageGenerator.HomusSymbol method*), 11

E

Edirom_Bargheer (*omrdataset-tools.OmrDataset.OmrDataset attribute*), 6
Edirom_FreischuetzDigital (*omrdataset-tools.OmrDataset.OmrDataset attribute*), 6
extract_and_render_all_symbol_masks() (*omrdataset-tools.MuscimaPlusPlusSymbolImageGenerator.MuscimaPlusPlusSymbolImageGenerator method*), 14
extract_symbols() (*omrdataset-tools.AudiverisOmrImageGenerator.AudiverisOmrImageGenerator method*), 8

F

Fornes (*omrdataset-tools.OmrDataset.OmrDataset attribute*), 6

G

get_dataset_download_url() (*omrdataset-tools.OmrDataset.OmrDataset method*), 8
get_dataset_filename() (*omrdataset-tools.OmrDataset.OmrDataset method*), 8

H

Homus_V1 (*omrdataset-tools.OmrDataset.OmrDataset attribute*), 6
Homus_V2 (*omrdataset-tools.OmrDataset.OmrDataset attribute*), 6
HomusImageGenerator (*class in omrdataset-tools.HomusImageGenerator*), 9
HomusSymbol (*class in omrdataset-tools.HomusImageGenerator*), 10

I

initialize_from_string() (*omrdataset-tools.HomusImageGenerator.HomusSymbol static method*), 10

M

MaskType (*class in omrdataset-tools.MuscimaPlusPlusMaskImageGenerator*), 13
MeasureVisualizer (*class in omrdataset-tools.MeasureVisualizer*), 12
module
    omrdatasettools, 8
    omrdatasettools.Downloader, 3
MScoreLib_All (*omrdataset-tools.OmrDataset.OmrDataset attribute*), 6
MScoreLib_Prokofiev (*omrdataset-tools.OmrDataset.OmrDataset attribute*), 7
MScoreLib_Scriabin (*omrdataset-tools.OmrDataset.OmrDataset attribute*), 7
MuscimaPlusPlus_Images (*omrdataset-tools.OmrDataset.OmrDataset attribute*), 7
MuscimaPlusPlus_MeasureAnnotations (*omrdataset-tools.OmrDataset.OmrDataset attribute*), 7

MuscimaPlusPlus_V1 (*omrdataset-tools.OmrDataset.OmrDataset attribute*), 7
MuscimaPlusPlus_V2 (*omrdataset-tools.OmrDataset.OmrDataset attribute*), 7
MuscimaPlusPlusMaskImageGenerator (*class in omrdataset-tools.MuscimaPlusPlusMaskImageGenerator*), 13
MuscimaPlusPlusSymbolImageGenerator (*class in omrdataset-tools.MuscimaPlusPlusSymbolImageGenerator*), 14

N

NODES_SEMANTIC_SEGMENTATION (*omrdataset-tools.MuscimaPlusPlusMaskImageGenerator.MaskType attribute*), 13

O

OmrDataset (*class in omrdataset-tools.OmrDataset*), 5
omrdatasettools
    module, 8
omrdatasettools.Downloader
    module, 3
OpenOmr (*omrdataset-tools.OmrDataset.OmrDataset attribute*), 7
OpenScoreLieder (*omrdataset-tools.OmrDataset.OmrDataset attribute*), 7
OpenScoreStringQuartets (*omrdataset-tools.OmrDataset.OmrDataset attribute*), 7

P

Printed (*omrdataset-tools.OmrDataset.OmrDataset attribute*), 7

R

Rebelo1 (*omrdataset-tools.OmrDataset.OmrDataset attribute*), 7
Rebelo2 (*omrdataset-tools.OmrDataset.OmrDataset attribute*), 7
render_node_masks() (*omrdataset-tools.MuscimaPlusPlusMaskImageGenerator.Muscim method*), 13

S

STAFF_BLOBS_INSTANCE_SEGMENTATION (*omrdataset-tools.MuscimaPlusPlusMaskImageGenerator.MaskTy attribute*), 13
STAFF_LINES_INSTANCE_SEGMENTATION (*omrdataset-tools.MuscimaPlusPlusMaskImageGenerator.MaskTy attribute*), 13