koboextractor Documentation

Release 0.2.0

Heiko Rothkranz

Jun 13, 2020

CONTENTS

CHAPTER

ONE

MODULE CONTENTS

class koboextractor.**KoboExtractor**(*token*, *endpoint*, *debug=False*)

Bases: `object`

Extracts collected data from KoBoToolbox.

This class provides methods to connect to the kpi API of KoBoToolbox, access information about surveys, their questions, choices, and responses.

token

Your authentication token, which can be obtained from https://kf.kobotoolbox.org/token/.

endpoint

The KoBoToolbox kpi API endpoint, e.g. https://kf.kobotoolbox.org/api/v2 or https://kobo.humanitarianresponse.info/api/v2.

debug

Set to True to enable debugging output. Default: False.

get_asset(*asset_uid*)

Gets information on an asset (survey).

Gets all information on an asset (survey) in the associated KoBoToolbox account.

Parameters **asset_uid** (`str`) – Unique ID of the asset. Obtainable e.g. through `list_assets()['results'][i]['uid']` (for your first asset, use `i=0`).

Return type `Dict[str, Any]`

Returns A dict containing information about your asset. Log into KoBoToolbox and visit https://kf.kobotoolbox.org/api/v2/assets/YOUR_ASSET_UID/ to see a description.

get_choices(*asset*)

Groups the choices (answer options) of a survey into a dict.

Groups all the choices (answer options) of a survey into a dict, arranged by their list. A 'sequence' number is added to allow restoring the original order of the choices from the inherently unordered dict.

Parameters **asset** – A dict as returned by `get_asset()`.

Returns

A dict of the form:

```
{
    LIST_NAME: {
        'label': CHOICE_LABEL,
        'sequence': SEQUENCE_NUMBER
    }
}
```

where CHOICE_LABEL is the label (text) of the choice in the survey's default language, and SEQUENCE_NUMBER is an incrementing number that can be used to restore the order of the choices in the survey from this unordered dict.

get_data (*asset_uid*, *query=None*, *start=None*, *limit=None*, *submitted_after=None*)
Gets the data (responses) of an asset (survey).

Gets all information on an asset (survey) in the associated KoBoToolbox account.

Parameters

- **asset_uid** – Unique ID of the asset. Obtainable e.g. through `list_assets()['results'][i]['uid']` (for your first asset, use i=0).
- **query** – Query string in the form `'{"field":"value"}'` or `'{"field":{"op": "value"}}'`, e.g. `'{"_submission_time": {"$gt": "2020-05-14T14:36:20"}}'`. See https://docs.mongodb.com/manual/reference/operator/query/ for operators.
- **start** – Index (zero-based) from which the results start (default: 0).
- **limit** – Number of results per page (max: 30000, default: 30000).
- **submitted_after** – Shorthand to query for submission time. String of date and time in ISO format (e.g. 2020-05-14T14:36:20, results) in query `'{"_submission_time": {"$gt": "2020-05-14T14:36:20"}}'`. Ignored when combined with 'query'.

Returns A dict containing the data associated with the asset. For a survey asset, the key 'count' provides the number of responses. The key 'results' contains a list of responses. Each response is a dict with several metadata keys (such as '_submission_time') and key/value pairs for each answered question in the form of 'GROUP_CODE/QUESTION_CODE': 'ANSWER_CODE'. Log into KoBoToolbox and visit https://kf.kobotoolbox.org/api/v2/assets/YOUR_ASSET_UID/data/ for a more detailed description.

get_questions (*asset*, *unpack_multiples*)
Groups the choices (answer options) of a survey into a dict.

Groups all the choices (answer options) of a survey into a dict, arranged by their list. A 'sequence' number is added to allow restoring the original order of the choices from the inherently unordered dict.

Parameters

- **asset** – A dict as returned by `get_asset()`.
- **unpack_multiples** – If True, the corresponding choices from `get_choices()` are added as subsequent questions following a multiple choice question (type 'select_multiple'). The type of these additional questions is set to 'select_multiple_option'.

Returns

A dict of the form:

```
{
    'groups': {
        GROUP_CODE: {
            'label': GROUP_LABEL,
            'sequence': SEQUENCE_NUMBER,
            'repeat': True/False,
            'questions': {
                QUESTION_CODE: {
                    'type': QUESTION_TYPE,
```

(continues on next page)

(continued from previous page)

```
                        'sequence': SEQUENCE_NUMBER,
                        'label': QUESTION_LABEL,
                        'list_name': CHOICE_LIST_NAME,
                        'choices': {
                            CHOICE_CODE: {
                                'label': CHOICE_LABEL,
                                'type': 'select_multiple_option',
                                'sequence': SEQUENCE_NUMBER
                            }
                        },
                        'other': {
                            'type': '_or_other',
                            'label': 'Other',
                            'sequence': SEQUENCE_NUMBER
                        }
                    }
                },
                'groups': {
                    GROUP_CODE: {
                        ...
                    }
                }
            },
        'questions': {
            QUESTION_CODE: {
                ...
            }
}
```

where GROUP_LABEL, QUESTION_LABEL and CHOICE_LABEL are the labels (text) of the group or question in the survey's default language. SEQUENCE_NUMBER is an incrementing number that can be used to restore the order of the questions in the survey from this unordered dict.

Depending on the question, not all keys may be present.

An additional question of the type '_or_other' is inserted after any question which type ends in '_or_other', to cover the reponses to such questions.

label_result (*unlabeled_result*, *choice_lists*, *questions*, *unpack_multiples*)

Adds labels for questions and answers to a response.

Adds labels corresponding the the question group codes, question codes and answer codes to a response.

Example

```
from KoboExtractor import KoboExtractor
kobo = KoboExtractor(KOBO_TOKEN, 'https://kf.kobotoolbox.org/api/v2')

assets = kobo.list_assets()
asset_uid = assets['results'][0]['uid']
asset = kobo.get_asset(asset_uid)
choice_lists = kobo.get_choices(asset)
questions = kobo.get_questions(asset=asset, unpack_multiples=True)

asset_data = kobo.get_data(asset_uid)
```

(continues on next page)

(continued from previous page)

```
results = kobo.sort_results_by_time(asset_data['results'])
labeled_results = []
for result in results:
    labeled_results.append(kobo.label_result(unlabeled_result=result, choice_
↪lists=choice_lists, questions=questions, unpack_multiples=True))
```

Parameters

- **unlabeled_result** – A single result (dict) of the form:

  ```
  {
      (GROUP_CODES)/)QUESTION_CODE: ANSWER_CODE,
      (GROUP_CODE(S)/)REPEAT_GROUP_CODE: [
          {
              (GROUP_CODE(S)/)REPEAT_GROUP_CODE/(GROUP_CODE(S)/
  ↪)QUESTION_CODE: ANSWER_CODE,
              (GROUP_CODE(S)/)REPEAT_GROUP_CODE/(GROUP_CODE(S)/
  ↪)REPEAT_GROUP_CODE: [
                  ...
              ]
          }
      ],
      METADATA_KEY: METADATA_VALUE
  }
  ```

  (e.g. one of the list items in `get_data(asset_uid)['results']`).

- **choice_lists** – Dict of choice lists as returned by `get_choices(asset)`.
- **questions** – Dict of questions as returned by `get_questions(asset)`
- **unpack_multiples** – If True, the corresponding choices from `get_choices()` are added as subsequent questions following a multiple choice question (type 'select_multiple').

Returns

A dict of the form:

```
{
    'meta': {
        'start': '2020-05-15T08:07:24.705+08:00',
        '_version_': 'vf4kqJPWTbsMrZSw5RZQ7H',
        '_submission_time': '2020-05-15T00:17:51',
        ...
    },
    results: {
        (GROUP_CODE(S)/)QUESTION_CODE: {
            'label': 'Question label',
            'answer_code': ANSWER_CODE,
            'answer_label': 'Answer label',
            'sequence': QUESTION_SEQUENCE,
            'choices': {
                'CHOICE_CODE': {
                    'sequence': CHOICE_SEQUENCE,
                    'label': CHOICE_LABEL,
                    'answer_code': 0 or 1,
                    'answer_label': 'Yes' or 'No'
```

(continues on next page)

(continued from previous page)

```
                    }
                }
            },
            (GROUP_CODE(S)/)REPEAT_GROUP_CODE: {
                0: {
                    (GROUP_CODE(S)/)QUESTION_CODE: {
                        'label': 'Question label',
                        'answer_code': ANSWER_CODE,
                        'answer_label': 'Answer label',
                        'sequence': QUESTION_SEQUENCE
                    },
                    (GROUP_CODE(S)/)QUESTION_CODE: {
                        ...
                    },
                    ...
                },
                1: {
                    ...
                }
            },
            ...
        }
}
```

() denote optional parts, depending on how deep the groups are nested. QUESTION_SEQUENCE reflects the order of the questions (and choices) in the survey.

list_assets()
Lists all assets (surveys).

Lists all assets (surveys) in the associated KoBoToolbox account.

Return type `Dict[str, Any]`

Returns A dict containing information about your assets. Log into KoBoToolbox and visit https://kf.kobotoolbox.org/api/v2/assets/ to see a description.

sort_results_by_time(*unsorted_results*, *reverse=False*)
Sorts an unordered list of responses by their submission time.

Sorts a list of responses in random order (e.g. as obtained by `get_data(asset_uid)['results']` by the value of their `_submission_time` key.

Example:

```
from koboextractor import KoboExtractor
kobo = KoboExtractor(KOBO_TOKEN, 'https://kf.kobotoolbox.org/api/v2')
assets = kobo.list_assets()
asset_uid = assets['results'][0]['uid']
new_data = kobo.get_data(asset_uid)
new_results = kobo.sort_results_by_time(new_data['results'])
```

Parameters

- **unsorted_results** – A list of results as returned by `kobo.get_data(asset_uid)['results']`.
- **reverse** – If True, sort in descending order. Default: False.

Returns A list of results as provided in `unsorted_results`, but sorted by the value of their `_submission_time` key.

CHAPTER

TWO

KOBOEXTRACTOR

This Python package provides a wrapper around part of the KoBoToolbox kpi API, with the main goal being to ease the downloading of survey responses. It provides methods to download data from the KoBoToolbox kpi API (e.g. https://kf.kobotoolbox.org/ or https://kc.humanitarianresponse.info/) and to rearrange this data into useable structures.

2.1 Installation

KoboExtractor requires Python 3.6+.

Simply install from PyPI with:

```
pip3 install koboextractor
```

2.2 Example usage

In this example, response data is downloaded from KoBoToolbox and arranged in a form that is convenient for further processing, e.g. for storing in a different database or uploading to Google Sheets.

Import and initialise the KoboExtractor:

```
from koboextractor import KoboExtractor
kobo = KoboExtractor(KOBO_TOKEN, 'https://kf.kobotoolbox.org/api/v2', debug=debug)
```

Get the unique ID of the first asset in your KoBoToolbox account:

```
assets = kobo.list_assets()
asset_uid = assets['results'][0]['uid']
```

Information on the questions and choices in your survey can be obtained with:

```
asset = kobo.get_asset(asset_uid)
choice_lists = kobo.get_choices(asset)
questions = kobo.get_questions(asset=asset, unpack_multiples=True)
```

`questions` is a dictionary of the form:

```
    {
    'groups': {
        GROUP_CODE: {
            'label': GROUP_LABEL,
```

(continues on next page)

(continued from previous page)

```
            'sequence': SEQUENCE_NUMBER,
            'repeat': True/False,
            'questions': {
                QUESTION_CODE: {
                    'type': QUESTION_TYPE,
                    'sequence': SEQUENCE_NUMBER,
                    'label': QUESTION_LABEL,
                    'list_name': CHOICE_LIST_NAME,
                    'choices': {
                        CHOICE_CODE: {
                            'label': CHOICE_LABEL,
                            'type': 'select_multiple_option',
                            'sequence': SEQUENCE_NUMBER
                        }
                    },
                    'other': {
                        'type': '_or_other',
                        'label': 'Other',
                        'sequence': SEQUENCE_NUMBER
                    }
                }
            },
            'groups': {
                GROUP_CODE: {
                    ...
                }
            }
        },
    'questions': {
        QUESTION_CODE: {
            ...
        }
}
```

`choices` is a dictionary of the form:

```
{
        LIST_NAME: {
                'label': CHOICE_LABEL,
                'sequence': SEQUENCE_NUMBER
        }
}
```

Download all responses submitted after a certain point in time:

```
new_data = kobo.get_data(asset_uid, submitted_after='2020-05-20T17:29:30')
```

The number of downloaded results is available in `new_data['count']`.

`new_data` will be an unordered list of form submissions. We can sort this list by submission time by calling:

```
new_results = kobo.sort_results_by_time(new_data['results'])
```

Each response (list item) is a dict with several metadata keys (such as '_submission_time') and key/value pairs for each answered question in the form of 'GROUP_CODE/QUESTION_CODE': 'ANSWER_CODE'. Map the question and answer labels from your survey onto the coded answers in the responses:

```
labeled_results = []
for result in new_results: # new_results is a list of list of dicts
        # Unpack answers to select_multiple questions
        labeled_results.append(kobo.label_result(unlabeled_result=result, choice_
↪lists=choice_lists, questions=questions, unpack_multiples=True))
```

2.3 Documentation

The full documentation is available at https://koboextractor.readthedocs.io .

PYTHON MODULE INDEX

INDEX