dexparser

Apr 23, 2023

Contents:

CHAPTER 1

dexparser

1.1 Modules

class dexparser.**AABParser**(*filedir=None*, *fileobj=None*, *deepscan=False*)
Bases: *dexparser.APKParser*

AAB (Android App Bundle) file format parser class :param string filedir: AAB file path :param bytes fileobj: AAB file object :param boolean deepscan: Scan all assets of AAB file for detect adex file

class dexparser.**APKParser**(*filedir=None*, *fileobj=None*, *deepscan=False*)
Bases: object

APK file format parser class :param string filedir: APK file path :param bytes fileobj: APK file object :param boolean deepscan: Scan all assets of APK file for detect adex file

get_all_dex_filenames()
Get all name of dex files :returns: list of dex filenames

example:

```
>>> APKParser(filedir='path/to/file.apk').get_all_dex_filenames()
['classes.dex', 'classes1.dex']
```

get_dex(*filename='classes.dex'*)
Get dex file with DEX parsed object

Params name of dexfile (default: classes.dex)

Returns DEXParser object

example:

```
>>> APKParser(filedir='path/to/file.apk').get_dex()
True
```

is_multidex
Detect if APK is a multidex https://developer.android.com/studio/build/multidex

Returns boolean

example:

```
>>> APKParser(filedir='path/to/file.apk').is_multidex
True
```

class dexparser.**DEXParser**(*filedir=None*, *fileobj=None*)
Bases: *dexparser.Dexparser*

DEX file format parser subclass :param string filedir: DEX file path :param bytes fileobj: DEX file object

class dexparser.**Dexparser**(*filedir=None*, *fileobj=None*)
Bases: object

DEX file format parser class :param string filedir: DEX file path :param bytes fileobj: DEX file object

checksum
Get checksum value of DEX file

Returns hexlify value of checksum

example:

```
>>> Dexparser(filedir='path/to/classes.dex').checksum
0x30405060
```

get_annotations(*offset*)
Get annotation data from DEX file

Parameters **offset** (*integer*) – annotation_off offset value

Returns specific data of annotation

example:

```
>>> dex = Dexparser(filedir='path/to/classes.dex')
>>> dex.get_annotations(offset=3022)
{
    'visibility': 3403,
    'type_idx_diff': 3024,
    'size_diff': 64,
    'name_idx_diff': 30,
    'value_type': 302,
    'encoded_value': 7483
}
```

get_class_data(*offset*)
Get class specific data from DEX file

Parameters **offset** (*integer*) – class_idx offset value

Returns specific data of class

example:

```
>>> dex = Dexparser(filedir='path/to/classes.dex')
>>> dex.get_class_data(offset=3022)
{
    'static_fields': [
        {
            'diff': 30, 'access_flags': 4000
        }
    ],
    'instance_fields': [
        {
            'diff': 32, 'access_flags': 4000
        }
    ],
    'direct_methods': [
        {
            'diff': 30, 'access_flags': 4000, 'code_off': 384304
        }
    ],
    'virtual_methods': [
        {
            'diff': 63, 'access_flags': 4000, 'code_off': 483933
        }
    ]
}
```

get_classdef_data()
Get class definition data from DEX file

Returns list of class definition data extracted from class_def_item

example:

```
>>> dex = Dexparser(filedir='path/to/classes.dex')
>>> dex.get_classdef_data()
[
    {
        'class_idx': 3049,
        'access_flags': 4000,
        'superclass_idx': 200,
        'interfaces_off': 343,
        'source_file_idx': 3182,
        'annotation_off': 343,
        'class_data_off': 345,
        'static_values_off': 8830
    },
    ...
]
```

get_fieldids()
Get field idx from DEX file

Returns list of field ids defined at field_id_item

example:

```
>>> dex = Dexparser(filedir='path/to/classes.dex')
>>> dex.get_fieldids()
[{'class_idx': 339, 'type_idx': 334, 'name_idx': 340}, ...]
```

get_methods()
Get methods from DEX file

Returns list of methods defined at DEX file

example:

```
>>> dex = Dexparser(filedir='path/to/classes.dex')
>>> dex.get_methods()
[{'class_idx': 132, 'proto_idx': 253, 'name_idx': 3005}, ...]
```

get_protoids()
Get proto idx from DEX file

Returns list of proto ids defined at proto_id_item

example:

```
>>> dex = Dexparser(filedir='path/to/classes.dex')
>>> dex.get_protoids()
[{'shorty_idx': 3000, 'return_type_idx': 330, 'param_off': 0}, ...]
```

get_static_values(*offset*)
Get all static values parsed from 'static_values_off' classdef_data section.

Parameters **offset** (*integer*) – static_values_off offset value

Returns specific data of static values

example:

```
>>> dex = Dexparser(filedir='path/to/classes.dex')
>>> dex.get_static_values(offset=3022)
[b'android.annotation', 0.0, False, None]
```

get_strings()
Get string items from DEX file

Returns strings extracted from string_data_item section

example:

```
>>> dex = Dexparser(filedir='path/to/classes.dex')
>>> dex.get_strings()
['Ljava/utils/getJavaUtils', ...]
```

get_typeids()
Get type ids from DEX file

Returns descriptor_idx extracted from type_id_item section

example:

```
>>> dex = Dexparser(filedir='path/to/classes.dex')
>>> dex.get_typeids()
[133, 355, 773, 494, ...]
```

header

Get header data from DEX

Returns header data

example:

```
>>> Dexparser(filedir='path/to/classes.dex').header
{'magic': 'dex5' ...}
```

CHAPTER 2

Indices and tables

- genindex
- modindex
- search

Python Module Index

d

Index