

---

# **fhirpath-guillotina Documentation**

***Release 0.4.0***

**Md Nazrul Islam**

**May 16, 2020**



## CONTENTS:

<b>1</b>	<b>Introduction</b>	<b>1</b>
1.1	Quickstart . . . . .	1
1.2	Credits . . . . .	3
<b>2</b>	<b>Installation</b>	<b>5</b>
2.1	Stable release . . . . .	5
2.2	From sources . . . . .	5
<b>3</b>	<b>Usage</b>	<b>7</b>
<b>4</b>	<b>fhirpath_guillotina</b>	<b>9</b>
4.1	fhirpath_guillotina package . . . . .	9
<b>5</b>	<b>Contributing</b>	<b>13</b>
5.1	Types of Contributions . . . . .	13
5.2	Get Started! . . . . .	14
5.3	Pull Request Guidelines . . . . .	15
5.4	Tips . . . . .	15
5.5	Deploying . . . . .	15
<b>6</b>	<b>Credits</b>	<b>17</b>
6.1	Development Lead . . . . .	17
6.2	Contributors . . . . .	17
<b>7</b>	<b>History</b>	<b>19</b>
7.1	0.4.0 (2020-05-16) . . . . .	19
7.2	0.3.0 (2019-11-13) . . . . .	19
7.3	0.2.0 (2019-10-10) . . . . .	19
7.4	0.1.0 (2019-08-24) . . . . .	19
<b>8</b>	<b>Indices and tables</b>	<b>21</b>
	<b>Python Module Index</b>	<b>23</b>
	<b>Index</b>	<b>25</b>



## INTRODUCTION

A guillotina framework powered fhirpath provider. Full battery included to use [fhirpath](#) more efficiently.

If you don't know about FHIR® and HL7®, Please [have a look here](#).

### 1.1 Quickstart

1. Make sure `fhirpath_guillotina` is added as addon in your guillotina configure file.
2. Make sure elasticsearch server configured properly.
3. Make sure you have FHIR resource contenttypes registered (see example bellow).
4. Make sure you have FHIR resources mapping (correct version) for elasticsearch.

Example: Add Contents:

```
>>> from fhirpath_guillotina.field import FhirField

class IOrganization(IFhirContent, IContentIndex):
    index_field(
        "organization_resource",
        type="object",
        field_mapping=fhir_resource_mapping("Organization"),
        fhirpath_enabled=True,
        resource_type="Organization",
        fhir_release=FHIR_VERSION.R4,
    )
    index_field("org_type", type="keyword")
    org_type = TextLine(title="Organization Type", required=False)
    organization_resource = FhirField(
        title="Organization Resource", resource_type="Organization", fhir_release="R4"
    )

@configure.contenttype(type_name="Organization", schema=IOrganization)
class Organization(Folder):
    """ """
```

(continues on next page)

(continued from previous page)

```
index(schemas=[IOrganization], settings={})
resource_type = "Organization"
```

Example Search:

```
>>> from guillotina.component import query_utility
>>> from fhirpath.interfaces import ISearchContextFactory
>>> from fhirpath.interfaces import IFhirSearch
>>> search_context = query_utility(ISearchContextFactory).get(
...     resource_type="Organization"
... )
>>> search_tool = query_utility(IFhirSearch)
>>> params = (
...     ("active", "true"),
...     ("_lastUpdated", "2010-05-28T05:35:56+00:00"),
...     ("_profile", "http://hl7.org/fhir/Organization"),
...     ("identifier", "urn:oid:2.16.528.1|91654"),
...     ("type", "http://hl7.org/fhir/organization-type|prov"),
...     ("address-postalcode", "9100 AA")
... )
>>> fhir_bundle = await search_tool(params, context=search_context)
>>> fhir_bundle.total == len(fhir_bundle.entry)
```

Example FhirPath Query:

```
>>> from fhirpath.interfaces import IElasticsearchEngineFactory
>>> from guillotina.component import query_utility
>>> from fhirpath.enums import SortOrderType
>>> from fhirpath.fql import Q_
>>> from fhirpath.fql import T_
>>> from fhirpath.fql import V_
>>> from fhirpath.fql import sort_
>>> engine = query_utility(IElasticsearchEngineFactory).get()
>>> query_builder = Q_(resource="Organization", engine=engine)
>>> query_builder = (
...     query_builder.where(T_("Organization.active") == V_("true"))
...     .where(T_("Organization.meta.lastUpdated", "2010-05-28T05:35:56+00:00"))
...     .sort(sort_("Organization.meta.lastUpdated", SortOrderType.DES))
...     .limit(20)
... )
>>> query_result = query_builder(async_result=True)
>>> result = query_result.fetchall()
>>> result.header.total == 100
True
>>> len(result.body) == 20
True
>>> async for resource in query_result:
...     assert resource.resource_type == "Organization"
```

- Free software: GNU General Public License v3
- Documentation: <https://fhirpath-guillotina.readthedocs.io>.

## 1.2 Credits

This package skeleton was created with [Cookiecutter](#) and the [audreyr/cookiecutter-pypackage](#) project template.





## INSTALLATION

### 2.1 Stable release

To install fhirpath-guillotina, run this command in your terminal:

```
$ pip install fhirpath_guillotina
```

This is the preferred method to install fhirpath-guillotina, as it will always install the most recent stable release.

If you don't have `pip` installed, this [Python installation guide](#) can guide you through the process.

### 2.2 From sources

The sources for fhirpath-guillotina can be downloaded from the [Github repo](#).

You can either clone the public repository:

```
$ git clone git://github.com/nazrulworld/fhirpath_guillotina
```

Or download the [tarball](#):

```
$ curl -OL https://github.com/nazrulworld/fhirpath_guillotina/tarball/master
```

Once you have a copy of the source, you can install it with:

```
$ python setup.py install
```



## USAGE

To use fhirpath-guillotina in a project:

```
import fhirpath_guillotina
```



## FHIRPATH\_GUILLOTINA

### 4.1 fhirpath\_guillotina package

#### 4.1.1 Submodules

#### 4.1.2 fhirpath\_guillotina.engine module

```
class fhirpath_guillotina.engine.ElasticsearchConnection(conn)
    Bases: fhirpath.connectors.factory.es.ElasticsearchConnection
    Elasticsearch Connection

    async count (index, compiled_query)

    async fetch (index, compiled_query)
        xxx: must have use scroll+slice https://stackoverflow.com/questions/43211387/
        what-does-elasticsearch-automatic-slicing-do https://stackoverflow.com/questions/50376713/
        elasticsearch-scroll-api-with-multi-threading

    classmethod from_url (url: str)

    async scroll (scroll_id, scroll='30s')

    async server_info ()

class fhirpath_guillotina.engine.ElasticsearchEngine(fhir_release, conn_factory, dialect_factory)
    Bases: fhirpath.engine.es.ElasticsearchEngine
    Elasticsearch Engine

    build_security_query (query)

    calculate_field_index_name (resource_type=None, index_config=None)
        1.) xxx: should be cached

    async execute (query, unrestricted=False, query_type=<EngineQueryType.DML: 'DML'>)

    extract_hits (selects, hits, container)

    static field_index_config_from_factory (factory, resource_type=None)

    async get_index_name (container=None)

    get_mapping (resource_type=None, index_config=None)

    async process_raw_result (rawresult, selects)

    wrapped_with_bundle (result)
```

### 4.1.3 fhirpath\_guillotina.field module

```
class fhirpath_guillotina.field.DefaultFhirFieldSchemaSerializer(field, schema,  
                                                                request)  
    Bases: guillotina.json.serialize_schema_field.DefaultSchemaFieldSerializer  
  
    property field_type
```

```
class fhirpath_guillotina.field.FhirField(resource_class=None,                                re-  
                                         source_interface=None, resource_type=None,  
                                         fhir_release=None, **kw)  
  
    Bases: guillotina.schema._field.Object
```

FhirResource also known as FHIR field is the schema field derived from z3c.form's field.

It takes all initial arguments those are derived from standard schema field, with additionally `model`, `resource_type` and `resource_interface`

---

**Note:** field name must be start with lowercase name of FHIR Resource.

---

```
ensure_fhir_abstract (klass)
```

```
from_dict (dict_value)
```

```
from_unicode (str_val)
```

```
class fhirpath_guillotina.field.FhirFieldValue(obj: NewType.<locals>.new_type =  
                                              None)
```

Bases: object

FhirResourceValue is a proxy class for holding any object derived from `fhir.resources.resource.Resource`

```
foreground_origin ()
```

Return the original object of FHIR model that is proxied!

```
patch (patch_data)
```

```
stringify (prettyify=False)
```

```
fhirpath_guillotina.field.fhir_field_deserializer(fhirfield, value, context=None)
```

```
fhirpath_guillotina.field.fhir_field_from_resource_type(resource_type: str, cache:  
                                                         bool = True) → Op-  
                                                         tional[dict]
```

```
fhirpath_guillotina.field.fhir_field_from_schema(schema: <InterfaceClass  
                                                  zope.interface.Interface>, re-  
                                                  source_type: str = None) → Op-  
                                                  tional[fhirpath_guillotina.field.FhirField]
```

```
fhirpath_guillotina.field.fhir_field_value_serializer(value)
```

#### 4.1.4 fhirpath\_guillotina.helpers module

`fhirpath_guillotina.helpers.parse_json_str(str_val: str) → Optional[dict]`

#### 4.1.5 fhirpath\_guillotina.interfaces module

Module where all interfaces, events and exceptions live.

#### 4.1.6 fhirpath\_guillotina.utilities module

**class** `fhirpath_guillotina.utilities.ElasticsearchEngineFactory`

Bases: `object`

**get** (*fhir\_release=None*)

**class** `fhirpath_guillotina.utilities.FhirSearch`

Bases: `object`

**create\_context** (*resource\_type, fhir\_release=None, unrestricted=False*)

**class** `fhirpath_guillotina.utilities.SearchContextFactory`

Bases: `object`

**get** (*resource\_type, fhir\_release=None, unrestricted=False*)

`fhirpath_guillotina.utilities.create_engine(fhir_release=None)`

`fhirpath_guillotina.utilities.default_settings()`

#### 4.1.7 Module contents

`fhirpath_guillotina.includeme(root)`





## CONTRIBUTING

Contributions are welcome, and they are greatly appreciated! Every little bit helps, and credit will always be given. You can contribute in many ways:

### 5.1 Types of Contributions

#### 5.1.1 Report Bugs

Report bugs at [https://github.com/nazrulworld/fhirpath\\_guillotina/issues](https://github.com/nazrulworld/fhirpath_guillotina/issues).

If you are reporting a bug, please include:

- Your operating system name and version.
- Any details about your local setup that might be helpful in troubleshooting.
- Detailed steps to reproduce the bug.

#### 5.1.2 Fix Bugs

Look through the GitHub issues for bugs. Anything tagged with “bug” and “help wanted” is open to whoever wants to implement it.

#### 5.1.3 Implement Features

Look through the GitHub issues for features. Anything tagged with “enhancement” and “help wanted” is open to whoever wants to implement it.

#### 5.1.4 Write Documentation

fhirpath-guillotina could always use more documentation, whether as part of the official fhirpath-guillotina docs, in docstrings, or even on the web in blog posts, articles, and such.

### 5.1.5 Submit Feedback

The best way to send feedback is to file an issue at [https://github.com/nazrulworld/fhirpath\\_guillotina/issues](https://github.com/nazrulworld/fhirpath_guillotina/issues).

If you are proposing a feature:

- Explain in detail how it would work.
- Keep the scope as narrow as possible, to make it easier to implement.
- Remember that this is a volunteer-driven project, and that contributions are welcome :)

## 5.2 Get Started!

Ready to contribute? Here's how to set up *fhirpath\_guillotina* for local development.

1. Fork the *fhirpath\_guillotina* repo on GitHub.

2. Clone your fork locally:

```
$ git clone git@github.com:your_name_here/fhirpath_guillotina.git
```

3. Install your local copy into a virtualenv. Assuming you have `pipenv` installed, this is how you set up your fork for local development:

```
$ cd fhirpath_guillotina/  
$ pipenv install --dev -e .[test]  
$ pipenv shell  
$ pytest tests
```

4. Create a branch for local development:

```
$ git checkout -b name-of-your-bugfix-or-feature
```

Now you can make your changes locally.

5. When you're done making changes, check that your changes pass `flake8` and the tests, including testing other Python versions with `tox`:

```
$ make lint  
$ pytest tests  
$ tox
```

To get `flake8` and `tox`, just `pip` install them into your virtualenv.

6. Commit your changes and push your branch to GitHub:

```
$ git add .  
$ git commit -m "Your detailed description of your changes."  
$ git push origin name-of-your-bugfix-or-feature
```

7. Submit a pull request through the GitHub website.

## 5.3 Pull Request Guidelines

Before you submit a pull request, check that it meets these guidelines:

1. The pull request should include tests.
2. If the pull request adds functionality, the docs should be updated. Put your new functionality into a function with a docstring, and add the feature to the list in README.rst.
3. The pull request should work for Python 3.7, 3.8, and optionally for PyPy. Check [https://travis-ci.org/nazrulworld/fhirpath\\_guillotina/pull\\_requests](https://travis-ci.org/nazrulworld/fhirpath_guillotina/pull_requests) and make sure that the tests pass for all supported Python versions.

## 5.4 Tips

To run a subset of tests:

```
$ pytest tests.test_fhirpath_guillotina
```

## 5.5 Deploying

A reminder for the maintainers on how to deploy. Make sure all your changes are committed (including an entry in HISTORY.rst). Then run:

```
$ bumpversion patch # possible: major / minor / patch
$ git push
$ git push --tags
```

Travis will then deploy to PyPI if tests pass.



## CREDITS

### 6.1 Development Lead

- Md Nazrul Islam <[email2nazrul@gmail.com](mailto:email2nazrul@gmail.com)>

### 6.2 Contributors

None yet. Why not be the first?



## HISTORY

### 7.1 0.4.0 (2020-05-16)

#### Improvements

- supports `fhirpath` 0.6.1 or later version.
- supports `guillotina` 5.3.x but not 6.x.x

### 7.2 0.3.0 (2019-11-13)

#### Improvements

- supports <https://pypi.org/project/fhirpath/> 0.4.0 or later version.

### 7.3 0.2.0 (2019-10-10)

- Full (almost) featured [FHIR Search R4](#) supports.

### 7.4 0.1.0 (2019-08-24)

- First release on PyPI. [nazrulworld]





## INDICES AND TABLES

- `genindex`
- `modindex`
- `search`



## PYTHON MODULE INDEX

### f

- `fhirpath_guillotina`, [11](#)
- `fhirpath_guillotina.engine`, [9](#)
- `fhirpath_guillotina.field`, [10](#)
- `fhirpath_guillotina.helpers`, [11](#)
- `fhirpath_guillotina.interfaces`, [11](#)
- `fhirpath_guillotina.utilities`, [11](#)



## INDEX

### B

`build_security_query()`  
(*fhirpath\_guillotina.engine.ElasticsearchEngine*  
method), 9

### C

`calculate_field_index_name()`  
(*fhirpath\_guillotina.engine.ElasticsearchEngine*  
method), 9

`count()` (*fhirpath\_guillotina.engine.ElasticsearchConnection*  
method), 9

`create_context()` (*fhirpath\_guillotina.utilities.FhirSearch*  
method), 11

`create_engine()` (in module  
*fhirpath\_guillotina.utilities*), 11

### D

`default_settings()` (in module  
*fhirpath\_guillotina.utilities*), 11

`DefaultFhirFieldSchemaSerializer` (class in  
*fhirpath\_guillotina.field*), 10

### E

`ElasticsearchConnection` (class in  
*fhirpath\_guillotina.engine*), 9

`ElasticsearchEngine` (class in  
*fhirpath\_guillotina.engine*), 9

`ElasticsearchEngineFactory` (class in  
*fhirpath\_guillotina.utilities*), 11

`ensure_fhir_abstract()`  
(*fhirpath\_guillotina.field.FhirField* method),  
10

`execute()` (*fhirpath\_guillotina.engine.ElasticsearchEngine*  
method), 9

`extract_hits()` (*fhirpath\_guillotina.engine.ElasticsearchEngine*  
method), 9

### F

`fetch()` (*fhirpath\_guillotina.engine.ElasticsearchConnection*  
method), 9

`fhir_field_deserializer()` (in module  
*fhirpath\_guillotina.field*), 10

`fhir_field_from_resource_type()` (in mod-  
ule *fhirpath\_guillotina.field*), 10

`fhir_field_from_schema()` (in module  
*fhirpath\_guillotina.field*), 10

`fhir_field_value_serializer()` (in module  
*fhirpath\_guillotina.field*), 10

`FhirField` (class in *fhirpath\_guillotina.field*), 10

`FhirFieldValue` (class in *fhirpath\_guillotina.field*),  
10

`fhirpath_guillotina`  
module, 11

`fhirpath_guillotina.engine`  
module, 9

`fhirpath_guillotina.field`  
module, 10

`fhirpath_guillotina.helpers`  
module, 11

`fhirpath_guillotina.interfaces`  
module, 11

`fhirpath_guillotina.utilities`  
module, 11

`FhirSearch` (class in *fhirpath\_guillotina.utilities*), 11

`field_index_config_from_factory()`  
(*fhirpath\_guillotina.engine.ElasticsearchEngine*  
static method), 9

`field_type()` (*fhirpath\_guillotina.field.DefaultFhirFieldSchemaSeriali*  
property), 10

`foreground_origin()`  
(*fhirpath\_guillotina.field.FhirFieldValue*  
method), 10

`from_dict()` (*fhirpath\_guillotina.field.FhirField*  
method), 10

`from_unicode()` (*fhirpath\_guillotina.field.FhirField*  
method), 10

`from_url()` (*fhirpath\_guillotina.engine.ElasticsearchConnection*  
class method), 9

### G

`get()` (*fhirpath\_guillotina.utilities.ElasticsearchEngineFactory*  
method), 11

`get()` (*fhirpath\_guillotina.utilities.SearchContextFactory*  
method), 11

`get_index_name()` (*fhirpath\_guillotina.engine.ElasticsearchEngine method*), 9  
`get_mapping()` (*fhirpath\_guillotina.engine.ElasticsearchEngine method*), 9

## I

`includeme()` (*in module fhirpath\_guillotina*), 11

## M

`module`  
    *fhirpath\_guillotina*, 11  
    *fhirpath\_guillotina.engine*, 9  
    *fhirpath\_guillotina.field*, 10  
    *fhirpath\_guillotina.helpers*, 11  
    *fhirpath\_guillotina.interfaces*, 11  
    *fhirpath\_guillotina.utilities*, 11

## P

`parse_json_str()` (*in module fhirpath\_guillotina.helpers*), 11  
`patch()` (*fhirpath\_guillotina.field.FhirFieldValue method*), 10  
`process_raw_result()` (*fhirpath\_guillotina.engine.ElasticsearchEngine method*), 9

## S

`scroll()` (*fhirpath\_guillotina.engine.ElasticsearchConnection method*), 9  
`SearchContextFactory` (*class in fhirpath\_guillotina.utilities*), 11  
`server_info()` (*fhirpath\_guillotina.engine.ElasticsearchConnection method*), 9  
`stringify()` (*fhirpath\_guillotina.field.FhirFieldValue method*), 10

## W

`wrapped_with_bundle()` (*fhirpath\_guillotina.engine.ElasticsearchEngine method*), 9