
AnyBlok IO Documentation

Release 1.2.0

Jean-Sebastien SUZANNE

Mar 13, 2023

Contents

1	Front Matter	3
1.1	Project Homepage	3
1.2	Installation	3
1.3	Running Tests	3
1.4	Contributing (hackers needed!)	4
1.5	Author	4
1.6	Contributors	4
1.7	Bugs	4
2	MEMENTO	5
2.1	Blok	5
3	Builtin Bloks	7
3.1	Blok anyblok-io	7
3.2	Blok anyblok-io-csv	18
3.3	Blok anyblok-io-xml	23
4	CHANGELOG	29
4.1	2.0.0 (unreleased)	29
4.2	1.2.0 (2021-08-16)	29
4.3	1.1.1 (2020-10-16)	29
4.4	1.1.0 (2020-05-06)	29
4.5	1.0.0 (2018-05-30)	30
5	Mozilla Public License Version 2.0	31
5.1	1. Definitions	31
5.2	2. License Grants and Conditions	33
5.3	3. Responsibilities	34
5.4	4. Inability to Comply Due to Statute or Regulation	35
5.5	5. Termination	35
5.6	6. Disclaimer of Warranty	35
5.7	7. Limitation of Liability	36
5.8	8. Litigation	36
5.9	9. Miscellaneous	36
5.10	10. Versions of the License	36
5.11	Exhibit A - Source Code Form License Notice	37
5.12	Exhibit B - “Incompatible With Secondary Licenses” Notice	37

6 Indices and tables	39
Python Module Index	41
Index	43

AnyBlok is a Python framework allowing to create highly dynamic and modular applications on top of the SQLAlchemy ORM. Applications are made of “bloks” that can be installed, extended, replaced, upgraded or uninstalled.

AnyBlok is released under the terms of the *Mozilla Public License version 2*.

Contents

- *Front Matter*
 - *Project Homepage*
 - *Installation*
 - *Running Tests*
 - *Contributing (hackers needed!)*
 - *Author*
 - *Contributors*
 - *Bugs*

Information about the AnyBlok project.

1.1 Project Homepage

AnyBlok is hosted on [github](#) - the main project page is at http://github.com/AnyBlok/anyblok_io. Source code is tracked here using [GIT](#).

Releases and project status are available on Pypi at http://pypi.python.org/pypi/anyblok_io.

The most recent published version of this documentation should be at <https://anyblok-io.readthedocs.io/en/latest/>.

1.2 Installation

Install released versions of AnyBlok from the Python package index with [pip](#) or a similar tool:

```
pip install anyblok_io
```

1.3 Running Tests

To run framework tests with `pytest` and PostgreSQL:

```
pip install pytest pytest-cov
ANYBLOK_DATABASE_DRIVER=postgresql ANYBLOK_DATABASE_NAME=anyblok_test pytest anyblok_
↪ io/tests
```

To run bloks tests with `pytest` and PostgreSQL:

```
pip install pytest pytest-cov
ANYBLOK_DATABASE_DRIVER=postgresql ANYBLOK_DATABASE_NAME=anyblok_test anyblok_
↪createdb --install-all-bloks
ANYBLOK_DATABASE_DRIVER=postgresql ANYBLOK_DATABASE_NAME=anyblok_test pytest anyblok_
↪io/tests
```

AnyBlok is tested continuously using Github actions

1.4 Contributing (hackers needed!)

Anyblok is at a very early stage, feel free to fork, talk with core dev, and spread the word!

1.5 Author

Jean-Sébastien Suzanne

1.6 Contributors

- Jean-Sébastien Suzanne

1.7 Bugs

Bugs and feature enhancements to AnyBlok should be reported on the [Issue tracker](#).

Contents

- *MEMENTO*
 - *Blok*

2.1 Blok

The **BlokImporter** give some behaviour to help to import file from the blok:

```
from anyblok.blok import Blok
from anyblok_io.blok import BlokImporter

class MyBlok(BlokImporter, Blok):

    required = ['anyblok-io-csv', 'anyblok-io-xml']

    def upload(self, latest_version=None):
        self.import_file_csv(Model, *file_path, **ImporterOptions) # need anyblok-io-
↪CSV
        self.import_file_xml(Model, *file_path, **ImporterOptions) # need anyblok-io-
↪xml
```


Covered Bloks

- *Blok anyblok-io*
- *Blok anyblok-io-csv*
- *Blok anyblok-io-xml*

3.1 Blok anyblok-io

```
class anyblok_io.bloks.io.AnyBlokIO (registry)
```

```
    Bases: anyblok.blok.Blok
```

```
    In / Out tool's:
```

- Formater: convert value 2 str or str 2 value in function of the field,
- Importer: main model to define an import,
- Exporter: main model to define an export,

```
    name = 'anyblok-io'
```

```
    version = '2.0.0'
```

```
    author = 'Suzanne Jean-Sébastien'
```

```
    autoinstall = False
```

```
    priority = 100
```

Note: Require the anyblok-io blok

3.1.1 Mapping

`Model.IO.Mapping` allows to link a `Model` instance by a `Model` namespace and str key. this key is an external *ID*

Save an instance with a key:

```
Blok = self.registry.System.Blok
blok = Blok.query().filter(Blok.name == 'anyblok-core').first()
self.registry.IO.Mapping.set('External ID', blok)
```

Warning: By default if you save another instance with the same key and the same model, an `IOMappingSetException` will be raised. If really you want this mapping you must call the set method with the named argument **`raiseifexist=False`**:

```
self.registry.IO.Mapping.set('External ID', blok, raiseifexist=False)
```

Get an entry in the mapping:

```
blok2 = self.registry.IO.Mapping.get('Model.System.Blok', 'External ID')
assert blok2 is blok
```

3.1.2 Formater

The goal of the formater is to get:

- value from string
- value from mapping key
- string from value
- mapping key from value

The value is the value of the field.

Warning: The relationships are particular cases. The value is the json of the primary keys. The `Many2Many` and the `One2Many` are the json of the list of the primary keys

3.1.3 Exporter

The `Model.IO.Exporter` export some entries in fonction of configuration. `anyblok-io` blok doesn't give complete exporter, just the base `Model` to standardize all the possible export:

```
exporter = registry.IO.Exporter.insert(...) # create a exporter
entries = ... # entries are instance of model
fp = exporter.run(entries)
# fp is un handler on the opened file (StringIO)
```

3.1.4 Importer

The `Model.IO.Importer` import some entries in fonction of configuration. `anyblok-io` blok doesn't give complete importer, just the base `Model` to standardize all the possible import:

```
importer = registry.IO.Importer.insert(...) # create an importer
# the file to import are filled in the parameter
entries = importer.run()
```

3.1.5 API doc

exceptions

exception `anyblok_io.bloks.io.exceptions.IOException`
 Bases: `Exception`

IO exception

exception `anyblok_io.bloks.io.exceptions.IOMappingCheckException`
 Bases: `anyblok_io.bloks.io.exceptions.IOException`

IO Exception for setter

exception `anyblok_io.bloks.io.exceptions.IOMappingSetException`
 Bases: `anyblok_io.bloks.io.exceptions.IOException`

IO Exception for setter

exception `anyblok_io.bloks.io.exceptions.ImporterException`
 Bases: `Exception`

Simple Exception for importer

exception `anyblok_io.bloks.io.exceptions.ExporterException`
 Bases: `Exception`

Simple Exception for exporter

exception `anyblok_io.bloks.io.exceptions.FormatterException`
 Bases: `Exception`

Simple Exception for importer

core

class `anyblok_io.bloks.io.core.SqlBase`
 Bases: `object`

AnyBlok registration:

- Type: Core
- Registry name: `Core.SqlBase`

delete (**args, **kwargs*)
 Inherit the `Model.delete` methods.:

```
instance.delete(remove_mapping=True)
```

Parameters `remove_mapping` – boolean, if check (default) the mapping is removed

mapping

class anyblok_io.bloks.io.mapping.**Mapping**

Bases: object

AnyBlok registration:

- Type: Model
- Registry name: Model.IO.Mapping
- Tablename: io_mapping

Fields	
key	<ul style="list-style-type: none">• Type - anyblok.column.String• primary_key - True• default - anyblok.column.NoDefaultValue• size - 64
model	<ul style="list-style-type: none">• Type - anyblok.column.String• primary_key - True• foreign_key - Model.System.Model => name• default - anyblok.column.NoDefaultValue• size - 256
primary_key	<ul style="list-style-type: none">• Type - anyblok.column.Json• nullable - False• default - anyblok.column.NoDefaultValue
blokname	<ul style="list-style-type: none">• Type - anyblok.column.String• Label - 'Blok name'• foreign_key - Model.System.Blok => name• default - anyblok.column.NoDefaultValue• size - 64

classmethod **check_primary_keys** (*model*, **pks*)

check if the all the primary keys match with primary keys of the model

Parameters

- **model** – model to check
- **pks** – list of the primary keys to check

Exception IOMappingCheckException

classmethod **clean** (*bloknames=None*, *models=None*)

Clean all mapping with removed object linked:

```
Mapping.clean(bloknames=['My blok'])
```

Warning: For filter only the no blokname:

```
Mapping.clean(bloknames=[None])
```

Params bloknames filter by blok, keep the order to remove the mapping

Params models filter by model, keep the order to remove the mapping

classmethod delete (*model, key, mapping_only=True, byquery=False*)

Delete the key for this model

Parameters

- **model** – model of the mapping
- **key** – string of the key

Return type Boolean True if the mapping is removed

classmethod delete_for_blokname (*blokname, models=None, byquery=False*)

Clean all mapping with removed object linked:

```
Mapping.clean('My blok')
```

Warning: For filter only the no blokname:

```
Mapping.clean(None)
```

Params blokname filter by blok

Params models filter by model, keep the order to remove the mapping

filter_by_model_and_key (*model, key*)

SQLAlchemy hybrid method to filter by model and key

Parameters

- **model** – model of the mapping
- **key** – external key of the mapping

Hybrid method

filter_by_model_and_keys (*model, *keys*)

SQLAlchemy hybrid method to filter by model and key

Parameters

- **model** – model of the mapping
- **key** – external key of the mapping

Hybrid method

classmethod get (*model, key*)

return instance of the model with this external key

Parameters

- **model** – model of the mapping
- **key** – string of the key

Return type instance of the model

classmethod `get_mapping_primary_keys` (*model, key*)
return primary key for a model and an external key

Parameters

- **model** – model of the mapping
- **key** – string of the key

Return type dict primary key: value or None

classmethod `multi_delete` (*model, *keys, **kwargs*)
Delete all the keys for this model

Parameters

- **model** – model of the mapping
- **keys** – list of the key

Return type Boolean True if the mappings are removed

classmethod `set` (*key, instance, raiseifexist=True, blokname=None*)
Add or update a mmping with a model and a external key

Parameters

- **model** – model to check
- **key** – string of the key
- **instance** – instance of the model to save
- **raiseifexist** – boolean (True by default), if True and the entry exist then an exception is raised
- **blokname** – name of the blok where come from the mapping

Exception IOMappingSetException

classmethod `set_primary_keys` (*model, key, pks, raiseifexist=True, blokname=None*)
Add or update a mmping with a model and a external key

Parameters

- **model** – model to check
- **key** – string of the key
- **pks** – dict of the primary key to save
- **raiseifexist** – boolean (True by default), if True and the entry exist then an exception is raised
- **blokname** – name of the blok where come from the mapping

Exception IOMappingSetException

mixin

class anyblok_io.bloks.io.mixin.IOMixin

Bases: object

AnyBlok registration:

- Type: Mixin
- Registry name: Mixin.IOMixin

Fields	
id	<ul style="list-style-type: none"> • Type - anyblok.column.Integer • primary_key - True • autoincrement - True • default - anyblok.column.NoDefaultValue
mode	<ul style="list-style-type: none"> • Type - anyblok.column.Selection • nullable - False • default - anyblok.column.NoDefaultValue • selections - 'get_mode_choices' • size - 64
model	<ul style="list-style-type: none"> • Type - anyblok.column.String • nullable - False • foreign_key - Model.System.Model => name • default - anyblok.column.NoDefaultValue • size - 256

importer

class anyblok_io.bloks.io.importer.Importer

Bases: anyblok.mixin.IOMixin

AnyBlok registration:

- Type: Model
- Registry name: Model.IO.Importer
- Tablename: io_importer
- Inherited Models or Mixins:
 - anyblok.mixin.IOMixin

Fields	
file_to_import	<ul style="list-style-type: none">• Type - <code>anyblok.column.LargeBinary</code>• nullable - <code>False</code>• default - <code>anyblok.column.NoDefaultValue</code>
offset	<ul style="list-style-type: none">• Type - <code>anyblok.column.Integer</code>• default - <code>0</code>
nb_grouped_lines	<ul style="list-style-type: none">• Type - <code>anyblok.column.Integer</code>• nullable - <code>False</code>• default - <code>50</code>
commit_at_each_grouped	<ul style="list-style-type: none">• Type - <code>anyblok.column.Boolean</code>• default - <code>True</code>
check_import	<ul style="list-style-type: none">• Type - <code>anyblok.column.Boolean</code>• default - <code>False</code>

exporter

class `anyblok_io.bloks.io.exporter.Exporter`

Bases: `anyblok.mixin.IOMixin`

AnyBlok registration:

- Type: `Model`
- Registry name: `Model.IO.Exporter`
- Tablename: `io_exporter`
- Inherited Models or Mixins:
 - `anyblok.mixin.IOMixin`

formater

class `anyblok_io.bloks.io.formater.Formatter`

Bases: `object`

AnyBlok registration:

- Type: `Model`
- Registry name: `Model.IO.Formatter`
- Tablename: `io_formater`

class `anyblok_io.bloks.io.formater.Float`

Bases: `anyblok.model.Formatter`

AnyBlok registration:

- Type: `Model`

- Registry name: Model.IO.Formatter.Float
- Tablename: io_formatter_float
- Inherited Models or Mixins:
 - anyblok.model.Formatter

class anyblok_io.bloks.io.formatter.Decimal
Bases: anyblok.model.Formatter

AnyBlok registration:

- Type: Model
- Registry name: Model.IO.Formatter.Decimal
- Tablename: io_formatter_decimal
- Inherited Models or Mixins:
 - anyblok.model.Formatter

class anyblok_io.bloks.io.formatter.Json
Bases: anyblok.model.Formatter

AnyBlok registration:

- Type: Model
- Registry name: Model.IO.Formatter.Json
- Tablename: io_formatter_json
- Inherited Models or Mixins:
 - anyblok.model.Formatter

class anyblok_io.bloks.io.formatter.Interval
Bases: anyblok.model.Formatter

AnyBlok registration:

- Type: Model
- Registry name: Model.IO.Formatter.Interval
- Tablename: io_formatter_interval
- Inherited Models or Mixins:
 - anyblok.model.Formatter

class anyblok_io.bloks.io.formatter.Integer
Bases: anyblok.model.Formatter

AnyBlok registration:

- Type: Model
- Registry name: Model.IO.Formatter.Integer
- Tablename: io_formatter_integer
- Inherited Models or Mixins:
 - anyblok.model.Formatter

class anyblok_io.bloks.io.formater.**BigInteger**

Bases: anyblok.model.Integer

AnyBlok registration:

- Type: Model
- Registry name: Model.IO.Formatter.BigInteger
- Tablename: io_formatter_biginteger
- Inherited Models or Mixins:
 - anyblok.model.Integer

class anyblok_io.bloks.io.formater.**Boolean**

Bases: anyblok.model.Formatter

AnyBlok registration:

- Type: Model
- Registry name: Model.IO.Formatter.Boolean
- Tablename: io_formatter_boolean
- Inherited Models or Mixins:
 - anyblok.model.Formatter

class anyblok_io.bloks.io.formater.**Time**

Bases: anyblok.model.Formatter

AnyBlok registration:

- Type: Model
- Registry name: Model.IO.Formatter.Time
- Tablename: io_formatter_time
- Inherited Models or Mixins:
 - anyblok.model.Formatter

class anyblok_io.bloks.io.formater.**Date**

Bases: anyblok.model.Formatter

AnyBlok registration:

- Type: Model
- Registry name: Model.IO.Formatter.Date
- Tablename: io_formatter_date
- Inherited Models or Mixins:
 - anyblok.model.Formatter

class anyblok_io.bloks.io.formater.**DateTime**

Bases: anyblok.model.Formatter

AnyBlok registration:

- Type: Model
- Registry name: Model.IO.Formatter.DateTime
- Tablename: io_formatter_datetime

- Inherited Models or Mixins:
 - `anyblok.model.Formatter`

class `anyblok_io.bloks.io.formater.Many2One`
Bases: `anyblok.model.Formatter`

AnyBlok registration:

- Type: Model
- Registry name: `Model.IO.Formatter.Many2One`
- Tablename: `io_formatter_many2one`
- Inherited Models or Mixins:
 - `anyblok.model.Formatter`

class `anyblok_io.bloks.io.formater.One2One`
Bases: `anyblok.model.Many2One`

AnyBlok registration:

- Type: Model
- Registry name: `Model.IO.Formatter.One2One`
- Tablename: `io_formatter_one2one`
- Inherited Models or Mixins:
 - `anyblok.model.Many2One`

class `anyblok_io.bloks.io.formater.Many2Many`
Bases: `anyblok.model.Formatter`

AnyBlok registration:

- Type: Model
- Registry name: `Model.IO.Formatter.Many2Many`
- Tablename: `io_formatter_many2many`
- Inherited Models or Mixins:
 - `anyblok.model.Formatter`

class `anyblok_io.bloks.io.formater.One2Many`
Bases: `anyblok.model.Many2Many`

AnyBlok registration:

- Type: Model
- Registry name: `Model.IO.Formatter.One2Many`
- Tablename: `io_formatter_one2many`
- Inherited Models or Mixins:
 - `anyblok.model.Many2Many`

3.2 Blok anyblok-io-csv

```
class anyblok_io.bloks.io_csv.AnyBlokIOCSV (registry)
    Bases: anyblok.blok.Blok

    CSV Importer / Exporter behaviour

    name = 'anyblok-io-csv'
    version = '2.0.0'
    author = 'Suzanne Jean-Sébastien'
    autoinstall = False
    priority = 100
```

Note: Require the anyblok-io-csv blok

3.2.1 Exporter

Add an exporter mode (CSV) in AnyBlok:

```
Exporter = registry.IO.Exporter.CSV
```

Define what export:

```
csv_delimiter = ','
csv_quotechar = '"'
model = ``Existing model name``
fields = [
    {'name': 'field name'},
    {'name': 'fieldname1.fieldname2. ... .fieldnamen} # fieldname1, fieldname 2_
    ↳ must be Many2One or have foreign keys
    {'name': 'field', model="external_id"} # field must be Many2One or foreign_keys_
    ↳ or primary keys
    ...
]
```

Create the Exporter:

```
exporter = Exporter.insert(csv_delimiter=csv_delimiter,
                           csv_quotechar=csv_quotechar,
                           model=model,
                           fields=fields)
```

Warning: You can also make insert with registry.IO.Exporter directly

Run the export:

```
fp = exporter.run(entries) # entries are instance of the ``model``
```

3.2.2 Importer

Add an importer mode (CSV) in AnyBlok:

```
Importer = registry.IO.Importer.CSV
```

Define what import:

```
csv_delimiter = ','
csv_quotechar = '"'
model = ``Existing model name``
with open(..., 'rb') as fp:
    file_to_import = fp.read()
```

Create the Importer:

```
importer = Importer.insert(csv_delimiter=csv_delimiter,
                           csv_quotechar=csv_quotechar,
                           model=model,
                           file_to_import=file_to_import)
```

Warning: You can also make insert with registry.IO.Importer directly

Run the import:

```
res = importer.run()
```

The result is a dict with:

- **error_found:** List the error, durring the import
- **created_entries:** Entries created by the import
- **updated_entries:** Entries updated by the import

List of the options for the import:

- **csv_on_error:**
 - **raise_now:** Raise now
 - **raise_at_the_end** (default): Raise at the end
 - **ignore:** Ignore and continue
- **csv_if_exist:**
 - **pass:** Pass to the next record
 - **overwrite** (default): Update the record
 - **create:** Create another record
 - **raise:** Raise an exception
- **csv_if_does_not_exist:**
 - **pass:** Pass to the next record
 - **create** (default): Create another record
 - **raise:** Raise an exception

3.2.3 API doc

exceptions

exception `anyblok_io.bloks.io_csv.exceptions.CSVImporterException`

Bases: `anyblok_io.bloks.io.exceptions.ImporterException`

Simple exception for CSV importer

exception `anyblok_io.bloks.io_csv.exceptions.CSVExporterException`

Bases: `anyblok_io.bloks.io.exceptions.ExporterException`

Simple exception for CSV exporter

mixin

class `anyblok_io.bloks.io_csv.mixin.IOCSVFieldMixin`

Bases: `object`

AnyBlok registration:

- Type: Mixin
- Registry name: `Mixin.IOCSVFieldMixin`

Fields	
id	<ul style="list-style-type: none">• Type - <code>anyblok.column.Integer</code>• <code>primary_key</code> - <code>True</code>• <code>autoincrement</code> - <code>True</code>• default - <code>anyblok.column.NoDefaultValue</code>
name	<ul style="list-style-type: none">• Type - <code>anyblok.column.String</code>• <code>nullable</code> - <code>False</code>• default - <code>anyblok.column.NoDefaultValue</code>• <code>size</code> - <code>64</code>

class `anyblok_io.bloks.io_csv.mixin.IOCSVMixin`

Bases: `object`

AnyBlok registration:

- Type: Mixin
- Registry name: `Mixin.IOCSVMixin`

Fields	
csv_delimiter	<ul style="list-style-type: none"> • Type - <code>anyblok.column.String</code> • nullable - <code>False</code> • default - <code>','</code> • size - <code>64</code>
csv_quotechar	<ul style="list-style-type: none"> • Type - <code>anyblok.column.String</code> • nullable - <code>False</code> • default - <code>'''</code> • size - <code>64</code>

importer

class `anyblok_io.bloks.io_csv.importer.Importer`

Bases: `anyblok.mixin.IOCVMixin`

AnyBlok registration:

- Type: `Model`
- Registry name: `Model.IO.Importer`
- Tablename: `io_importer`
- Inherited Models or Mixins:
 - `anyblok.mixin.IOCVMixin`

Fields	
csv_on_error	<ul style="list-style-type: none"> • Type - <code>anyblok.column.Selection</code> • default - <code>'raise_at_the_end'</code> • size - <code>64</code>
csv_if_exist	<ul style="list-style-type: none"> • Type - <code>anyblok.column.Selection</code> • default - <code>'overwrite'</code> • size - <code>64</code>
csv_if_does_not_exist	<ul style="list-style-type: none"> • Type - <code>anyblok.column.Selection</code> • default - <code>'create'</code> • size - <code>64</code>

class `anyblok_io.bloks.io_csv.importer.CSV (importer, blokname=None)`

Bases: `object`

AnyBlok registration:

- Type: `Model`
- Registry name: `Model.IO.Importer.CSV`
- Tablename: `io_importer_csv`

exporter**class** anyblok_io.bloks.io_csv.exporter.**Exporter**

Bases: anyblok.mixin.IOCSVMixin

AnyBlok registration:

- Type: Model
- Registry name: Model.IO.Exporter
- Tablename: io_exporter
- Inherited Models or Mixins:
 - anyblok.mixin.IOCSVMixin

class anyblok_io.bloks.io_csv.exporter.**Field**

Bases: anyblok.mixin.IOCSVFieldMixin

AnyBlok registration:

- Type: Model
- Registry name: Model.IO.Exporter.Field
- Tablename: io_exporter_field
- Inherited Models or Mixins:
 - anyblok.mixin.IOCSVFieldMixin

Fields	
exporter	<ul style="list-style-type: none">• Type - anyblok.relationship.Many2One• backref - 'fields_to_export'• model - Model.IO.Exporter• index - False
mode	<ul style="list-style-type: none">• Type - anyblok.column.Selection• nullable - False• default - 'any'• selections - 'get_selection'• size - 64
mapping	<ul style="list-style-type: none">• Type - anyblok.column.String• default - anyblok.column.NoDefaultValue• size - 64

class anyblok_io.bloks.io_csv.exporter.**CSV** (*exporter*)

Bases: object

AnyBlok registration:

- Type: Model
- Registry name: Model.IO.Exporter.CSV

- Tablename: io_exporter_csv

3.3 Blok anyblok-io-xml

class anyblok_io.bloks.io_xml.**AnyBlokIOXML** (*registry*)
 Bases: anyblok.blok.Blok
 XML Importer / Exporter behaviour

Warning: Importer and Exporter are not implemented yet

```
name = 'anyblok-io-xml'
version = '2.0.0'
author = 'Suzanne Jean-Sébastien'
autoinstall = False
priority = 100
```

Note: Require the anyblok-io-xml blok

3.3.1 Exporter

TODO

3.3.2 Importer

Add an importer mode (XML) in AnyBlok:

```
Importer = registry.IO.Importer.XML
```

Define what import:

```
model = ``Existing model name``
with open(..., 'rb') as fp:
    file_to_import = fp.read()
```

Create the Importer:

```
importer = Importer.insert(model=model,
                           file_to_import=file_to_import)
```

Warning: You can also make insert with registry.IO.Importer directly

Run the import:

```
res = importer.run()
```

The result is a dict with:

- `error_found`: List the error, durring the import
- `created_entries`: Entries created by the import
- `updated_entries`: Entries updated by the import

Root structure of the XML file:

```
<records on_error="...">
  ...
</records>
```

raise can have the value:

- `raise` (dafault)
- `ignore`

records node can have:

- `commit`: commit the import, only if no error found
- `record`: import one record

Add a record:

```
<records>
  <record>
    ...
    <field name="..." />
    ...
  </record>
</records>
```

Record attribute:

- `model`: if not filled, then the importer will indicate the model
- `external_id`: Mapping key
- `param`: Mapping key only for the import (not save)
- **`on_error`:**
 - `raise`
 - `ignore` (default)
- **`if_exist`:**
 - `overwrite` (default)
 - `create`
 - `pass`: continue to the next record
 - `continue`: continue on the sub record without take this record
 - `raise`
- **`id_does_not_exist`:**
 - `create` (default)
 - `pass`

- raise

The field node represents a Field, a Column or Relationship, the attributes are:

- **name** (required): name of the field, column or relationship

Case of the relationship, they have some more attributes:

- **external_id**:
- **param**:
- **on_error**:
 - raise
 - ignore (default)
- **if_exist**:
 - overwrite (default)
 - create
 - pass: continue to the next record
 - continue: continue on the sub record without take this record
 - raise
- **id_does_not_exist**:
 - create (default)
 - pass
 - raise

Many2One and One2One declaration is directly in the field node:

```
<records
  <record
    ...
    <field name="Many2One or One2One">
      ...
      <field name="..." />
      ...
    </field>
    ...
  </record>
</records>
```

One2Many and Many2Many declarations are also in the field but with a record node:

```
<records
  <record
    ...
    <field name="Many2Many or One2Many">
      ...
      <record>
        ...
        <field name="..." />
        ...
      </record>
    ...
  </record>
  ...
</records>
```

(continues on next page)

(continued from previous page)

```
        </field>
        ...
    </record>
</records>
```

In the case of polymorphisme you may use the attribute model on record:

```
<records
  <record
    ...
    <field name="Many2Many or One2Many">
      ...
      <record model="`polymorphic model`">
        ...
        <field name="..." />
        ...
      </record>
      <record model="`another polymorphic model`">
        ...
        <field name="..." />
        ...
      </record>
      ...
    </field>
    ...
  </record>
</records>
```

3.3.3 API doc

exceptions

exception anyblok_io.bloks.io_xml.exceptions.**XMLImporterException**

Bases: anyblok_io.bloks.io.exceptions.ImporterException

Simple exception for XML importer

exception anyblok_io.bloks.io_xml.exceptions.**XMLExporterException**

Bases: anyblok_io.bloks.io.exceptions.ExporterException

Simple exception for XML exporter

importer

class anyblok_io.bloks.io_xml.importer.**Importer**

Bases: object

AnyBlok registration:

- Type: Model
- Registry name: Model.IO.Importer
- Tablename: io_importer

class anyblok_io.bloks.io_xml.importer.**XML** (*importer, blokname=None*)

Bases: object

AnyBlok registration:

- Type: Model
- Registry name: Model.IO.Importer.XML
- Tablename: io_importer_xml

exporter

class anyblok_io.bloks.io_xml.exporter.**Exporter**
Bases: object

AnyBlok registration:

- Type: Model
- Registry name: Model.IO.Exporter
- Tablename: io_exporter

class anyblok_io.bloks.io_xml.exporter.**XML** (*exporter*)
Bases: object

AnyBlok registration:

- Type: Model
- Registry name: Model.IO.Exporter.XML
- Tablename: io_exporter_xml

4.1 2.0.0 (unreleased)

- Put on github action
- Used pre commit hook
- Increased velocity in mapping

4.2 1.2.0 (2021-08-16)

- Adapted to **AnyBlok** ≥ 1.2

4.3 1.1.1 (2020-10-16)

- Fixed the size of the fields **model**, because they have a foreign key to the model **Model.System.Model** on the field **name**. The next version of AnyBlok check that the size are the same

4.4 1.1.0 (2020-05-06)

- Removed **Python 3.4** capability
- Removed **Python 3.5** capability
- Refactored unittest, replaced nose by pytest
- Fixed #1 added hook to render primary key serializable

4.5 1.0.0 (2018-05-30)

- Cherry pick io bloks from AnyBlok distribution

5.1 1. Definitions

5.1.1 1.1. “Contributor”

Means each individual or legal entity that creates, contributes to the creation of, or owns Covered Software.

5.1.2 1.2. “Contributor Version”

Means the combination of the Contributions of others (if any) used by a Contributor and that particular Contributor’s Contribution.

5.1.3 1.3. “Contribution”

Means Covered Software of a particular Contributor.

5.1.4 1.4. “Covered Software”

Means Source Code Form to which the initial Contributor has attached the notice in Exhibit A, the Executable Form of such Source Code Form, and Modifications of such Source Code Form, in each case including portions thereof.

5.1.5 1.5. “Incompatible With Secondary Licenses”

Means:

- **That the initial Contributor has attached the notice described in Exhibit B** to the Covered Software; or
- **That the Covered Software was made available under the terms of version 1.1** or earlier of the License, but not also under the terms of a Secondary License.

5.1.6 1.6. “Executable Form”

Means any form of the work other than Source Code Form.

5.1.7 1.7. “Larger Work”

Means a work that combines Covered Software with other material, in a separate file or files, that is not Covered Software.

5.1.8 1.8. “License”

Means this document.

5.1.9 1.9. “Licensable”

Means having the right to grant, to the maximum extent possible, whether at the time of the initial grant or subsequently, any and all of the rights conveyed by this License.

5.1.10 1.10. “Modifications”

Means any of the following:

- **Any file in Source Code Form that results from an addition to, deletion from,** or modification of the contents of Covered Software; or
- Any new file in Source Code Form that contains any Covered Software.

5.1.11 1.11. “Patent Claims” of a Contributor

Means any patent claim(s), including without limitation, method, process, and apparatus claims, in any patent Licensable by such Contributor that would be infringed, but for the grant of the License, by the making, using, selling, offering for sale, having made, import, or transfer of either its Contributions or its Contributor Version.

5.1.12 1.12. “Secondary License”

Means either the GNU General Public License, Version 2.0, the GNU Lesser General Public License, Version 2.1, the GNU Affero General Public License, Version 3.0, or any later versions of those licenses.

5.1.13 1.13. “Source Code Form”

Means the form of the work preferred for making modifications.

5.1.14 1.14. “You” (or “Your”)

Means an individual or a legal entity exercising rights under this License. For legal entities, “You” includes any entity that controls, is controlled by, or is under common control with You. For purposes of this definition, “control” means (a) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (b) ownership of more than fifty percent (50%) of the outstanding shares or beneficial ownership of such entity.

5.2 2. License Grants and Conditions

5.2.1 2.1. Grants

Each Contributor hereby grants You a world-wide, royalty-free, non-exclusive license:

- **Under intellectual property rights (other than patent or trademark)** Licensable by such Contributor to use, reproduce, make available, modify, display, perform, distribute, and otherwise exploit its Contributions, either on an unmodified basis, with Modifications, or as part of a Larger Work; and
- **Under Patent Claims of such Contributor to make, use, sell, offer for sale,** have made, import, and otherwise transfer either its Contributions or its Contributor Version.

5.2.2 2.2. Effective Date

The licenses granted in Section 2.1 with respect to any Contribution become effective for each Contribution on the date the Contributor first distributes such Contribution.

5.2.3 2.3. Limitations on Grant Scope

The licenses granted in this Section 2 are the only rights granted under this License. No additional rights or licenses will be implied from the distribution or licensing of Covered Software under this License. Notwithstanding Section 2.1(b) above, no patent license is granted by a Contributor:

- For any code that a Contributor has removed from Covered Software; or
- **For infringements caused by: (i) Your and any other third party's** modifications of Covered Software, or (ii) the combination of its Contributions with other software (except as part of its Contributor Version); or
- **Under Patent Claims infringed by Covered Software in the absence of its** Contributions.

This License does not grant any rights in the trademarks, service marks, or logos of any Contributor (except as may be necessary to comply with the notice requirements in Section 3.4).

5.2.4 2.4. Subsequent Licenses

No Contributor makes additional grants as a result of Your choice to distribute the Covered Software under a subsequent version of this License (see Section 10.2) or under the terms of a Secondary License (if permitted under the terms of Section 3.3).

5.2.5 2.5. Representation

Each Contributor represents that the Contributor believes its Contributions are its original creation(s) or it has sufficient rights to grant the rights to its Contributions conveyed by this License.

5.2.6 2.6. Fair Use

This License is not intended to limit any rights You have under applicable copyright doctrines of fair use, fair dealing, or other equivalents.

5.2.7 2.7. Conditions

Sections 3.1, 3.2, 3.3, and 3.4 are conditions of the licenses granted in Section 2.1.

5.3 3. Responsibilities

5.3.1 3.1. Distribution of Source Form

All distribution of Covered Software in Source Code Form, including any Modifications that You create or to which You contribute, must be under the terms of this License. You must inform recipients that the Source Code Form of the Covered Software is governed by the terms of this License, and how they can obtain a copy of this License. You may not attempt to alter or restrict the recipients' rights in the Source Code Form.

5.3.2 3.2. Distribution of Executable Form

If You distribute Covered Software in Executable Form then:

- **Such Covered Software must also be made available in Source Code Form, as** described in Section 3.1, and You must inform recipients of the Executable Form how they can obtain a copy of such Source Code Form by reasonable means in a timely manner, at a charge no more than the cost of distribution to the recipient; and
- **You may distribute such Executable Form under the terms of this License, or** sublicense it under different terms, provided that the license for the Executable Form does not attempt to limit or alter the recipients' rights in the Source Code Form under this License.

5.3.3 3.3. Distribution of a Larger Work

You may create and distribute a Larger Work under terms of Your choice, provided that You also comply with the requirements of this License for the Covered Software. If the Larger Work is a combination of Covered Software with a work governed by one or more Secondary Licenses, and the Covered Software is not Incompatible With Secondary Licenses, this License permits You to additionally distribute such Covered Software under the terms of such Secondary License(s), so that the recipient of the Larger Work may, at their option, further distribute the Covered Software under the terms of either this License or such Secondary License(s).

5.3.4 3.4. Notices

You may not remove or alter the substance of any license notices (including copyright notices, patent notices, disclaimers of warranty, or limitations of liability) contained within the Source Code Form of the Covered Software, except that You may alter any license notices to the extent required to remedy known factual inaccuracies.

5.3.5 3.5. Application of Additional Terms

You may choose to offer, and to charge a fee for, warranty, support, indemnity or liability obligations to one or more recipients of Covered Software. However, You may do so only on Your own behalf, and not on behalf of any Contributor. You must make it absolutely clear that any such warranty, support, indemnity, or liability obligation is offered by You alone, and You hereby agree to indemnify every Contributor for any liability incurred by such Contributor as a result of warranty, support, indemnity or liability terms You offer. You may include additional disclaimers of warranty and limitations of liability specific to any jurisdiction.

5.4 4. Inability to Comply Due to Statute or Regulation

If it is impossible for You to comply with any of the terms of this License with respect to some or all of the Covered Software due to statute, judicial order, or regulation then You must: (a) comply with the terms of this License to the maximum extent possible; and (b) describe the limitations and the code they affect. Such description must be placed in a text file included with all distributions of the Covered Software under this License. Except to the extent prohibited by statute or regulation, such description must be sufficiently detailed for a recipient of ordinary skill to be able to understand it.

5.5 5. Termination

5.5.1 5.1.

The rights granted under this License will terminate automatically if You fail to comply with any of its terms. However, if You become compliant, then the rights granted under this License from a particular Contributor are reinstated (a) provisionally, unless and until such Contributor explicitly and finally terminates Your grants, and (b) on an ongoing basis, if such Contributor fails to notify You of the non-compliance by some reasonable means prior to 60 days after You have come back into compliance. Moreover, Your grants from a particular Contributor are reinstated on an ongoing basis if such Contributor notifies You of the non-compliance by some reasonable means, this is the first time You have received notice of non-compliance with this License from such Contributor, and You become compliant prior to 30 days after Your receipt of the notice.

5.5.2 5.2.

If You initiate litigation against any entity by asserting a patent infringement claim (excluding declaratory judgment actions, counter-claims, and cross-claims) alleging that a Contributor Version directly or indirectly infringes any patent, then the rights granted to You by any and all Contributors for the Covered Software under Section 2.1 of this License shall terminate.

5.5.3 5.3.

In the event of termination under Sections 5.1 or 5.2 above, all end user license agreements (excluding distributors and resellers) which have been validly granted by You or Your distributors under this License prior to termination shall survive termination.

5.6 6. Disclaimer of Warranty

Warning: Covered Software is provided under this License on an “as is” basis, without warranty of any kind, either expressed, implied, or statutory, including, without limitation, warranties that the Covered Software is free of defects, merchantable, fit for a particular purpose or non-infringing. The entire risk as to the quality and performance of the Covered Software is with You. Should any Covered Software prove defective in any respect, You (not any Contributor) assume the cost of any necessary servicing, repair, or correction. This disclaimer of warranty constitutes an essential part of this License. No use of any Covered Software is authorized under this License except under this disclaimer.

5.7 7. Limitation of Liability

Warning: Under no circumstances and under no legal theory, whether tort (including negligence), contract, or otherwise, shall any Contributor, or anyone who distributes Covered Software as permitted above, be liable to You for any direct, indirect, special, incidental, or consequential damages of any character including, without limitation, damages for lost profits, loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses, even if such party shall have been informed of the possibility of such damages. This limitation of liability shall not apply to liability for death or personal injury resulting from such party's negligence to the extent applicable law prohibits such limitation. Some jurisdictions do not allow the exclusion or limitation of incidental or consequential damages, so this exclusion and limitation may not apply to You.

5.8 8. Litigation

Any litigation relating to this License may be brought only in the courts of a jurisdiction where the defendant maintains its principal place of business and such litigation shall be governed by laws of that jurisdiction, without reference to its conflict-of-law provisions. Nothing in this Section shall prevent a party's ability to bring cross-claims or counter-claims.

5.9 9. Miscellaneous

This License represents the complete agreement concerning the subject matter hereof. If any provision of this License is held to be unenforceable, such provision shall be reformed only to the extent necessary to make it enforceable. Any law or regulation which provides that the language of a contract shall be construed against the drafter shall not be used to construe this License against a Contributor.

5.10 10. Versions of the License

5.10.1 10.1. New Versions

Mozilla Foundation is the license steward. Except as provided in Section 10.3, no one other than the license steward has the right to modify or publish new versions of this License. Each version will be given a distinguishing version number.

5.10.2 10.2. Effect of New Versions

You may distribute the Covered Software under the terms of the version of the License under which You originally received the Covered Software, or under the terms of any subsequent version published by the license steward.

5.10.3 10.3. Modified Versions

If you create software not governed by this License, and you want to create a new license for such software, you may create and use a modified version of this License if you rename the license and remove any references to the name of the license steward (except to note that such modified license differs from this License).

5.10.4 10.4. Distributing Source Code Form that is Incompatible With Secondary Licenses

If You choose to distribute Source Code Form that is Incompatible With Secondary Licenses under the terms of this version of the License, the notice described in Exhibit B of this License must be attached.

5.11 Exhibit A - Source Code Form License Notice

This Source Code Form **is** subject to the terms of the Mozilla Public License, v. 2.0. If a copy of the MPL was **not** distributed **with** this file, You can obtain one at <http://mozilla.org/MPL/2.0/>.

If it is not possible or desirable to put the notice in a particular file, then You may include the notice in a location (such as a LICENSE file in a relevant directory) where a recipient would be likely to look for such a notice.

Note: You may add additional accurate notices of copyright ownership.

5.12 Exhibit B - “Incompatible With Secondary Licenses” Notice

This Source Code Form is “Incompatible With Secondary Licenses”, as defined by the Mozilla Public License, v. 2.0.

CHAPTER 6

Indices and tables

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

a

- `anyblok_io.bloks.io`, [7](#)
- `anyblok_io.bloks.io.core`, [9](#)
- `anyblok_io.bloks.io.exceptions`, [9](#)
- `anyblok_io.bloks.io.exporter`, [14](#)
- `anyblok_io.bloks.io.formater`, [14](#)
- `anyblok_io.bloks.io.importer`, [13](#)
- `anyblok_io.bloks.io.mapping`, [9](#)
- `anyblok_io.bloks.io.mixin`, [12](#)
- `anyblok_io.bloks.io_csv`, [18](#)
- `anyblok_io.bloks.io_csv.exceptions`, [20](#)
- `anyblok_io.bloks.io_csv.exporter`, [22](#)
- `anyblok_io.bloks.io_csv.importer`, [21](#)
- `anyblok_io.bloks.io_csv.mixin`, [20](#)
- `anyblok_io.bloks.io_xml`, [23](#)
- `anyblok_io.bloks.io_xml.exceptions`, [26](#)
- `anyblok_io.bloks.io_xml.exporter`, [27](#)
- `anyblok_io.bloks.io_xml.importer`, [26](#)

A

`anyblok_io.bloks.io` (module), 7
`anyblok_io.bloks.io.core` (module), 9
`anyblok_io.bloks.io.exceptions` (module), 9
`anyblok_io.bloks.io.exporter` (module), 14
`anyblok_io.bloks.io.formater` (module), 14
`anyblok_io.bloks.io.importer` (module), 13
`anyblok_io.bloks.io.mapping` (module), 9
`anyblok_io.bloks.io.mixin` (module), 12
`anyblok_io.bloks.io_csv` (module), 18
`anyblok_io.bloks.io_csv.exceptions` (module), 20
`anyblok_io.bloks.io_csv.exporter` (module), 22
`anyblok_io.bloks.io_csv.importer` (module), 21
`anyblok_io.bloks.io_csv.mixin` (module), 20
`anyblok_io.bloks.io_xml` (module), 23
`anyblok_io.bloks.io_xml.exceptions` (module), 26
`anyblok_io.bloks.io_xml.exporter` (module), 27
`anyblok_io.bloks.io_xml.importer` (module), 26
`AnyBlokIO` (class in `anyblok_io.bloks.io`), 7
`AnyBlokIOCSV` (class in `anyblok_io.bloks.io_csv`), 18
`AnyBlokIOXML` (class in `anyblok_io.bloks.io_xml`), 23
`author` (`anyblok_io.bloks.io.AnyBlokIO` attribute), 7
`author` (`anyblok_io.bloks.io_csv.AnyBlokIOCSV` attribute), 18
`author` (`anyblok_io.bloks.io_xml.AnyBlokIOXML` attribute), 23
`autoinstall` (`anyblok_io.bloks.io.AnyBlokIO` attribute), 7
`autoinstall` (`anyblok_io.bloks.io_csv.AnyBlokIOCSV` attribute), 18
`autoinstall` (`anyblok_io.bloks.io_xml.AnyBlokIOXML` attribute), 23

N

`name` (`anyblok_io.bloks.io.AnyBlokIO` attribute), 7
`name` (`anyblok_io.bloks.io_csv.AnyBlokIOCSV` attribute), 18
`name` (`anyblok_io.bloks.io_xml.AnyBlokIOXML` attribute), 23

P

`priority` (`anyblok_io.bloks.io.AnyBlokIO` attribute), 7
`priority` (`anyblok_io.bloks.io_csv.AnyBlokIOCSV` attribute), 18
`priority` (`anyblok_io.bloks.io_xml.AnyBlokIOXML` attribute), 23

V

`version` (`anyblok_io.bloks.io.AnyBlokIO` attribute), 7
`version` (`anyblok_io.bloks.io_csv.AnyBlokIOCSV` attribute), 18
`version` (`anyblok_io.bloks.io_xml.AnyBlokIOXML` attribute), 23