AnyBlok / Bus Documentation

Release 1.1.0

jssuzanne

Jun 24, 2019

Contents:

1 Front Matter **1**
- 1.1 Project Homepage . . . 1
- 1.2 Installation . . . 1
- 1.3 Unit Test . . . 1
- 1.4 Script . . . 2
- 1.5 Dependencies . . . 2
- 1.6 Author . . . 2
- 1.7 Contributors . . . 2
- 1.8 Bugs . . . 2

2 Usage **3**
- 2.1 Declare a new consumer . . . 3
- 2.2 Publish a message through rabbitmq . . . 3

3 Code **5**
- 3.1 Declare a consumer on queue with a marshmallow schema . . . 5
- 3.2 Worker . . . 6

4 Bloks **11**
- 4.1 Blok **bus** . . . 11

5 CHANGELOG **17**
- 5.1 1.2.0 . . . 17
- 5.2 1.1.0 (2018-09-15) . . . 17
- 5.3 1.0.0 (2018-06-05) . . . 17

6 Mozilla Public License Version 2.0 **19**
- 6.1 1. Definitions . . . 19
- 6.2 2. License Grants and Conditions . . . 21
- 6.3 3. Responsibilities . . . 22
- 6.4 4. Inability to Comply Due to Statute or Regulation . . . 23
- 6.5 5. Termination . . . 23
- 6.6 6. Disclaimer of Warranty . . . 23
- 6.7 7. Limitation of Liability . . . 24
- 6.8 8. Litigation . . . 24
- 6.9 9. Miscellaneous . . . 24
- 6.10 10. Versions of the License . . . 24

6.11 Exhibit A - Source Code Form License Notice 25
6.12 Exhibit B - “Incompatible With Secondary Licenses” Notice 25

7 Indices and tables **27**

Python Module Index **29**

Index **31**

CHAPTER 1

Front Matter

Information about the AnyBlok / Bus project.

1.1 Project Homepage

AnyBlok is hosted on github - the main project page is at https://github.com/AnyBlok/anyblok_bus. Source code is tracked here using GIT.

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

The most recent published version of this documentation should be at http://doc.anyblok-bus.anyblok.org.

1.2 Installation

Install released versions of AnyBlok from the Python package index with pip or a similar tool:

```
pip install anyblok_bus
```

Installation via source distribution is via the `setup.py` script:

```
python setup.py install
```

Installation will add the `anyblok` commands to the environment.

1.3 Unit Test

Run the test with `nose`:

```
pip install nose
nosetests anyblok_bus/tests
```

1.4 Script

anyblok_bus add `console_script` to launch worker. A worker consume a queue defined by the decorator **anyblok_bus.bus_consumer**:

```
anyblok_bus -c anyblok_config_file.cfg
```

..note:: The profile name in the configuration is used to find the correct url to connect to rabbitmq

1.5 Dependencies

AnyBlok / Bus works with **Python 3.3** and later and pika. The install process will ensure that AnyBlok is installed, in addition to other dependencies. The latest version of them is strongly recommended.

1.6 Author

Jean-Sébastien Suzanne

1.7 Contributors

Anybox team:

- Jean-Sébastien Suzanne
- Florent Jouatte

Sensee team:

- Julien SZKUDLAPSKI

1.8 Bugs

Bugs and feature enhancements to AnyBlok should be reported on the Issue tracker.

Contents

- *Usage*
  - *Declare a new consumer*
  - *Publish a message through rabbitmq*

CHAPTER 2

Usage

2.1 Declare a new consumer

In an AnyBlok Model you have to decorate a method with **bus_consumer**

```
from  anyblok_bus import bus_consumer
from anyblok import Declarations
from .schema import MySchema

@Declarations.register(Declarations.Model)
class MyModel:

    @bus_consumer(queue_name='name of the queue', schema=MySchema())
    def my_consumer(cls, body):
        # do something
```

The schema must be an instance of marshmallow.Schema, see the marshmallow documentation

Note: The decorated method become a classmethod with always the same prototype (cls, body) body is the desarialization of the message from the queue by the schema.

2.2 Publish a message through rabbitmq

The publication is done by **registry.Bus.publish** method:

```
registry.Bus.publish('exchange', 'routing_key', message, mimestype)
```

if the message have not be send, then an exception is raised

..warning:

```
A profile must be defined and selected by the AnyBlok configuration **bus_profile**
```

Contents

- *Code*
  - *Declare a consumer on queue with a marshmallow schema*
    - *decorator* `bus_consumer`
    - *anyblok model plugin*
  - *Worker*
    - *Exceptions*

CHAPTER 3

Code

3.1 Declare a consumer on queue with a marshmallow schema

3.1.1 decorator `bus_consumer`

anyblok_bus.consumer.**bus_consumer**(*queue_name=None*, *adapter=None*, *processes=0*, ***kwargs*)

3.1.2 anyblok model plugin

class anyblok_bus.consumer.**BusConsumerPlugin**(*registry*)

Bases: anyblok.model.plugins.ModelPluginBase

anyblok.model.plugin to allow the build of the anyblok_bus.bus_consumer

apply_consumer(*consumer*, *new_base*, *properties*, *transformation_properties*)

Insert in a base the overload

Parameters

- **new_base** – the base to be put on front of all bases
- **properties** – the properties declared in the model
- **transformation_properties** – the properties of the model

initialisation_tranformation_properties(*properties*, *transformation_properties*)

Initialise the transform properties

Parameters

- **properties** – the properties declared in the model
- **new_type_properties** – param to add in a new base if need

insert_in_bases(*new_base*, *namespace*, *properties*, *transformation_properties*)

Insert in a base the overload

Parameters

- **new_base** – the base to be put on front of all bases
- **namespace** – the namespace of the model
- **properties** – the properties declared in the model
- **transformation_properties** – the properties of the model

transform_base_attribute(*attr*, *method*, *namespace*, *base*, *transformation_properties*, *new_type_properties*)

transform the attribute for the final Model

Parameters

- **attr** – attribute name
- **method** – method pointer of the attribute
- **namespace** – the namespace of the model
- **base** – One of the base of the model
- **transformation_properties** – the properties of the model
- **new_type_properties** – param to add in a new base if need

3.2 Worker

class anyblok_bus.worker.**Worker**(*registry*, *profile*, *consumers*, *withautocommit=True*)

Bases: object

Define consumers to consume the queue défined in the AnyBlok registry by the bus_consumer decorator

```
worker = Worker(anyblokregistry, profilename)
worker.start()  # blocking loop
worker.is_ready()  # return True if all the consumer are started
worker.stop()  # stop the loop and close the connection with rabbitmq
```

This is an example consumer that will handle unexpected interactions with RabbitMQ such as channel and connection closures.

If RabbitMQ closes the connection, this class will stop and indicate that reconnection is necessary. You should look at the output, as there are limited reasons why the connection may be closed, which usually are tied to permission related issues or socket timeouts.

If the channel is closed, it will indicate a problem with one of the commands that were issued and that should surface in the output as well.

Parameters

- **registry** – anyblok registry instance
- **profile** – the name of the profile which give the url of rabbitmq
- **consumers** – list of the consumer to consum
- **withautocommit** – default True, commit all the transaction

add_on_cancel_callback()

Add a callback that will be invoked if RabbitMQ cancels the consumer for some reason. If RabbitMQ does cancel the consumer, on_consumer_cancelled will be invoked by pika.

add_on_channel_close_callback()
This method tells pika to call the on_channel_closed method if RabbitMQ unexpectedly closes the channel.

close_channel()
Call to close the channel with RabbitMQ cleanly by issuing the Channel.Close RPC command.

close_connection()

connect()
This method connects to RabbitMQ, returning the connection handle. When the connection is established, the on_connection_open method will be invoked by pika.

Return type pika.SelectConnection

declare_consumer(*queue*, *model*, *method*)

get_url()
Retrieve connection url

is_ready()

on_basic_qos_ok(*_unused_frame*)
Invoked by pika when the Basic.QoS method has completed. At this point we will start consuming messages by calling start_consuming which will invoke the needed RPC commands to start the process.

Parameters **_unused_frame** (*pika.frame.Method*) – The Basic.QosOk response frame

on_bindok(*_unused_frame*, *userdata*)
Invoked by pika when the Queue.Bind method has completed. At this point we will set the prefetch count for the channel.

Parameters

- **_unused_frame** (*pika.frame.Method*) – The Queue.BindOk response frame
- **userdata** (*str|unicode*) – Extra user data (queue name)

on_cancelok(*_unused_frame*, *userdata*)
This method is invoked by pika when RabbitMQ acknowledges the cancellation of a consumer. At this point we will close the channel. This will invoke the on_channel_closed method once the channel has been closed, which will in-turn close the connection.

Parameters

- **_unused_frame** (*pika.frame.Method*) – The Basic.CancelOk frame
- **userdata** (*str|unicode*) – Extra user data (consumer tag)

on_channel_closed(*channel*, *reason*)
Invoked by pika when RabbitMQ unexpectedly closes the channel. Channels are usually closed if you attempt to do something that violates the protocol, such as re-declare an exchange or queue with different parameters. In this case, we'll close the connection to shutdown the object.

Parameters

- **pika.channel.Channel** – The closed channel
- **reason** (*Exception*) – why the channel was closed

on_channel_open(*channel*)
This method is invoked by pika when the channel has been opened. The channel object is passed in so we can make use of it.

Since the channel is now open, we'll declare the exchange to use.

Parameters **channel** (*pika.channel.Channel*) – The channel object

on_connection_closed(*_unused_connection*, *reason*)
This method is invoked by pika when the connection to RabbitMQ is closed unexpectedly. Since it is unexpected, we will reconnect to RabbitMQ if it disconnects.

Parameters

- **connection** (*pika.connection.Connection*) – The closed connection obj
- **reason** (*Exception*) – exception representing reason for loss of connection.

on_connection_open(*_unused_connection*)
This method is called by pika once the connection to RabbitMQ has been established. It passes the handle to the connection object in case we need it, but in this case, we'll just mark it unused.

Parameters **_unused_connection** (*pika.SelectConnection*) – The connection

on_connection_open_error(*_unused_connection*, *err*)
This method is called by pika if the connection to RabbitMQ can't be established.

Parameters

- **_unused_connection** (*pika.SelectConnection*) – The connection
- **err** (*Exception*) – The error

on_consumer_cancelled(*method_frame*)
Invoked by pika when RabbitMQ sends a Basic.Cancel for a consumer receiving messages.

Parameters **method_frame** (*pika.frame.Method*) – The Basic.Cancel frame

open_channel()
Open a new channel with RabbitMQ by issuing the Channel.Open RPC command. When RabbitMQ responds that the channel is open, the on_channel_open callback will be invoked by pika.

reconnect()
Will be invoked if the connection can't be opened or is closed. Indicates that a reconnect is necessary then stops the ioloop.

set_qos()
This method sets up the consumer prefetch to only be delivered one message at a time. The consumer must acknowledge this message before RabbitMQ will deliver another one. You should experiment with different prefetch values to achieve desired performance.

start()
Run the example consumer by connecting to RabbitMQ and then starting the IOLoop to block and allow the SelectConnection to operate.

start_consuming()
This method sets up the consumer by first calling add_on_cancel_callback so that the object is notified if RabbitMQ cancels the consumer. It then issues the Basic.Consume RPC command which returns the consumer tag that is used to uniquely identify the consumer with RabbitMQ. We keep the value to use it when we want to cancel consuming. The on_message method is passed in as a callback pika will invoke when a message is fully received.

stop()
Cleanly shutdown the connection to RabbitMQ by stopping the consumer with RabbitMQ. When RabbitMQ confirms the cancellation, on_cancelok will be invoked by pika, which will then closing the channel and connection. The IOLoop is started again because this method is invoked when CTRL-C is pressed raising a KeyboardInterrupt exception. This exception stops the IOLoop which needs to be running for pika to communicate with RabbitMQ. All of the commands issued prior to starting the IOLoop will be buffered but not processed.

stop_consuming()
Tell RabbitMQ that you would like to stop consuming by sending the Basic.Cancel RPC command.

3.2.1 Exceptions

Contents

- *Bloks*
  - *Blok **bus***
    - *Memento*
    - *API doc*
      - ***Bus***
      - ***Profile***
      - ***Message***
      - ***Exceptions***

CHAPTER 4

Bloks

4.1 Blok bus

class anyblok_bus.bloks.bus.**Bus**(*registry*)

Bases: anyblok.blok.Blok

Add bus configuration in AnyBlok

author = 'Suzanne Jean-Sébastien'

conditional_by = []

conflicting_by = []

classmethod import_declaration_module()

Do the python import for the Declaration of the model or other

name = 'bus'

optional_by = []

classmethod reload_declaration_module(*reload*)

required = ['anyblok-core']

required_by = []

version = '1.1.0'

4.1.1 Memento

This blok define two Models:

- **Model.Bus.Profile**: list the connection available to a rabbitmq server
- **Model.Bus.Message**: Give the received message witch did not be imported correctly by the consumer

4.1.2 API doc

Bus

class anyblok_bus.bloks.bus.bus.**Bus**
Bases: object

Namespace Bus

AnyBlok registration:

- Type: Model
- Registry name: Model.Bus
- Tablename: bus

classmethod get_consumers()
Return the list of the consumers

classmethod publish(*exchange*, *routing_key*, *data*, *contenttype*)
Publish a message in an exchange with a routing key through rabbitmq with the profile given by the anyblok configuration

Parameters

- **exchange** – name of the exchange
- **routing_key** – name of the routing key
- **data** – str or unitcode to send through rabbitmq
- **contenttype** – the mimestype of the data

Exception PublishException

Profile

class anyblok_bus.bloks.bus.profile.**Profile**
Bases: object

AnyBlok registration:

- Type: Model
- Registry name: Model.Bus.Profile
- Tablename: bus_profile

Fields	
name	• Type - anyblok.column.String • primary_key - True • unique - True • nullable - False • default - anyblok.column.NoDefaultValue • size - 64
description	• Type - anyblok.column.String • default - anyblok.column.NoDefaultValue • size - 64
url	• Type - anyblok.column.URL • nullable - False • default - anyblok.column.NoDefaultValue
state	• Type - anyblok.column.Selection • nullable - False • default - 'disconnected' • size - 64

Message

class anyblok_bus.bloks.bus.message.**Message**

Bases: object

AnyBlok registration:

- Type: Model
- Registry name: Model.Bus.Message
- Tablename: bus_message

Fields	
id	• Type - anyblok.column.Integer • primary_key - True • autoincrement - True • default - anyblok.column.NoDefaultValue
create_date	• Type - anyblok.column.DateTime • nullable - False • is auto updated - False • default timezone - <UTC>
edit_date	• Type - anyblok.column.DateTime • nullable - False • is auto updated - True • default timezone - <UTC>
content_type	• Type - anyblok.column.String • nullable - False • default - 'application/json' • size - 64
message	• Type - anyblok.column.LargeBinary • nullable - False • default - anyblok.column.NoDefaultValue
sequence	• Type - anyblok.column.Integer • nullable - False • default - 100
error	• Type - anyblok.column.Text • default - anyblok.column.NoDefaultValue
queue	• Type - anyblok.column.String • nullable - False • default - anyblok.column.NoDefaultValue • size - 64
model	• Type - anyblok.column.String • nullable - False • default - anyblok.column.NoDefaultValue • size - 64
method	• Type - anyblok.column.String • nullable - False

consume()
Try to consume on message to import it in database

classmethod consume_all()
Try to consume all the message, ordered by the sequence

Exceptions

exception anyblok_bus.bloks.bus.exceptions.**PublishException**
Bases: Exception

Exception Error for Publish a message through rabbitmq

Contents

- *CHANGELOG*
  - *1.2.0*
  - *1.1.0 (2018-09-15)*
  - *1.0.0 (2018-06-05)*

CHAPTER 5

CHANGELOG

5.1 1.2.0

- Fixed Multiple consumer on the same model
- Refactored bus console script, Added processes parameter on bus_consumer. The goal is to define processes for one queue, by default all the queues are in the same process
- Add better logging when a queue is missing. If a queue is missing, then workers won't start.
- Added adapter parameter to transform bus message, the schema attribute become now a simple kwargs argument give to adapter.

  The adapter is not required.

  Note: To keep the compatibility, if no adapter is defined with a schema then the adapter is schema_adapter

5.2 1.1.0 (2018-09-15)

- Improved logging: for helping to debug the messages
- Added create and update date columns
- fixed `consume_all` method. now the method does not stop when an exception is raised
- Used marsmallow version >= 3.0.0

5.3 1.0.0 (2018-06-05)

- add Worker to consume the message from rabbitmq

- add publish method to publish a message to rabbitmq
- add **anyblok_bus.bus_consumer** add decorator to défine the consumer

CHAPTER 6

Mozilla Public License Version 2.0

6.1 1. Definitions

6.1.1 1.1. “Contributor”

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

6.1.2 1.2. “Contributor Version”

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

6.1.3 1.3. “Contribution”

Means Covered Software of a particular Contributor.

6.1.4 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.

6.1.5 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.

6.1.6 1.6. “Executable Form”

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

6.1.7 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.

6.1.8 1.8. “License”

Means this document.

6.1.9 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.

6.1.10 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.

6.1.11 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.

6.1.12 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.

6.1.13 1.13. “Source Code Form”

Means the form of the work preferred for making modifications.

6.1.14 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.

6.2 2. License Grants and Conditions

6.2.1 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.

6.2.2 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.

6.2.3 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).

6.2.4 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).

6.2.5 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.

6.2.6 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.

6.2.7 2.7. Conditions

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

6.3 3. Responsibilities

6.3.1 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.

6.3.2 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.

6.3.3 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).

6.3.4 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.

6.3.5 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.

6.4 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.

6.5 5. Termination

6.5.1 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.

6.5.2 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.

6.5.3 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.

6.6 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.

6.7 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.

6.8 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.

6.9 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.

6.10 10. Versions of the License

6.10.1 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.

6.10.2 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.

6.10.3 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).

6.10.4 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.

6.11 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.

6.12 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 7

Indices and tables

- genindex
- modindex
- search

Python Module Index

a

anyblok_bus.bloks.bus, 11
anyblok_bus.bloks.bus.bus, 12
anyblok_bus.bloks.bus.exceptions, 15
anyblok_bus.bloks.bus.message, 13
anyblok_bus.bloks.bus.profile, 12
anyblok_bus.consumer, 5
anyblok_bus.worker, 6

Index

A

anyblok_bus.bloks.bus (*module*), 11
anyblok_bus.bloks.bus.bus (*module*), 12
anyblok_bus.bloks.bus.exceptions (*module*), 15
anyblok_bus.bloks.bus.message (*module*), 13
anyblok_bus.bloks.bus.profile (*module*), 12
anyblok_bus.consumer (*module*), 5
anyblok_bus.worker (*module*), 6