

---

# **pybeeryaml Documentation**

***Release 1.0***

**TROUVERIE Joachim**

**Apr 23, 2018**



---

## Contents

---

<b>1</b>	<b>Quick start</b>	<b>3</b>
<b>2</b>	<b>Contributing</b>	<b>5</b>
<b>3</b>	<b>API</b>	<b>7</b>



A Python YAML format parser for beer storage.

Parse a recipe from a [beeryaml](#) file and returns an object containing ingredients and metadata. Supports export to [beerxml](#) format.

For more information about the beeryaml format please read the following documentation:

<http://beeryaml.rtfid.io/>



## 1.1 Installation

```
pip install pybeeryaml
```

## 1.2 Usage

```
from pybeeryaml import Recipe

path_to_beeryaml_file = "/tmp/my_recipe.yml"

# create recipe from file
recipe = Recipe.from_file(path_to_beeryaml_file)

# or from string
with open(path_to_beeryaml_file, "r") as mybeer:
    recipe2 = Recipe.from_yaml(mybeer.read())

assert recipe == recipe2 # True

# convert to beerxml format
recipexml = recipe.to_xml()

# create your recipe
recipe = Recipe(
    name="Test", brewer="TROUVERIE Joachim", type="Pale Ale",
    batch_size=10.0, boil_time=60.0, boil_size=15.0, style="Test"
)
```

```
# convert it to yaml
yaml_data = recipe.to_yaml()
```

## 1.3 Testing

Unit tests can be run with `pytest`.

```
python -m pytest tests
```



## CHAPTER 2

---

### Contributing

---

All PRs and issues on this project are very welcomed.

If you want to use it, want to talk about the modification about this format or want to discuss the utility of this format, feel free to open an issue at <http://github.com/j0ack/pybeeryaml/>.

All the code contribution must follow the [PEP8](#) guide.



## CHAPTER 3

---

API

---

### 3.1 Module contents

### 3.2 Meta

### 3.3 pybeeryaml.meta module