

---

# **fantastico Documentation**

***Release 0.0.1-b31***

**Radu Viorel Cosnita**

April 15, 2013



## CONTENTS

<b>1</b>	<b>Getting started</b>	<b>1</b>
1.1	Why another python framework? . . . . .	1
1.2	Installation manual . . . . .	1
<b>2</b>	<b>Build status</b>	<b>3</b>



## GETTING STARTED

### 1.1 Why another python framework?

The main reason for developing a new framework is simple: I want to use it for teaching purposes. I have seen many projects which fail either because of poor coding or because they become legacy very fast. I will not get into details why and what could have been done. It defeats the purpose.

Each piece of code that is being added to fantastico will follow these simple rules:

1. *The code is written because is needed and there is no clean way to achieve the requirement with existing fantastico features.*
2. The code is developed using TDD (Test Driven Development).
3. The code quality is 9+ (reported by pylint).
4. The code coverage is 90%+ (reported by nose coverage).
5. The code is fully documented and included into documentation.

#### 1.1.1 What do you want to teach who?

I am a big fan of Agile practices and currently I own a domain called [scrum-expert.ro](http://scrum-expert.ro). This is meant to become a collection of hands on resource of how to develop good software with high quality and in a reasonable amount of time. Resources will cover topics like

1. Incremental development always ready for rollout.
2. TDD (Test Driven Development)
3. XP (eXtreme programming)
4. Scrum
5. Projects setup for Continuous Delivery

and many other topics that are required for delivering high quality software but apparently so many companies are ignoring nowadays.

### 1.2 Installation manual

In this section you can find out how to configure fantastico framework for different purposes.

### 1.2.1 Developing a new fantastico project

Currently fantastico is in early stages so we did not really use it to create new projects. The desired way we want to provide this is presented below:

pip-3.2 install fantastico

Done, now you are ready to follow our tutorials about creating new projects.

### 1.2.2 Contributing to fantastico framework

Fantastico is an open source MIT licensed project to which any contribution is welcomed. If you like this framework idea and you want to contribute do the following (I assume you are on an ubuntu machine):

```
#. Create a github account.
#. Ask for permissions to contribute to this project (send an email to radu.cosnita@gmail.com) - I w
#. Create a folder where you want to hold fantastico framework files. (e.g worspace_fantastico)
#. cd ~/workspace_fantastico
#. git clone git@github.com:rcosnita/fantastico
#. sudo apt-get install python3-setuptools
#. sh virtual_env/setup_dev_env.sh
#. cd ~/workspace_fantastico
#. git clone git@github.com:rcosnita/fantastico fantastico-doc
#. git checkout gh-pages
```

Now you have a fully functional fantastico workspace. I personally use PyDev and spring toolsuite but you are free to use whatever editor you want. The only rule we follow is *always keep the code stable*. To check the stability of your contribution before committing the code follow the steps below:

```
#. cd ~/workspace_fantastico/fantastico/fantastico
#. sh run_tests.sh (we expect no failure in here)
#. sh run_pylint.sh (we expect 9+ rated code otherwise the build will fail).
#. cd ~/workspace_fantastico/fantastico
#. export BUILD_NUMBER=1
#. ./build_docs.sh (this will autogenerate documentation).
#. Look into ~/workspace_fantastico/fantastico-doc
#. Here you can see the autogenerated documentation (do not commit this as Jenkins will do this for y
#. Be brave and push your newly awesome contribution.
```

## BUILD STATUS

If you want to see the current build status of the project visit [Build status](#).