# Publishing a Pillar Booklet

Stéphane Ducasse

November 6, 2017
tag: V6 @ 5136df2




Layout and typography based on the `sbabook` LaTeX class by Damien Pollet.

# Contents

# Illustrations

The current document is a little tutorial to show you how to begin with the creation of booklet. It describes the installation, customisation and integration with travis and bintray online services. From that perspective it can be mapped to other integration services.

We will use bintray to store the latest pdf produced by travis and we will use github to store the released pdf.

CHAPTER 1

# Creating and Configuring a Booklet

In this part we will describe the complete setup starting just after the creation of a git repository to version the booklet files. The example used for this is this booklet itself.

We take as hypothesis that you want to version your document on gihub but Pillar does not care about this. Now in the next chapter we will explain how to use github service integration. Such services only work with github.

## 1.1 Installing pillar

Installing pillar is simple. Now since Pillar can be used to generate different families of document, once you download you should install the template you want to use. But let us get started!

### Cloning a git repository

Let us clone the repository which is empty right now.

```
git clone
    git@github.com:SquareBracketAssociates/Booklet-PublishingAPillarBooklet.git
```

```
cd Booklet-PublishingAPillarBooklet
```

### Installing Pillar

Now you are ready to install Pillar using the following script:

```
wget
    https://raw.githubusercontent.com/pillar-markup/pillar/master/download.sh
chmod +x download.sh
./download.sh
```

### Selecting and installing book

Pillar come with several templates and you can define your own using mustache (see the book Entreprise Pharo: a Web perspective). By default pillar offers the following templates hosted at https://github.com/pillar-markup/Pillar-Archetype:

- Book. This is the template used in recent Pharo books and booklets.
- Presentation. These are the beamer templates used for example in the Pharo mooc.
- Basic-latex. This template is good to generate basic reports. What is nice is that you do not need the latest version of latex packages.
- Welcome. This template is to teach how to build templates.

```
./pillar archetype book
```

```
make book
```

or

```
make book spiral
```

produces a book in the folder `build`. Indeed the folder Chapters and figures contain some samples.

## 1.2 Adapting Pillar to your book

Now we are ready to produce your own book. For this we edit some configuration files, in addition to edit some chapters. We focus on the configuration of files.

You need to edit two files:

- pillar.conf which defines all the concerns such as author name, book tile but also the book template.
- book.pillar which lists the chapters in order and is the place to add some glue text.

### Editing pillar.conf

You can start to edit the file `pillar.conf`. The main points are:

- title:

- attribution: Give the author of the booklet.
- chapters. Here you should give the list of the files. You can put them in separate folders or not.

Here is the

```
  {
    "metadata" : {
      "title": "Publishing a Pillar Booklet",
      "attribution": "StéphaneDucasse",
      "series": "Square Bracket tutorials",
      "keywords": "project template, Pillar, Pharo, Smalltalk"
    },
    "outputDirectory": "build",
    "mainDocument": "book",
    "latexTemplate": "support/templates/main.latex.mustache",
    "latexChapterTemplate":
     "support/templates/chapter.latex.mustache",
    "htmlTemplate": "support/templates/html.mustache",
    "htmlChapterTemplate": "support/templates/html.mustache",
    "chapters": [
        "Chapters/GettingStarted",
        "Chapters/TravisBintray"
    "newLine": #unix,
    "configurations": {
      "LaTeX" : {
        "outputType": #'latex:sbabook',
        "separateOutputFiles": true
      },
      "HTML" : {
        "outputType": #html,
        "separateOutputFiles": true
      }
    }
  }
```

### Editing book.pillar

Now we will edit the file `book.pillar` (the file listed in `"mainDocument": "book"` in the `pillar.conf` file).

```
more book.pillar

The current document is a little tutorial to show you how to begin
    with the creation of booklet.
It describes the installation, customisation and integration with
    travis and bintray online services.
${inputFile:Chapters/GettingStarted.pillar}$

${inputFile:Chapters/TravisBintray.pillar}$
```

### Adding a .gitignore

Not all the downloaded files are worth versionning. We suggest the following .gitignore configuration.

```
more .gitignore
>>>
/Pharo.*
/pharo
/pharo-ui
/pillar
/mustache
/pharo-vm/
/build/
```

You can remove the folder figures and the chapter samples. Now you are ready to commit your files.

CHAPTER 2

# Setting Up Travis

Travis is a service similar to Jenkins that is useful to automate your process. Once configured, a travis job will run each time you commit to your book github repository. This is what we will see now.

## 2.1 Configuring your github account

The first to start with is to enable the travis integration from your github repository. You will find the integration in Settings and the menu integration services. You should look for Travis CI and enable it as shown in Figure 2-1.

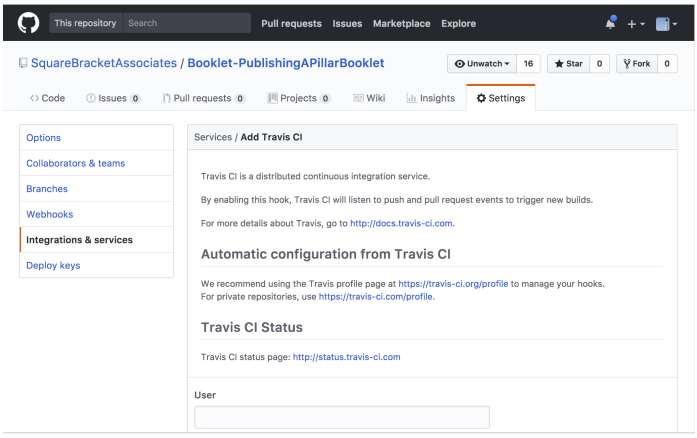


**Figure 2-1** Enabling the travis service in your github account.

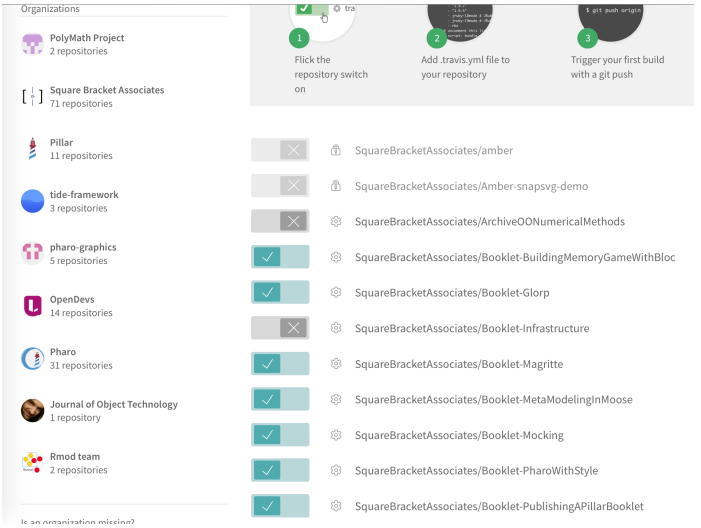


**Figure 2-2** Enabling your github project on travis.

Once this is done, you should go to your travis account http://travis.com and enable your project as shown in Figure .

Note that this setup may take a while to sync. So do not worry if these systems start to off right in the minute. It may take a while, so let us pass to the next item.

## 2.2 Add and edit the .travis.yml file

Here is the `.travis.yml` of this booklet. You can find it online at https://github.com/SquareBracketAssociates/Booklet-PublishingAPillarBooklet:

```
sudo: required
services:
- docker

before_install:
- docker pull dpollet/texlive:pillar

script:
- docker run --tty --rm --volume $PWD:/work dpollet/texlive:pillar
    make spiralbook
```

Add a similar configuration file to your repository. Once the travis github get synchronised and travis is kicked in, you will be able to check on the travis log that your project has been successfully built.

## 2.3 Some explanations

Let us explain some parts:

```
sudo: required
services:
- docker

before_install:
- docker pull dpollet/texlive:pillar

script:
- docker run --tty --rm --volume $PWD:/work dpollet/texlive:pillar
    make spiralbook
```

Our script uses the docker container containing an installed version of texlive and pillar. The script invokes make to build the book. `make spiralbook` builds the book in spiral lulu format. You may want to adapt it to other formats.

## 2.4 Using badges to indicate build status

You can now use the status of a travis build right in the github repository using markdown badges in the project `README.md` file.

```
# A booklet explaining how to build a booklet

[![Build status][badge]][travis]

[travis]:
    https://travis-ci.org/SquareBracketAssociates/Booklet-PublishingAPillarBooklet
[badge]: https://travis-ci.org/SquareBracketAssociates/
Booklet-PublishingAPillarBooklet.svg?branch=master

## To contribute
- Fork
- Do pull Request

## To latex it locally

```
make
make bootstrap
make book spiral
```
```

At the stage you should get already important feedback since you will know if your project fully builds or not.

CHAPTER 3

# Deploying work in progresss artefacts in bintray

Bintray is a service that can host artefacts produced by Travis. We will use bintray to store the latest pdf produced by travis and we will use github to store the released (tagged) pdf.

## 3.1 Create a bintray account

You should have an account on bintray https://bintray.com/. You may link it or not to your github account.

## 3.2 Encoding your bintray API key for Travis

In your bintray profile, get your API key (copy it to your copy paste buffer). This is key is important because without it you will not be able to make Travis use Bintray. To be used in Travis it should be encrypted.

Note that this encrypted key cannot be shared between different project, therefore it should be generated from the bintray API key for each new project.

In your machine in the folder of your project execute the following expressions:

```
travis login --auto
travis encrypt 'Your API Key from Bintray'
```

You will use the resulting key in your .travis.yml configuration file.

## 3.3 Enabling bintray

To enable bintray, we extend the `.travis.yml` description with the following section:

```
deploy:
- provider: bintray
  file: .bintray.json
  on:
    repo: SquareBracketAssociates/Booklet-PublishingAPillarBooklet
  skip_cleanup: true
  user: ducasse
  key:
    secure:
    DLT5e+3+U3jaooTLVN6JPcavMS+5XzFwNIfsK7QLYefzOZfBR8lt2vHtX2OxcdyKNfuzh0EaIDUR9a6+/
```

The deploy section indicates to use the file `.bintray.` as a specification to store the result of the build to the bintray repository for the account `ducasse` and this is there that you should paste your encrypted key.

Note that the `skyp_cleanup: true` is a way to circumvent some strange behavior.

## 3.4 Add and edit the .bintray.json file

Now we should add a `.bintray.json` configuration file that you should version with your project.

Here is the one of the https://github.com/SquareBracketAssociates/Booklet-PublishingAPillarBookle

```
more .bintray.json
```

```
{
  "package": {
      "name": "Booklet-PublishingABooklet",
      "repo": "wip",
      "subject": "squarebracketassociates",
      "desc": "Distributing",
      "licenses": ["MIT"],
      "website_url":
    "https://github.com/SquareBracketAssociates/Booklet-PublishingAPillarBooklet",
      "issue_tracker_url":
    "https://github.com/SquareBracketAssociates/Booklet-PublishingAPillarBooklet/issu
      "vcs_url":
    "https://github.com/SquareBracketAssociates/Booklet-PublishingAPillarBooklet.git"
      "labels": ["work-in-progress"]
    },

    "version": {
```

```
        "name": "latest",
        "desc": "Latest successful build"
    },

    "files": [
        {
            "includePattern": "build/(learningoop).pdf",
            "uploadPattern": "$1-wip.pdf",
            "matrixParams": { "override": 1 }
        }
    ],
    "publish": true
}
```

The most important is probably the files part. It lets you specify the files that you select in the build and how the files can be renamed when uploaded to bintray. `$1` denotes the part in `()`.

- `"includePattern": "build/(learningoop).pdf"`, states that the file to grab is named learningoop.pdf and is found in the `build` folder.
- `"uploadPattern": "$1-wip.pdf"`, states that when the file will be uploaded to bintray we renaming it `learningoop-wip.pdf`.

## 3.5 Put a point to bintray in your README

Now

```
The result from the latest successful Travis build can be found [on
    Bintray
    (PDF)](https://bintray.com/squarebracketassociates/wip/download_file?file_path
```

CHAPTER 4

# For Github integration

For a given checkout you should generate the associated token. In your github repository you should mention to travis that you will be interested by sotring releases files in the github release tab.

```
> travis setup releases
```

Here is the full .travis.yml with the github integration

```
sudo: required
services:
- docker

before_install:
- docker pull dpollet/texlive:pillar

script:
- docker run --tty --rm --volume $PWD:/work dpollet/texlive:pillar
    make spiralbook

deploy:
- provider: bintray
  file: .bintray.json
  on:
    repo: SquareBracketAssociates/LearningOOPWithPharo
  skip_cleanup: true
  user: ducasse
  key:
    secure:
    ThnjhD8OMRlL5NmsK8dGQR4mpzXG2Kj2NeWHA5oHCO4YTaK0fKLPHF8YcjlyQorESX6+Z/Q0ihtTq1
- provider: releases
  file: build/learningoop.spiral.pdf
```

```
  on:
    tags: true
  skip_cleanup: true
  api_key:
    secure:
     pRfu3fr35BzqQDi1ul0SubPSwO8zcyIidqLRZPz3XIrH4qoy1Zgnq785dmB7ziNQqH+VJyh7askSFY3JL
```

The expression file: `build/learningoop.pdf` identifies the file produced by `make`. Pay attention because spiralbook produces learningoop.spiral.pdf

The expression

```
  on:
    tags: true
```

expresses the fact that the script will only trigger for

Note: We should add how we generated the key. It is not your bintray key. Probably travis encrypt your github token key.

CHAPTER 5

# How to add a new released file in your git hub account

To release a pdf that will be stored on github in the booklet repository, we should create an annotated tag.

```
git tag -a v1.0-Pharo50
git push --tags
```