

# Jenkins

Systemtechnik Labor  
5BHIT 2015/16, Gruppe A

Hagen Aad Fock

Version 0.1

Betreuer: Prof. Dolzeal

Begonnen am 20.11.2015

Note:

Beendet am 20.11.2015

## Inhaltsverzeichnis

Aufgabenstellung.....	3
Zeitaufwand.....	4
Github Repo Link .....	4
Ergebnisse .....	4
Quellen .....	9

## Aufgabenstellung

*"Continuous Integration is a software development practice where members of a team integrate their work frequently, usually each person integrates at least daily - leading to multiple integrations per day. Each integration is verified by an automated build (including test) to detect integration errors as quickly as possible. Many teams find that this approach leads to significantly reduced integration problems and allows a team to develop cohesive software more rapidly. This article is a quick overview of Continuous Integration summarizing the technique and its current usage." M.Fowler*

Schreibe fünf Testfälle für dein CSV-Projekt und lass diese mithilfe von Jenkins automatisch bei jedem Build testen!

- Installiere auf deinem Rechner bzw. einer virtuellen Instanz das Continuous Integration System Jenkins
- Installiere die notwendigen Plugins für Jenkins (Git Plugin, Violations, Cobertura)
- Installiere Nose und Pylint (mithilfe von pip)
- Integriere dein CSV-Projekt in Jenkins, indem du es mit Git verbindest
- Schreibe fünf Unit Tests für dein CSV-Projekt
- Konfiguriere Jenkins so, dass deine Unit Tests automatisch bei jedem Build durchgeführt werden inkl. Berichte über erfolgreiche / fehlgeschlagene Tests und Coverage
- Protokolliere deine Vorgehensweise (inkl. Zeitaufwand, Konfiguration, Probleme) und die Ergebnisse (viele Screenshots!)

Viel Spaß!

## Zeitaufwand

Installation der einzelnen Komponenten	0.6 h
Code + Tests	0.6 h
Einrichtung vom Jenkins	0.5 h
Troubleshooting Coverage	0.5 h
Dokumentation	0.6 h
Gesamt	2.8 h

## Github Repo Link

<https://github.com/hfock-tgm/JenkinsCSV>

## Ergebnisse

Für die Installation von dem Continuous Integration System Jenkins verwende ich eine virtuelle Instanz (Ubuntu V15).

Für die Installation von Jenkins habe ich auf der offiziellen Seite einen Link [1] gefunden und befolgt.

Mit folgender URL bin ich dann auf den Jenkins Server gekommen

<code>http://localhost:8080/</code>
-------------------------------------

Für die Installation von den Plugins muss man bei der Weboberfläche vom Jenkins den Plugin Manager benutzen und dort einfach die Plugins auswählen und installieren.












(Git Plugin, Violations, Cobertura)

Kleiner Hinweis: Bei mir hat es ausgesehen als hätte das Violations plugin ewig irgendetwas heruntergeladen, jedoch hat der Jenkins sich anscheinend nicht mehr selbst aktualisiert, deswegen musste man nur das Fenster refreshen.


## Installing Plugins/Upgrades

### Preparation

- Checking internet connectivity
- Checking update center connectivity
- Success

Credentials Plugin	 credentials plugin is already installed. Jenkins needs to be restarted for the update to take effect
SSH Credentials Plugin	 ssh-credentials plugin is already installed. Jenkins needs to be restarted for the update to take effect
Git client plugin	 Success
SCM API Plugin	 Success
JUnit Plugin	 junit plugin is already installed. Jenkins needs to be restarted for the update to take effect
Matrix Project Plugin	 matrix-project plugin is already installed. Jenkins needs to be restarted for the update to take effect
Mailer Plugin	 mailer plugin is already installed. Jenkins needs to be restarted for the update to take effect
Git plugin	 Success
Cobertura Plugin	 Success
Violations plugin	 Installing <div data-bbox="496 835 659 846" style="width: 100%;"></div>
Restarting Jenkins	 Pending

➡ [Go back to the top page](#)  
(you can start using the installed plugins right away)

➡  Restart Jenkins when installation is complete and no jobs are running

Für die Installation von Nose und Pylint habe ich folgende Kommandos eingegeben.

```
pip install Nose
pip install pylint
```

Für die Erstellung des Jenkins Projektes/Job habe ich fast 1 zu 1 das Tutorial [6] nachgemacht, bis auf den PYTHONPATH Teil, welchen ich leicht abgewandelt habe.

Anstatt von

```
PYTHONPATH=' '
nosetests --with-xunit --all-modules --traverse-namespace --with-coverage --cover-package=project1 --cover-inclusive
python -m coverage xml --include=project1*
pylint -f parseable -d I0011,R0801 project1 | tee pylint.out
```

habe ich

```
PYTHONPATH=' '
Nosetests-3.4 --with-xunit --all-modules --traverse-namespace --with-coverage --cover-inclusive
python3 -m coverage xml
pylint -f parseable -d I0011,R0801 CSVReader.py | tee pylint.out
```

verwendet.

Nachdem ich alles eingerichtet hatte, hat alles bis auf die Coverage funktioniert.

```
Started by an SCM change
Building in workspace /var/lib/jenkins/jobs/csv/workspace
> git rev-parse --is-inside-work-tree # timeout=10
Fetching changes from the remote Git repository
> git config remote.origin.url https://github.com/hfock-tgm/JenkinsCSV.git #
timeout=10
Fetching upstream changes from https://github.com/hfock-tgm/JenkinsCSV.git
> git --version # timeout=10
> git -c core.askpass=true fetch --tags --progress https://github.com/hfock-
tgm/JenkinsCSV.git +refs/heads/*:refs/remotes/origin/*
> git rev-parse refs/remotes/origin/master^{commit} # timeout=10
> git rev-parse refs/remotes/origin/origin/master^{commit} # timeout=10
Checking out Revision 8c6b0b6d136c67dbdda6e213fe310bfde36b7321
(refs/remotes/origin/master)
> git config core.sparsecheckout # timeout=10
> git checkout -f 8c6b0b6d136c67dbdda6e213fe310bfde36b7321
> git rev-list 01c4c4cfef6607d5edc5955cc0a5507d901e9873 # timeout=10
[workspace] $ /bin/sh -xe /tmp/hudson6894571619213337267.sh
+ PYTHONPATH=
+ nosetests --with-xunit --all-modules --traverse-namespace --with-coverage --
cover-inclusive
nose.plugins.cover: ERROR: Coverage not available: unable to import coverage module
.....
-----
Ran 5 tests in 0.040s

OK
+ python -m coverage xml
No data to report.
Build step 'Execute shell' marked build as failure
Skipping Cobertura coverage report as build was not UNSTABLE or better ...
Recording test results
Finished: FAILURE
```

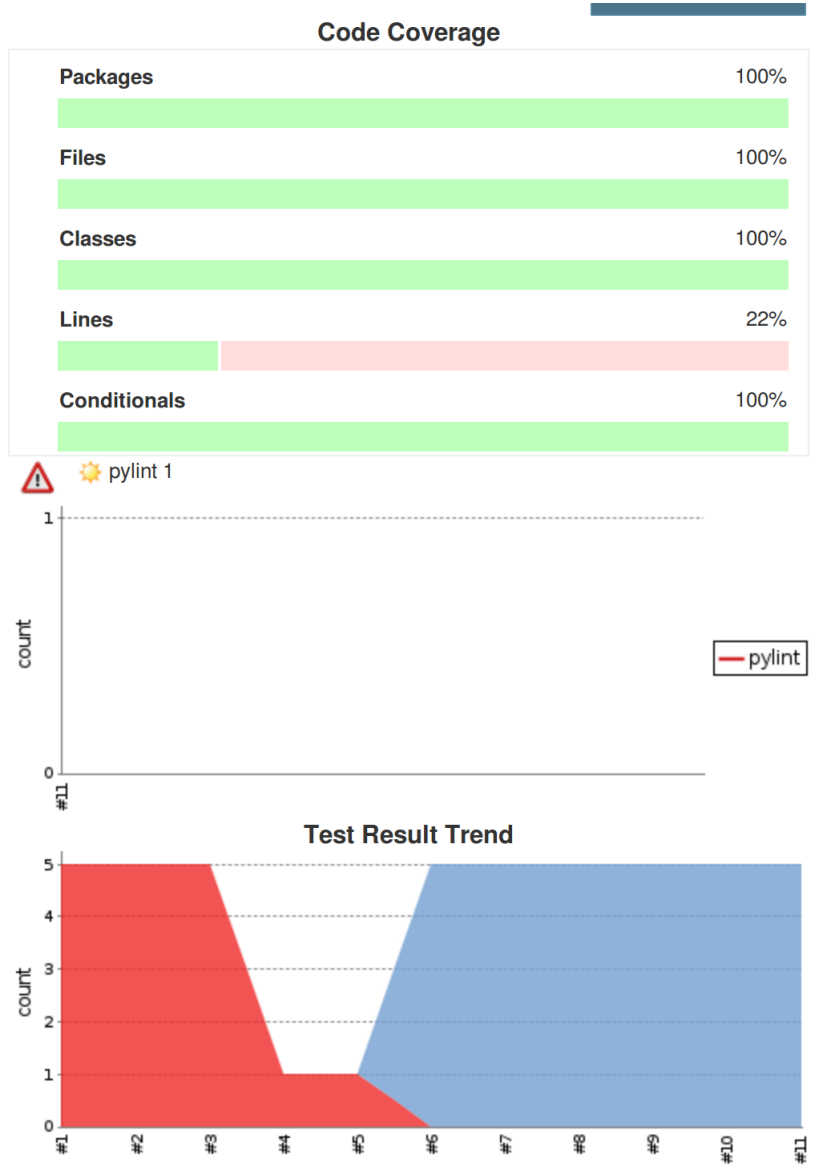
Nachdem ich folgende Befehle ausgeführt habe hat es dann endlich funktioniert.

```
sudo apt-get install python3-coverage
```

```
sudo chown jenkins:jenkins  
/var/lib/jenkins/jobs/csv/workspace/.coverage
```

```
fock@fock-VirtualBox:/var/lib/jenkins/jobs/csv/workspace$ sudo chown jenkins:jenkins .coverage  
fock@fock-VirtualBox:/var/lib/jenkins/jobs/csv/workspace$ ls -la  
total 284  
drwxr-xr-x 6 jenkins jenkins 4096 Feb 16 13:42 .  
drwxr-xr-x 4 jenkins jenkins 4096 Feb 16 13:42 ..  
drwxr-xr-x 2 jenkins jenkins 4096 Feb 16 10:29 Bilder  
-rw-r--r-- 1 jenkins jenkins 7008 Feb 16 13:40 .coverage  
-rwxr-xr-x 1 jenkins jenkins 10 Feb 16 13:42 csv_file.csv  
-rwxr-xr-x 1 jenkins jenkins 241334 Feb 16 10:55 Fock_Hagen_Jenkins.docx  
drwxr-xr-x 8 jenkins jenkins 4096 Feb 16 13:42 .git  
drwxr-xr-x 3 jenkins jenkins 4096 Feb 16 10:55 .idea  
-rwxr-xr-x 1 jenkins jenkins 850 Feb 16 13:42 nosetests.xml  
-rwxr-xr-x 1 jenkins jenkins 1282 Feb 16 10:29 QT-CSV.ui  
-rwxr-xr-x 1 jenkins jenkins 13 Feb 16 10:29 README.md  
drwxr-xr-x 4 jenkins jenkins 4096 Feb 16 10:29 src  
fock@fock-VirtualBox:/var/lib/jenkins/jobs/csv/workspace$
```

Schlussendlich hat dann alles funktioniert.





## Quellen

- [1] Installation von Jenkins auf Debian/Ubuntu
  - Online: <http://pkg.jenkins-ci.org/debian/>
    - Zuletzt gesehen am 09.02.2016
- [2] Python Guide; Continuous Integration
  - Online: <http://docs.python-guide.org/en/latest/scenarios/ci/>
    - Zuletzt gesehen am 15.02.2016
- [3] Jenkins
  - Online: <https://jenkins-ci.org/>
    - Zuletzt gesehen am 15.02.2016
- [4] Steve's Blog; Automated python unit testing Part1
  - Online: <http://bhfsteve.blogspot.co.at/2012/04/automated-python-unit-testing-code.html>
    - Zuletzt gesehen am 15.02.2016
- [5] Steve's Blog; Automated python unit testing Part2
  - Online: [http://bhfsteve.blogspot.co.at/2012/04/automated-python-unit-testing-code\\_20.html](http://bhfsteve.blogspot.co.at/2012/04/automated-python-unit-testing-code_20.html)
    - Zuletzt gesehen am 15.02.2016
- [6] Steve's Blog; Automated python unit testing Part3
  - Online: [http://bhfsteve.blogspot.co.at/2012/04/automated-python-unit-testing-code\\_27.html](http://bhfsteve.blogspot.co.at/2012/04/automated-python-unit-testing-code_27.html)
    - Zuletzt gesehen am 15.02.2016