

PyCharm

Projekt: DigitalSchoolNotes

Projekt Team: Adler, Brinnich, Hohenwarter, Karic, Stedronsky

Version 2.0

15.12.2015

Status: [RELEASE]

	Datum	Name	Unterschrift
Erstellt	07.10.2015	Selina Brinnich	
Geprüft	15.12.2015	Niklas Hohenwarter	
Freigegeben			
Git-Pfad: /doc/technologien Dokument: pycharm_technologie.doc			

Inhaltsverzeichnis

1	CHANGELOG.....	3
2	EINLEITUNG	4
3	KONFIGURATION.....	4
3.1	REMOTE INTERPRETER	4
3.2	FILE SYNCHRONISATION.....	5
3.3	START CONFIGURATIONS.....	5
3.4	SOURCE ROOT.....	6
4	ABBILDUNGSVERZEICHNIS	7
5	QUELLEN.....	7

1 Changelog

Version	Datum	Status	Bearbeiter	Kommentar
0.1	2015-10-07	Erstellt	Selina Brinnich	Dokument erstellt
1.0	2015-10-07	Geprüft	Niklas Hohenwarter	QA
1.1	2015-12-15	Bearbeitet	Thomas Stedronsky	Abbildungsverzeichnis und passive Sprache, SCM mit Git
2.0	2015-12-15	Geprüft	Niklas Hohenwarter	QA

2 Einleitung

Das Team entscheidet sich dazu für die Entwicklung eine IDE zu verwenden. Die Wahl fiel auf PyCharm, da wir im Unterricht bereits die verwandte IDE IntelliJ verwendet hatten und daher mit der Handhabung von PyCharm schon ein bisschen vertraut waren. Des Weiteren wurde die IDE von Prof. Rafeiner empfohlen. Dieser half auch bei der Erst-Konfiguration mit.

3 Konfiguration

3.1 Remote Interpreter

In PyCharm kann man sogenannte Remote Interpreter hinzufügen[1]. Dies bedeutet, dass Lokal in PyCharm programmiert werden kann und sobald ein Test durchgeführt werden soll, wird es auf den Server geladen und dort ausgeführt. Der Vorteil hiervon ist, dass Lokal keine Libraries oder Datenbanken installiert werden müssen. Als erster Schritt muss der Remote Interpreter hinzugefügt werden. Dies geht in den Einstellungen des jeweiligen Projektes (STRG+ALT+S).

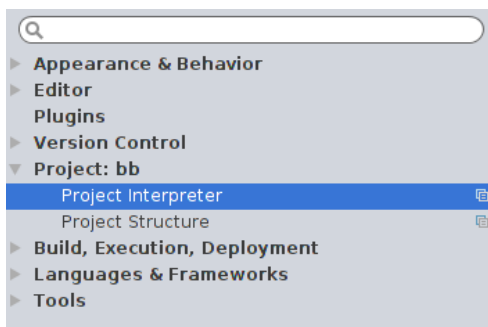


Abbildung 1 Projekt-Einstellung

Hier klickt man nun aufs Zahnrad oben rechts und wählt „Add Remote“ aus.

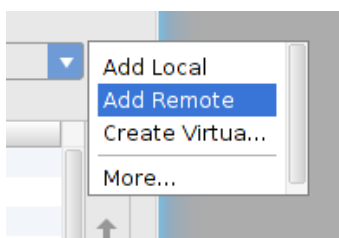


Abbildung 2 Add Remote

Hier gibt man nun die entsprechenden Verbindungsinformationen an. Danach muss der OK-Button gedrückt werden. Wenn dies beendet ist, kann man das Konfigurationsfenster wieder mit dem OK-Button schließen. Nun sieht man am unteren Fensterrand noch einen Fortschrittsbalken. Dieser indiziert den Downloadfortschritt der Library Skelettons vom Server.

3.2 File Synchronisation

Um die bearbeiteten Files in PyCharm automatisch am Server hochzuladen, muss die Deployment Konfiguration erstellt werden[2]. Dies lässt sich unter dem Menüpunkt „Tools > Deployment > Configuration“ machen. Hier drückt man auf das Plus um eine neue Konfiguration hinzuzufügen. Dann gibt man die Verbindungsinformationen ein.

The screenshot shows the 'Deployment Configuration' dialog box in PyCharm, specifically the 'Connection' tab. The 'Type' is set to 'SFTP'. Below this, it states 'Project files are deployed to a remote host via SFTP'. The 'Upload/download project files' section contains the following fields: 'SFTP host' (178.62.1.99), 'Port' (22), 'Root path' (/home/niklas/bb), 'User name' (niklas), 'Auth type' (Key pair (OpenSSH or PuTTY)), 'Private key file' (/home/user/.ssh/id_rsa), and 'Passphrase' (empty). There are buttons for 'Test SFTP connection...', 'Autodetect', and 'Advanced options...'. At the bottom, the 'Web server root URL' is set to http://bringbuy.net:5002 with an 'Open' button.

Abbildung 3 File Synchronisation

Unter Mappings muss nun noch der Pfad zum Source angepasst werden.

3.3 Start Configurations

Um das Programm starten zu können muss noch eine entsprechende „Start Configuration“ angelegt werden [3]. Die Einstellung findet man unter „Run > Edit Configurations“.

The screenshot shows the 'Run Configuration' dialog box in PyCharm for a 'Start' configuration. The 'Name' is 'Start'. The 'Script' is /home/niklas/bb/src/run.py. The 'Script parameters' are empty. Under 'Environment', the 'Environment variables' are set to PYTHONUNBUFFERED=1. The 'Python interpreter' is 'Project Default (Remote Python 2.7.5 (ssh://niklas@178.62.1.99:22/usr/bin/python))'. The 'Working directory' is /home/niklas/bb/src/. The 'Path mappings' are empty. There are checkboxes for 'Add content roots to PYTHONPATH' (checked), 'Add source roots to PYTHONPATH' (checked), and 'Show command line afterwards' (unchecked).

Abbildung 4 Start Configurations

Unter Script setzt man den Pfad zum Startfile am Remoteserver (NICHT der lokale Pfad). Das „Working Directory“ muss ebenfalls gesetzt werden (ebenfalls nicht lokal).

3.4 Source Root

Damit PyCharm den Code und die Pakete richtig interpretiert, muss der Source Root geändert werden [4]. Dies kann man im Project Explorer machen. Nehmen wir an wir wollten „src“ als Root angeben.

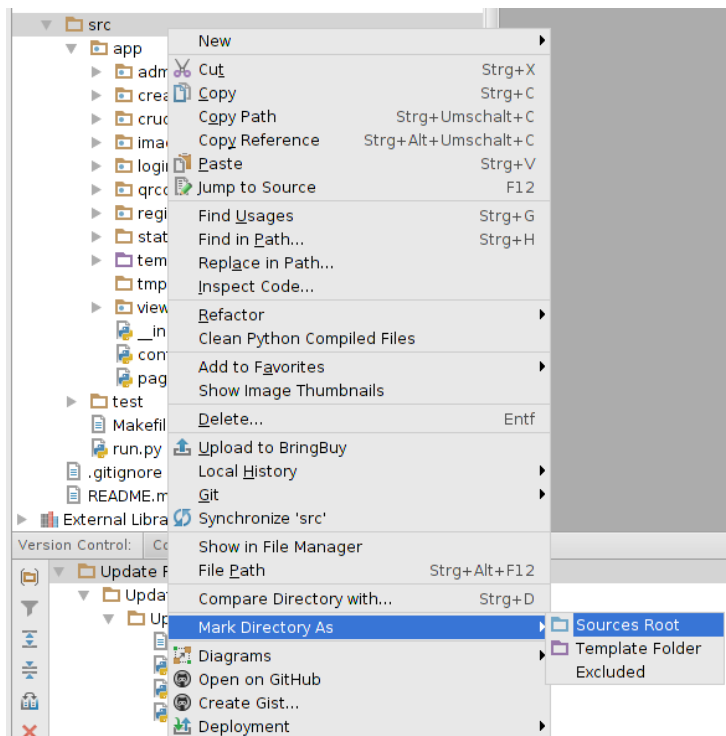


Abbildung 5 Source Root

4 Git in Pycharm

Falls sich das Projekt innerhalb eines Git Repositories befindet, erkennt PyCharm dies automatisch. Es kann auf drei Arten mit dem Repository interagiert werden:

- Auf ein File rechtsklicken und dann das Untermenü „Git“ auswählen
- Mithilfe der Werkzeugleiste
- Durch die Hauptmenüleiste unter „VCS > Git“

5 Abbildungsverzeichnis

Abbildung 1 Projekt-Einstellung	4
Abbildung 2 Add Remote.....	4
Abbildung 3 File Synchronisation.....	5
Abbildung 4 Start Configurations.....	5
Abbildung 5 Source Root	6

6 Quellen

[1], PyCharm, „Configuring Remote Python Interpreters“,
<https://www.jetbrains.com/pycharm/help/configuring-remote-python-interpreters.html>, zuletzt
besucht: 29.03.2015

[2], PyCharm, „Creating a Remote Server Configuration“,
<https://www.jetbrains.com/pycharm/help/creating-a-remote-server-configuration.html>, zuletzt
besucht: 29.03.2015

[3], PyCharm, „Creating and Editing Run/Debug Configurations“,
<https://www.jetbrains.com/pycharm/help/creating-and-editing-run-debug-configurations.html>,
zuletzt besucht: 29.03.2015

[4], PyCharm, „Content Root“,
<https://www.jetbrains.com/pycharm/help/content-root.html>, zuletzt besucht: 29.03.2015