

PyCharm

Projekt: DigitalSchoolNotes

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

Version 1.0

07.10.2015

Status: [RELEASE]

	Datum	Name	Unterschrift
Erstellt	07.10.2015	Selina Brinnich	
Geprüft	07.10.2015	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	QUELLEN.....	6

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

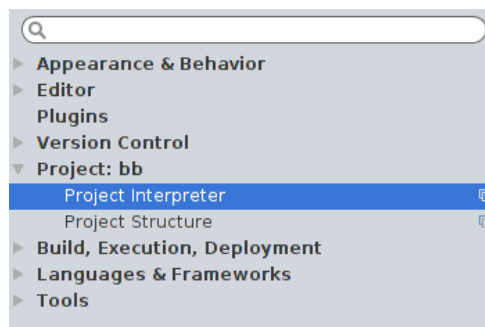
2 Einleitung

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 uns die IDE von Prof. Rafeiner empfohlen. Dieser half auch bei der Erst-Konfiguration mit.

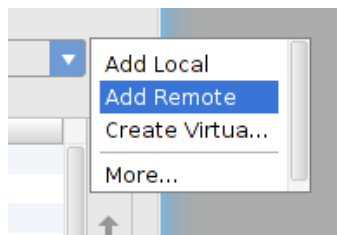
3 Konfiguration

3.1 Remote Interpreter

In PyCharm kann man sogenannte Remote Interpreter hinzufügen[1]. Dies bedeutet, dass ich lokal in PyCharm programmieren kann und sobald ich es Testen will, 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 macht man in den Einstellungen des Projektes (STRG+ALT+S).



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



Hier gibt man nun die entsprechenden Verbindungsinformationen an. Sobald der OK Button gedrückt wurde, holt sich PyCharm Informationen vom Server. 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 'SFTP' configuration window in PyCharm. It has three tabs: 'Connection', 'Mappings', and 'Excluded Paths'. The 'Connection' tab is active. At the top, it says 'Type: SFTP' and 'Project files are deployed to a remote host via SFTP'. Below this, there's a section 'Upload/download project files'. It contains fields for '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, there's a section 'Browse files on server' with a 'Web server root URL:' field (http://bringbuy.net:5002) and an 'Open' button.

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

3.3 Start Configurations

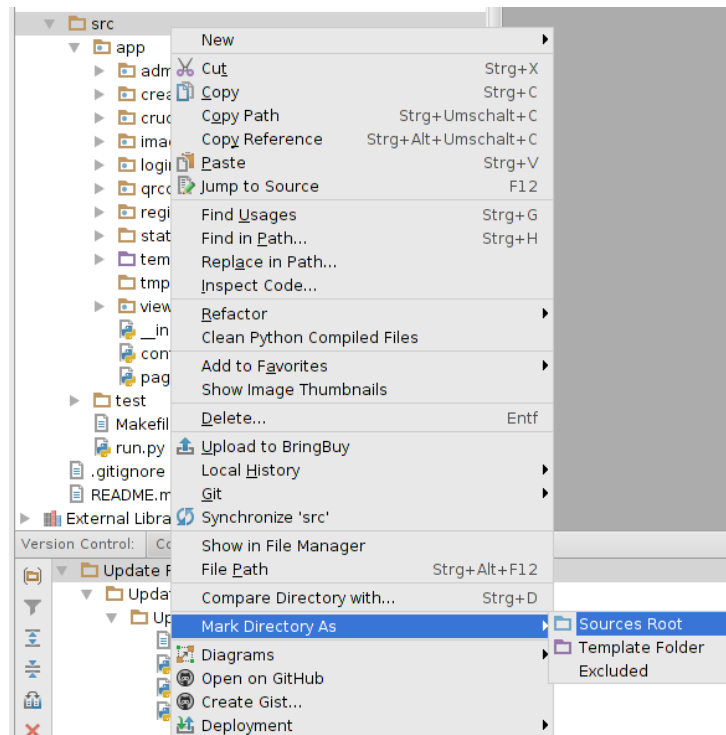
Um das Programm starten zu können muss noch eine entsprechende Start Configuration angelegt werden [3]. Die Einstellung findet man unter Run > Edit Configurations.

The screenshot shows the 'Run Configuration' dialog in PyCharm. It has two tabs: 'Configuration' and 'Logs'. The 'Configuration' tab is active. At the top, there's a 'Name:' field (Start) and checkboxes for 'Share' and 'Single instance only'. Below this, there's a section 'Script'. It contains fields for 'Script:' (/home/niklas/bb/src/run.py), 'Script parameters:' (empty), 'Environment' (expanded), 'Environment variables:' (PYTHONUNBUFFERED=1), 'Python interpreter:' (Project Default (Remote Python 2.7.5 (ssh://niklas@178.62.1.99:22/usr/bin/python))), 'Interpreter options:' (empty), 'Working directory:' (/home/niklas/bb/src/), and 'Path mappings:' (empty). At the bottom, there are checkboxes for 'Add content roots to PYTHONPATH' (checked), 'Add source roots to PYTHONPATH' (checked), and 'Show command line afterwards' (unchecked).

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.



4 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