EP2 Eierlegende Wollmilchsau

12.03.2021

Installation & Einrichtung

Notwendige Pakete/Software

- Python >= 3.6 **(!)** mit `pip`
- Git
- ssh (wird meistens mit Git mitinstalliert)

> Python 2.* ist EOL seit 01.01.2020, deswegen gibt es keinen Support mehr.
>
> Unter Windows ist die Installation von Python, wie die Erfahrung zeigt, auf viele verschiedene Varianten möglich.
> Bitte stellt sicher, dass ihr `python` oder `python3` ausführen könnt.

SSH Keys einrichten

SSH Keys können mit `ssh-keygen` erzeugt werden (`ssh-keygen.exe` unter Windows)

```
ssh-keygen
```

- Schlüssel befinden sich im Verzeichnis `~/.ssh`
- Public keys enden mit `.pub`

> **Für Windows 10**:
>
> ```
> Set-Service -Name ssh-agent -StartupType Automatic
> Set-Service -Name ssh-agent -Status Running
> ```

SSH Config

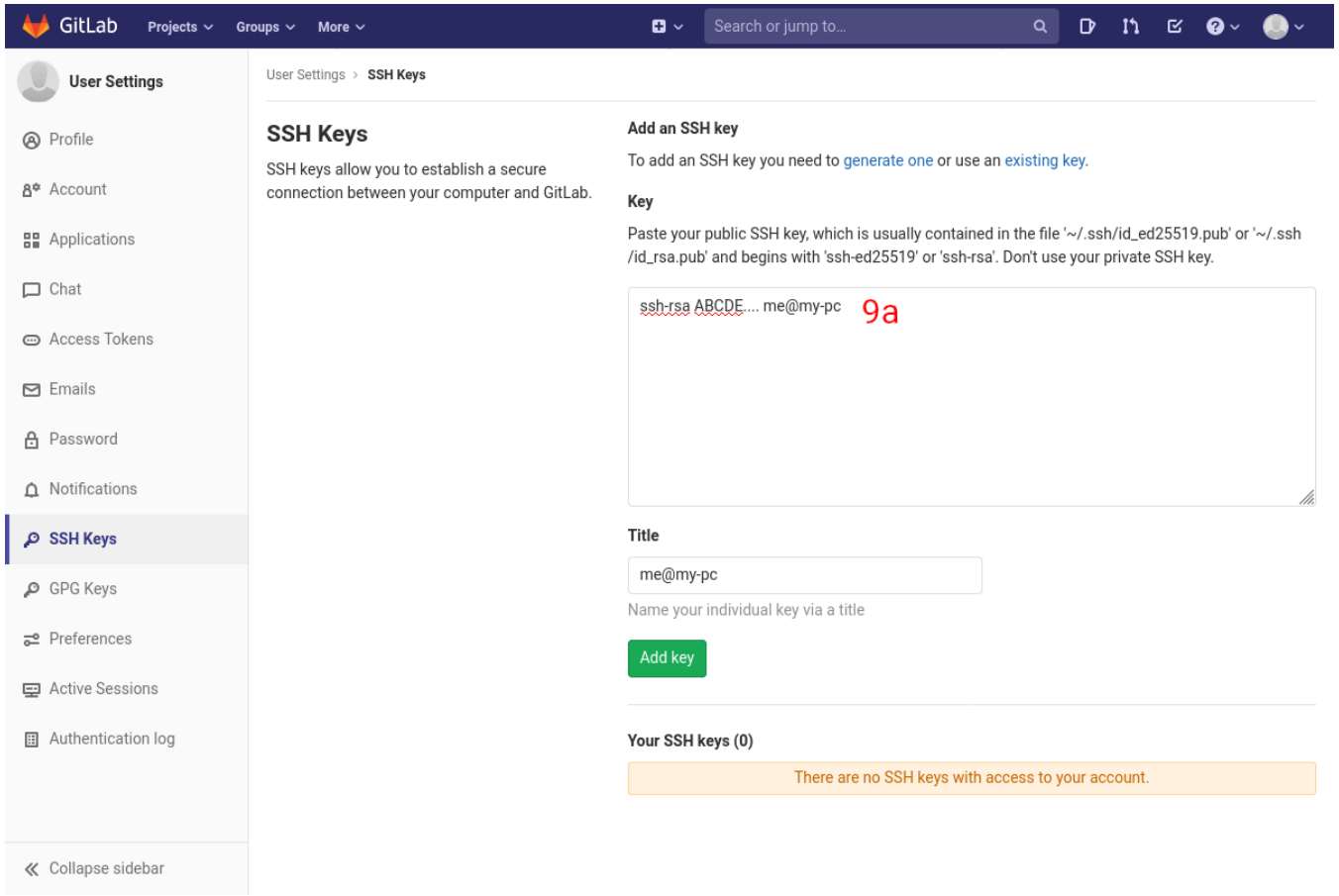
Falls die Schlüssel nicht im `~/.ssh` Verzeichnis liegen, oder ein anderer Schlüssel verwendet werden soll:

Datei `~/.ssh/config` mit folgendem Inhalt erzeugen:

```
Host b3.complang.tuwien.ac.at
    IdentityFile <Pfad zu Key>
```

SSH Keys einrichten

Schlüssel auf Gitlab hinzufügen: https://b3.complang.tuwien.ac.at/profile/keys



SSH Keys testen

Zum Schlüssel testen:

```
ssh git@b3.complang.tuwien.ac.at
```

GitLab sollte dann mit eurem Namen grüßen!

Tools installieren

Tools können direkt über `pip` installiert werden

```
pip3 install --user ep2-tutor-scripts==0.4.2
```

> Zum Aktualisieren zusätzlich noch die Flag `-U` angeben. z.B.:
>
> ```
> pip3 install -U --user ep2-tutor-scripts==0.4.2
> ```

Tools installieren (Linux/Unix)

Um die Tools direkt ausführen zu können muss der Pfad ergänzt werden.
Dazu die folgenden Zeilen in `~/.bashrc` oder in einem anderen Startup File hinzufügen.

```
export PY_USER_BIN=$(python -c 'import site; print(site.USER_BASE + "/bin")')
export PATH=$PY_USER_BIN:$PATH
```

Tools installieren (Windows)

Um die Tools direkt ausführen zu können muss der Pfad ergänzt werden.
Dazu untenstehendes Kommando ausführen und das Ergebnis zur `PATH` Umgebungsvariable hinzuzufügen.

```
python -c "import site; print(site.USER_BASE + '\\Scripts')"
```

> Alternativ: `%APPDATA%\Python\Python37\Scripts`

GitLab Access Token

Token kann auf https://b3.complang.tuwien.ac.at/profile/personal_access_tokens erzeugt werden.

Wichtig:

- Token muss das ganze Semester gelten!
- Token braucht Zugriffsberechtigung `api`
- Sobald jemand diesen Token hat, hat die Person Zugriff auf alles, auf das ihr auch Zugriff habt

> ⇒ **Token = Private Key!**

Tools einrichten

```
ep2_util init --semester=2021s \
              --git-home=$EP2_GIT_HOME \
              --idea-eval-dir=$EP2_IDEA_EVAL_PATH
```

`--semester=2021s` setzt Werte für `--gitlab-url` und `--gitlab-repo-prefix` für dieses Semester.

`--git-home` In diesem Verzeichnis werden die verschiedenen Repositories angelegt. Auch eine eigene Instanz des Tutorinnen Repositories!

`--idea-eval-dir` Ist ein **leeres** Verzeichnis, in das Projekte kopiert werden, um sie in IntelliJ schnell anzeigen zu können.

Mit `--tutor-repo` kann der Pfad zu einem bestehenden Clone des `tutorinnen` Repositories angegeben werden. Dann wird dieses verwendet.

Tools testen

Zum Testen einfach

```
ep2_util test
```

aufrufen

Bei Encoding Problemen

Es werden für euch User auf der `b3` angelegt.

Mit diesen könnt ihr dann die Issues erzeugen (Infos folgen)

Verwendung

Tools

- `ep2_util` Tool um die Skripten einzurichten und zu testen, sowie Tools, die allgemein verwendet werden können
- `ep2_ex_test` Tool um die Übungstests zu bewerten
- `ep2_eval` Tool um die Vorbewertung durchzuführen

Vorbewertung `ep2_eval`

Alle Repositories auschecken

```
ep2_eval checkout --group <group> --ue <ue>
```

Bewerten mit Tool oder direkt im CSV

```
ep2_eval grade --group <group> --ue <ue> [--student <mat. no> ...]
```

Noch ausstehende Bewertungen:

```
ep2_eval list ungraded --group <group> --ue <ue>
```

Wenn alle Bewertungen eingetragen sind, Issues erzeugen:

```
ep2_eval submit --group <group> --ue <ue>
```

Vorbewertung `ep2_eval`

Alle Repositories auschecken

```
ep2_eval checkout --group test --ue 1
```

Bewerten mit Tool oder direkt im CSV

```
ep2_eval grade --group test --ue 1 --student 11777729
```

Noch ausstehende Bewertungen:

```
ep2_eval list ungraded --group test --ue 1
```

Wenn alle Bewertungen eingetragen sind, Issues erzeugen:

```
ep2_eval submit --group test --ue 1
```

CSV Files

- Alle LVA-relevanten Daten werden in CSV (**Comma* Separated Values) Dateien gespeichert
- Zum Eintragen entweder das Tool oder ein Tabellenkalkulationstool (LibreOffice Calc, MS Excel, sc-im, ...) verwenden

Alle CSVs benutzen Kommas als Trennzeichen und Texte werden mit Anführungszeichen `"` umgeben.
Encoding ist `UTF-8` besonders bei Windows darauf achten (gab schon häufig Probleme)

Vorbewertung `ep2_eval` - CSV

Spalte	Gültige Werte	Beschreibung
`id`	Matrikelnummer	Matrikelnummer
`attendend`	`0` = no/ `1` = yes	Anwesenheitskontrolle
`grading`	`((\d*\.\d+|\d+\.?),)*` `(\d*\.\d+|\d+\.?)`	Bewertung
`pre_eval_ex_<x>`	`(?<points>\d*)\/(?<of>\d*)` where `of` ≥ `points`	Ergebnis der automatischen Testung
`remarks`	Text	Infos für die ÜGL
`feedback`	Text	Feedback für Studis

Übungstests `ep2_ex_test`

Nach der Abgabe alle Repositories taggen

```
ep2_ex_test tag --group <group> --ue <ue>
```

dann alle Repositories aktualisieren:

```
ep2_ex_test checkout --group <group> --ue <ue>
```

Übungstests `ep2_ex_text` cont.

Dann Abgaben eingeben (Eingabe von der Matrikelnummer des Repositories oder des Gruppennames reicht)

```
ep2_ex_test submission --group <group> --ue <ue>
```

Kommando ist interaktiv und kann mit `--idea` und `--grade` als ultimative Wollmilchsau verwendet werden.

Alternativ kann die Bewertung über ein separates Kommando oder ein externes Tool eingegeben werden:

```
ep2_ex_test grade --group <group> --ue <ue> --points <points> --repo-owner <working repo owner>
```

Abschließend Issues erzeugen:

```
ep2_ex_test submit --group <group> --ue <ue>
```

Übungstests `ep2_ex_test` - CSV

Spalte	Gültige Werte	Beschreibung
`id`	Matrikelnummer	Matrikelnummer
`role`	`o` = owner/ `e` = editor/ `t` = third	Rolle bei der Bearbeitung
`points`	`_`/Zahl	Bewertung
`team`	Teambezeichnung	Team
`remarks`	Text	Infos für die ÜGL
`feedback`	Text	Feedback für Studis

Weitere hilfreiche Tools

Liste aller noch zu bewertenden Projekte:

```
ep2_eval/ep2_ex_test list ungraded --group <group> --ue <ue>
```

Zusätzliche tags hinzufügen (wann auch immer das gebraucht wird...)

```
ep2_util tag --group <group> --tag <tag>
```

Geplante Funktionen (zumindest von mir)

- Erzeugen von Breakout Listen
- Eingabe von Teams für Übungstests über Teamnummer
- ... (was auch immer ihr braucht)