

ELO Business Solutions command line interface

Das ELO Business Solutions CLI hilft beim erstellen und verwalten von unterschiedlichen Paketen der ELO Business Solutions.

Installation empfohlener GIT-Tools

Git Bash:

<https://git-scm.com/>

SourceTree:

Komfortable GIT Oberfläche zur Verwaltung von unterschiedlichen Projekten.

<https://www.sourcetreeapp.com/>

TortoiseGIT:

Integration von GIT in den Windows Explorer. Zeigt in der Exploreransicht an, welche Dateien sind geändert haben.

<https://tortoisegit.org/>

Ersteinrichtung GIT

SSH-Key erstellen und in GitLab hinterlegen.

```
ssh-keygen -t rsa -C "e.stach@elo.com"  
notepad C:\Users\Stach\.ssh\id_rsa.pub
```

Diesen Key anschließend in den gitlab Einstellungen hinterlegen.

Eigenen User konfigurieren.

```
git config --global user.name "Erik Stach"  
git config --global user.email e.stach@elo.com
```

Einrichtung für ELO Business Solutions

ELO Business Solutions Dateien müssen teils mit CRLF und LF Zeilenumbrüchen eingecheckt werden. Es ist daher essentiell die automatische CRLF Konvertierung auszuschalten.

```
git config --global core.autocrlf false
```

ELO erlaubt zudem eine tiefere Verzeichnisstruktur als Windows im Standard. In GIT müssen daher global lange Pfade zugelassen werden.

```
git config --system core.longpaths true
```

Damit Änderungen an der Groß-/Kleinschreibung von Git berücksichtigt werden, muss folgender Befehl ausgeführt werden.

```
git config --global core.ignorecase false
```

Es kann passieren, dass git commits in UTC angezeigt werden anstatt mit der lokalen Zeit. Dieser Befehl setzt die Anzeige global auf lokale Zeit.

```
git config --global log.date local
```

Verzeichnisstruktur anlegen

ELO Business Solutions Projekte erwarten einen gewissen Verzeichnis Aufbau, sodass Abhängigkeiten korrekt aufgelöst werden können.

```
C:\
  git
    solutions
      common.git
      contract.git
    dev
      elo-cli.git
```

Bevor Projekte geclost werden, müssen daher die Verzeichnisse C:\git\dev und c:\git\solutions angelegt werden.

ELO Command Line Interface vorbereiten

Das Projekt elo-cli stellt grundlegende Operationen für das Verwalten von Docker-Containern und Business Solutions bereit. Dieses Projekt muss nach C:\git\dev\elo-cli.git\ geklost werden.

```
git clone git@srvpGit01vm.elo.local:bs/dev/elo-cli.git elo-cli.git
```

Benötigt wird PowerShell mindest in der Version 4. Das betrifft Anwender unter Windows 7 und Windows 8.

<https://www.microsoft.com/de-de/download/details.aspx?id=40855>

Anschließend muss das ELO CLI Projekt in die Windows-Path-Variable aufgenommen werden. Somit steht der Befehl ELO überall zur Verfügung.

C:\Git\dev\elo-cli.git in die Path-Umgebungsvariablen eintragen

Benötigt wird zudem die Angabe des JDKs in den Umgebungsvariablen von Windows

```
JDK_HOME      C:\Program Files\Java\jdk1.8.0_121
```

Damit Powershell Skripte wie auch der "elo"-Befehl in der Powershell gestartet werden können, muss einmalig folgender Befehl in der Powershell ausgeführt werden:

```
Set-ExecutionPolicy RemoteSigned -Scope CurrentUser
```

Erste Solution klonen

Business Solutions benötigen ein Common Paket. Die Pfad-Struktur sollte der obigen entsprechen. Es ist darauf zu achten, dass GIT-Projekte ein ".git" im Namen enthalten.

```
git clone git@srvpGit01vm.elo.local:bs/common.git common.git
git clone git@srvpGit01vm.elo.local:bs/contract.git contract.git
```

Getting started

Eine Neue Entwicklungsumgebung kann über "elo prepare" gestartet und vorebereitet werden.

```
elo prepare -stack mosbachn-contract-1-03-000 -workspace de
```

Der Stackname ist der Name der Docker-Umgebung, Workspaces definieren Abhängigkeiten zu anderen Paketen sowie welche Pakete installiert werden sollen.

Anschließend können Änderungen über "elo pull" zurückgespielt werden.

```
elo pull -workspace de
```

Wichtige Hinweise

Ein Pull der Daten aus ELO schreibt alle Pakete neu. GIT ist dabei in der Lage Änderungen zu erkennen. Dadurch ist gewährleistet, dass lediglich der Diff in GIT committed wird.

Damit das funktioniert muss ein Pull von ELO auf dem Commit ausgeführt werden, der verwendet wurde um die Entwicklungsumgebung vorzubereiten. Andernfalls werden eventuell Änderungen von Kollegen überschrieben.

Zudem ist darauf zu achten, dass der common-Stand stets mit der Lösung übereinstimmt. Ist bspw. common 1.02 ausgecheckt und es wird versucht contract 1.03 zu bauen, führt das zu Fehlern.