Betriebsdokumentation: Antragsverwaltungstool (I3)

Inhaltsverzeichnis

Ziel der Dokumentation

Ziel der Dokumentation ist es den Administrator beim Einrichten, Konifgurieren, Innbetriebnhemen sowie Betreuen des Systems Rahmenbedingungen zu geben.

Anforderungen an die Produktivumgebung

Systemvorraussetzungen

Zur Ausführung des Systems, in Hinblick auf Skalierbarkeit, wird benötigt:

Hardware

1. Für den Server:
   a. CPU mit 4 Kernen
   b. 4 GB Arbeitsspeicher
   c. 15 GB Festplattenspeicher
   d. Internetanbindung mit mindestens 50 Mbit/s

Software

1. Für den Server:
    a. Ubuntu (Version 20.04 LTS) oder Debian (Version 10)
    b. Python3 bzw. Pip3
    c. *mysqlclient*
    d. mindestens Django (Version 3.2.4 LTS)
    e. Apache2
    f. Ubuntu (Version 20.04 LTS) oder Debian (Version 10)
    g. MySQL Community Edition (mindestens Version 8.0)

Systemeinrichtung

Komponenten

Die Anwednung besteht im ganzen aus einer Django-Applikation welche sich in kleine Unterkomponenten gliedert (im Folgenden aufgelistet) . HTML: Alle Seiten die die Applikation bereitstellt sind im Ordner *Templates* zu finden. . CSS: Die CSS-Dateien bilden das Styling für die HTML-Seiten und sind im Applikationsordner *Antragsverwaltungstool* unter *static/css* zu finden. . PNG: In der Applikation befinden sich ebenfalls einzelne Bilder zum darstellen auf der Webseite, ebenfalls unter *static/png* . JS: Es wird JavaScript benutzt um input Felder anzuzeigen oder zu verstecken. Zusätzlich um Texte einfach zu displayen. . PY: Die Anwendung besteht zum Großteil aus Python-Dateien welche einerseits das MVC-Architekturprinzip wiederspiegeln (*views.py*, *models.py*) und andererseits die Funktionen des Frameworks bereitstellen. Wichtig ist auch hier die *settings.py* indem für die Applikation spezifische Einstellungen festgelegt sind. . MySQL-API-TReiber: Zum Betreiben wird ebenfalls ein Treiber zur Kommunikation mit der Datenbank benötigt. Hier wird *mysqlclient* empfohlen.

Installation / Verzeichnisstruktur

Für die Installation ist nur Python und Django nötig sowie der Apache Webserver. Es kann entweder ein neues Django Projekt mit dem gleichen Namen erstellt werden, um dann die Dateien aus der Applikationen in das neue Projekt zu kopieren. Oder es wird der gesamte Ordner aus der fertigen Applikation kopiert. Hier ist es nötig die Python-Laufzeitumgebung entsprechend anzupassen, da in unserem Falle mit einem Python Virtual Environment gearbeitet wird. Die Verzeichnisstruktur wird beim Erstellen des Projektes automatisch erzeugt, Hier ist es nur notwendig die HTML-Unterordner in *Templates* und den static-Ordner in den Root-Folder der Applikation zu verschieben. Auf dem Webserver wurden die Dateien im root Verzeichnis unter ~/Antragsverwaltungstool abgelegt.

Installation und Konfiguration von Django und Apache mit wsgi

Um unsere Server passent einzurichten und zu Konfigutieren sind wir diesem Guide gefolgt. https://www.digitalocean.com/community/tutorials/how-to-serve-django-applications-with-apache-and-mod_wsgi-on-ubuntu-14-04 nachdem wir unser Projekt angelegt hatten haben wir die Erstellten dateien des neuen Projektes mit unseren ausgetauscht und danach nochmals die Befehle

```
python3 ./manage.py makemigrations
python3 ./manage.py migrate
```

Konfiguration

Zur Konfiguration muss einerseits im der *settings.py* die IP-Adresse des Datenbankservers angegeben werden mit einer zusätzlichen Angabe des MySQL-Treibers. Da der Server auf der gleichen Maschine läuft, wurde *localhost* verwendet. Dafür muss in der Datenbank nur ein Nutzer für die Applikation angelegt werden. Damit die Applikation unter Apache2 lauffähig ist muss *WSGI* verwendet werden. Dazu benötigt der Webserver Berechtigungen um die Dateien der Applikation ausführen und lesen zu können.

Systembetreuung

Datensicherung und -wiederherstellung

Die Daten des Webservers können nach Notwendigkeit regelmäßig gesichert werden. Es ist systemseitige keine automatische Sicherung der Daten vorhanden.

Erreichbarkeit des Servers (Portfreigabe)

Zur Kommunikation mit dem Webserver muss der Port 8000 über die Firewall geöffnet werden. Das geschieht mit dem Kommando "sudo ufw allow 8000". Dies ist leider eine "Bad Practice" da der Traffic eigentlich auch über Port 80 oder den http/tcp Port laufen könnte. Leider sind wir noch zu keiner Möglichkeit gekommen dies so zu realisieren. Selbige Umsetzung ist aus Sicherheitsgründen jedoch anzustreben. Der Port kann über "sudo ufw disallow 8000" wieder geschlossen werden.

Neustart des Servers

Der Django Server ist in einer Screen session im Hintergrund und wird bei einem Reboot des Haubtsrevers **Nicht** Automatisch gestartet. Der Start muss manuell ausgelöst werden. Sollte man den Server Starten wollen muss man sich über ssh auf den Server einlogen und unter dem Benutzer "django_admin" folgende Befehle eingeben:

```
cd Antragsverwaltungstool
screen ./Serverstart.sh
```

Danach gelangt man in eine screen session. Um aus dieser Herauszukommen um den Prozess in den Hintergrund zu bringen muss man die Tastenkombination **STRG+A** danach **D** drücken dann kommt man zurück zu der Linux konsole und kann sich beispielsweise mit "exit" abmelden.

Um den Serverprozess wieder in den Vordergrund zu bekommen benutzt man folgende befehle:

```
screen -ls
screen -r pts-2.antragsverwaltung
```

Sollte sich der Name der screen session ändern genügt es auch screen -r **TAB** zu drücken da der server gerade die einzige screen session sein sollte

Verwalten des Systems als Administrator

Zugang und Rechte

Der Administrator hat das alleinige Recht das Tool zu verwalten. Dieser besitzt eine seperaten Account mit allumfänglichen Rechten auf der Datenbank. Die Adminseite ist unter "141.56.51.110:8000/admin" zu erreichen. Hier können Datenbank Einträge angezeigt und bearbeitet bzw. gelöscht oder hinzugefügt werden. Dies sollte jedoch nur in Außnahemfällen genutzt werden.

Sollte hier eine Antragsart nicht aufgelistet sein, sollte in der Datei *admin.py* überprüft werden ob das zugehörige Model auch registriert ist. Die Anzeige der jeweiligen Datensätze der Antragsarten hängt von der *str* Methode im jeweiligen Model ab. Diese sollte **unbedingt** bei Erstellung eines Models auf Korrektheit überprüft werden.

Der Admin kann ebenfalls neue Benutzer und Benutzergruppen einrichten. Hier ist es zu empfehlen den entsprechenden Accounts nicht vollumfängliche Rechte einzuräumen sondern nur Lese- bzw. Schreibrechte zu erteilen, damit keine Manipulation der Daten Auftritt.