h_bdl

HAUSCHILD BÜRODIENSTLEISTUNGEN
UNTERNEHMENSGESELLSCHAFT (HAFTUNGSBESCHRÄNKT)

Archivierung durch QR-Codes

Dokumentation

12.01.2020,Julia Hauschild

1. KURZBESCHREIBUNG 4

2. ÄNDERUNGEN 4

3. BESCHREIBUNG DER ANWENDUNG 5

3.1. Upload von Dokumenten 5

3.2. Lesen der QR-Codes 5

3.3. Erzeugen neuer Dokumente im PDF-Format 5

3.4. Erstellen von Metadaten 5

3.5. Umwandlung aller Daten und Dokumente in ein XML-File 5

4. TECHNISCHE BESCHREIBUNG 6

4.1. Architektur 6

4.2. Plattform 6

4.3. Systemkomponenten 6

4.4. Schnittstellen 6

5. INSTALLATION UND KONFIGURATION 8

5.1. Installation Grundsystem 8

5.1.1 Voraussetzungen 8

5.1.2 Webserver 8

5.2. Konfiguration Mapping 8

6. SYSTEM-/ANWENDUNGSBETRIEB 9

6.1. Logging 9

6.2. Löschkonzept 9

6.3. Archivierung 9

6.4. Update der Systemmodule 9

7. SECURITY 9

7.1. Zugriffsberechtigung 9

7.2. Ports 9

8. STÖRUNGSMANAGEMENT 9

8.1. Fehlermanagement 9

8.2. Restart und Wiederanlauf 10

8.3. Alarmierung Eskalation 10

1. Kurzbeschreibung

Die Aufgabe der Software ist es Archivierungsinformationen auf Basis eines QR-Codes zu lesen und diese als Metadaten für die spätere Archivierung mit archivierfähigen Dokumenten in ein festes Verzeichnis abgelegt.

2. Änderungen

Änderungshistorie				
Version	Status	Datum	Bearbeiter	Änderung/Bemerkung
0.5	Stable (POC)	12.01.2020	Julia Hauschild	Die Testfunktionalität Download eingefügt

3. Beschreibung der Anwendung

3.1. Upload von Dokumenten

Die Dokumente können in der Startseite des Services hochgeladen werden. Es können Dateien in den Formaten JPEG, TIFF und PDF weiterverarbeitet. Andere Dateien werden durch eine Fehlermeldung gekennzeichnet

3.2. Lesen der QR-Codes

Auf dem hochgeladenen Dokument wird nach einem QR-Code gesucht. Falls ein QR-Code gefunden wird, wird dieser automatisch ausgelesen. Falls kein Code, mehrere Codes oder fehlerhafte Codes gefunden werden, werden entsprechende Fehlermeldungen ausgegeben.

3.3. Erzeugen neuer Dokumente im PDF-Format

Alle hochgeladenen Dokumente werden nach Seiten geordnet und evtl. zusammengefasst und in ein PDF konvertiert.

3.4. Erstellen von Metadaten

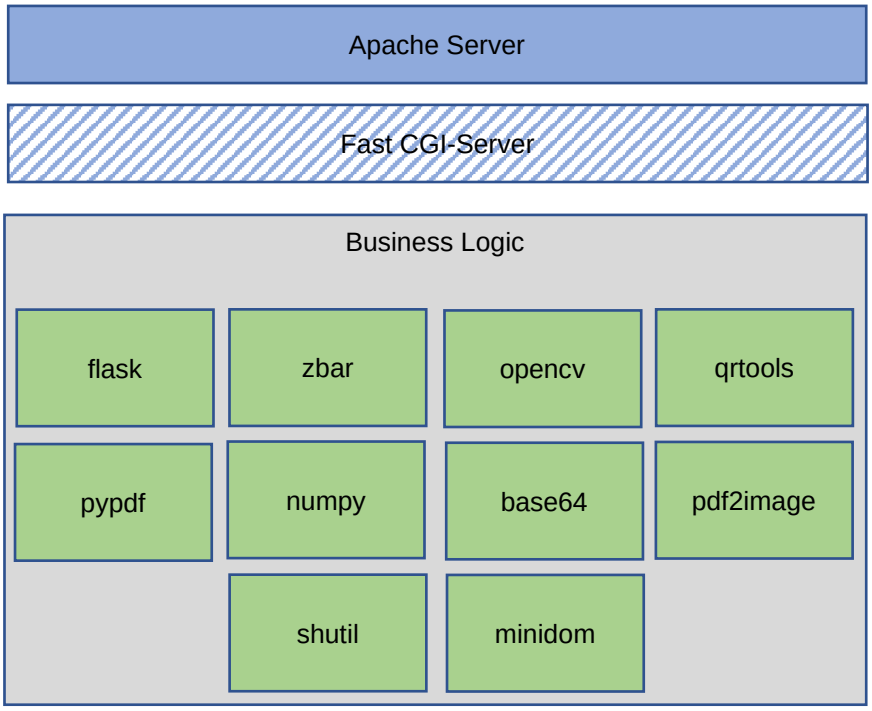
Die Informationen des QR-Codes werden interpretiert und als Metadaten für die weitere Archivierung in einem XML-File zur Verfügung gestellt.

3.5. Umwandlung aller Daten und Dokumente in ein XML-File

Das PDF wird durch base64 in das XML-File integriert, sodass die gesamte Datenlieferung im XML-Format zurückgegeben wird.

4. Technische Beschreibung

4.1. Architektur



4.2. Plattform

Als Plattform kann jede Linux Distribution verwendet werden. Die Software kommt als One-File Installation um alle notwendige Module und Bibliotheken zu installieren. Die Dimensionierung der Plattform ist abhängig von der Nutzung und Anzahl der parallelen Sessions. Dabei kann die Performance sowohl vertikal als auch horizontal skaliert werden.

4.3. Systemkomponenten

Siehe Architekturbild.

4.4. Schnittstellen

Der Aufruf kann über http bzw. über https erfolgen.

Die Ausgabe des XML-Files erfolgt auf einem festen Verzeichnis. Für den Test kann auch die Datei nach erfolgreichen Verarbeitung heruntergeladen werden. Hierzu wird in der Ergebnisseite der Herunterladen-Button angezeigt.

Anbei ein Beispiel der XML-Struktur.

```xml
<?xml version="1.0" ?>
<upload uploadID="1">
        <document>
                <file>
                        <mime>application/pdf</mime>
                        <data>JVBERi0xLjMKMSAwIG9iago8PAovS2lkcyBbIDMgMCBSIF0
                </file>
                <metadata>
                        <offernumber>BK300-682266-00E</offernumber>
                        <regnr>07</regnr>
                        <doctypenr>31</doctypenr>
                        <docnamenr>12</docnamenr>
                        <register>Vertrag</register>
                        <doctype>Zusatzvereinbarungen</doctype>
                        <documentname>Vertragsdokument</documentname>
                </metadata>
        </document>
        <document>
                <file>
                        <mime>application/pdf</mime>
                        <data>JVBERi0xLjMKMSAwIG9iago8PAovS2lkcyBbIDMgMCBSIF0
                </file>
                <metadata>
                        <offernumber>BK300-682266-00E</offernumber>
                        <regnr>07</regnr>
                        <doctypenr>31</doctypenr>
                        <docnamenr>12</docnamenr>
                        <register>Vertrag</register>
                        <doctype>Zusatzvereinbarungen</doctype>
                        <documentname>Vertragsdokument</documentname>
                </metadata>
        </document>
</upload>
```

5. Installation und Konfiguration

5.1. Installation Grundsystem

5.1.1 Voraussetzungen

OpenCV
In der Anwendung wird OpenCV zur Erkennung von Mustern eingesetzt. Dieses ist vorab auf dem Server zu installieren, da einige verwendete Bibliotheken mit OpenCV arbeiten.

5.1.2 Webserver

Für das produktive Umfeld wird ein Apache Webserver benötigt. Dieser ist für den Einsatz mit Flask speziell zu konfigurieren. Siehe Anlage Konfigurationsanleitung Apache Server. Für den Testbetrieb stellt Flask einen Webserver zur Verfügung, welcher für den nicht-produktiven Testbetrieb verwendet werden kann.

5.2. Konfiguration Mapping

Es werden drei csv Dateien benötigt (Dokumentenname, Registernummer und Dokumententyp). Mithilfe dieser drei csv-Dateien wird der Code aus dem QR-Code entschlüsselt und den einzelnen Rubriken zugeordnet. Bei jedem Neustart werden diese geladen, dadurch sind sie beliebig erweiterbar und anpassbar.

6. System-/Anwendungsbetrieb

6.1. Logging

Das Logging der Anwendung erfolgt filegebunden im Pfad /app/logs. Hier werden Infos, Warnings und Error dargestellt. Für den produktiven Betrieb ist diese auf eine Share oder auf einen Server umzuleiten.

6.2. Löschkonzept

Die Prozesse der Anwendung sind statless angelegt. Es werden ausschließlich temporäre Daten über den Verarbeitungsprozess gehalten werden und nach erfolgreichen Abschluss der Session gelöscht. Dadurch wird kein weiteres Löschkonzept benötigt.

6.3. Archivierung

Zum archivieren sind ausschließlich die Logging- und XML-Daten erforderlich.

6.4. Update der Systemmodule

Folgende Module sind aktuell enthalten:

- flask
- pathlib
- pdf2image
- numpy
- qrtools
- shutil
- img2pdf
- PyPDF2
- os
- base64
- minidom
- datetime

7. Security

7.1. Zugriffsberechtigung

Für den produktiven Betrieb können technische Benutzer eingerichtet werden

7.2. Ports

Für http ist der Port 80, für https der Port 443 vorgesehen.

8. Störungsmanagement

8.1. Fehlermanagement

Für den produktiven Einsatz wird ein Wartungsvertrag empfohlen. Hier können separat alle wichtigen Details festgelegt werden.
Soll das Fehlermanagement intern erfolgen kann der mitgelieferte Source-code selbst oder durch HBDL customized, um auf die jeweiligen Anforderungen angepasst zu werden.

8.2. Restart und Wiederanlauf

Bei einem Restart gehen die Daten einer offenen Session verloren.
Beim Wiederanlauf werden alle vorhandenen temporäre Daten bereinigt und neue Konfigurationsdaten geladen.
Somit sind keine zusätzlichen Aktionen erforderlich

8.3. Alarmierung Eskalation

Die Alarmierung kann über die Logs mittels Monitoringsoftware erfolgen. Eine Eskalation kann auch automatisiert, bei einem vorhandenen Wartungsvertrag, an den Support geliefert werden.