



PDF-Downloader

Opgavebeskrivelse

I denne uge skal I udvikle et værktøj til at downloade PDF-filer. Opgaven er baseret på en case fra en kunde, hvor krav og behov ikke nødvendigvis er fuldt specificerede fra starten. Det er derfor vigtigt, at I læser case-beskrivelsen grundigt, og forholder jer kritisk til kundens krav.

Casen afspejler en typisk situation, hvor en kunde har et konkret behov, men ikke nødvendigvis har overblik over, hvordan den tekniske løsning bedst udformes, eller hvilke krav der bør stilles til robusthed, vedligeholdelse og fremtidig brug. Det er jeres opgave at omsætte disse behov til en velstruktureret og holdbar løsning.

OBS:

For ikke at overbelaste Specialisternes netværk - Start med at lave en prototype som kun downloader maks. 10 PDF'er ad gangen. Når I arbejder hjemme, kan I forsøge med det fulde antal PDF'er og fuld concurrency.

Casebeskrivelse

I din rolle som IT-konsulent hos Specialisterne, har du modtaget en opgave fra en kunde, der har et ældre Python-script, som ikke længere fungerer korrekt. Kunden har brug for et pålideligt og effektivt program, til at downloade PDF-rapporter fra en liste med dynamiske URL'er.

Case fra kunden:

"Kære Konsulent,

Som aftalt har du hermed listen (inkl. metadata) angående de rapporter vi gerne vil have downloaded ("GRI_2017_2020").



Det er adressen i kolonne AL (Pdf_URL) vi har forsøgt at downloade, men som jeg forklarede ville det være fedt, at hvis dette link ikke virker, så prøvede programmet linket i kolonne AM. Hvis første link virker, behøver den ikke prøve AM.

Jeg har også vedhæftet vores Python-program, som vi har brugt til at downloade med. Som sagt kører koden, men den er meget ustabil, og det går til tider meget langsomt.

Desuden er der vedhæftet en Excel-fil med rapporter fra 2006-2016 ("Metadata2006-2016"), hvor man i kolonne AT kan se, om en rapport er blevet downloadet eller ej (som vi også gerne vil have information om med de nye rapporter). Vi behøver ikke en specificering af, hvorfor rapporten ikke er hentet, blot en variabel der f.eks. antager værdien "Downloadet" eller "Ikke downloadet".

Kravsdefinitionen for dette projekt er, at vi ønsker, at I laver et program, der effektivt kan downloade alle de rapporter, der har et virkende link. Programmet skal downloade disse PDF-rapporter og dele dem med os via NAS (eller anden måde). De downloadede rapporter skal navngives efter kolonnen "BRNummer". Derudover ønsker vi en liste over, hvilke rapporter (fra GRI_2017_2020), der er blevet downloadet, og hvilke der ikke er. Den endelige kode afleveres ligeledes til os, da der formegentligt kommer flere rapporter, der skal hentes senere.

Bliver koden lavet i Python, eller er det ikke det rigtige program til denne type opgave?"

Opfølgning på kundecase

Du og din nærmeste leder gennemgår kundecasen og vurderer, at den nuværende løsning mangler tilstrækkelig fejlhåndtering og er svær at genbruge i andre sammenhænge. Kunden har ikke selv stor erfaring med programmering og har derfor haft svært ved at formulere præcise tekniske krav.

Efter yderligere dialog med din leder beslutter I jer for at vende retur til kunden. Her presserer I signifikansen i at udvikle en mere robust løsning der er mere effektivt, bedre struktureret ift. fejlbehandling og nemmere at vedligeholde. Der er behov for at håndtere alternative URL'er,



hvis det første link ikke virker, og at registrere downloadstatus for hver rapport. Det er ikke angivet, hvordan linkene eller hvor de downloadede filer skal lagres. Du skal derfor selv finde en passende løsning.

Din opgave er at udvikle en brugbar løsning til kunden med udgangspunkt i casen samt opfølgningen på casen. Det indebærer, at du selv tager stilling til, hvordan kravene bedst omsættes til en robust og gennemtænkt løsning. Det er også relevant at overveje, hvordan PDF-rapporterne lagres på en hensigtsmæssig måde og *GRI_2017_2020* metadaten er kun til inspiration. Kodesproget til opgaven er også valgfrit!

GitHub

Fordi det er en tidsbegrænset og ret åben opgave, aftaler din leder og dig at en af dine kollegaer skal udføre et code review på projektet der skal garantere implementeringen af tests og kvalitet ift. projektstruktur & separation of concerns. For at simulere denne øvelse skal du i slutningen af ugen overlevere dit projekt til en anden kursist der skal kvalitetssikre dit projekt alt imens at du gør det samme på deres eller en anden kursists PDF-downloader.

Derfor skal I skal uploade jeres projekt og kodebase til GitHub. Det er vigtigt, at jeres projekt er let at forstå for andre. Sørg for en tydelig README med vejledning til opsætning og kørsel af koden, samt relevante kommentarer i koden.

- Link til et Github-repository, der indeholder følgende:
 - Kildekode med kommentarer.
 - Kravspecifikation (brug skabelonen, der er uploadet på Teams).
 - UML klassediagram eller et UML sekvensdiagram, der viser systemets opbygning.
 - En README.md fil, der beskriver hvordan man kører jeres kode.
-



Opmærksomhedspunkter

- Det er vigtigt at have fokus på “separations of concerns”, SOLID og kommentar/dokumentation.
- Løsningen skal designes med henblik på stabilitet, skalerbarhed, testbarhed og fremtidig anvendelse – ikke blot som et engangsscript. Husk på de refleksioner og erfaringer du gjorde dig med unit-tests da du udviklede dit spil.
- For ikke at overbelaste Specialisternes netværk - Start med at lave en prototype som kun downloader maks. 10 PDF'er ad gangen. Når I arbejder hjemme, kan I forsøge med det fulde antal PDF'er og fuld concurrency.

Pensum og Ressourcer

- [Filhåndtering](#)

God arbejdslyst!