PDF-Downloader

Opgavebeskrivelse

I denne uge skal I arbejde med fejlhåndtering, exception-handling og multithreading ved at bygge en PDF-downloader. Opgaven er baseret på en case fra en kunde, som har behov for et pålideligt og effektivt program til at downloade PDF-rapporter fra en liste med dynamiske URL'er. Der er behov for at håndtere alternative URL'er, hvor det første link ikke virker, og at registrere downloadstatus for hver rapport. Læs casebeskrivelsen længere nede grundigt, inden du går i gang.

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.

Afleveringsformat

I slutningen af ugen skal i lave en præsentation, der gennemgår en demo af jeres løsning, nedslag i forskellig dele af jeres kodebase. Det kan være en god ide at komme omkring styrker og svagheder ved ens løsning.

I skal dog også uploade jeres projekt og kodebase til Github da vi skal arbejde videre på opgaven i næste uge.

- 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.

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. Der er behov for at håndtere alternative URL'er, såfremt det første link ikke virker, og at registrere downloadstatus for hver rapport.

Beskrivelse 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".

Kravspecifikationen 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?"

Opgaver og opmærksomhedspunkter

- Det er vigtigt at have fokus på "separations of concerns", SOLID og kommentar/dokumentation, da dit færdige projekt skal afleveres
- 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

- Multithreading
- Filhåndtering

God arbejdslyst!