

Software Test Rapport – Unit & Integration Testing

1. Generel Information

Projekt / Systemnavn: PDF Downloader

Testansvarlig: Robert Pallesen

Versionsnummer / Build ID: SHA Commit 8403c20

2. Testformål

Validere at URL-håndtering, HTTP-fejl, PDF-verifikation, fil-I/O, logging og re-run-filtrering opfører sig som specificeret. Bekræfte at både sekventiel og concurrent eksekvering gemmer korrekte status-tupler i JSON-loggen og kun skriver gyldige PDF-filer.

3. Testresumé

Testtype	Antal tests	Bestået	Fejlet	Kommentar
Unit tests	19	19	0	<i>verify_pdf</i> og <i>download_pdf_file</i> inkl. 200, 400, 403, 404→200, 408, 415, 500, 503, I/O-fejl
Integration tests	7	7	0	Lokal HTTP-server, Excel I/O, JSON-form, re-run skip, fallback-logik
Samlet resultat	26	26	0	Alle tests bestod

4. Testet Funktionalitet

Klasse / Modul	Metoder / Funktioner	Formål med testen	Resultat
<i>download_files</i>	<code>verify_pdf(bytes None)</code>	Detekterer PDF via magic bytes	Bestået
<i>download_files</i>	<code>download_pdf_file(row_id, urls, cfg)</code>	Henter m. timeout og headers, mapper koder, gem kun gyldig PDF, håndter I/O-fejl	Bestået
<i>download_files</i>	<code>filter_data(df, data_cfg, batch_size)</code>	Filtrér allerede loggede rækker, bevar indeks	Bestået
<i>download_files</i>	<code>extract_urls(df, data_cfg)</code>	Trim og prioriter kandidat-URL'er pr. ID	Bestået
<i>download_files</i>	<code>main_sequential(data_cfg, dl_cfg)</code>	End-to-end, deterministisk rækkefølge, korrekt JSON-log	Bestået
<i>download_files</i>	<code>main_concurrent(data_cfg, dl_cfg)</code>	Samme semantik som sekventiel i multi-tråd	Bestået

5. Testtilgang og Miljø

- Automatiserede tests med *pytest*
- Unit tests bruger *responses* til at stubbe HTTP-svar samt monkeypatch for undtagelser og I/O-fejl
- Integrationstests starter en lokal *ThreadingHTTPServer*, der serverer en temp-mappe med fixtures: *valid.pdf*, *notpdf.txt*, simuleret 403 og sleep-baseret timeout. Excel input skrives med pandas + openpyxl

Komponent	Version / Setup
Windows	11
Python	3.13.8
Packages	<i>pytest, responses, requests, pandas & openpyxl</i>
IDE	Visual Studio Code

6. Testresultater

Test ID	Testnavn / Enhed	Input	Forventet Output	Faktisk Output	Resultat	Kommentar
UT-001	verify_pdf()	gyldig PDF-header	Returnerer True	Returnerer True	Bestået	Magic bytes %PDFdetekteret
UT-002	verify_pdf()	tom payload	Returnerer False	Returnerer False	Bestået	None/empty håndteres
UT-003	download_pdf_file()	primær 404, sekundær 200	Fil gemmes, kode 200, brugt sekundær URL	Fil gemmes, kode 200, brugt sekundær URL	Bestået	Fallback virker
UT-004	download_pdf_file()	ikke-PDF payload	Ingen fil, kode 415	Ingen fil, kode 415	Bestået	Indholdsvalidering
UT-005	download_pdf_file()	timeout	Ingen fil, kode 408/500	Ingen fil, kode 408/500	Bestået	Mapping dokumenteret
IT-001	End-to-end: main_concurrent	Excel med 3 rækker, 2 gyldige	2 filer i downloads/ korrekt log	2 filer i downloads/ korrekt log	Bestået	Lokalt HTTP-miljø
IT-002	Re-run filter	1 ID gentages i 2. kørsel	Gentaget ID springes over	Gentaget ID springes over	Bestået	Matcher log
IT-003	JSON log form	Efter kørsel	{id: [ok, code, url]}	{id: [ok, code, url]}	Bestået	Struktur verificeret

7. Fejl og Observationer

Fejl-ID	Beskrivelse	Sværhedsgrad	Forslag til løsning	Status
BUG-01	Timeout-mapping kan variere pr. requests-version (408 vs 500). Tests accepterer begge pt.	Høj	Overvej eksplicit Content-Type-tjek ud over magic bytes hvis upstream servere fejlagtigt rapporterer typer	Åben

8. Opsummering og Konklusion

Downloaderen opfylder de nuværende krav. Fejlhåndtering er forudsigelig. Logging og re-run-filtrering virker. Sekventiel og concurrent er på linje. Klar til CI.

9. Refleksion og Videre Arbejde

- Property-baserede tests for URL-normalisering og header-variationer.
 - Checksum-verificering til deduplikering af identiske PDF'er.
 - Maks filstørrelse og rate limiting i integrationstests.
 - Tilføj tests til CI med upload af artefakter (logs og sample downloads).
-