

MSDB - Testdatenverarbeitung

Fachrichtung

Informatik HF
Schwerpunkt
Softwareentwicklung

Auftraggeber

SmartDrives AG
Christian Staudenmann

Experte/-in & Dozent/-in

Jean-Jacques Pittet
&
Alain Rohr



```
jobState.setMotorId(motorId);
jobState.setTestType(testType);
jobState.setAnalysisParameters(analysisParameters); // Parameter speichern
jobRepository.save(jobState);

String dataTypeForStorage = testType.toUpperCase();
DataRequest dataReq = new DataRequest();
dataReq.setRequestId(jobId);
dataReq.setTestTypeForData(dataTypeForStorage);
dataReq.setMotorId(motorId);
storageGateway.requestDataFromStorage(dataReq);

return jobState;

@Override
public void processDataResponseFromStorage(DataResponse dataResponse) {
    if (dataResponse == null || dataResponse.getRequestId() == null) {
        LOG.warn("Ungültige DataResponse empfangen (null oder ohne RequestId).");
        return;
    }
    String jobId = dataResponse.getRequestId();
    Optional<AnalyseResult> jobStateOpt = jobRepository.findById(jobId);

    if (jobStateOpt.isEmpty()) {
        LOG.warnf("DataResponse für unbekannte Job-ID %s empfangen. Ignoriere.", jobId);
        return;
    }
    AnalyseResult jobState = jobStateOpt.get();

    if (!dataResponse.isSuccess()) {
        String errorMsg = "StorageService: " + dataResponse.getErrorMessage();
        LOG.errorf("Datenabruf fehlgeschlagen für Job-ID: %s. Fehler: %s", jobId, errorMsg);
        jobRepository.updateJobStatus(jobId, "FAILED_DATA_FETCH", errorMsg);
        notifyBatchCoordinatorIfApplicable(jobId);
        return;
    }
    String errorMsg = "StorageService für Job-ID %s empfangen. Bearbeite Job für DataService neu.", jobId;
    LOG.errorf(errorMsg);
    jobRepository.updateJobStatus(jobId, "DATA_RECEIVED_PROCESSING", null);
    notifyBatchCoordinatorIfApplicable(jobId);

    if (dataResponse.isSuccess()) {
        String errorMsg = "StorageService: " + dataResponse.getErrorMessage();
        LOG.errorf("Datenabruf fehlgeschlagen für Job-ID: %s. Fehler: %s", jobId, errorMsg);
        jobRepository.updateJobStatus(jobId, "FAILED_DATA_FETCH", errorMsg);
        notifyBatchCoordinatorIfApplicable(jobId);
        return;
    }
    AnalyseResult jobState = jobStateOpt.get();
    jobRepository.updateJobStatus(jobId, "DATA_RECEIVED_PROCESSING", null);
    notifyBatchCoordinatorIfApplicable(jobId);
    return;
}
```

Ausgangslage & Ziel

Bisher wurden Motor-Testdaten von Internen Messtool als JSON-Dateien unstrukturiert auf Netzlaufwerken gespeichert. Das auswerten war fehleranfällig, mühsam und schwer zu pflegen. Ziel war die Entwicklung eines zentralen Systems, das die Daten automatisch importiert, prüft, speichert und einfacher nutzbar macht.

Ergebnis & Nutzen

Das neue System verarbeitet die Testdaten und soll automatisch, zentral und zuverlässig die Daten verwalten. Dadurch wird die Arbeit schneller, einfacher und transparenter. Fehlerquellen werden reduziert, Auswertungen vereinfacht und die Grundlage für zukünftige Erweiterungen wurde geebnet.