

Documentation

Das Hauptziel unseres Projekts ist die Entwicklung einer Anwendung, die den gegebenen großen Satz von Daten/XML-Dateien in die Datenbank/Neo4J konvertiert.

Dabei erhalten wir 6 wichtige Datenstrukturen, nämlich Klassen in der OOP (objektorientierte Programmierung). Jeder dieser Baupläne hat eine eindeutige Nummer (ID) und weiß, wie man sich selbst für die Datenbank vorbereiten muss.

I. Grundlegende Strukturen & Datenmodell

IEntity: Das IEntity-Interface ist die Standardvereinbarung, die sicherstellt, dass alle Entitäten im System identifiziert, gespeichert, in JSON konvertiert und konsistent angezeigt werden können – was das gesamte System organisierter und zuverlässiger macht.

AbstractEntity: Diese Java-Klasse bietet eine Vorlage, der alle Hauptobjekte im System folgen. Sie stellt die grundlegenden Regeln und gemeinsamen Funktionalitäten bereit, die jede Entität (wie Abgeordnete, Reden, Fraktionen) haben muss. Sie gewährleistet, dass jedes Objekt im System eindeutig identifiziert werden kann. Sie bietet eine hashCode()-Methode, die Sammlungen (wie Maps und Sets) dabei hilft, Entitäten schnell zu speichern und zu finden.

CypherQuery: Dies ist das Paket von Anweisungen oder Befehlen, das an die Datenbank gesendet wird. Dadurch konnten wir die Daten- und Befehlsstruktur trennen.

Plenarprotokoll: Sie enthält alle wesentlichen Informationen darüber, wann und wo die Sitzung stattfand, und dient als Container für alle Reden, die während dieser Sitzung gehalten wurden. Sie exportiert auch alle ihre Daten als JSON zur Verwendung in Websites, APIs oder zum Datenaustausch, was sowohl Basisinformationen als auch Beziehungen (über IDs) beinhaltet.

Datenklassen (Entitäten): Die Klassen sind Kommentar, Redner (Abgeordneter), Rede, Abgeordneter (Redner), Fraktion, Sitzung (Plenarprotokoll). Sie haben ihre eigene Eigenschaften(attributes) und operationen(Methode)

II. Kernsteuerung und Verarbeitung

PortalFactory: Diese Java-Klasse ist wie das Kontrollzentrum aller Operationen im Projekt. Die Erstellung von Objekten im System und die Zuweisung eindeutiger IDs für unsere Datenstrukturen erfolgen in dieser Klasse. Die erstellten Objekte werden in ihren internen Ablageschränken/Map-Sammlungen gespeichert, und diese Klasse bietet Zugriff auf zwei weitere Klassen: XMLProcessor und DatabaseService.

XMLProcessor: Seine Hauptaufgabe ist es, die offiziellen XML-Dateien des Parlaments zu lesen und die Rohdaten in verwendbare Objekte innerhalb des Systems zu konvertieren. Er arbeitet eng mit der PortalFactory zusammen, um alle Objekte zu erstellen. Die XML-Dateien des Parlaments haben eine spezifische, verschachtelte Struktur. Der XMLProcessor weiß genau, wie er diese Struktur navigieren muss, um die richtigen Informationen zu finden. Verschiedene XML-Dateien schreiben Parteinamen möglicherweise

unterschiedlich (z.B. "CDU/CSU", "CDUCSU", nur "CDU"). Der Prozessor verfügt über ein integriertes "Übersetzungswörterbuch" (fraktionMapping), um sicherzustellen, dass alle Variationen in einen standardisierten Namen konvertiert werden.

Neo4jConnection: Sie kümmert sich um die gesamte Kommunikation mit der Datenbank – das Senden von Daten zum Speichern und das Abrufen von Daten bei Bedarf. Sie erstellt und verwaltet eine lokale eingebettete Datenbank (die Datenbank läuft innerhalb Ihrer Anwendung, nicht auf einem separaten Server). Sie erstellt eindeutige Constraints für alle Hauptobjekttypen (Abgeordnete, Reden, Fraktionen usw.).

DatabaseService: Der DatabaseService dient als Datenexporteur oder Speicher-Manager für das Parlamentsportal. Seine Hauptaufgabe ist es, alle vom System erstellten Objekte zu nehmen und sie dauerhaft in der Neo4j-Datenbank zu speichern. Wenn der XMLProcessor Daten importiert, exportiert der DatabaseService sie in die Datenbank. Er sammelt alle erstellten Objekte: Politiker, Politische Gruppen, Sitzungen, Reden und Kommentare. Anstatt ein Objekt nach dem anderen zu speichern (was sehr langsam wäre), verwendet er Stapelverarbeitung: Er gruppiert Speicherbefehle in Bündeln von 1000 Stück. Er sendet jedes Bündel als einzelnes Paket an die Datenbank. Das ist viel schneller, als 1000 einzelne Speicherbefehle zu senden.

III. Analytik und Ausführung

Statistik: Sie nimmt alle gespeicherten Daten und wandelt sie in aussagekräftige Einblicke über politisches Verhalten und parlamentarische Aktivität um. Die Befehle wie printDataCounts(), statistics.speechLengthProPerson(), statistics.speechLengthProFaction(), statistics.commentFrequencyPerMember() usw. liefern die gewünschten Informationen für die Benutzer dieses Redenportals. Der Prozess innerhalb dieser Klasse erfolgt in dieser Reihenfolge: Verbindung zur Neo4j-Datenbank herstellen, Abfrage mittels Cypher-Sprache durchführen, Ergebnisse verarbeiten, in lesbare Tabellen formatieren, Einblick anzeigen.

MainApp: Dies ist das Kommandozentrum, das das gesamte parlamentarische Datenverarbeitungssystem orchestriert. Es importiert die Komponenten, die wir benötigen (import de.bundestag.factory.PortalFactory;, import de.bundestag.service.XMLProcessor;, import de.bundestag.service.DatabaseService;, import de.bundestag.service.Statistik;, import java.nio.file.Paths;). Es stellt sicher, dass der Benutzer einen Datenordner angegeben hat (Fehlerprüfung: if (args.length < 1) {...}). Auf diese Weise ist MainApp dafür verantwortlich, alle Komponenten des Projekts (RedePortal) aufzurufen und auszuführen.

Anwendungsfall diagramm:Das RedePortal ist für zwei Gruppen von Nutzern gedacht: den Admin (Verwalter) und den Benutzer (Leser) Der Admin ist für die gesamte technische Voraarbeit zuständig. Er liest die Protokolldateien ein und sortiert alle Informationen sofort in klare digitale Schubladen (z.B. Reden, Abgeordnete und Kommentare). Und er speichert diese sortierten Daten in einem digitalen Lager, der Datenbank, und kümmert sich um die allgemeine Ordnung und Pflege dieses Lagers. Als Erweiterung seiner Aufgaben kann der Admin auch verschiedene Statistiken aus den gespeicherten Daten erstellen. Der Benutzer interagiert nur mit den fertigen Ergebnissen, indem er sich anmeldet, um die aufbereiteten Informationen einzusehen. Er kann gezielt nach Reden filtern und diese lesen oder sich die

entsprechenden Kommentare und Zwischenrufe anzeigen lassen. RedenPortal sorgt der Verwalter dafür, dass das System funktioniert und die Datenbasis vorhanden ist, während der Leser das fertige Informationsangebot nutzt.