

Kursnamn: Integrationsplattformar
Klass: JIN23
Termin: Hösten 2023

Designa en Web Service i Java

INLEDNING

Bakgrundsbeskrivning	Apache Kafka är en distribuerad strömmande plattform som används för att hantera och behandla stora mängder data i realtid. Det används ofta för att bygga skalbara och pålitliga dataflöden mellan olika applikationer och system.
Varför ska ni utföra detta arbete?	I denna uppgift kommer du att få möjlighet att sätta upp och konfigurera ett Apache Kafka-kluster samt hantera dataflödet från ett web API till en Java Maven-applikation och en MySQL-databas, eller annan databas service.
Vad ska ni leverera?	<p>Inlämningen skall bestå av följande delar:</p> <p>En Maven Java applikation där användaren kan skriva in data i ett formulär (GUI eller konsol). Denna data skickas i JSON format till ett localhost Web API. Naturligtvis kan applikationen bestå av flera sub-applikationer, som alla sköter sin egen del.</p> <p>En Kafka Kluster som använder en producent för att lyssna på ovan nämnda Web API och hämta inkommande data. Datan skall sparas i en Topic. Via 2st Consumers skall datan skickas tillbaka till Java applikationen som bekräftelse samt också sparas i en lokal MySQL databas.</p> <p>En MySQL databas modellfil som initierar en databas passande för ändamålet.</p> <p>Java applikationen skall versionshanteras via en repository på GitHub. Repositoryn skall innehålla en ReadMe fil med väsentlig information för set-up och drift av er applikation och Kafka Kluster på en ny maskin och localhost.</p> <p>Vid inlämning skall ni lämna in projektet i en komprimerad zip/rar fil på inlämningsportalen. En invite/länk till GitHub repository är också ett godkänt inlämningsmetod via portalen.</p> <p>Inlämningen skall också inkludera installation eller instruktioner för att sätta upp er Kafka kluster på ny maskin.</p> <p>Självklart skall koden i projektet vara kommenterad och D.R.Y (Don't Repeat Yourself).</p>

ER PROJEKTUPPGIFT

Vad ska ni göra?	Din uppgift är att skapa och konfigurera ett Apache Kafka-kluster som tar emot data från ett web API i form av JSON-payload och sedan distribuerar denna data till både en Java Maven-applikation och en MySQL-databas. Uppgiften består av två huvuddelar: grundläggande (G) och väl genomförande (VG). Varje del innehåller specifika krav och mål.
Hur löser ni uppgiften?	Börja med att bestämma en idé om vilken typ av data ni vill er Java applikation skall hämta och presentera. Lista av era favorit-musik och

data om genre, information om era husdjur eller stats av ert nuvarande pokemon team, som exempel.

Fortsätt sedan med att skapa Servicen och Kafka servern först. Tänk på att skapa tester för er service och kontroller att servicen fortfarande fungerar under utvecklingens gång.

Skapa också klienten och testa den mot er Web API tidigt i utvecklingen innan ni fortsätter med nästa, för att veta att er klient får kontakt med servicen.

Efter detta, fortsätt med resterande tjänster er service skall ha.

Veckoplanering

Under detta arbete så kommer ni att ha delmål som sträcker sig över en eller två veckor. Detta för att ni skall vara i fas med kursens planering samt att veta om ni ligger efter med arbetet.

Vecka 35 - Denna vecka skall du ha färdigt din Apache Kafka server samt kan skicka text-meddelanden till den via ett Web API som hanteras via en Java Maven applikation.

Vecka 36 - Denna vecka skall du ha vidareutvecklat ditt Web API för att kunna skicka text i JSON format till Kafka servern.

Vecka 37 - Denna vecka skall du ha skapat en Consumer som skickar datan från din Kafka server till en Databas så att data sparas permanent.

Vecka 38 - Denna vecka skall du ha börjat din Java klient applikation som kommer att skicka och hämta data till och från Kafka servern. Bygg upp Unit Tester denna vecka som testar kommunikationen till din Web API och får ett respons meddelande tillbaka.

Vecka 39 - Denna vecka skall du bygga färdigt din Java applikation så att användaren (User) kan både skicka och ta emot JSON data till och från din Kafka server.

Vecka 40 - Denna vecka skall du skriva färdigt din dokumentation och se till att beskrivningen inkluderar den hjälp som en ny användaren skulle kunna behöva för att lansera ditt projekt på en ny maskin.

Vecka 41 - Sista veckan. Du skall färdigställa hela projektet och göra det redo för inlämning.

Struktur för arbetet
Vid grupparbete: arbetsfördelning och tidsplan

Uppgiften är en individuell uppgift.

INLÄMNING OCH REDOVISNING

Inlämning

Arbetet skall vara inlämnat innan slutet av kursen. Fredagen den 13 oktober.

Redovisning

Arbetet skall inte redovisas inför klass. Arbetet måste vara dokumenterat och redo för bedömning efter avslutad kurs.

BEDÖMNING OCH ÅTERKOPPLING

Bedömning sker med följande betygskriterier

Din uppgift kommer att bedömas baserat på hur väl du uppfyller både grundläggande och väl genomförande krav. Bedömningen kommer att inkludera följande aspekter:

- Korrekt konfiguration av Apache Kafka-kluster.
- Framgångsrik implementering av producenter och konsumenter.
- Effektivitet och skalbarhet av dataflöden.
- Korrekt implementerad felhantering och säkerhetsåtgärder.
- Kvaliteten på den skriftliga dokumentationen.

För godkänt (G) på projektarbetet skall följande krav uppfyllas:

- Sätt upp ett Apache Kafka-kluster bestående av minst tre noder (API, Databas, Java-applikation)
- Implementera en producent som hämtar data från det angivna web API:et och skickar det till en Kafka-topic.
- Skapa två separata Kafka-konsumenter: a. En konsument som lyssnar på Kafka-topic och skickar meddelandet till en Java Maven-applikation. b. En konsument som lyssnar på samma Kafka-topic och lagrar meddelandet i en MySQL-databas.
- Skriv dokumentation som beskriver ditt Apache Kafka-klusters konfiguration och hur producenten och konsumenterna är implementerade.

Väl Godkänt (VG) krav:

- Implementera felhantering i producer och consumer för att hantera eventuella nätverksproblem eller andra problem som kan uppstå vid dataflödet.
- Optimering: Utvärdera och implementera effektiva sätt att skicka och behandla meddelanden i ditt Kafka-kluster.
- Konfigurera Apache Kafka för att hantera replikering och felhantering för att säkerställa hög tillgänglighet.
- Utöka dokumentationen med en djupare förståelse av de val du gjort avseende konfiguration, säkerhet och optimering.

Återkoppling

Senast 2 arbetsveckor efter avslutad kurs.