**Setup:**

First install Kafka from the official website. <https://kafka.apache.org/downloads>

Make sure to download the binary version or you will run into errors when you start up zookeeper or kafka.

Graphical user interface, text, application, email

Description automatically generated

**Zookeeper**

**Navigate to the kafka folder and run the below command to start up zookeeper.**

C:\kafka > .\bin\windows\zookeeper-server-start.bat .\config\zookeeper.properties

Text

Description automatically generated

**Kafka**

**Navigate to the kafka folder and run the below command to start up kafka.**

C:\kafka> .\bin\windows\kafka-server-start.bat .\config\server.properties

Text

Description automatically generated

**Topic**

**Navigate to the kafka folder and run the below command create topic.**

C:\kafka\bin\windows>kafka-topics.bat --create --bootstrap-server localhost:9092 --replication-factor 1 --partitions 1 --topic employeetopic

Text

Description automatically generated

**Producer**

**Navigate to the kafka\bin\windows folder and run the below command start producer.**

C:\kafka\bin\windows>kafka-console-producer.bat --broker-list localhost:9092 --topic employeetopic

Text

Description automatically generated

**Consumer**

**Navigate to the kafka\bin\windows folder and run the below command start consumer.**

C:\kafka\bin\windows>kafka-console-consumer.bat --bootstrap-server localhost:9092 --topic employeetopic

Text

Description automatically generated

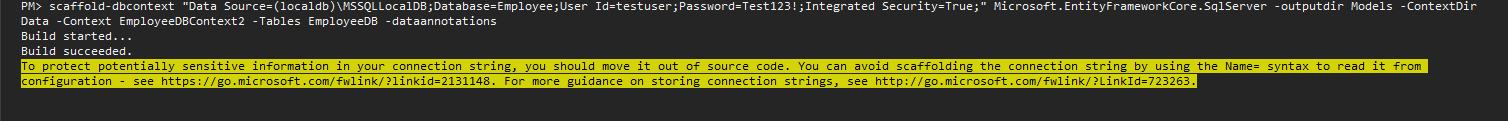
**Installed Client Library for SignalR**

Graphical user interface, text

Description automatically generated

Importing existing database using EF core

scaffold-dbcontext "Data Source=(localdb)\MSSQLLocalDB;Database=Employee;User Id=testuser;Password=Test123!;Integrated Security=True;" Microsoft.EntityFrameworkCore.SqlServer -outputdir Models -ContextDir Data -Context EmployeeDBContext -Tables EmployeeDB -dataannotations



**REST API to Consume data from database**

**API Endpoint :** [**https://localhost:7042/api/Employee?cursor=0&count=50**](https://localhost:7042/api/Employee?cursor=0&count=50)

Retrieves and displays only 50 records at a time ( 1 to 50 records) – achieved through cursor pagination. Next set of cursor value to use is passed in response header through “X-Pagination” value. Using the “X-Pagination” value, clients can easily pass the value in query parameters to retrieve the set of data they want.

Graphical user interface, text, application, email

Description automatically generated

Graphical user interface, text, application, email

Description automatically generated

**API Endpoint :** <https://localhost:7042/api/Employee?cursor=50&count=50>

Retrieves next set of 50 records from id 51 to 100.

Graphical user interface, text, application, email

Description automatically generated

Graphical user interface, text, application, email

Description automatically generated