Ministerul Educaţiei al Republicii Moldova

Universitatea Tehnică a Moldovei

Facultatea Calculatoare Informatică şi Microelectronică

Departamentul Ingineria Software și Automatică

**Raport**

Disciplina: Programarea aplicațiilor mobile.

Lucrarea de laborator nr. 1

**Tema:** Agent de mesagerie. Invocarea la distanță.

A efectuat: Zavorot Daniel, Prodan Marcel, Ceban Vitalie

gr. TI-194

A verificat: asis. univ Fiștic Cristofor

Chişinău 2022

**Scopul lucrării:**

Integrarea bazată pe agenți de mesaje care ar permite o comunicare asincronă dintre componentele   
distribuite ale unui sistem. Studierea mecanismului de invocare la distanță.

**Obiectivele lucrării:**

Realizarea sistemului utilizînd „Socket-uri” (Partea 1) și realizarea sistemului utilizînd invocările   
la distanță, “gRPC” (Partea 2).

1. **Realizarea sistemului utilizând ,,Socket-uri”.**

Partea 1 a lucrării de laborator ,,Agenție de mesagerie” constă în crearea unui sistem folosind ajutorul socket-urilor, acest sistem ar avea o componentă fizică care ar gestiona comunicarea dintre componentele unei aplicații distribuite. Avantajele acestei metode sunt:

* Reduce cuplarea - transmițătorii comunică doar cu brokerul, astfel o potențială grupare a mai multor receptori sub un nume logic comun poate deveni transparentă transmițătorilor.
* Mărește integrabilitatea - aplicațiile care comunică cu brokerul nu trebuie să aibă aceiași interfață, astfel brokerul poate deveni o punte dintre aplicații cu diferite nivele de securitate și calitate a serviciilor QoS.
* Mărește evolutivitatea - brokerul protejează componentele de modificările individuale ale aplicațiilor integrate, deseori oferînd capacități de configurare dinamică.

Dezavantajele acestei metode sunt:

* Crește complexitatea - brokerul comunicând cu toți participanții trebuie să implementeze multiple interfețe (protocoale) și în perspectiva performanței utilizează multithreadingul;
* Crește efortul pentru mentenanță - toți participanții trebuie să fie înregsitrați la brocker și se cere un mecanism de identificare a acestora;
* Reduce disponibilitatea - o singură componentă care intermediază comunicarea este singurul punct de eșec (single point of failure - eng.), căderea acestuia implică blocarea activității întregului sistem; această problemă se remediază prin dublarea brokerului și sincronizarea stărilor agentului primar și secundar;
* Reduce performanța - agentul de mesaje adaugă un pas adăugător, care implică cheltuieli suplimentare (overhead - eng.).
  1. Desfășurarea lucrării. Implimentarea publisher.

Inițializarea socketului în constructorul publisher, primul parametru corespunde de tipul adresei, în cazul nostu IPv4, al doilea parametru defineste tipul socketului și al 3-lea ce protocol este folosit în schimbul de date.

Изображение выглядит как текст

Автоматически созданное описание

BeginConnect – este varianta asincrona a functiei connect, care crează în background un alt fir de execuție.

Изображение выглядит как текст

Автоматически созданное описание

Aceasta are nevoie de un call back care ar verifica dacă socketul a fost conectat cu succes la broker.

Изображение выглядит как текст, снимок экрана, экран

Автоматически созданное описание

Metoda de transmitere a mesajului.

Изображение выглядит как текст, экран, снимок экрана

Автоматически созданное описание

* 1. Desfășurarea lucrării. Implimentarea broker.

Initializarea socketului în interiorul constructorului.

Изображение выглядит как текст

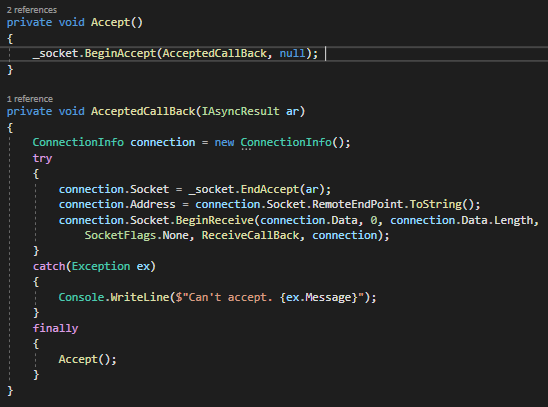
Автоматически созданное описание

Metoda de start a brokerului.

Изображение выглядит как текст

Автоматически созданное описание

Metoda accept care apelezea metoda asinhrona de a incepe accepta packetele a socketului.



* 1. Desfășurarea lucrării. Implimentarea subscriber.

Inițializarea socketului cu aceeași parametri ca sender socket și broker socket.

Изображение выглядит как текст

Автоматически созданное описание

Metoda de conectare la socketul brokerului.

Изображение выглядит как текст

Автоматически созданное описание

Beggin connect este metoda asinhrona a socket connect și are nevoie de un callback. Acest callback verifică daca socketul e connectat și începe să recepționeze date de la broker.

Изображение выглядит как текст

Автоматически созданное описание

Metoda de subscribe, în ea se declara topicul la care subscriberul este connectat și se trimit aceste date către broker.

Изображение выглядит как текст

Автоматически созданное описание

Metoda de a începe recepționarea datelor de la broker.

Изображение выглядит как текст

Автоматически созданное описание

1. **Realizarea sistemului utilizând ,,gRPC”.**

Partea 2 a lucrării de laborator numărul 1 constă în a integra sistemul de mesagerie la un nivel mai înalt și anume utilizând RPC-uri (Remote Procedure Call), sistemul nostru va fi integrat cu gRPC care este un framework modern bazat pe RPC. Agentul de mesaje (message broker - eng.) Este o componentă fizică care gestionează comunicarea dintre componentele unei aplicații distribuite. Avantajul utilizării acestei tehnici constă în decuplarea receptorului de transmițătorul mesajelor. Prin urmare o aplicație participantă transmite mesaje doar agentului, indicând un nume logic la receptor. Agentul poate expune diverse interfețe aplicații în colaborare și poate transfera mesajele între acestea, ne impunând o interfață comună tuturor participanților întru asigurarea interacțiunii.

* 1. Implimentarea agentului de mesagerie cu ajutorul gRPC.

Serviciul publisher contine metota PublishMessage care lucrează ca un task, ceea ce ar însemna ca execută o operație asinhronă. Această metodă corespunde de primirea mesagelor de la un publisher și adăugarea lui într-un container de message.

Изображение выглядит как текст

Автоматически созданное описание

Serviciul subscriber corespunde de conectarea unui subscribe la broker-ul nostru, aceasta clasă inițializeaza o conexiune nouă și o adăugă într-un container de conexiuni. Fiind o metodă care returnează un task acceasta execută operațiile în mod asinhron. În caz că conexiunea eșuează se afișează un mesaj de notificare în clientul de broker.

Изображение выглядит как текст

Автоматически созданное описание

Această clasă este senderwoker-ul nostru, ea moștenește de la interfața IhostedService care reprezintă o colecții de servicii care lucrează în background de la inițializarea aplicației. Această interfață implimentează 2 metode StartAsync care este apelată la inițializarea aplicației și StopAsync o metodă pentru dezactivarea serviciului care se apeleză la momentul inchiderii aplicației.

Изображение выглядит как текст, снимок экрана, экран

Автоматически созданное описание

Metoda StartAsync intializează un task nou la fiecare 2 secunde ce apelează metoda DoSendWork. Această metodă corespunde de tratarea tuturor mesajelor care se afla în container și trimiterea lui la subscriberi. Pentru aceasta el apelează și containerul de conexiuni pentru a prelua adresa subscriberului.

Изображение выглядит как текст

Автоматически созданное описание

1. **Compararea metodelor de implimentare.**

Prima metoda a fost una low level unde s-a folosit socket-urile, iar a doua metodă a fost implimentată cu gRPC. RPC este protocolul, socketul oferă acces la transport pentru a implementa acel protocol. RPC este serviciul și protocolul oferit de sistemul de operare pentru a permite declanșarea codului pentru executarea unei aplicații la distanță. Are un protocol definit prin care procedurile sau obiectele pot fi accesate de un alt dispozitiv printr-o rețea. O implementare a RPC se poate face practic în orice transport de rețea (de ex. TCP, UDP, cupe cu șiruri). Socket-ul este doar o abstracție de programare, astfel încât aplicația să poată trimite și primi date cu un alt dispozitiv printr-un anumit transport de rețea. Implementați protocoale (cum ar fi RPC) deasupra unui transport (cum ar fi TCP) cu un socket.

Aplicația realizată cu ajutorul socket-urilor, a atras multă atenție la interacțiunea dintre mai multe dispozitive, cum s-ar face acest transfer de date dintre dispozitive, însă gRPC a permis lucrul la un nivel mai înalt, adică s-a putut de atras atenția la nuanțele agenției de mesagerie. Astfel gRPC a permis o implementare mai ușoară și din această cauză a permis realizarea altor funcționalități ale agentului de mesagerie.

Anexe

Codul sursă: