Tapi Messenger

Dokumentacja

Autor: Artur Margielewski

Wstęp

Moduł Tapi Messenger pozwala na komunikację pomiędzy przeglądarką internetową a modułem **Tapi Server**, czyli programem obsługującym centralkę telefoniczną.
Tapi Messenger do komunikacji z przeglądarką wykorzystuje WebSockety i bibliotekę SockJS i STOMP, czyli protokół oparty na wiadomościach tekstowych. Do komunikacji z programem Tapi Server wykorzystano komunikację przez gniazda (sockety). Program wykorzystuje w swoim działaniu szkielet aplikacyjny Java Spring oraz posiada wbudowane serwer Tomcat.

Użyte technologie:
Java Spring
SockJS
STOMP

Użytkowanie

Aby skorzystać z modułu należy uruchomić aplikację w postacie pojedynczego pliku z rozszerzeniem .jar.

Moduł TapiMessenger wykona próbę nawiązania połączenia z TapiServer, który powinien być uruchomiony na tej samej maszynie. Komunikacja między moduami będzie odbywała się na porcie numer 44444. Jeżeli próba połączenia zakończy się niepowodzeniem po 3 sekundach próba zostanie powtórzona.

WebSockety, czyli przeglądarki internetowe lub inne programy, które z nich korzystają, mogą połączyć się z serwerem TapiMessenger pod adresami:

172.16.35.51:8081/tapi-messenger

Następnie wiadomości do klienta wysyłane są pod adresami:

/user/queue/incoming-call-user{{SimpSessionId}}

SimpSessionId to id negocjowane w momencie połączenia z serwerem. Można je odczytać przez manipulację adresem.

/topic/is-listen-init'

Przykładowa funkcja nawiązująca połączenie:

```
function connect() {
    var socket = new SockJS('172.16.35.51:8081/tapi-messenger');
//172.16.35.51:8081
    stompClient = Stomp.over(socket);
    stompClient.connect('mylogin', 'mypasswd', function (frame) {
        var apexSession = $v("pInstance");
        var url = stompClient.ws._transport.url;
```

```
        url = url.replace("http","").replace("ws","");
        url = url.replace(
            "://172.16.35.51:8081/tapi-messenger/",  ""); //172.16.35.51:8081
        url = url.replace("/websocket", "");
        url = url.replace(/^[0-9]+\//, "");
        console.log("Your current session is: " + url);
        setConnected(true);
        console.log('Connected: ' + frame);
        stompClient.subscribe('/user/queue/incoming-call-user' + url, function
(caller) {
            setCallerBadge(JSON.parse(caller.body));
        });
        stompClient.subscribe('/topic/is-listen-init', function (resp) {
console.log(resp); });
        stompClient.subscribe('/user/queue/incoming-call-user' + apexSession,
function (caller) {
            setCallerBadge(JSON.parse(caller.body));
        });
        introduce(apexSession);
    }, function (frame) {
   console.log('Error occured');
   reportError();});
}
```

Nawiązanie połączenia

/tapi/listen

Po stronie przeglądarki wywoływana jest funkcja connect(). Przeglądarka subskrybuje na ramki przychodzące pod danymi adresami i ustala funkcje callback dla każdego adresu. Przed nasłuchiwania na numer telefonu użytkownika konieczne jest podanie aplikacyjnego numeru sesji na podstawie którego zostanie ustalony numer telefonu na którym rozpocząć nasłuchiwanie. Jest on podawany jako nagłówek opisany kluczem ‘apex_session’. Wiadomość z takim nagłówkiem i dowolną treścią wysyłana jest na adres ‘tapi/listen’.

TapiMessenger nie wykonuje żadnych akcji w momencie połączenia się klienta. Podejmuje akcje w momencie gdy klient przedstawia się aplikacyjnym numerem sesji, czyli numerem sesji z Apex’a. (Numer sesji można uzyskać wywoływując w przeglądarce funkcję $v(„pInstance”)). Wtedy TapiMessenger sięga do bazy danych, w której przechowywane są numery aplikacyjne numery sesji, sprawdza nazwę użytkownika i powiązany z nim numer telefonu. Jeżeli rekord został odnaleziony TM dodaje wpis do mapy łączącej numery telefonu z numerami sesji. Jeżeli dodana sesja jest jedyną sesją związaną z tym numerem telefonu TM wysyła informację do TS, aby ten rozpoczął nasłuchiwanie.

Połączenia przychodzące

/user/queue/incoming-call-user{{sessionId}

W momencie gdy TapiMessenger odbierze od TapiServer wiadomość o połączeniu przychodzącym wysyła tę informację do wszystkich niezamkniętych WebSocketów związanych z tym numerem telefonu. To, które sesje (WebSockety) związane są z danym numerem telefonu zapisane jest w mapi asocjacyjnej, która jest modyfikowana podczas nawiązywania nowych połączeń przeglądarka-TapiMessenger lub rozłączania połączeń przeglądarka-TapiMessenger.

Informacja o połączeniu przychodzącym wysyłana jest na adres:

/user/queue/incoming-call-user + id sesji (SimpSessionId).

Połączenie przychodzące reprezentowane jest przez klasę Caller.

```
public class Caller {
    private String name;
    private String phone; … }
```

Pole name ma wartość UNKNOWN, ponieważ odczyt danych z bazy został przeniesiony do aplikacji OracleApex.

Połączenia wchodzące

/tapi/call

W momencie gdy użytkownik chce rozpocząć nowe połączenie powinien wysłać wiadomość na adres:

/tapi/call

i przesłać obiekt zawierający pole ‘toNumer’ ze stringiem opisującym numer do którego chce zadzwonić.

```
stompClient.send("/tapi/call", JSON.stringify({'toNumber': number}))
```

Serwer w tym momencie pobierze z mapy numer telefonu skojarzony z tym numerem sesji WebSocketu wysyłającego wiadomość (numer sesji pobrany jest z nagłówka) i wyślę do TapiServer wiadomość z prośbą o nawiązanie nowego połączenia. Dokona też weryfikacji czy z tym numerem sesji związany jest jakiś numer telefonu.

Zamykanie połączenia

W momencie gdy połączenie zostanie zamknięte lub zerwane serwer usuwa z mapy asocjacyjnej wpis na temat danego numeru sesji. Gdy ostatni WebSocket związany z danym numerem zostanie zamknięty TapiMessenger wysyła do TapiServer wiadomość proszącą o zaprzestanie nasłuchiwania.

Komunikacja TapiMessenger TapiServer

Moduły te porozumiewają się przez tradycyjny socket na porcie 44444. Korzystają z protokołu opartego o stringi.