دانشگاه صنعتی امیرکبیر

(پلی تکنیک تهران)

دانشکده مهندسی برق

پروژه درس تئوری صف

نویسنده : حسین غلامی - ۹۷۱۲۳۰۲۱

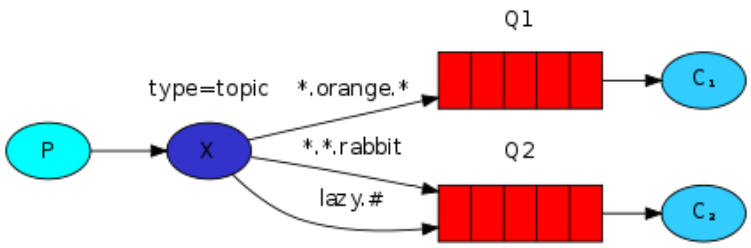
استاد درس

سیدمصطفی صفوی همامی

فهرست

چکیده

Rabbitmq یک بروکر و دریافت کننده پیام بر بسستر اینترنت است که در زمینه اینترنت اشیا کاربرد فراوانی دارد ، از این پلتفرم میتوان به عنوان بخشی که پیام ها را دریافت میکند ، استفاده کرد ، چرا که قابلیت های فراوانی را در اختیار کاربران قرار میدهد که در این بین میتوان به پشتیبانی از پروتکل هایی مانند AMQP و STOMP و MQTT اشاره کرد. همچنین این نرم افزار علاوه بر دریافت ، میتواند پیام ها دسته بندی کرده و صف ایجاد کند



هدف پروژه

در این پروژه سعی شد با کمک پارامترهای ارائه شده توسط نوع صف Classic Mirrored ،مدل های مختلفی از صف را ارائه داد، هدف در این پروژه دریافت بسته توسط پروتکل MQTT با QoS ۰و۱ ، انتقال به صف Classic Mirrored و دریافت و پردازش توسط m ، consumer یا سرویس دهنده است که در این بین بتوان از اکثر ویژگی هایی که این صف در اختیار ما قرار میدهد استفاده نمود. پیاده سازی شد

۱-مقدمه

۱-۱ اینترنت اشیا و کاربردها

اینترنت اشیاء ، شبکه ای از اشیاء فیزیکی یا چیزهای تعبیه شده با قطعات الکترونیکی، نرم افزار، وسنسورها و اتصالات است ؛که این اتصالات می توانند توسط تبادل اطلاعات با تولید کننده اطلاعات، اپراتورها و یا دیگر دستگاه ها ارزش و خدمات تولید کنند. هر شئ به تنهایی توسط سیستم تعبیه شده ، قادر به شناسایی است و همچنین میتواند با زیر ساخت اینترنت موجود نیز تعامل داشته باشد.

اینترنت اشیاء مفهومی محاسباتی است، برای توصیف آینده‌ای که در آن اشیای فیزیکی یکی پس از دیگری به اینترنت وصل می‌شوند و با اشیای دیگر در ارتباط قرار می‌گیرند. مفهوم اینترنت اشیاء، به نظر شبیه داستان‌هایی علمی تخیلی است که در آن یخچال‌ها حرف می‌زنند و خودروها به طور خودکار استارت می‌خورند. اما ارتباط با بستر اینترنت در وسایلی که دائم به یکدیگر متصل هستند، بسیار بیشتر از یک خانه یا خودرو هوشمند بر زندگی ما اثر خواهد گذاشت.

در این تکنولوژی به هر چیز یک شناسه منحصر به فرد تعلق می¬گیرد که بتواند داده¬ها را برای پایگاه داده مشخص شده ارسال کند. دنیای امروزی دنیایی است که در آن، داده و اطلاعات حرف اول را می زنند و به نوعی می¬توان به آن¬ها مفهوم طلای مجازی را اطلاق کرد، اینترنت اشیاء نیز با دانستن این مهم، بر پایه داده شکل گرفته و آن را به تمامی اشیای پیرامون محیط زندگیمان بسط داده است. به طور مثال فرض کنید

که یخچال منزل شما دارای این تکنولوژی می باشد (یعنی امکاناتی تعبیه شده دارد که می تواند داده را ذخیره و توسط زیرساخت اینترنت، آن را به پایگاه داده ها بفرستد که شما قادر به دیدن اطلاعات آن باشید) حال فردی از اعضای خانواده¬تان بطری شیر را برداشته و آن را مصرف میکند. یخچال نیز بعد از گذشت زمان تعیین شده¬ای، به تمام شدن شیر پی برده و آن را به شما اطلاع می دهد.

طبق تعریف ارائه شده در ویکی پدیا عبارت اینترنت اشیا، برای نخستین بار در سال ۱۹۹۹ توسط کوین اشتون مورد استفاده قرار گرفت و جهانی را توصیف کرد که در آن هر چیزی، از جمله اشیا بی‌جان، برای خود هویت دیجیتال داشته باشند و به کامپیوترها اجازه دهند ، آن ها را سازماندهی و مدیریت کنند. اینترنت در حال حاضر همه مردم را به هم متصل میکند ولی با اینترنت اشیاء تمام اشیاء به هم متصل می شوند. اینترنت اشیاء مفهومی جدید در دنیای فناوری و ارتباطات بوده و به طور خلاصه فناوری مدرنی است که در آن برای هر موجودی (انسان، حیوان و یا اشیاء) قابلیت ارسال داده از طریق شبکه‌های ارتباطی، اعم از اینترنت یا اینترانت، فراهم می‌شود. بستر اینترنت اشیاء بر امواج رادیویی بی‌سیمی قرارداده شده که به دستگاه‌های مختلف این امکان را میدهند تا از طریق اینترنت با یکدیگر به برقراری ارتباط بپردازند. این بستر شامل استانداردهایی مانند وای‌فای، بلوتوث کم‌مصرف ZigBee و LORAWLAN SIGFOX و ... است.

۱-۲معماری اینترنت اشیاء

معماری در اینترنت اشیاء از یک هرم تبعیت میکند ، که از طریق شکل زیر به توضیح پیرامون هر بخش آن پرداخته می شود.

مدل کسب و کار

قبل از اینکه بخواهیم پروژه ای در زمینه اینترنت اشیاء انجام دهیم ، باید مدل کسب و کار وکاربرد اصلی را پیش بینی و طراحی کنیم. در ابتدا می بایست به این سوال پاسخ دهیم که هدف اصلی پردازش ما چیست و از انجام این پروژه سعی در رفع کدام نیاز داریم. در ادامه می بایست اطلاعاتی که قصد پردازش آن را داریم را

تجمیع کنیم ، آنگاه اطلاعاتی را که برای تصمیم گیری نیاز داریم را در شاخه بندی متناسب با تصمیم قرار دهیم تا به بهترین تصمیم برسیم و در ادامه ویژگی های قابل بهبود را بررسی کنیم و برای آن ها راهکار ارائه دهیم.

پلتفرم یا سکو

یکی از مهم ترین بخش ها در هر پروژه در زمینه اینترنت اشیاء بخشی است که وظیفه دریافت اطلاعات را بر عهده دارد این دریافت با توجه به حجم انبوه پیام های دریافتی بسیار زیاد است و میبایست در یک صف اطلاعات را دریافت و پردازش کند به طور تخصصی به این فرایند تجمیع سازی یا integration می گویند همچنین پلتفرم علاوه بر وظیفه دریافت اطلاعات ، وظیفه پردازش آن ها نیز بر عهده دارد که در این بین با مسئله دادگان انبوه یا Big data مواجه هستیم ، یرای ساده ترین تعریف Big Data میتوان گفت حجم اطلاعات بیش از اندازه رم ما باشد . پس باید بتوانیم با کمک تکنیک های پردازش توزیع شده و استفاده از خوشه ای از پردازنده ها این مسئله را حل کنیم

دستگاه های ارتباطی

دستگاه های ارتباطی به طور کلی چند وظیفه کلی دارند ، که به شرح زیر است:

ایجاد درگاه امن ، تبادل اطلاعات با پلتفرم

قابلیت پیاده سازی پروتکل های تبادل اطلاعات با پلتفرم) MQTT,REST,(...,

پشتیبانی از پروتکل های تبادل اطلاعات با قطعات الکترونیکی) GPIO,UART,SPI,I2C,CAN(

ذخیره سازی داده ها به منظور انجام تحلیل های کوتاه مدت

حسگر ها و عملگر ها

به طور کلی حسگر ها و عملگر ها ، تولید کنندگان اطلاعات و مصرف کنندگان اطلاعات می‌باشند.

به نحوی که حسگر ها اطلاعات(مثلا دما اتاق) را تولید کرده و از طریق دستگاه های ارتباطی به پلتفرم ارسال می‌کنند.همچنین دستگاه های ارتباطی متناسب با داده هایی که از پلتفرم دریافت میکنند ، عملگر ها را کنترل میکند و عملگر ها رخ دادهایی در دنیای فیزیکی ایجاد میکنند که منجر به رسیدن ما به هدف مطلوب میشوند.

۲- نقش تئوری صف در پلتفرم های اینترنت اشیا

همانطور که پیشتر گفته شد ، پلتفرم وظیفه دریافت اطلاعات و پردازش آن ها را دارد که در این پروژه تنها به بخش دریافت اطلاعات و تشریح آن و پروتکل MQTT

۲-۱ MQTT چیست؟

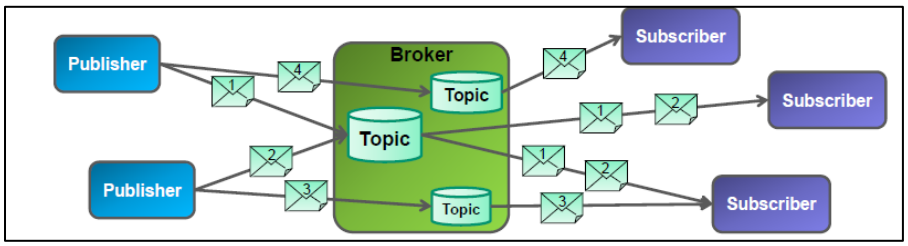
MQTT یک پروتکل سبک انتقال اطلاعات صف بندی شده است. همانطور که از نام آن پیداست (انتقال از راه دور پیام های صف بندی شده) برای ارسال اطلاعات از راه دور مناسب است. از آن جایی که MQTT بسیار سبک و ساده است بنابراین برای سناریو های M2M، WSN، IOT که سنسور ها و عملگر ها از طریق یک MQTT broker باهم به تبادل اطلاعات می‌پردازند بسیار کاربردی است. مثال :

- یک سنسور نور به صورت زمان پیوسته داده های خود را به سمت broker ارسال می‌کند .
- ساختن یک نرم افزار که داده ها را از broker دریافت کند و تصمیم بگیرد که آیا نور مناسب است یا خیر و ارسال فرمان به broker
- ساختن یک عملگر که داده ها (فرمان ها) را از broker دریافت کند و اقدام عملی محیط پیرامون که روشن و خاموش کردن چراغ های محیط است را انجام دهد.

.

۲-۲ ویژگی های کلیدی MQTT

اولین و شاید مهمترین ویژگی MQTT را می‌توان سبک(ساده) بودن پیام ها ی صف‌بندی شده وقوانین انتقال آن را در نظر گرفت. ویژگی بعدی را می‌توان مخابره آسنکرون اطلاعات در نظر گرفت ، که به صورت رویداد[1] محور مخابره‌ها انجام می‌شود. که این امکان را می‌دهد بعضی امکانات را غیر فعال کرده تا رویداد جدید ظاهر شود ، و با رخ داد یک رویداد ، فرایندی مشخص را انجام داد.ویژگی کلیدی دیگری که می‌توان در نظر گرفت ، این است که پهنای باند کمی را اشغال می‌کند (تنها ۲ بایت هِدر به پکت‌های داده اضافه می‌کند) که برای شبکه ها با پهنای باند پایین بسیار مناسب است.از دیگر مزیت های استفاده از MQTT استفاده از مدل PubSub است که از ساده ترین روش های انتقال اطلاعات است. به نحوی که ارسال کننده ، داده ها را Publish و شنونده، داده ها را Subscribe می‌کند.از ویژگی های اساسی این پروتکل ، جدا کردن اطلاعات فرستنده ،شنونده های مختلف از طریق topic هاست. به نحوی که مثلا فرستنده شماره ۱ داده های خود را در تاپیک A وفرستنده شماره ۲ داده های خود در تاپیک B ارسال می‌کند و شنونده ۱ داده های خود را از تاپیک A و شنونده۲ داده های خود را از تاپیک B دریافت می‌کند.یکی دیگر از ویژگی ها این است که ارتباط فرستنده و شنونده با broker از طریق پروتکل TCP [2] است، که ارسال داده و امنیت آن را (نسبت به UDP) تضمین می‌کند.



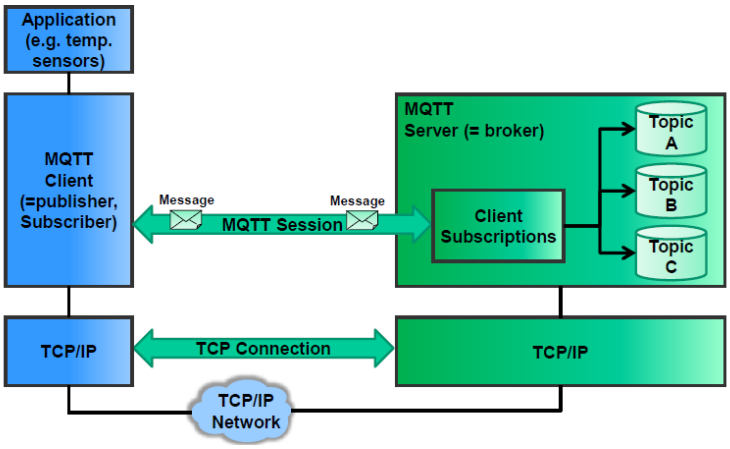
[1] event

[2] Transmission Control Protocol

بخش های اصلی یک ارتباط از طریق MQTT شامل :

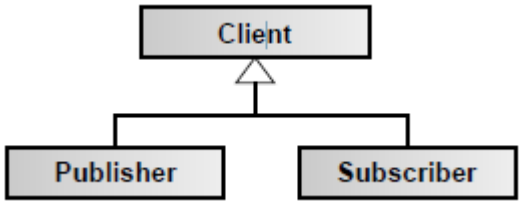
۱)کاربران MQTT[3](فرستنده و شنونده) ۲) سرور (broker) ۳) جلسه [۴] ۴) تاپیک‌ها ۵) اشتراک[۵]

به معماری زیر دقت کنید:



کاربران MQTT ، شامل فرستندگان(Publisher) و مشترک‌شوندگان(Subsciber) می‌شوند،آن ها می‌بایست

در یک تاپیک مشترک شوند (subscribe کنند) تا بتوانندداده را ارسال و دریافت کنند.



[3] MQTT client

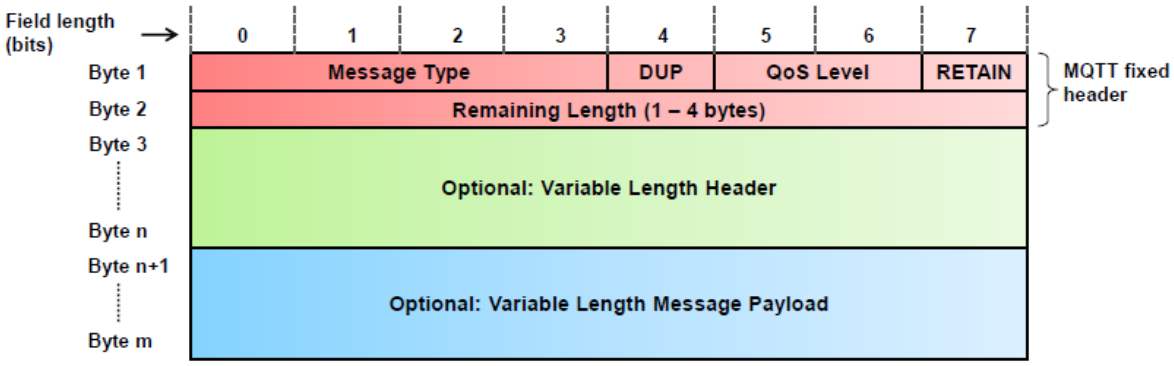
[4] sessions

[5] subscription

سـرور ها تاپیک ها را می‌سـازند و اجرا می‌کنند. یعنی هنگامی که کاربران (فرسـتنده‌ها و شـنونده‌ها) روی یک تاپیک مشترک می‌شوند، (subscribe می‌کنند) ، وظیفه دریافت داده ها و انتقال آن را برعهده دارند.

-ساختار پیام ها در پروتکل MQTT

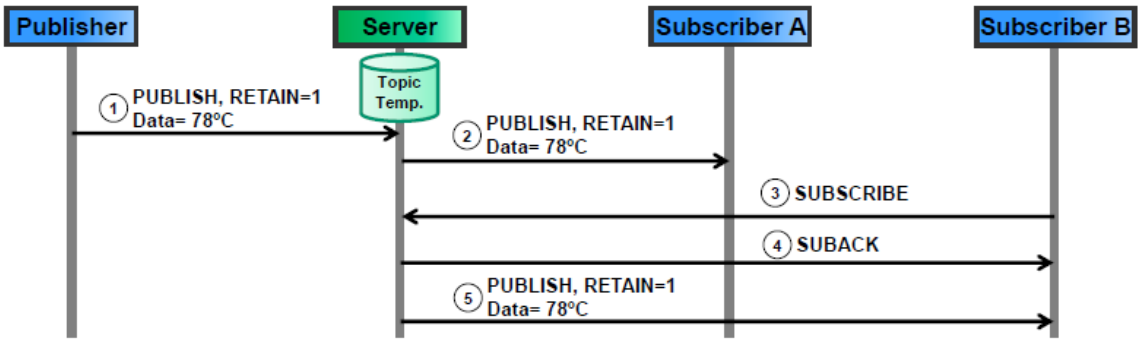
به طور کلی پیام ها در این پروتکل از شـکل زیر (شـکل ۳-۵)پیروی می‌کنند و با تغیر نوع پیام سـاختار ها کمی تغییر می‌کنند ، در این بخش به صورت مختصر با برخی از جزئیات این پروتکل آشنا می‌شویم.



همانطور که در دو شکل فوق دیده می‌شود به طور خلاصه:

- Massage Type : ۱۴ نوع متفاوت پیام منتقل میشود (۱ تا ۱۵) که در بخش بعدی اندکی با جزئیات آن آشنا می‌شویم. البته نوع فریم بندی دقیق آن در فصل ششم آورده شد.

- DUP[6]: این پرچم نشان می‌دهد که آیا پیام ارسال شده تکراری است؟ یاخیر.
- RETAIN :اگر این پرچم ۱ باشد ، سرور موظف است اخرین پیام را نگه دارد و در صورت اشتراک کاربر جدید اطلاعات ذخیره شده را به آن ارسال کند .



- remaining length : تعداد بایت های پیامی که منتقل خواهد شد را در بر خواهد داشت.
- QoS[7] : MQTT به مقصد رساندن داده ها را تضمین می‌کند،با وجود اینکه TCP/IP به مقصد رساندن اطلاعات را تضمین می‌کند ولی اگر مسیر ارتباطی TCP قطع شود ، داده ها ممکن است از بین بروند، بنابراین MQTT سه سطح کیفت از سرویس را ارائه می‌دهد.
  - QoS level 0 : حداکثر یک بار ارسال : در این سطح تضمین همان ، تضمین TCP/IP است

[6] Duplicate Massage Flag
[7] Quality of Service

و اطلاعات تنها یک بار ارسال می‌شوند؛ کاربرد :مثلا اطلاعات سنسور دما ، چرا که اگر یکی از اطلاعات را از دست دهیم برایمان اهمیتی چندانی ندارد ، چرا که میزان متوسط آن برایمان اهمیت دارد، و اگر داده تکراری داشته باشیم ، متوسط‌گیری غلط خواهد بود.

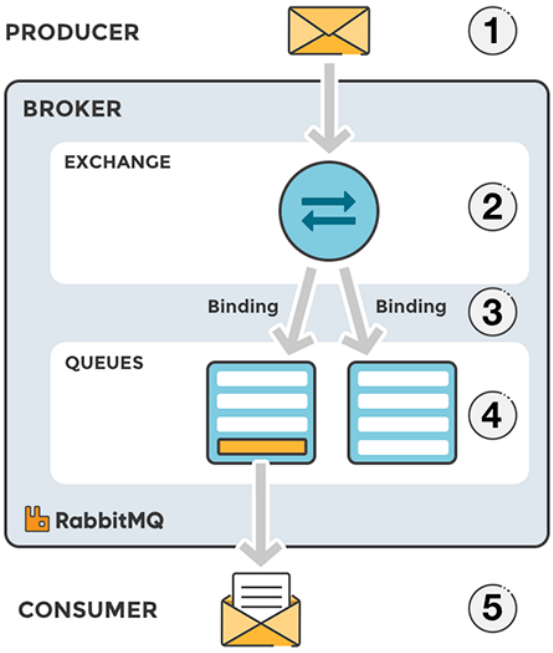
- QoS level 1 : حداقل یک بار ارسال : در این سطح تضمین دریافت داده ها قطعی است ولی داده اضافی ایجاد می‌شود.کاربرد : مثلا سنسور درب منزل باز و بسته بودن درب را گزارش می‌دهد ولی برای ما تغییرات در حالت درب اهمیت دارد و داده تکراری مهم نیست و همچنین نباید این داده ازبین برود، چرا که در عملگر شاید باید یک دزدگیر فعال شود.
- QoS level 2 : دقیقا یک بار ارسال : این بالاترین کیفیت ارائه شده است که تضمین می‌کند داده دقیقا یک بار به مقصد می‌رسد ،البته برای این سطح از کیفیت باید پیام را ذخیره کرد. کاربرد ؛ در محیط هایی که وجود اطلاعات اضافی باعث ایجاد اخلال می‌شود.مثلا؛ به صدا در آمدن یک صدا به عنوان عملگرکه دقیقا باید یک بار پیام به مقصد برسد.

۳-طراحی و آماده سازی دریافت کننده پیام (RabbitMQ)

حال که با مفاهیم پروتکل استاندارد MQTT آشنا شدیم به پیاده سازی و طراحی هر یک از اعضای آن میپردازیم ، در این بین RabbitMQ یک بروکر پیام است ، که از پروتکل های بسیاری نظیر AMQP و STOMP و MQTT ، پشتیبانی میکند. این نرم افزار که به سادگی میتواند به صورت Dockerized روی Cloud اجرا شود که به این معناست برای پیاده سازی پروتکل مورد بحث ما مناسب است و متوانیم یک کلاستر از آن را بر روی Cloud اجرا کنیم و این Container فرایند ایجاد Session را با امنیت بسیار بالا برای ما انجام می دهد ، پس در این پروژه ما باید به ترتیب ۱- پیاده سازی و طراحی Topic ها (در این نرم افزار به آن Exchange میگوید) آماده سازی صف ها و اتصال (Bind) کردن ۲- گیرنده پیام (subscriber یا consumer) ۳- ایجاد یک داشبورد و محیط برای دیدن و مشاهده کردن و تغییرات (یک وب سرور Django بر روی کامپیوتر بالا می آوریم تا بتواند بخش های فوق را به هم متصل کند و نمایش آن قابل فهم و درک باشد و بتوان در همان جا بتوان وضعیت صف ها را بررسی نمود) انجام شد. حال به بررسی و طراحی هر یک از بخش های فوق میپردازیم

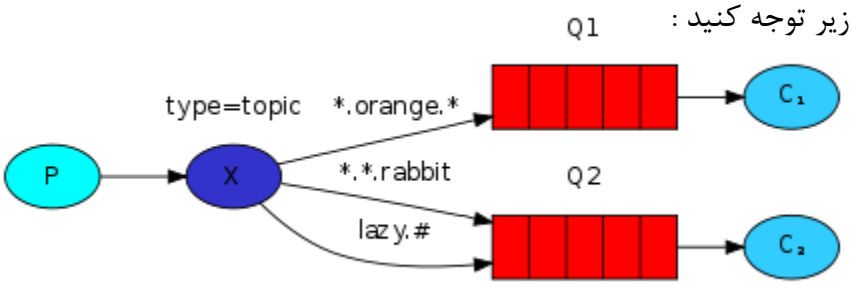
۱- برای پیاده سازی و طراحی تاپیک ها در صف ابتدا به امر به بررسی چند ویژگی که rabbitMq در اختیار ما قرار میدهد ، می پردازیم ، همانطور که در تصویر زیر مشاهده میکنید ، پیام ها توسط پروتکل

MQTT به ex.mqtt منتقل میشـند حال باید پیام ها را با توجه به نوع Topic و Routing Key که هر پیام برای خود دارد به صف مربوطه خود منتقل شود.



در این بخش به انواع تعریف Routing key ها میپردازیم ، همانطور که در پروتکل ذکر شد در هدر بسته بخشی تحت عنوان topic وجود دارد این هدر مشخص میکند که بسته در کدام یکی از صف ها قرار بگیرد . برای مثال stock.usd.ny و quick.orange.rabbit می توانند هر دو نام یک تاپیک برای بسته باشند که بسته ها را به صف مربوطه متصل میکنند. برای صف نیز یک Routing key تعریف میشود که برای اینکه بسته درون آن صف قرار گیرد ، باید topic با Routing Key یکی باشد همچین در این بین دو کارکتر * و # در Routing Key استفاده کرد که عملکردی متفاوت برای کنترل بسته ها در اختیار ما قرار میدهند

* به معنی این است که میتواند در آن بخش متمایز باشند ولی همچنان باید فرمت کلی حفظ شود به این معنا که بسته ای با تاپیک quick.orange.rabbit میتواند وارد صفی با routing key ، *.orange.* قرار بگیرد ولی حتما باید از ۳ بخش تشکیل شده باشد برای باز تر کردن موضوع کارکتر # به این معنا است که میتواند در همان بخش یکسان نباشد و همچنین میتواند از تعداد بخش های یکسان تشکیل نشده باشد برای مثال بسته ای با topic ، quick.orange.rabbit.salam میتواند در صفی با Routing key ، quick# قرار بگیرد ، لازم به ذکر است میتوان برای یک صف میتوان از چند routing key استفاده نمود. به شکل زیر توجه کنید :



در این سیستم یک تولید کننده اطلاعات داریم که همه اطلاعات خود را به exchange X منتقل میکند همچین ۲ صف Q1,Q2 را داریم که صف Q1 با routing key ، *.orange.* روی exchange X تعریف کرده (Bind) کرده و میتواند بسته هایی با topic ۳ قسمتی که بخش دوم آن orange هست را دریافت کند و توسط مصرف کننده c1 اطلاعات را مصرف کند ، همچنین صف Q2 بسته های ۳ قسمتی که بخش سوم آنها rabbit است را دریافت میکند علاوه بر این ، بسته هایی که بخش اول آن ها lazy است را نیز دریافت میکند

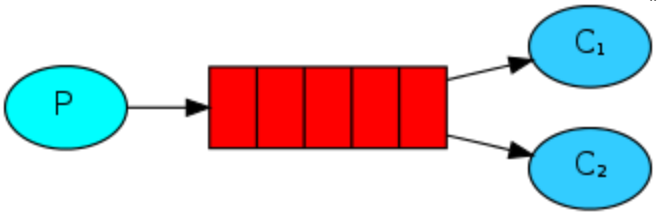
این مکانیزنم میتواند برای دسته بندی پیام های ورودی و بسیار موثر باشد.

۳-۱: نحوه پیاده سازی صف در RabbitMQ

به طور کلی پیام ها به دو صورت کلی تقسیم بندی میشوند پیام های transient و پیام های persist شده پیام ها ، به طور عمومی هر دو این پیام ها در حافظه نوشته میشوند و و در صورتی که حد آستانه ای که برای صف تنظیم شده بیشتر شوند پیام های persist شده روی دیسک نوشته میشوند و تنها یک اشاره گر که محل ابتدای ادرس دیسک ثبت شده در حافظه نگهداری میشوند ، پیام هایی که به مصرف کننده ارسال میشوند قبل از ارسال وارد حافظه شده که به این پیام ها transient گفته میشود ، در رم نگه داری میشوند و بعد از اینکه دریافت کننده ACK آنها را ارسال کرد از حافظه نیز پاک میشوند ، در این بین پس میزانی که حافظه اشغال میشود در حالتی که بر روی یک سخت افزار اجرا شود بسیار بهینه است ، حال درصورتی که این نرم افزار را در حال توزیع شده استفاده میکنیم اطلاعات در بخش های مختلفی از این نرم افزار اجرا میشود میتوان از حالت آینه شده نیز استفاده کرد به این معنا که اگر یکی از سخت افزار هایی که اطلاعات روی دیسک آن یا حافظه آن قرار گرفته میتواند به صورت duplicate در جایی دیگر ذخیره شود و با همین شیوه توسط node مرکزی کنترل شود

۳-۲سرویس دهنده یا مصرف کننده اطلاعات

همانطور که درمباحث کلاس ذکر شد میتوانیم چندین سرویس دهنده به طور همزمان استفاده کنیم تا بتوانیم حجم بسته های داخل صف را مدریت کنیم علاوه بر مباحثی که در کلاس ذکر شد ، در این سیستم یک پارامتر دیگر تحت عنوان prefetch تعریف می شود به این معنا که تعدادی از بسته ها مشخصی از بسته را به یکی از سرویسی دهنده ها تحویل می شود و همانطور که بخش قبل گفته شد ، بسته ها به حالت transient در می آیند ، تا هنگامی که سرویس دهی این بسته ها تمام شده و مجدد یک تعداد جدید بسته را تقاضا کند ، این امر موجب میشود تعداد پیام هایی که برای ارسال دریافت بین سرور صف و سرویس دهنده منتقل میشود کاهش یابد و در نتیجه راندمان بالا تر رود. برای مثال به شکل زیر توجه کنید



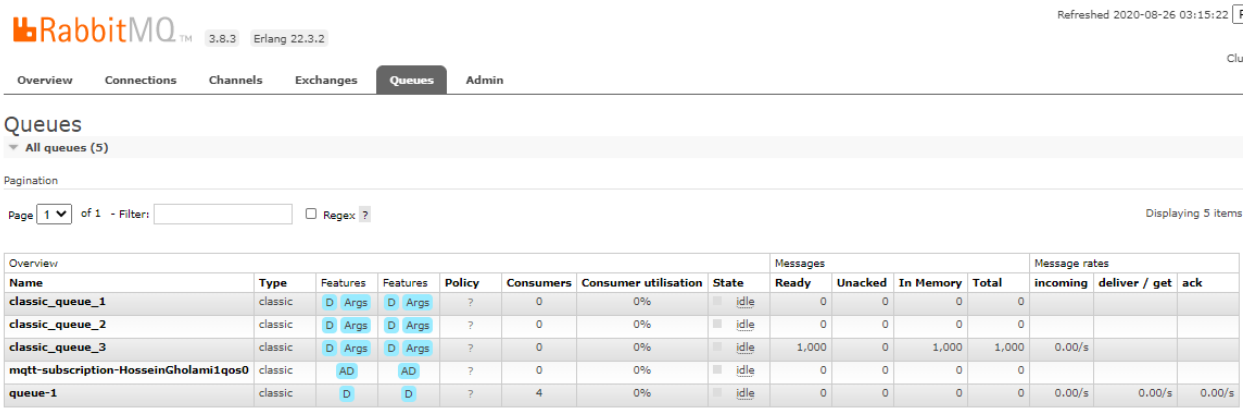
در این شکل دو سرویس دهنده یا مصرف کننده به این صف متصل شده اند و در صف ۵۰ بسته وجود دارد ، C1 با prefetch ۵ به این صف متصل شده و C2 که کامپیوتری با امکانات سخت افزاری قوی تری است ، با prefetch ۱۰ به این صف اتصال یافته ،حال در لحظه صفر c1 در خواست ۵ بسته ، و c2 درخواست ۱۰ بسته میکند ، در این حالت همچنان ۵۰ بسته در صف وجود دارد با این تفاوت که ۱۵ بسته به حالت transient در آمده اند ، و بعد از پردازش توسط c1,c2 اعلام تکمیل میکننده ، و بسته ها از صف حذف میشوند.

نتایج و تصاویر

محیط برنامه نوشته شده و کد های بخش های مختلف در ضمیمه آورده شد همچنین یک ویدیو از نحوه کار

کردن گرفته شد که در ضمیمه قرار گرفت ، ولی تصاویری از اجرای پروژه نیز در زیر مشاهده میکنید:

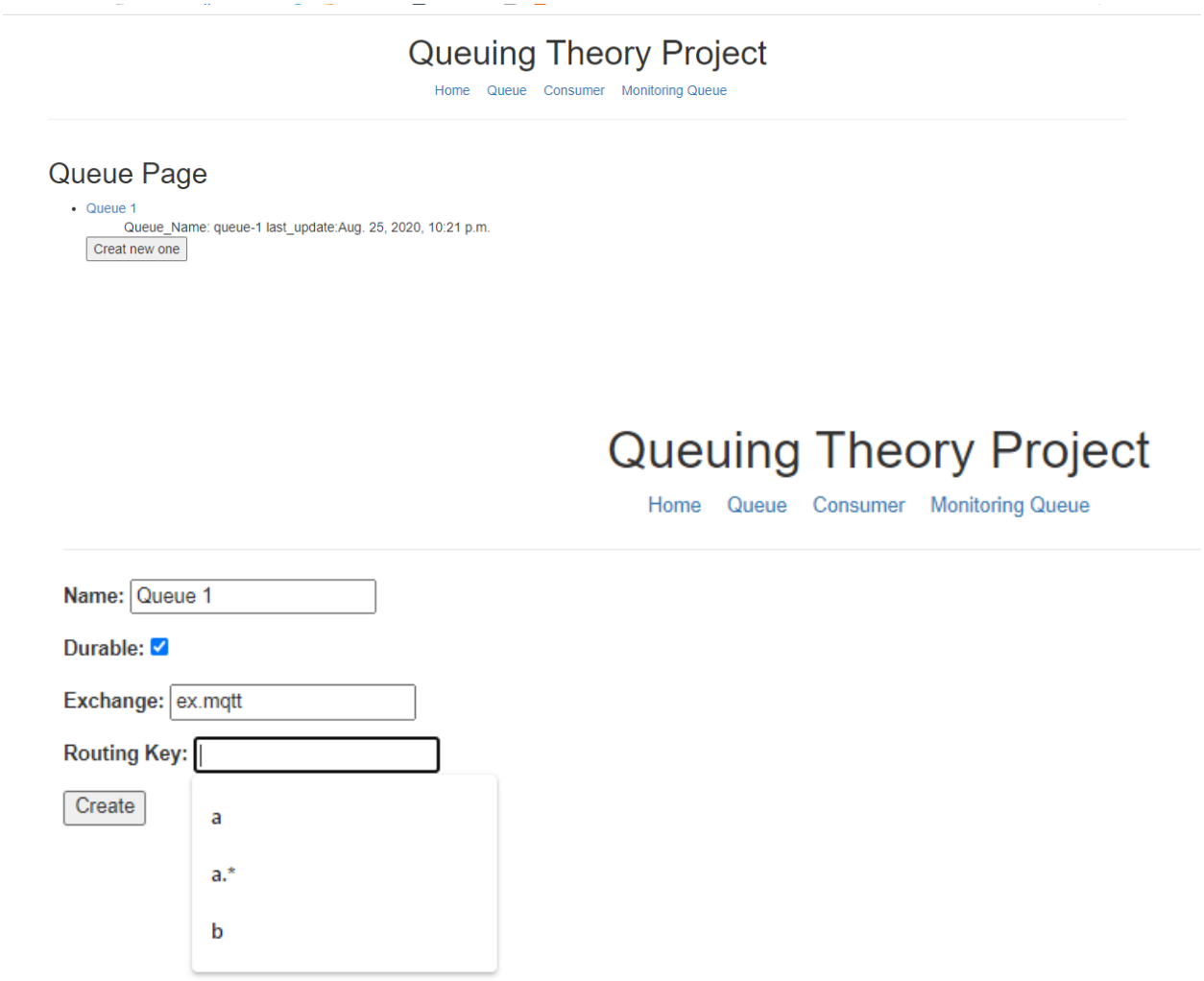
تصویر پنل ادمین Rabbit MQ



در این تصویر تعداد صف ها و برخی از جزئیات آن ها قابل مشاهده است

توضیحات بیشتر در ویدیو آمده است

تصویر داشبورد ایجاد و تنظیم صف :



خلاصه ای از اطلاعات صفی که در حال تبادل اطلاعات با ۳ سرویس دهنده است

monitoring Queue Page

queue-1

- Nuber of consumers : 3
- consumer_utilisation: 1.0
- durable: True

consumer_details

- consumer_tag : counsumer-1
- prefetch_count : 3

- consumer_tag : counsumer-2
- prefetch_count : 8

- consumer_tag : counsumer-2
- prefetch_count : 8

message_stats

- messages Number: 2
- messages_ready: 0
  - messages_ready_rate: 0.0
- messages_unacknowledged: 2
  - messages_unacknowledgedr_rate: 0.0

Queue_history

- Number of which acked: 1194
  - ack rate: 1.8
- Delivered message: 1197
  - Delivered rate: 1.8
- published message: 1151
  - publish rate: 1.8
- redelivered message: 1
  - redeliver rate: 0.0

تصویری از بخش ارسال اطلاعات :

TEST_SEND.py

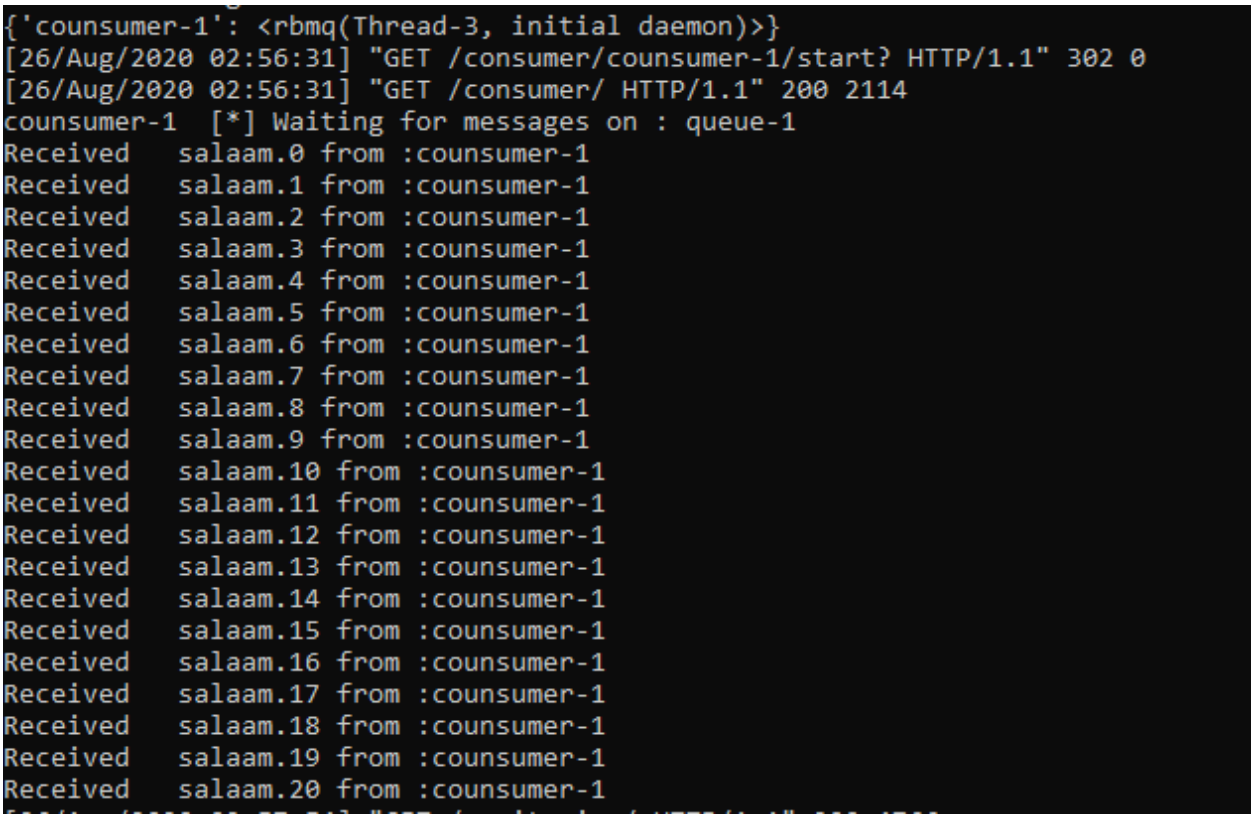
C: > Users > hosse > OneDrive > Desktop > interface > TEST_SEND.py > ...

```python
import pika
import time
credentials = pika.PlainCredentials('hgh', 'guest')
parameters = pika.ConnectionParameters('localhost',
                                       5672,
                                       '/',
                                       credentials)
for i in range (1000):
    connection = pika.BlockingConnection(parameters)
    channel = connection.channel()

    message = "salaam."+str(i)


    channel.basic_publish(exchange='ex.mqtt',
                          routing_key='a',
                          body=message,
                          properties=pika.BasicProperties(
                              delivery_mode = 2, # make message persistent
                              )
                          )


    print(" [x] Sent %r" % message)
    connection.close()
    time.sleep(0.5)
```

PROBLEMS OUTPUT TERMINAL DEBUG CONSOLE

```
[x] Sent 'salaam.319'
[x] Sent 'salaam.320'
[x] Sent 'salaam.321'
[x] Sent 'salaam.322'
[x] Sent 'salaam.323'
[x] Sent 'salaam.324'
[x] Sent 'salaam.325'
[x] Sent 'salaam.326'
[x] Sent 'salaam.327'
[x] Sent 'salaam.328'
[x] Sent 'salaam.329'
[x] Sent 'salaam.330'
[x] Sent 'salaam.331'
[x] Sent 'salaam.332'
[x] Sent 'salaam.333'
[x] Sent 'salaam.334'
```

ترمینالی از اطلاعات دریافت شده:



منابع:

S. C. Mukhopadhyay, Internet of Things Challenges and Opportunities, Massey University,New Zealand: Springer.

E. Pietrosemoli, "Wireless standards for IoT," in *International Centre for Theoretical Physics*, italy.

D. Gilliam, Introduction to Mechatronics,Temperature Sensors, 2003.

ACS712, Worcester, Massachusetts, U.S.A: Allegro MicroSystems, Inc.

l. ada, DHTxx Sensors, Adafruit Industries.

E. International Business Machines Corporation (IBM), MQ Telemetry Transport (MQTT) V3.1 Protocol, MQTT.org, 2010.

همچنین برای بخش پیاده سازی documentation های سایت های :

https://www.rabbitmq.com/documentation.html

https://docs.docker.com/

https://docs.djangoproject.com/en/3.1/

نامه ای به استاد :

سلام جناب دکتر با تشکر از زحمات شما در طول ترم

اقـای دکتـر ایـن درس آخـرین درس مـن در دوره کارشناسـی ارشـد هسـت و بـرای اینکـه بتـوانم از مزایـای معـدل جهـت سـربازی(لینـک خبـر از سـایت نظـام وظیفـه: اینجـا کلیـک کنید) اسـتفاده کـنم نیـاز بـه معـدل بـالای ۱۷ دارم در حـال حاضـر بـا ۲۳ واحـد پـاس شـده ، معـدل ۱۶.۷۷ دارم ، و بـرای رسـیدن بـه معـدل فـوق بـه حـداقل نمـره ی ۱۸.۷۷ نیـاز دارم ، امیدوارم با توجه به پـروژه انجـام شـده ، و اینکـه کـاملا پیـاده سـازی اسـت و نـه شـبیه سـازی، فعالیت کلاسی در طول ترم ، تمرین ها ، ... مساعدت فرمایید.