بسمه تعالی

درس معماری نرم افزار

گزارش پروژه

Push Notification SDK

دانشجویان:

ملیکا مسیبی

امیرحسین مسیبی

علی علمداری

حسین ناصری زاده

استاد:

دکتر آشتیانی

بهمن ۱۴۰۲

Feature list:

معماری push notification SDK دارای سه feature اصلی می باشد که برای آن تعریف شده است به نام‌های:

Server broker

Notification sdk

Client Application

ماژولی که در پروژه طراحی شده است، notification SDK می باشد که این ماژول بین Client Application که موبایل یا Device است و Server Broker قرار گرفته است. Server Broker ای که در شکل داریم قرار است برای ما Channel ها را نگه دارد و در واقع سروری است که از آن برای برای Push Notification استفاده می شود. همانطور که در تصویر نشان داده شده است، ماژولnotification SDK دارای زیر ماژول هایی به نام های subscribe،publish، SDK Manager و Identifyهستند. در ادامه به جزئیات معماری می-پردازیم.

معماری Push Notification SDK:

همانطور که در بخش قبل توضیح داده شد این معماری شامل سه بخش اصلی Server broker وNotification sdk و Client Application می باشد که فرض این هست که SDK قرار است از جایی سرویس بگیرد. اکنون SDk را می توان به سه بخش تقسیم کرد:

۱. ماژولی که subscribe انجام می دهد.

۲. ماژولی که Publish انجام داده.

۳. ماژول Identify که وظیفه احراز هویت را برعهده دارد. به بیان دیگر وظیفه این ماژول این است که به Client می گوید باید در ابتدا به ماژول Identify وصل شد و سپس احراز هویت شود که Client چه کسی است. درواقع Identifier، یک SDK است که باید بگوید از کدوم سرور، با چه مشخصاتی، با چه آدرسی و با چه configی دارد سرویس می گیرد. (همه چیز باید در ماژول identify مشخص شود و باید به Client اعلام نماید که از چه سروری، سرویس می گیرد و به چه سروری وصل است.). Client نیز یک سری Id, Device Id را به ماژول Identify داده و در نهایت خروجی این ماژول Identify یک session ای است که کاربر می تواند در آن session که توسط ماژول identify ایجاد شده، subscribe و publish کند و همینطور Push

Notification را بگیرد. بنابراین، خروجی یک سری فاز ها که شناسایی کلاینت و سرور هستند باید در identify انجام شود. منظور از بدون خروجی یعنی اینکه اگر session نداشته باشیم، بدون آن session نمی توان کاری انجام داد و عملا نباید کار کند چون اول باید گفته شود که Clientچه کسی است که بتوان از این سرویس ها استفاده کرد.

سیستم Push Notification تحت Channel انجام می شود که در این کانال یک سری topic وجود دارد که افراد آن پیام را subscribe می کنند. channel موجود در این سرویس دارای ID است.

فرض میکنیم در ابتدای کار، یک سری channel بایک سری id وجود دارد که افراد می توانند پیام ها راpush و یا subscribe کنند و کسانی که subscribe کرده اند اگه کسی پیامی را push کند، پیام Notif (اعلان) برای آن ها می آید. در channel هم پیام در حال رد و بدل شدن است و توسط یک سری افراد پیام subscribe شده در آن channel و هر پیامی که توسط publisher ها ارسال شده در channel، توسط subscriber ها دریافت شده و توسط client نشان داده می شود.

بعد از اینکه identify کاربر انجام شد و session ایجاد شد، کلاینت برای اینکه Push Notificaiton را دریافت کند نیاز داردکه channel ای را subscribe کنه و اگر subscribe نکند یعنی هیچ کدام از این تاپیک هایی که وجود دارد را نمی خواهد pushش را دریافت کند. پس قدم اول این است که کلاینت میخواهد channel ی را با id مشخص subscribe کند و بعد این درخواست برای سرور فرستاده شده و از طریق ماژول subscribe ای که داریم یک سری متد داخلش قرار دارد.

هر دو طرف یک ماژول encode, decode ای داریم که رمز نگاری را انجام می دهد. هر ارتباطی که به دو طرف خارج از ماژول SDK انجام شده، باید یک رمزنگاری روی آن انجام شود برای رفت و آمد پیام ها.

بنابراین subscribe فرستاده می شود و سرور متوجه شده که آن session ی که متعلق به user است با آن device id، آن رخداد را subscribe کرده است و سپس در دیتا بیس سمت سرور خودش این را نگه میدارد.

حال اگر یک publisher ای بیاید و در سمت خودش بخواهد publish کند، می گوید این channel id من و message من است، اکنون پیام را برای channel هایی که این channel id را دارند برایشان بفرست (برایشان push کن) . سپس این action را در ماژول publish خودش (پیامش را منتشر نموده) و سپس سمت سرور می رود و در اینجا دیگر سرور دست ما نیست و فرض میکنیم که publish را گرفته است و channel id را خودش map می کند بدین صورت که می گوید، این channel id وجود دارد و حال باید message برایشان

فرستاده شود. سپس برای خروجی، سرور که listener subscribe دارد، گوش داده و متوجه شده که در آن زمان یک نفر push کرده است و من پیامی را دریافت کردم و در اینجا پیام دریافت شده و decode و منتقل شده و سپس به کلاینت نشان داده می شود.

ارتباط subscribe هم می تواند دو طرفه باشد یعنی می تواند developer تنظیم کند که وقتی از یک SDK استفاده می کند، به محض اینکه applicaiton بالا آمد، subscribe را انجام دهد و دیگر نیازی به کلاینت نیست و یا می توان گفت که حتما دکمه باید زده شود که subscribe مثلا youtube مانند یک دکمه زده شده که subscribe انجام شود. درنهایت نیز sdk هم آن پیامی که از سرور گرفته است منتقل می کند به کلاینت و و کلاینت می تواند آن پیام را ببیند.

در کلاس دیاگرام، این موارد ها را داریم: ماژول SDK Manager در این کلاس نقطه ارتباطی با broker, client هست. کل مدیریت را SDK Manager برعهده دارد و با همه ماژول ها نیز در ارتباط است.

در قسمت class diagram بخش SDK Manager در خود تابع SDK Manager از هر ماژولی یک object گذاشته شده است که در این قسمت سه تابع به درد کلاینت خورده که وقتی sdk manager مارا نصب می کند، اولین چیزی که به آن داده می شود، یک object از کلاس sdk manager است.

اولین و مهم ترین کار SDK Manager این است که init کند. به بیان دیگر در syntaxش هم یک توکن داراست که از جنس string بوده و کلاس کلاینت را حذف کرده است زیرا بیشتر تمرکز این پروژه روی SDKاست. بنابراین token از broker گرفته شده که این token شناسنامه کلاینت است که توسط broker شناخته شده بوده و شناسایی شده که این توکن برای چه کسی است.

زمانی که تابع init را call نمود، یک سری کارها باید انجام شود و اصلی ترین کارهایی هم که انجام شده شامل connection با بروکر است. سپس connection یک securesession درست شده و مقداردهی در sdk manager می شود (یعنی init اول کارهای ماژول identify انجام داده که خروجی آن secure session می شود.). درنهایت securesession تولید و در sdk manager ذخیره شده و سپس اگر securesession امن بود آن را subscribe می کنیم.

بعد از صدا زدن تابع init می تواند کلاینت از طریق sdk با server ارتباط برقرار کند.

تابع هایی که دراختیارش قرار داده شده است:

+ subsrcibeTopics(topics: List<String> , callBackFn: CallBackFn) : void

+ publishToTopics(topics: List<String> , message: String) : void

+ getConnectedChannel() : Channel

+ unSubscribeTopics(topics : List<String>) : Channel

نکته دیگر در این پروژه در مورد فرق بین channelوTopic است که هر کلاینت وقتی به broker وصل شده، یک channel به آن اختصاص داده می شود و توی اون channel می تواند پیام ها رد و بدل شود یعنی یک کلاینت فقط بایک channel ارتباط برقرار کرده است. حال با استفاده از topic می توان موضوعات مختلف را از هم جدا کرد یعنی تاپیک های مختلف در همان channel آمده (تاپیک های مختلف را subscribe می کنیم).

channel فقط یک راه ارتباطی است که همه پیام‌هایی که رفت و آمد دارند، از این طریق بوده ولی تاپیک مشخص کرده که نوع پیام ها می تواند فرق کند و کلاینت می تواند هرچقدر تاپیک را که بخواهد subscribe کند و افراد دیگر هم که publish میکنند در تاپیک های مختلف میتوانند publish کنند یعنی می گویند هرکس تاپیک A را دارد پیام رو دریافت کند و بروکر میاد نگاه کرده که چه کسانی هستد که تاپیک A را subscribe کرده اند و بعد به channelشان پیام فرستاده می شود.

همچنین sdk manager نیاز دارد تا رمزنگاری و رمز گشایی را انجام دهد و زمانی رمزنگاری می کند که بخواهد publish یک چیزی را publishکند و رمزگشایی می کند که میخواهد پیام را از بروکر دریافت کند و به کلاینت برساند.

ماژول SubscribeModule در کلاس دیاگرام دارای یک متد داره به اسم subscribetopic است که یک لیستی از تاپیک ها گرفته و پارامتر دومش هم callback function است که گفته می شود هر وقت یک پیامی در هر کدام از این تاپیک ها آمد، این تابع را برای من صدا بزن پس در این callbackFn پیام و تاپیک وجود دارد یعنی با پیام و تاپیک صدا زده شده است.

در ماژول SubscribeModule وقتی پیام از طرف broker آمد میاد callbackFn صدا زده شده با استفاده از پیام و دیتایی که از بروکر آمده هست. همچنین در publish هم مثل subscribe همین کار را می توان انجام داد با این تفاوت که topic list را گرفته و کلاینت topics را صدا می زند. در پارامتر اول گفته می شود که می خواهیم این پارامتر ها رو پابلیش کنیم یعنی هر کسی این تاپیک ها را دارد پیام ما را دریافت کند. پارامتر دوم خود message است که string بوده و می گوید پیام را برایشان بفرست.