به نام خدا

پروژه ای که جهت کارآموزی در شرکت رایان آروین الگوریتم شروع کردم یک پیام رسان با تکنولوژی MQTT بود که باید بر معماری یک Restful api نوشته می شد تا بتوان از طریق اینترنت درخواست هایی به سمت بروکر MQTT فرستاد

این پیام رسان که برای استفاده از ماژول های MQTT توسعه داده شده از زبان برنامه نویسی Python و میکرو فریموورک FastAPI استفاده میکند.

تا به وسیله آنها بتواند محیط پرسرعتی به همراه تجربه کاربری متفاوت و راحت تر بر بستر یک Rest-ful API به کاربران ارائه دهد.

در واقع این ماژول کمک میکند که همزمان با ران بودن لوپ سرور و کلاینت پیام های شما به مقصد مخابره شود و به کمک تکنولوژی `Async/Await`

اتصال به سرور را بهینه تر و پیام رسانی را سریع تر میکند تا به هنگام پیام رسانی تعداد خطاهای کمتری تجربه بکنیم.

* چرا FastAPI ؟

ابتدا باید بررسی کنیم ما از این پروژه ای که بر پایه Rest-ful میخواهیم توسعه دهیم به چه چیز هایی نیاز دارد یا ما از آن چه چیزی میخواهیم ؟

* سرعت

اولین و مهم ترین عاملی که ما باید برای این پروژه در نظر بگیریم سرعت بازگردانی Response در این پروژه میباشد، به این صورت که این نرم افزار با توجه به استفاده از ماژول MQTT که سرعت بسیار زیادی برای پیامرسانی دارد، نیاز به یک فریموورک پر سرعت در پاسخگویی ها داشتیم که FastAPI در فراخوانی های مرتبط به پروژه های کوچک سرعت زیادی دارد و از این رو انتخاب ما برای توسعه Fast-MQTT شد.

* حجم بسیار کم

یک از عوامل مهم در بهینه بودن نرم افزار کم حجم بودن آن نسبت به کاری که انجام میدهد می باشد، با توجه به اینکه تکنولوژی FastAPIحجم بسیار کمی را در برمیگیرد بهترین گزینه برای توسعه این میکروماژول به حساب می آمد.

* ماژولار بودن

با توجه به اینکه خود میکروفریموورک FastAPI از معماری خاصی پیروی نمیکند میتوانست بهترین گزینه برای توسعه دهنده ای باشد که با توجه به ماژول های مورد استفاده در نرم افزار معماری خود را انتخاب میکند؛ همچنین این موضوع برای این میکروفریموورک باعث استفاده از دیزاین پترن های متفاوتی برای نرم افزار می شود.

* Paho-Mqtt

Paho-MQTT یک کتابخانه نرم‌افزاری متن‌باز و پرکاربرد است که برای پیاده‌سازی پروتکل MQTT در زبان‌های برنامه‌نویسی مختلف از جمله پایتون طراحی شده است. MQTT یک پروتکل سبک‌وزن و منتشر-مشترک است که به طور گسترده در اینترنت اشیا (IoT) و سیستم‌های تعبیه‌شده برای ارتباط بین دستگاه‌ها استفاده می‌شود.

* ویژگی‌های کلیدی Paho-MQTT:
  + پشتیبانی از نسخه‌های مختلف MQTT: Paho-MQTT از نسخه‌های مختلف پروتکل MQTT از جمله MQTT 5.0، 3.1.1 و 3.1 پشتیبانی می‌کند.
  + سادگی استفاده: این کتابخانه رابط کاربری ساده و روانی دارد که به توسعه‌دهندگان اجازه می‌دهد به‌راحتی به یک بروکر MQTT متصل شوند، پیام‌ها را منتشر کنند و به موضوعات مختلف مشترک شوند.
  + قابلیت اطمینان: Paho-MQTT مکانیزم‌های مختلفی برای اطمینان از قابلیت اطمینان ارتباطات، مانند مکانیزم‌های بازاتصال خودکار و تایید دریافت پیام‌ها، ارائه می‌دهد.
  + پلتفرم‌های پشتیبانی شده: این کتابخانه برای پلتفرم‌های مختلف از جمله ویندوز، لینوکس و macOS قابل استفاده است.
  + جامعه بزرگ: Paho-MQTT یک پروژه متن‌باز با یک جامعه بزرگ از توسعه‌دهندگان است که به این معنی است که شما می‌توانید به راحتی به منابع و پشتیبانی دسترسی پیدا کنید.

و در نهایت با توجه به این ویژگی ها نرم افزار مورد نظر را تولید کردم !  
  
لینک گیت هاب این پروژه : <https://github.com/alirezazainy/fast-mqtt>

لینک مستندات : <https://alirezazainy.github.io/fast-mqtt>

از روزی که در شرکت مستقر شدم شروع کردم به بروزرسانی دانش برنامه نویسی خودم، بعد از اتمام این برنامه سرپرست کارآموزی یک پروژه به من محول کرد که انجام بدهم. این پروژه یک API است که به یک سرور MQTT متصل میشود و میتواند با دستورات API به آن پیام بدهد و نتیجه را دریافت کند.   
به منظور اینکه بفهمم باید چه کاری بکنم ابتدا رفتم سراغ مستندات MQTT و آقای مطهری که خودشان هم اطلاعات کاملی در مورد MQTT داشتند، پس از اینکه فهمیدم ساز و کار این تکنولوژی به چه صورت است، رفتم سراع اینکه ببینم با دانش فنی من، چگونه میتوان یک نرم افزار به این شکل توسعه داد.   
برای اینکه بتوانم به یک سرور MQTT متصل بشوم خیلی جست و جو کردم تا به یک کتابخانه مطلوب به نام paho-mqtt رسیدم. یک کد آماده برای subscriber و یک کد اماده دیگر هم برای publisher پیدا کردم و با توجه به این ها که از زبان پایتون استفاده میکردند یک ماژول که بتوانم آن را از طریق FastAPI فراخوانی و استفاده کنم طراحی کردم.  
معماری اصلی برنامه و کد ها را منظم طراحی کردم و کاملا کامنت گذاری کردم و در نهایت به کمک زبان markdown و تکنولوژی material for mkdocs یک دوره مستندات فنی از همین ورژن هم نوشتم.   
همچنین برای بارگذاری روی سرور هم یک Dockerfile نوشتم و آن را با تمام مستندات و متعلقات از ابتدای پروژه روی github بارگذاری کردم.  
سپس تست در فضای حقیقی آن را برای یک دستگاه به صورت عملی گرفتم تا ببینم برنامه درست کار میکند.