前回のレッスンでは、ストリームと呼ばれるものに実際に出会ったのは初めてでした。そして、メソッドが再度呼び出される場所でそれを使用することの効果を見ることができます-メッセージコレクションに追加される新しいデータを取得するたびに。そして、これはこのコレクションにサブスクライブしているようになり、新しいデータが入るたびにアプリで通知されます。それで、私たちが使用しているこれらのDartストリームは正確には何ですか？

それでは、類推から始めましょう。

そして、私を知って、それは食べ物の類推になるでしょう。

今、あなたが日本のレストランにいて、本当に寿司を食べたいと思っていると想像してみましょう。

もちろん、シェフが料理を準備するには時間がかかります。

魚を切っておにぎりなどを作らなければなりません。少し待たなければなりません。

今、この権利を待つには2つの方法がありますか？

一つは、「ここに私が食べたいものがすべてある」と言うことができるということです。

そして注文すると、領収書がもらえます。

これは注文番号です。

準備ができたら、収集できます。

そしてこれが先物の仕組みです。

そして、私たちはすでにそれを見て、Climaモジュールで詳細に説明しました。

しかし、別のシナリオを検討してください。

「一度にすべての注文が欲しい」と言う代わりに、一度にすべての注文をお願いします。

代わりにテーブルに席があったらどうしますか？

だから、あなたが回転寿司屋にいて、シェフがこれらの料理を作ると想像しましょう。

そして、彼がそれらを作ると、彼らはコンベヤーベルトの下に送られ、コンベヤーベルトの下流のどこかに座っている人は、彼がそれを作っているときにこれらの寿司を拾うことができます。

これはもう少しストリームのようなものです。繰り返しますが、あなたが待っているイベントがあります。つまり、シェフが寿司の準備を終えなければならないということです。しかし、彼は寿司の準備を終えるとすぐに、それらの寿司を送り出します。

したがって、このストリームにサブスクライブするか、このコンベアベルト寿司テーブルに座ることで、データが入ってくるときやストリームに追加されたときにデータを取得することができます。

このように、シェフが寿司を作るのにかかる時間に応じて、あなたが待っているイベントがまだあります。しかし、すべてを一度に取得し、最終的な注文を家に持ち帰るのを待つのではなく、ここで、それが通過するときにそれを取得するようにします。

それが私たちのメッセージがやってきた方法です。メッセージのストリームにサブスクライブしており、そのコードはスナップショットメソッド内にあります。

そのメソッドの定義の内部を見ると、クエリスナップショットのストリームを返すメソッドを作成していることがわかります。

そして、このコレクションにさらにデータが追加されると、データはメッセージストリームを介して受信され、最新のデータを印刷するか、メッセージコレクション内にあるようにすべてのデータを印刷するかを聞くことができます。 。

これについてのもう1つの考え方は、これを2 x 2のマトリックスと考えた場合です。

したがって、すでに利用可能なものがあるとしましょう。つまり、通常のオブジェクト、int、double、stringなどです。

そして、あなたが待つ必要があるものがあります。そして、これらのものの単一バージョンが欲しかったとしましょう。

既に利用可能なものの単一バージョンは、単なる文字列オブジェクトであるデータの一部であるかもしれません。

変数または定数のように。

しかし、待機する必要のある単一のオブジェクトがある場合はどうでしょうか？

それは将来のStringオブジェクトかもしれません。

したがって、これは非同期に処理されるものです。

インターネットから取得するのに時間がかかる場合がありますが、いったん到着すると、通常の文字列として使用できます。

しかしその間、エンコードするときと計画しているとき、データ型は将来の文字列です。

しかし、複数のアイテムや複数のアイテムはどうですか？

簡単に入手できる文字列の複数バージョンは、文字列のリストになりますよね？

多くの文字列を含むリストになります。

しかし、この文字列のリストが届くまで待たなければならない場合はどうでしょうか？

将来の文字列の複数形とは何ですか？

まあ、それは、いつでもストリームに追加できるコンベアベルトに追加されている寿司のように、文字列のストリームです。

そして、そのストリームにサブスクライブしている限り、パイプラインまたはストリームに新しいデータが追加されるたびに通知されます。食べ物に関連するアナロジーが十分にない場合は、長い間そうめんと呼ばれるこの素晴らしいものがあります。これは、日本の人々が夏に竹を割って、小さな川のように水を流すということです。 。そして、誰かが上流にいて、通常、麺をストリームに流し込み、それらはストリームの中に流れ込みます。

そして、ストリームを購読している人は、うどんが流れ落ちるのをキャッチして、ボウルに加えることができます。したがって、彼らはストリームを効果的に使用し、通過するデータを聴いています。

今、データがどれほど定期的に送られるかわかりませんか？

それは毎秒である可能性があります、それは1秒であり、そしておそらく5分後です

ただし、サブスクライブしている限り、データがいつ送信されるかを通知されます。

ここでも同様に、ユーザーがメッセージを入力する割合が不規則であることがわかっています。

今日はお母さんにメッセージを送るかもしれませんが、1か月間は彼女にメッセージを送りません。

文書を取得するようなものを使用してデータをプルし続けることは意味がありません。その場合、タイマーに置く必要があるかもしれません。

今すぐメッセージを確認する必要がありますか？彼らのメッセージは？彼らのメッセージは？そして、あなたはたぶんそれを毎秒、あるいは本当に非常に集中的な一秒のほんの一部をする必要があります。

しかし、代わりに、ストリームがあった場合、基本的には新しいデータがあるとすぐに通知され、データを処理することができます（つまり、アプリに追加することで）。ですから、ストリームをもう少しクリアするか、少なくともここで何が起こっているのかをもう少し伝えてほしいと思います。

そして、コードをいじり終わったら、次のレッスンに進みます。ここでは、データストリームを画面に表示できる実際のウィジェットに変換する方法について説明します。