天気アプリを開始するには、コースリソースにリストされているGitHubリポジトリにアクセスしてください。次に、以前のアプリで行ったように、リポジトリをクローンし、Android Studioからプロジェクトを開きます。

最後に、プロジェクトを開いた後、依存関係の取得をクリックして、すべてのエラーがなくなり、プロジェクトが正常にビルドされるようにしてください。

次に、libフォルダー内にいくつかのフォルダーがあることを確認します。それで、私はすべての異なるファイルを異なるセクションに分類しました。

さまざまな画面があり、そのうち3つの画面があります：読み込み画面、場所画面、都市名に基づいて天気を取得する画面です。

そして、location.dartやネットワーキングの部分、天気の部分などのサービスがあります。そして、私たちはついに定数ファイルだけを持っているユーティリティをいくつか手に入れました。そして、すべてがmain.dartファイルから始まります。

だから、先に進み、私たちがあなたのためにすでに作成したデザインを見てください。

しかし、基本的にUIは非常にシンプルに保たれます。これは、多くのバックエンド処理とネットワーキングを行うからです。

そのため、ユーザーインターフェイスを比較的理解しやすくしました。ロケーション画面である最も複雑な画面でも、読み進めるだけで非常に簡単になります。

それは、いくつかのテキストウィジェットだけでなく多くのボタンを含む列を内部に持つ背景画像を備えた単純なコンテナです。

これまでのところ、これは非常に単純なものです。アプリの読み込み時に最初に発生することは、最初のルートとして読み込み画面を読み込むことです。

したがって、ロード画面内では、位置を取得するという1つの上げられたボタンを含むステートフルウィジェット以外に、ここにはほとんど何もないことがわかります。

したがって、これをそのまま実行すると、今のところ非常にエキサイティングなことは何もありません。1つのボタンが付いた黒い画面です。しかし、ここからアプリが起動します。最初に行うことは、このボタンをタップして場所を取得するときにユーザーの場所を取得できるようにすることです。

これを行うために、Flutterパッケージのパワーを活用します。また、iOSとAndroidの両方から位置情報サービスを取得できるようにするために使用するgeolocatorというパッケージがあります。そのため、ドキュメントで説明されているこのようなコードを1行呼び出す必要があり、パッケージは舞台裏で実行され、コードが実行されているかどうかにかかわらず、場所を取得するために必要なすべてのことを行いますiOSまたはAndroidで実行します。

これを依存関係として追加して始めましょう。

これをコピーしてpubspec.yamlに移動し、cupertino\_iconsのすぐ下にジオロケーターを追加して、Cupertinoアイコンと同じレベルにフォーマットされていることを確認します。

そして、それが文字列として入ってくる瞬間に、私はそれを削除するつもりです。

保持するかどうかは問題ではありません。

同じように解釈されます。

しかし、私は物事に一貫性を持たせるのが好きです。

その依存関係を追加したので、パッケージをクリックして実際にプロジェクトにインポートします。

そして、それが完了すると、外部ライブラリとFlutterプラグイン内で確認できるはずです。

そのため、ジオロケーターをすぐに見ることができます。

これで実際に使用できるようになりましたので、今はロード画面内で使用します。

したがって、これらのパッケージを使用するための最初のステップは、もちろんそれをインポートすることです。

そのため、一番上でジオロケーターをインポートします。

そこにはgeolocator.dartがあります。

そして、今度は何も返さない新しいメソッドを作成しますが、実行するのは場所を取得するだけなので、場所を取得します。そして、このメソッドの中括弧内で、現在のユーザーの位置を取得するために必要なすべてのことを行います。

したがって、ドキュメントを見ると、デバイスの現在の場所を照会するように指示されます。

getCurrentPositionメソッドを呼び出すだけです。

そして、これがどのように行われるかを示しています。

したがって、データ型の位置であり、これは位置と呼ばれるものです。そして、awaitというキーワードがあり、ジオロケーターを使用して現在の位置を取得します。

そして、入力として、その場所をどのくらい正確にしたいかを指定します。

それでは、このコード行をプロジェクトにコピーして、作成したgetLocationという新しいメソッドの中に貼り付けましょう。

これを実際に機能させるために最初にしなければならないことは、この関数に修飾子キーワードを追加することです。これは非同期という言葉になります。

さて、この非同期と待機は正確には何ですか？

これは、電話からGPS位置情報を取得するなど、時間のかかるタスクを実行できる方法です。電話機に現在の場所を尋ね、それを取得して処理し、最終的にその場所にアクセスできるようになるには少し時間がかかることを想像できます。

また、インターネットからデータをダウンロードしようとしたり、ファイルから何かを読み取ろうとするなど、他にもあります。

これらはすべて、予測できない時間がかかる可能性がありますが、非常に長い時間がかかる場合もあります。非同期プログラミングを使用することで、これらの時間のかかるタスクをフォアグラウンドで実行し、UIをブロックしてアプリをフリーズする代わりに、バックグラウンドで実行することができます。

次のレッスン全体は、Dartでの非同期プログラミングに専念します。非同期待機の未来とそのすべてについて詳細に説明します。

ただし、今のところは、まず現在の場所を取得してみましょう。

したがって、geolocatatorパッケージに由来するGeolocatorと呼ばれるこのオブジェクトがあります。これを使用して、必要な精度を提供することで現在の位置を取得します。

そして、これを高から低に変更します。そして、ナビゲーションに最適な最低、中、最低、最高のさまざまなオプションがたくさんあることがわかります。覚えておくべきことは、取得しようとしている場所が正確であればあるほど、バッテリーが集中することですあります。

そのため、ユーザーとその携帯電話のバッテリーに優しくしたい場合は、必要な精度だけを使用してください。

したがって、人々が実際にターンバイターンの方向を必要とするナビゲーションアプリを作成している場合、おそらくナビゲーションに最適なような非常に正確なものを選択する必要があります。

しかし、天気だけを取得しようとしている場合は、iOAで最も近いキロメートル、Androidで最も近い500メートルの精度を提供する低さのようなもので十分でしょう。そして、最低のものでもおそらく大丈夫ですが、どちらの方法でもこれらの精度レベルのいずれかを選択してください。

そして今、私たちは自分の位置を取り戻すことができるはずです。そして、位置を取り戻したら、コンソールにそれを印刷して、GPSに従って現在の位置を確認します。これが機能するために必要なもう1つのステップがあります。それは、ユーザーに現在の場所を取得する許可を求めることです。危険な。そしてiOSと

Androidでは、位置データを明示的に要求する必要があります。

それでは、どうすればそれができるかを見てみましょう。 geolocatorのドキュメントで、少し下にスクロールすると、Androidでアクセス許可を取得するには、Androidのマニフェストにアクセスの粗い場所またはアクセスの細かい場所のいずれかを追加する必要があることがわかります。低レベルの精度の位置データのみを使用するため、粗い位置を取得するための許可のみが必要です。

では、この特定のタグのコピーを押して、ここでAndroidプロジェクトフォルダーに移動してから、アプリフォルダー、ソース、メインに移動して、AndroidManifest.xmlファイルを見つけます。

これをダブルクリックして開くと、これらがAndroidプロジェクト用に設定したさまざまな設定であることがわかります。

そして、マニフェストタグの直接の子として、次の2行のいずれかを追加する必要があることがわかります。

それはどういう意味ですか？

これはマニフェストタグであり、この特定のファイルの最高レベルのタグです。

その中のすべてはその子ですが、アプリケーションタグ、アクティビティタグ、メタデータタグなどの下位階層もあります。

また、HTMLを使用したことがある場合、このフォーマットは拡張マークアップ言語を表すXMLでスタイル設定されているため、非常に馴染みのあるものになります。

キーはこれらのタグ内にあり、値はタグ内に保持されるというキー値ペアと本質的に同じです。

そのため、マニフェストタグ内で、閉じ山括弧の直後に、ユーザーのアクセス許可タグを貼り付けます。

これは、Androidアプリに、大まかな場所にアクセスする必要があり、ユーザーから要求する必要があることを伝えます。

これでAndroidの部分が完了したので、Androidフォルダーを閉じることができます。

そして、iOSユーザーがアプリを使用するときに、iOSで許可を求める必要があります。そして、私たちがしなければならないことは、キーと文字列の2行をコピーすることです。これは、ユーザーに場所を使用する必要がある理由を伝えます。

そのため、必要に応じてこのテキストを変更できます。実際にデータを要求しているときに、ユーザーによりわかりやすいものにしたい場合は。

しかし、これらの2行をコピーして、info.plistファイルに追加します。

したがって、iOSフォルダーに移動する必要があります。ランナーフォルダー内には、iOSアプリに特に関連するすべてのファイルがあります。

そして、ここでinfo.plistというファイルを見つける必要があります。

それをダブルクリックすると、これは再びXMLファイルであることがわかります。このファイルのキーと文字列を、dictまたは辞書と呼ばれる場所のすぐ下に追加します。

したがって、これらの2行を入力し、キーと値のペアの辞書を作成します。

これにより、iOSアプリに、ユーザーの場所を要求するときに、これがユーザーに場所を必要とする理由を伝えるために提供する説明であることがわかります。

では、保存をクリックします。

これらの2つのファイルを閉じましょう。

そして最後にやらなければならないのは、実際にこのメソッドを呼び出して位置を取得することです。そして、ユーザーが位置を取得するというボタンを押したときに呼び出すので、onPressedで実行します。そして、ここでget locationを呼び出して、アプリをテストする準備ができました。

そして、テストする前に、依存関係を追加し、AndroidマニフェストとiOS plistをいじったため、現在のアプリの実行を停止してからコールドから起動することをお勧めします。

通常は、一度それをゼロから開始することをお勧めします。少し時間がかかりますが、舞台裏でリンクしなければならないことがたくさんあります。

このボタンをクリックして位置を取得するときにアプリが稼働しているので、今作成した位置取得メソッドを呼び出して、現在の位置を取得してコンソールに印刷する必要があります。

しかし、ボタンをクリックしたときに最初に行うことは、現在の場所を与える許可を求めることです。

そのため、このステップで[許可]をクリックしてください。これで、緯度と経度がコンソールに出力されます。その場所はどこから来たのでしょうか？

シミュレータを使用している場合、それは通常、アップル本社にあるデフォルトの場所、つまりクパチーノのどこかにあります。 Androidでアプリを実行している場合は、ここの右下にあるこれらの3つのドットをクリックできます。

これにより、拡張コントロールが表示され、[場所]タブで選択して、この携帯電話から報告する緯度と経度を設定できます。

位置データを手動で設定してアプリに送信できます。

ここで、Androidでスマートフォンを実行しているときにAndroid Xに関する何かに関連するエラーがコンソールに表示される場合は、このリソースへのリンクがあるコースリソースに移動してください。アプリのAndroidX互換性を実現する方法についてはFlutterのドキュメントをご覧ください。

今、この問題は誰にも影響しません。

アプリが実際にクラッシュし、ビルドが拒否された場合のみです。

そのため、現在のGPS位置を実際に取得するのは簡単です。

そして、あなたがそれについて考えるならば、舞台裏で多くのことが起こらなければなりませんでした。

iOSの対応者は、そのデバイスから位置を取得できるようにiOSオペレーティングシステムと通信する必要があり、Androidの部分はAndroid OSと通信し、まったく異なる方法で位置を取得する必要がありました。

しかし、すべてのことを処理してくれたgeolocatorパッケージを使用するだけで、それを行うことができました。そして、単にプロジェクトに追加する以外に何もする必要はありませんでした。

ジオロケーターがこれをどのように管理しているかに興味がある場合は、外部ライブラリフォルダーに移動し、Flutterプラグインに移動してジオロケーターライブラリを展開できます。

ここで、iOSおよびAndroidプロジェクトが表示されます。

iOSプロジェクトを展開してクラスに入ると、iOSロケーションAPIと連携してロケーションデータを取得して戻ることができるように、Swiftで記述されたファイルがここにたくさんあることがわかります。それをあなたに。

Androidフォルダー内を見て、src.mainに移動してからJavaフォルダーに移動すると、データフォルダーとタスクおよびutilsフォルダーの両方が存在します。

そして、Androidロケーションマネージャーを処理し、そのロケーションデータを取得できるように作成されたすべてのJavaコードがここにあります。

今、あなたはこのJavaコードのすべてを書き、このSwiftコードのすべてが多くの専門知識を要するだけでなく、多くの時間もかかると想像できます。しかし、このプラグインの作成者は非常に親切なプラグインを非常に使いやすいプラグインにパッケージ化しているため、インポートしてコードを1行だけ呼び出すだけで、すべての困難な作業に便乗することができます。

また、他の人が役立つと思うカスタム機能を構築するときに、他のFlutter開発者と共有して支払うことができるように、その周りにパッケージを作成することもできます。

しかし、今のところ、これまで見たことのないこれら2つのキーワードasyncとwaitに焦点を当てたいと思います。これを理解するには、同期および非同期プログラミングについてすべて学ぶ必要があります。

したがって、このコードで実際に何が起こっているのかを理解する必要があります。

次のレッスンに進んで、非同期プログラミングについてすべて学びましょう。