Dokumentation Portfolioapp

Die vorliegende Dokumentation erläutert die Funktionsweise der Navigation in der entwickelten Applikation und beschreibt die zur Erstellung der verschiedenen Seiten verwendeten Widgets. Diese App wurde mit Flutter, einem Open-Source-Framework von Google, entwickelt, das zur Erstellung nativ kompilierter Anwendungen für mobile Endgeräte, Web und Desktop verwendet wird.

1. Navigation in der Applikation

Die Navigation innerhalb der Applikation erfolgt mittels des Flutter-Navigators. Der Navigator ist ein Widget, das eine Methode zur Steuerung des Stapels von Routen bereitstellt, die für die Navigation zwischen verschiedenen Bildschirmen verwendet werden. In unserer App nutzen wir das Konzept der benannten Routen, das es uns ermöglicht, Routen im Voraus zu definieren und sie bei Bedarf aufzurufen.

Beim Start der Applikation wird die `main.dart`-Datei ausgeführt, die das Grundgerüst der Applikation enthält. Innerhalb dieser Datei wird die Hauptklasse `MyApp` definiert, welche das `MaterialApp`-Widget zurückgibt. Das `MaterialApp`-Widget stellt die grundlegende Struktur und Designprinzipien der App bereit und ermöglicht die Definition von benannten Routen. Diese Routen verweisen auf die verschiedenen Seiten der Applikation, wie die Startseite, die Über-mich-Seite, die Fähigkeiten-Seite, die Projekte-Seite, die Kontakt-Seite und die Lebenslauf-Seite.

Die benannten Routen werden in einer Map innerhalb des `MaterialApp`-Widgets definiert. Jede Route wird einem spezifischen Widget zugewiesen, das den Inhalt der jeweiligen Seite darstellt. Zum Beispiel wird die Route `'/about'` dem `AboutPage`-Widget zugewiesen. Wenn der Benutzer auf einen Navigationslink klickt oder eine spezifische Aktion ausführt, wird der Navigator verwendet, um die entsprechende Route zu laden und das damit verknüpfte Widget anzuzeigen.

1. Widgets zur Erstellung der Seiten

MaterialApp und Scaffold

Das `MaterialApp`-Widget ist das Herzstück jeder Flutter-Applikation, die auf dem Material Design basiert. Es ist verantwortlich für das grundlegende Layout und die Navigation der App. Innerhalb des `MaterialApp`-Widgets werden die globalen Einstellungen, wie das Thema der App und die benannten Routen, definiert.

Jede Seite in der Applikation wird durch das `Scaffold`-Widget strukturiert. Das `Scaffold`-Widget bietet eine Grundstruktur mit AppBar, Body und anderen grundlegenden Layout-Elementen. Es ermöglicht die einfache Integration von App-Leisten, Navigationsschubladen und anderen wichtigen UI-Komponenten.

1. AppBar und Drawer

Die `AppBar` ist ein Bestandteil des `Scaffold`-Widgets und wird verwendet, um eine Anwendung-Leiste am oberen Rand der Seite zu erstellen. Sie enthält den Titel der Seite und kann weitere Aktionen, wie Menüoptionen oder Suchfunktionen, beinhalten. In unserer Applikation wird die `AppBar` dazu verwendet, den Seitentitel anzuzeigen und ein konsistentes Erscheinungsbild zu gewährleisten.

Der `Drawer` ist ein weiteres Widget des `Scaffold`, das eine seitliche Navigationsleiste darstellt. Durch das Antippen oder Wischen von links nach rechts öffnet sich der `Drawer`, der Links zu den verschiedenen Seiten der Applikation enthält. Dies ermöglicht eine intuitive und einfache Navigation innerhalb der App.

1. Container und BoxDecoration

Um das visuelle Erscheinungsbild der Seiten zu gestalten, verwenden wir das `Container`-Widget in Kombination mit `BoxDecoration`. Das `Container`-Widget bietet ein Rechteck, das andere Widgets aufnehmen kann und deren Ausrichtung, Größe und Ränder steuert. In unserer App wird es verwendet, um den Hintergrund der Seiten zu gestalten und eine konsistente Farbgebung zu gewährleisten.

Die `BoxDecoration` ermöglicht die Anpassung des Aussehens eines Containers durch Hintergrundfarben, -bilder, Ränder und Schatten. In der Applikation wird `BoxDecoration` verwendet, um Hintergründe zu setzen und visuelle Akzente zu schaffen.

1. SingleChildScrollView und Column

Um sicherzustellen, dass der Inhalt der Seiten scrollbar ist und nicht über den Bildschirm hinausgeht, verwenden wir das `SingleChildScrollView`-Widget. Dieses Widget ermöglicht das Scrollen einer einzigen Kindkomponente, wodurch der gesamte Inhalt zugänglich bleibt, auch wenn dieser die Bildschirmgröße überschreitet.

Innerhalb des `SingleChildScrollView`-Widgets verwenden wir das `Column`-Widget, um die Anordnung der Kinder-Widgets vertikal zu strukturieren. Das `Column`-Widget ermöglicht die Ausrichtung und Platzierung der Elemente in einer Spalte, was eine klare und geordnete Darstellung des Inhalts gewährleistet.

1. Card und Padding

Für die Darstellung von Projekten und anderen spezifischen Inhalten verwenden wir das `Card`-Widget. Das `Card`-Widget bietet eine Material-Design-Karte, die visuell hervorgehoben wird und Informationen auf strukturierte Weise präsentiert. Es wird in Kombination mit dem `Padding`-Widget verwendet, um den Innenabstand der Karteninhalte zu steuern und eine klare Trennung zwischen den einzelnen Elementen zu schaffen.

1. Text und TextStyle

Zur Darstellung von Textinhalten verwenden wir das `Text`-Widget. Um die Lesbarkeit und das Erscheinungsbild des Textes zu verbessern, wird das `TextStyle`-Widget eingesetzt. Es ermöglicht die Anpassung von Schriftgröße, -farbe, -gewicht und anderen typografischen Eigenschaften. In unserer Applikation verwenden wir `TextStyle`, um Titel und Inhalte visuell ansprechend und konsistent zu gestalten.

1. ElevatedButton und InkWell

Für interaktive Elemente wie Schaltflächen verwenden wir das `ElevatedButton`-Widget. Dieses Widget bietet eine erhöhte Schaltfläche im Material-Design-Stil, die auf Benutzerinteraktionen reagiert. In unserer Applikation wird es verwendet, um Aktionen auszulösen, wie das Öffnen eines Popup-Fensters mit zusätzlichen Informationen.

Das `InkWell`-Widget wird verwendet, um Berührungs- und Klickereignisse auf Widgets zu registrieren. Es wird häufig für Links und andere interaktive Text-Elemente verwendet, um visuelles Feedback bei Berührung zu bieten und Aktionen auszuführen, wie das Öffnen einer URL im Browser.

1. Schlussfolgerung

Die Navigation in unserer Applikation erfolgt über den Flutter-Navigator und benannte Routen, die eine einfache und intuitive Steuerung zwischen den verschiedenen Seiten ermöglichen. Zur Erstellung der Seiten verwenden wir eine Vielzahl von Widgets des Flutter-Frameworks, darunter `MaterialApp`, `Scaffold`, `AppBar`, `Drawer`, `Container`, `BoxDecoration`, `SingleChildScrollView`, `Column`, `Card`, `Padding`, `Text`, `TextStyle`, `ElevatedButton` und `InkWell`. Diese Widgets bieten die notwendigen Werkzeuge, um eine optisch ansprechende, funktionale und benutzerfreundliche Applikation zu entwickeln.