

Aplikacja Mobilna CountYourDollars

1

Wygenerowano za pomocą Doxygen 1.12.0

| | |
|------------------------------------------------|-----------|
| 1 Indeks przestrzeni nazw | 1 |
| 1.1 Lista przestrzeni nazw | 1 |
| 2 Indeks hierarchiczny | 3 |
| 2.1 Hierarchia klas | 3 |
| 3 Indeks klas | 5 |
| 3.1 Lista klas | 5 |
| 4 Indeks plików | 7 |
| 4.1 Lista plików | 7 |
| 5 Dokumentacja przestrzeni nazw | 9 |
| 5.1 Dokumentacja przestrzeni nazw CYD | 9 |
| 5.2 Dokumentacja przestrzeni nazw CYD.Pages | 9 |
| 5.3 Dokumentacja przestrzeni nazw CYD.Services | 10 |
| 6 Dokumentacja klas | 11 |
| 6.1 Dokumentacja klasy CYD.Pages.AboutPage | 11 |
| 6.1.1 Opis szczegółowy | 11 |
| 6.1.2 Dokumentacja konstruktora i destruktora | 12 |
| 6.1.2.1 AboutPage() | 12 |
| 6.1.3 Dokumentacja funkcji składowych | 12 |
| 6.1.3.1 LoadDataAsync() | 12 |
| 6.1.4 Dokumentacja atrybutów składowych | 12 |
| 6.1.4.1 _coinGeckoService | 12 |
| 6.1.5 Dokumentacja właściwości | 12 |
| 6.1.5.1 Cryptocurrencies | 12 |
| 6.2 Dokumentacja klasy CYD.App | 12 |
| 6.2.1 Opis szczegółowy | 13 |
| 6.2.2 Dokumentacja konstruktora i destruktora | 13 |
| 6.2.2.1 App() | 13 |
| 6.2.3 Dokumentacja funkcji składowych | 13 |
| 6.2.3.1 CreateWindow() | 13 |
| 6.2.3.2 LoadInitialDataAsync() | 14 |
| 6.2.3.3 SubscribeToLoginSuccessful() | 14 |
| 6.2.4 Dokumentacja atrybutów składowych | 14 |
| 6.2.4.1 _currencyLayerService | 14 |
| 6.2.4.2 _navigationPage | 14 |
| 6.2.5 Dokumentacja właściwości | 15 |
| 6.2.5.1 CurrencyRates | 15 |
| 6.3 Dokumentacja klasy CYD.AppShell | 15 |
| 6.3.1 Opis szczegółowy | 16 |
| 6.3.2 Dokumentacja konstruktora i destruktora | 16 |

| | |
|----------------------------------------------------------|----|
| 6.3.2.1 AppShell() | 16 |
| 6.3.3 Dokumentacja funkcji składowych | 16 |
| 6.3.3.1 DisplaySnackBarAsync() | 16 |
| 6.3.3.2 DisplayToastAsync() | 16 |
| 6.3.3.3 OnLogoutButtonClicked() | 17 |
| 6.3.3.4 SfSegmentedControl_SelectionChanged() | 17 |
| 6.3.4 Dokumentacja atrybutów składowych | 17 |
| 6.3.4.1 _authService | 17 |
| 6.4 Dokumentacja klasy CYD.Services.CoinGeckoService | 18 |
| 6.4.1 Opis szczegółowy | 18 |
| 6.4.2 Dokumentacja konstruktora i destruktora | 18 |
| 6.4.2.1 CoinGeckoService() | 18 |
| 6.4.3 Dokumentacja funkcji składowych | 18 |
| 6.4.3.1 GetTopCryptocurrenciesAsync() | 18 |
| 6.4.4 Dokumentacja atrybutów składowych | 19 |
| 6.4.4.1 _httpClient | 19 |
| 6.5 Dokumentacja klasy CYD.Services.CryptoCurrency | 19 |
| 6.5.1 Opis szczegółowy | 19 |
| 6.5.2 Dokumentacja właściwości | 19 |
| 6.5.2.1 Current_price | 19 |
| 6.5.2.2 Id | 19 |
| 6.5.2.3 Image | 19 |
| 6.5.2.4 Name | 20 |
| 6.5.2.5 Symbol | 20 |
| 6.6 Dokumentacja klasy CYD.Services.CurrencyLayerService | 20 |
| 6.6.1 Opis szczegółowy | 20 |
| 6.6.2 Dokumentacja konstruktora i destruktora | 20 |
| 6.6.2.1 CurrencyLayerService() | 20 |
| 6.6.3 Dokumentacja funkcji składowych | 21 |
| 6.6.3.1 GetTopCurrenciesAsync() | 21 |
| 6.6.4 Dokumentacja atrybutów składowych | 21 |
| 6.6.4.1 _httpClient | 21 |
| 6.6.4.2 ApiKey | 21 |
| 6.6.4.3 BaseUrl | 21 |
| 6.7 Dokumentacja klasy CYD.Pages.CurrencyPage | 21 |
| 6.7.1 Opis szczegółowy | 22 |
| 6.7.2 Dokumentacja konstruktora i destruktora | 22 |
| 6.7.2.1 CurrencyPage() | 22 |
| 6.7.3 Dokumentacja funkcji składowych | 22 |
| 6.7.3.1 LoadCurrencyRates() | 22 |
| 6.7.3.2 LoadCurrencyRatesAsync() | 22 |
| 6.7.4 Dokumentacja właściwości | 23 |

| | |
|--------------------------------------------------------------------------------|----|
| 6.7.4.1 CurrencyRates | 23 |
| 6.8 Dokumentacja klasy CYD.Pages.CurrencyRate | 23 |
| 6.8.1 Opis szczegółowy | 23 |
| 6.8.2 Dokumentacja właściwości | 23 |
| 6.8.2.1 DisplayValue | 23 |
| 6.8.2.2 Key | 23 |
| 6.8.2.3 Value | 24 |
| 6.9 Dokumentacja klasy CYD.Services.CurrencyLayerService.CurrencyRatesResponse | 24 |
| 6.9.1 Opis szczegółowy | 24 |
| 6.9.2 Dokumentacja właściwości | 24 |
| 6.9.2.1 ConversionRates | 24 |
| 6.10 Dokumentacja klasy FirebaseAuthService | 24 |
| 6.10.1 Opis szczegółowy | 25 |
| 6.10.2 Dokumentacja konstruktora i destruktora | 25 |
| 6.10.2.1 FirebaseAuthService() | 25 |
| 6.10.3 Dokumentacja funkcji składowych | 25 |
| 6.10.3.1 GetCurrentUserEmail() | 25 |
| 6.10.3.2 GetCurrentUserEmailAsync() | 25 |
| 6.10.3.3 GetCurrentUserEmailAsync() | 25 |
| 6.10.3.4 IsUserLoggedInAsync() | 25 |
| 6.10.3.5 LogoutAsync() | 26 |
| 6.10.3.6 RegisterWithEmailAndPasswordAsync() | 26 |
| 6.10.3.7 SignInWithEmailAndPasswordAsync() | 26 |
| 6.10.4 Dokumentacja atrybutów składowych | 26 |
| 6.10.4.1 _authLink | 26 |
| 6.10.4.2 _authProvider | 26 |
| 6.10.4.3 _firebaseApiKey | 26 |
| 6.11 Dokumentacja klasy CYD.Services.FirebaseDatabaseService | 27 |
| 6.11.1 Opis szczegółowy | 27 |
| 6.11.2 Dokumentacja konstruktora i destruktora | 27 |
| 6.11.2.1 FirebaseDatabaseService() | 27 |
| 6.11.3 Dokumentacja funkcji składowych | 27 |
| 6.11.3.1 GetSpendingAsync() | 27 |
| 6.11.3.2 SaveSpendingAsync() | 27 |
| 6.11.4 Dokumentacja atrybutów składowych | 28 |
| 6.11.4.1 _firebaseClient | 28 |
| 6.12 Dokumentacja klasy CYD.Pages.ForgotPassword | 28 |
| 6.12.1 Opis szczegółowy | 28 |
| 6.12.2 Dokumentacja konstruktora i destruktora | 28 |
| 6.12.2.1 ForgotPassword() | 28 |
| 6.12.3 Dokumentacja funkcji składowych | 29 |
| 6.12.3.1 OnLoginTapped() | 29 |

| | |
|-------------------------------------------------|----|
| 6.12.3.2 OnSendResetLinkClicked() | 29 |
| 6.13 Dokumentacja klasy CYD.Pages.LoginPage | 29 |
| 6.13.1 Opis szczegółowy | 30 |
| 6.13.2 Dokumentacja konstruktora i destruktora | 30 |
| 6.13.2.1 LoginPage() | 30 |
| 6.13.3 Dokumentacja funkcji składowych | 30 |
| 6.13.3.1 Button_Clicked_2() | 30 |
| 6.13.3.2 OnForgotPasswordTapped() | 30 |
| 6.13.3.3 OnLoginButtonClicked() | 31 |
| 6.13.3.4 OnSignUpTapped() | 31 |
| 6.13.4 Dokumentacja atrybutów w składowych | 31 |
| 6.13.4.1 _authService | 31 |
| 6.13.4.2 fingerprint | 32 |
| 6.13.5 Dokumentacja zdarzeń | 32 |
| 6.13.5.1 LoginSuccessful | 32 |
| 6.14 Dokumentacja klasy CYD.MauiProgram | 32 |
| 6.14.1 Opis szczegółowy | 32 |
| 6.14.2 Dokumentacja funkcji składowych | 32 |
| 6.14.2.1 CreateMauiApp() | 32 |
| 6.15 Dokumentacja klasy CYD.Pages.SignUpPage | 33 |
| 6.15.1 Opis szczegółowy | 33 |
| 6.15.2 Dokumentacja konstruktora i destruktora | 33 |
| 6.15.2.1 SignUpPage() | 33 |
| 6.15.3 Dokumentacja funkcji składowych | 33 |
| 6.15.3.1 OnLoginTapped() | 33 |
| 6.15.3.2 OnSignUpClicked() | 34 |
| 6.15.4 Dokumentacja atrybutów składowych | 34 |
| 6.15.4.1 _authService | 34 |
| 6.16 Dokumentacja klasy CYD.Services.Spending | 34 |
| 6.16.1 Opis szczegółowy | 34 |
| 6.16.2 Dokumentacja właściwości | 35 |
| 6.16.2.1 Amount | 35 |
| 6.16.2.2 Category | 35 |
| 6.17 Dokumentacja klasy CYD.Pages.SpendingsPage | 35 |
| 6.17.1 Opis szczegółowy | 36 |
| 6.17.2 Dokumentacja konstruktora i destruktora | 36 |
| 6.17.2.1 SpendingsPage() | 36 |
| 6.17.3 Dokumentacja funkcji składowych | 36 |
| 6.17.3.1 LoadSpendings() | 36 |
| 6.17.3.2 OnRefreshButtonClicked() | 36 |
| 6.17.4 Dokumentacja atrybutów w składowych | 36 |
| 6.17.4.1 _authService | 36 |

| | |
|------------------------------------------------------------------------|-----------|
| 6.17.4.2 _firebaseService | 36 |
| 7 Dokumentacja plików | 37 |
| 7.1 Dokumentacja pliku CYD/App.xaml.cs | 37 |
| 7.2 App.xaml.cs | 37 |
| 7.3 Dokumentacja pliku CYD/AppShell.xaml.cs | 38 |
| 7.3.1 Dokumentacja definicji typów | 38 |
| 7.3.1.1 Font | 38 |
| 7.4 AppShell.xaml.cs | 39 |
| 7.5 Dokumentacja pliku CYD/MauiProgram.cs | 40 |
| 7.6 MauiProgram.cs | 40 |
| 7.7 Dokumentacja pliku CYD/Pages/AboutPage.xaml.cs | 41 |
| 7.8 AboutPage.xaml.cs | 41 |
| 7.9 Dokumentacja pliku CYD/Pages/CurrencyPage.xaml.cs | 42 |
| 7.10 CurrencyPage.xaml.cs | 42 |
| 7.11 Dokumentacja pliku CYD/Pages/ForgotPassword.xaml.cs | 43 |
| 7.12 ForgotPassword.xaml.cs | 43 |
| 7.13 Dokumentacja pliku CYD/Pages/LoginPage.xaml.cs | 44 |
| 7.14 LoginPage.xaml.cs | 44 |
| 7.15 Dokumentacja pliku CYD/Pages/SignUpPage.xaml.cs | 45 |
| 7.16 SignUpPage.xaml.cs | 45 |
| 7.17 Dokumentacja pliku CYD/Pages/SpendingPage.xaml.cs | 46 |
| 7.18 SpendingPage.xaml.cs | 47 |
| 7.19 Dokumentacja pliku CYD/Services/CoinGeckoService.cs | 47 |
| 7.20 CoinGeckoService.cs | 48 |
| 7.21 Dokumentacja pliku CYD/Services/CurrencyLayerService.cs | 48 |
| 7.22 CurrencyLayerService.cs | 49 |
| 7.23 Dokumentacja pliku CYD/Services/FirebaseAuthService.cs | 50 |
| 7.24 FirebaseAuthService.cs | 50 |
| 7.25 Dokumentacja pliku CYD/Services/FirebaseService.cs | 51 |
| 7.26 FirebaseService.cs | 52 |
| Skorowidz | 53 |

Rozdział 1

Indeks przestrzeni nazw

1.1 Lista przestrzeni nazw

Tutaj znajdują się wszystkie przestrzenie nazw wraz z ich krótkimi opisami:

| | |
|----------------------------------------|----|
| CYD | 9 |
| CYD.Pages | 9 |
| CYD.Services | 10 |

Rozdział 2

Indeks hierarchiczny

2.1 Hierarchia klas

Ta lista dziedziczenia posortowana jest z grubsza, choć nie całkowicie, alfabetycznie:

| | |
|-------------------------------------------------------------------|----|
| Application | |
| CYD.App | 12 |
| CYD.Services.CoinGeckoService | 18 |
| ContentPage | |
| CYD.Pages.AboutPage | 11 |
| CYD.Pages.CurrencyPage | 21 |
| CYD.Pages.ForgotPassword | 28 |
| CYD.Pages.LoginPage | 29 |
| CYD.Pages.SignUpPage | 33 |
| CYD.Pages.SpendingsPage | 35 |
| CYD.Services.CryptoCurrency | 19 |
| CYD.Services.CurrencyLayerService | 20 |
| CYD.Pages.CurrencyRate | 23 |
| CYD.Services.CurrencyLayerService.CurrencyRatesResponse | 24 |
| FirebaseAuthService | 24 |
| CYD.Services.FirebaseDatabaseService | 27 |
| CYD.MauiProgram | 32 |
| Shell | |
| CYD.AppShell | 15 |
| CYD.Services.Spending | 34 |

Rozdział 3

Indeks klas

3.1 Lista klas

Tutaj znajdują się klasy, struktury, unie i interfejsy wraz z ich krótkimi opisami:

| | |
|-------------------------------------------------------------------------|----|
| CYD.Pages.AboutPage | 11 |
| CYD.App | |
| Główna klasa aplikacji zarządzająca nawigacją i danymi | 12 |
| CYD.AppShell | |
| Główna klasa powłoki aplikacji | 15 |
| CYD.Services.CoinGeckoService | 18 |
| CYD.Services.CryptoCurrency | 19 |
| CYD.Services.CurrencyLayerService | 20 |
| CYD.Pages.CurrencyPage | |
| Strona wyświetlająca kursy walut | 21 |
| CYD.Pages.CurrencyRate | |
| Klasa reprezentująca pojedynczy kurs waluty | 23 |
| CYD.Services.CurrencyLayerService.CurrencyRatesResponse | 24 |
| FirebaseAuthService | 24 |
| CYD.Services.FirebaseDatabaseService | 27 |
| CYD.Pages.ForgotPassword | 28 |
| CYD.Pages.LoginPage | 29 |
| CYD.MauiProgram | |
| Główna klasa konfiguruje Maui | 32 |
| CYD.Pages.SignUpPage | |
| Strona rejestracji użytkownika z obsługą Firebase | 33 |
| CYD.Services.Spending | 34 |
| CYD.Pages.SpendingsPage | |
| Strona zarządzająca wyświetlaniem wydatków użytkownika | 35 |

Rozdział 4

Indeks plików

4.1 Lista plików

Tutaj znajduje się lista wszystkich plików wraz z ich krótkimi opisami:

| | |
|-----------------------------------------------------------------|----|
| CYD/ App.xaml.cs | 37 |
| CYD/ AppShell.xaml.cs | 38 |
| CYD/ MauiProgram.cs | 40 |
| CYD/Pages/ AboutPage.xaml.cs | 41 |
| CYD/Pages/ CurrencyPage.xaml.cs | 42 |
| CYD/Pages/ ForgotPassword.xaml.cs | 43 |
| CYD/Pages/ LoginPage.xaml.cs | 44 |
| CYD/Pages/ SignUpPage.xaml.cs | 45 |
| CYD/Pages/ SpendingsPage.xaml.cs | 46 |
| CYD/Services/ CoinGeckoService.cs | 47 |
| CYD/Services/ CurrencyLayerService.cs | 48 |
| CYD/Services/ FirebaseAuthService.cs | 50 |
| CYD/Services/ FirebaseService.cs | 51 |

Rozdział 5

Dokumentacja przestrzeni nazw

5.1 Dokumentacja przestrzeni nazw CYD

Przestrzenie nazw

- namespace [Pages](#)
- namespace [Services](#)

Komponenty

- class [App](#)
Główna klasa aplikacji zarządzająca nawigacją i danymi.
- class [AppShell](#)
Główna klasa powłoki aplikacji.
- class [MauiProgram](#)
Główna klasa konfigurująca Maui.

5.2 Dokumentacja przestrzeni nazw CYD.Pages

Komponenty

- class [AboutPage](#)
- class [CurrencyPage](#)
Strona wyświetlająca kursy walut.
- class [CurrencyRate](#)
Klasa reprezentująca pojedynczy kurs waluty.
- class [ForgotPassword](#)
- class [LoginPage](#)
- class [SignUpPage](#)
Strona rejestracji użytkownika z obsługą Firebase.
- class [SpendingsPage](#)
Strona zarządzająca wyświetlaniem wydatków użytkownika.

5.3 Dokumentacja przestrzeni nazw CYD.Services

Komponenty

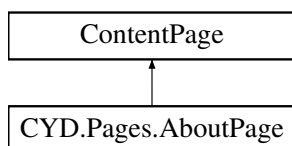
- class [CoinGeckoService](#)
- class [CryptoCurrency](#)
- class [CurrencyLayerService](#)
- class [FirebaseDatabaseService](#)
- class [Spending](#)

Rozdział 6

Dokumentacja klas

6.1 Dokumentacja klasy CYD.Pages.AboutPage

Diagram dziedziczenia dla CYD.Pages.AboutPage



Metody publiczne

- [AboutPage](#) ()

Właściwości

- `List< CryptoCurrency > Cryptocurrencies = new()` [get, set]

Metody prywatne

- `async void LoadDataAsync ()`

Atrybuty prywatne

- `CoinGeckoService _coinGeckoService`

6.1.1 Opis szczegółowy

Definicja w linii 5 pliku [AboutPage.xaml.cs](#).

6.1.2 Dokumentacja konstruktora i destruktora

6.1.2.1 AboutPage()

```
CYD.Pages.AboutPage.AboutPage () [inline]
```

Definicja w linii 10 pliku [AboutPage.xaml.cs](#).

6.1.3 Dokumentacja funkcji składowych

6.1.3.1 LoadDataAsync()

```
async void CYD.Pages.AboutPage.LoadDataAsync () [inline], [private]
```

Definicja w linii 24 pliku [AboutPage.xaml.cs](#).

6.1.4 Dokumentacja atrybutów składowych

6.1.4.1 _coinGeckoService

```
CoinGeckoService CYD.Pages.AboutPage._coinGeckoService [private]
```

Definicja w linii 7 pliku [AboutPage.xaml.cs](#).

6.1.5 Dokumentacja właściwości

6.1.5.1 Cryptocurrencies

```
List<CryptoCurrency> CYD.Pages.AboutPage.Cryptocurrencies = new() [get], [set]
```

Definicja w linii 21 pliku [AboutPage.xaml.cs](#).

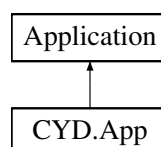
Dokumentacja dla tej klasy została wygenerowana z pliku:

- CYD/Pages/[AboutPage.xaml.cs](#)

6.2 Dokumentacja klasy CYD.App

Główna klasa aplikacji zarządzająca nawigacją i danymi.

Diagram dziedziczenia dla CYD.App



Metody publiczne

- [App \(\)](#)
Konstruktor aplikacji.

Metody chronione

- override Window [CreateWindow](#) (IActivationState? activationState)
Tworzy okno aplikacji.

Właściwości

- static Dictionary< string, decimal > [CurrencyRates](#) [get, private set]
Aktualne kursy walut.

Metody prywatne

- void [SubscribeToLoginSuccessful](#) ()
Subskrybuje zdarzenie pomyślnego logowania.
- async void [LoadInitialDataAsync](#) ()
Ładuje początkowe dane walutowe asynchronicznie.

Atrybuty prywatne

- NavigationPage [_navigationPage](#)
Główna strona nawigacyjna aplikacji.
- readonly [CurrencyLayerService _currencyLayerService](#)
Serwis do pobierania danych o walutach.

6.2.1 Opis szczegółowy

Główna klasa aplikacji zarządzająca nawigacją i danymi.

Definicja w linii 10 pliku [App.xaml.cs](#).

6.2.2 Dokumentacja konstruktora i destruktora

6.2.2.1 App()

```
CYD.App.App () [inline]
```

Konstruktor aplikacji.

Inicjalizuje komponenty, ładuje dane walutowe i ustawia stronę logowania.

Definicja w linii 22 pliku [App.xaml.cs](#).

6.2.3 Dokumentacja funkcji składowych

6.2.3.1 CreateWindow()

```
override Window CYD.App.CreateWindow (  
    IActivationState? activationState) [inline], [protected]
```

Tworzy okno aplikacji.

Parametry

| | |
|------------------------------|---------------------------|
| <code>activationState</code> | Stan aktywacji aplikacji. |
|------------------------------|---------------------------|

Zwraca

Zwraca główne okno aplikacji.

Definicja w linii 45 pliku [App.xaml.cs](#).

6.2.3.2 LoadInitialDataAsync()

```
async void CYD.App.LoadInitialDataAsync () [inline], [private]
```

Ładuje początkowe dane walutowe asynchronicznie.

Definicja w linii 64 pliku [App.xaml.cs](#).

6.2.3.3 SubscribeToLoginSuccessful()

```
void CYD.App.SubscribeToLoginSuccessful () [inline], [private]
```

Subskrybuje zdarzenie pomyślnego logowania.

Definicja w linii 53 pliku [App.xaml.cs](#).

6.2.4 Dokumentacja atrybutów składowych

6.2.4.1 _currencyLayerService

```
readonly CurrencyLayerService CYD.App._currencyLayerService [private]
```

Serwis do pobierania danych o walutach.

Definicja w linii 13 pliku [App.xaml.cs](#).

6.2.4.2 _navigationPage

```
NavigationPage CYD.App._navigationPage [private]
```

Główna strona nawigacyjna aplikacji.

Definicja w linii 12 pliku [App.xaml.cs](#).

6.2.5 Dokumentacja właściwości

6.2.5.1 CurrencyRates

```
Dictionary<string, decimal> CYD.App.CurrencyRates [static], [get], [private set]
```

Aktualne kursy walut.

Definicja w linii 15 pliku [App.xaml.cs](#).

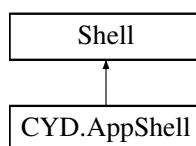
Dokumentacja dla tej klasy została wygenerowana z pliku:

- CYD/[App.xaml.cs](#)

6.3 Dokumentacja klasy CYD.AppShell

Główna klasa powłoki aplikacji.

Diagram dziedziczenia dla CYD.AppShell



Metody publiczne

- [AppShell](#) ()
Konstruktor klasy [AppShell](#).

Statyczne metody publiczne

- static async Task [DisplaySnackbarAsync](#) (string message)
Wyświetla komunikat Snackbar.
- static async Task [DisplayToastAsync](#) (string message)
Wyświetla komunikat Toast.

Metody prywatne

- void [SfSegmentedControl_SelectionChanged](#) (object sender, Syncfusion.Maui.Toolkit.SegmentedControl.↔ SelectionChangedEventArgs e)
Obsługuje zmianę motywu aplikacji.
- async void [OnLogoutButtonClicked](#) (object sender, EventArgs e)
Obsługuje kliknięcie przycisku wylogowania.

Atrybuty prywatne

- [FirebaseAuthService _authService](#)

Serwis uwierzytelniania Firebase.

6.3.1 Opis szczegółowy

Główna klasa powłoki aplikacji.

Zarządza interfejsem użytkownika oraz operacjami związanymi z motywem, komunikatami i wylogowaniem użytkownika.

Definicja w linii 16 pliku [AppShell.xaml.cs](#).

6.3.2 Dokumentacja konstruktora i destruktora

6.3.2.1 AppShell()

```
CYD.AppShell.AppShell () [inline]
```

Konstruktor klasy [AppShell](#).

Inicjalizuje komponenty, serwis uwierzytelniania oraz ustawia motyw aplikacji.

Definicja w linii 25 pliku [AppShell.xaml.cs](#).

6.3.3 Dokumentacja funkcji składowych

6.3.3.1 DisplaySnackbarAsync()

```
static async Task CYD.AppShell.DisplaySnackbarAsync (  
    string message) [inline], [static]
```

Wyświetla komunikat Snackbar.

Parametry

| | |
|----------------|-----------------------------------|
| <i>message</i> | Treść komunikatu do wyświetlenia. |
|----------------|-----------------------------------|

Tworzy i wyświetla snackbar z określonymi opcjami wizualnymi.

Definicja w linii 39 pliku [AppShell.xaml.cs](#).

6.3.3.2 DisplayToastAsync()

```
static async Task CYD.AppShell.DisplayToastAsync (  
    string message) [inline], [static]
```

Wyświetla komunikat Toast.

Parametry

| | |
|----------------|-----------------------------------|
| <i>message</i> | Treść komunikatu do wyświetlenia. |
|----------------|-----------------------------------|

Wyświetla komunikat typu Toast, z pominięciem na platformie Windows.

Definicja w linii 63 pliku [AppShell.xaml.cs](#).

6.3.3.3 OnLogoutButtonClicked()

```
async void CYD.AppShell.OnLogoutButtonClicked (  
    object sender,  
    EventArgs e) [inline], [private]
```

Obsługuje kliknięcie przycisku wylogowania.

Parametry

| | |
|---------------|------------------------------|
| <i>sender</i> | Obiekt wywołujący zdarzenie. |
| <i>e</i> | Argumenty zdarzenia. |

Definicja w linii 89 pliku [AppShell.xaml.cs](#).

6.3.3.4 SfSegmentedControl_SelectionChanged()

```
void CYD.AppShell.SfSegmentedControl_SelectionChanged (  
    object sender,  
    Syncfusion.Maui.Toolkit.SegmentedControl.SelectionChangedEventArgs e) [inline],  
[private]
```

Obsługuje zmianę motywu aplikacji.

Parametry

| | |
|---------------|------------------------------|
| <i>sender</i> | Obiekt wywołujący zdarzenie. |
| <i>e</i> | Argumenty zdarzenia. |

Definicja w linii 79 pliku [AppShell.xaml.cs](#).

6.3.4 Dokumentacja atrybutów składowych

6.3.4.1 _authService

```
FirestoreAuthService CYD.AppShell._authService [private]
```

Serwis uwierzytelniania Firebase.

Definicja w linii 18 pliku [AppShell.xaml.cs](#).

Dokumentacja dla tej klasy została wygenerowana z pliku:

- CYD/[AppShell.xaml.cs](#)

6.4 Dokumentacja klasy CYD.Services.CoinGeckoService

Metody publiczne

- [CoinGeckoService](#) ()
- async Task< List< [CryptoCurrency](#) > > [GetTopCryptocurrenciesAsync](#) (int count=10)

Atrybuty prywatne

- readonly HttpClient [_httpClient](#)

6.4.1 Opis szczegółowy

Service for interacting with the CoinGecko API to fetch cryptocurrency data.

Definicja w linii 13 pliku [CoinGeckoService.cs](#).

6.4.2 Dokumentacja konstruktora i destruktora

6.4.2.1 CoinGeckoService()

```
CYD.Services.CoinGeckoService.CoinGeckoService () [inline]
```

Initializes a new instance of the [CoinGeckoService](#) class.

Definicja w linii 20 pliku [CoinGeckoService.cs](#).

6.4.3 Dokumentacja funkcji składowych

6.4.3.1 GetTopCryptocurrenciesAsync()

```
async Task< List< CryptoCurrency >> CYD.Services.CoinGeckoService.GetTopCryptocurrencies↔  
Async (  
    int count = 10) [inline]
```

Fetches the top cryptocurrencies sorted by market cap.

Parametry

| | |
|--------------|---------------------------------------------------------|
| <i>count</i> | The number of cryptocurrencies to fetch. Default is 10. |
|--------------|---------------------------------------------------------|

Zwraca

A list of top cryptocurrencies.

Definicja w linii 33 pliku [CoinGeckoService.cs](#).

6.4.4 Dokumentacja atrybutów składowych

6.4.4.1 _httpClient

```
readonly HttpClient CYD.Services.CoinGeckoService._httpClient [private]
```

Definicja w linii 15 pliku [CoinGeckoService.cs](#).

Dokumentacja dla tej klasy została wygenerowana z pliku:

- CYD/Services/[CoinGeckoService.cs](#)

6.5 Dokumentacja klasy CYD.Services.CryptoCurrency

Właściwości

- string [Id](#) [get, set]
- string [Symbol](#) [get, set]
- string [Name](#) [get, set]
- decimal [Current_price](#) [get, set]
- string [Image](#) [get, set]

6.5.1 Opis szczegółowy

Represents a cryptocurrency with basic information.

Definicja w linii 52 pliku [CoinGeckoService.cs](#).

6.5.2 Dokumentacja właściwości

6.5.2.1 Current_price

```
decimal CYD.Services.CryptoCurrency.Current_price [get], [set]
```

Definicja w linii 57 pliku [CoinGeckoService.cs](#).

6.5.2.2 Id

```
string CYD.Services.CryptoCurrency.Id [get], [set]
```

Definicja w linii 54 pliku [CoinGeckoService.cs](#).

6.5.2.3 Image

```
string CYD.Services.CryptoCurrency.Image [get], [set]
```

Definicja w linii 58 pliku [CoinGeckoService.cs](#).

6.5.2.4 Name

```
string CYD.Services.CryptoCurrency.Name [get], [set]
```

Definicja w linii 56 pliku [CoinGeckoService.cs](#).

6.5.2.5 Symbol

```
string CYD.Services.CryptoCurrency.Symbol [get], [set]
```

Definicja w linii 55 pliku [CoinGeckoService.cs](#).

Dokumentacja dla tej klasy została wygenerowana z pliku:

- CYD/Services/[CoinGeckoService.cs](#)

6.6 Dokumentacja klasy CYD.Services.CurrencyLayerService

Komponenty

- class [CurrencyRatesResponse](#)

Metody publiczne

- [CurrencyLayerService](#) ()
- async Task< Dictionary< string, decimal > > [GetTopCurrenciesAsync](#) (string baseCurrency="USD", int count=5)

Atrybuty prywatne

- readonly HttpClient [_httpClient](#)

Statyczne atrybuty prywatne

- const string [ApiKey](#) = "42c7f538c6126b3d6b10018c"
- const string [BaseUrl](#) = "https://v6.exchangerate-api.com/v6/"

6.6.1 Opis szczegółowy

Definicja w linii 11 pliku [CurrencyLayerService.cs](#).

6.6.2 Dokumentacja konstruktora i destruktora

6.6.2.1 CurrencyLayerService()

```
CYD.Services.CurrencyLayerService.CurrencyLayerService () [inline]
```

Definicja w linii 18 pliku [CurrencyLayerService.cs](#).

6.6.3 Dokumentacja funkcji składowych

6.6.3.1 GetTopCurrenciesAsync()

```
async Task< Dictionary< string, decimal > > CYD.Services.CurrencyLayerService.GetTopCurrencies←  
Async (   
    string baseCurrency = "USD",  
    int count = 5) [inline]
```

Definicja w linii 27 pliku [CurrencyLayerService.cs](#).

6.6.4 Dokumentacja atrybutów składowych

6.6.4.1 _httpClient

```
readonly HttpClient CYD.Services.CurrencyLayerService._httpClient [private]
```

Definicja w linii 13 pliku [CurrencyLayerService.cs](#).

6.6.4.2 ApiKey

```
const string CYD.Services.CurrencyLayerService.ApiKey = "42c7f538c6126b3d6b10018c" [static],  
[private]
```

Definicja w linii 14 pliku [CurrencyLayerService.cs](#).

6.6.4.3 BaseUrl

```
const string CYD.Services.CurrencyLayerService.BaseUrl = "https://v6.exchangerate-api.com/v6/"  
[static], [private]
```

Definicja w linii 15 pliku [CurrencyLayerService.cs](#).

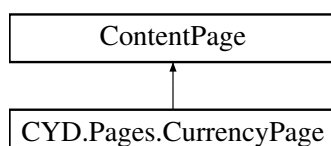
Dokumentacja dla tej klasy została wygenerowana z pliku:

- CYD/Services/[CurrencyLayerService.cs](#)

6.7 Dokumentacja klasy CYD.Pages.CurrencyPage

Strona wyświetlająca kursy walut.

Diagram dziedziczenia dla CYD.Pages.CurrencyPage



Metody publiczne

- [CurrencyPage](#) ()
Konstruktor strony walut.

Właściwości

- List< [CurrencyRate](#) > [CurrencyRates](#) [get, set]

Metody prywatne

- async void [LoadCurrencyRatesAsync](#) ()
Metoda asynchroniczna ³adująca dane o kursach walut.
- void [LoadCurrencyRates](#) (Dictionary< string, decimal > rates)
Metoda do za³adowania kursów walut do listy.

6.7.1 Opis szczegółowy

Strona wyświetlająca kursy walut.

Definicja w linii 10 pliku [CurrencyPage.xaml.cs](#).

6.7.2 Dokumentacja konstruktora i destruktora

6.7.2.1 CurrencyPage()

```
CYD.Pages.CurrencyPage.CurrencyPage () [inline]
```

Konstruktor strony walut.

Definicja w linii 15 pliku [CurrencyPage.xaml.cs](#).

6.7.3 Dokumentacja funkcji składowych

6.7.3.1 LoadCurrencyRates()

```
void CYD.Pages.CurrencyPage.LoadCurrencyRates (
    Dictionary< string, decimal > rates) [inline], [private]
```

Metoda do za³adowania kursów walut do listy.

Definicja w linii 50 pliku [CurrencyPage.xaml.cs](#).

6.7.3.2 LoadCurrencyRatesAsync()

```
async void CYD.Pages.CurrencyPage.LoadCurrencyRatesAsync () [inline], [private]
```

Metoda asynchroniczna ³adująca dane o kursach walut.

Definicja w linii 25 pliku [CurrencyPage.xaml.cs](#).

6.7.4 Dokumentacja wÃciwoÃci

6.7.4.1 CurrencyRates

```
List<CurrencyRate> CYD.Pages.CurrencyPage.CurrencyRates [get], [set]
```

Definicja w linii 12 pliku [CurrencyPage.xaml.cs](#).

Dokumentacja dla tej klasy zostaÅa wygenerowana z pliku:

- CYD/Pages/[CurrencyPage.xaml.cs](#)

6.8 Dokumentacja klasy CYD.Pages.CurrencyRate

Klasa reprezentujca pojedynczy kurs waluty.

WÃciwoÃci

- string [Key](#) [get, set]
- decimal [Value](#) [get, set]
- string [DisplayValue](#) [get]

Formatuje wartoÅ kursu waluty do wywietlenia.

6.8.1 Opis szczegÃÅwy

Klasa reprezentujca pojedynczy kurs waluty.

Definicja w linii 65 pliku [CurrencyPage.xaml.cs](#).

6.8.2 Dokumentacja wÃciwoÃci

6.8.2.1 DisplayValue

```
string CYD.Pages.CurrencyRate.DisplayValue [get]
```

Formatuje wartoÅ kursu waluty do wywietlenia.

Definicja w linii 71 pliku [CurrencyPage.xaml.cs](#).

6.8.2.2 Key

```
string CYD.Pages.CurrencyRate.Key [get], [set]
```

Definicja w linii 67 pliku [CurrencyPage.xaml.cs](#).

6.8.2.3 Value

```
decimal CYD.Pages.CurrencyRate.Value [get], [set]
```

Definicja w linii 68 pliku [CurrencyPage.xaml.cs](#).

Dokumentacja dla tej klasy została wygenerowana z pliku:

- CYD/Pages/[CurrencyPage.xaml.cs](#)

6.9 Dokumentacja klasy CYD.Services.CurrencyLayerService.CurrencyRatesResponse

Właściwości

- Dictionary< string, decimal > [ConversionRates](#) [get, set]

6.9.1 Opis szczegółowy

Definicja w linii 54 pliku [CurrencyLayerService.cs](#).

6.9.2 Dokumentacja właściwości

6.9.2.1 ConversionRates

```
Dictionary<string, decimal> CYD.Services.CurrencyLayerService.CurrencyRatesResponse.Conversion↵  
Rates [get], [set]
```

Definicja w linii 57 pliku [CurrencyLayerService.cs](#).

Dokumentacja dla tej klasy została wygenerowana z pliku:

- CYD/Services/[CurrencyLayerService.cs](#)

6.10 Dokumentacja klasy FirebaseAuthService

Metody publiczne

- [FirebaseAuthService](#) ()
- async Task< string > [RegisterWithEmailAndPasswordAsync](#) (string email, string password)
- async Task< string > [SignInWithEmailAndPasswordAsync](#) (string email, string password)
- string [GetCurrentUserEmail](#) ()
- async Task< string > [GetCurrentUserEmailAsync](#) ()
- async Task< string > [GetCurrentUserIdAsync](#) ()
- async Task< bool > [IsUserLoggedInAsync](#) ()
- async Task [LogoutAsync](#) ()

Atrybuty prywatne

- readonly FirebaseAuthProvider [_authProvider](#)
- readonly string [_firebaseApiKey](#) = "AlzaSyCsPbeDGli0slQWhIcPr2hI0NShLn7dE7M"
- FirebaseAuthLink [_authLink](#)

6.10.1 Opis szczegółowy

Definicja w linii 6 pliku [FirebaseAuthService.cs](#).

6.10.2 Dokumentacja konstruktora i destruktora

6.10.2.1 FirebaseAuthService()

```
FirebaseAuthService.FirebaseAuthService () [inline]
```

Definicja w linii 13 pliku [FirebaseAuthService.cs](#).

6.10.3 Dokumentacja funkcji składowych

6.10.3.1 GetCurrentUserEmail()

```
string FirebaseAuthService.GetCurrentUserEmail () [inline]
```

Definicja w linii 75 pliku [FirebaseAuthService.cs](#).

6.10.3.2 GetCurrentUserEmailAsync()

```
async Task< string > FirebaseAuthService.GetCurrentUserEmailAsync () [inline]
```

Definicja w linii 81 pliku [FirebaseAuthService.cs](#).

6.10.3.3 GetCurrentUserIdAsync()

```
async Task< string > FirebaseAuthService.GetCurrentUserIdAsync () [inline]
```

Definicja w linii 88 pliku [FirebaseAuthService.cs](#).

6.10.3.4 IsUserLoggedInAsync()

```
async Task< bool > FirebaseAuthService.IsUserLoggedInAsync () [inline]
```

Definicja w linii 105 pliku [FirebaseAuthService.cs](#).

6.10.3.5 LogoutAsync()

```
async Task FirebaseAuthService.LogoutAsync () [inline]
```

Definicja w linii 112 pliku [FirebaseAuthService.cs](#).

6.10.3.6 RegisterWithEmailAndPasswordAsync()

```
async Task< string > FirebaseAuthService.RegisterWithEmailAndPasswordAsync (
    string email,
    string password) [inline]
```

Definicja w linii 21 pliku [FirebaseAuthService.cs](#).

6.10.3.7 SignInWithEmailAndPasswordAsync()

```
async Task< string > FirebaseAuthService.SignInWithEmailAndPasswordAsync (
    string email,
    string password) [inline]
```

Definicja w linii 50 pliku [FirebaseAuthService.cs](#).

6.10.4 Dokumentacja atrybutów składowych

6.10.4.1 _authLink

```
FirebaseAuthLink FirebaseAuthService._authLink [private]
```

Definicja w linii 10 pliku [FirebaseAuthService.cs](#).

6.10.4.2 _authProvider

```
readonly FirebaseAuthProvider FirebaseAuthService._authProvider [private]
```

Definicja w linii 8 pliku [FirebaseAuthService.cs](#).

6.10.4.3 _firebaseApiKey

```
readonly string FirebaseAuthService._firebaseApiKey = "AIzaSyCsPbeDGii0slQWhIcPr2hI0NShLn7d↔E7M" [private]
```

Definicja w linii 9 pliku [FirebaseAuthService.cs](#).

Dokumentacja dla tej klasy została wygenerowana z pliku:

- CYD/Services/[FirebaseAuthService.cs](#)

6.11 Dokumentacja klasy CYD.Services.FirebaseDatabaseService

Metody publiczne

- [FirebaseDatabaseService](#) ()
- async Task [SaveSpendingAsync](#) (string category, decimal amount, string userEmail)
- async Task< List< [Spending](#) > > [GetSpendingsAsync](#) (string userEmail)

Atrybuty prywatne

- readonly FirebaseClient [_firebaseClient](#)

6.11.1 Opis szczegółowy

Definicja w linii 8 pliku [FirebaseService.cs](#).

6.11.2 Dokumentacja konstruktora i destruktora

6.11.2.1 FirebaseDatabaseService()

```
CYD.Services.FirebaseDatabaseService.FirebaseDatabaseService () [inline]
```

Definicja w linii 13 pliku [FirebaseService.cs](#).

6.11.3 Dokumentacja funkcji składowych

6.11.3.1 GetSpendingsAsync()

```
async Task< List< Spending > > CYD.Services.FirebaseDatabaseService.GetSpendingsAsync (  
    string userEmail) [inline]
```

Definicja w linii 38 pliku [FirebaseService.cs](#).

6.11.3.2 SaveSpendingAsync()

```
async Task CYD.Services.FirebaseDatabaseService.SaveSpendingAsync (  
    string category,  
    decimal amount,  
    string userEmail) [inline]
```

Definicja w linii 20 pliku [FirebaseService.cs](#).

6.11.4 Dokumentacja atrybutów składowych

6.11.4.1 `_firebaseClient`

```
readonly FirebaseClient CYD.Services.FirebaseDatabaseService._firebaseClient [private]
```

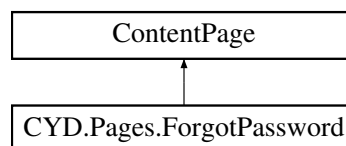
Definicja w linii 10 pliku [FirebaseService.cs](#).

Dokumentacja dla tej klasy została wygenerowana z pliku:

- CYD/Services/[FirebaseService.cs](#)

6.12 Dokumentacja klasy `CYD.Pages.ForgotPassword`

Diagram dziedziczenia dla `CYD.Pages.ForgotPassword`



Metody publiczne

- [ForgotPassword\(\)](#)
Konstruktor strony "ForgotPassword".

Metody prywatne

- async void [OnSendResetLinkClicked](#) (object sender, EventArgs e)
Obsługuje kliknięcie przycisku "Send Reset Link".
- async void [OnLoginTapped](#) (object sender, EventArgs e)
Obsługuje kliknięcie przycisku "Login" w przypadku, gdy użytkownik pamięta hasło.

6.12.1 Opis szczegółowy

Definicja w linii 5 pliku [ForgotPassword.xaml.cs](#).

6.12.2 Dokumentacja konstruktora i destruktora

6.12.2.1 `ForgotPassword()`

```
CYD.Pages.ForgotPassword.ForgotPassword () [inline]
```

Konstruktor strony "ForgotPassword".

Definicja w linii 8 pliku [ForgotPassword.xaml.cs](#).

6.12.3 Dokumentacja funkcji składowych

6.12.3.1 OnLoginTapped()

```
async void CYD.Pages.ForgotPassword.OnLoginTapped (
    object sender,
    EventArgs e) [inline], [private]
```

Obsługuje kliknięcie przycisku "Login" w przypadku, gdy użytkownik pamięta hasło.

Definicja w linii 30 pliku [ForgotPassword.xaml.cs](#).

6.12.3.2 OnSendResetLinkClicked()

```
async void CYD.Pages.ForgotPassword.OnSendResetLinkClicked (
    object sender,
    EventArgs e) [inline], [private]
```

Obsługuje kliknięcie przycisku "Send Reset Link".

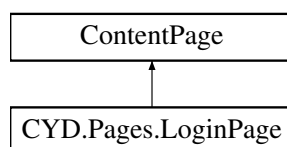
Definicja w linii 14 pliku [ForgotPassword.xaml.cs](#).

Dokumentacja dla tej klasy została wygenerowana z pliku:

- CYD/Pages/[ForgotPassword.xaml.cs](#)

6.13 Dokumentacja klasy CYD.Pages.LoginPage

Diagram dziedziczenia dla CYD.Pages.LoginPage



Metody publiczne

- [LoginPage](#) (IFingerprint [fingerprint](#))

Zdarzenia

- static EventHandler [LoginSuccessful](#)

Metody prywatne

- async void [OnLoginButtonClicked](#) (object sender, EventArgs e)
- async void [OnSignUpTapped](#) (object sender, EventArgs e)
- async void [OnForgotPasswordTapped](#) (object sender, EventArgs e)
- async void [Button_Clicked_2](#) (object sender, EventArgs e)

Atrybuty prywatne

- readonly IFingerprint [fingerprint](#)
- readonly [FirebaseAuthService](#) [_authService](#)

6.13.1 Opis szczegółowy

Reprezentuje stronę logowania w aplikacji. Strona ta obsługuje uwierzytelnianie użytkownika zarówno przez tradycyjne logowanie (e-mail/hasło), jak i uwierzytelnianie za pomocą odcisku palca. Komunikuje się z usługą uwierzytelniania Firebase, aby zalogować użytkownika i obsługiwać błądy związane z uwierzytelnianiem.

Definicja w linii 15 pliku [LoginPage.xaml.cs](#).

6.13.2 Dokumentacja konstruktora i destruktora

6.13.2.1 LoginPage()

```
CYD.Pages.LoginPage.LoginPage (
    IFingerprint fingerprint) [inline]
```

Inicjalizuje nową instancję klasy [LoginPage](#).

Parametry

| | |
|--------------------|----------------------------------------------------|
| <i>fingerprint</i> | Instancja usługi uwierzytelniania odciskiem palca. |
|--------------------|----------------------------------------------------|

Definicja w linii 29 pliku [LoginPage.xaml.cs](#).

6.13.3 Dokumentacja funkcji składowych

6.13.3.1 Button_Clicked_2()

```
async void CYD.Pages.LoginPage.Button_Clicked_2 (
    object sender,
    EventArgs e) [inline], [private]
```

Obsługuje kliknięcie przycisku uwierzytelniania za pomocą odcisku palca. Próbuje uwierzytelnić użytkownika za pomocą czytnika odcisków palców. Jeśli uwierzytelnienie jest udane, użytkownik jest zalogowany, a odpowiednie zdarzenie jest wyzwalane.

Parametry

| | |
|---------------|--------------------------------------|
| <i>sender</i> | Obiekt, który wyzwołał to zdarzenie. |
| <i>e</i> | Argumenty zdarzenia. |

Definicja w linii 121 pliku [LoginPage.xaml.cs](#).

6.13.3.2 OnForgotPasswordTapped()

```
async void CYD.Pages.LoginPage.OnForgotPasswordTapped (
    object sender,
    EventArgs e) [inline], [private]
```

Obsługuje kliknięcie linku "Zapomniałem hasła?". Nawigujemy do strony resetowania hasła.

Parametry

| | |
|---------------|-------------------------------------------------|
| <i>sender</i> | Obiekt, który wyzwoi ³ to zdarzenie. |
| <i>e</i> | Argumenty zdarzenia. |

Definicja w linii 109 pliku [LoginPage.xaml.cs](#).

6.13.3.3 OnLoginButtonClicked()

```
async void CYD.Pages.LoginPage.OnLoginButtonClicked (  
    object sender,  
    EventArgs e) [inline], [private]
```

Obs³uguje kliknięcie przycisku logowania. Próbuje uwierzytelnić użytkownika za pomocą podanego e-maila i has³a. Jeśli logowanie jest udane, użytkownik zostaje zalogowany, a odpowiednie zdarzenie jest wyzwalane. Jeśli logowanie się nie uda, wyświetlany jest odpowiedni komunikat o b³ędzie.

Parametry

| | |
|---------------|-------------------------------------------------|
| <i>sender</i> | Obiekt, który wyzwoi ³ to zdarzenie. |
| <i>e</i> | Argumenty zdarzenia. |

Definicja w linii 44 pliku [LoginPage.xaml.cs](#).

6.13.3.4 OnSignUpTapped()

```
async void CYD.Pages.LoginPage.OnSignUpTapped (  
    object sender,  
    EventArgs e) [inline], [private]
```

Obs³uguje kliknięcie linku "Zarejestruj się". Nawigujemy do strony rejestracji.

Parametry

| | |
|---------------|-------------------------------------------------|
| <i>sender</i> | Obiekt, który wyzwoi ³ to zdarzenie. |
| <i>e</i> | Argumenty zdarzenia. |

Definicja w linii 98 pliku [LoginPage.xaml.cs](#).

6.13.4 Dokumentacja atrybut³ów składowych**6.13.4.1 _authService**

```
readonly FirebaseAuthService CYD.Pages.LoginPage._authService [private]
```

Definicja w linii 18 pliku [LoginPage.xaml.cs](#).

6.13.4.2 fingerprint

```
readonly IFingerprint CYD.Pages.LoginPage.fingerprint [private]
```

Definicja w linii 17 pliku [LoginPage.xaml.cs](#).

6.13.5 Dokumentacja zdarzeń

6.13.5.1 LoginSuccessful

```
EventHandler CYD.Pages.LoginPage.LoginSuccessful [static]
```

Zdarzenie wyzwalane, gdy logowanie się powiedzie.

Definicja w linii 23 pliku [LoginPage.xaml.cs](#).

Dokumentacja dla tej klasy została wygenerowana z pliku:

- CYD/Pages/[LoginPage.xaml.cs](#)

6.14 Dokumentacja klasy CYD.MauiProgram

Główna klasa konfiguruje Maui.

Statyczne metody publiczne

- static MauiApp [CreateMauiApp](#) ()
Tworzenie aplikacji Maui.

6.14.1 Opis szczegółowy

Główna klasa konfiguruje Maui.

Definicja w linii 10 pliku [MauiProgram.cs](#).

6.14.2 Dokumentacja funkcji składowych

6.14.2.1 CreateMauiApp()

```
static MauiApp CYD.MauiProgram.CreateMauiApp () [inline], [static]
```

Tworzenie aplikacji Maui.

Rejestracja usług w kontenerze DI (Dependency Injection)

Rejestracja stron z trasami (routing) jako transient (krótkoterminowe)

Definicja w linii 13 pliku [MauiProgram.cs](#).

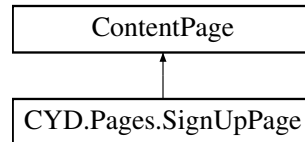
Dokumentacja dla tej klasy została wygenerowana z pliku:

- CYD/[MauiProgram.cs](#)

6.15 Dokumentacja klasy CYD.Pages.SignUpPage

Strona rejestracji użytkownika z obsługą Firebase.

Diagram dziedziczenia dla CYD.Pages.SignUpPage



Metody publiczne

- [SignUpPage \(\)](#)
Konstruktor inicjalizujący komponenty UI i serwis autoryzacji.

Metody prywatne

- async void [OnSignUpClicked](#) (object sender, EventArgs e)
Obsługuje kliknięcie przycisku rejestracji.
- async void [OnLoginTapped](#) (object sender, EventArgs e)
Obsługuje kliknięcie linku do logowania.

Atrybuty prywatne

- [FirebaseAuthService _authService](#)
Serwis uwierzytelniania Firebase.

6.15.1 Opis szczegółowy

Strona rejestracji użytkownika z obsługą Firebase.

Definicja w linii 9 pliku [SignUpPage.xaml.cs](#).

6.15.2 Dokumentacja konstruktora i destruktora

6.15.2.1 SignUpPage()

```
CYD.Pages.SignUpPage.SignUpPage () [inline]
```

Konstruktor inicjalizujący komponenty UI i serwis autoryzacji.

Definicja w linii 16 pliku [SignUpPage.xaml.cs](#).

6.15.3 Dokumentacja funkcji składowych

6.15.3.1 OnLoginTapped()

```
async void CYD.Pages.SignUpPage.OnLoginTapped (  
    object sender,  
    EventArgs e) [inline], [private]
```

Obsługuje kliknięcie linku do logowania.

Parametry

| | |
|---------------|------------------------------|
| <i>sender</i> | Obiekt wywołujący zdarzenie. |
| <i>e</i> | Argumenty zdarzenia. |

Definicja w linii 78 pliku [SignUpPage.xaml.cs](#).

6.15.3.2 OnSignUpClicked()

```
async void CYD.Pages.SignUpPage.OnSignUpClicked (  
    object sender,  
    EventArgs e) [inline], [private]
```

Obsługuje kliknięcie przycisku rejestracji.

Parametry

| | |
|---------------|------------------------------|
| <i>sender</i> | Obiekt wywołujący zdarzenie. |
| <i>e</i> | Argumenty zdarzenia. |

Definicja w linii 27 pliku [SignUpPage.xaml.cs](#).

6.15.4 Dokumentacja atrybutów składowych**6.15.4.1 _authService**

```
FirebaseAuthService CYD.Pages.SignUpPage._authService [private]
```

Serwis uwierzytelniania Firebase.

Definicja w linii 11 pliku [SignUpPage.xaml.cs](#).

Dokumentacja dla tej klasy została wygenerowana z pliku:

- [CYD/Pages/SignUpPage.xaml.cs](#)

6.16 Dokumentacja klasy CYD.Services.Spending**Właściwości**

- string [Category](#) [get, set]
- decimal [Amount](#) [get, set]

6.16.1 Opis szczegółowy

Definicja w linii 64 pliku [FirebaseService.cs](#).

6.16.2 Dokumentacja właściwości

6.16.2.1 Amount

```
decimal CYD.Services.Spending.Amount [get], [set]
```

Definicja w linii 67 pliku [FirebaseService.cs](#).

6.16.2.2 Category

```
string CYD.Services.Spending.Category [get], [set]
```

Definicja w linii 66 pliku [FirebaseService.cs](#).

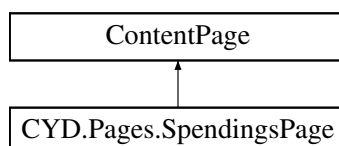
Dokumentacja dla tej klasy została wygenerowana z pliku:

- CYD/Services/[FirebaseService.cs](#)

6.17 Dokumentacja klasy CYD.Pages.SpendingsPage

Strona zarządzająca wyświetlaniem wydatków użytkownika.

Diagram dziedziczenia dla CYD.Pages.SpendingsPage



Metody publiczne

- [SpendingsPage \(\)](#)
Konstruktor inicjalizujący komponenty oraz serwisy.

Metody prywatne

- async Task [LoadSpendings \(\)](#)
Asynchroniczne³ adowanie wydatków użytkownika.
- async void [OnRefreshButtonClicked](#) (object sender, EventArgs e)
Obs³uguje kliknięcie przycisku "Refresh".

Atrybuty prywatne

- readonly [FirebaseDatabaseService _firebaseService](#)
Serwis bazy danych Firebase.
- readonly [FirebaseAuthService _authService](#)
Serwis uwierzytelniania Firebase.

6.17.1 Opis szczegółowy

Strona zarządzająca wyświetlaniem wydatków użytkownika.

Definicja w linii 13 pliku [SpendingPage.xaml.cs](#).

6.17.2 Dokumentacja konstruktora i destruktora

6.17.2.1 SpendingPage()

```
CYD.Pages.SpendingPage.SpendingPage () [inline]
```

Konstruktor inicjalizujący komponenty oraz serwisy.

Definicja w linii 21 pliku [SpendingPage.xaml.cs](#).

6.17.3 Dokumentacja funkcji składowych

6.17.3.1 LoadSpending()

```
async Task CYD.Pages.SpendingPage.LoadSpending () [inline], [private]
```

Asynchroniczne ładowanie wydatków użytkownika.

Definicja w linii 32 pliku [SpendingPage.xaml.cs](#).

6.17.3.2 OnRefreshButtonClicked()

```
async void CYD.Pages.SpendingPage.OnRefreshButtonClicked (  
    object sender,  
    EventArgs e) [inline], [private]
```

Obsługuje kliknięcie przycisku "Refresh".

Parametry

| | |
|---------------|------------------------------|
| <i>sender</i> | Obiekt wywołujący zdarzenie. |
| <i>e</i> | Argumenty zdarzenia. |

Definicja w linii 66 pliku [SpendingPage.xaml.cs](#).

6.17.4 Dokumentacja atrybutów w składowych

6.17.4.1 _authService

```
readonly FirebaseAuthService CYD.Pages.SpendingPage._authService [private]
```

Serwis uwierzytelniania Firebase.

Definicja w linii 16 pliku [SpendingPage.xaml.cs](#).

6.17.4.2 _firebaseService

```
readonly FirebaseDatabaseService CYD.Pages.SpendingPage._firebaseService [private]
```

Serwis bazy danych Firebase.

Definicja w linii 15 pliku [SpendingPage.xaml.cs](#).

Dokumentacja dla tej klasy została wygenerowana z pliku:

- [CYD/Pages/SpendingPage.xaml.cs](#)

Rozdział 7

Dokumentacja plików

7.1 Dokumentacja pliku CYD/App.xaml.cs

Komponenty

- class [CYD.App](#)
Główna klasa aplikacji zarządzająca nawigacją i danymi.

Przestrzenie nazw

- namespace [CYD](#)

7.2 App.xaml.cs

[Idź do dokumentacji tego pliku.](#)

```
00001 using CYD.Services;
00002 using Plugin.Fingerprint;
00003
00004 namespace CYD
00005 {
00010     public partial class App : Application
00011     {
00012         private NavigationPage _navigationPage;
00013         private readonly CurrencyLayerService _currencyLayerService;
00014
00015         public static Dictionary<string, decimal> CurrencyRates { get; private set; }
00016
00022         public App()
00023         {
00024             InitializeComponent(); //<--- Inicjalizacja komponentów UI.
00025             _currencyLayerService = new CurrencyLayerService(); //<--- Inicjalizacja serwisu
walutowego.
00026             LoadInitialDataAsync(); //<--- Asynchroniczne ładowanie danych walutowych.
00027
00028             var loginPage = new LoginPage(CrossFingerprint.Current); //<--- Tworzenie strony logowania
z uwierzytelnianiem odcisków palców.
00029
00030             _navigationPage = new NavigationPage(loginPage)
00031             {
00032                 BarBackgroundColor = Colors.Transparent, //<--- Przezroczysty kolor paska nawigacji.
00033                 BarTextColor = Colors.Transparent //<--- Przezroczysty kolor tekstu paska nawigacji.
00034             };
00035
00036             SubscribeToLoginSuccessful(); //<--- Subskrypcja zdarzenia pomyślnego logowania.
00037             MainPage = _navigationPage; //<--- Ustawienie głównej strony aplikacji.
00038         }
00039     }
```

```

00045         protected override Window CreateWindow(IActivationState? activationState)
00046         {
00047             return new Window(_navigationPage); //<--- Tworzenie nowego okna aplikacji.
00048         }
00049
00053         private void SubscribeToLoginSuccessful()
00054         {
00055             LoginPage.LoginSuccessful += (sender, args) =>
00056             {
00057                 _navigationPage.PushAsync(new AppShell()); //<--- Nawigacja do głównej powłoki
00058                 aplikacji.
00059             };
00060
00064         private async void LoadInitialDataAsync()
00065         {
00066             try
00067             {
00068                 CurrencyRates = await _currencyLayerService.GetTopCurrenciesAsync("USD"); //<---
00069                 Pobieranie kursów walut z API.
00070                 if (CurrencyRates != null)
00071                 {
00072                     Console.WriteLine("Currency data successfully loaded."); //<--- Informacja o
00073                     załadowaniu danych.
00074                 }
00075                 catch (Exception ex)
00076                 {
00077                     Console.WriteLine($"Error loading currency data: {ex.Message}"); //<--- Obsługa błędu
00078                     podczas ładowania danych.
00079                 }
00080             }
00081         }

```

7.3 Dokumentacja pliku CYD/AppShell.xaml.cs

Komponenty

- class `CYD.AppShell`
Główna klasa powłoki aplikacji.

Przestrzenie nazw

- namespace `CYD`

Definicje typów

- using `Font` = `Microsoft.Maui.Font`

7.3.1 Dokumentacja definicji typów

7.3.1.1 Font

using `Font` = `Microsoft.Maui.Font`

Definicja w linii 3 pliku `AppShell.xaml.cs`.

7.4 AppShell.xaml.cs

[Idź do dokumentacji tego pliku.](#)

```

00001 using CommunityToolkit.Maui.Alerts;
00002 using CommunityToolkit.Maui.Core;
00003 using Font = Microsoft.Maui.Font;
00004 using Firebase.Auth;
00005 using Plugin.Fingerprint;
00006
00007 namespace CYD
00008 {
00016     public partial class AppShell : Shell
00017     {
00018         private FirebaseAuthService _authService;
00019
00025         public AppShell()
00026         {
00027             InitializeComponent(); //<--- Inicjalizacja komponentów interfejsu użytkownika.
00028             _authService = new FirebaseAuthService(); //<--- Inicjalizacja serwisu Firebase.
00029             var currentTheme = Application.Current!.UserAppTheme; //<--- Pobranie aktualnego motywu
aplikacji.
00030             ThemeSegmentedControl.SelectedIndex = currentTheme == AppTheme.Light ? 0 : 1; //<---
Ustawienie wybranego motywu w kontrolce.
00031         }
00032
00039         public static async Task DisplaySnackbarAsync(string message)
00040         {
00041             CancellationTokensource cancellationTokensource = new CancellationTokensource(); //<---
Źródło tokena anulowania dla Snackbar.
00042
00043             var snackbarOptions = new SnackbarOptions
00044             {
00045                 BackgroundColor = Color.FromArgb("#FF3300"), //<--- Ustawienie koloru tła.
00046                 TextColor = Colors.White, //<--- Ustawienie koloru tekstu.
00047                 ActionButtonTextColor = Colors.Yellow, //<--- Ustawienie koloru przycisku akcji.
00048                 CornerRadius = new CornerRadius(0), //<--- Ustawienie promienia zaokrąglenia
narożników.
00049                 Font = Font.SystemFontOfSize(18), //<--- Ustawienie rozmiaru czcionki dla tekstu.
00050                 ActionButtonFont = Font.SystemFontOfSize(14) //<--- Ustawienie rozmiaru czcionki dla
przycisku.
00051             };
00052
00053             var snackbar = Snackbar.Make(message, visualOptions: snackbarOptions); //<--- Utworzenie
obiektu Snackbar z określonymi opcjami.
00054             await snackbar.Show(cancellationTokensource.Token); //<--- Wyświetlenie Snackbar z obsługą
anulowania.
00055         }
00056
00063         public static async Task DisplayToastAsync(string message)
00064         {
00065             if (OperatingSystem.IsWindows()) //<--- Sprawdzenie, czy system to Windows.
00066                 return; //<--- Anulowanie dla systemu Windows.
00067
00068             var toast = Toast.Make(message, textSize: 18); //<--- Utworzenie obiektu Toast.
00069
00070             var cts = new CancellationTokensource(TimeSpan.FromSeconds(5)); //<--- Źródło tokena
anulowania na 5 sekund.
00071             await toast.Show(cts.Token); //<--- Wyświetlenie komunikatu Toast.
00072         }
00073
00079         private void SfSegmentedControl_SelectionChanged(object sender,
Syncfusion.Maui.Toolkit.SegmentedControl.SelectionChangedEventArgs e)
00080         {
00081             Application.Current!.UserAppTheme = e.NewIndex == 0 ? AppTheme.Light : AppTheme.Dark;
//<--- Zmiana motywu aplikacji.
00082         }
00083
00089         private async void OnLogoutButtonClicked(object sender, EventArgs e)
00090         {
00091             await _authService.LogoutAsync(); //<--- Wylogowanie użytkownika przez Firebase.
00092
00093             Application.Current.MainPage = new NavigationPage(new
LoginPage(CrossFingerprint.Current)); //<--- Nawigacja do strony logowania.
00094             SecureStorage.Remove("user_email"); //<--- Usunięcie zapisanych danych użytkownika.
00095             SecureStorage.Remove("user_uid"); //<--- Usunięcie zapisanego identyfikatora użytkownika.
00096         }
00097     }
00098 }

```



```

00046         builder.Services.AddSingleton<ManageMetaPageModel>(); // <--- Rejestracja modelu strony
zarządzania metadanymi jako singleton
00047
00049         builder.Services.AddTransientWithShellRoute<ProjectDetailPage,
ProjectDetailPageModel>("project"); // <--- Rejestracja strony szczegółów projektu z trasą
00050         builder.Services.AddTransientWithShellRoute<TaskDetailPage, TaskDetailPageModel>("task");
// <--- Rejestracja strony szczegółów zadania z trasą
00051
00052         return builder.Build(); // <--- Budowanie i zwrócenie aplikacji Maui
00053     }
00054 }
00055 }

```

7.7 Dokumentacja pliku CYD/Pages/AboutPage.xaml.cs

Komponenty

- class [CYD.Pages.AboutPage](#)

Przestrzenie nazw

- namespace [CYD](#)
- namespace [CYD.Pages](#)

7.8 AboutPage.xaml.cs

Idź do dokumentacji tego pliku.

```

00001 using CYD.Services; // <--- Użycie serwisu do pobierania danych o kryptowalutach
00002
00003 namespace CYD.Pages
00004 {
00005     public partial class AboutPage : ContentPage
00006     {
00007         private CoinGeckoService _coinGeckoService; // <--- Inicjalizacja zmiennej przechowującej
usługę CoinGeckoService, odpowiedzialną za pobieranie danych o kryptowalutach
00008
00009         // Konstruktor strony AboutPage
00010         public AboutPage()
00011         {
00012             InitializeComponent(); // <--- Inicjalizacja komponentów strony (np. UI)
00013
00014             _coinGeckoService = new CoinGeckoService(); // <--- Tworzenie instancji CoinGeckoService,
która będzie używana do pobierania danych
00015
00016             BindingContext = this; // <--- Ustawienie kontekstu powiązania (BindingContext) na bieżący
obiekt (AboutPage), umożliwia bindowanie danych do UI
00017             LoadDataAsync(); // <--- Wywołanie metody asynchronicznej w celu załadowania danych
kryptowalut
00018         }
00019
00020         // Właściwość przechowująca listę kryptowalut
00021         public List<Cryptocurrency> Cryptocurrencies { get; set; } = new(); // <--- Lista kryptowalut,
która będzie powiązana z UI
00022
00023         // Asynchroniczna metoda do ładowania danych kryptowalut
00024         private async void LoadDataAsync()
00025         {
00026             Cryptocurrencies = await _coinGeckoService.GetTopCryptocurrenciesAsync(); // <--- Pobiera
dane o kryptowalutach z serwisu CoinGecko
00027             OnPropertyChanged(nameof(Cryptocurrencies)); // <--- Powiadomienie o zmianie właściwości
Cryptocurrencies, co odwieża UI
00028         }
00029     }
00030 }

```

7.9 Dokumentacja pliku CYD/Pages/CurrencyPage.xaml.cs

Komponenty

- class [CYD.Pages.CurrencyPage](#)
Strona wyświetlająca kursy walut.
- class [CYD.Pages.CurrencyRate](#)
Klasa reprezentująca pojedynczy kurs waluty.

Przestrzenie nazw

- namespace [CYD](#)
- namespace [CYD.Pages](#)

7.10 CurrencyPage.xaml.cs

Idź do dokumentacji tego pliku.

```
00001 using Microsoft.Maui.Controls; // <--- Użycie przestrzeni nazw do kontrolki w aplikacji Maui
00002 using System; // <--- Użycie przestrzeni nazw do ogólnych operacji systemowych
00003 using System.Collections.Generic; // <--- Użycie przestrzeni nazw do operacji na kolekcjach
00004 using System.Linq; // <--- Użycie przestrzeni nazw do operacji na kolekcjach
00005 using System.Threading.Tasks; // <--- Użycie przestrzeni nazw do operacji asynchronicznych
00006
00007 namespace CYD.Pages
00008 {
00010     public partial class CurrencyPage : ContentPage
00011     {
00012         public List<CurrencyRate> CurrencyRates { get; set; } // <--- Lista obiektów CurrencyRate,
zawierająca kursy walut
00013
00015         public CurrencyPage()
00016         {
00017             InitializeComponent(); // <--- Inicjalizacja komponentów strony
00018             BindingContext = this; // <--- Ustawienie bieżącej strony jako kontekst powiązania
00019
00020             // Asynchronicznie załaduj dane walut
00021             LoadCurrencyRatesAsync(); // <--- Wywołanie asynchronicznej metody wczytania kursów walut
00022         }
00023
00025         private async void LoadCurrencyRatesAsync()
00026         {
00027             try
00028             {
00029                 // Używamy CurrencyLayerService do pobrania walut
00030                 var service = new Services.CurrencyLayerService(); // <--- Inicjalizacja serwisu do
pobierania kursów walut
00031                 var rates = await service.GetTopCurrenciesAsync("USD", 5); // <--- Pobierz tylko 5
najlepszych walut względem USD
00032
00033                 if (rates != null && rates.Any()) // <--- Sprawdzamy, czy dane zostały poprawnie
pobrane
00034                 {
00035                     LoadCurrencyRates(rates); // <--- Załaduj dane walut do widoku
00036                 }
00037                 else
00038                 {
00039                     await DisplayAlert("Error", "No currency rates available.", "OK"); // <---
Wyświetlamy komunikat o braku dostępnych danych
00040                 }
00041             }
00042             catch (Exception ex)
00043             {
00044                 Console.WriteLine($"Error loading currency rates: {ex.Message}"); // <--- Zapisujemy
błąd w konsoli
00045                 await DisplayAlert("Error", "Failed to load currency rates. Please try again later.",
"OK"); // <--- Wyświetlamy komunikat o błędzie
00046             }
00047         }
00048
00050         private void LoadCurrencyRates(Dictionary<string, decimal> rates)
00051         {
```

```

00052         // Przekształcamy dane w listę do powiązania
00053         CurrencyRates = rates.Select(rate => new CurrencyRate
00054         {
00055             Key = rate.Key, // <--- Przypisanie klucza (nazwa waluty)
00056             Value = rate.Value // <--- Przypisanie wartości (kurs waluty)
00057         }).ToList();
00058
00059         // Powiadomienie o zmianie danych
00060         OnPropertyChanged(nameof(CurrencyRates)); // <--- Powiadomienie widoku o zmianie danych
00061     }
00062 }
00063
00064 public class CurrencyRate
00065 {
00066     public string Key { get; set; } // <--- Klucz waluty (np. "USD")
00067     public decimal Value { get; set; } // <--- Wartość kursu waluty
00068     public string DisplayValue => $"{Key}: {Value:F2}"; // <--- Formatowanie do wyświetlenia jako
00069     "USD: 1.23"
00070 }
00071 }
00072 }
00073 }

```

7.11 Dokumentacja pliku CYD/Pages/ForgotPassword.xaml.cs

Komponenty

- class [CYD.Pages.ForgotPassword](#)

Przestrzenie nazw

- namespace [CYD](#)
- namespace [CYD.Pages](#)

7.12 ForgotPassword.xaml.cs

[Idź do dokumentacji tego pliku.](#)

```

00001 using Microsoft.Maui.Controls; // <--- Użycie przestrzeni nazw dla kontrolki w aplikacji Maui
00002
00003 namespace CYD.Pages;
00004
00005 public partial class ForgotPassword : ContentPage
00006 {
00007     public ForgotPassword()
00008     {
00009         InitializeComponent(); // <--- Inicjalizacja komponentów strony
00010     }
00011
00012     private async void OnSendResetLinkClicked(object sender, EventArgs e)
00013     {
00014         string email = EmailEntry.Text; // <--- Pobranie adresu e-mail z kontrolki
00015
00016         // Weryfikacja poprawności e-maila
00017         if (string.IsNullOrEmpty(email) || !email.Contains("@"))
00018         {
00019             await DisplayAlert("Error", "Please enter a valid email address.", "OK"); // <---
00020             Wyświetlanie komunikatu o błędzie, gdy e-mail jest niepoprawny
00021             return;
00022         }
00023
00024         // Logika do wysłania e-maila z instrukcjami resetowania hasła
00025         await DisplayAlert("Reset Link Sent", "A reset link has been sent to your email address.",
00026             "OK"); // <--- Informowanie użytkownika o wysłaniu linku do resetowania hasła
00027     }
00028
00029     private async void OnLoginTapped(object sender, EventArgs e)
00030     {
00031         await Navigation.PopAsync(); // <--- Powrót do poprzedniej strony
00032     }
00033 }
00034 }

```

7.13 Dokumentacja pliku CYD/Pages/LoginPage.xaml.cs

Komponenty

- class [CYD.Pages.LoginPage](#)

Przestrzenie nazw

- namespace [CYD](#)
- namespace [CYD.Pages](#)

7.14 LoginPage.xaml.cs

[Idź do dokumentacji tego pliku.](#)

```
00001 namespace CYD.Pages
00002 {
00003     using Microsoft.Maui.Controls;
00004     using Microsoft.Maui.Storage;
00005     using Plugin.Fingerprint.Abstractions;
00006     using System;
00007     using System.Net.Http.Headers;
00008     using System.Threading.Tasks;
00009
00015     public partial class LoginPage : ContentPage
00016     {
00017         private readonly IFingerprint fingerprint;
00018         private readonly FirebaseAuthService _authService;
00019
00023         public static event EventHandler LoginSuccessful;
00024
00029         public LoginPage(IFingerprint fingerprint)
00030         {
00031             InitializeComponent();
00032             _authService = new FirebaseAuthService();
00033             this.fingerprint = fingerprint;
00034         }
00035
00044         private async void OnLoginButtonClicked(object sender, EventArgs e)
00045         {
00046             string email = EmailEntry.Text;
00047             string password = PasswordEntry.Text;
00048
00049             // Sprawdzamy, czy e-mail lub has³o s¹ puste
00050             if (string.IsNullOrEmpty(email) || string.IsNullOrEmpty(password)) // <--- Sprawda, czy
e-mail lub has³o s¹ puste
00051             {
00052                 await DisplayAlert("B³¹d", "E-mail i has³o nie mog¹ byæ puste.", "OK");
00053                 return;
00054             }
00055
00056             try
00057             {
00058                 // Próbujemy zalogowaæ u¿ytkownika za pomoc¹ e-maila i has³a
00059                 var result = await _authService.SignInWithEmailAndPasswordAsync(email, password);
00060
00061                 if (!string.IsNullOrEmpty(result) && result.StartsWith("Error")) // <--- Sprawda, czy
wynik zawiera b³¹d
00062                 {
00063                     // Obs³ugujemy konkretne b³êdy uwierzytelniania
00064                     if (result.Contains("INVALID_LOGIN_CREDENTIALS")) // <--- Sprawda, czy b³¹d
wynika z nieprawid³owych danych logowania
00065                     {
00066                         await DisplayAlert("B³¹d", "Nieprawid³owy e-mail lub has³o.", "OK");
00067                     }
00068                     else if (result.Contains("INVALID_EMAIL")) // <--- Sprawda, czy b³¹d wynika z
nieprawid³owego e-maila
00069                     {
00070                         await DisplayAlert("B³¹d", "Nieprawid³owy e-mail.", "OK");
00071                     }
00072                     else
00073                     {
00074                         await DisplayAlert("B³¹d", "Nieznany b³¹d. Spróbuj ponownie.", "OK");
00075                     }
00076                 }
00077             }
00078         }
00079     }
00080 }
```

```

00077         else
00078         {
00079             // Logowanie się powiodło
00080             await DisplayAlert("Sukces", "Logowanie udane!", "OK");
00081             LoginSuccessful?.Invoke(this, EventArgs.Empty); // <--- Wyzwala zdarzenie sukcesu
logowania
00082         }
00083     }
00084     catch (Exception ex)
00085     {
00086         // Przechowujemy e-mail użytkownika w bezpiecznej pamięci i obsługujemy błędy ogólne
00087         await SecureStorage.SetAsync("user_email", email);
00088         await DisplayAlert("Błąd", $"Błąd: {ex.Message}", "OK");
00089     }
00090 }
00091
00092 private async void OnSignUpTapped(object sender, EventArgs e)
00093 {
00094     await Navigation.PushAsync(new SignUpPage());
00095 }
00096
00097 private async void OnForgotPasswordTapped(object sender, EventArgs e)
00098 {
00099     await Navigation.PushAsync(new ForgotPassword());
00100 }
00101
00102 private async void Button_Clicked_2(object sender, EventArgs e)
00103 {
00104     var request = new AuthenticationRequestConfiguration("Logowanie", "Za pomocą odcisku
00105     palca");
00106     var result = await fingerprint.AuthenticateAsync(request);
00107     if (result.Authenticated) // <--- Sprawdza, czy uwierzytelnienie odciskiem palca jest
00108     udane
00109     {
00110         await DisplayAlert("Sukces", "Logowanie udane!", "OK");
00111         LoginSuccessful?.Invoke(this, EventArgs.Empty); // <--- Wyzwala zdarzenie sukcesu
logowania
00112     }
00113     else
00114     {
00115         await DisplayAlert("Błąd", "Logowanie nieudane.", "OK");
00116     }
00117 }

```

7.15 Dokumentacja pliku CYD/Pages/SignUpPage.xaml.cs

Komponenty

- class [CYD.Pages.SignUpPage](#)
Strona rejestracji użytkownika z obsługą Firebase.

Przestrzenie nazw

- namespace [CYD](#)
- namespace [CYD.Pages](#)

7.16 SignUpPage.xaml.cs

[Idź do dokumentacji tego pliku.](#)

```

00001 using Microsoft.Maui.Controls;
00002
00003 namespace CYD.Pages;
00004
00009 public partial class SignUpPage : ContentPage
00010 {
00011     private FirebaseAuthService _authService;

```

```

00012
00016     public SignUpPage()
00017     {
00018         InitializeComponent(); //<--- Inicjalizacja komponentów UI.
00019         _authService = new FirebaseAuthService(); //<--- Inicjalizacja serwisu Firebase.
00020     }
00021
00027     private async void OnSignUpClicked(object sender, EventArgs e)
00028     {
00029         string firstName = FirstNameEntry.Text; //<--- Pobranie imienia użytkownika.
00030         string lastName = LastNameEntry.Text; //<--- Pobranie nazwiska użytkownika.
00031         string email = EmailEntry.Text; //<--- Pobranie adresu e-mail.
00032         string password = PasswordEntry.Text; //<--- Pobranie hasła.
00033         string confirmPassword = ConfirmPasswordEntry.Text; //<--- Pobranie potwierdzenia hasła.
00034
00035         if (string.IsNullOrEmpty(firstName) || string.IsNullOrEmpty(lastName) ||
00036             string.IsNullOrEmpty(email) || string.IsNullOrEmpty(password) ||
00037             string.IsNullOrEmpty(confirmPassword))
00038         {
00039             await DisplayAlert("Error", "Please fill in all fields.", "OK"); //<--- Sprawdzenie
00040             pustych pól.
00041             return;
00042         }
00043         if (password != confirmPassword)
00044         {
00045             await DisplayAlert("Error", "Passwords do not match.", "OK"); //<--- Sprawdzenie zgodności
00046             haseł.
00047             return;
00048         }
00049         if (password.Length < 6)
00050         {
00051             await DisplayAlert("Error", "Password must be at least 6 signs!", "OK"); //<---
00052             Sprawdzenie długości hasła.
00053             return;
00054         }
00055         string result = await _authService.RegisterWithEmailAndPasswordAsync(email, password); //<---
00056             Rejestracja użytkownika.
00057         if (!string.IsNullOrEmpty(result) && result.StartsWith("Error"))
00058         {
00059             await DisplayAlert("Error", "Unable to register account", "OK"); //<--- Obsługa błędu
00060             rejestracji.
00061             return;
00062         }
00063         else
00064         {
00065             await DisplayAlert("Registration Success", "Account created successfully!", "OK"); //<---
00066             Sukces rejestracji.
00067             await Navigation.PopAsync(); //<--- Powrót do poprzedniej strony.
00068             EmailEntry.Text = string.Empty; //<--- Wyczyszczenie pola e-mail.
00069             PasswordEntry.Text = string.Empty; //<--- Wyczyszczenie pola hasła.
00070             ConfirmPasswordEntry.Text = string.Empty; //<--- Wyczyszczenie pola potwierdzenia hasła.
00071         }
00072     }
00078     private async void OnLoginTapped(object sender, EventArgs e)
00079     {
00080         await Navigation.PopAsync(); //<--- Powrót do poprzedniej strony.
00081     }
00082 }

```

7.17 Dokumentacja pliku CYD/Pages/SpendingPage.xaml.cs

Komponenty

- class [CYD.Pages.SpendingPage](#)
Strona zarządzająca wyświetlaniem wydatków użytkownika.

Przestrzenie nazw

- namespace [CYD](#)
- namespace [CYD.Pages](#)

7.18 SpendingsPage.xaml.cs

Idź do dokumentacji tego pliku.

```
00001 using System;
00002 using System.Collections.Generic;
00003 using System.Threading.Tasks;
00004 using CYD.Services;
00005 using Microsoft.Maui.Controls;
00006
00007 namespace CYD.Pages
00008 {
00013     public partial class SpendingsPage : ContentPage
00014     {
00015         private readonly FirebaseDatabaseService _firebaseService;
00016         private readonly FirebaseAuthService _authService;
00017
00021         public SpendingsPage()
00022         {
00023             InitializeComponent(); //<--- Inicjalizacja komponentów UI.
00024             _firebaseService = new FirebaseDatabaseService(); //<--- Inicjalizacja serwisu bazy
danych.
00025             _authService = new FirebaseAuthService(); //<--- Inicjalizacja serwisu uwierzytelniania.
00026             LoadSpendings(); //<--- ładowanie wydatków.
00027         }
00028
00032         private async Task LoadSpendings()
00033         {
00034             LoadingIndicator.IsRunning = true; //<--- Pokazanie wskanika ładowania.
00035             LoadingIndicator.IsVisible = true;
00036
00037             var isLoggedIn = await _authService.IsUserLoggedInAsync(); //<--- Sprawdzenie, czy
użytkownik jest zalogowany.
00038             if (!isLoggedIn)
00039             {
00040                 await DisplayAlert("Error", "User not logged in.", "OK"); //<--- Komunikat o błądzie.
00041                 return;
00042             }
00043
00044             var userEmail = await _authService.GetCurrentUserEmailAsync(); //<--- Pobranie adresu
e-mail użytkownika.
00045             if (string.IsNullOrEmpty(userEmail) || userEmail == "User not logged in")
00046             {
00047                 await DisplayAlert("Error", "Failed to retrieve user email.", "OK"); //<--- Komunikat
o błądzie.
00048                 return;
00049             }
00050
00051             var sanitizedEmail = userEmail.Replace("@", "-at-").Replace(".", "-dot-"); //<---
Sanitacja adresu e-mail.
00052             var spendings = await _firebaseService.GetSpendingsAsync(sanitizedEmail); //<--- Pobranie
wydatków.
00053
00054             SpendingsListView.ItemsSource = spendings; //<--- Przypisanie danych do ListView.
00055             SpendingsListView.IsVisible = true;
00056
00057             LoadingIndicator.IsRunning = false; //<--- Ukrycie wskanika ładowania.
00058             LoadingIndicator.IsVisible = false;
00059         }
00060
00066         private async void OnRefreshButtonClicked(object sender, EventArgs e)
00067         {
00068             await LoadSpendings(); //<--- Ponowne załadowanie wydatków.
00069         }
00070     }
00071 }
```

7.19 Dokumentacja pliku CYD/Services/CoinGeckoService.cs

Komponenty

- class [CYD.Services.CoinGeckoService](#)
- class [CYD.Services.CryptoCurrency](#)

Przestrzenie nazw

- namespace [CYD](#)
- namespace [CYD.Services](#)

7.20 CoinGeckoService.cs

[Idź do dokumentacji tego pliku.](#)

```
00001 using System;
00002 using System.Collections.Generic;
00003 using System.Linq;
00004 using System.Text;
00005 using System.Threading.Tasks;
00006 using System.Net.Http.Json;
00007
00008 namespace CYD.Services
00009 {
00010     public class CoinGeckoService
00011     {
00012         private readonly HttpClient _httpClient; ///<--- HTTP client for making requests to CoinGecko
00013         API.
00014
00015         public CoinGeckoService()
00016         {
00017             _httpClient = new HttpClient
00018             {
00019                 BaseAddress = new Uri("https://api.coingecko.com/api/v3/") ///<--- Base address for the
00020                 CoinGecko API.
00021             };
00022         }
00023
00024         public async Task<List<CryptoCurrency>> GetTopCryptocurrenciesAsync(int count = 10)
00025         {
00026             try
00027             {
00028                 // Fetching cryptocurrency data from CoinGecko API.
00029                 var result = await
00030                 _httpClient.GetFromJsonAsync<List<CryptoCurrency>>($"coins/markets?vs_currency=usd&order=market_cap_desc&per_page={count}");
00031                 return result ?? new List<CryptoCurrency>(); ///<--- Return fetched data or an empty
00032                 list if result is null.
00033             }
00034             catch (Exception ex)
00035             {
00036                 Console.WriteLine($"Error fetching data from CoinGecko: {ex.Message}"); ///<--- Log the
00037                 error message.
00038                 return new List<CryptoCurrency>(); ///<--- Return an empty list in case of error.
00039             }
00040         }
00041
00042         public class CryptoCurrency
00043         {
00044             public string Id { get; set; } ///<--- The unique identifier for the cryptocurrency.
00045             public string Symbol { get; set; } ///<--- The ticker symbol for the cryptocurrency.
00046             public string Name { get; set; } ///<--- The name of the cryptocurrency.
00047             public decimal Current_price { get; set; } ///<--- The current price of the cryptocurrency.
00048             public string Image { get; set; } ///<--- The URL of the cryptocurrency's image.
00049         }
00050     }
00051 }
```

7.21 Dokumentacja pliku CYD/Services/CurrencyLayerService.cs

Komponenty

- class [CYD.Services.CurrencyLayerService](#)
- class [CYD.Services.CurrencyLayerService.CurrencyRatesResponse](#)

Przestrzenie nazw

- namespace [CYD](#)
- namespace [CYD.Services](#)

7.22 CurrencyLayerService.cs

[Idź do dokumentacji tego pliku.](#)

```
00001 using System; // <--- Użycie przestrzeni nazw System, potrzebne dla podstawowych klas, takich jak Uri
      czy Exception
00002 using System.Collections.Generic; // <--- Użycie przestrzeni nazw do pracy z kolekcjami, takimi jak
      słowniki i listy
00003 using System.Linq; // <--- Użycie przestrzeni nazw do LINQ, umożliwia operacje na kolekcjach
00004 using System.Net.Http; // <--- Użycie przestrzeni nazw do komunikacji HTTP (pobieranie danych z API)
00005 using System.Net.Http.Json; // <--- Użycie rozszerzenia HttpClient do pracy z danymi w formacie JSON
00006 using System.Text.Json.Serialization; // <--- Użycie przestrzeni nazw do serializacji JSON
00007 using System.Threading.Tasks; // <--- Użycie przestrzeni nazw do operacji asynchronicznych
00008
00009 namespace CYD.Services
00010 {
00011     public class CurrencyLayerService
00012     {
00013         private readonly HttpClient _httpClient; // <--- Obiekt HttpClient, który będzie
      wykorzystywany do wysyłania zapytań HTTP
00014         private const string ApiKey = "42c7f538c6126b3d6b10018c"; // <--- Klucz API, który będzie
      używany do uwierzytelniania zapytań do API
00015         private const string BaseUrl = "https://v6.exchangerate-api.com/v6/"; // <--- Podstawowy URL
      API do pobierania kursów walut
00016
00017         // Konstruktor inicjalizujący HttpClient z podstawowym adresem URL
00018         public CurrencyLayerService()
00019         {
00020             _httpClient = new HttpClient
00021             {
00022                 BaseAddress = new Uri(BaseUrl) // <--- Ustawienie podstawowego URL dla HttpClient
00023             };
00024         }
00025
00026         // Asynchroniczna metoda pobierająca top X walut na podstawie wybranej waluty bazowej
00027         public async Task<Dictionary<string, decimal>> GetTopCurrenciesAsync(string baseCurrency =
      "USD", int count = 5)
00028         {
00029             try
00030             {
00031                 var endpoint = $"{ApiKey}/latest/{baseCurrency}"; // <--- Tworzymy pełny endpoint z
      kluczem API i walutą bazową
00032                 var response = await _httpClient.GetFromJsonAsync<CurrencyRatesResponse>(endpoint); //
      <--- Wysyłamy zapytanie do API i deserializujemy odpowiedź do obiektu CurrencyRatesResponse
00033
00034                 if (response?.ConversionRates != null) // <--- Sprawdzamy, czy odpowiedź zawiera dane
      o kursach walut
00035                 {
00036                     Console.WriteLine("API Request Successful."); // <--- Informujemy, że zapytanie
      API powiodło się
00037
      // Ograniczamy do top X walut
00038                     return response.ConversionRates // <--- Zwracamy konwersje walut
      .Take(count) // <--- Pobieramy tylko pierwsze 'count' walut
00039                     .ToDictionary(kvp => kvp.Key, kvp => kvp.Value); // <--- Zamieniamy wynik na
      słownik
00040
00041                 }
00042
00043                 Console.WriteLine("API response is null or invalid."); // <--- Informujemy o
      problemach z odpowiedzią API
00044                 return new Dictionary<string, decimal>(); // <--- Zwracamy pusty słownik w przypadku
      braku wyników
00045             }
00046             catch (Exception ex) // <--- Łapanie wyjątków, które mogą wystąpić podczas zapytania HTTP
00047             {
00048                 Console.WriteLine($"Error fetching currency rates: {ex.Message}"); // <---
      Wyświetlanie błędu w przypadku niepowodzenia zapytania
00049                 return new Dictionary<string, decimal>(); // <--- Zwracamy pusty słownik w przypadku
      błędu
00050             }
00051         }
00052
00053         // Klasa pomocnicza do deserializacji odpowiedzi API
00054         public class CurrencyRatesResponse
00055         {
00056             [JsonPropertyName("conversion_rates")] // <--- Atrybut pozwalający zmapować pole w
      odpowiedzi JSON na właściwość ConversionRates
00057             public Dictionary<string, decimal> ConversionRates { get; set; } // <--- Słownik
      przechowujący kursy walut (klucz: waluta, wartość: kurs)
00058         }
00059     }
00060 }
```

7.23 Dokumentacja pliku CYD/Services/FirebaseAuthService.cs

Komponenty

- class [FirebaseAuthService](#)

7.24 FirebaseAuthService.cs

Idź do dokumentacji tego pliku.

```

00001 using Firebase.Auth; // <--- Użycie pakietu Firebase.Auth, który zawiera funkcje do autoryzacji
00002 using System; // <--- Użycie przestrzeni nazw System, zapewniającej podstawowe funkcje systemowe
00003 using System.Threading.Tasks; // <--- Użycie przestrzeni nazw Tasków, pozwalających na operacje
    asynchroniczne
00004 using Microsoft.Maui.Storage; // <--- Użycie przestrzeni nazw do pracy z przechowywaniem danych
    aplikacji
00005
00006 public class FirebaseAuthService
00007 {
00008     private readonly FirebaseAuthProvider _authProvider; // <--- Provider do autentykacji z Firebase
00009     private readonly string _firebaseApiKey = "AIzaSyCsPbeDGIi0slQWhIcPr2hI0NShLn7dE7M"; // <--- Klucz
    API dla Firebase
00010     private FirebaseAuthLink _authLink; // <--- Przechowuje dane o zalogowanym użytkowniku
00011
00012     // Konstruktor inicjalizujący usługę Firebase
00013     public FirebaseAuthService()
00014     {
00015         _authProvider = new FirebaseAuthProvider(new FirebaseConfig(_firebaseApiKey)); // <---
    Inicjalizacja FirebaseAuthProvider z kluczem API
00016     }
00017
00018     // -----
00019     // Rejestracja nowego użytkownika
00020     // -----
00021     public async Task<string> RegisterWithEmailAndPasswordAsync(string email, string password)
00022     {
00023         try
00024         {
00025             // Tworzy nowego użytkownika za pomocą podanego e-maila i hasła
00026             _authLink = await _authProvider.CreateUserWithEmailAndPasswordAsync(email, password); //
    <--- Rejestracja użytkownika w Firebase
00027
00028             // Zapisuje email do SecureStorage
00029             await SecureStorage.SetAsync("user_email", email); // <--- Zapisuje e-mail użytkownika w
    SecureStorage
00030
00031             // Pobiera token Firebase
00032             string token = _authLink.FirebaseToken; // <--- Pobiera token autentykacji Firebase
00033             return token; // Zwraca token, jeśli rejestracja się powiedzie
00034         }
00035         catch (FirebaseAuthException ex)
00036         {
00037             // Obsługuje specyficzne błędy Firebase
00038             return $"Error: {ex.Reason}"; // <--- Zwraca błąd, jeśli wystąpił problem z rejestracją
00039         }
00040         catch (Exception ex)
00041         {
00042             // Obsługuje ogólne błędy
00043             return $"Error: {ex.Message}"; // <--- Zwraca ogólny błąd
00044         }
00045     }
00046
00047     // -----
00048     // Logowanie istniejącego użytkownika
00049     // -----
00050     public async Task<string> SignInWithEmailAndPasswordAsync(string email, string password)
00051     {
00052         try
00053         {
00054             // Loguje użytkownika za pomocą podanego e-maila i hasła
00055             _authLink = await _authProvider.SignInWithEmailAndPasswordAsync(email, password); // <---
    Logowanie użytkownika w Firebase
00056
00057             // Zapisuje email do SecureStorage
00058             await SecureStorage.SetAsync("user_email", email); // <--- Zapisuje e-mail użytkownika w
    SecureStorage
00059
00060             await SecureStorage.SetAsync("user_uid", _authLink.User.LocalId); // <--- Zapisuje UID
    użytkownika w SecureStorage

```

```

00061         // Pobiera token Firebase
00062         string token = _authLink.FirebaseToken; // <--- Pobiera token autentykacji Firebase
00063         return token; // Zwraca token, jeśli logowanie się powiedzie
00064     }
00065     catch (Exception ex)
00066     {
00067         // Zwraca błąd, jeśli logowanie nie powiedzie się
00068         return $"Error: {ex.Message}"; // <--- Zwraca błąd, jeśli wystąpił problem z logowaniem
00069     }
00070 }
00071
00072 // -----
00073 // Pobiera adres e-mail aktualnie zalogowanego użytkownika
00074 // -----
00075 public string GetCurrentUserEmail()
00076 {
00077     return _authLink?.User.Email; // <--- Zwraca e-mail użytkownika lub null, jeśli użytkownik nie
jest zalogowany
00078 }
00079
00080 // Asynchronicznie pobiera e-mail aktualnie zalogowanego użytkownika
00081 public async Task<string> GetCurrentUserEmailAsync()
00082 {
00083     var email = await SecureStorage.GetAsync("user_email"); // <--- Pobiera zapisany e-mail z
SecureStorage
00084     return email ?? "Nie jesteś zalogowany"; // <--- Zwraca e-mail lub komunikat, że użytkownik
nie jest zalogowany
00085 }
00086
00087 // Asynchronicznie pobiera UID (identyfikator użytkownika) aktualnie zalogowanego użytkownika
00088 public async Task<string> GetCurrentUserIdAsync()
00089 {
00090     // Zwróci UID użytkownika (LocalId)
00091     var email = await SecureStorage.GetAsync("user_email"); // <--- Pobiera e-mail z SecureStorage
00092     if (string.IsNullOrEmpty(email)) // <--- Sprawdza, czy e-mail jest pusty (czy użytkownik nie
jest zalogowany)
00093     {
00094         return string.Empty; // <--- Jeśli użytkownik nie jest zalogowany, zwraca pusty ciąg
00095     }
00096
00097     // Logowanie się na podstawie zapisanej lokalnie wiadomości e-mail
00098     var user = await _authProvider.SignInWithEmailAndPasswordAsync(email, "userPassword"); // <---
Loguje się na podstawie zapisanego e-maila (tu domyślne hasło "userPassword")
00099     return user.User.LocalId; // <--- Zwraca UID (LocalId) użytkownika
00100 }
00101
00102 // -----
00103 // Wylogowanie użytkownika
00104 // -----
00105 public async Task<bool> IsUserLoggedInAsync()
00106 {
00107     var email = await SecureStorage.GetAsync("user_email"); // <--- Pobiera zapisany e-mail
użytkownika z SecureStorage
00108     return !string.IsNullOrEmpty(email); // <--- Sprawdza, czy e-mail jest zapisany (czy
użytkownik jest zalogowany)
00109 }
00110
00111 // Wylogowanie użytkownika
00112 public async Task LogoutAsync()
00113 {
00114     // Usuwa e-mail użytkownika z SecureStorage podczas wylogowania
00115     SecureStorage.Remove("user_email"); // <--- Usuwa e-mail z SecureStorage, aby wylogować
użytkownika
00116 }
00117 }

```

7.25 Dokumentacja pliku CYD/Services/FirebaseService.cs

Komponenty

- class [CYD.Services.FirebaseDatabaseService](#)
- class [CYD.Services.Spending](#)

Przestrzenie nazw

- namespace [CYD](#)
- namespace [CYD.Services](#)

7.26 FirebaseService.cs

Idź do dokumentacji tego pliku.

```
00001 using Firebase.Database; // <--- Użycie Firebase.Database do komunikacji z bazą danych Firebase
00002 using Firebase.Database.Query; // <--- Użycie zapytań do bazy danych Firebase
00003 using System.Collections.Generic; // <--- Użycie przestrzeni nazw do pracy z kolekcjami, takimi jak
        lista
00004 using System.Threading.Tasks; // <--- Użycie przestrzeni nazw do operacji asynchronicznych
00005
00006 namespace CYD.Services
00007 {
00008     public class FirebaseDatabaseService
00009     {
00010         private readonly FirebaseClient _firebaseClient; // <--- Inicjalizacja klienta Firebase, który
        będzie służył do komunikacji z bazą danych
00011
00012         // Konstruktor inicjalizujący klienta Firebase
00013         public FirebaseDatabaseService()
00014         {
00015             _firebaseClient = new
        FirebaseClient("https://countyourdollars-default-rtdb.europe-west1.firebaseio.com/"); // <---
        Tworzenie klienta Firebase z podanym URL bazy danych
00016         }
00017
00018         // Zapisanie wydatku
00019         // Zaktualizowana metoda SaveSpendingAsync
00020         public async Task SaveSpendingAsync(string category, decimal amount, string userEmail)
00021         {
00022             // Zamieniamy specjalne znaki w e-mailu na bezpieczne dla Firebase
00023             var sanitizedEmail = userEmail.Replace("@", "-at-").Replace(".", "-dot-"); // <---
        Bezpieczne przekształcenie e-maila użytkownika na format odpowiedni dla Firebase
00024
00025             // Tworzymy obiekt wydatku
00026             var spending = new Spending
00027             {
00028                 Category = category, // <--- Przypisanie kategorii wydatku
00029                 Amount = amount // <--- Przypisanie kwoty wydatku
00030             };
00031
00032             // Tworzymy ścieżkę do zapisania wydatku w Firebase
00033             var refPath = $"users/{sanitizedEmail}/spendings"; // <--- Tworzenie ścieżki do zapisania
        danych wydatku w Firebase
00034             await _firebaseClient.Child(refPath).PostAsync(spending); // <--- Zapisanie obiektu
        wydatku w Firebase w odpowiedniej ścieżce
00035         }
00036
00037         // Pobranie wydatków użytkownika na podstawie e-maila
00038         public async Task<List<Spending>> GetSpendingAsync(string userEmail)
00039         {
00040             // Zamieniamy e-mail użytkownika na format bezpieczny dla Firebase
00041             var sanitizedEmail = userEmail.Replace("@", "-at-").Replace(".", "-dot-"); // <---
        Bezpieczne przekształcenie e-maila użytkownika
00042
00043             // Ścieżka do wydatków użytkownika
00044             var refPath = $"users/{sanitizedEmail}/spendings"; // <--- Ścieżka w bazie danych do
        wydatków użytkownika
00045
00046             // Pobieramy dane z Firebase
00047             var spendings = await _firebaseClient
        .Child(refPath) // <--- Dostęp do węzła w bazie danych, gdzie przechowywane są wydatki
00048             .OnceAsync<Spending>(); // <--- Pobieranie wszystkich wydatków w postaci obiektów
        Spending
00049
00050             // Konwertujemy wyniki na listę wydatków
00051             var spendingList = new List<Spending>(); // <--- Tworzymy pustą listę do przechowywania
        pobranych wydatków
00052             foreach (var item in spendings)
00053             {
00054                 spendingList.Add(item.Object); // <--- Dodajemy każdy wydatek do listy
00055             }
00056
00057             return spendingList; // <--- Zwracamy listę wydatków
00058         }
00059     }
00060
00061 }
00062
00063 // Klasa reprezentująca wydatek
00064 public class Spending
00065 {
00066     public string Category { get; set; } // <--- Kategoria wydatku (np. "zakupy", "jedzenie")
00067     public decimal Amount { get; set; } // <--- Kwota wydatku
00068 }
00069 }
```

Skorowidz

- [_authLink](#)
 - [FirebaseAuthService, 26](#)
 - [_authProvider](#)
 - [FirebaseAuthService, 26](#)
 - [_authService](#)
 - [CYD.AppShell, 17](#)
 - [CYD.Pages.LoginPage, 31](#)
 - [CYD.Pages.SignUpPage, 34](#)
 - [CYD.Pages.SpendingsPage, 36](#)
 - [_coinGeckoService](#)
 - [CYD.Pages.AboutPage, 12](#)
 - [_currencyLayerService](#)
 - [CYD.App, 14](#)
 - [_firebaseApiKey](#)
 - [FirebaseAuthService, 26](#)
 - [_firebaseClient](#)
 - [CYD.Services.FirebaseDatabaseService, 28](#)
 - [_firebaseService](#)
 - [CYD.Pages.SpendingsPage, 36](#)
 - [_httpClient](#)
 - [CYD.Services.CoinGeckoService, 19](#)
 - [CYD.Services.CurrencyLayerService, 21](#)
 - [_navigationPage](#)
 - [CYD.App, 14](#)
- [AboutPage](#)
 - [CYD.Pages.AboutPage, 12](#)
- [Amount](#)
 - [CYD.Services.Spending, 35](#)
- [ApiKey](#)
 - [CYD.Services.CurrencyLayerService, 21](#)
- [App](#)
 - [CYD.App, 13](#)
- [AppShell](#)
 - [CYD.AppShell, 16](#)
- [AppShell.xaml.cs](#)
 - [Font, 38](#)
- [BaseUrl](#)
 - [CYD.Services.CurrencyLayerService, 21](#)
- [Button_Clicked_2](#)
 - [CYD.Pages.LoginPage, 30](#)
- [Category](#)
 - [CYD.Services.Spending, 35](#)
- [CoinGeckoService](#)
 - [CYD.Services.CoinGeckoService, 18](#)
- [ConversionRates](#)
 - [CYD.Services.CurrencyLayerService.CurrencyRatesResponse, 24](#)
- [CreateMauiApp](#)
 - [CYD.MauiProgram, 32](#)
- [CreateWindow](#)
 - [CYD.App, 13](#)
- [Cryptocurrencies](#)
 - [CYD.Pages.AboutPage, 12](#)
- [CurrencyLayerService](#)
 - [CYD.Services.CurrencyLayerService, 20](#)
- [CurrencyPage](#)
 - [CYD.Pages.CurrencyPage, 22](#)
- [CurrencyRates](#)
 - [CYD.App, 15](#)
 - [CYD.Pages.CurrencyPage, 23](#)
- [Current_price](#)
 - [CYD.Services.CryptoCurrency, 19](#)
- [CYD, 9](#)
- [CYD.App, 12](#)
 - [_currencyLayerService, 14](#)
 - [_navigationPage, 14](#)
 - [App, 13](#)
 - [CreateWindow, 13](#)
 - [CurrencyRates, 15](#)
 - [LoadInitialDataAsync, 14](#)
 - [SubscribeToLoginSuccessful, 14](#)
- [CYD.AppShell, 15](#)
 - [_authService, 17](#)
 - [AppShell, 16](#)
 - [DisplaySnackbarAsync, 16](#)
 - [DisplayToastAsync, 16](#)
 - [OnLogoutButtonClicked, 17](#)
 - [SfSegmentedControl_SelectionChanged, 17](#)
- [CYD.MauiProgram, 32](#)
 - [CreateMauiApp, 32](#)
- [CYD.Pages, 9](#)
- [CYD.Pages.AboutPage, 11](#)
 - [_coinGeckoService, 12](#)
 - [AboutPage, 12](#)
 - [Cryptocurrencies, 12](#)
 - [LoadDataAsync, 12](#)
- [CYD.Pages.CurrencyPage, 21](#)
 - [CurrencyPage, 22](#)
 - [CurrencyRates, 23](#)
 - [LoadCurrencyRates, 22](#)
 - [LoadCurrencyRatesAsync, 22](#)
- [CYD.Pages.CurrencyRate, 23](#)
 - [DisplayValue, 23](#)
 - [Key, 23](#)
 - [Value, 23](#)
- [CYD.Pages.ForgotPassword, 28](#)

- ForgotPassword, 28
- OnLoginTapped, 29
- OnSendResetLinkClicked, 29
- CYD.Pages.LoginPage, 29
 - _authService, 31
 - Button_Clicked_2, 30
 - fingerprint, 31
 - LoginPage, 30
 - LoginSuccessful, 32
 - OnForgotPasswordTapped, 30
 - OnLoginButtonClicked, 31
 - OnSignUpTapped, 31
- CYD.Pages.SignUpPage, 33
 - _authService, 34
 - OnLoginTapped, 33
 - OnSignUpClicked, 34
 - SignUpPage, 33
- CYD.Pages.SpendingsPage, 35
 - _authService, 36
 - _firebaseService, 36
 - LoadSpendings, 36
 - OnRefreshButtonClicked, 36
 - SpendingsPage, 36
- CYD.Services, 10
- CYD.Services.CoinGeckoService, 18
 - _httpClient, 19
 - CoinGeckoService, 18
 - GetTopCryptocurrenciesAsync, 18
- CYD.Services.CryptoCurrency, 19
 - Current_price, 19
 - Id, 19
 - Image, 19
 - Name, 19
 - Symbol, 20
- CYD.Services.CurrencyLayerService, 20
 - _httpClient, 21
 - ApiKey, 21
 - BaseUrl, 21
 - CurrencyLayerService, 20
 - GetTopCurrenciesAsync, 21
- CYD.Services.CurrencyLayerService.CurrencyRatesResponse, 24
 - ConversionRates, 24
- CYD.Services.FirebaseDatabaseService, 27
 - _firebaseClient, 28
 - FirestoreDatabaseService, 27
 - GetSpendingsAsync, 27
 - SaveSpendingAsync, 27
- CYD.Services.Spending, 34
 - Amount, 35
 - Category, 35
- CYD/App.xaml.cs, 37
- CYD/AppShell.xaml.cs, 38, 39
- CYD/MauiProgram.cs, 40
- CYD/Pages/AboutPage.xaml.cs, 41
- CYD/Pages/CurrencyPage.xaml.cs, 42
- CYD/Pages/ForgotPassword.xaml.cs, 43
- CYD/Pages/LoginPage.xaml.cs, 44
- CYD/Pages/SignUpPage.xaml.cs, 45
- CYD/Pages/SpendingsPage.xaml.cs, 46, 47
- CYD/Services/CoinGeckoService.cs, 47, 48
- CYD/Services/CurrencyLayerService.cs, 48, 49
- CYD/Services/FirebaseAuthService.cs, 50
- CYD/Services/FirebaseService.cs, 51, 52
- DisplaySnackBarAsync
 - CYD.AppShell, 16
- DisplayToastAsync
 - CYD.AppShell, 16
- DisplayValue
 - CYD.Pages.CurrencyRate, 23
- fingerprint
 - CYD.Pages.LoginPage, 31
- FirestoreAuthService, 24
 - _authLink, 26
 - _authProvider, 26
 - _firebaseApiKey, 26
 - FirestoreAuthService, 25
 - GetCurrentUserEmail, 25
 - GetCurrentUserEmailAsync, 25
 - GetCurrentUserIdAsync, 25
 - IsUserLoggedInAsync, 25
 - LogoutAsync, 25
 - RegisterWithEmailAndPasswordAsync, 26
 - SignInWithEmailAndPasswordAsync, 26
- FirestoreDatabaseService
 - CYD.Services.FirebaseDatabaseService, 27
- Font
 - AppShell.xaml.cs, 38
- ForgotPassword
 - CYD.Pages.ForgotPassword, 28
- GetCurrentUserEmail
 - FirestoreAuthService, 25
- GetCurrentUserEmailAsync
 - FirestoreAuthService, 25
- GetCurrentUserIdAsync
 - FirestoreAuthService, 25
- GetSpendingsAsync
 - CYD.Services.FirebaseDatabaseService, 27
- GetTopCryptocurrenciesAsync
 - CYD.Services.CoinGeckoService, 18
- GetTopCurrenciesAsync
 - CYD.Services.CurrencyLayerService, 21
- Id
 - CYD.Services.CryptoCurrency, 19
- Image
 - CYD.Services.CryptoCurrency, 19
- IsUserLoggedInAsync
 - FirestoreAuthService, 25
- Key
 - CYD.Pages.CurrencyRate, 23
- LoadCurrencyRates
 - CYD.Pages.CurrencyPage, 22

LoadCurrencyRatesAsync
 CYD.Pages.CurrencyPage, [22](#)

LoadDataAsync
 CYD.Pages.AboutPage, [12](#)

LoadInitialDataAsync
 CYD.App, [14](#)

LoadSpending
 CYD.Pages.SpendingPage, [36](#)

LoginPage
 CYD.Pages.LoginPage, [30](#)

LoginSuccessful
 CYD.Pages.LoginPage, [32](#)

LogoutAsync
 FirebaseAuthService, [25](#)

Name
 CYD.Services.CryptoCurrency, [19](#)

OnForgotPasswordTapped
 CYD.Pages.LoginPage, [30](#)

OnLoginButtonClicked
 CYD.Pages.LoginPage, [31](#)

OnLoginTapped
 CYD.Pages.ForgotPassword, [29](#)
 CYD.Pages.SignUpPage, [33](#)

OnLogoutButtonClicked
 CYD.AppShell, [17](#)

OnRefreshButtonClicked
 CYD.Pages.SpendingPage, [36](#)

OnSendResetLinkClicked
 CYD.Pages.ForgotPassword, [29](#)

OnSignUpClicked
 CYD.Pages.SignUpPage, [34](#)

OnSignUpTapped
 CYD.Pages.LoginPage, [31](#)

RegisterWithEmailAndPasswordAsync
 FirebaseAuthService, [26](#)

SaveSpendingAsync
 CYD.Services.FirebaseDatabaseService, [27](#)

SfSegmentedControl_SelectionChanged
 CYD.AppShell, [17](#)

SignInWithEmailAndPasswordAsync
 FirebaseAuthService, [26](#)

SignUpPage
 CYD.Pages.SignUpPage, [33](#)

SpendingPage
 CYD.Pages.SpendingPage, [36](#)

SubscribeToLoginSuccessful
 CYD.App, [14](#)

Symbol
 CYD.Services.CryptoCurrency, [20](#)

Value
 CYD.Pages.CurrencyRate, [23](#)