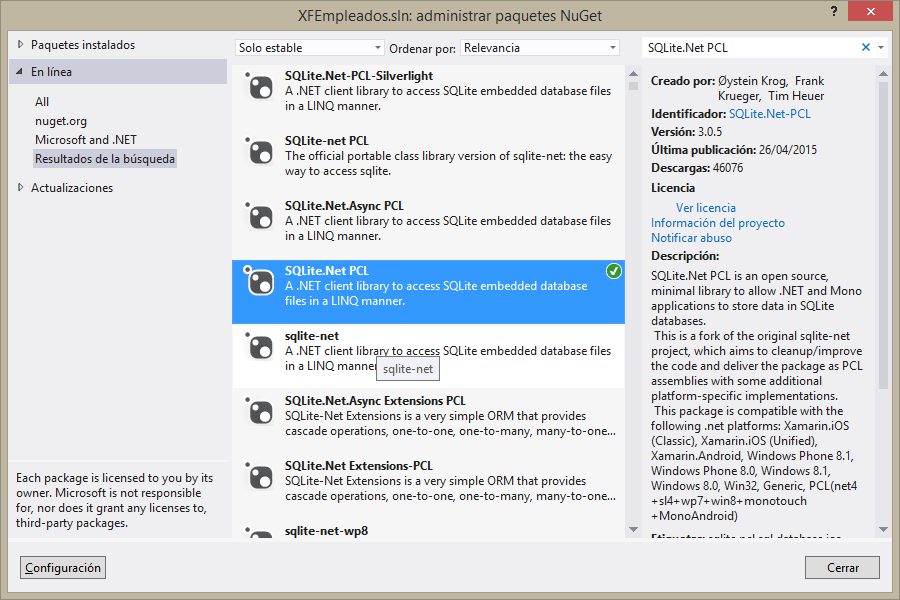
Xamarin.Forms

Guía para construir ejemplo de SQLite

Juan Carlos Zuluaga – <http://zulu-software.com>

1. Cree el proyecto portable de Xamarin.Forms
2. Agregue a todos los proyectos el paquete SQLite.Net PCL



1. En el proyecto compartido agregue la clase : **Employee.cs** con:

using SQLite.Net.Attributes;

using System;

namespace XConceptThree

{

public class Employee

{

[PrimaryKey, AutoIncrement]

public int EmployeeId { get; set; }

public string FirstName { get; set; }

public string LastName { get; set; }

public DateTime ContractDate { get; set; }

public decimal Salary { get; set; }

public bool Active { get; set; }

public string FullName { get { return string.Format("{0} {1}", FirstName, LastName); } }

}

}

1. En el proyecto compartido agregue la interfaz **IConfig.cs** con el siguiente código:

using SQLite.Net.Interop;

namespace XConceptThree

{

public interface IConfig

{

string DirectoryDB { get; }

ISQLitePlatform Platform { get; }

}

}

1. En el proyecto iOS implemente la interfaz **IConfig** con el siguiente código:

using SQLite.Net.Interop;

using System;

using Xamarin.Forms;

[assembly: Dependency(typeof(XConceptThree.iOS.Config))]

namespace XConceptThree.iOS

{

public class Config : IConfig

{

private string directoryDB;

private ISQLitePlatform platform;

public string DirectoryDB

{

get

{

if (string.IsNullOrEmpty(directoryDB))

{

var directorio = System.Environment.GetFolderPath(Environment.SpecialFolder.Personal);

directoryDB = System.IO.Path.Combine(directorio, "..", "Library");

}

return directoryDB;

}

}

public ISQLitePlatform Platform

{

get

{

if (platform == null)

{

platform = new SQLite.Net.Platform.XamarinIOS.SQLitePlatformIOS();

}

return platform;

}

}

}

}

1. En el proyecto Droid implemente la interfaz **IConfig** con el siguiente código:

using SQLite.Net.Interop;

using Xamarin.Forms;

[assembly: Dependency(typeof(XConceptThree.Droid.Config))]

namespace XConceptThree.Droid

{

public class Config : IConfig

{

private string directoryDB;

private ISQLitePlatform platform;

public string DirectoryDB

{

get

{

if (string.IsNullOrEmpty(directoryDB))

{

directoryDB = System.Environment.GetFolderPath(System.Environment.SpecialFolder.Personal);

}

return directoryDB;

}

}

public ISQLitePlatform Platform

{

get

{

if (platform == null)

{

platform = new SQLite.Net.Platform.XamarinAndroid.SQLitePlatformAndroid();

}

return platform;

}

}

}

}

1. En el proyecto Windows Phone implemente la interfaz **IConfig** con el siguiente código:

using SQLite.Net.Interop;

using Windows.Storage;

using Xamarin.Forms;

[assembly: Dependency(typeof(XConceptThree.Windows.Config))]

namespace XConceptThree.Windows

{

public class Config : IConfig

{

private string directoryDB;

private ISQLitePlatform platform;

public string DirectoryDB

{

get

{

if (string.IsNullOrEmpty(directoryDB))

{

directoryDB = ApplicationData.Current.LocalFolder.Path;

}

return directoryDB;

}

}

public ISQLitePlatform Platform

{

get

{

if (platform == null)

{

platform = new SQLite.Net.Platform.WinRT.SQLitePlatformWinRT();

}

return platform;

}

}

}

}

1. En el proyecto compartido agregue la clase **DataAccess.cs** con el siguiente código:

using SQLite.Net;

using System;

using System.Collections.Generic;

using System.Linq;

using Xamarin.Forms;

namespace XConceptThree

{

public class DataAccess : IDisposable

{

private SQLiteConnection connection;

public DataAccess()

{

var config = DependencyService.Get<IConfig>();

connection = new SQLiteConnection(config.Platform,

System.IO.Path.Combine(config.DirectoryDB, "Employees.db3"));

connection.CreateTable<Employee>();

}

public void Insert<T>(T model)

{

connection.Insert(model);

}

public void Update<T>(T model)

{

connection.Update(model);

}

public void Delete<T>(T model)

{

connection.Delete(model);

}

public T Find<T>(int id) where T : class

{

return connection.Table<T>().FirstOrDefault(model => model.GetHashCode() == id);

}

public T First<T>() where T : class

{

return connection.Table<T>().FirstOrDefault();

}

public List<T> GetList<T>() where T : class

{

return connection.Table<T>().ToList();

}

public void Dispose()

{

connection.Dispose();

}

}

}

1. Agregue la página: **HomePage.xaml** con:

<?xml version="1.0" encoding="utf-8" ?>

<ContentPage xmlns="http://xamarin.com/schemas/2014/forms"

xmlns:x="http://schemas.microsoft.com/winfx/2009/xaml"

x:Class="XConceptThree.HomePage"

Title="Employees">

<ScrollView>

<StackLayout Spacing="10">

<Entry x:Name="firstNameEntry"

Placeholder="First Name" />

<Entry x:Name="lastNameEntry"

Placeholder="Last Name" />

<Entry x:Name="salaryEntry"

Placeholder="Salary"

Keyboard="Numeric" />

<StackLayout Orientation="Horizontal">

<Label Text="Contract Date: "

VerticalOptions="Center" />

<DatePicker x:Name="contractDateDatePicker"

HorizontalOptions="FillAndExpand"

VerticalOptions="Center"/>

<Label Text="Active: "

HorizontalOptions="End"

VerticalOptions="Center" />

<Switch x:Name="activeSwitch"

VerticalOptions="Center"

IsToggled="True" />

</StackLayout>

<Button x:Name="addButton"

Text="Add"

BackgroundColor="Navy"

TextColor="White" />

<ListView x:Name="employeesListView" />

</StackLayout>

</ScrollView>

</ContentPage>

1. Cambie el inicio de la aplicación **App.cs** por:

public App()

{

MainPage = new NavigationPage(new HomePage());

}

Pruebe lo que lleve hasta el momento

|  |  |
| --- | --- |
|  |  |

1. Modifique el siguiente código a la página: **HomePage.xaml.cs**

using System;

using Xamarin.Forms;

namespace XConceptThree

{

public partial class HomePage : ContentPage

{

public HomePage()

{

InitializeComponent();

Padding = Device.OnPlatform(

new Thickness(10, 20, 10, 10),

new Thickness(10),

new Thickness(10));

addButton.Clicked += AddButton\_Clicked;

}

private async void AddButton\_Clicked(object sender, EventArgs e)

{

if (string.IsNullOrEmpty(firstNameEntry.Text))

{

await DisplayAlert("Error", "You must enter a first name", "Acept");

firstNameEntry.Focus();

return;

}

if (string.IsNullOrEmpty(lastNameEntry.Text))

{

await DisplayAlert("Error", "You must enter a last name", "Acept");

lastNameEntry.Focus();

return;

}

if (string.IsNullOrEmpty(salaryEntry.Text))

{

await DisplayAlert("Error", "You must enter a salary", "Acept");

salaryEntry.Focus();

return;

}

var employee = new Employee

{

FirstName = firstNameEntry.Text,

LastName = lastNameEntry.Text,

ContractDate = contractDateDatePicker.Date,

Salary = decimal.Parse(salaryEntry.Text),

Active = activeSwitch.IsToggled

};

using (var data = new DataAccess())

{

data.Insert(employee);

employeesListView.ItemsSource = data.GetList<Employee>();

}

firstNameEntry.Text = string.Empty;

lastNameEntry.Text = string.Empty;

salaryEntry.Text = string.Empty;

contractDateDatePicker.Date = DateTime.Now;

activeSwitch.IsToggled = true;

await DisplayAlert("Message", "Employee added", "Acept");

}

protected override void OnAppearing()

{

base.OnAppearing();

using (var data = new DataAccess())

{

employeesListView.ItemsSource = data.GetList<Employee>();

}

}

}

}

Pruebe lo que lleve hasta el momento

1. Agregue la clase **EmployeeCell.cs** con:

using Xamarin.Forms;

namespace XConceptThree

{

public class EmployeeCell : ViewCell

{

public EmployeeCell()

{

var employeeIdLabel = new Label

{

HorizontalTextAlignment = TextAlignment.End,

HorizontalOptions = LayoutOptions.Start,

FontSize = 20,

FontAttributes = FontAttributes.Bold,

};

employeeIdLabel.SetBinding(Label.TextProperty, new Binding("EmployeeId"));

var fullNameLabel = new Label

{

FontSize = 20,

FontAttributes = FontAttributes.Bold,

HorizontalOptions = LayoutOptions.StartAndExpand

};

fullNameLabel.SetBinding(Label.TextProperty, new Binding("FullName"));

var contractDateLabel = new Label

{

HorizontalOptions = LayoutOptions.StartAndExpand

};

contractDateLabel.SetBinding(Label.TextProperty, new Binding("ContractDate", stringFormat: "{0:yyyy/MM/dd}"));

var salaryLabel = new Label

{

HorizontalTextAlignment = TextAlignment.End,

HorizontalOptions = LayoutOptions.StartAndExpand

};

salaryLabel.SetBinding(Label.TextProperty, new Binding("Salary", stringFormat: "{0:C2}"));

var activeSwitch = new Switch

{

IsEnabled = false,

HorizontalOptions = LayoutOptions.End

};

activeSwitch.SetBinding(Switch.IsToggledProperty, new Binding("Active"));

var line1 = new StackLayout

{

Orientation = StackOrientation.Horizontal,

Children = {

employeeIdLabel, fullNameLabel

},

};

var line2 = new StackLayout

{

Orientation = StackOrientation.Horizontal,

Children = {

contractDateLabel, salaryLabel, activeSwitch,

},

};

View = new StackLayout

{

Orientation = StackOrientation.Vertical,

Children = {

line1, line2,

},

};

}

}

}

1. Agregue estas líneas al constructor del **HomePage.xaml.cs**:

employeesListView.ItemTemplate = new DataTemplate(typeof(EmployeeCell));

employeesListView.RowHeight = 70;

Pruebe el impacto de los cambios

|  |  |
| --- | --- |
|  |  |

1. Agregue la página **EditPage.xaml** con:

<?xml version="1.0" encoding="utf-8" ?>

<ContentPage xmlns="http://xamarin.com/schemas/2014/forms"

xmlns:x="http://schemas.microsoft.com/winfx/2009/xaml"

x:Class="XConceptThree.EditPage"

Title="Edit Employee">

<ScrollView>

<StackLayout Padding="10" Spacing="10">

<Entry x:Name="firstNameEntry"

Placeholder="First Name" />

<Entry x:Name="lastNameEntry"

Placeholder="Last Name" />

<Entry x:Name="salaryEntry"

Placeholder="Salary"

Keyboard="Numeric" />

<StackLayout Orientation="Horizontal">

<Label Text="Contract Date: "

VerticalOptions="Center" />

<DatePicker x:Name="contractDateDatePicker" />

<Label Text="Active: "

VerticalOptions="Center" />

<Switch x:Name="activeSwitch"

VerticalOptions="Center"

IsToggled="True" />

</StackLayout>

<StackLayout Orientation="Horizontal">

<Button x:Name="updateButton"

Text="Update"

BackgroundColor="Teal"

TextColor="White"

HorizontalOptions="FillAndExpand"

VerticalOptions="Center" />

<Button x:Name="deleteButton"

Text="Delete"

BackgroundColor="Red"

TextColor="White"

HorizontalOptions="FillAndExpand"

VerticalOptions="Center" />

</StackLayout>

</StackLayout>

</ScrollView>

</ContentPage>

1. Agregue el siguiente código a **EditPage.xaml.cs**:

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using Xamarin.Forms;

namespace XConceptThree

{

public partial class EditPage : ContentPage

{

private Employee employee;

public EditPage(Employee employee)

{

InitializeComponent();

Padding = Device.OnPlatform(

new Thickness(10, 20, 10, 10),

new Thickness(10),

new Thickness(10));

this.employee = employee;

firstNameEntry.Text = employee.FirstName;

lastNameEntry.Text = employee.LastName;

contractDateDatePicker.Date = employee.ContractDate;

salaryEntry.Text = employee.Salary.ToString();

activeSwitch.IsToggled = employee.Active;

updateButton.Clicked += UpdateButton\_Clicked;

deleteButton.Clicked += DeleteButton\_Clicked;

}

private async void DeleteButton\_Clicked(object sender, EventArgs e)

{

var rta = await DisplayAlert("Confirm", "Are you sure to delete the record?", "Yes", "No");

if (!rta)

{

return;

}

using (var data = new DataAccess())

{

data.Delete(employee);

}

await DisplayAlert("Message", "The record was deleted", "Acept");

await Navigation.PopAsync();

}

private async void UpdateButton\_Clicked(object sender, EventArgs e)

{

if (string.IsNullOrEmpty(firstNameEntry.Text))

{

await DisplayAlert("Error", "You must enter a first name", "Acept");

firstNameEntry.Focus();

return;

}

if (string.IsNullOrEmpty(lastNameEntry.Text))

{

await DisplayAlert("Error", "You must enter a last name", "Acept");

lastNameEntry.Focus();

return;

}

if (string.IsNullOrEmpty(salaryEntry.Text))

{

await DisplayAlert("Error", "You must enter a salary", "Acept");

salaryEntry.Focus();

return;

}

employee.FirstName = firstNameEntry.Text;

employee.LastName = lastNameEntry.Text;

employee.Salary = decimal.Parse(salaryEntry.Text);

employee.ContractDate = contractDateDatePicker.Date;

employee.Active = activeSwitch.IsToggled;

using (var data = new DataAccess())

{

data.Update(employee);

}

await DisplayAlert("Message", "The record was updated", "Acept");

await Navigation.PopAsync();

}

}

}

1. Agregue la siguiente línea al constructor de la clase **HomePage.xaml.cs**:

employeesListView.ItemSelected += EmployeesListView\_ItemSelected;

1. Agregue el siguiente método a la clase **HomePage.xaml.cs**:

private async void EmployeesListView\_ItemSelected(object sender, SelectedItemChangedEventArgs e)

{

await Navigation.PushAsync(new EditPage((Employee)e.SelectedItem));

}

Pruebe como quedo la aplicación finalizada

|  |  |
| --- | --- |
|  |  |