Lecture 6 – Data Access

Mr. Yousif Garabet Arshak
Computer Science Department
University of Zakho
yousif.arshak@uoz.edu.krd

Outlines

- **Introduction**
- **Application Property**
- **Files**
- **Local Databases**

Introduction

- **Application Property**

  Property is a dictionary that hold object in it and it can be access after restarting application, it's better to use it for saving simple variables.

- **Files**

  File handling with Xamarin.Forms can be achieved using code in a .NET Standard library, or by using embedded resources.

- **Local Databases**

  Xamarin.Forms supports database-driven applications using the SQLite database engine, which makes it possible to load and save objects in shared code. This lecture describes how Xamarin.Forms applications can read and write data to a local SQLite database using SQLite.Net.

File Handling in Xamarin.Forms

- *File handling with Xamarin.Forms can be achieved using code in a .NET Standard library, or by using embedded resources.*
- Xamarin.Forms code runs on multiple platforms - each of which has its own filesystem. Previously, this meant that reading and writing files was most easily performed using the native file APIs on each platform. Alternatively, embedded resources are a simpler solution to distribute data files with an app. However, with .NET Standard 2.0 and over it's possible to share file access code in .NET Standard libraries.

Saving and Loading Files

- The System.IO classes can be used to access the file system on each platform. The File class lets you create, delete, and read files, and the Directory class allows you to create, delete, or enumerate the contents of directories. You can also use the Stream subclasses, which can provide a greater degree of control over file operations (such as compression or position search within a file).
- A text file can be written using the File.WriteAllText method:

  File.WriteAllText(fileName, text);
- A text file can be read using the File.ReadAllText method:

  string text = File.ReadAllText(fileName);
- File.Exists method determines whether the specified file exists:

  bool doesExist = File.Exists(fileName);

- The path of the file on each platform can be determined from a .NET Standard library by using a value of the Environment.SpecialFolder enumeration as the first argument to the Environment.GetFolderPath method. This can then be combined with a filename with the Path.Combine method:

```
string fileName =
Path.Combine(Environment.GetFolderPath(Environment.SpecialFolder.LocalApp
licationData), “file.txt");
```

Xamarin.Forms Local Databases

- The SQLite database engine allows Xamarin.Forms applications to load and save data objects in shared code. The sample application uses a SQLite database table to store todo items.
- Integrate SQLite.NET into mobile apps by following these steps:
  1. Install the NuGet package.
  2. Configure constants.
  3. Create a database access class.
  4. Access data in Xamarin.Forms.
  5. Advanced configuration.

Install the SQLite NuGet package

- Use the NuGet package manager to search for **sqlite-net-pcl** and add the latest version to the shared code project.
- There are a number of NuGet packages with similar names. The correct package has these attributes:
  - **ID:** sqlite-net-pcl
  - **Authors:** SQLite-net
  - **Owners:** praeclarum
  - **NuGet link:** sqlite-net-pcl

Any Questions?