

5-10テキストファイルに保存する。

ボタンを配置するツールバーはTextEditorにモディファイアのtoolbarを追加して作る。

ツールバーにボタンを追加するには,toolbarのコンテンツとしてToolbarItemを作る。

ToolbarItem()の引数placementに.navigationBarTrailingを指定するとツールバーの右端に寄った位置に表示される。

編集を終了してキーボードを下げるにはUIApplicationクラスをextensionで拡張子したendEditing()で実行する。

ツールバーに追加した保存ボタンではsaveText(入力されたパラメータ,保存ファイル名)=saveText(theText,"sample.txt"),定義された\$theTextに文字列を保存し他ものをsaveTextの第一引数に渡し、ファイル名を第二引数に書く。ファイル名だけでは保存できないのでユーザ定義関数のdocURL(filename)を実行する。保存するURLを取得して、そのURLのパス(url.path)に書き込まれる。

URLが得られずにdocURLがnilを返す場合があるのでguard let-elseによるオプションバインディングでurlを確定している。

例外処理 do-try-catchの中でデータ保存を実行する。

テキストデータの保存はtextData.write()で行う。

write()の挙動

引数toFile=指定したパスのファイルにテキストデータを保存。指定したファイルがない場合は引数atomically=ファイル書き出し中にファイルが壊れないように一時ファイルを作って書き込

このexData.write()はdo-try-catchという構文の中で実行されている。エラーをキャッチして、catchに書いたステートメントを実行することで処理を異常処理せずに実行する。

docURL()定義:保存ファイルのパスを作る

データを保存するファイルパスのURLはユーザー定義したdocURL(fileName)で取得。ファイルはiPhoneデバイス内のどこでも保存できるというわけではなく、その場所は目的に応じて決まっている。ユーザーが読み書きするファイルはDocumentsフォルダ内に保存する。DocumentフォルダまでのURLはFileManager.defaultでfileManagerオブジェクトを作り、fileManager.url()で取得することができる。

DocumentフォルダまでのURLを得られたならば,appendingPathComponent(fileName)で保存ファイル名を追加したURLを作り,そのURLをdocURL(fileName)の値として返す。

ここでもfileManager.urlになる可能性があるので、do-try-catchの例外処理の中で実行する。もしエラーになったならばURLではなくnilを返す。