# Simple Storage Example

Using Xamarin.Essentials allows access to the SecureStorage mechanism for saving string key value pairs.

App class: using Xamarin.Essentials;

We’ll be creating a quick app that will tie the state of a switch object to the value within the SecureStorage.

Add an asynchronous method that will initially read what the value is for the key. If no value is set will set to “false”, otherwise it will set the value of the switch toggled property to either true or false. The switch will be initialized later:

bool lightOn;

Switch lightSwitch;

async void isLightOn()

{

string result = await SecureStorage.GetAsync("light");

if (result != null)

{

lightOn = result.Equals("false") ? false : true;

}

else

{

await SecureStorage.SetAsync("light", "false"); // give default value

}

}

Create a layout with a Switch that is bound to the property we just created after initialize component):

var label = new Label { Text = "Light?" };

isLightOn(); // runs to get the starting value

lightSwitch = new Switch { IsToggled = lightOn };

lightSwitch.Toggled += (sender, e) =>

{

if (lightSwitch.IsToggled) // saves the setting as we go

{

SecureStorage.SetAsync("light", "true");

}

else

{

SecureStorage.SetAsync("light", "false");

}

};

MainPage = new ContentPage

{

Content = new StackLayout

{

HorizontalOptions = LayoutOptions.CenterAndExpand,

VerticalOptions = LayoutOptions.CenterAndExpand,

Children = { label, lightSwitch },

},

};

Try it out, note switch starts on (default Get). Try changing to off, killing app, then opening again.

## Optional: Binding to a Property of a Different Type

Views and Pages can be bound to a BindingContext which can be any object. We’ll try binding the MainPage’s background color to the light switch’s toggled property. Toggled is Bool and color is a color objet, we need to convert.

MainPage.BindingContext = lightSwitch;

MainPage.SetBinding(VisualElement.BackgroundColorProperty, "IsToggled", converter: new ToggledToColorConverter()); //setting how conversion b/w bool to color

Creating an IValueConverter (just put in same file as App class):

class ToggledToColorConverter : IValueConverter

{

public object Convert(object value, Type targetType, object parameter, CultureInfo culture)

{

return (bool)value ? Color.White : Color.Gray;

}

public object ConvertBack(object value, Type targetType, object parameter, CultureInfo culture)

{

throw new NotImplementedException();

}}