UWP

Background Task

MainPage.xaml

⇒ <button click="btn_clicked" ... />

MainPage.xaml.cs

⇒ private async void btn_clicked(...)

```
{ bool taskRegistered = false
  foreach (var task in BackgroundTaskRegistration.AllTasks)
  {  if (task.Value.Name == "TaskName")
       { taskRegistered = true; break; } }

  if (taskRegistered == false)
  { BackgroundAccessStatus access =
      await BackgroundExecutionManager.RequestAccessAsync();
    switch (access)
    {  case BackgroundAccessStatus.Denied:
         ...... }

    var builder = new BackgroundTaskBuilder();
    builder.Name = "TaskName";
    builder.TaskEntryPoint = typeof(A.B).ToString();   // "B.Run"
    builder.SetTrigger(new TimeTrigger(15, false));    // ⇒ 定时15分钟执行一次.
    BackgroundTaskRegistration t = builder.Register();
    t.Completed += Task_Completed;   // ⇒ 创建同名方法 doSomething.
    t.Progress += Task_Progress;     // ⇒ 创建同名方法 doSomething.
}
```

或

```
var trigger = new ApplicationTrigger();
builder.SetTrigger(trigger);
await trigger.RequestAsync();   (立即执行)
```

[1] go to App.xaml.cs under App() { } → two new methods available:

```
{ this.EnteredBackground += App_EnteredBackground;
  this.LeavingBackground += App_LeavingBackground; }
protected override void OnBackgroundActivated(BackgroundActivatedEventArgs args)
{
    ..... 此Task做的事情.
    base.OnBackgroundActivated(args);
}
```

[2] 添加 TaskEntryPoint.

创建 -> 附加 project : RuntimeComponent [A] → 新组件项目名

① 右击后 project → References → add References... → Solution → 勾上新建的 RuntimeComp.

② 双击后 project → Package.appxmanifest → add "Background Tasks" → ☑ System event → ☑ ExecutableOrStartPageIsRequired => Entry Point: A.B → 新组件 class name

```
namespace [A]
{ public sealed class [B] : IBackgroundTask
  { BackgroundTaskDeferral _deferral = null;   // 用于延时执行其它Task. 否则项目可能终止某一task结束.
    public void Run(IBackgroundTaskInstance taskInstance)
    {
      ..... 此Task 做的事情.
    } } }
```

```
try { dostuff
} catch (Exception ex
{ ....
} finally
{ _deferral.Complet
}
```