

现代操作系统应用开发实验报告

姓名：张晓帆

学号：16340293

实验名称：实验 2

一、参考资料

<https://msdn.microsoft.com/zh-cn/library/system.runtime.serialization.collectiondatacontractattribute.aspx>

<https://docs.microsoft.com/zh-cn/windows/uwp/design/shell/tiles-and-notifications/create-adaptive-tiles>

每周老师给的 PPT，以及相应的作业要求，和视频文件。

• ImageSource -> Bitmap

```
1 // ImageSource --> Bitmap
2 public static System.Drawing.Bitmap ImageSourceToBitmap(ImageSource imageSource)
3 {
4     BitmapSource m = (BitmapSource)imageSource;
5
6     System.Drawing.Bitmap bmp = new System.Drawing.Bitmap(m.PixelWidth, m.PixelHeight, System.Drawing.Imaging.PixelFormat.Default);
7
8     System.Drawing.Imaging.BitmapData data = bmp.LockBits(
9     new System.Drawing.Rectangle(System.Drawing.Point.Empty, bmp.Size), System.Drawing.Imaging.ImageLockMode.ReadOnly, bmp.PixelFormat);
10
11     m.CopyPixels(Int32Rect.Empty, data.Scan0, data.Height * data.Stride, data.Stride);
12     bmp.UnlockBits(data);
13
14     return bmp;
15 }
```

好像可以了
 刘恒伟(935841375) 17:43:18
 这是什么原理
 16340117李江涛(530012814) 17:45:50
 可能是要求一定要有内容吧
 蔡梓珩(342078162) 20:07:21
 只有小论坛不滚动的的问题请问有人解决了吗
 刘恒伟(935841375) 20:31:58
 就算我死了，死在棺材里了，我也要把这个链接发出来：
https://docs.microsoft.com/en-us/uwp/api/windows.applicationmodel.datatransfer.datapackage.setdataprovider#Windows_ApplicationModel_DataTransfer_DataPackage_SetDataProvider_System_String_Windows_ApplicationModel_DataTransfer_DataProviderHandle
 王品<wj645349357@qq.com> 21:25:24
 还行
 2018-04-10
 梁肖诚(1042467380) 1:32:01

OnSuspending

```
// Get the frame navigation state serialized as a string and save in settings
Frame frame = Window.Current.Content as Frame;
ApplicationData.Current.LocalSettings.Values["NavigationState"] = frame.GetNavigationState();
```

OnNavigatedFrom

```
public override void OnNavigatedFrom(NavigationEventArgs e)
{
    state["FirstName"] = this.FirstName;
    state["LastName"] = this.LastName;
    state["Email"] = this.Email;
}
```

```
public override void OnNavigatedTo(string parameter, NavigationMode mode,
    IDictionary<string, object> state)
{
    try
    {
        this.FirstName = state["FirstName"].ToString();
        this.LastName = state["LastName"].ToString();
        this.Email = state["Email"].ToString();
    }
    finally { state.Clear(); }
```

App.xaml.cs

- OnSuspending函数
- 在程序挂起时保存应用程序的状态

```
private void OnSuspending(object sender, SuspendingEventArgs e)
{
    var deferral = e.SuspendingOperation.GetDeferral();
    //TODO: 保存应用程序状态并防止任何后台活动

    // Get the frame navigation state serialized as a string and save in settings
    Frame frame = Window.Current.Content as Frame;
    ApplicationData.Current.LocalSettings.Values["NavigationState"] = frame.GetNavigationState();
    deferral.Complete();
}
```

App.xaml.cs

- OnLaunched函数
- 在重新打开程序时加载应用程序的状态

```
if (e.PreviousExecutionState == ApplicationExecutionState.Terminated)
{
    //TODO: 从之前挂起的运行程序加载状态
    if (ApplicationData.Current.LocalSettings.Values.ContainsKey("NavigationState"))
    {
        rootFrame.SetNavigationState((string)ApplicationData.Current.LocalSettings.Values["NavigationState"]);
    }
}
```

二、实验步骤

第四周：挂起并关闭，保存状态

首先修改 App.xaml.cs 文件：

在 OnSuspending 函数处修改添加在程序挂起时保存应用程序的状态。

在 OnLaunched 函数处修改添加在程序打开程序时加载应用程序的状态

在 MainPage.xaml.cs 文件中重写 OnNavigatedTo 函数和 OnNavigatedFrom 函数。

OnNavigatedTo 函数指跳转到当前页面时执行的函数。函数处理的是将从挂起关闭保存时的数据，读出。将存在 LocalSetting 中的数据一个一个取出，并赋给相关变量，实现恢复状态功能。

OnNavigatedFrom 函数指从当前页面跳转走时执行的函数。函数将页面中的状态储存在 LocalSetting 中。

在 Newpage1.xaml.cs 文件中重写 OnNavigatedTo 函数和 OnNavigatedFrom 函数。实现与 Mainpage 中相应函数的功能。

主要步骤：

将数据进行处理，使得需要保存的数据能够实现保存在 LocalSetting 中。但是这只能储存基本类型，不能储存自定义的类型。由于需要储存 Todoitem 这个类，其中有 imagesource 属性的值。所以实现起来踩了好多大坑。通过和舍友，同学的讨论，我使用了在选取图片时，将其他文件夹中的图片暂时复制到应用的 LocalFolder 文件中。然后将复制后的文件流 储存在一个 LocalSetting 中。在实现读取数据时，使用文件选择器，将暂时保存的图片选中，然后赋值给相应的 Todoitem.image。这样才最终实现了对应的图片的绑定，保存。

实现对 Todoitem 的储存，由于不能储存自己定义的类型，所以将 viewmodel 转成单例模式。这样就可以实现对 viewmodel 中的个实例的访问。由于需要保存 viewlist 的状态，将 checked 的值与 checkbox 的 checked 和 line 的 visibility 绑定在一起，需要将 bool 类型转成 visibility 的 Visible 和 Collapsed。

第五周：动态磁贴与分享

要求使用标准的处理 XML DOM 方式创建动态磁贴

要求采用 Adaptive Tile （覆盖至少 small、medium、wide）

实现效果：要求每添加一条项目，磁贴能进行更新，并且更新的内容循环展示（1-2-3-4-5-1-2-3-4.....）

了解到 UWP 创建磁贴可以使用多种方式，一种单独创建.xml 文件，在文件中使用 XML 来动态生成磁贴，在 C#文件中实现数据的绑定，以及更新。

主要步骤：

在项目中添加 Tile.xml 文件，文件中使用 xml 语法创建 TileSmall,TileMedium,TileWide,TileLarge，四个标签。并在各自 **template** 中定义文本 Title,Description,以及图片 Image。在 Mainpage.cs 中实现更新磁贴的函数 UpdateTile。在函数中使用 TileUpdateManager 来创建磁贴更新队列。使用 xml.LoadXml 来读取 xml 文件内容，并以 xml.GetElementsByTagName("**text**")方式，获取各个需要更改的目标，将 Todoitem 对应的 Title, Description, Image 输入到相应的内容中。

分享功能：

使用 DataTransferManager.ShowShareUI();并在页面跳转时将要分享的数据传出。

第六周：数据持久化，SQLite的使用，模糊查询

利用数据库保存及恢复应用状态以及实现 todo 表项的增、删、改、查。

根据所给的参考，将数据库的操作抽象为四部：

使用 SQLiteConnection 来查找的相应的数据库。

将 SQL 语言保存在 String 中

使用 SQLiteConnection.Prepare(string)将 SQL 语句传入 SQLite 中

使用 SQLiteConnection.Step()。来执行数据库的相关操作

```
using (var statement = conn.Prepare(sql))  
  
{  
  
    statement.Step();  
  
}
```

LocalFolder RoamingFolder 等存储位置的作用：

LocalFolder：应用文件中，保存应用的数据

LocalSetting：应用文件中，小型的数据保存，保存应用的设置。

RoamingFolder：操作系统中，保存应用程序数据,可以使用云同步的数据

RoamSetting：操作系统中，可以同步到云上

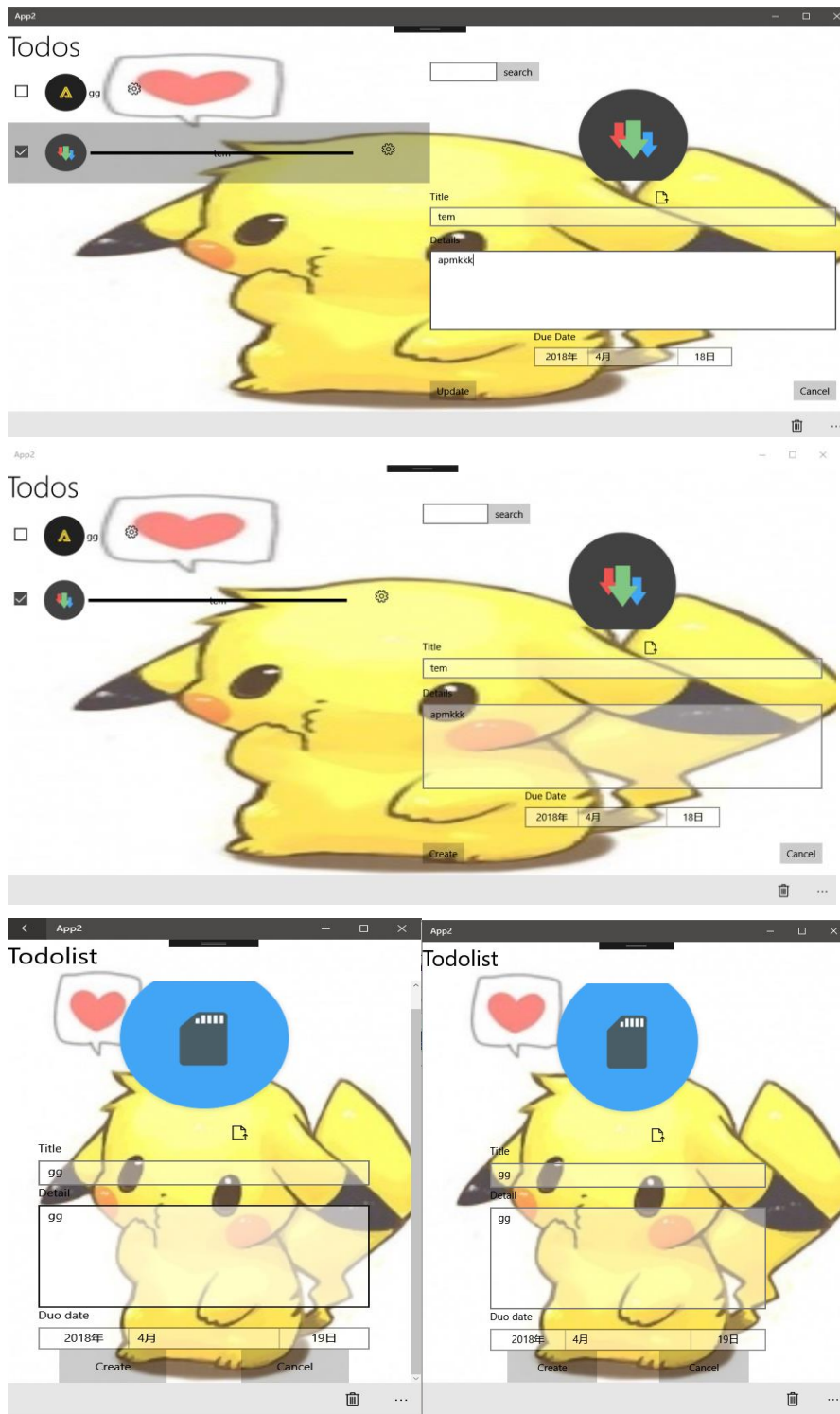
Credential Locker：操作系统中，保存密码文件，被加密了。

Temp:操作系统中，保存应用产生的临时文件

StringBuilder 的作用：用于频繁大量的字符串拼接操作，提高 string 类型的拼接效率。

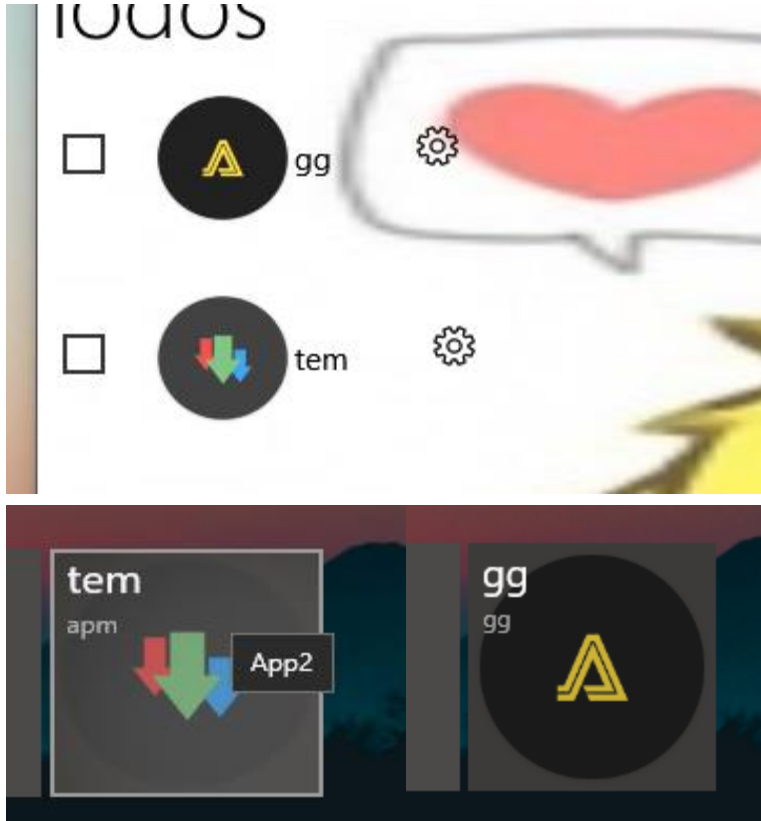
三、关键步骤截图

第四周：挂起并关闭

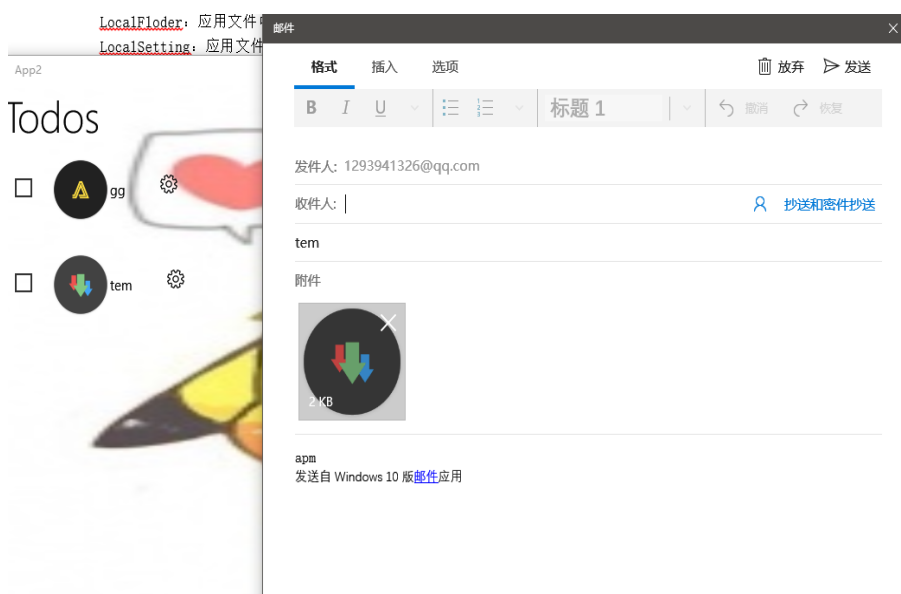


第五周:

磁贴

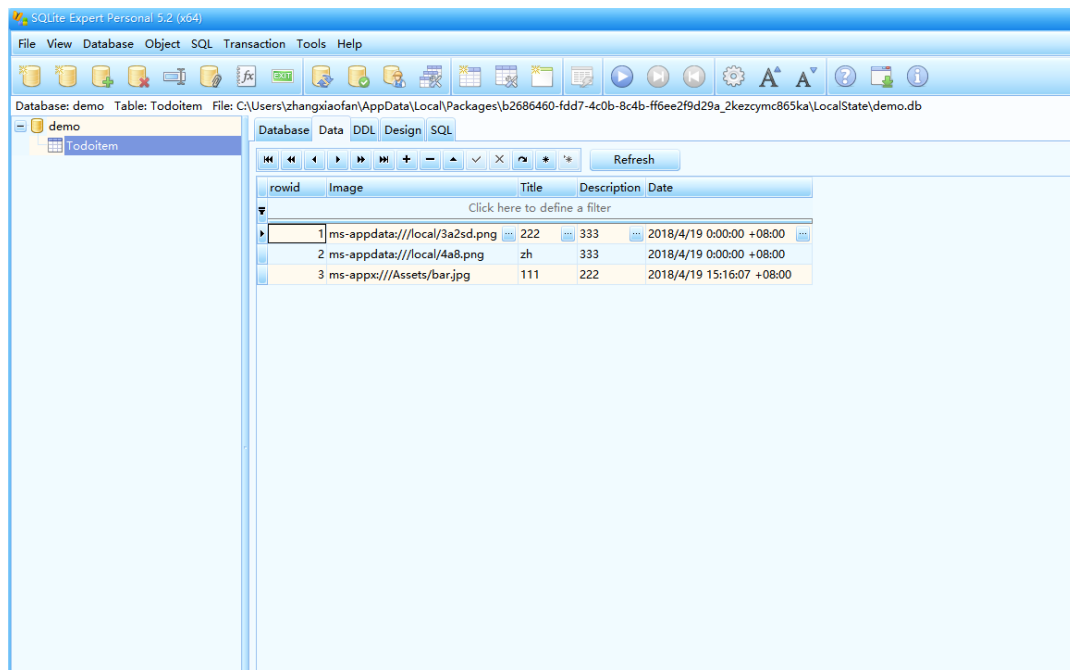


分享功能:



第六周：使用数据库将数据持久化，模糊搜索

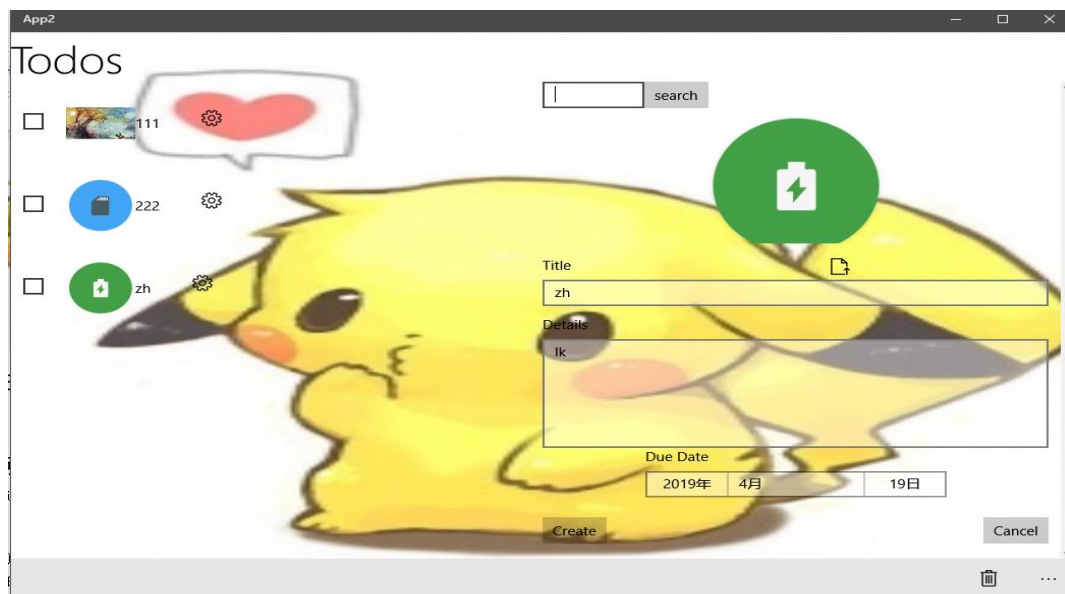
数据库的可视化：

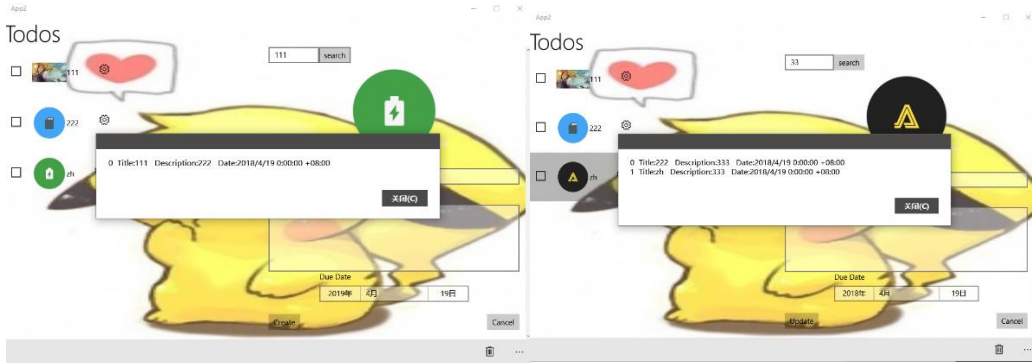


Database: demo Table: Todoitem File: C:\Users\zhangxiaofan\AppData\Local\Packages\b2686460-fdd7-4c0b-8c4b-ff6ee2f9d29a_2kezcymc865ka\LocalState\demo.db

rowid	Image	Title	Description	Date
1	ms-appdata://local/3a2sd.png	222	333	2018/4/19 0:00:00 +08:00
2	ms-appdata://local/4a8.png	zh	333	2018/4/19 0:00:00 +08:00
3	ms-appx:///Assets/bar.jpg	111	222	2018/4/19 15:16:07 +08:00

应用的截图





四、亮点与改进（可选）

第四周：实现了图片的选取，绑定功能。可以动态保存显示挂起关闭时的 Todoitem 中的图片，以及保存 Select 图片的状态。

第五周：实现了动态磁贴的图片的绑定，可以在磁贴上显示当前 Todoitem 中的图片。实现了分享图片的功能，在分享邮件时可以将当前 Todoitem 中的图片作为附件发送。

第六周：实现使用数据库将数据进行持久化的处理。对 Todoitem 中的图片进行数据库的保存。

遇到的问题

最头疼的问题就是如何处理图片。一开始是将图片以 ImageSource 形式来处理，但是在挂起并关闭时，就遇到了无法将其储存在 LocalSetting 中的问题，在经过查询官方文档后，将类型转换为 BitmapImage 类型，这种类型的好处是能够读取文件路径，从而能方便的处理比如复制修改等操作，还有的问题就是本地文件夹访问权限的问题。最终将需要的文件复制到应用文件夹中，暂时解决了问题。

五、思考与总结

经过前三周的学习,已经掌握了部分 UWP 应用开发的基础。在这三周中通过对挂起并关闭的处理,磁贴和分享,以及数据库的使用。加深了对 UWP 应用开发的理解。对 C#的委托,事件也有了新的认识。每一节课的内容都很多,需要下课后花费更多的时间去理解,作业的难度也还好,每次作业 TA 也都会讲解作业要实现的难点。老师在课上也会给出范例。整个过程,没有了初学时的无措,学会了在官方文档中查找需要使用的 API,以及能够和同学一起分享解决问题的过程。我认为现在的 Todolist 已经可以作为一个不联网的 APP 使用了,但是还有一些问题,由于前期的开发没有比较清晰的思路,在部分函数的调用变量的使用方面没有完全遵守相应的规则,导致需要改进时优化时面临一些困难。