

# 现代操作系统应用开发实验报告

学号： 15331418

班级： 晚上班

姓名： 郑柏川 实验名称： Live Tiles & App to App Communication

## 一. 参考资料

参考的 ppt

## 二. 实验步骤

磁贴的设计：

```
<?xml version="1.0" encoding="utf-8" ?>
<tile>
  <visual>
    <binding template="TileSmall" branding="none" hint-textStacking="center">
      <image src="Assets\background.jpg" placement="background"/>
      <text hint-style="subtitle"></text>
    </binding>

    <binding template="TileMedium" branding="Name" displayName="Todos">
      <image src="Assets\background.jpg" placement="background"/>
      <text hint-style="subtitle"></text>
      <text></text>
    </binding>

    <binding template="TileWide" branding="Name" displayName="Todos">
      <image src="Assets\background.jpg" placement="background"/>
      <text hint-style="subtitle"></text>
      <text></text>
      <text></text>
    </binding>
  </visual>
</tile>
```

其中在 text 标签加内容：

```

foreach (TodoItem item in ViewModel.AllItems)
{
    int count = 0;

    // small
    ((XmlElement)texts[count]).InnerText = item.title;
    count++;

    // medium
    ((XmlElement)texts[count]).InnerText = item.title;
    count++;
    ((XmlElement)texts[count]).InnerText = item.description;
    count++;

    // wide
    ((XmlElement)texts[count]).InnerText = item.title;
    count++;
    ((XmlElement)texts[count]).InnerText = item.description;
    count++;
    ((XmlElement)texts[count]).InnerText = item.date.ToString();
    count++;

    TileNotification notification = new TileNotification(xmlDoc);
    updatemanager.Update(notification);
}

```

分析：对列表中的每一个 todoitem 都创建对应的 tilenotification 并更新

### 分享的实现：

先找到对应的 item：

```

private void ShareButtonClick(object sender, RoutedEventArgs e)
{
    //找到要分享的item
    DataTransferManager.ShowShareUI();
    ViewModel.SelectedItem = ((MenuFlyoutItem)e.OriginalSource).DataContext as TodoItem;
}

```

这样就定位到 selecteditem 了，然后创建 `DataPackage` 类的 data 对象，并将一个属性放入

```

//所要传输的数据
DataPackage data = args.Request.Data;
if (ViewModel.SelectedItem != null)
{
    //为数据添加值
    string textTitle = ViewModel.SelectedItem.title;
    string textDescription = ViewModel.SelectedItem.description;

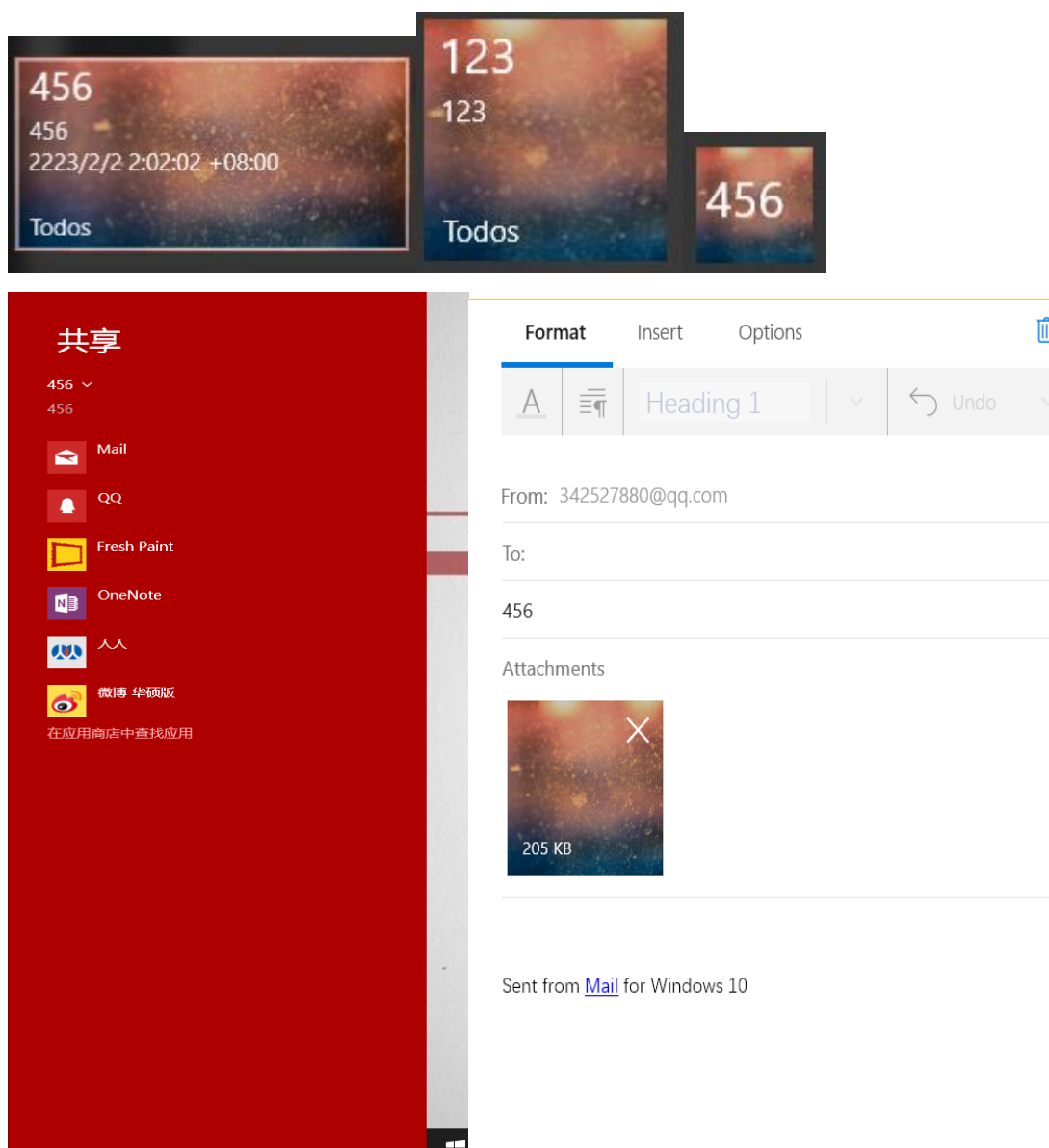
    data.Properties.Title = textTitle;
    data.Properties.Description = textDescription;
}
else
{
    //在侧边拿数据
    data.Properties.Title = title_grid.Text;
    data.Properties.Description = description_grid.Text;
}

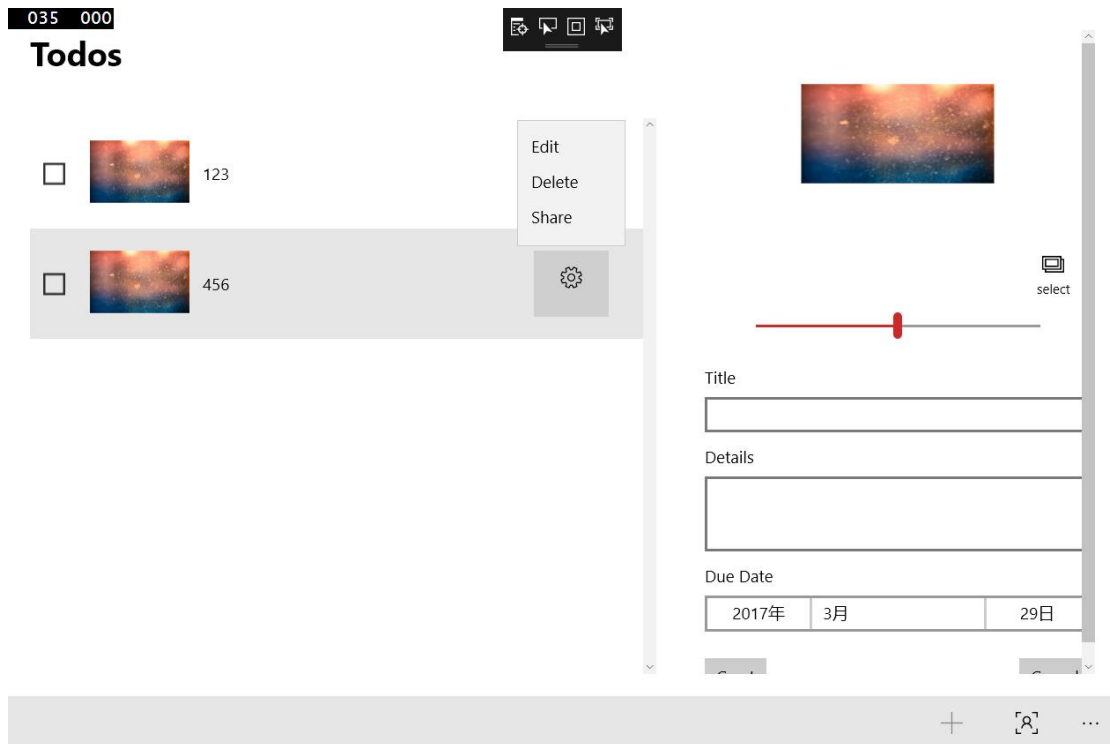
```

再实现图片的分享：

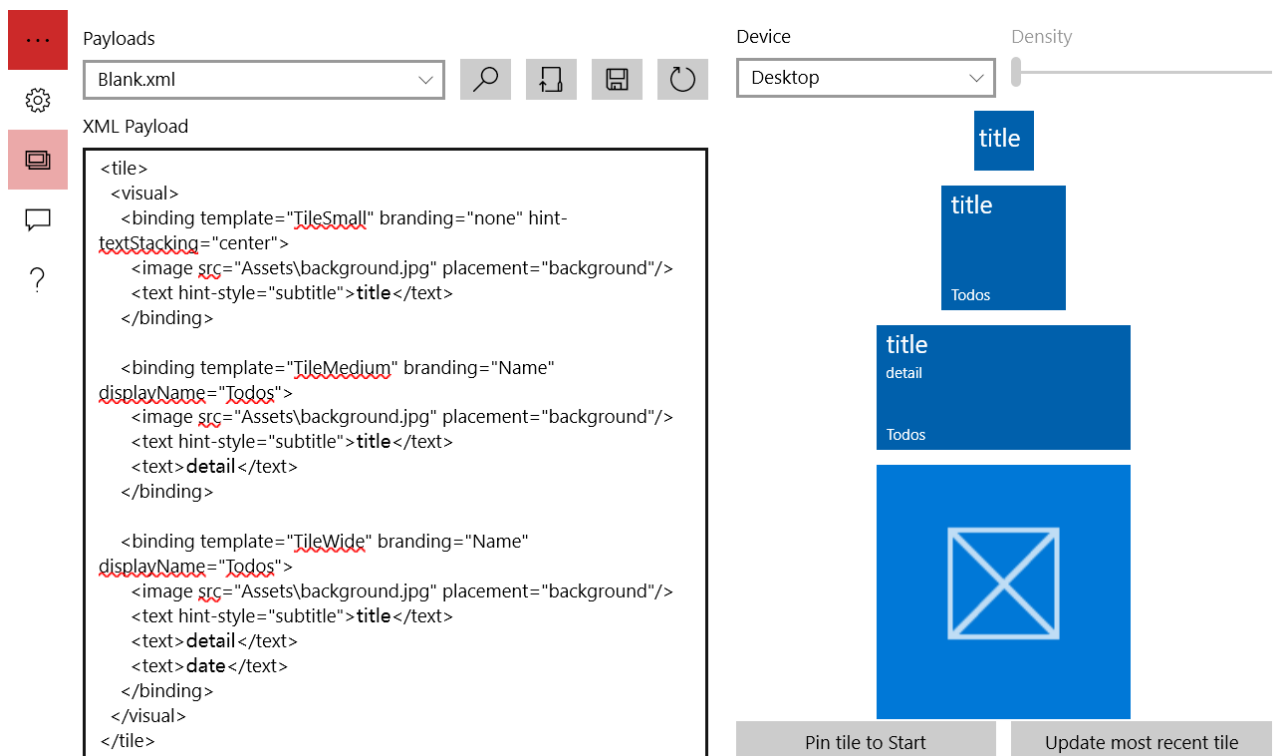
```
// 图片
DataRequestDeferral GetFiles = args.Request.GetDeferral();
try
{
    StorageFile imageFile = await Package.Current.InstalledLocation.GetFileAsync("Assets\\background.jpg");
    data.Properties.Thumbnail = RandomAccessStreamReference.CreateFromFile(imageFile);
    data.SetBitmap(RandomAccessStreamReference.CreateFromFile(imageFile));
}
finally
{
    GetFiles.Complete();
}
```

### 三．实验结果截图





#### Notification visualizer 的开发过程:



#### 四 . 实验过程遇到的问题

图片问题还是用上一次保存图片用的 api 一致

Tile 的制作一开始照着 ppt 打就行，分享部分也是一样

## 五 . 思考与总结

还是要多看群，例如分享的 item 定位就是看到群里一位大神贴的代码，之前是用 id 定位的