

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 3

一、参考资料

<https://www.juhe.cn/docs/api/id/39>

<https://code.juhe.cn/docs/1305>

```
//1.根据城市名/id查询天气
string url1 = "http://v.juhe.cn/weather/index";

var parameters1 = new Dictionary<string, string>();

parameters1.Add("cityname", ""); //城市名或城市ID, 如: &quot;苏州&quot;, 需要utf8 urlencode
parameters1.Add("dtype", ""); //返回数据格式: json或xml, 默认json
parameters1.Add("format", ""); //未来6天预报(future)两种返回格式, 1或2, 默认1
parameters1.Add("key", appkey); //你申请的key

string result1 = sendPost(url1, parameters1, "get");

JsonObject newObj1 = new JsonObject(result1);
String errorCode1 = newObj1["error_code"].Value;

if (errorCode1 == "0")
{
    Debug.WriteLine("成功");
    Debug.WriteLine(newObj1);
}
else
{
    //Debug.WriteLine("失败");
    Debug.WriteLine(newObj1["error_code"].Value+"-"+newObj1["reason"].Value);
}

//2.天气种类及标识列表
string url2 = "http://v.juhe.cn/weather/uni";

var parameters2 = new Dictionary<string, string>();

parameters2.Add("key", appkey); //你申请的key
parameters2.Add("dtype", ""); //返回数据的格式, xml或json, 默认json

string result2 = sendPost(url2, parameters2, "get");

JsonObject newObj2 = new JsonObject(result2);
String errorCode2 = newObj2["error_code"].Value;

if (errorCode2 == "0")
{
    Debug.WriteLine("成功");
    Debug.WriteLine(newObj2);
}
else
{
    //Debug.WriteLine("失败");
    Debug.WriteLine(newObj2["error_code"].Value+"-"+newObj2["reason"].Value);
}
```

<https://docs.microsoft.com/en-us/uwp/api/Windows.UI.Xaml.Controls.MediaPlayerElement>
<https://docs.microsoft.com/en-us/windows/uwp/design/controls-and-patterns/slider>
<https://docs.microsoft.com/zh-cn/windows/uwp/design/motion/storyboarded-animations>

实现播放控件

- 使用 `OpenOperationCompleted` 处理程序有机会发现媒体源内容的持续时间。确定持续时间后，Slider 控件的最大值设置为媒体项的总秒数。此值在 `RunAsync` 调用中设置，以确保它在 UI 线程上运行。
- `private async void MediaSource_OpenOperationCompleted(MediaSource sender, MediaSourceOpenOperationCompletedEventArgs args)`
- {
- `_duration = sender.Duration.GetValueOrDefault();`
- `await Dispatcher.RunAsync(Windows.UI.Core.CoreDispatcherPriority.Normal, () =>`
- {
- `_positionSlider.Minimum = 0;`
- `_positionSlider.Maximum = _duration.TotalSeconds;`
- `_positionSlider.StepFrequency = 1;`
- });
- }

二、实验步骤

第七周：使用网络，使用已有的 API，获得 JSON 或 XML，并将其解析。

首先百度到一些可用的 API 接口，并阅读 API 接口说明。

使用 `HttpClient httpClient = new HttpClient();`

`HttpResponseMessage httpResponseMessage = new HttpResponseMessage();`

创建 HTTP 服务，将对应 API 接口的 uri 传入 `httpClient` 中得到返回值 `httpResponseMessage`。

`httpResponseMessage = await httpClient.GetAsync(uri);`

使用 `await httpResponseMessage.Content.ReadAsStringAsync()`

将返回值转化为 string 类型。

根据得到的 string 类型和返回的格式 (json/xml) 选用不同的解析方法。

XML 的解析方法较为容易，将所得到了 string 载入到一个新的 XML document 中，利用 `xmlDocument.GetElementsByTagName("weather")` 方法得到所有标签名为

“weather”的元素,在根据所需 weather 出现的位置,获得其 element 中 innerText 的内容。

Json 的解析方法较为复杂,一开始是准备使用第三方库来解析的但是需要建立很多类,很麻烦,所以就简单使用了微软提供的解析方法。使用 JsonReader 来读取 Json 中的内容,将 string 类转换成 JsonReader 可以读的类型,JsonReader jsonReader = new JsonTextReader(new StringReader(httpResponseBody));
在根据要找到的内容的 path 来获取相应内容: jsonReader.Path == "result.today.weather"。

第八周:简单的媒体播放器

使用 MediaPlayerElement 控件制作简单的媒体播放器,根据老师的 PPT 使用 MediaPlayer 将 MediaSource 类用作所有媒体项目的容器。在 MediaPlayer 设置好后可以使用 MediaPlayerElement.SetMediaPlayer(mediaPlayer);将媒体的来源绑定到 MediaPlayerElement 元素上实现在 xaml 上的媒体播放。

播放,暂停,重放功能:

使用 MediaTimelineController 类,获得每一个 MediaPlayer 的 MediaTimelineController,然后使用时间控制线来实现播放,暂停,重放功能:
mediaTimeline.Play();mediaTimeline.Pause();mediaTimeline.Resume();

实现进度条功能:对每一个 MediaSource 设置的 OpenOperationCompleted 注册处理程序。这个用来获取媒体的最大播放时间的,用来定义 Slider 的 maximum 的值:
mediaSource.OpenOperationCompleted += MediaSourceOpenOperationCompleted;在 MediaSourceOpenOperationCompleted 函数中实现

为了使 Slider 的指针跟随播放时间的进行而移动,需要为时间线控制器的 PositionChanged 事件注册处理程序: mediaTimeline.PositionChanged += MediaTimelinePositionChanged;在 MediaTimelinePositionChanged 函数实现。

实现移动 Slider 的指针,媒体播放跳转到相应位置:Slider 中的特有方法 ValueChanged。将 Slider 的值赋给时间控制线的 Position。达到媒体播放跳转到相应位置。

全屏显示:在一开始定义 xaml 时就将 MediaPlayerElement 设置成填满应用界面,在使用全屏之后将应用放大到全屏,将多余的元素设置为不可见即可。

三、关键步骤截图

Select the city

广州

Weather_XML

Weather_JSON

雷阵雨转中雨

雷阵雨转中雨

Select ip

65.233.22.2

Country_JSON

美国

Select the city

广州

Weather_XML

Weather_JSON

雷阵雨转中雨

雷阵雨转中雨

Select ip

65.233.22.2

Country_JSON

美国

MediaPlayer



0s



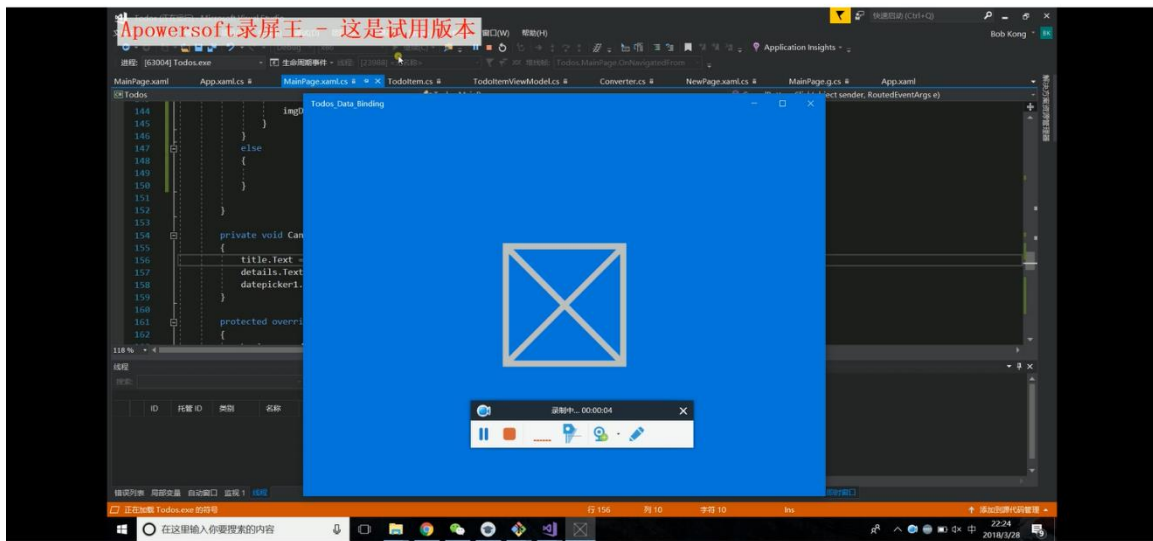
阮豆 - 遥不可及的你 (Cover: 7



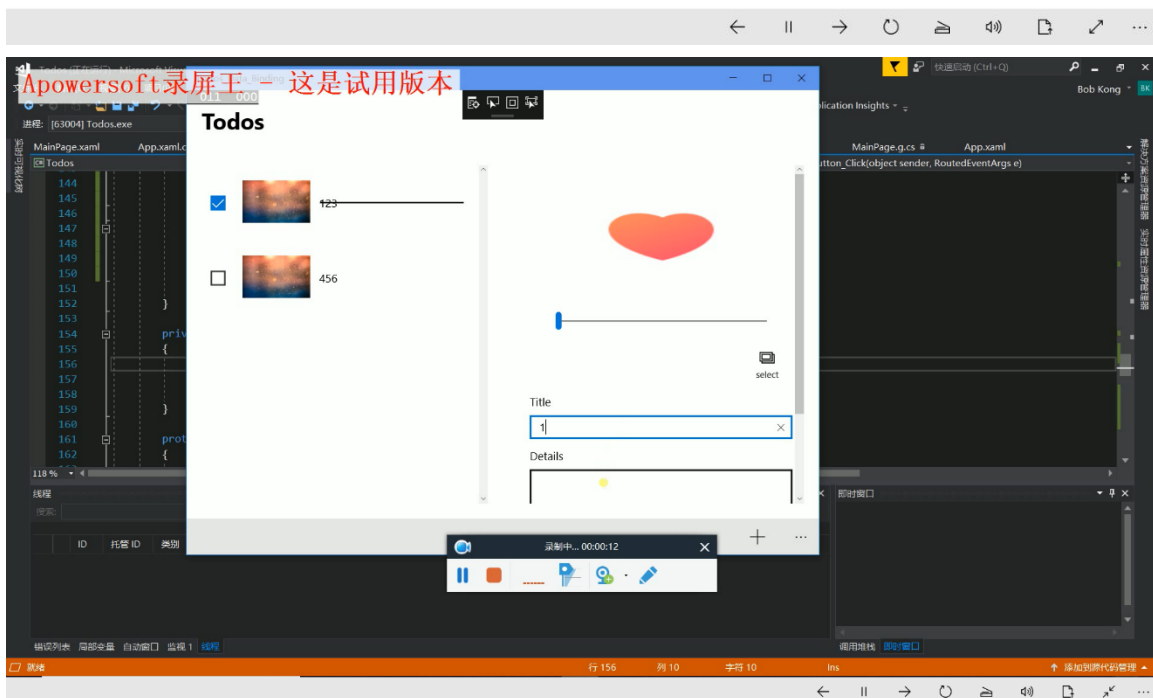
7s

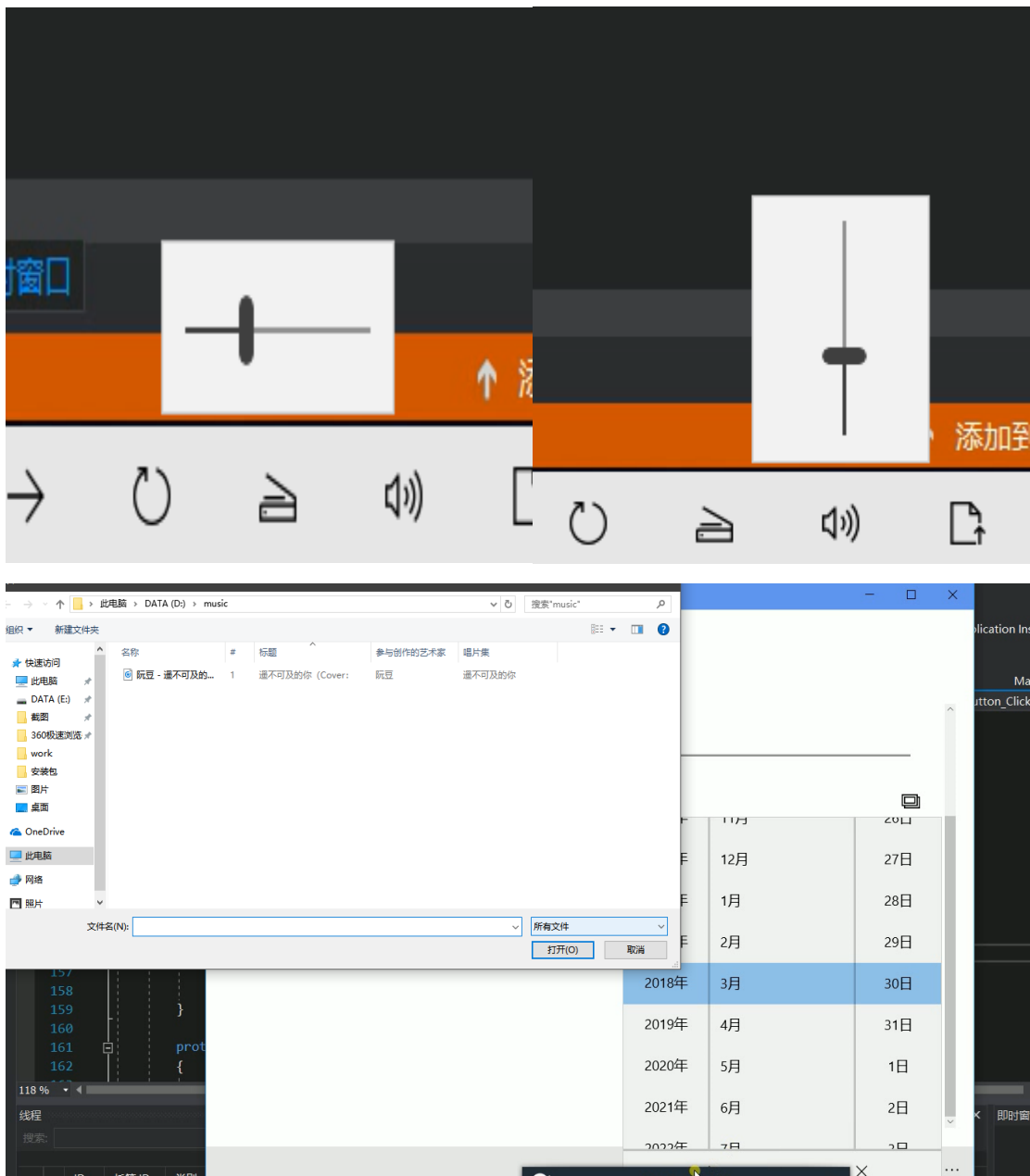


week4



4s





四、亮点与改进（可选）

第六周：使用 XML 和 JSON 方法对同一个 API 进行解析。

第七周：

- 1.可手动选择本地的媒体资源进行播放
- 2.实现封面旋转：播放音乐时，封面开始旋转，播放结束，封面复位，暂停时，封面的旋转动作暂停，播放视频的时候封面隐藏
- 3.实现调节播放声音的功能
- 4.实现对媒体的 1.5 倍加速和 0.5 倍慢速的功能。

五、遇到的问题

第六周:

在第六周使用 JSON 类型解析的时候，一开始希望将所有信息都保存在一个类中，就是使用第三方库。但是由于 API 的返回值中的数据太多，（没有想到将不用的数据移除掉）就没有使用第三方库。根据 JsonReader 类型的属性，可以根据 Path 来获取相应的数据。只是这种方法开销有点大。

第七周:

因为不能使用自带的功能，需要自己重写。一开始是使用 MediaElement 类，但是由于需要设计进度条，MediaElement 类实现起来比较麻烦，最后使用 MediaPlayerElement 控件，和 MediaPlayer 类以及 MediaTimeLineController 类实现。选取本地媒体文件时使用 OpenFilePicker，由于播放视频和音频 UI 是不相同的，所以需要判断文件类型，一开始以为是“.mp3”后缀，但是测试之后发现没有相应的跳转。后面查找文档后发现类型应该是“audio/mpeg”。实现封面的旋转动画，根据 TA 给的文档，简单了解了一些 StoryBoard。但还是没有头绪，在阅读了一些博客后，根据他们所提供的一些例子才完成封面的旋转

六、思考与总结

这两周的学习内容已经较为简单了，主要是如何使用网络，进行网络连接，使用网络 API 接口，解析 Json，XML 文档，使用官方的组件自定义媒体播放器。但还是有一些小坑，由于对 Json 文件的不熟悉，在如何解析它时花费了不少时间，由于当时没有想到将不需要的部分给删除，所以没有使用推荐的 Json 解析方法，而是使用了较为简单的方法。UWP 的学习马上就要结束了，个人觉得 UWP 应用还是挺好设计和使用的，官方的控件，以及各种功能都挺完善，第三方库的支持也还不错，那为什么不火呢。