Web Role 服务

# 概述:

客户机应该能够给云上载照片和故事的配置文件.我们在 blob 存储中存储所有这些文件.由于视频编码可能需要一些时间才能完成，我们使用标准Web Role前沿Worker Role背景模型.

# 期望的结果:

* 创建客户端上传文件服务.

# 不期望的结果:

* 编码视频.

# 设计:

首先我们必须决定哪个组件负责上传文件.我们有几个选项:

1. 在电话上使用其他 API 直接将文件上载到 blob 存储.这就需要在客户端应用程序中嵌入存储帐户密码.
2. 将文件上载到 Web Role服务，然后将它们存储在 blob.这比 # 1需要更多时间，但并不要求客户端知道存储帐户密钥.
3. 完善 # 2、 使用流传输模式.这可能需要一些努力来实现.
4. 使用 Web 角色服务生成的 SAS，并允许手机客户端上载的文件直接blob存储.此选项需要一些对 SAS 的精细控制.

至少我们不应在 Windows Phone 客户端应用程序中嵌入我们存储帐户键.所以 # 1 是有问题的.其余的所有选项都有优点和缺点.对于小的文件(如配置文件)， # 2 不引入太多延迟，特别是如果 blob 存储服务和 Web Role在相同的数据中心上.对于大型文件(如照片) # 3 和 # 4 似乎是不错的选择.

在此版本中，我们使用 # 2 上传 xml 配置文件.我们不需要在客户端上传一个物理文件.相反，客户端只是将 xml 文件内容放在请求主体.我们使用 # 4 上传的照片.为了提供更好的安全性，我们选择了短暂的 SAS.目前的决定是 30 分钟，这应该足以上载一个典型的故事里面的所有照片..

Web 服务本身使用 WCF Web API，它是创建其他服务的最佳解决方案.

下面是上传过程的概述:

* 客户端发送 xml 配置文件服务.
* 服务分析配置文件.它将为每张照片生成带SAS的blob URI.
* 服务修改 每张照片的blob URI（不带SAS），并使用表示的短影唯一的 ID将其存储在blob存储，.
* 服务发回响应，其中包含 blob URI 的每张照片 （与 SAS) 和唯一的 id.
* 客户端将照片上载到 blob 存储.
* 所有的照片都上载后，客户端通知服务这是短影使用的唯一 ID.确认短影.
* 服务队列存储中创建了一条消息.消息中包含短影的唯一 ID.
* Worker Role队列和处理编码作业.

我们设计的服务协议如下所示:

上传配置文件:

Request:

POST http://.../stoies

Request 内文就是xml 配置文件. 例如:

<Story PhotoCount="2">

<Photo Name="Blue hills.jpg" PhotoDuration="3">

<Transition Name="Fade Transition" Duration="20" />

</Photo>

<Photo Name="Test.jpg" PhotoDuration="7">

<Transition Name="Fade Transition" Duration="30" />

</Photo>

</Story>

Response:

如果操作成功, 返回201. Response内文是个简单的xml文件:

<Story ID=”some GUID”>

<Photo Name="Blue hills.jpg" Uri="A blob URI with SAS"/>

<Photo Name="Test.jpg" Uri="A blob URI with SAS "/>

</Story>

如果操作失败，返回正确的错误代码，并Response内文中包含错误消息.

确认短影:

Request:

PUT http://.../stories/GUID?commit=true

Request 内文: 空.

Response: 如果操作成功，则返回204和空内文.如果失败，返回正确的错误代码，并在Response内文中包含错误消息.

列表显示已编码视频:

Request:

GET <http://.../stories/>

Response: 如果操作成功，则返回200以及描述视频的JSON Response内文. 如果失败，返回正确的错误代码，并在Response内文中包含错误消息.目前我们只是支持 JSON. 在将来我们可能需要更改设计，使其依接受Request头返回 XML 或 JSON.