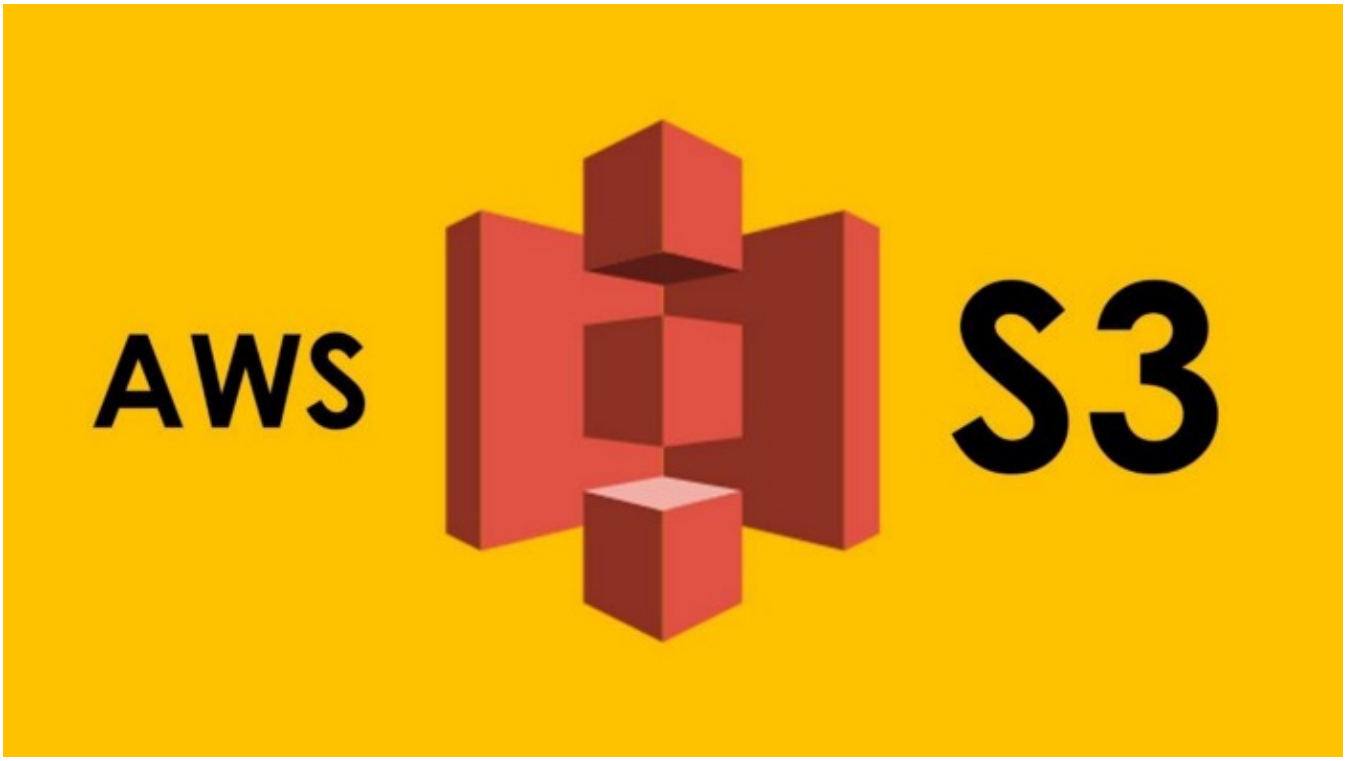
Uploading to AWS fi

Ever found yourself needing to uploa
finding up-to-date documentation? W
explain how to upload to S3 using simple C# in Unity, WebRequests and streams. The code was tested in Unity 2019.x.

Tomasz Witczak
Jun 12, 2020 · 4 min read



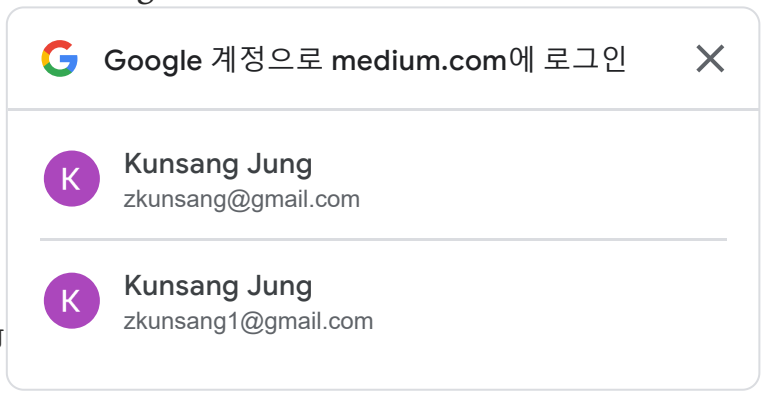
AMAZON

Implementation

Before you write any actual code, you will first need to create a bucket via the AWS Web GUI — this will define the root location where all files will be uploaded. Whilst you're there, you'll also need to collect the following information:

- **Bucket name** — the name of the created bucket.
- **Access key — r**equired for establishing a connection with AWS3.

- **Secret key — a**lso required for establishing a connection with AWS3.



Here are our using directives:

```
using System;
using System.IO;
using System.Net;
using System.Security.Cryptog
using System.Text;
using UnityEngine;
```

With them we can create our very first members at the top of our class:

```
private const string awsBucketName = "MyTestBucket";
private const string awsAccessKey = "XXXxxxXXXxxx";
private const string awsSecretKey = "XXXXxxxxxxXXXxxxXXX";
```

We will also need a member to store our base virtual URL :

```
private string awsURLBaseVirtual = "";
```

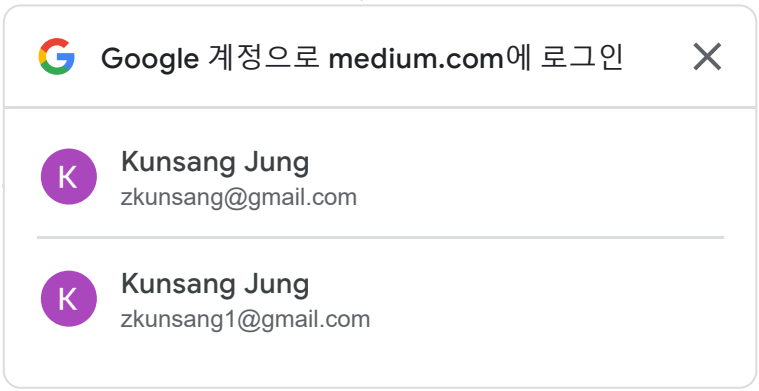
We then fill in the base virtual URL in MonoBehavior's Start:

```
void Start()
{
   awsURLBaseVirtual = "https://" +
      awsBucketName + ".s3.amazonaws.com/";
}
```

OK, so let's assume that you want to upload a single file to your test bucket. Let's create a function named `UploadFileToAWS3` , which takes the filepath and filename as it's arguments.

```
public void UploadFileToAWS3(string FileName, string FilePath)
```

We'll start the function by creating an AWS3 header. To do so, first we need to create a current-date string:




```
string currentAWS3Date = Syst
   "ddd, dd MMM yyyy HH:mm:ss
   "GMT";

string canonicalString =
   "PUT\n\n\n\nx-amz-date:" +
   currentAWS3Date + "\n/" +
   awsBucketName + "/" + FileName;
```

Now we need to encode our secret:

```
   UTF8Encoding encode = new UTF8Encoding();
   HMACSHA1 signature = new HMACSHA1();
   signature.Key = encode.GetBytes(awsSecretKey);
   byte[] bytes = encode.GetBytes(canonicalString);
   byte[] moreBytes = signature.ComputeHash(bytes);
   string encodedCanonical = Convert.ToBase64String(moreBytes);
```

With the encoded secret, we can create an AWS3 header:

```
string aws3Header = "AWS " + awsAccessKey + ":" + encodedCanonical;
```

Now, it's time for a WebRequest. To do that, we have to create a URL that defines the destination path. We do this by combining the base virtual URL with the specified filename.

```
string URL3 = awsURLBaseVirtual + FileName;
```

The WebRequest itself needs our newly created URL (to know where to point at), our header and date.

```
WebRequest requestS3 = (HttpWebRequest)WebRequest.Create(URL3);
requestS3.Headers.Add("Authorization", aws3Header);
requestS3.Headers.Add("x-amz-date", currentAWS3Date);
```

Next, let's read byte data from our loc


```
byte[] fileRawBytes = File.Re
requestS3.ContentLength = fil
```

Now, it's very important to set a prope
are uploading, we use PUT.

```
requestS3.Method = "PUT";
```

Finally, we can upload the file to AWS via a Stream.

```
Stream S3Stream = requestS3.GetRequestStream();

S3Stream.Write(fileRawBytes, 0, fileRawBytes.Length);

Debug.Log(
   "Sent bytes: " +
   requestS3.ContentLength +
   ", for file: " +
   FileName);
```

Don't forget to close your Stream after!

```
S3Stream.Close();
```

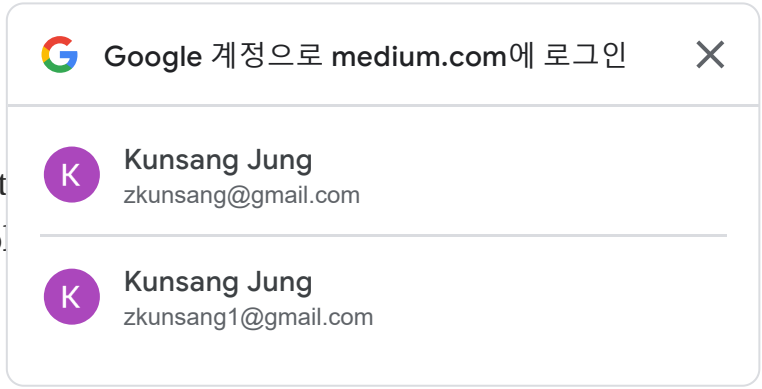
Putting it all together

Now we have all the pieces, we can put them together in a complete class!

```
using System;
using System.IO;
using System.Net;
using System.Security.Cryptography;
using System.Text;
using UnityEngine;
```

```
public class AWS3 : MonoBehaviour
{
    private const string awsB
    private const string awsA
    private const string awsS

    private string awsURLBase

    void Start()
    {
        awsURLBaseVirtual = "
           awsBucketName +
           ".s3.amazonaws.com/";
    }

    public void UploadFileToAWS3(string FileName, string FilePath)
    {
        string currentAWS3Date =
            System.DateTime.UtcNow.ToString(
                "ddd, dd MMM yyyy HH:mm:ss ") +
                "GMT";
        string canonicalString =
            "PUT\n\n\n\nx-amz-date:" +
            currentAWS3Date + "\n/" +
            awsBucketName + "/" + FileName;

        UTF8Encoding encode = new UTF8Encoding();
        HMACSHA1 signature = new HMACSHA1();
        signature.Key = encode.GetBytes(awsSecretKey);
        byte[] bytes = encode.GetBytes(canonicalString);
        byte[] moreBytes = signature.ComputeHash(bytes);
        string encodedCanonical = Convert.ToBase64String(moreBytes);

        string aws3Header = "AWS " +
            awsAccessKey + ":" +
            encodedCanonical;

        string URL3 = awsURLBaseVirtual + FileName;

        WebRequest requestS3 =
           (HttpWebRequest)WebRequest.Create(URL3);
        requestS3.Headers.Add("Authorization", aws3Header);
        requestS3.Headers.Add("x-amz-date", currentAWS3Date);

        byte[] fileRawBytes = File.ReadAllBytes(FilePath);
        requestS3.ContentLength = fileRawBytes.Length;

        requestS3.Method = "PUT";

        Stream S3Stream = requestS3.GetRequestStream();
        S3Stream.Write(fileRawBytes, 0, fileRawBytes.Length);
        Debug.Log("Sent bytes: " +
            requestS3.ContentLength +
            ", for file: " +
            FileName);
```

```
            S3Stream.Close();
        }
    }
```



You can easily extend this functionalit
AWS3, and even encapsulate the whol

> *XRLO: eXtended Reality Lowdown is brought to you by REWIND, an immersive design and innovation company. If you want to talk tech, ideas, and the future, get in touch here.*
>
> *Your claps and follows help us understand what our readers like. If you liked our articles, show them some love!* ♡
>
> *We'd also love to hear from you. If you're passionate about all things XR, you can apply to contribute to XRLO here.* ✍

Unity Amazon S3 AWS Rnd