

Building an Amazon S3 Client with Application Express 4.0

*An Oracle White Paper
December 2010*

Note: The following is intended to outline our general product direction. It is intended for information purposes only, and may not be incorporated into any contract. It is not a commitment to deliver any material, code, or functionality, and should not be relied upon in making purchasing decisions. The development, release, and timing of any features or functionality described for Oracle's products remains at the sole discretion of Oracle.

Building an Amazon S3 Client with Application Express 4.0

Terminology.....	2
Overview.....	5
Prerequisites.....	5
Amazon S3 Technical Details.....	5
API's.....	5
Authentication.....	5
Common Request HTTP Headers.....	6
Create the Application and Supporting Objects.....	6
Create Packages and Functions.....	6
Create the Application.....	8
Create an Application Item for Date Header and Computations.....	8
Create a Page to View All Buckets.....	9
Create a RESTful Web Reference to View All Buckets.....	9
Create a Page to View all Buckets.....	10
Create a Page to Create a New Bucket.....	11
Create the RESTful Web Service Reference.....	12
Create the Page.....	12
Create a Page to List Bucket Contents.....	14
Create the RESTful Web Service Reference.....	14
Create the Page.....	15
Modify Report to Provide Download Link for Objects.....	16
Create a Page to Upload an Object to a Bucket.....	17
Create the RESTful Web Service Reference.....	18
Create the Page.....	18
Add Ability to Delete an Object.....	20
Create the RESTful Web Service Reference.....	20
Modify the Report on Page 3 to Include a Delete Link.....	21
Create Process to Call the Delete Object Web Service Reference.....	22
Conclusion.....	24

TERMINOLOGY

S3	Amazon S3 is storage for the Internet. It is designed to make web-scale computing easier for developers.
Object	A file and any metadata that describes the file.
Bucket	A container for objects. Buckets can have distinct access control lists as well as distinct geographical regions.
AWS Account	To access any Amazon Web Services you must first create an AWS Account at http://aws.amazon.com .
SOAP	SOAP is a simple XML-based protocol to let applications exchange information over HTTP. ¹
REST	Resource oriented architecture for interacting with web services where the resource is described by the URI and the method is described by the HTTP verb (GET, PUT, POST, DELETE).
Epoch	Number of seconds since January 1, 1970.

¹ http://www.w3schools.com/soap/soap_intro.asp

INTRODUCTION

Oracle Application Express (APEX), a feature of the Oracle Database 11g, combines rapid web application development with the power of the Oracle database. Application Builder features an easy-to-use browser-based interface which enables developers and non-programmers to develop and deploy data driven web applications in very little time.

Oracle Application Express doesn't depend on any client software for developing or deploying web applications. Simple architecture, and browser-based features make the transition for developers and end-users seamlessly provide the URL for the cloud environment.

The multi-tenant capabilities of Oracle Application Express allow multiple users and their associated applications to co-exist within one Oracle Database, minimizing cost. Only one instance is needed and users work in a dedicated work area called a workspace. An added advantage, when you create a database backup you also create backup of application source. Oracle Application Express can integrate with other applications by consuming web services.

Amazon Simple Storage Service (Amazon S3) provides highly available and highly scalable Internet storage. It was designed to sustain the concurrent loss of data in two facilities.² Common use cases include content storage and distribution as well as backup and disaster recovery. Amazon S3 allows for storing objects (files) in containers called *buckets* and has web service API's to allow for S3 interactions.

Oracle Application Express 4.0 introduces support for consuming RESTful web services. RESTful web services conform to a simpler architecture than traditional SOAP based web services. Oracle APEX 4.0's support for consuming RESTful web services makes it possible to build a client on Amazon S3, using its RESTful web service API.

² <http://aws.amazon.com/s3/>

OVERVIEW

The purpose of this white paper is to describe how to build an Oracle Application Express application that is a fully functional Amazon S3 client. The application will allow you to view all your buckets, create a new bucket, view contents of a bucket, add objects to a bucket, and finally the ability to delete objects.

PREREQUISITES

- Oracle Application Express 4.0 instance that can make network callouts (note that you cannot use apex.oracle.com which does not support making network callouts)
- Amazon Web Services account (<http://aws.amazon.com>)
- Amazon S3 Account (<http://aws.amazon.com/s3>)
- Download and install S3Fox Organizer (<http://www.s3fox.net>)
- Grant execute on DBMS_CRYPT to the schema associated with your workspace

AMAZON S3 TECHNICAL DETAILS

API's

Amazon S3 is a web service that provides both a SOAP and a REST API to interact with the service. This document describes interactions using the REST API.

Authentication

Authentication is the process of establishing an identity. For the purposes of Amazon S3, this is done through the use of the poorly named HTTP header called Authorization. When you sign up for an AWS account, you are provided with an AWS Access Key ID and a Secret Access Key. All requests to Amazon S3 need to include the HTTP header Authorization in the following form:

```
AWS <AWSAccessKeyId>:<Signature>
```

<AWSAccessKeyId> should be replaced with your actual AWS Access Key ID and <Signature> is the HMAC-SHA1 hash of a string composed of elements of the request using your AWS Secret Key as the key. The hash is then Base64 encoded. Included in this paper is a function to return this header based on the string you passed in. This function will be created in the next section.

The string that is hashed is a concatenation of the HTTP method, select HTTP headers sent with the request (such as Date), and the path to the resource.

Common Request HTTP Headers

The following table lists the HTTP headers that are usually sent with each request to the Amazon S3 service.

Header Name	Description	Required
Content-Length	Length of the message (without the headers) according to RFC 2616. Condition: Required for PUTs and operations that load XML, such as logging and ACLs.	Conditional
Content-Type	The content type of the resource. Example: text/plain	No
Date	The current date and time according to the requester. Example: Wed, 25 Nov 2009 12:00:00 GMT	Yes
Host	Normally, the value of Host is s3.amazonaws.com. A Host header with a value other than s3.amazonaws.com selects the bucket for the request as described in Virtual Hosting of Buckets . Condition: Required for HTTP 1.1 (most toolkits add this header automatically); optional for HTTP/1.0 requests.	Conditional
Authorization	The information required for request authentication. For more information, see The Authentication Header for details about the format.	Yes

³

CREATE THE APPLICATION AND SUPPORTING OBJECTS

Create Packages and Functions

First you compile a package that contains constants holding your AWS Access Key ID and your AWS Secret Key. To compile the package, do the following:

1. Login to your Oracle Application Express workspace
2. Click **SQL Workshop**
3. Click **SQL Commands**

³ <http://docs.amazonwebservices.com/AmazonS3/2006-03-01/dev/UsingRESTOperations.html>

4. Paste the code in code listing 1 into the SQL Commands text area
5. Replace <YourAWSAccessKeyID> and <YourAWSSecretKey> with the actual values from your account
6. Click **Run**

Code Listing 1

```
create or replace package amazon_secrets
as
    g_aws_id constant varchar2(20) :=
'<YourAWSAccessKeyID>';
    g_aws_key constant varchar2(40) :=
'<YourAWSSecretKey>';

end amazon_secrets;
/
```

Next, create a function that will return the Authorization header string with your AWS Access Key ID and the string from the request hashed with your AWS Secret Key. Use steps 1-6 above and the code from code listing 2. The schema associated with your workspace must be granted execute on DBMS_CCRYPTO.

Code Listing 2

```
create or replace function amazon_signature_sh1(
    p_string in varchar2) return varchar2
as
    output_string          VARCHAR2 (32000);
    encrypted_raw          RAW (2000);           -- stores
encrypted binary text
    decrypted_raw          RAW (2000);           -- stores
decrypted binary text
    key_bytes_raw          RAW (64);             -- stores
256-bit encryption key
begin

    key_bytes_raw := UTL_I18N.STRING_TO_RAW
(amazon_secrets.g_aws_key, 'AL32UTF8');
    decrypted_raw := UTL_I18N.STRING_TO_RAW (p_string,
'AL32UTF8');

    encrypted_raw := dbms_crypto.mac(src =>
decrypted_raw, typ => DBMS_CCRYPTO.HMAC_SH1, key =>
key_bytes_raw);
    output_string := UTL_I18N.RAW_TO_CHAR
(utl_encode.base64_encode(encrypted_raw), 'AL32UTF8');

    return 'AWS
'||amazon_secrets.g_aws_id||':'||output_string;
end amazon_signature_sh1;
/
```


Finally, create functions that will later be used in a query that return just the signature portion of the Authorization header and another function to return just your AWS Access Key ID. Use the steps above and the code in code listing 3.

Code Listing 3

```
create or replace function amazon_sig_only(
    p_string    in varchar2) return varchar2
as
begin
    return substr(amazon_signature_sh1(p_string),26);
end amazon_sig_only;
/

create or replace function amazon_aws_id_only
return varchar2
as
begin
    return amazon_secrets.g_aws_id;
end;
/
```

CREATE THE APPLICATION

Once you have your package and functions compiled in the schema associated with your workspace, you create the application that will become the Amazon S3 client. Follow the steps below to create the application.

1. Click the Application Builder tab
2. Click **Create**
3. Choose Database and click **Next >**
4. Choose From Scratch and click **Next >**
5. Enter **Amazon S3 Client** in the Name field and click **Next >**
6. In the Add Page section, choose blank page, enter **Buckets** in the Page Name field and Click **Add Page**
7. Click **Create**
8. Click **Create**

Create an Application Item for Date Header and Computations

The Date HTTP header must be passed with each request to Amazon S3 in the format “Dy, DD Mon YYYY HH24:MI:SS GMT.” Create an application item

with a corresponding computation so that the current date and time is computed when the page is loaded and when the page is submitted. To create the application item and computations:

1. Click **Shared Components** on the Application Builder home page
2. Click Application Items under the Logic heading
3. Click **Create**
4. Enter **S3_DATE_TIME** in the Name field and click **Create**
5. Click the **Shared Components** breadcrumb
6. Click **Application Computations**
7. Click **Create**
8. Choose **S3_DATE_TIME** from the Computation Item list
9. Choose **Before Header** from the Computation Point list
10. Choose **PL/SQL Function Body** from the Computation Type list
11. Enter the following in the Computation field. You may need to adjust the computation based on the time zone of your Oracle Application Express instance. You can determine the time zone of the instance by running the following query:

```
select dbtimezone from dual;
```

For example, if you are currently 7 hours behind GMT, you would change `sysdate + 4/24` to `sysdate + 7/24` in the computation below.

```
return to_char(sysdate+4/24,'Dy, DD Mon YYYY  
HH24:MI:SS') || ' GMT';
```

12. Click **Create Computation**
13. Repeat steps 7 – 12 above, choosing **After Submit** instead of Before Header in step 9

CREATE A PAGE TO VIEW ALL BUCKETS

Create a RESTful Web Reference to View All Buckets

To interact with a web service in Oracle Application Express, you first create a Web Service Reference to that service. To create a Web Service Reference to list all buckets owned by an AWS account:

1. Click **Shared Components** on the Application Builder home page
2. Click **Web Service References**

3. Click **Create**
4. Choose **REST** and click **Next**
5. Enter the following on the REST Details page
 - a. Enter **List All Buckets** in the Name field
 - b. Enter <http://s3.amazonaws.com/> in the URL field
 - c. Accept the defaults for HTTP Method and Basic Authentication and then enter **Host** in the Name field under REST HTTP Headers and click **Add Header**
 - d. Repeat step c and create **Authorization** and **Date** headers
 - e. Click **Next**
6. Click **Next** on the REST Inputs page since this service has no parameters
7. Enter the following on the REST Outputs page
 - a. Accept the default output format and enter **/ListAllMyBucketsResult/Buckets/Bucket** in the XPath to Output Parameters field
 - b. Enter <http://s3.amazonaws.com/doc/2006-03-01/> in the Response Namespace field
 - c. Enter **Name** in the Name field and tab to the Path field
 - d. Click **Create**

Create a Page to View all Buckets

Now create the page using the Web Service Reference.

1. On the Web Service Reference success page, click **Create Form and Report on Web Service**
2. Choose **List All Buckets** from the Web Service Reference select list
3. Click **Next**
4. Change the page to page **1** to re-use that page
5. Change Report Region title to **Buckets** and click **Next**
6. Choose **No** to create items for Host, Authorization, and Date. They will be populated with values other than items on this page
7. Click **Next**
8. Check Name in the parameter list and click **Finish**
9. Click **Edit Page**

10. Delete the Buckets HTML region (not the report region) which was created during the create application wizard. This region is not necessary.
11. Edit the **doREST** region select **Never** for the Condition Type under the Conditions section.
12. Edit the page process called Web Service Request and make the following changes
 - a. Process Point: **On Load - Before Header**
 - b. Input Parameters
 - i. Host: Static Value, **s3.amazonaws.com**
 - ii. Authorization: Function, **return amazon_signature_sh1('GET'||chr(10)||chr(10)||chr(10)||:S3_DATE_TIME||chr(10)||'/');**
 - iii. Date: Item, **S3_DATE_TIME**
 - c. Click **Apply Changes**
13. Run the page and verify that you see a report of bucket names (you should create a test bucket and add some test objects with another client, like S3Fox⁴ prior to running this page)
- 14.

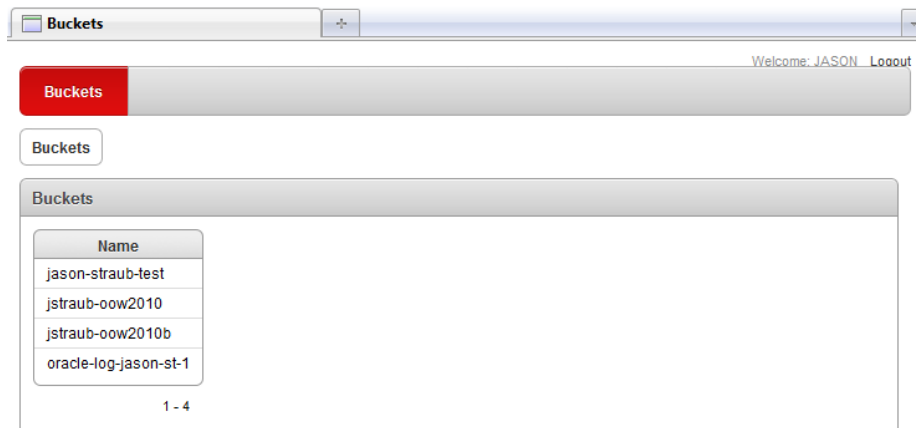


Figure 1, List all Buckets Page

CREATE A PAGE TO CREATE A NEW BUCKET

Now that you can view your buckets, your application should provide the ability to create a new one.

⁴ <http://www.s3fox.net/>

Create the RESTful Web Service Reference

1. Navigate to Shared Components, Web Service References
2. Click **Create**
3. Choose **REST** from the Web reference type and click **Next**
4. Enter the following on the REST Details page
 - a. Enter **Create Bucket** in the Name field
 - b. Enter http://&P2_NAME..s3.amazonaws.com/ in the URL field
 - c. Choose **PUT** as the HTTP Method and accept the default for Basic Authentication
 - d. Enter **Host** in the Name field under HTTP Headers and click **Add Header**
 - e. Repeat step d and create **Authorization** and **Date** headers
 - f. Click **Next**
5. Click **Next** on the REST Inputs page since this service has no parameters
6. Choose **Text** on the REST Output Format with no other modifications and click **Create**. The only response from the service should be an HTTP status code if the bucket is successfully created.

Create the Page

1. On the Web Service Reference success page, click **Create Form on Web Service**
2. Choose **Create Bucket** from the Web Service Reference List and click **Next**
3. On the Page and Region Attributes step, change the following:
 - a. Page Number: **2**
 - b. Region Title: **Create Bucket**
 - c. Breadcrumb: **Breadcrumb**, Parent Entry – **Buckets**
4. Click **Next**
5. Choose **No** for Host, Authorization, and Date to not create those items, they will be populated with function results and an application item.
6. Click **Next**
7. Click **Finish**
8. Click **Edit Page**
9. Add text item **P2_NAME** for the bucket name
 - a. Right-click on the Create Bucket region and choose **Create Page Item**

- b. Choose **Text Field** and click **Next**
 - c. Enter **P2_NAME** in the Item Name field and click **Next**
 - d. Accept Item Attribute defaults and click **Next**
 - e. Accept Settings defaults and click **Next**
 - f. Click **Create Item**
10. Edit the page process called Web Service Request and make the following changes
 - a. Input Parameters
 - i. Host: Function, **return**
:P2_NAME||'.s3.amazonaws.com';
 - ii. Authorization: Function, **return**
amazon_signature_sh1('PUT'||chr(10)||chr(10)||chr(10)||:S3_DATE_TIME||chr(10)||'/'||:P2_NAME||'/');
 - iii. Date: Item, **S3_DATE_TIME**
 - b. Success Message: **Bucket created**
 - c. Click **Apply Changes**
11. Edit the branch to Page 2 and change it to Page 1
12. Go to the page definition of page 1
13. Create a button with the following attributes in the region called Buckets:
 - a. Right click on the Buckets region and choose **Create Region Button**
 - b. Button Name: **CREATE**
 - c. Label: **Create Bucket**
 - d. Position: **Region Template Position #CREATE#**
 - e. Action: Redirect to Page
 - i. Page **2**
 - ii. Clear Cache **2**
 - f. Click **Create Button**
14. Run page 1, click **Create Bucket**, enter a name, and then verify you can create a bucket



Figure 2, Bucket successfully created

CREATE A PAGE TO LIST BUCKET CONTENTS

Logically the next step is to create a page that will list all the objects contained in a bucket.

Create the RESTful Web Service Reference

1. Navigate to Shared Components, Web Service References
2. Click **Create**
3. Choose **REST** from the Web reference type and click **Next**
4. Enter the following on the REST Details page
 - a. Enter **List Bucket Contents** in the Name field
 - b. Enter http://s3.amazonaws.com/&P3_NAME in the URL field
 - c. Enter **Host** in the Name field under HTTP Headers and click **Add Header**
 - d. Repeat step c and create **Authorization** and **Date** headers
 - e. Click **Next**
5. Since there are no input parameters, click **Next**
6. Enter **/ListBucketResult/Contents** in the XPath field
7. Enter <http://s3.amazonaws.com/doc/2006-03-01/> in the Response Namespace field
8. Enter **LastModified** in the output parameter Name field, tab to Path field

9. Click **Add Parameter**. Enter **Key** in the Name field and tab to Path field.
10. Click **Create**

Create the Page

1. On the Web Service Reference success page, click **Create Form and Report on Web Service**
2. Choose **List Bucket Contents** from the Web Service Reference select list and click **Next**
3. Change the following on the Page and Region Attributes step:
 - a. Page: **3**
 - b. Report Region Title: **Contents**
 - c. Breadcrumb: **Breadcrumb**, Entry Name - **&P3_NAME**.
 - d. Parent Entry: **Buckets**
4. Click **Next**
5. Choose **No** for Host, Authorization, and Date. They will be populated differently, click **Next**
6. On the Report Parameters page:
 - a. Change collection name to **P3_BUCKET_CONTENTS**
 - b. Check all parameters
7. Click **Finish**
8. Click **Edit Page**
9. Add a hidden item to the doREST region called **P3_NAME**
10. Edit the doREST region and select **Never** for the Condition Type under Conditions
11. Edit the page process called Web Service Request and make the following changes
 - a. Process Point: **On Load - Before Header**
 - b. Input Parameters
 - i. Host: Function, **return**
:P3_NAME||'.s3.amazonaws.com';
 - ii. Authorization: Function, **return**
amazon_signature_sh1('GET'||chr(10)||chr(10)||chr(10)||:S3_DATE_TIME||chr(10)||'/:P3_NAME);
 - iii. Date: Item, **S3_DATE_TIME**

- c. Click **Apply Changes**
12. Go to the page definition of page 1
13. Modify the report attributes of the Buckets region, column Name, supplying the following link information:
 - a. Link Text: **#Name#**
 - b. Page: **3**
 - c. Item 1: **P3_NAME**
 - d. Value: **#Name#**
14. Run page 1 and then click on the test bucket you created with objects in it (if you have not done this yet, now would be a good time). Verify that you can view objects in a bucket.

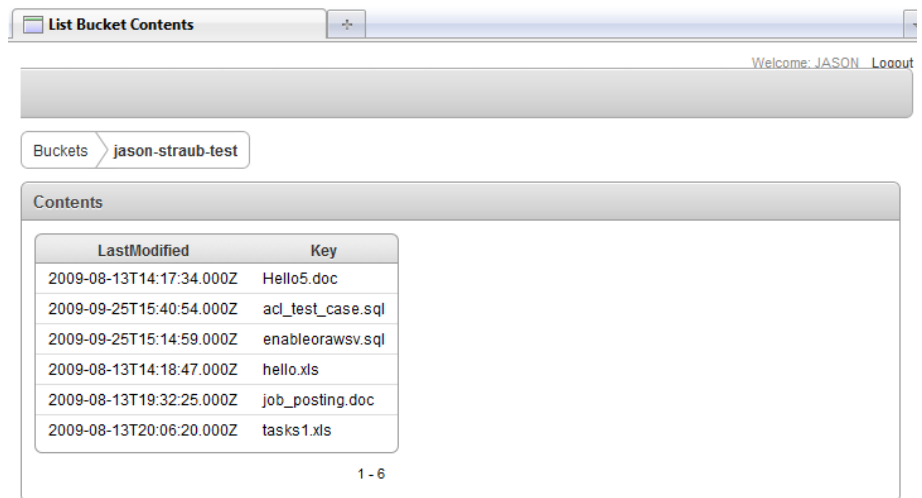


Figure 3, Listing bucket contents

Modify Report to Provide Download Link for Objects

You construct a link to objects using query string authentication, so you can provide direct access to the objects through a URL. The signature is added to the query string of the URL. You provide a URL expiration which is represented as an epoch (number of seconds since January 1, 1970). You create an item on the page and a computation to calculate the epoch and then modify the query that displays the bucket contents to include the link to download the object.

1. Edit page 3 and create a hidden item name **P3_EPOCH** in the doREST region

2. Right-click on **P3_EPOCH** and choose **Create Computation**. Create a computation with the following options
 - a. Computation Point: **Before Header**
 - b. Computation Type: **PL/SQL Function Body**
 - c. Click **Next** >
 - d. Enter **return (trunc(sysdate + 2) - (to_date('01-01-1970','MM-DD-YYYY')))*24*60*60;** in the Computation text area
 - e. Click **Create**
3. Modify the query in the Contents region, changing it to the following:

```

select
extractValue(value(t), '/*/LastModified', 'xmlns="http://s
3.amazonaws.com/doc/2006-03-01/"') "LastModified"
, '<a
href="http://'||:P3_NAME||'.s3.amazonaws.com/'||extractV
alue(value(t), '/*/Key', 'xmlns="http://s3.amazonaws.com/d
oc/2006-03-
01/"')||'?AWSAccessKeyId='||amazon_aws_id_only||'&Expire
s='||:P3_EPOCH||'&Signature='||wwv_flow_utilities.url_en
code2(amazon_sig_only('GET' ||chr(10)||chr(10)||chr(10)||
:P3_EPOCH||chr(10)||'/'||:P3_NAME||'/'||extractValue(val
ue(t), '/*/Key', 'xmlns="http://s3.amazonaws.com/doc/2006-
03-
01/"'))||'">'||extractValue(value(t), '/*/Key', 'xmlns="h
ttp://s3.amazonaws.com/doc/2006-03-01/"')||'</a>' "Name"
from wwv_flow_collections c,

table(xmlsequence(extract(c.xmltype001, '/ListBucketResul
t/Contents', 'xmlns="http://s3.amazonaws.com/doc/2006-03-
01/"')) t
where c.collection_name = 'P3_BUCKET_CONTENTS'

```

4. Modify the new Name column, changing Display as to **Standard Report Column**
5. Run page 3 and verify you can download documents

CREATE A PAGE TO UPLOAD AN OBJECT TO A BUCKET

Uploading new files for storage on Amazon S3 should be one of the primary functions of any Amazon S3 client. This client is no different and you create a page to upload new files (objects). This page will demonstrate Oracle Application Express's support for using the PUT method in RESTful requests as well as support for using a file from a file upload item as the input to a web service.

Create the RESTful Web Service Reference

1. Navigate to Shared Components, Web Service References
2. Click **Create**
3. Choose **REST** from the Web reference type and click **Next**
4. Enter the following on the REST Details page
 - a. Enter **Create Object** in the Name field
 - b. Enter http://s3.amazonaws.com/&P3_NAME./&P4_NAME. in the URL field
 - c. Choose **PUT** as the HTTP Method
 - d. Enter **Host** in the Name field under HTTP Headers and click **Add Header**
 - e. Repeat step d and create **Authorization**, **Date**, and **Content-Type** headers
 - f. Click **Next**
5. Choose **File Upload Item** on the REST Inputs step and click **Next**
6. Choose **Text** as the Output Format on the REST Outputs step and click **Create**

Create the Page

1. On the Web Service Reference success page, click **Create Form on Web Service**
2. Choose Create Object and click **Next**
3. Change the following on the Page and Region Attributes step:
 - a. Page: **4**
 - b. Region Title: **Create Object**
 - c. Breadcrumb: **Breadcrumb**
 - d. Parent Entry: **&P3_NAME.**
4. Click **Next**
5. Choose not to create items for Host, Authorization, and Date on the REST Input Parameters page, change P4_CONTENT-TYPE name to **P4_CONTENT_TYPE** and click **Next**
6. Click **Finish**
7. Click **Edit Page**
8. Edit the P4_CONTENT_TYPE item and make it hidden

9. Create a hidden item **P4_NAME**
10. Right click the **P4_CONTENT_TYPE** item and choose **Create Computation**. Create the computation with the following options:
 - a. Computation Point: **After Submit**
 - b. Computation Type: **SQL Query (return single value)**
 - c. Click **Next**
 - d. Enter the following in the Computation text area:
select mime_type from apex_application_files where name = :P4_FILE
 - e. Click **Create**
11. Right click the **P4_NAME** item and choose **Create Computation**. Create a computation with the following options:
 - a. Computation Point: **After Submit**
 - b. Computation Type: **SQL Query (return single value)**
 - c. Click **Next**
 - d. Enter the following in the Computation text area:
select filename from apex_application_files where name = :P4_FILE
 - e. Click **Create**
12. Edit the page process called Web Service Request and make the following changes
 - a. Input Parameters
 - i. Host: Function, **return**
:P3_NAME||'.s3.amazonaws.com';
 - ii. Authorization: Function, **return**
amazon_signature_sh1('PUT'||chr(10)||chr(10)||:P4_CONTENT_TYPE||chr(10)||:S3_DATE_TIME||chr(10)||'/'||:P3_NAME||'/'||:P4_NAME);
 - iii. Date: Item, **S3_DATE_TIME**
 - b. Process Success Message: **Object added**
 - c. Click **Apply Changes**
13. Edit the Branch to Page 4, change the branch to page 3
14. Go to the page definition of page 3
15. Create a button with the following attributes in the region called Contents:
 - a. Create a button in a region position

- b. Button Name: **CREATE**
- c. Label: **Create Object**
- d. Position: Region Template Position #CREATE#
- e. Action: Redirect to Page in this Application
- f. Page **4**
- g. Clear Cache **4**
- h. Click **Create Button**

16. Run page 3, click **Create Object** and verify that you can upload a new item

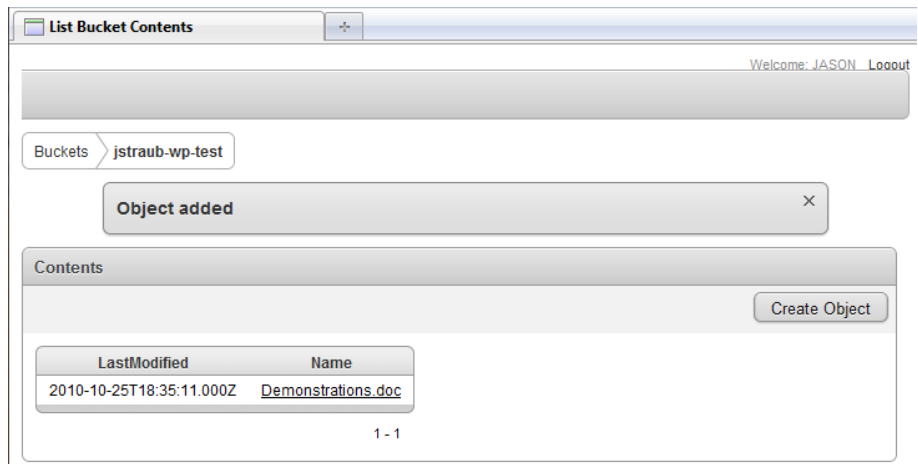


Figure 4, Add object to a bucket

ADD ABILITY TO DELETE AN OBJECT

The final feature to add to the S3 client is the ability to delete an object. The ability to delete a bucket and its contents is an exercise left to the reader.

Create the RESTful Web Service Reference

1. Navigate to Shared Components, Web Service References
2. Click **Create**
3. Choose **REST** from the Web reference type and click **Next**
4. Enter the following on the REST Details page
 - a. Enter **Delete Object** in the Name field
 - b. Enter http://s3.amazonaws.com/&P3_NAME./&P3_DEL_NAME. in the URL field
 - c. Choose **DELETE** as the HTTP Method

- d. Enter **Host** in the Name field under HTTP Headers and click **Add Header**
 - e. Repeat step d and create **Authorization** and **Date** headers
 - f. Click **Next**
5. There are no other inputs to the service so on the REST Inputs step simply click **Next**
 6. Choose **Text** as the Output Format on the REST Outputs step and click **Create**

Modify the Report on Page 3 to Include a Delete Link

First, create a hidden item on page 3 called P3_DEL_NAME. This item will hold the name of the object to be deleted. Then modify the query of the Contents region to select a third column, which is simply the object name. Next, modify the column properties of that new column to link to a URL which calls JavaScript to set the P3_DEL_NAME item and submit the page.

1. Navigate to the page definition of page 3
2. Right click on the Contents region and choose **Create Page Item**
3. Complete the wizard to create a hidden item called **P3_DEL_NAME**. Ensure the Value Protected attribute is set to **No** because you will need to set the value of this item through JavaScript
4. Edit the query and replace it with the following new query:

```

select
extractValue(value(t), '/*/LastModified', 'xmlns="http://s
3.amazonaws.com/doc/2006-03-01/"') "LastModified"
, '<a
href="http://'||:P3_NAME||'.s3.amazonaws.com/'||extractV
alue(value(t), '/*/Key', 'xmlns="http://s3.amazonaws.com/d
oc/2006-03-
01/"')||'?AWSAccessKeyId='||amazon_aws_id_only||'&Expire
s='||:P3_EPOCH||'&Signature='||wwv_flow_utilities.url_en
code2(amazon_sig_only('GET' ||chr(10)||chr(10)||chr(10)||
:P3_EPOCH||chr(10)||'/'||:P3_NAME||'/'||extractValue(val
ue(t), '/*/Key', 'xmlns="http://s3.amazonaws.com/doc/2006-
03-
01/"'))||'">'||extractValue(value(t), '/*/Key', 'xmlns="h
ttp://s3.amazonaws.com/doc/2006-03-01/"')||'</a>'
"Name" ,
extractValue(value(t), '/*/Key', 'xmlns="http://s3.amazona
ws.com/doc/2006-03-01/"') d
from wwv_flow_collections c,

table(xmlsequence(extract(c.xmltype001, '/ListBucketResul
t/Contents', 'xmlns="http://s3.amazonaws.com/doc/2006-03-
01/"')) t
where c.collection_name = 'P3_BUCKET_CONTENTS'

```

5. Edit the D report column and make the following changes:

- a. Column Heading: ** **
- b. Link Text: [**Delete**]
- c. Target: **URL**
- d. URL: **javascript:\$s('P3_DEL_NAME', '#D#'); apex.submit('DELETE');**
- e. Click Apply Changes

Create Process to Call the Delete Object Web Service Reference

The next step is to create a process on the page that uses the Delete Object web service reference to call the Amazon S3 web service and delete the object. To create the process:

1. Navigate to the page definition of page 3
2. Under Page Processing, right-click on Processing and choose **Create Process**
3. Run the Create Page Process wizard specifying the following options:
 - a. Process Category: **Web Services**
 - b. Name: **call delete object**
 - c. Web Service Reference: **Delete Object**

- d. Operation: **doREST**
- e. Input Parameters:
 - i. Host: Function, **return**
:P3_NAME||'.s3.amazonaws.com';
 - ii. Authorization: Function, **return**
amazon_signature_sh1('DELETE'||chr(10)||chr(10)||chr(10)||:S3_DATE_TIME||chr(10)||'/'||:P3_NAME||'/'||:P3_DEL_NAME);
 - iii. Date: Item, **S3_DATE_TIME**
4. Click **Next**
5. Enter **Object deleted** in the Success Message
6. Enter **Error deleting object** in the Failure Message
7. Click **Next**
8. Choose **Request = Expression 1** from the Condition Type list
9. Enter **DELETE** in the Expression 1 field
10. Click **Create Process**
11. Run page 3 and verify that you can successfully delete an object

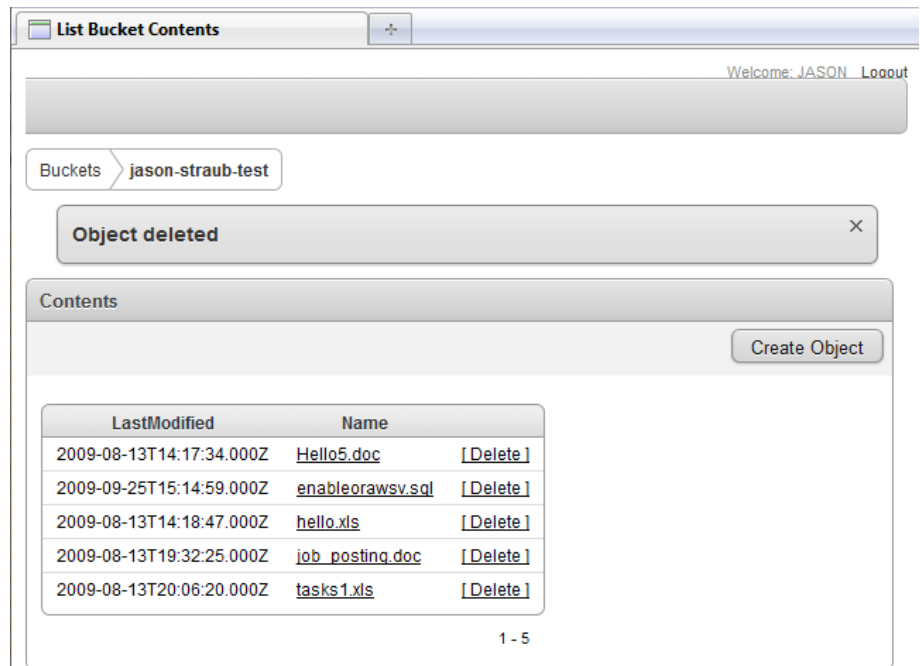


Figure 5, Object successfully deleted

CONCLUSION

Amazon S3 is a very useful platform for storing content or for off-site backups on the internet. The Amazon S3 service makes it easy to build clients on the S3 platform.

Oracle Application Express 4.0 introduces the ability to consume and interact with RESTful style web services. This paper has shown how you can quickly and declaratively build an Amazon S3 client using Oracle Application Express's new support for RESTful style web services.



Building an Amazon S3 Client with Application Express 4.0

December 2010

Author: Jason Straub

Contributors: Terri Jennings, Chaitanya Koratamaddi

**Oracle Corporation
World Headquarters
500 Oracle Parkway
Redwood Shores, CA 94065
U.S.A.**

**Worldwide Inquiries:
Phone: +1.650.506.7000
Fax: +1.650.506.7200
www.oracle.com**

**Oracle Corporation provides the software
that powers the internet.**

**Oracle is a registered trademark of Oracle Corporation. Various
product and service names referenced herein may be trademarks
of Oracle Corporation. All other product and service names
mentioned may be trademarks of their respective owners.**

**Copyright © 2010 Oracle Corporation
All rights reserved.**