

User Manual:

This application has the ability to upload/download files with any size and also users can get the list of available files in cache and distributed data store. Moreover, they can get the location and availability of files in cache or distributed data store. The steps of using this application is described below:

1. Connect to the application:
use this link <http://3.swift-casing-752.appspot.com/> to see the webpage of the application. The interface of the application is shown in figure 1. Google Chrome and Firefox are the recommended browser for using this application.

Please browse to select file(s) to upload to the distributed storage system:

Upload using Google Cloud Storage API - Files with size less than 100MB:

No file chosen

Upload using Blobstore API, for any size:

No file chosen

Write the name of the file to check if it is in the distributed storage system or not:

Files in Google Cloud Storage:

Click on the file-name to download the file

0D7mSFR0f.txt	1024 Bytes	<input type="button" value="Delete"/>	<input type="text"/>	<input type="button" value="FindInFile"/>
4QoWgOaD0l.txt	10483760 Bytes	<input type="button" value="Delete"/>	<input type="text"/>	<input type="button" value="FindInFile"/>
G3lsTlCa's.txt	10240 Bytes	<input type="button" value="Delete"/>	<input type="text"/>	<input type="button" value="FindInFile"/>
K3xM2SGYR.txt	102400 Bytes	<input type="button" value="Delete"/>	<input type="text"/>	<input type="button" value="FindInFile"/>
No7Lfjwst.txt	102400 Bytes	<input type="button" value="Delete"/>	<input type="text"/>	<input type="button" value="FindInFile"/>
ON7HCa7Ga.txt	102400 Bytes	<input type="button" value="Delete"/>	<input type="text"/>	<input type="button" value="FindInFile"/>
UUn1S7EB.txt	102400 Bytes	<input type="button" value="Delete"/>	<input type="text"/>	<input type="button" value="FindInFile"/>
ZuxKZisQYT.txt	1048376 Bytes	<input type="button" value="Delete"/>	<input type="text"/>	<input type="button" value="FindInFile"/>
zUG0IH4E6a.txt	10240 Bytes	<input type="button" value="Delete"/>	<input type="text"/>	<input type="button" value="FindInFile"/>

Files in Cache:

Click on the file-name to download the file

Remove all files in Google Cloud Storage:

Remove all files in Cache:

Find Total Number of Files in Google Cloud Storage:

Find Total Number of Files in Cache:

Find total space allocated in Google Cloud Storage:

Find total space allocated in Cache:

Search for regular expression in all files:

Figure 1

2. Uploading files:

According to [fig. 1], in order to upload a file, users should press “Choose File” button and then select the files which they want. (It is possible to select numbers of file together). After choosing file/files, users should press the “upload” button. At this time the files will be uploaded to cache and/or Google Cloud Storage (based on their size). There are two methods to upload files.

- ✓ Using Google Cloud Storage API
Through this API, users can upload files with the size less than 100MB.
- ✓ Using BlobStore API
Through this API, users can upload files with any size.

3. Listing available files in Google Cloud Storage and Cache:

There is two list at the end of the web page which belongs to Cache and Google Cloud Storage. [fig. 2] At the time which web page starts these lists are available. As long as users upload their files these lists will be updated.

Files in Google Cloud Storage:

Click on the file-name to download the file

0D7rmSFR0f.txt	1024 Bytes	Delete
0Lgv1vr2pU.txt	102400 Bytes	Delete
0aQrFC7cDZ.txt	10240 Bytes	Delete
0pzgm6Ffag.txt	10240 Bytes	Delete
0rxByUlg2.txt	10240 Bytes	Delete
GdY3a2Zzz3.txt	102400 Bytes	Delete
gpcaaUDMiW.txt	1048576 Bytes	Delete

Files in Cache:

Click on the file-name to download the file

GdY3a2Zzz3.txt	102400 Bytes	Delete
0rxByUlg2.txt	10240 Bytes	Delete
0pzgm6Ffag.txt	10240 Bytes	Delete
0Lgv1vr2pU.txt	102400 Bytes	Delete
0D7rmSFR0f.txt	1024 Bytes	Delete
0aQrFC7cDZ.txt	10240 Bytes	Delete

Figure 2

4. Checking the location and the availability of files:

According to Figure 3, there is a text box which user can type the name of desired file. After pressing the “Check” button one alert appears in browser which says the availability of file in Cache and/or Google Cloud Storage.

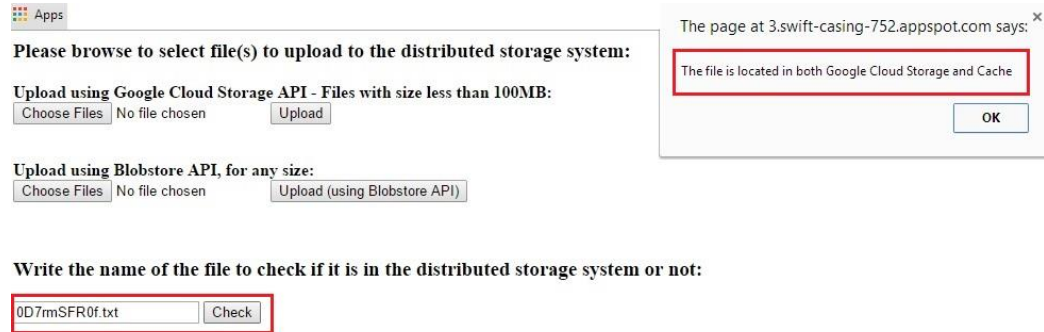


Figure 3

5. Removing Files:

According to Figure 2, there are two lists which shows the available files in Cache and Google Cloud Storage. In front of each file there is a “Delete” button. Users can remove files by pressing this button.

6. Downloading files:

In order to download a file, first, users should press “right click” and second they should choose “save link as...” option. [Fig. 4]

Files in Google Cloud Storage:

Click on the file-name to download the file

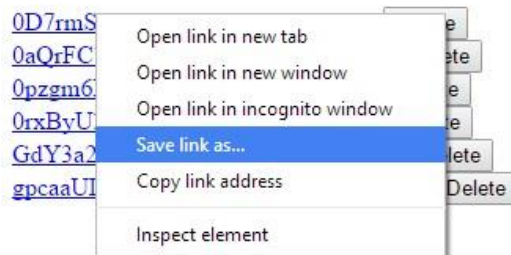


Figure 4

7. Remove All Files:

The application has the ability to remove all files at one time. It can remove files which is located at distributed dataset or cache. In order to remove all files users should click one of the button which is shown in [Figure 5]

Remove all files in Google Cloud Storage:

Remove All Files (GCS)

Remove all files in Cache:

Remove All Files (Cache)

Figure 5

8. Find total number of available files in storage:

The application has the ability to show the number of available files in cache or distributed dataset. User should press the button which is shown in [Fig. 6].

Find Total Number of Files in Google Cloud Storage:

count All Files (GCS)

Find Total Number of Files in Cache:

count All Files (Cache)

Figure 6

9. Find the size of free space in storage:

Users can see the free space of distributed dataset or cache. [Fig. 7]

Find total space allocated in Google Cloud Storage:

Files Storage Size (GCS)

Find total space allocated in Cache:

Files Storage Size (Cache)

Figure 7

10. List all files which has one desired regular expression in their File:

Users should type the regular expression in the text box [fig. 8]. After pressing the “Search Files” button, one new tab open which contain the list of desire files.

Search for regular expression in all files:



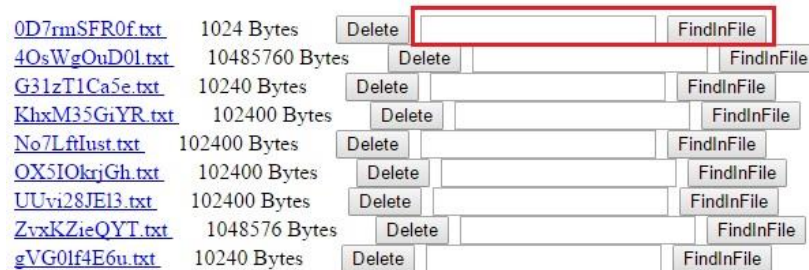
The figure shows a simple search interface. It consists of a rectangular text input field followed by a button labeled "Search Files".

Figure 8

11. Find a regular expression in the name of files:

This application has the ability to search regular expression in each file.[Fig. 9]

In order to find desired file, users should type the regular expression in front of files and then they should press “FindInFile” button. After pressing the button one alert appear which shown the status of availability of that regular expression on selected file.



The figure shows a table of files with the following columns: File Name, Size, Delete, and FindInFile. The first row is highlighted with a red border.

File Name	Size	Delete	FindInFile
0D7rmSFR0f.txt	1024 Bytes	Delete	FindInFile
4OsWgOuD0l.txt	10485760 Bytes	Delete	FindInFile
G31zTlCa5e.txt	10240 Bytes	Delete	FindInFile
KhxM35GiYR.txt	102400 Bytes	Delete	FindInFile
No7LftIust.txt	102400 Bytes	Delete	FindInFile
OX5IOkrjGh.txt	102400 Bytes	Delete	FindInFile
UUvi28JE13.txt	102400 Bytes	Delete	FindInFile
ZvxKZieQYT.txt	1048576 Bytes	Delete	FindInFile
gVG0lf4E6u.txt	10240 Bytes	Delete	FindInFile

Figure 9

Deployment Manual:

There are numbers of software which are used during the deployment of this application. The list of these software is shown below:

- ✓ Apache Maven, version: 3.2.3
- ✓ Java, version: 1.7.0_17
- ✓ Google Cloud SDK, version: 0.9.34

The steps of compiling this project is explained below:

1. Change the bucket name in all source files (JSP and Java Files)
2. The created bucket needs to be publicly accessible for read. Users should issue the following commands to add public-read access to the ACL of their bucket from their console:
 - ✓ To login to gsutil if users haven't already logged in:
gsutil config
 - ✓ To give public-read access to the current files on the bucket (if any):
gsutil -m setacl -R -a public-read gs://BUCKET_NAME
 - ✓ To set public-read access as the default ACL for future uploaded files:
gsutil -m setdefacl public-read gs://BUCKET_NAME
3. In order to compile and upload the application on google cloud. First, users should change the directory to the directory which the project is available. Second, They should issue this command:
mvn appengine:update
4. After users have deployed the app on Google App Engine, they need to open the app using the following url: **<http://<app-name>.appspot.com/upload>**
Google Chrome is the recommended browser.

The current version of the code right now is 3. Users can change the version in "appengine-web.xml".