JAVA SWIFT EXAMPLES

SETUP

The following examples may require some or all of the following Java classes to be imported:

```
import java.io.File;
import java.util.List;
import java.util.Map;
import com.rackspacecloud.client.cloudfiles.FilesClient;
import com.rackspacecloud.client.cloudfiles.FilesConstants;
import com.rackspacecloud.client.cloudfiles.FilesContainer;
import com.rackspacecloud.client.cloudfiles.FilesContainerExistsException;
import com.rackspacecloud.client.cloudfiles.FilesObject;
import com.rackspacecloud.client.cloudfiles.FilesObjectMetaData;
```

CREATE A CONNECTION

This creates a connection so that you can interact with the server:

```
String username = "USERNAME";
String password = "PASSWORD";
String authUrl  = "https://objects.dreamhost.com/auth";

FilesClient client = new FilesClient(username, password, authUrl);
if (!client.login()) {
        throw new RuntimeException("Failed to log in");
}
```

CREATE A CONTAINER

This creates a new container called `my-new-container`:

```
client.createContainer("my-new-container");
```

CREATE AN OBJECT

This creates an object `foo.txt` from the file named `foo.txt` in the container `my-new-container`:

```
File file = new File("foo.txt");
String mimeType = FilesConstants.getMimetype("txt");
client.storeObject("my-new-container", file, mimeType);
```

ADD/UPDATE OBJECT METADATA

This adds the metadata key-value pair key:value to the object named `foo.txt` in the container `my-new-container`:

```
FilesObjectMetaData metaData = client.getObjectMetaData("my-new-container", "foo.txt");
metaData.addMetaData("key", "value");

Map<String, String> metamap = metaData.getMetaData();
client.updateObjectMetadata("my-new-container", "foo.txt", metamap);
```

LIST OWNED CONTAINERS

This gets a list of Containers that you own. This also prints out the container name.

```
List<FilesContainer> containers = client.listContainers();
for (FilesContainer container : containers) {
        System.out.println("  " + container.getName());
}
```

The output will look something like this:

```
mahbuckat1
mahbuckat2
mahbuckat3
```

LIST A CONTAINER’S CONTENT

This gets a list of objects in the container `my-new-container`; and, it also prints out each object’s name, the file size, and last modified date:

```
List<FilesObject> objects = client.listObjects("my-new-container");
for (FilesObject object : objects) {
        System.out.println("  " + object.getName());
}
```

The output will look something like this:

```
myphoto1.jpg
myphoto2.jpg
```

RETRIEVE AN OBJECT’S METADATA

This retrieves metadata and gets the MIME type for an object named `foo.txt` in a container named `my-new-container`:

```
FilesObjectMetaData metaData =  client.getObjectMetaData("my-new-container", "foo.txt");
String mimeType = metaData.getMimeType();
```

RETRIEVE AN OBJECT

This downloads the object `foo.txt` in the container `my-new-container` and saves it in `./outfile.txt`:

```
FilesObject obj;
File outfile = new File("outfile.txt");

List<FilesObject> objects = client.listObjects("my-new-container");
for (FilesObject object : objects) {
        String name = object.getName();
        if (name.equals("foo.txt")) {
                obj = object;
                obj.writeObjectToFile(outfile);
        }
}
```

DELETE AN OBJECT

This deletes the object goodbye`.txt` in the container “my-new-container”:

```
client.deleteObject("my-new-container", "goodbye.txt");
```

DELETE A CONTAINER

This deletes a container named “my-new-container”:

```
client.deleteContainer("my-new-container");
```

Note: The container must be empty! Otherwise it won’t work!