# JAVA SWIFT EXAMPLES

## SETUP

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

```
import org.javaswift.joss.client.factory.AccountConfig;
import org.javaswift.joss.client.factory.AccountFactory;
import org.javaswift.joss.client.factory.AuthenticationMethod;
import org.javaswift.joss.model.Account;
import org.javaswift.joss.model.Container;
import org.javaswift.joss.model.StoredObject;
import java.io.File;
import java.io.IOException;
import java.util.*;
```

## CREATE A CONNECTION

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

```
String username = "USERNAME";
String password = "PASSWORD";
String authUrl  = "https://radosgw.endpoint/auth/1.0";

AccountConfig config = new AccountConfig();
config.setUsername(username);
config.setPassword(password);
config.setAuthUrl(authUrl);
config.setAuthenticationMethod(AuthenticationMethod.BASIC);
Account account = new AccountFactory(config).createAccount();
```

## CREATE A CONTAINER

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

```
Container container = account.getContainer("my-new-container");
container.create();
```

## CREATE AN OBJECT

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

```
Container container = account.getContainer("my-new-container");
StoredObject object = container.getObject("foo.txt");
object.uploadObject(new File("foo.txt"));
```

## 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`:

```
Container container = account.getContainer("my-new-container");
StoredObject object = container.getObject("foo.txt");
Map<String, Object> metadata = new TreeMap<String, Object>();
metadata.put("key", "value");
object.setMetadata(metadata);
```

## LIST OWNED CONTAINERS

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

```
Collection<Container> containers = account.list();
for (Container currentContainer : containers) {
    System.out.println(currentContainer.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:

```
Container container = account.getContainer("my-new-container");
Collection<StoredObject> objects = container.list();
for (StoredObject currentObject : objects) {
    System.out.println(currentObject.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`:

```
Container container = account.getContainer("my-new-container");
StoredObject object = container.getObject("foo.txt");
Map<String, Object> returnedMetadata = object.getMetadata();
for (String name : returnedMetadata.keySet()) {
    System.out.println("META / "+name+": "+returnedMetadata.get(name));
}
```

## RETRIEVE AN OBJECT

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

```
Container container = account.getContainer("my-new-container");
StoredObject object = container.getObject("foo.txt");
object.downloadObject(new File("outfile.txt"));
```

## DELETE AN OBJECT

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

```
Container container = account.getContainer("my-new-container");
StoredObject object = container.getObject("foo.txt");
```

```
object.delete();
```

## DELETE A CONTAINER

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

```
Container container = account.getContainer("my-new-container");
container.delete();
```

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