

public byte[] download(String urlString) {

log.info(" -> Image url = {}", urlString);

KeyStore ks = null;

try (**InputStream clientCert = this.getClass().getClassLoader().getResourceAsStream("test-com.p12"**);

**InputStream trustStoreIS = this.getClass().getClassLoader().getResourceAsStream("jssecacerts")**

) {

**ks = KeyStore.getInstance("PKCS12");**

**ks.load(clientCert, "testplm".toCharArray());**

**KeyManagerFactory** kmf = KeyManagerFactory.getInstance(KeyManagerFactory.getDefaultAlgorithm());

**kmf.init(ks, "testplm".toCharArray());**

KeyManager[] kms = kmf.getKeyManagers();

**KeyStore trustStore = KeyStore.getInstance("JKS");**

**trustStore.load(trustStoreIS, "changeit".toCharArray());**

**TrustManagerFactory** tmf = TrustManagerFactory.getInstance(TrustManagerFactory.getDefaultAlgorithm());

**tmf.init(trustStore);**

TrustManager[] tms = tmf.getTrustManagers();

SSLContext sc = SSLContext.getInstance("TLS");

**sc.init(kms, tms, new SecureRandom());**

HttpsURLConnection.setDefaultSSLSocketFactory(sc.getSocketFactory());

} catch (KeyStoreException | IOException | NoSuchAlgorithmException | CertificateException | KeyManagementException | UnrecoverableKeyException e) {

log.error(e.getMessage(), e);

}

// end of example. Further there is a usage

URL url;

String imagePostfix;

File image = null;

HttpsURLConnection connection = null;

try {

url = new URL(urlString);

if (url.getHost().equalsIgnoreCase("prod.com")) {

user = "ee";

password = "bb";

} else if (url.getHost().equalsIgnoreCase("qa.com")) {

user = "st";

password = "ql";

} else if (url.getHost().equalsIgnoreCase("test.com")) {

user = "s";

password = "A";

}

imagePostfix = getImagePostfix(url);

connection = (HttpsURLConnection) url.openConnection();

if (user != null && password != null) {

log.info("User's name and password for auth in cloud is recieved: TRUE");

} else log.error("User's name and password for auth in cloud is recieved: FALSE");

String authString = user + ":" + password;

byte[] authEncBytes = Base64.encodeBase64(authString.getBytes());

String authStringEnc = new String(authEncBytes);

connection.setRequestProperty("Authorization", "Basic " + authStringEnc);

log.info("-> connection to {}. Response code = {}", url, connection.getResponseCode());

InputStream inputStream = connection.getInputStream();

// image = File.createTempFile("plm-", imagePostfix);

image = new File("/media/data/plm-service" + File.separator + "plm-" + new Date().getTime() + imagePostfix);

image.createNewFile();

log.info("Temp file created. Name = " + image.getAbsolutePath());

OutputStream out = new FileOutputStream(image);

IOUtils.copy(inputStream, out);

out.flush();

out.close();

inputStream.close();

connection.disconnect();

return IOUtils.readBytesFromStream(new FileInputStream(image));

} catch (MalformedURLException e) {

log.error("Error in Image Downloader's URL = " + urlString.toString(), e);

} catch (IOException e) {

log.error(e.getClass().getSimpleName() + ": " + e.getMessage(), e);

}

return null;