Ejemplo de creación del encabezado y envío de petición en JAVA

```
import java.io.IOException;
import java.io.StringReader;
import java.io.StringWriter;
import java.net.URI;
import java.security.GeneralSecurityException;

import javax.crypto.Mac;
import javax.crypto.spec.SecretKeySpec;
import javax.xml.transform.OutputKeys;
import javax.xml.transform.Source;
import javax.xml.transform.Transformer;
import javax.xml.transform.TransformerFactory;
import javax.xml.transform.stream.StreamResult;
import javax.xml.transform.stream.StreamSource;

import org.apache.commons.codec.binary.Base64;
import org.apache.http.HttpEntity;
import org.apache.http.HttpResponse;
import org.apache.http.client.methods.HttpGet;
import org.apache.http.impl.client.CloseableHttpClient;
import org.apache.http.impl.client.HttpClientBuilder;
import org.apache.http.message.BasicHeader;
import org.apache.http.util.EntityUtils;

/**
 * Pricing API call sample code.
 *
 * @author payu.com.co
 * @version 1.0
 * @since 1/09/2015
 */
public class PruebaPricing {

	private final String API_URL = "http://stg.api.payulatam.com/payments-
api/rest/v4.3/pricing?accountId=509171&currency=ARS&amount=500&paymentMethod=VIS
A";
	private final String MERCHANT_PUBLIC_KEY = "PKfm6N499761MiCW566M9okj0N";
	private final String MERCHANT_API_KEY = "6u39nqhq8ftd0hlvnjfs66eh8c";

	/**
	 * The main method.
	 * @param args the arguments
	 */
	public static void main(String[] args) {

		PruebaPricing sample = new PruebaPricing();
		sample.doRequest();
```

```
	}

	/**
	 * Do the request.
	 */
	public void doRequest() {

		CloseableHttpClient client = null;
		try {
			// send request
			client = HttpClientBuilder.create().build();
			HttpResponse response =
client.execute(buildGetRequest(API_URL));

			HttpEntity entity = response.getEntity();
			String responseString = EntityUtils.toString(entity, "UTF-
8");

			System.out.println("");
			System.out.println("Response: ");
			System.out.println(responseString);

			int status = response.getStatusLine().getStatusCode();
			System.out.println("response status: " + status);

		} catch (Exception e) {
			e.printStackTrace();
		}
		finally {
			if (client != null) {
				try {
					client.close();
				} catch (IOException e) {
					e.printStackTrace();
				}
			}
		}
	}

	/**
	 * Builds the get request.
	 * @param url the URL
	 * @return the HTTP get
	 * @throws Exception the exception
	 */
	private HttpGet buildGetRequest(String url) throws Exception {

		URI path = new URI(url);
		HttpGet request = new HttpGet(path);
```

```
            // create signature: method + content md5 + content-type + date +
uri
            String method = "GET";
            String content = "";
            String contentType = "";
            String date = "Thu, 13 Aug 2015 15:51:01 GMT";
            String uriPath = "/payments-api/rest/v4.3/pricing";

          StringBuilder signature = new StringBuilder();
            signature.append(method).append("\n")
                .append(content).append("\n")
                .append(contentType).append("\n")
                .append(date).append("\n")
                .append(uriPath);

            System.out.println("Signature content:");
            System.out.println(signature.toString());

            String auth = "Hmac " + MERCHANT_PUBLIC_KEY + ":" +
buildHmacSignature(MERCHANT_API_KEY, signature.toString());

            System.out.println("");
            System.out.println("Authorization Header:");
            System.out.println(auth);

            request.addHeader(new BasicHeader("Date", date));
            request.addHeader(new BasicHeader("Authorization", auth));
            request.addHeader(new BasicHeader("Accept",
"application/xml"));//aplication/json

            return request;
      }

      /**
       * Builds the HMAC signature.
       * @param secret the secret
       * @param data the data
       * @return the string
       */
      private String buildHmacSignature(String secret, String data) {

            try {
                  SecretKeySpec signingKey = new
SecretKeySpec(secret.getBytes(), "HmacSHA256");
                  Mac mac = Mac.getInstance("HmacSHA256");
                  mac.init(signingKey);
                  byte[] rawHmac = mac.doFinal(data.getBytes());

                  // Encode with Base64
                  String result = new String(Base64.encodeBase64(rawHmac));
                  return result;
```

```
            }
            catch (GeneralSecurityException e) {
                  throw new IllegalArgumentException();
            }
      }

}
```