

Unblock Signature Generation

```
// Requiere CryptoJS para generar HMAC
const crypto = require('crypto-js');

// Tu clave secreta
// const secretKey = "your-secret-key";
const secretKey = pm.environment.get('secret-key');
console.log('secret-key:', secretKey);

// Obtener la URL de la petición
let url = pm.request.url.getPath();
const queryParams = pm.request.url.query.toString();
if (queryParams) {
    url += '?' + queryParams; // Concatenar los parámetros a la URL
}
console.log('url:', url);

// Obtener el body si existe (asumimos que el body está en formato JSON)
let body = pm.request.body && pm.request.body.raw ? pm.request.body.raw : "";
let jsonString = '';
if (body) {
    try {
        console.log('body:', body);
        // Asegurarte de que el body esté en formato string
        body = JSON.stringify(JSON.parse(body));
        // body = JSON.stringify(body);
        // jsonString = JSON.stringify(body).replace(/\s+/g, "").trim();
        // jsonString =
`{"phone":"+123456789","isoCodePhone":"US","name":"John","lastName":"Doe","email":"john.doe@example.com","isoCodeCountryResidence":"US","address":"1234 Main St","dateOfBirth":"1990-02-15","zipCode":"33140"}`;
    } catch (error) {
        console.log("Error parsing body to JSON string:", error);
        // Manejar el error si el JSON no es válido
    }
}
console.log('jsonString:', body);

// Concatenar URL + Body
const dataToSign = url + body;
```

```
console.log('dataToSign:', dataToSign);
// Generar el HMAC utilizando SHA256
const signature = crypto.HmacSHA256(dataToSign, secretKey).toString(crypto.enc.Hex);

// Agregar el `signature` como un header
pm.request.headers.add({
  key: 'signature',
  value: signature
});

// Log para depuración
console.log('Signature:', signature);

//*****Setear header de
idempotency***** */

var uuid = require('uuid');
const uuid1 = uuid.v4();

pm.request.headers.add({ key: "X-Idempotency-Key", value: uuid1 });
```

Revision #3

Created 19 June 2026 14:15:49 by Krzysztof Wereski

Updated 19 June 2026 14:18:34 by Krzysztof Wereski