**REST API – Postman HMAC Pre-request script examples**

***GET***

// Set environment variables

var clientId = "{your\_client\_id}";

var clientSecret = "{your\_client\_secret}";

var resource = "/rest/v1/get-brands";

var host = "api.price2spy.com:443";

var method = "GET";

// Create a timestamp

var timestamp = Math.floor(new Date().getTime() / 1000).toString();

// Create the string to sign

var payload = ""; // mandatory even if there is no body

var stringToSign = method + "\n" + host + "\n" + resource + "\n" + timestamp + "\n" + payload;

// Generate HMAC signature

var signature = CryptoJS.HmacSHA256(stringToSign, clientSecret);

var base64signature = CryptoJS.enc.Base64.stringify(signature);

// Set headers

pm.request.headers.add({

    key: "X-P2S-Date",

    value: timestamp

});

pm.request.headers.add({

    key: "Authorization",

    value: "HmacSHA256 " + clientId + ":" + base64signature

});

***POST***

// Set environment variables

var clientId = "{your\_client\_id}";

var clientSecret = "{your\_client\_secret}";

var resource = "/rest/v1/insert-brand";

var host = "api.price2spy.com:443";

var method = "POST";

var content\_type = "application/json"

// Create a timestamp

var timestamp = Math.floor(new Date().getTime() / 1000).toString();

// Define the request payload

var payload = {

  "active": true,

  "emailRecipients": "test@oaza.rs",

  "name": "Test",

  "priority": 0

};

// Convert the payload to a string to include in the string to sign

var payload = JSON.stringify(payload);

// Create the string to sign

var stringToSign = method + "\n" + host + "\n" + content\_type + "\n" + resource + "\n" + timestamp + "\n" + payload;

// Generate HMAC signature

var signature = CryptoJS.HmacSHA256(stringToSign, clientSecret);

var base64signature = CryptoJS.enc.Base64.stringify(signature);

// Set headers

pm.request.headers.add({

    key: "X-P2S-Date",

    value: timestamp

});

pm.request.headers.add({

    key: "Authorization",

    value: "HmacSHA256 " + clientId + ":" + base64signature

});

// Set the body

pm.request.body = {

    mode: 'raw',

    raw: payload,

    options: {

        raw: {

            language: 'json'

        }

    }

};

***PUT***

// Set environment variables

var clientId = "{your\_client\_id}";

var clientSecret = "{your\_client\_secret}";

var resource = "/rest/v1/update-brand";

var host = "api.price2spy.com:443";

var method = "PUT";

var content\_type = "application/json"

// Create a timestamp

var timestamp = Math.floor(new Date().getTime() / 1000).toString();

// Define the request payload

var payload = {

  "active": true,

  "emailRecipients": "test@oaza.rs",

  "id": 473898,

  "name": "API Update brand",

  "priority": 0

};

// Convert the payload to a string to include in the string to sign

var payload = JSON.stringify(payload);

// Create the string to sign

var stringToSign = method + "\n" + host + "\n" + content\_type + "\n" + resource + "\n" + timestamp + "\n" + payload;

// Generate HMAC signature

var signature = CryptoJS.HmacSHA256(stringToSign, clientSecret);

var base64signature = CryptoJS.enc.Base64.stringify(signature);

// Set headers

pm.request.headers.add({

    key: "X-P2S-Date",

    value: timestamp

});

pm.request.headers.add({

    key: "Authorization",

    value: "HmacSHA256 " + clientId + ":" + base64signature

});

// Set the body

pm.request.body = {

    mode: 'raw',

    raw: payload,

    options: {

        raw: {

            language: 'json'

        }

    }

};

***DELETE WITH BODY***

// Set environment variables

var clientId = "{your\_client\_id}";

var clientSecret = "{your\_client\_secret}";

var resource = "/rest/v1/delete-brand";

var host = "api.price2spy.com:443";

var method = "DELETE";

var content\_type = "application/json"

// Create a timestamp

var timestamp = Math.floor(new Date().getTime() / 1000).toString();

// Define the request payload

var payload = {

  "id": 473898

};

// Convert the payload to a string to include in the string to sign

var payload = JSON.stringify(payload);

// Create the string to sign

var stringToSign = method + "\n" + host + "\n" + content\_type + "\n" + resource + "\n" + timestamp + "\n" + payload;

// Generate HMAC signature

var signature = CryptoJS.HmacSHA256(stringToSign, clientSecret);

var base64signature = CryptoJS.enc.Base64.stringify(signature);

// Set headers

pm.request.headers.add({

    key: "X-P2S-Date",

    value: timestamp

});

pm.request.headers.add({

    key: "Authorization",

    value: "HmacSHA256 " + clientId + ":" + base64signature

});

// Set the body

pm.request.body = {

    mode: 'raw',

    raw: payload,

    options: {

        raw: {

            language: 'json'

        }

    }

};

***DELETE WOTHOUT BODY***

// Set environment variables

var clientId = "{your\_client\_id}";

var clientSecret = "{your\_client\_secret}";

var resource = "/rest/v1/delete-product/1234567";

var host = "api.price2spy.com:443";

var method = "DELETE";

// Create a timestamp

var timestamp = Math.floor(new Date().getTime() / 1000).toString();

// Create the string to sign

var payload = ""; // mandatory even if there is no body

var stringToSign = method + "\n" + host + "\n" + resource + "\n" + timestamp + "\n" + payload;

// Generate HMAC signature

var signature = CryptoJS.HmacSHA256(stringToSign, clientSecret);

var base64signature = CryptoJS.enc.Base64.stringify(signature);

// Set headers

pm.request.headers.add({

    key: "X-P2S-Date",

    value: timestamp

});

pm.request.headers.add({

    key: "Authorization",

    value: "HmacSHA256 " + clientId + ":" + base64signature

});

SOAP API – **Postman HMAC Pre-request script examples**

***Pre – request script -*** same for every request

var clientId = "{your\_client\_id}";

var clientSecret = "{your\_client\_secret}";

var resource = "/soap";

var host = "api.price2spy.com:443";

var method = "POST";

var timestamp = Math.floor(new Date().getTime() / 1000).toString();

var stringToSign = clientId + "\n" + timestamp;

var signature = CryptoJS.HmacSHA256(stringToSign, clientSecret);

var base64signature = CryptoJS.enc.Base64.stringify(signature);

var clientSignature = "HmacSHA256 " + clientId + ":" + base64signature

pm.environment.set("Authorization", clientSignature);

pm.environment.set("X-P2S-Date", timestamp);

***GET***

<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/" xmlns:sear="http://www.price2spy.com/api/search">

   <soapenv:Header>

      <Authorization>{{Authorization}}

</Authorization>

      <X-P2S-Date>{{X-P2S-Date}}</X-P2S-Date>

   </soapenv:Header>

   <soapenv:Body>

      <sear:getBrandsRequest>

      </sear:getBrandsRequest>

   </soapenv:Body>

</soapenv:Envelope>

***INSERT***

<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/" xmlns:mod="http://www.price2spy.com/api/modify">

  <soapenv:Header>

      <Authorization>{{Authorization}}

</Authorization>

      <X-P2S-Date>{{X-P2S-Date}}</X-P2S-Date>

   </soapenv:Header>

   <soapenv:Body>

      <mod:insertBrandRequest>

         <mod:insert name="API test soap" active="true" priority="0" emailRecipient="test@oaza.rs"/>

      </mod:insertBrandRequest>

   </soapenv:Body>

</soapenv:Envelope>

***UPDATE***

<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/" xmlns:mod="http://www.price2spy.com/api/modify">

    <soapenv:Header>

      <Authorization>{{Authorization}}

</Authorization>

      <X-P2S-Date>{{X-P2S-Date}}</X-P2S-Date>

   </soapenv:Header>

   <soapenv:Body>

      <mod:updateBrandRequest>

         <mod:update brandId= "473898" name="update test brand" active="false" priority="10" emailRecipient="test@oaza.rs"/>

      </mod:updateBrandRequest>

   </soapenv:Body>

</soapenv:Envelope>

***DELETE***

<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/" xmlns:mod="http://www.price2spy.com/api/modify">

   <soapenv:Header>

      <Authorization>{{Authorization}}

</Authorization>

      <X-P2S-Date>{{X-P2S-Date}}</X-P2S-Date>

   </soapenv:Header>

   <soapenv:Body>

      <mod:deleteBrandRequest>

         <mod:brandId>12345678</mod:brandId>

    </mod:deleteBrandRequest>

   </soapenv:Body>

</soapenv:Envelope>