

ServiceUseRegistration REST API

1 Inledning

ServiceUseRegistration är en tjänst för att rapportera användningen av visnings- och nedladdningstjänster.


Det REST API som tjänsten har är främst tänkt att användas vid maskin-maskin kommunikation d.v.s. ett program eller skript hämtar och registrerar information. Se appendix A för ett enkelt exempel i Java för att hämta en rapport och appendix B för ett exempel på registrering.

2 Rest klient

Det finns i dagsläget inget grafiskt verktyg för tjänsterapporteringen så för att manuellt använda tjänsten kan man använda en sk. REST klient, som är ett generellt verktyg för att använda/testa REST tjänster.

Det finns många varianter men i det här dokumentet används ett tillägg till webbläsaren Chrome som heter Restlet Client.

2.1 Installation av Restlet Client

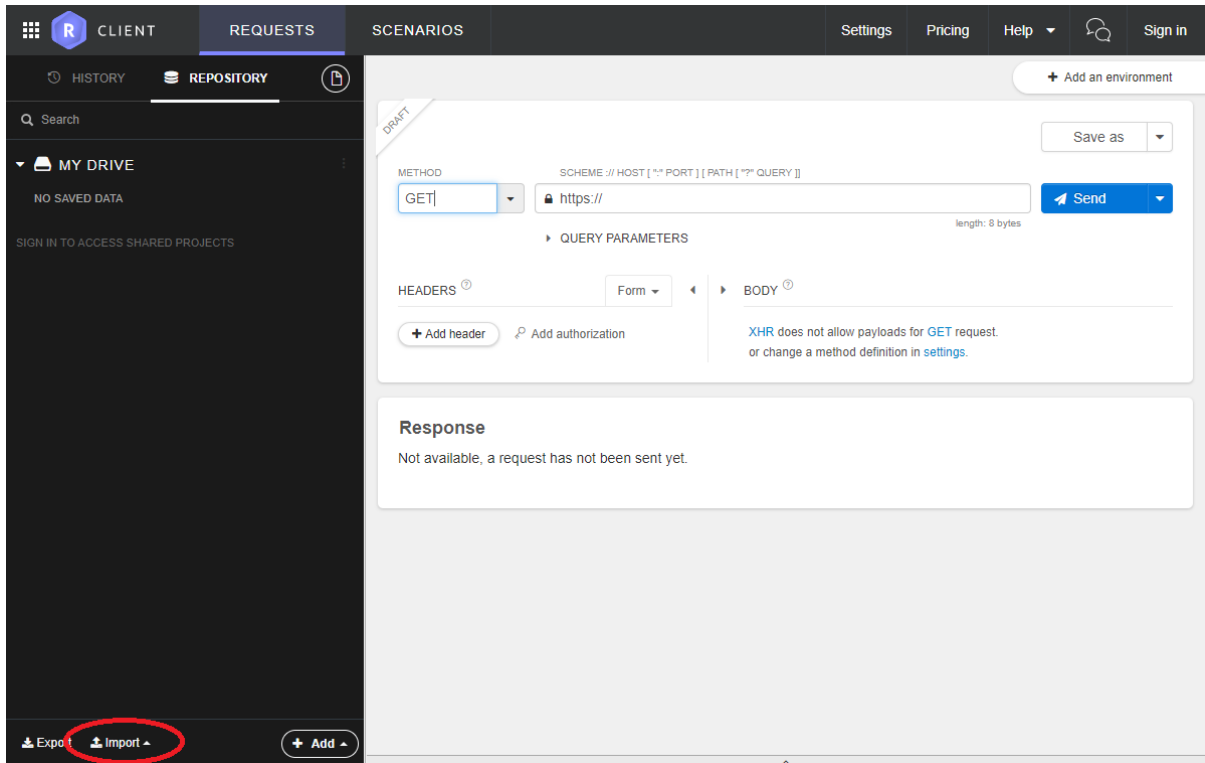
1. Gå till Chrome Web Store på adressen <https://chrome.google.com/webstore/>
2. I rutan "Sök i butiken", skriv "Restlet Client"
3. I sökresultaten, klicka på knappen "Lägg till" för tillägget
4. Klicka "Lägg till tillägg" i den bekräftelseruta som kommer upp
5. För att använda klienten, klicka på dess ikon  uppe till höger i webbläsaren

För att komma direkt till tillägget (utan att använda sökningarna beskrivet ovan) kan följande adress användas:

<https://chrome.google.com/webstore/detail/restlet-client/aejoelaoggembcahagimdiliamlcdmfm>

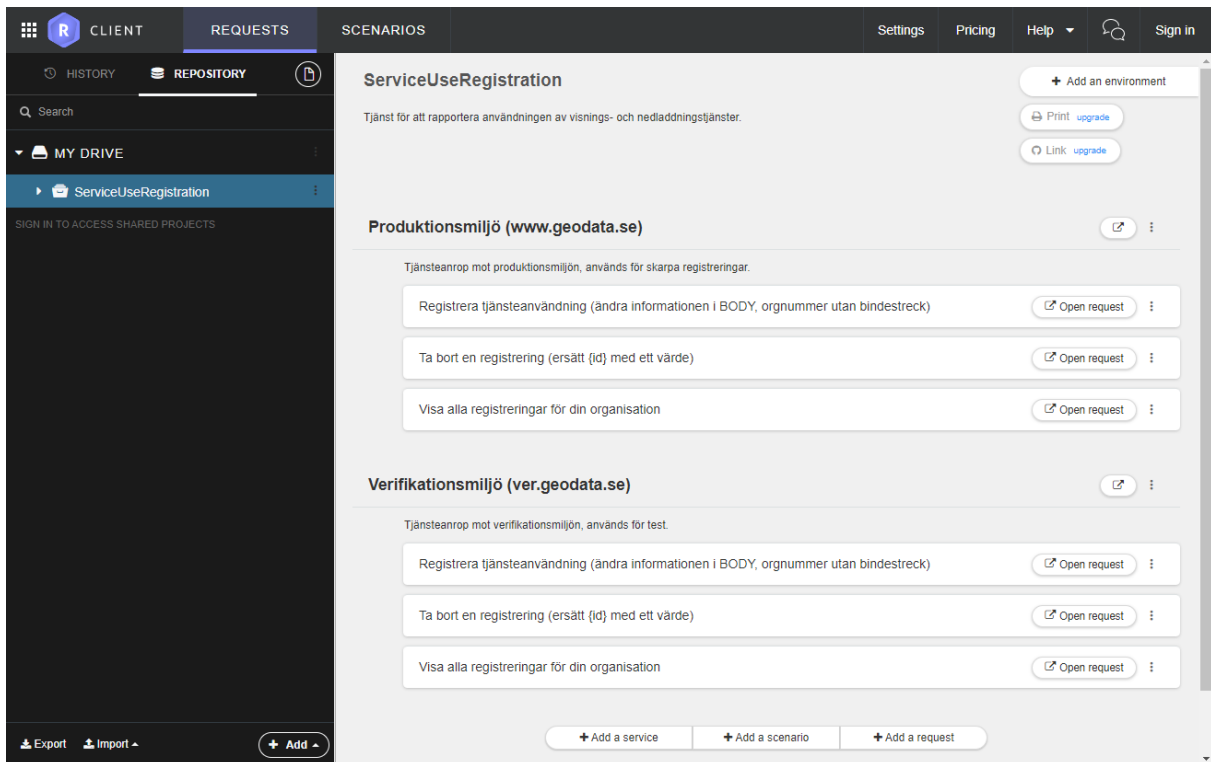
2.2 Färdiga exempel

Har du fått tillgång till de färdiga exemplen som finns kan dessa importeras till klienten.



1. Klicka på Import knappen enligt bilden ovan och välj "Import Restlet Client Repository"
2. Klicka på "Choose a file" och öppna den json-fil som innehåller exemplen
3. Klicka i rutan för "ServiceUseRegistration" och sedan "Import" knappen

Resultatet blir enligt nedan:



Exemplen visar hur man registrerar tjänsteanvändning, tar bort en registrering samt hur man kan se alla inrapporteringar. De är uppdelade i två sektioner där exemplen är samma men utförs mot två olika miljöer, en för skarpa registreringar och en för test.

2.3 Webbadress

Grundadressen till tjänsten är <https://www.geodata.se/ServiceUseRegistration> men sedan lägger man till olika kommandon.

Vill man testa tjänsten utan att påverka skarpa registreringar kan man använda följande adress <https://ver.geodata.se/ServiceUseRegistration> (ver istället för www)

Exemplen som följer använder ver-adressen men allt fungerar på samma sätt mot www.

2.4 Behörigheter

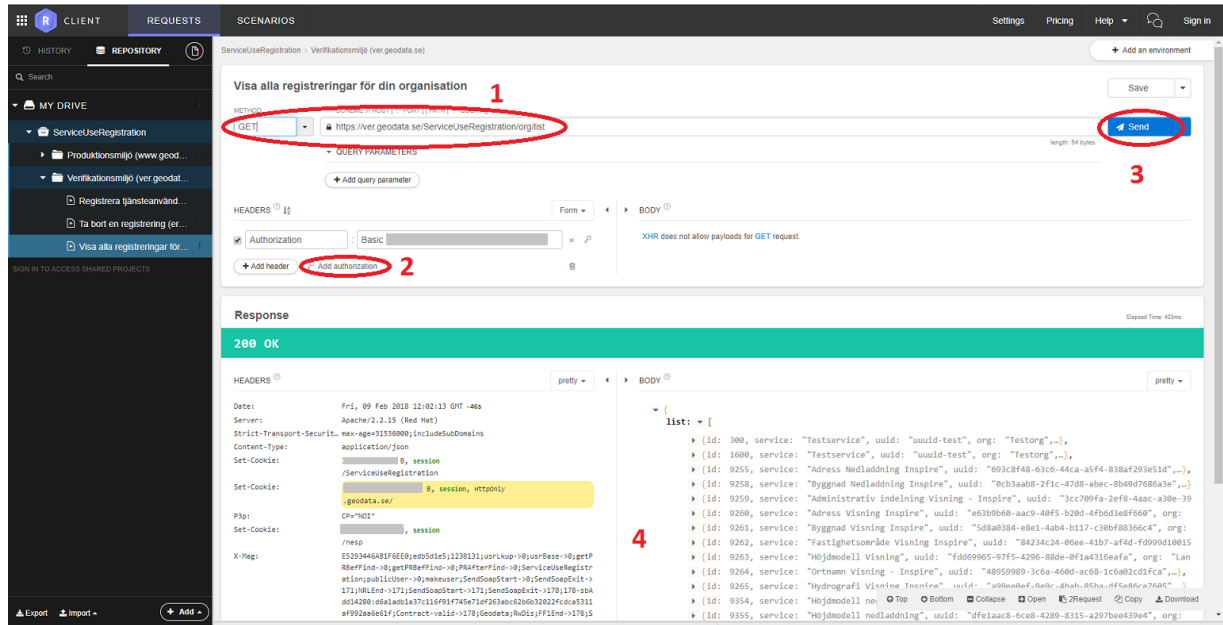
För att komma åt tjänsten behöver man vara behörig och logga in med användarnamn och lösenord.

Kontakta geodatasupport@lm.se för att få en tjänsteanvändare mot Lantmäteriets onlinetjänster (dessa har inte tidsbegränsade lösenord) och meddela sedan användarnamnet till support@geodata.se för att få rätt behörighet.

3 Exempel

3.1 Exempel 1: Visa alla registreringar

För att se alla registreringar som gjorts för din organisation:



1. Ange METHOD=GET och kommandot `org/list` (alltså adressen <https://ver.geodata.se/ServiceUseRegistration/org/list>)
2. Ange inloggningsuppgifter genom att klicka på "Add authorization" och ange användarnamn och lösenord.
3. Klicka på "Send" knappen
4. Svaret presenteras, klicka på de små pilarna för att se respektive inrapportering.

3.2 Exempel 2: Registrera tjänste användning

För att registrera tjänste användning ändrar man kommandot till `org/add` och METHOD från GET till POST. När man byter till POST visas ett textfält där man skriver in indata på ett speciellt sätt (som json). Under `HEADERS` ska det finnas, förutom inloggningsuppgifterna, ett `Content-Type` värde satt till `application/json`.

Exempel på indata:

```
{
  "service": "Testservice",
  "uuid": "uuuid-test",
  "org": "Testorg",
  "orgnummer": "0000",
  "requests": 666,
  "reporter": "Test Testsson",
  "fromDate": "2017-01-01",
  "toDate": "2017-01-02"
}
```

}

Naturligtvis ska exemplets värden ersättas med de riktiga.

OBS att organisationsnumret anges med endast siffror (inga bindestreck).

Man kan bara registrera med samma organisationsnummer som den inloggade användaren har.

Nedan visas hur Restlet klienten ser ut vid registrering av tjänsteanvändning:

The screenshot shows a REST client interface for configuring and sending a POST request. The request is to the URL `https://ver.geodata.se/ServiceUseRegistration/org/add`. The headers include `Authorization: Basic` and `Content-Type: application/json`. The body is a JSON object with fields: `service: "Testservice"`, `uid: "uuuid-test"`, `org: "Testorg"`, `orgnummer: "11055"`, `requests: 1234`, `reporter: "Test Testsson"`, `fromDate: "2018-01-01"`, and `toDate: "2018-01-02"`. The response is a `200 OK` with headers including `Date`, `Server`, `Strict-Transport-Security`, `Content-Type: application/json`, `Set-Cookie`, `P3p`, `Via`, `Keep-Alive`, `Connection`, and `Transfer-Encoding`. The response body is a JSON object with fields: `id: 11055`, `service: "Testservice"`, `uid: "uuuid-test"`, `org: "Testorg"`, `orgnummer: "2021004888"`, `requests: 1234`, `reporter: "Test Testsson"`, `fromDate: "2018-01-01"`, and `reported: "2018-02-09T12:23:50.373+0000"`.

1. Ange METHOD=POST och kommandot `org/add` (alltså adressen <https://ver.geodata.se/ServiceUseRegistration/org/add>)
2. Ange inloggningsuppgifter genom att klicka på "Add authorization" knappen och ange användarnamn samt lösenord. Se till att det finns en rad med Content-Type: application/json
3. Ange det som ska registreras (indata) enligt det speciella formatet (json)
4. Klicka på Send
5. Tjänsten svarar med samma indata kompletterat med uppgifter om id och inrapporteringstid

3.3 Exempel 3: Ta bort en inrapportering

För att ta bort inrapportering måste man känna till id:t för den. Gör en lista på inrapporteringar enligt Exempel 1 och notera siffran vid id:t för den inrapportering som ska tas bort.

Siffran anges som en del av kommandot `org/delete/{id}`. Om man vill ta bort en inrapportering med id 11055 blir alltså kommandot `org/delete/11055`.

Man kan bara ta bort inrapporteringar med samma organisationsnummer som den inloggade användaren har.

The screenshot shows a REST client interface with the following elements:

- Request Configuration:**
 - METHOD:** POST (circled in red with a '1').
 - URL:** <https://ver.geodata.se/ServiceUseRegistration/org/delete/{id}> (circled in red with a '1').
 - Authorization:** Basic (circled in red with a '2').
 - Send Button:** A blue button with a right-pointing arrow (circled in red with a '3').
- Response:**
 - Status:** 200 OK (highlighted in green).
 - Body:** A JSON object (circled in red with a '4'):

```
{
  "id": 11055,
  "service": "Testservice",
  "uuid": "uuuid-test",
  "org": "Testorg",
  "orgnummer": "2021004888",
  "requests": 1234,
  "reporter": "Test Testsson",
  "fromDate": "2018-01-01",
  "toDate": "2018-01-02",
  "reported": "2018-02-09T12:23:50.373+0000"
}
```

1. Ange METHOD=POST och kommandot `org/delete/{id}` (alltså adressen <https://ver.geodata.se/ServiceUseRegistration/org/delete/{id}>), ersätt {id} med rätt värde
2. Ange inloggningsuppgifter genom att klicka på "Add authorization" och ange användarnamn och lösenord.
3. Klicka på "Send" knappen
4. Den borttagna inrapporteringen visas i svarsfältet.

4 Appendix A

Här följer ett enkelt exempel på ett program i Java som hämtar en rapport för alla tjänster för januari 2017. Svaret visas som ett xml-dokument.

```
import java.io.IOException;
import java.io.InputStream;
import java.io.InputStreamReader;
import java.net.MalformedURLException;
import java.net.URL;
import java.net.URLConnection;

import org.apache.commons.codec.binary.Base64;

public class Example {

    private static final String USER = "<username>";
    private static final String PASS = "<password>";

    private static void getReport() throws IOException {
        InputStream is = null;
        try {
            String base = "https://ver.geodata.se/ServiceUseRegistration/";
            String service = "adm/report/xml";
            String query = "?fromDate=2017-01-01&toDate=2017-01-31";

            String request = base + service + query;

            String authString = USER + ":" + PASS;

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

            URL url = new URL(request);
            URLConnection urlConnection = url.openConnection();
            urlConnection.setRequestProperty("Authorization", "Basic " + authStringEnc);
            is = urlConnection.getInputStream();
            InputStreamReader isr = new InputStreamReader(is);

            int numCharsRead;
            char[] charArray = new char[1024];
            StringBuffer sb = new StringBuffer();
            while ((numCharsRead = isr.read(charArray)) > 0) {
                sb.append(charArray, 0, numCharsRead);
            }
            String result = sb.toString();

            System.out.println(result);

        } catch (MalformedURLException e) {
            e.printStackTrace();
        } catch (IOException e) {
            e.printStackTrace();
        } finally {
            if (is != null) {
                is.close();
            }
        }
    }

    public static void main(String[] args) throws IOException{
        getReport();
    }
}
```

5 Appendix B

Här visas exempel på registrering av tjänsteanvändning:

```
import java.io.DataOutputStream;
import java.io.IOException;
import java.io.InputStream;
import java.io.InputStreamReader;
import java.net.HttpURLConnection;
import java.net.MalformedURLException;
import java.net.URL;

import org.apache.commons.codec.binary.Base64;

public class Example
{
    private static final String USER = "<username>";
    private static final String PASS = "<password>";

    private static void register() throws IOException
    {
        InputStream is = null;
        try
        {
            String base = "https://ver.geodata.se/ServiceUseRegistration/";
            String path = "org/add";

            String request = base + path;

            String authString = USER + ":" + PASS;

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

            URL url = new URL(request);
            HttpURLConnection urlConnection = (HttpURLConnection) url.openConnection();
            urlConnection.setRequestMethod("POST");
            urlConnection.setRequestProperty("Content-Type", "application/json");
            urlConnection.setRequestProperty("Authorization", "Basic " + authStringEnc);
            urlConnection.setDoOutput(true);

            String data = "{" +
                "\"service\": \"Testservice\", " +
                "\"uid\": \"uuid-test\", " +
                "\"org\": \"Testorg\", " +
                "\"orgnummer\": \"0000\", " +
                "\"requests\": 666, " +
                "\"reporter\": \"Test Testsson\", " +
                "\"fromDate\": \"2017-02-01\", " +
                "\"toDate\": \"2017-02-02\" " +
                "}";
            DataOutputStream wr = new DataOutputStream(urlConnection.getOutputStream());
            wr.writeBytes(data);
            wr.flush();
            wr.close();

            is = urlConnection.getInputStream();
            InputStreamReader isr = new InputStreamReader(is);

            int numCharsRead;
            char[] charArray = new char[1024];
            StringBuffer sb = new StringBuffer();
            while ((numCharsRead = isr.read(charArray)) > 0)
            {
                sb.append(charArray, 0, numCharsRead);
            }
            String result = sb.toString();

            System.out.println(result);
        }
        catch (MalformedURLException e)
    }
}
```



```
{
    e.printStackTrace();
}
catch (IOException e)
{
    e.printStackTrace();
}
finally
{
    if (is != null)
    {
        is.close();
    }
}
}

public static void main(String[] args) throws IOException
{
    register();
}
}
```