#### 4. Jersey client

**1. Configure POM.xml with**

To use Jersey client APIs, declares **“jersey-client.jar”** in your pom.xml file.

<dependency>

<groupId>com.sun.jersey</groupId>

<artifactId>jersey-client</artifactId>

<version>1.8</version>

</dependency>

**2.GET Request Example**

package rest.clients;

import com.sun.jersey.api.client.Client;

import com.sun.jersey.api.client.ClientResponse;

import com.sun.jersey.api.client.WebResource;

public class JerseyClientGet {

public static void main(String[] args) {

try {

Client client = Client.*create*();

WebResource webResource = client

.resource("http://localhost:8080/JAXRS-JSON-JavaClients-Example/rest/json/getjson");

ClientResponse response = webResource.accept("application/json").get(ClientResponse.class);

if (response.getStatus() != 200) {

throw new RuntimeException("Failed : HTTP error code : " + response.getStatus());

}

String output = response.getEntity(String.class);

System.*out*.println("Output from Server .... \n");

System.*out*.println(output);

} catch (Exception e) {

e.printStackTrace();

}

}

}

Output

Output from Server ....

{"username":"satyakaveti@gmail.com","password":"XCersxg34CXeWER341DS@#we"}

**3.POST Request Example**

package rest.clients;

import com.sun.jersey.api.client.Client;

import com.sun.jersey.api.client.ClientResponse;

import com.sun.jersey.api.client.WebResource;

public class JerseyClientPost {

public static void main(String[] args) {

try {

Client client = Client.*create*();

WebResource webResource = client

.resource("http://localhost:8080/JAXRS-JSON-JavaClients-Example/rest/json/postjson");

String input = "{\"username\":\"satyakaveti@gmail.com\",\"password\":\"XCersxg34CXeWER341DS@#we\"}";

ClientResponse response = webResource.type("application/json").post(ClientResponse.class, input);

if (response.getStatus() != 201) {

throw new RuntimeException("Failed : HTTP error code : " + response.getStatus());

}

System.*out*.println("Output from Server .... \n");

String output = response.getEntity(String.class);

System.*out*.println(output);

} catch (Exception e) {

e.printStackTrace();

}

}

}

Output

Output from Server ....

USER DATA SAVED!! USER [username=satyakaveti@gmail.com, password=XCersxg34CXeWER341DS@#we]

# 27.How to Test (JAX-RS) RESTful Web Services

in real time projects we will use different tools to test RESTful web services

**1.Browser Addons**

* Postman [ Chrome Extension ]
* REST Client [ Chrome Extension ]
* Advanced REST Client [ Chrome Extension ]
* Rest Client [ Firefox Add-On ]

**2.JAX-RS Local System Tools**

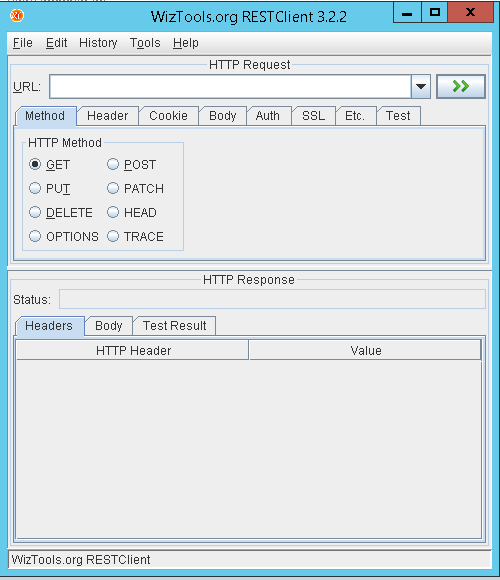
* [RESTClient UI](https://code.google.com/archive/p/rest-client/downloads)
* [SoupUi](https://www.soapui.org/)

RESTClient UI

1. Download .jar file from here <https://code.google.com/archive/p/rest-client/downloads>

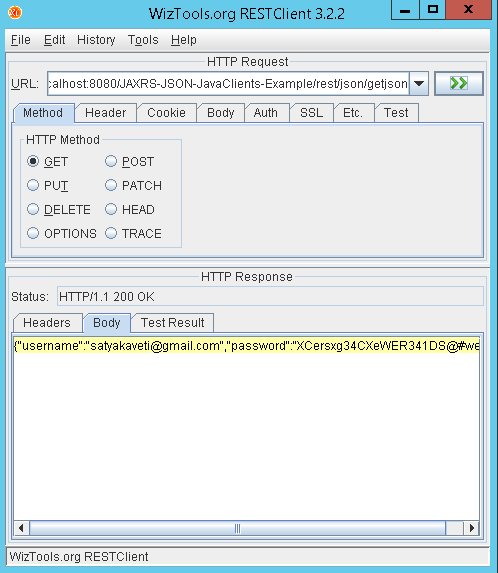
2.Run jar by giving >java -jar restclient-ui-3.2.2-jar-with-dependencies.jar

3.It will Opens the window as follows

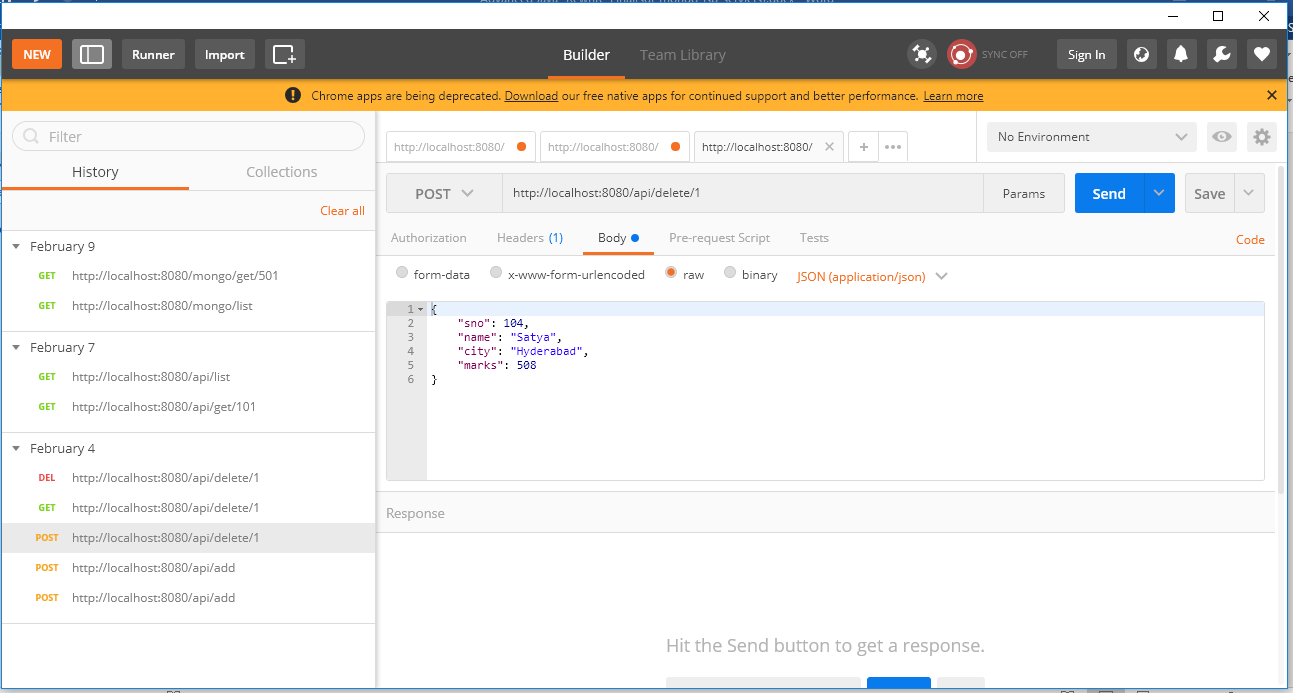


4.Test Your application by proving any running web service URL

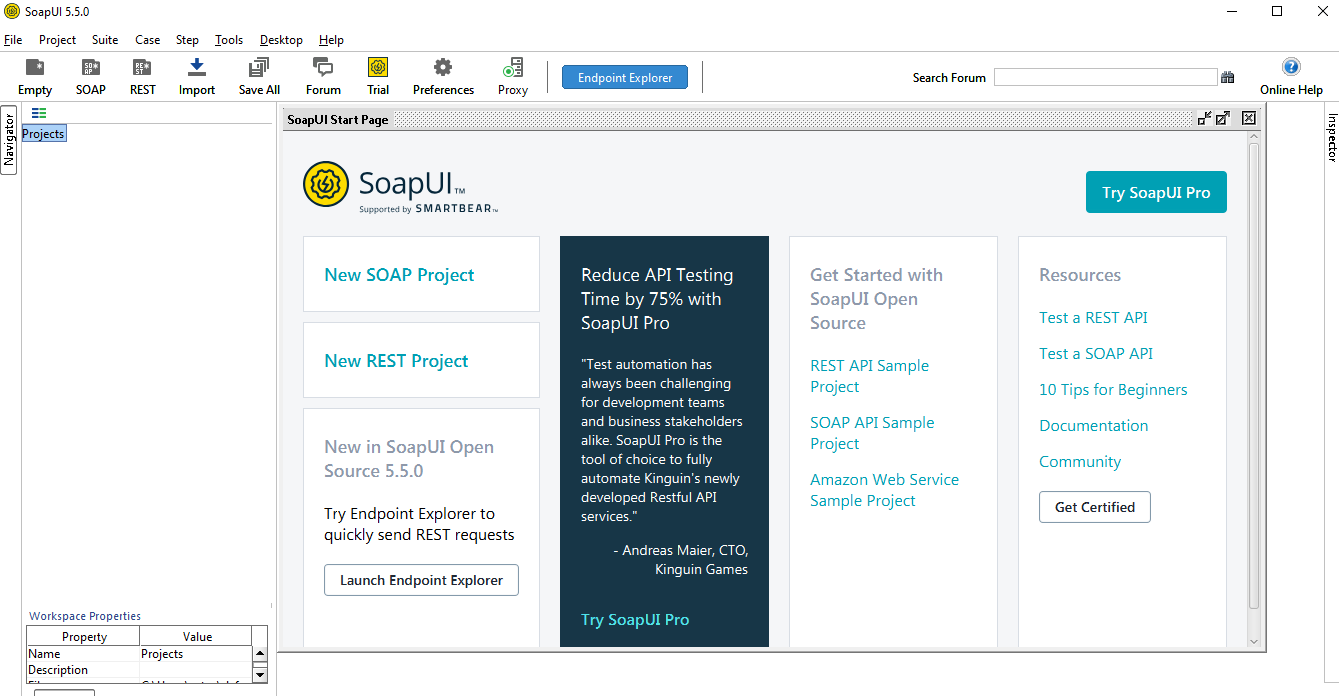
Ex: <http://localhost:8080/JAXRS-JSON-JavaClients-Example/rest/json/getjson>



Postman



Similarly we can work with SoapUI also



# References

JAX-WS

<http://www.java2blog.com/2013/03/soap-web-service-tutorial.html>

[https://examples.smlcodes.com/enterprise-java/jws/jax-ws-annotations-example/](https://examples.javacodegeeks.com/enterprise-java/jws/jax-ws-annotations-example/)

<http://cxf.apache.org/docs/developing-a-service.html>

<https://docs.oracle.com/cd/E13222_01/wls/docs92/webserv/annotations.html>

<http://docs.oracle.com/javaee/5/api/javax/jws/WebService.html>

<https://jax-ws.java.net/jax-ws-ea3/docs/annotations.html>

JAX-RS

<http://www.java4s.com/web-services/restful-web-services-jax-rs-annotations/>

<http://www.mkyong.com/tutorials/jax-rs-tutorials/>

<http://www.mkyong.com/tutorials/jax-rs-tutorials/>

<http://www.java4s.com/web-services/>