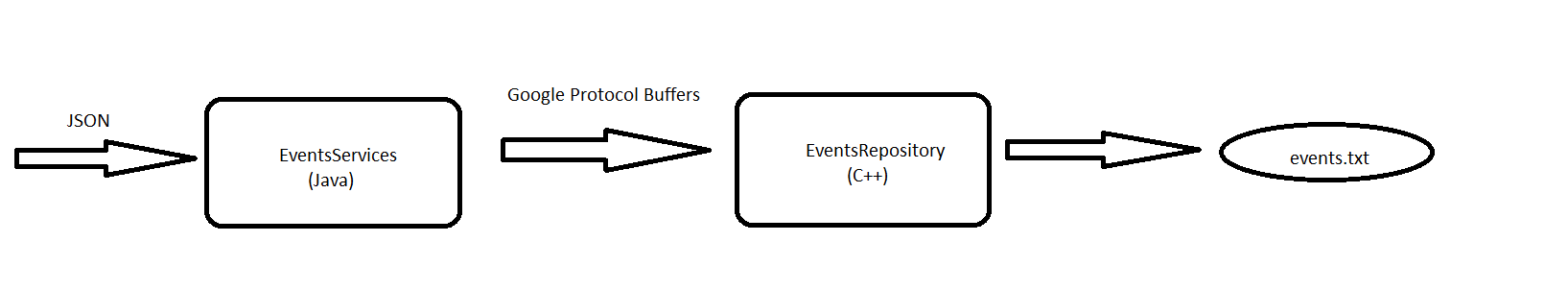
**Events Services**

1. **High Level Architecture**



1. **Description**

EventsServices is a RESTful web service written in Java using Spring Boot and Spring MVC.

EventsRepository is a server written in C++. TCP/IP is the protocol used for the communication between services using Sockets.

1. **EventsRepository(C++)**

This server is created using sockets. It has a listener for the requests coming from the EventsServices and it uses Google Protocol Buffers for serializing and deserializing. The DTO object stored in request is described in the events.proto file as the Event message. When a request is coming, the server tries to deserialize the event object. After that, it will get all the list of events stored in the “events.txt” file(as Events message described in the events.proto), and will append the new event to that list. The new list is then stored in the “events.txt” file.

To read all the events stored in that file, you have to run the service using the “print\_events” as the first argument. In this case, it will print all the events.

When you start the server, you have to run the service using the “start\_server” as the first argument.

1. **EventsServices(Java)**

This is a RESTful web service that has an endpoint. To access it you need to make a POST request to “/events/add”. The content-type is application/JSON and the body structure should be like in the following example:

{

"event": "event test",

"timestamp": "1533081600",

"userId": "123"

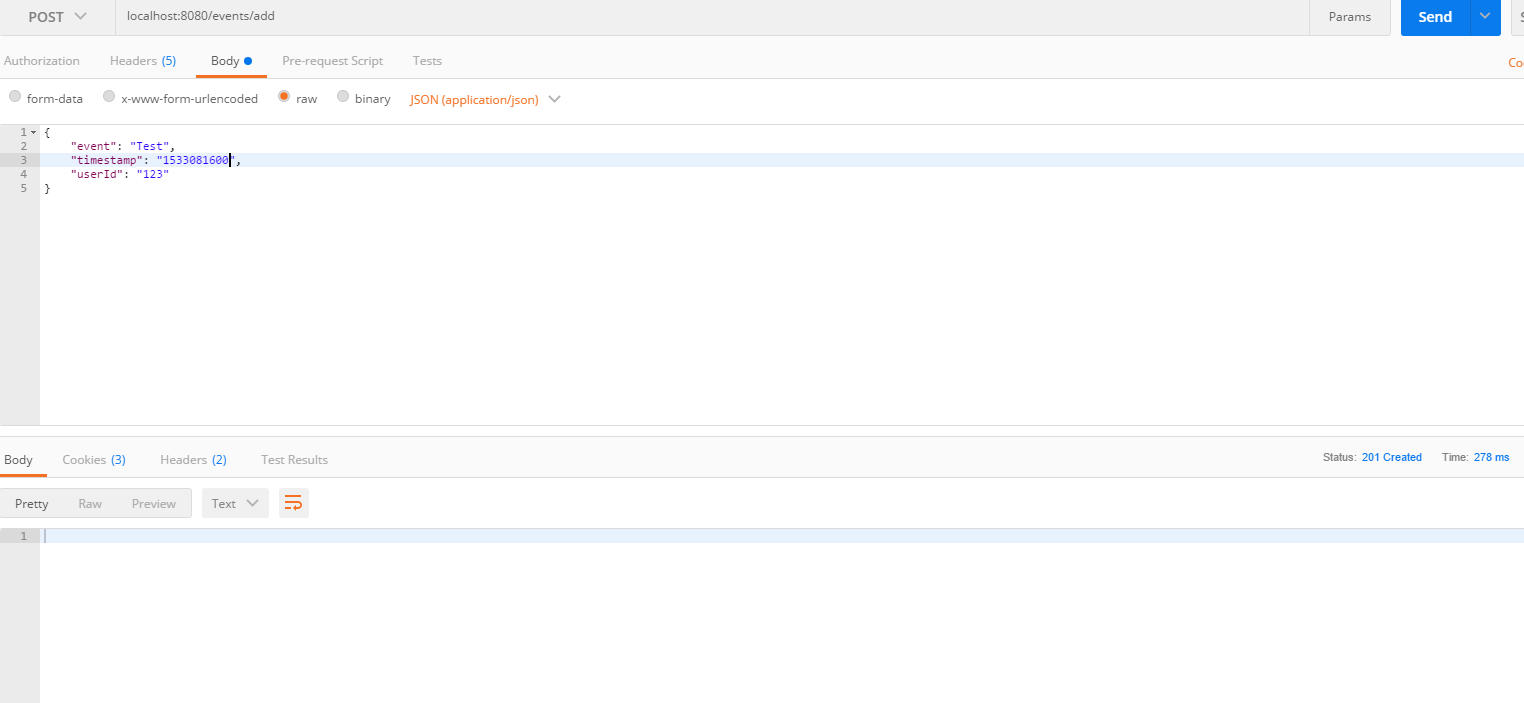
}

You can find the documentation of the code here: <https://github.com/laurbogdan/eventsServices/tree/master/Documentation/eventsServiceJavaDoc>

You need to download that folder and access index.html.

1. **Demo**

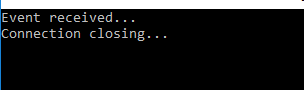
A POST request to the EventsServices written in Java. The body contains all the event information.



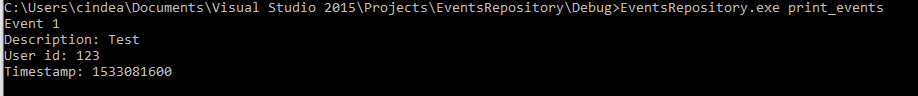
Event request in the EventsServices web service.



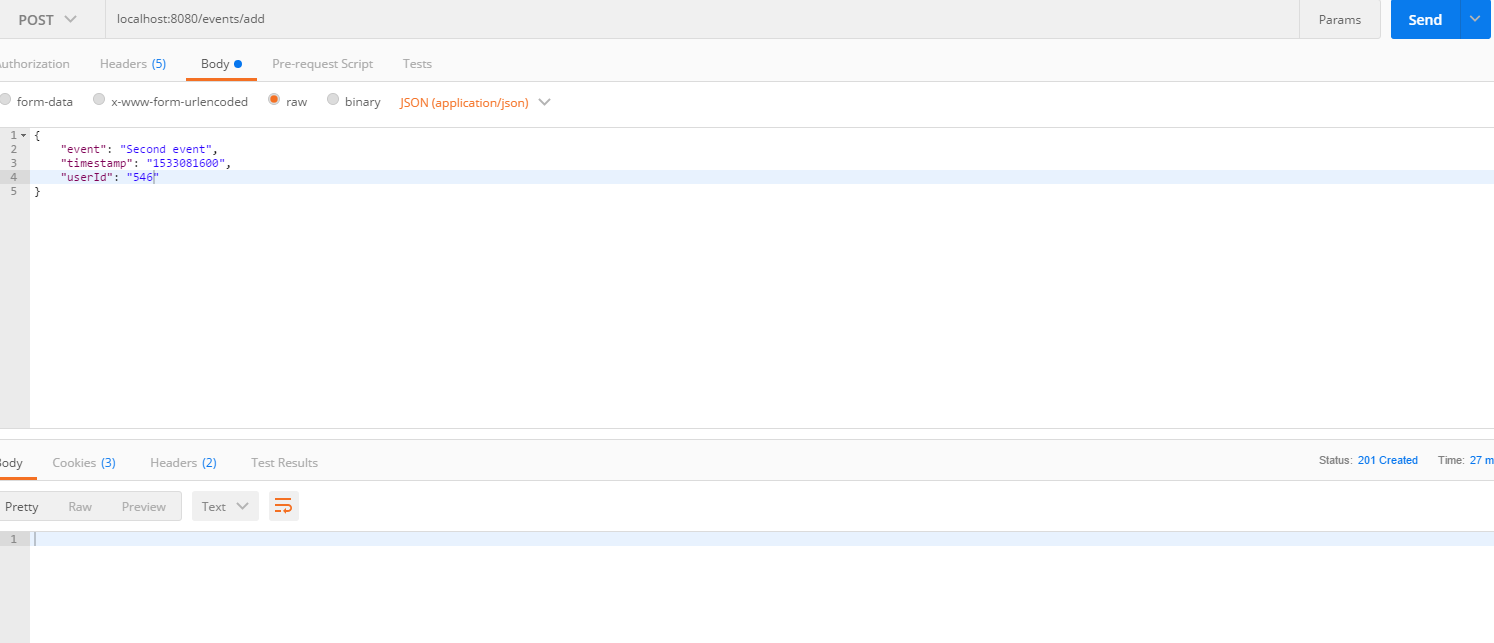
Request received in the EventsRepository service:



And now print the stored events:



Add new event:



The new event has been added:

