# 

# **Data Engineering - Test Task**

## Overview

We need to write a system that processes real-time events sent by **8ballpool game server** and, on top of that, creates real-time data quality transformations and aggregations.

The data that reaches the reports should be already cleaned and aggregated.

The goal of this task is to write such a system that does all this pipeline.

The events sent by the game server are the following:

* Init
  + Triggered when the player opens the @8ballpool app;
  + This is the first event of all. All other events should only be sent after this.
* Match
  + Triggered when two players play a match against each other;
  + Sent only at the end of the match.
* InAppPurchase
  + Triggered when a player does a purchase.

The schema of the events should be according to the schema sent in the attached folder.

## Details

This task should be **self-contained**, consider one of the following **suggestions**:

* Install in your local machine and run from there;
* Use Docker Compose with pre-built docker images;
* Create a Virtual machine, install and run from there;
* You can also provide your own option to self-contain the system.

All the code should also be commented and available.

You can choose the programming language you are more comfortable with. We suggest java, scala or python.

What is not specified, we let the developer decide the best approach. **In case of any doubt please contact us, we will be always free to help.**

## Tasks

### Kafka

* Install Kafka (see [Details](#_tbzw1xclflgr))
  + <https://kafka.apache.org/documentation/>
  + <https://kafka.apache.org/documentation/#quickstart>
  + <https://hub.docker.com/search?q=kafka>
* Create real-time match, init and in-app events and send them to a topic.
  + You can let kafka create automatically the topic, the defaults are enough for this challenge

### Spark – Batch aggregator

* Install Spark (see [Details](#_tbzw1xclflgr))
  + If you use java or scala, you can use the embedded runtime. It’s enough for this task.
  + <https://spark.apache.org/docs/latest/>
  + <https://spark.apache.org/docs/latest/quick-start.html>
* Create the following Spark aggregation scripts:
  + Daily Aggregated data. We want to have the number of distinct users during each day by country and platform.
* The component should have a set of rules to change each column that needs Data Quality transformations. Transformations that are required are:
  + Uppercase a field, e.g. platform.
  + Map a field ID to a name, e.g country\_id to country\_name.