Session 23: More Kafka Assignment 1

Task 1:

Create a java program MyKafkaProducer.java that takes a file name and delimiter as input arguments.

It should read the content of file line by line. Fields in the file are in following order

1. Kafka Topic Name
2. Key
3. value

For every line, insert the key and value to the respective Kafka broker in a fire and forget mode.

After record is sent, it should print appropriate message on screen.
Pass dataset_producer.txt as the input file and -as delimiter.
LINK: https://drive.google.com/file/d/0B_Qjau8wv1KoSnR5eHpKOF9rTFU/view?usp=sharing

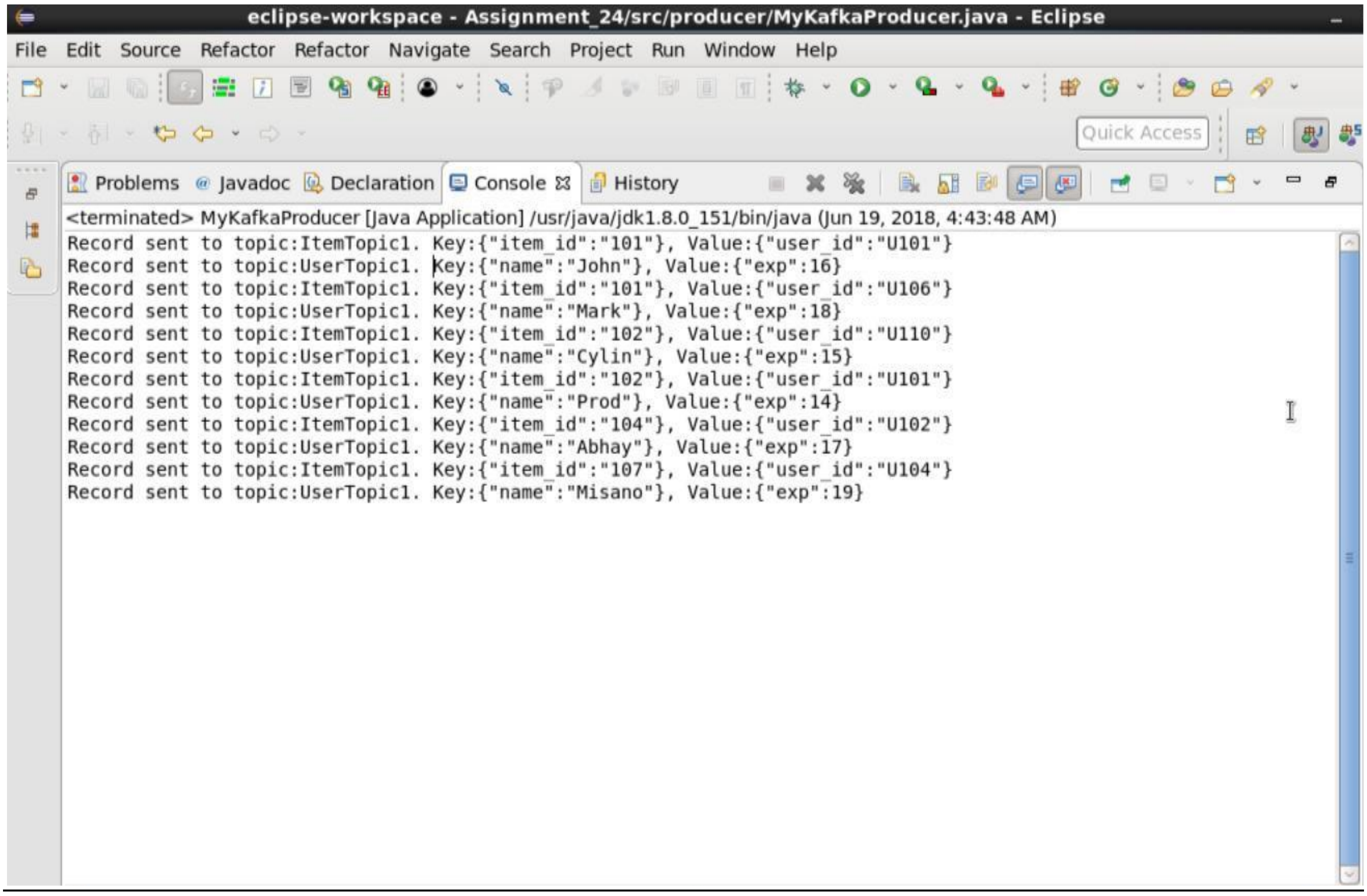
Solution:

Java Code

```
package producer;
import java.io.IOException;
import java.util.Properties;

import org.apache.kafka.clients.producer.KafkaProducer;
import org.apache.kafka.clients.producer.ProducerRecord;

public class MyKafkaProducer {
       public static void main(String[] args) throws IOException{
             Properties props = new Properties(); props.put("bootstrap.servers",
"localhost:9092");
             props.put("key.serializer",
"org.apache.kafka.common.serialization.StringSerializer");
props.put("value.serializer",
"org.apache.kafka.common.serialization.StringSerializer");
             KafkaProducer<String, String> producer = new KafkaProducer<>(props);
ProducerRecord<String, String> producerRecord = null;
             String fileName = “/home/acadgild/dataset_producer.txt”;
             String delimiter = “-“;
             try(BufferedReader br = new BufferedReader(new
FileReader(fileName))) { for(String line; (line = br.readLine()) != null; ) {
             String[] tempArray = line.split(delimiter);
             String topic = tempArray[0];
             String key = tempArray[1];
             String value = tempArray[2];
             producerRecord = new ProducerRecord<String, String>(topic, key,
value); producer.send(producerRecord);
             System.out.printf("Record sent to topic:%s. Key:%s, Value:%s\n",
topic, key, value);
             }
             }
             producer.close();
             }
```

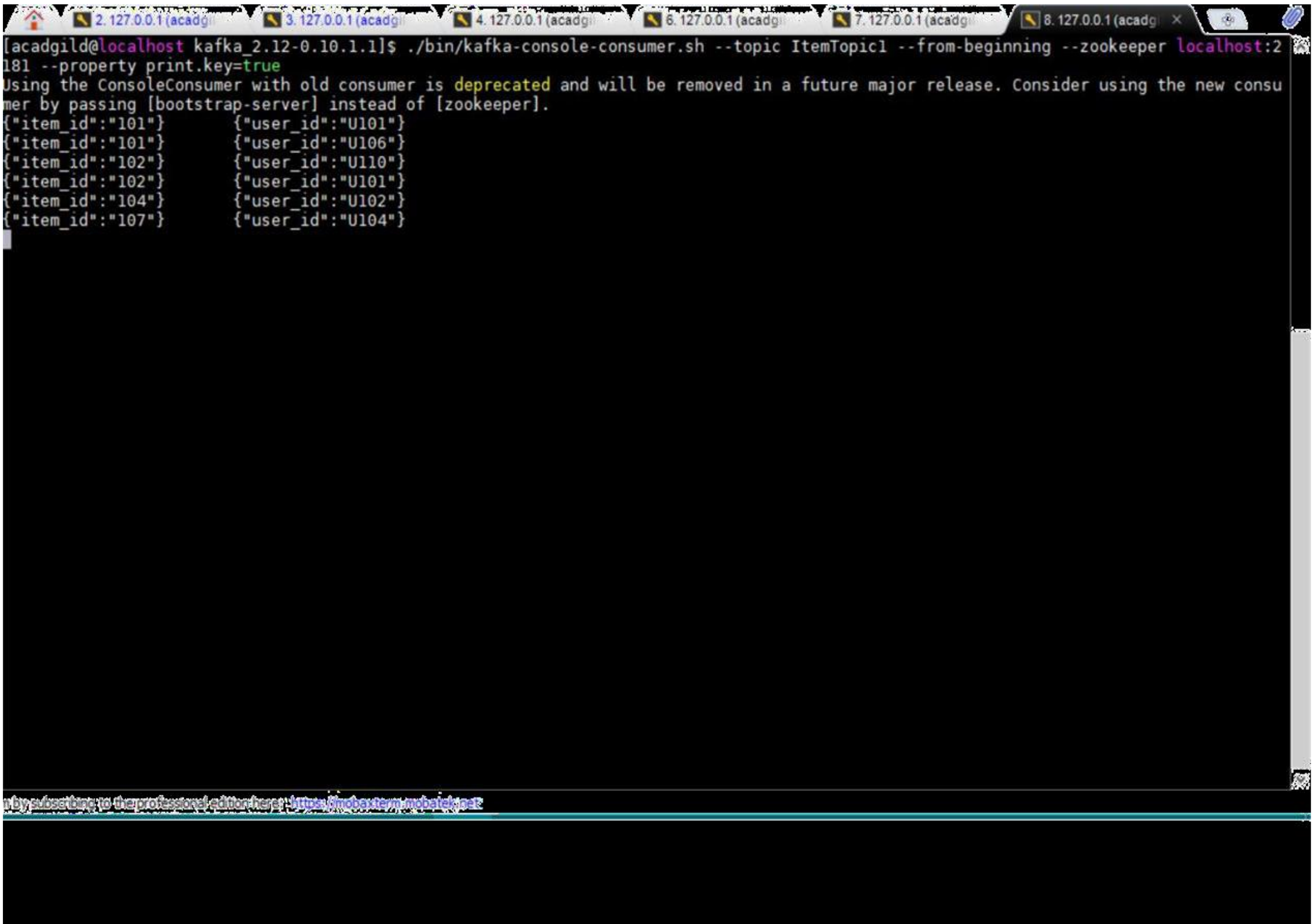
}

Output:



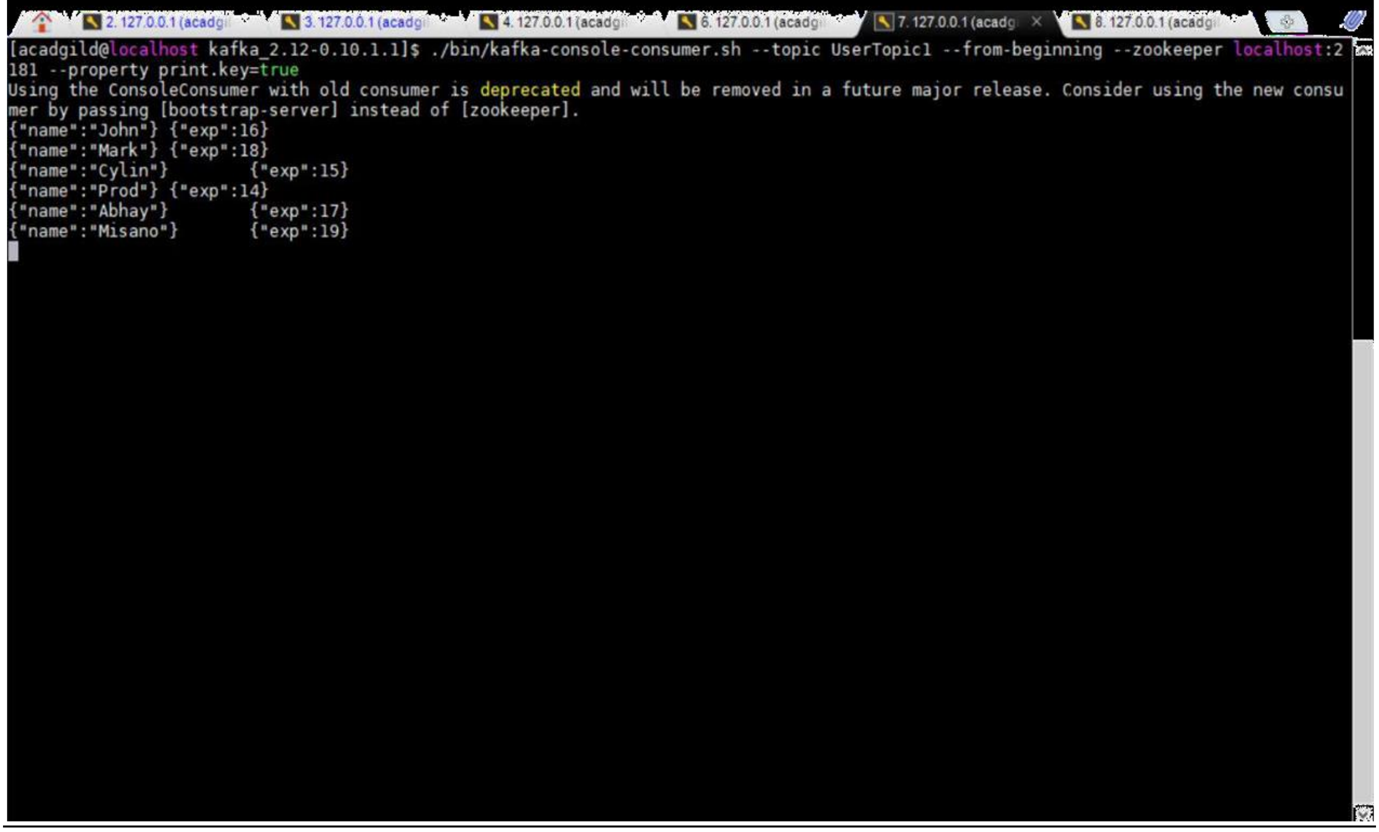
eclipse-workspace - Assignment_24/src/producer/MyKafkaProducer.java - Eclipse
File Edit Source Refactor Refactor Navigate Search Project Run Window Help
Quick Access
Problems
Javadoc
Declaration
Console
History

```
<terminated> MyKafkaProducer [Java Application] /usr/java/jdk1.8.0_151/bin/java (Jun 19, 2018, 4:43:48 AM)
Record sent to topic:ItemTopic1. Key:{"item_id":"101"}, Value:{"user_id":"U101"}
Record sent to topic:UserTopic1. Key:{"name":"John"}, Value:{"exp":16}
Record sent to topic:ItemTopic1. Key:{"item_id":"101"}, Value:{"user_id":"U106"}
Record sent to topic:UserTopic1. Key:{"name":"Mark"}, Value:{"exp":18}
Record sent to topic:ItemTopic1. Key:{"item_id":"102"}, Value:{"user_id":"U110"}
Record sent to topic:UserTopic1. Key:{"name":"Cylin"}, Value:{"exp":15}
Record sent to topic:ItemTopic1. Key:{"item_id":"102"}, Value:{"user_id":"U101"}
Record sent to topic:UserTopic1. Key:{"name":"Prod"}, Value:{"exp":14}
Record sent to topic:ItemTopic1. Key:{"item_id":"104"}, Value:{"user_id":"U102"}
Record sent to topic:UserTopic1. Key:{"name":"Abhay"}, Value:{"exp":17}
Record sent to topic:ItemTopic1. Key:{"item_id":"107"}, Value:{"user_id":"U104"}
Record sent to topic:UserTopic1. Key:{"name":"Misano"}, Value:{"exp":19}
```



```
[acadgild@localhost kafka_2.12-0.10.1.1]$ ./bin/kafka-console-consumer.sh --topic ItemTopic1 --from-beginning --zookeeper localhost:2
181 --property print.key=true
Using the ConsoleConsumer with old consumer is deprecated and will be removed in a future major release. Consider using the new consu
mer by passing [bootstrap-server] instead of [zookeeper].
{"item_id":"101"}       {"user_id":"U101"}
{"item_id":"101"}       {"user_id":"U106"}
{"item_id":"102"}       {"user_id":"U110"}
{"item_id":"102"}       {"user_id":"U101"}
{"item_id":"104"}       {"user_id":"U102"}
{"item_id":"107"}       {"user_id":"U104"}
```

n by subscribing to the professional edition here: https://mobaxterm.mobatek.net



```
[acadgild@localhost kafka_2.12-0.10.1.1]$ ./bin/kafka-console-consumer.sh --topic UserTopic1 --from-beginning --zookeeper localhost:2
181 --property print.key=true
Using the ConsoleConsumer with old consumer is deprecated and will be removed in a future major release. Consider using the new consu
mer by passing [bootstrap-server] instead of [zookeeper].
{"name":"John"} {"exp":16}
{"name":"Mark"} {"exp":18}
{"name":"Cylin"}        {"exp":15}
{"name":"Prod"} {"exp":14}
{"name":"Abhay"}        {"exp":17}
{"name":"Misano"}       {"exp":19}
```

Task 2:

Modify the previous program MyKafkaProducer.java and create a new Java program KafkaProducerWithAck.java

This should perform the same task as of KafkaProducer.java with some modification.

When passing any data to a topic, it should wait for acknowledgement.

After acknowledgement is received from the broker, it should print the key and value which has been written to a specified topic.

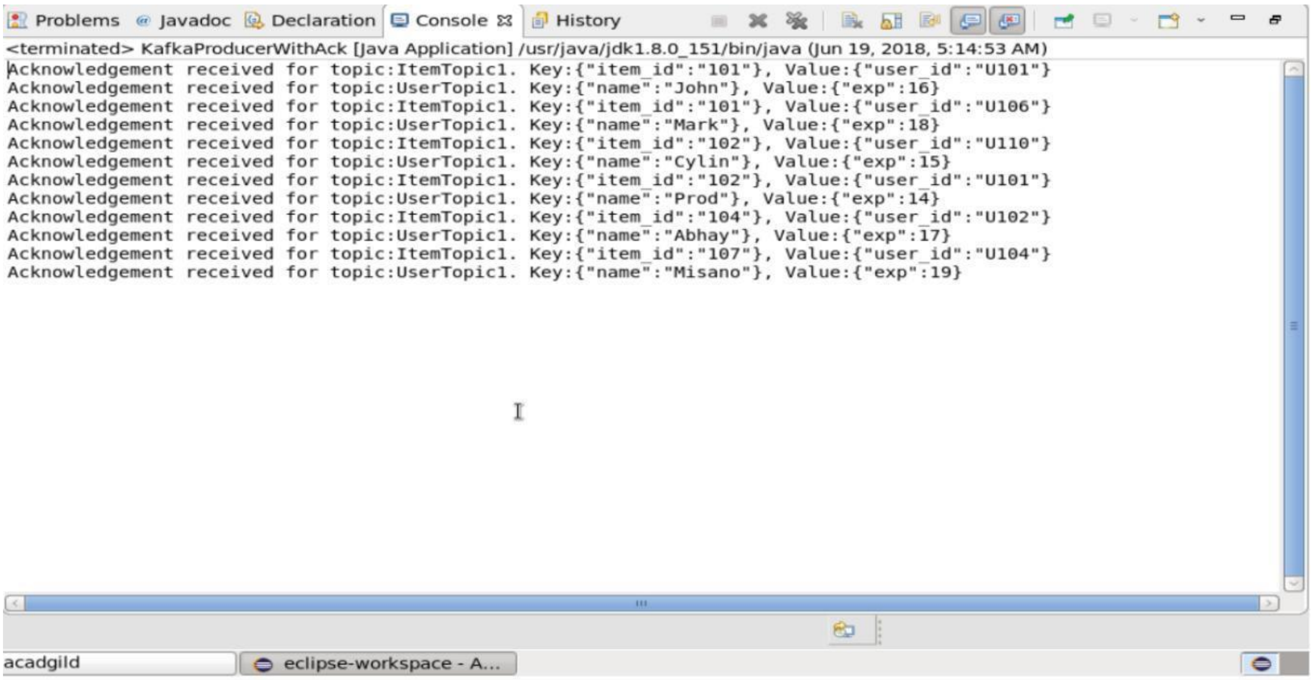
The application should attempt for 3 retries before giving any exception.

Pass **dataset_producer.txt** as the input file and -as delimiter.

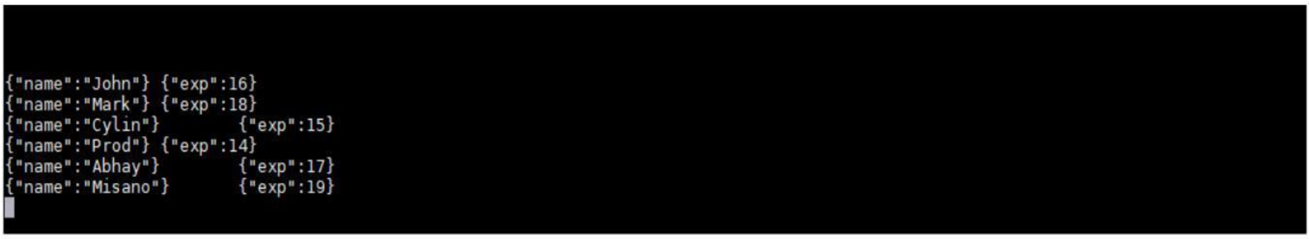
Java Code:

```java
package producer;

import org.apache.kafka.clients.producer.KafkaProducer;
import org.apache.kafka.clients.producer.ProducerRecord;
import java.io.BufferedReader;
import java.io.FileReader;
import java.io.IOException;
import java.util.Properties;

public class MyKafkaProducer {
public static void main(String[] args) throws IOException{
Properties props = new Properties();
props.put("bootstrap.servers", "localhost:9092");
props.put(“acks”,”all”);
props.put(“retries”,3);
props.put("key.serializer",
"org.apache.kafka.common.serialization.StringSerializer");
props.put("value.serializer",
"org.apache.kafka.common.serialization.StringSerializer");
KafkaProducer<String, String> producer = new KafkaProducer<>(props);
ProducerRecord<String, String> producerRecord = null;
String fileName = “/home/acadgild/dataset_producer.txt”; String delimiter = “-“;
try(BufferedReader br = new BufferedReader(new FileReader(fileName))) { for(String
line; (line = br.readLine()) != null; ) {
String[] tempArray = line.split(delimiter);
String topic = tempArray[0];
String key = tempArray[1];
String value = tempArray[2];
producerRecord = new ProducerRecord<String, String>(topic, key, value);
producer.send(producerRecord);
System.out.printf("Acknowledgement received for topic:%s. Key:%s, Value:%s\n",
topic, key, value);
}
}
producer.close();
}
}
```

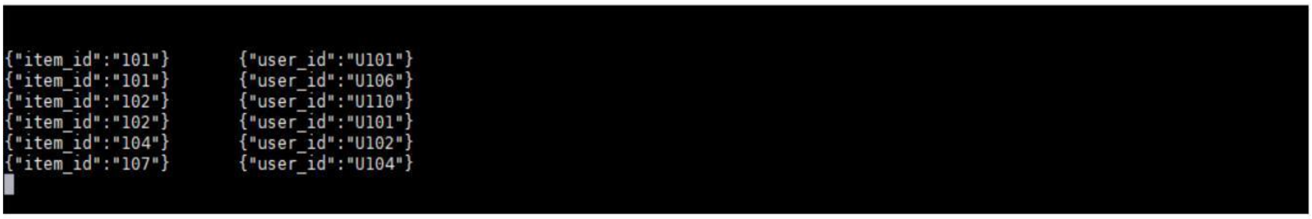
Output:



```
<terminated> KafkaProducerWithAck [Java Application] /usr/java/jdk1.8.0_151/bin/java (Jun 19, 2018, 5:14:53 AM)
Acknowledgement received for topic:ItemTopic1. Key:{"item_id":"101"}, Value:{"user_id":"U101"}
Acknowledgement received for topic:UserTopic1. Key:{"name":"John"}, Value:{"exp":16}
Acknowledgement received for topic:ItemTopic1. Key:{"item_id":"101"}, Value:{"user_id":"U106"}
Acknowledgement received for topic:UserTopic1. Key:{"name":"Mark"}, Value:{"exp":18}
Acknowledgement received for topic:ItemTopic1. Key:{"item_id":"102"}, Value:{"user_id":"U110"}
Acknowledgement received for topic:UserTopic1. Key:{"name":"Cylin"}, Value:{"exp":15}
Acknowledgement received for topic:ItemTopic1. Key:{"item_id":"102"}, Value:{"user_id":"U101"}
Acknowledgement received for topic:UserTopic1. Key:{"name":"Prod"}, Value:{"exp":14}
Acknowledgement received for topic:ItemTopic1. Key:{"item_id":"104"}, Value:{"user_id":"U102"}
Acknowledgement received for topic:UserTopic1. Key:{"name":"Abhay"}, Value:{"exp":17}
Acknowledgement received for topic:ItemTopic1. Key:{"item_id":"107"}, Value:{"user_id":"U104"}
Acknowledgement received for topic:UserTopic1. Key:{"name":"Misano"}, Value:{"exp":19}
```



```
{"name":"John"} {"exp":16}
{"name":"Mark"} {"exp":18}
{"name":"Cylin"}        {"exp":15}
{"name":"Prod"} {"exp":14}
{"name":"Abhay"}        {"exp":17}
{"name":"Misano"}       {"exp":19}
```



```
{"item_id":"101"}       {"user_id":"U101"}
{"item_id":"101"}       {"user_id":"U106"}
{"item_id":"102"}       {"user_id":"U110"}
{"item_id":"102"}       {"user_id":"U101"}
{"item_id":"104"}       {"user_id":"U102"}
{"item_id":"107"}       {"user_id":"U104"}
```