```
////////////// API  ////////////

// Captures all Hive connection info

class HiveStreamingEndPoint {
 public HiveStreamingEndPoint(String nameNodeUri, String metaStoreUri
                              , String database, String table
                              , Map<String,String> partitionSpec)
}


// Streaming Writer Class

class StreamingWriter {
  public enum TxState { NOT_ACTIVE, OPEN, COMMITTED, ABORTED }

    public StreamingWriter(HiveStreamingEndPoint hiveEndPoint)
          throws ConnectionError;

    public void close();


  // Transaction Support //
  public void fetchTransactionBatch(int batchCountHint)
                      throws ConnectionError;

  public boolean beginNextTransaction() throws InvalidTrasactionState;

  public void  commit() throws InvalidTrasactionState;

  public void  abort()  throws InvalidTrasactionState;

  public TxState getTransactionState();

  public int remainingTransactions();


  // Write Data //
  public void  write(byte[] record)
                  throws ConnectionError, InvalidTrasactionState;
  public void  write(Collection<byte[]> records)
                  throws ConnectionError, InvalidTrasactionState;


}
```

***** StreamingWriter(HiveStreamingEndPoint hiveEndPoint)**
Opens a connection to the hiveEndPoint

***** fetchTransactionBatch(int batchCountHint)** throws **...**
Acquires a new batch of transactions from Hive.
BatchCount is a hint from client indicating how many batches client needs

***** beginNextTransaction(...)** throws **...**
Switch to the next transaction in the batch.
returns false if there are no more transactions.

***** commit()**
Commits the currently open transaction.

***** abort() throws ...**
Aborts the currently open transaction.

***** write(**byte[] record**)**
Write a record.

***** write(**Collection<byte[]> records**)**
Write multiple records

***** getTransactionState()**
Get the current state of the transaction

***** remainingTransactions()**
Get a count of the unused transactions in the batch acquired by the last call to nextTransactionBatch.

***** close()**
Close any open connections and resources.

```
///// Simple Use case – write data indefinitely /////

Map<String, String> partInfo = new HashMap<String,String>();
partInfo.put("country", "USA");
partInfo.put("city", "San Jose");

HiveStreamingEndPoint hiveEP
     = new HiveStreamingEndPoint("namenode:123/path"
                                 , "metstore:4567", "db_name", "table_name"
                                 ,  partInfo );


StreamingWriter writer = new StreamingWriter(hiveEP);
int batchSize = 1000;
while(keepGoing) {
      try {
        if(writer.remainingTransactionCount()==0) {
            writer.fetchTransactionBatch(batchSizeHint);
        }
        writer.beginNextTransaction();
        for(int i = 0; i<batchSize; ++i)  {
          byte[] record = /* read Data from somewhere */;
          writer.write(record);
        }
        writer.commit();

      } catch( Whatever e)  {
        writer.abort();
      }
}
writer.close();
```