

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

```
// Captures all Hive connection info
```

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

---

```
// The Streaming connection
```

```
public class StreamingConnection {  
    public StreamingConnection(HiveStreamingEndPoint hiveEndPoint);  
  
    public TransactionBatch fetchTransactionBatch(int numTransactions)  
    throws ConnectionError;  
  
    public void close();  
}
```

**\*\*\* StreamingConnection(HiveStreamingEndPoint hiveEndPoint)**

Opens a connection to the hiveEndPoint

**\*\*\* fetchTransactionBatch(int numTransactions) throws ...**

Acquires a new batch of transactions from Hive.

numTransactions is a hint from client indicating how many transactions  
client needs

**\*\*\* close()**

Close any open connections and resources.

---

```
public class TransactionBatch {  
    public enum TxnState {INACTIVE, OPEN, COMMITTED, ABORTED }
```

```
    TransactionBatch(IMetaStoreClient msClient, int numTxns, LockRequest  
    lockRequest) throws ConnectionError ;
```

```
    public TxnState getCurrentTransactionState();
```

```
    public int remainingTransactions()
```

```
    public void beginNextTransaction() throws StreamingException;
```

```
    public void commit() throws StreamingException;
```

```
    public void abort() throws StreamingException;
```

```
// Write Data //
public void write(byte[] record);
public void write(Collection<byte[]> records);

}
```

**\*\*\* 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 fetchTransactionBatch. Current transaction is not considered part of remaining transactions.

```
///// 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"  
                                , "metastore:4567", "db_name", "table_name"  
                                , partInfo );
```

```
StreamingConnection conn = new StreamingConnection(hiveEP);  
int batchSize = 1000;  
while(keepGoing) {  
    for(TransactionBatch txnBatch = conn.fetchTransactionBatch(10);  
        txnBatch.remainingTransactions()==0;  
        txnBatch.beginNextTransaction() ) {  
        try {  
            for(int i=0; i<batchSize; ++i) {  
                byte[] record = /* read Data from somewhere */;  
                txnBatch.write(record);  
            }  
            txnBatch.commit();  
        } catch( Whatever e) {  
            txnBatch.abort();  
        }  
    } // for  
} // while  
conn.close();
```