

# ThriftExamples03

Cassandra <= 0.3

```
<?php
$GLOBALS['THRIFT_ROOT'] = '/usr/share/php/Thrift';

require_once $GLOBALS['THRIFT_ROOT'].'/packages/cassandra/Cassandra.php';
require_once $GLOBALS['THRIFT_ROOT'].'/transport/TSocket.php';
require_once $GLOBALS['THRIFT_ROOT'].'/protocol/TBinaryProtocol.php';
require_once $GLOBALS['THRIFT_ROOT'].'/transport/TFramedTransport.php';
require_once $GLOBALS['THRIFT_ROOT'].'/transport/TBufferedTransport.php';

try {
    $socket = new TSocket('localhost', 9160);
    $transport = new TBufferedTransport($socket, 1024, 1024);
    $protocol = new TBinaryProtocol($transport);
    $client = new CassandraClient($protocol);

    $transport->open();
    /* Insert some data into base attributes */
    $key_user_id = "1";
    $timestamp = time();
    $client->insert("users", $key_user_id, "base_attributes:name", "Chris Goffinet", $timestamp, false);
    $client->insert("users", $key_user_id, "base_attributes:age", "24", $timestamp, false);

    $start = -1;
    $end = -1;
    $result = $client->get_slice("users", $key_user_id, "base_attributes", $start, $end);

    print_r($result);
    $transport->close();
} catch (TException $tx) {
    print 'TException: '.$tx->why."\n";
}

?>
```

## Python

Make sure that when you generate the gen-py folder, you move the cassandra/ folder into your Python's site-packages.

```

from thrift import Thrift
from thrift.transport import TTransport
from thrift.transport import TSocket
from thrift.transport import THttpClient
from thrift.protocol import TBinaryProtocol
from cassandra import Cassandra
from cassandra.ttypes import *
import time

socket = TSocket.TSocket("localhost", 9160)
transport = TTransport.TBufferedTransport(socket)
protocol = TBinaryProtocol.TBinaryProtocol(transport)
client = Cassandra.Client(protocol)

try:
    transport.open()
    key_user_id = "1"
    timestamp = time.time()
    client.insert("users",key_user_id,"base_attributes:name", "Chris Goffinet", timestamp, False)
    client.insert("users",key_user_id,"base_attributes:age", "24", timestamp, False)
    start = -1
    end = -1
    result = client.get_slice("users", key_user_id,"base_attributes", start, end)
    print result
except Thrift.TException, tx:
    print 'Thrift: %s' % tx.message
finally:
    transport.close()

```

## Java

```

import java.io.UnsupportedEncodingException;
import java.util.List;

import org.apache.thrift.TException;
import org.apache.thrift.protocol.TBinaryProtocol;
import org.apache.thrift.protocol.TProtocol;
import org.apache.thrift.transport.TSocket;
import org.apache.thrift.transport.TTransport;
import org.apache.thrift.transport.TTransportException;

import org.apache.cassandra.service.InvalidRequestException;
import org.apache.cassandra.service.NotFoundException;
import org.apache.cassandra.service.UnavailableException;
import org.apache.cassandra.service.column_t;
import org.apache.cassandra.service.Cassandra.Client;

public class CClient {

    public static void main(String[] args) {

        TTransport tr = new TSocket("localhost", 9160);
        TProtocol proto = new TBinaryProtocol(tr);
        try {

            Client client = new Client(proto);
            tr.open();
            String key_user_id = "1";
            long timestamp = System.currentTimeMillis();
            client.insert("users", key_user_id, "base_attributes:name",
                "Chris Goffinet".getBytes("UTF-8"), timestamp, false);
            client.insert("users", key_user_id, "base_attributes:age",
                "24".getBytes("UTF-8"), timestamp, false);

            int start = -1;
            int end = -1;
            List<column_t> results = client.get_slice("users", key_user_id,
                "base_attributes", start, end);

            for (column_t col : results) {
                System.out.println(col.getColumnName() + " -> " +
                    new String(col.getValue(),"UTF-8"));
            }

        } catch (TTransportException e) {
            e.printStackTrace();
        } catch (UnsupportedEncodingException e) {
            e.printStackTrace();
        } catch (InvalidRequestException e) {
            e.printStackTrace();
        } catch (UnavailableException e) {
            e.printStackTrace();
        } catch (TException e) {
            e.printStackTrace();
        } catch (NotFoundException e) {
            e.printStackTrace();
        } finally {
            tr.close();
        }
    }
}

```