

## Panduan Pemrograman Apache Thrift 2015

(tidak dikumpulkan. Tugas yang dikumpulkan terdapat pada file Tugas 2.pdf)

### Lingkungan Linux

1. Untuk lingkungan Linux, ikuti <http://thrift.apache.org/docs/BuildingFromSource>

### Lingkungan Windows

1. Untuk lingkungan Windows, download file thrift-0.9.2.exe, libthrift-0.9.2.jar dan lib.zip (yg berisi library tambahan yang diperlukan) dari <http://kuliah.informatika.org>
2. libthrift-0.9.2.jar dan lib.zip dapat dibangun ulang dengan mendownload source dari <http://thrift.apache.org/download>, dan menjalankan ant command dari direktori ./lib/java  
Buatlah file definisi service thrift sederhana, dengan nama calculator.thrift  
namespace java if4031

```
typedef i32 int
service CalculatorService
{
    int multiply(1:int n1, 2:int n2),
    int add(1:int n1, 2:int n2),
}
```

3. untuk pembangkitan kode, pastikan thrift-0.9.2.exe berada dalam PATH yang ada pada Windows, dan jalankan perintah:

```
thrift-0.9.2 --gen java calculator.thrift
```

4. Implementasikan kelas handler sbb:

```
import org.apache.thrift.TException;
import if4031.CalculatorService;
public class CalculatorHandler implements CalculatorService.Iface
{
    @Override
    public int multiply(int n1, int n2) throws TException {
        System.out.println("Multiply(" + n1 + "," + n2 + ")");
        return n1 * n2;
    }
    @Override
    public int add(int n1, int n2) throws TException {
        System.out.println("Add(" + n1 + "," + n2 + ")");
        return n1 + n2;
    }
}
```

5. Implementasikan kelas server sbb:

```
import org.apache.thrift.server.TServer;
import org.apache.thrift.server.TServer.Args;
import org.apache.thrift.server.TSimpleServer;
```

```

import org.apache.thrift.transport.TServerSocket;
import org.apache.thrift.transport.TServerTransport;
import if4031.CalculatorService;
public class CalculatorServer {

    public static CalculatorHandler handler;

    public static CalculatorService.Processor processor;

    public static void main(String [] args) {
        try {
            handler = new CalculatorHandler();
            processor = new CalculatorService.Processor(handler);

            Runnable simple = new Runnable() {
                public void run() {
                    simple(processor);
                }
            };

            new Thread(simple).start();
        } catch (Exception x) {
            x.printStackTrace();
        }
    }

    public static void simple(CalculatorService.Processor processor) {
        try {
            TServerTransport serverTransport = new TServerSocket(9090);
            TServer server = new TSimpleServer(new
Args(serverTransport).processor(processor));

            System.out.println("Starting the simple server...");
            server.serve();
        } catch (Exception e) {
            e.printStackTrace();
        }
    }
}

```

#### 6. Implementasikan kelas client sbb:

```

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 if4031.CalculatorService;

```

```

public class CalculatorClient {
    public static void main(String [] args) {

        try {
            TTransport transport;

            transport = new TSocket("localhost", 9090);
            transport.open();

            TProtocol protocol = new TBinaryProtocol(transport);
            CalculatorService.Client client = new
CalculatorService.Client(protocol);

            perform(client);

            transport.close();
        } catch (TException x) {
            x.printStackTrace();
        }
    }

    private static void perform(CalculatorService.Client client)
throws TException
    {

        int product = client.multiply(3,5);
        System.out.println("3*5=" + product);
    }
}

```

7. Kompilasi dengan melakukan setting classpath yang berisi library yang diperlukan. (minimal: libthrift-0.9.2.jar, log4j-1.2.14.jar, slf4j-log4j12-1.5.8.jar, slf4j-api-1.5.8.jar)
8. Jalankan kelas server dengan melakukan setting classpath yang berisi library yang diperlukan (minimal: libthrift-0.9.2.jar, log4j-1.2.14.jar, slf4j-log4j12-1.5.8.jar, slf4j-api-1.5.8.jar)
9. Jalankan kelas client dengan melakukan setting classpath yang berisi library yang diperlukan (minimal: libthrift-0.9.2.jar, log4j-1.2.14.jar, slf4j-log4j12-1.5.8.jar, slf4j-api-1.5.8.jar)
10. Pelajari dan coba jenis server dan protokol yang berbeda