

例3-15 Event-driven Source的一个示例

```
package usingflume.ch03;
```

```
public class TransactionSource extends AbstractEventDrivenSource implements  
    FlumeCreditCardAuth {
```

```
    private static final String CONF_HOST = "host";  
    private static final String CONF_PORT = "port";  
    private static final String DELIMITER = ";";
```

```
    private String host;  
    private Integer port;  
    private Server srv;
```

```
    @Override
```

```
    protected void doConfigure(Context context) throws FlumeException {  
        host = context.getString(CONF_HOST);  
        Preconditions.checkArgument(host != null && !host.isEmpty(),  
            "Host must be specified");  
        port = Preconditions.checkNotNull(context.getInteger(CONF_PORT),  
            "Port must be specified");  
    }
```

```
    @Override
```

```
    protected void doStart() throws FlumeException {  
        srv = new NettyServer(  
            new SpecificResponder(FlumeCreditCardAuth.class, this),  
            new InetSocketAddress(host, port));  
        srv.start();  
        super.start();  
    }
```

```
    @Override
```

```
    protected void doStop() throws FlumeException {  
        srv.close();  
        try {  
            srv.join();  
        } catch (InterruptedException e) {  
            throw new FlumeException("Interrupted while waiting for Netty Server to" +  
                " shut down.", e);  
        }  
    }
```

```
    @Override
```

```
    public Status transactionsCompleted(List<CreditCardTransaction> transactions)
```