

Akka学习笔记：Actor消息处理-请求和响应(1)

Akka学习笔记系列文章：

[《Akka学习笔记：ACTORS介绍》](#)
[《Akka学习笔记：Actor消息传递\(1\)》](#)
[《Akka学习笔记：Actor消息传递\(2\)》](#)
[《Akka学习笔记：日志》](#)
[《Akka学习笔记：测试Actors》](#)
[《Akka学习笔记：Actor消息处理-请求和响应\(1\)》](#)
[《Akka学习笔记：Actor消息处理-请求和响应\(2\)》](#)
[《Akka学习笔记：ActorSystem\(配置\)》](#)
[《Akka学习笔记：ActorSystem\(调度\)》](#)

在前面的文章[《Akka学习笔记：Actor消息传递\(1\)》](#)和[《Akka学习笔记：Actor消息传递\(2\)》](#)。我们仅仅将消息发送到Actor，并没有期待从Actor发来的响应。

从技术上讲，我们给Actor发送消息，除了没有响应，目标的Actor将对这条消息进行如下操作：

- 1、给发送者发送一个响应，在我们的例子中，TeacherActor将发送一个quote到StudentActor作为响应。或者
- 2、给其他的Actor响应，而这个Actor可能也会作出响应或者转发等。Routers和Supervisors就是这个例子，我们将在后面看到。

请求和响应

这里我们只研究场景1



如果想及时了

解Spark、Hadoop或者Hbase相关的文章，欢迎关注微信公共帐号：iteblog_hadoop

上图传达了我們此刻想表达的意思。为了简单，我没有在图中画出ActorSystem、Dispatcher和Mailboxes。

- 1、DriverApp给StudentActor发送了一个InitSignal消息；
 - 2、StudentActor对InitSignal消息作出反应，并且发送了QuoteRequest消息给TeacherActor；
 - 3、TeacherActor用QuoteResponse作出了响应；
 - 4、StudentActor仅仅将QuoteResponse作为日志打印到控制台/logger。
- 我们将在testcase中进行确认。让我们先来看看上面4点：

一、DriverApp给StudentActor发送了一个InitSignal消息



如果想及时了

解Spark、Hadoop或者Hbase相关的文章，欢迎关注微信公共帐号：iteblog_hadoop

到现在为止，你可能已经猜到了DriverApp将要做的事，仅仅4件事

1)、初始化ActorSystem

```
//Initialize the ActorSystem  
val system = ActorSystem("UniversityMessageSystem")
```

2)、创建TeacherActor

```
//create the teacher actor  
val teacherRef = system.actorOf(Props[TeacherActor], "teacherActor")
```

3)、创建StudentActor

```
//create the Student Actor - pass the teacher actorref  
//as a constructor parameter to StudentActor  
val studentRef = system.actorOf(Props(new StudentActor(teacherRef)), "studentActor")
```

你可能注意到，我用TeacherActor 作为StudentActor构造参数，并传入到ActorRef中，这样StudentActor 将可以用ActorRef发送消息到TeacherActor。还有其他的方法可以实现类似的功能（比如利用Props），但是当我们想查看 Supervisors 和Routers的时候，这个方法很方便。我们很快看到一个自Actors，但是在这里的语义不对：Student创建Teacher听起来很不好。不是吗？

4)、最后，DriverApp将给StudentActor发送InitSignal。所以StudentActor可以给TeacherActor发送QuoteRequest 消息。

```
//send a message to the Student Actor  
studentRef ! InitSignal
```

这就是DriverClass所做的。完整的代码如下

```
package me.rerun.akkanotes.messaging.requestresponse

import akka.actor.ActorSystem
import akka.actor.Props
import me.rerun.akkanotes.messaging.protocols.StudentProtocol._
import akka.actor.ActorRef

object DriverApp extends App {

  //Initialize the ActorSystem
  val system = ActorSystem("UniversityMessageSystem")

  //construct the teacher actor
  val teacherRef = system.actorOf(Props[TeacherActor], "teacherActor")

  //construct the Student Actor - pass the teacher actorref
  // as a constructor parameter to StudentActor
  val studentRef = system.actorOf(Props(new StudentActor(teacherRef)), "studentActor")

  //send a message to the Student Actor
  studentRef ! InitSignal

  //Let's wait for a couple of seconds before we shut down the system
  Thread.sleep(2000)

  //Shut down the ActorSystem.
  system.shutdown()

}
```

本文英文地址：<http://rerun.me/2014/10/06/akka-notes-actor-messaging-request-and-response-3/>

本博客文章除特别声明，全部都是原创！
原创文章版权归过往记忆大数据（[过往记忆](#)）所有，未经许可不得转载。
本文链接：【】（）