

# MailBox



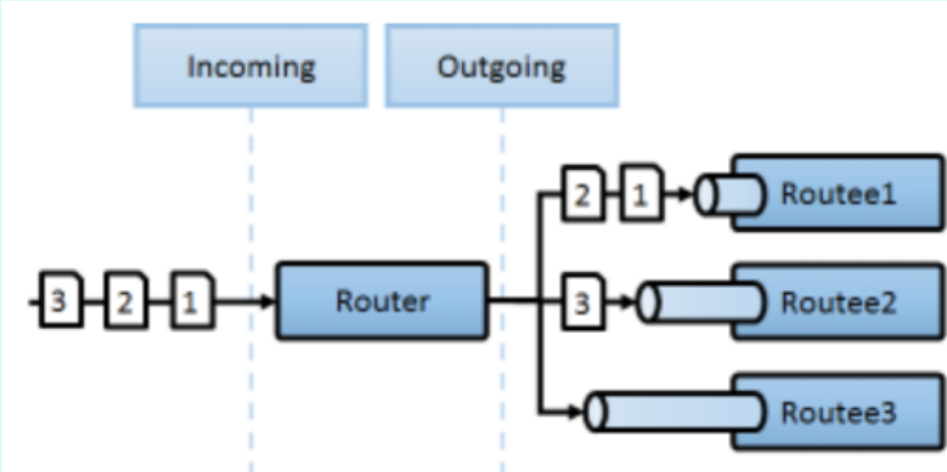
Actor	MailBox
-------	---------

•

;

•

smallest-mailbox-pool () .



```
:smallest-mailbox-pool
Routee3 , .
```

```
akka.actor.deployment {
  /some-pool {
    router = smallest-mailbox-pool
    nr-of-instances = 5
  }
}
```

( vs ) 5 (IsBug,IsSecurityFlaw,IsBug,IsBug,IsSecurityFlaw) .

;

## akka.conf

```
# .
# code : https://github.com/psmon/AkkaForNetCore/blob/master/AkkaNetCore/Models/Message/IssueTrackerMailbox.cs
# , ( ) .
my-custom-mailbox {
    mailbox-type : "AkkaNetCore.Models.Message.IssueTrackerMailbox, AkkaNetCore"
}

actor.deployment {
    /mymailbox {
        mailbox = my-custom-mailbox
    }
}
```

```
using System;
using Akka.Actor;
using Akka.TestKit;
using AkkaNetCore.Models.Message;
using Xunit;
using Xunit.Abstractions;

namespace AkkaNetCoreTest.Actors
{
    public class MailBoxTest : TestKitXunit
    {
        protected TestProbe probe;
        protected IActorRef mailBoxActor;

        public MailBoxTest(ITestOutputHelper output) : base(output)
        {
            // akkaconfig .
            // TestKitXunit
            // TestKitXunit.akkaConfig :
            Setup();
        }

        public void Setup()
        {
            // Qa .
            probe = this.CreateTestProbe();
            var mailboxOpt = Props.Create<MailBoxActor>(probe).WithMailbox("my-custom-mailbox");
            mailBoxActor = Sys.ActorOf(mailboxOpt, "mymailbox");
        }

        [Fact]
        public void ____()
        {
            // isBug false .
            Issue msg1 = new Issue("test1", true);
            Issue msg2 = new Issue("test2", false);
            Issue msg3 = new Issue("test3", true);
            Issue msg4 = new Issue("test4", true);
            Issue msg5 = new Issue("test5", false);
            mailBoxActor.Tell(msg1);
            mailBoxActor.Tell(msg2);
            mailBoxActor.Tell(msg3);
            mailBoxActor.Tell(msg4);
            mailBoxActor.Tell(msg5);

            for(int i = 0; i < 5; i++)
            {
                probe.ExpectMsg<Issue>(issue =>
                {
                    Console.WriteLine($"IssueInfo : Message:{issue.Message} IsSecurityFlaw:{issue.IsSecurityFlaw} IsBug:{issue.IsBug} ");
                });
            }
        }
    }
}
```

```

        });
    }
    /*
    IssueInfo: Message: test2 IsSecurityFlaw:True IsBug:False
    IssueInfo : Message: test5 IsSecurityFlaw:True IsBug:False
    IssueInfo : Message: test4 IsSecurityFlaw:False IsBug:True
    IssueInfo : Message: test3 IsSecurityFlaw:False IsBug:True
    IssueInfo : Message: test1 IsSecurityFlaw:False IsBug:True
    */
}

public class MailBoxActor : ReceiveActor
{
    IActorRef notifyQa;
    public MailBoxActor(IActorRef _notifyQa)
    {
        notifyQa = _notifyQa;
        //MailBoxTest
        Receive<Issue>(issue => {
            //Console.WriteLine($"IssueInfo : Message:{issue.Message} IsSecurityFlaw:{issue.IsSecurityFlaw}
IsBug:{issue.IsBug} ");
            notifyQa.Tell(issue);
        });
        //InboxTest
        Receive<string>(msg =>
        {
            if (msg == "hello")
                Sender.Tell("world");
        });
    }
}
}

```