

潤沁實業

潤沁網路大學

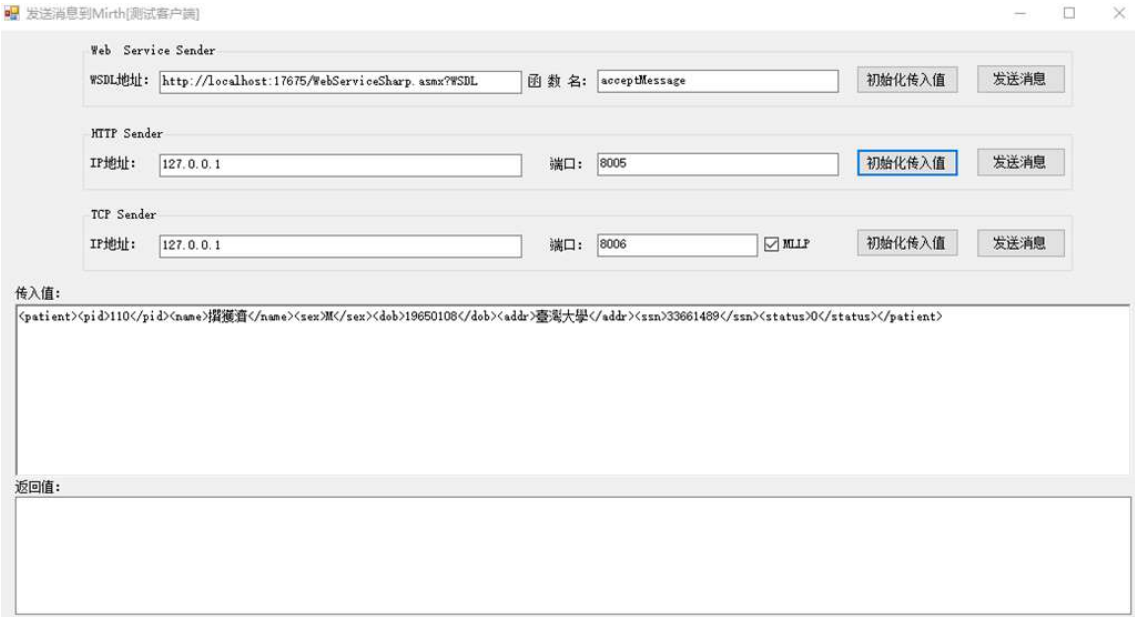
博客园 首页 新随笔 联系 订阅 管理

第六課-Channel Study For TCP Listener & HTTP Listener & Web Service Listener About Response Handler

经过前面章节的课程，对Mirth Connect在系统集成与数据交互中的使用有了一个大概的了解；大家一定有个疑惑，Mirth Connect如何组织响应消息并返回给调用者？今天我们就来继续深入讲解Response响应处理机制。

C#示例：

本课程我们将通过C#编程语言调用Mirth Connect接口服务来讲述，C#示例截图如下：



C#测试客户端

WebServiceSharp

潤沁實業开放给Mirth Connect的C#版本接口服务

支持下列操作，有关正式定义，请查看[服务说明](#)。

- [DBConnTest](#)
数据库连接测试
- [HelloWorld](#)
测试本Webservice服务是否搭建OK
- [acceptMessage](#)
取得患者信息

C#版本Webservice接口服务

TCP Listener 信道示例：

公告

昵称： 潤沁網路大學
园龄： 1年10个月
粉丝： 2
关注： 0

< 2021年3月				
日	一	二	三	
28	1	2	3	
7	8	9	10	
14	15	16	17	
21	22	23	24	
28	29	30	31	
4	5	6	7	

搜索

常用链接

我的随笔
我的评论
我的参与
最新评论
我的标签

我的标签

Mirth(18)
HL7(7)
DataBase(2)

随笔分类

DataBase(2)
HL7(7)
Mirth(15)

随笔档案

2021年2月(2)
2021年1月(20)

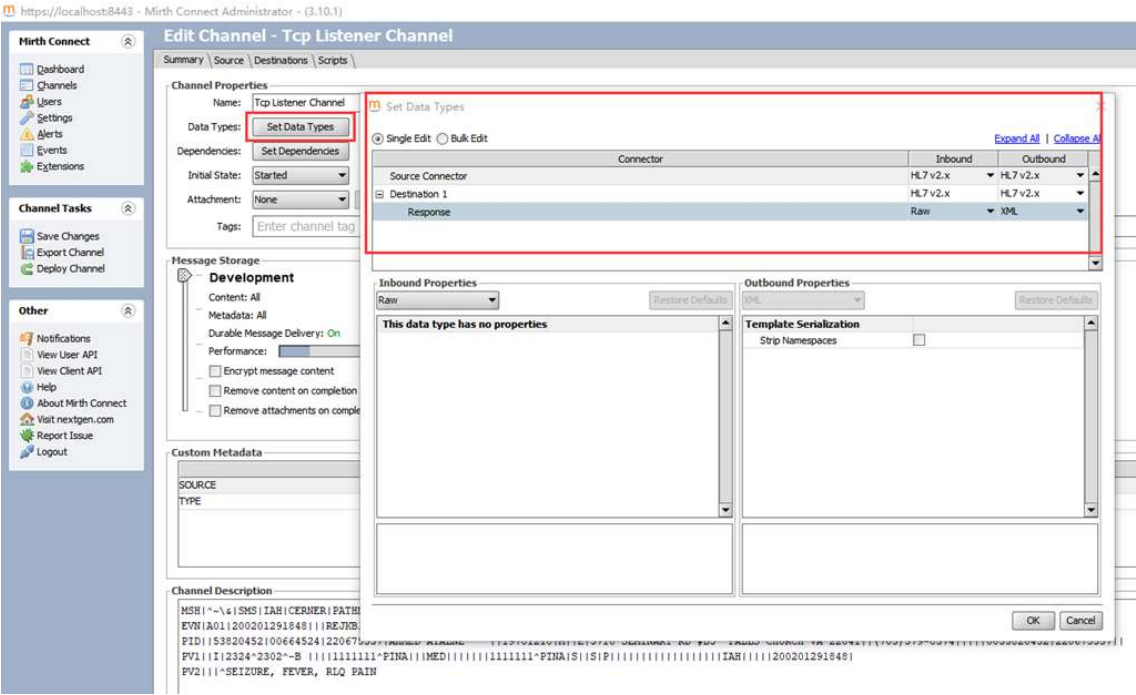
文章分类

Mirth(3)

最新评论

1. Re:第八課-Channel Study Custom JAR Lib
受益匪浅!!!

阅读排行榜

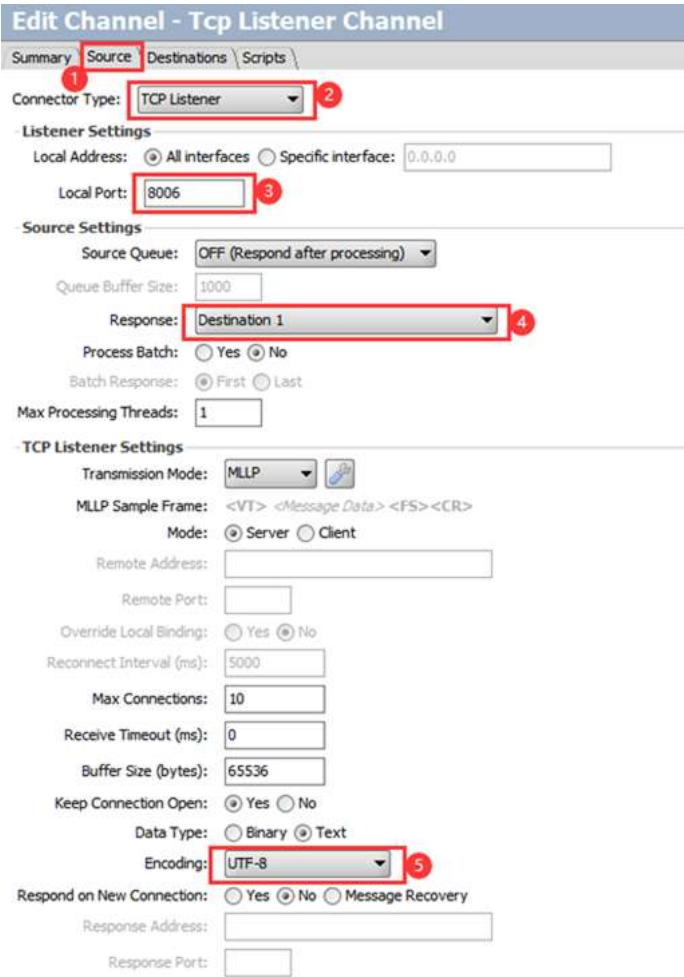


- 1. HL7传输协议(161)
- 2. 第壹課-Install: Mirth Connect 安装步骤(99)
- 3. 开篇: Mirth Connect 系统集(5)
- 4. HL7标准的版本(75)
- 5. 第三課: 信道学习Source C Destinations File Writer(60)

评论排行榜

- 1. 第八課-Channel Study For R Lib(1)

新建信道并设置源端输入输出参数格式、目的端输入输出格式、响应输入输出参数格式



设置源端TCP Listener参数

Edit Channel - Tcp Listener Channel

Summary | Source | Destinations | Scripts

Status	Destination
Enabled	Destination 1

Connector Type: **Channel Writer** ☐ Wait for previous destination

Destination Settings

Queue Messages: ☒ Never ☐ On Failure ☐ Always

Advanced Queue Settings: Retries

Validate Response: ☐ Yes ☒ No

Reattach Attachments: ☒ Yes ☐ No

Channel Writer Settings

Channel Id: **<None>**

Message Metadata:

Template: `{message.encodedData}`

设置目的端Channel Writer参数

https://localhost:8443 - Mirth Connect Administrator - (3.10.1)

Edit Channel - Tcp Listener Channel

Summary | **Source** | Destinations | Scripts

Connector Type: **TCP Listener**

Listener Settings

Local Address: ☒ All interfaces ☐ Specific interface:

Local Port:

Source Settings

Source Queue: **OFF (Respond after processing)**

Queue Buffer Size:

Response: **Destination 1**

Process Batch: ☐ Yes ☒ No

Batch Response: ☒ First ☐ Last

Max Processing Threads:

TCP Listener Settings

Transmission Mode: **MLLP**

MLLP Sample Frame: `<VT> <Message Data> <FS> <CR>`

Mode: ☒ Server ☐ Client

Remote Address:

Remote Port:

Override Local Binding: ☐ Yes ☒ No

Reconnect Interval (ms):

Max Connections:

Receive Timeout (ms):

Buffer Size (bytes):

Keep Connection Open: ☒ Yes ☐ No

Data Type: ☐ Binary ☒ Text

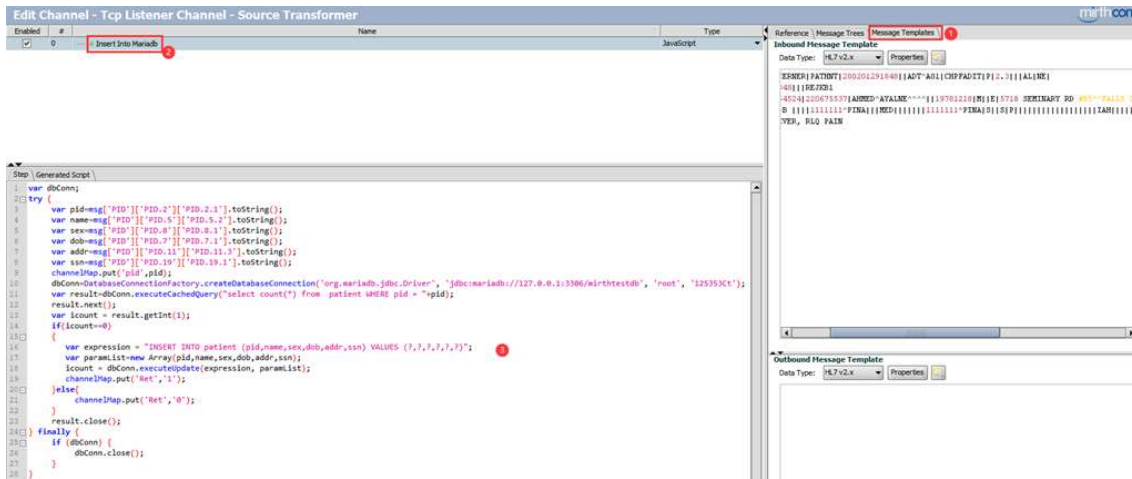
Encoding: **UTF-8**

Respond on New Connection: ☐ Yes ☒ No ☐ Message Recovery

Response Address:

Response Port:

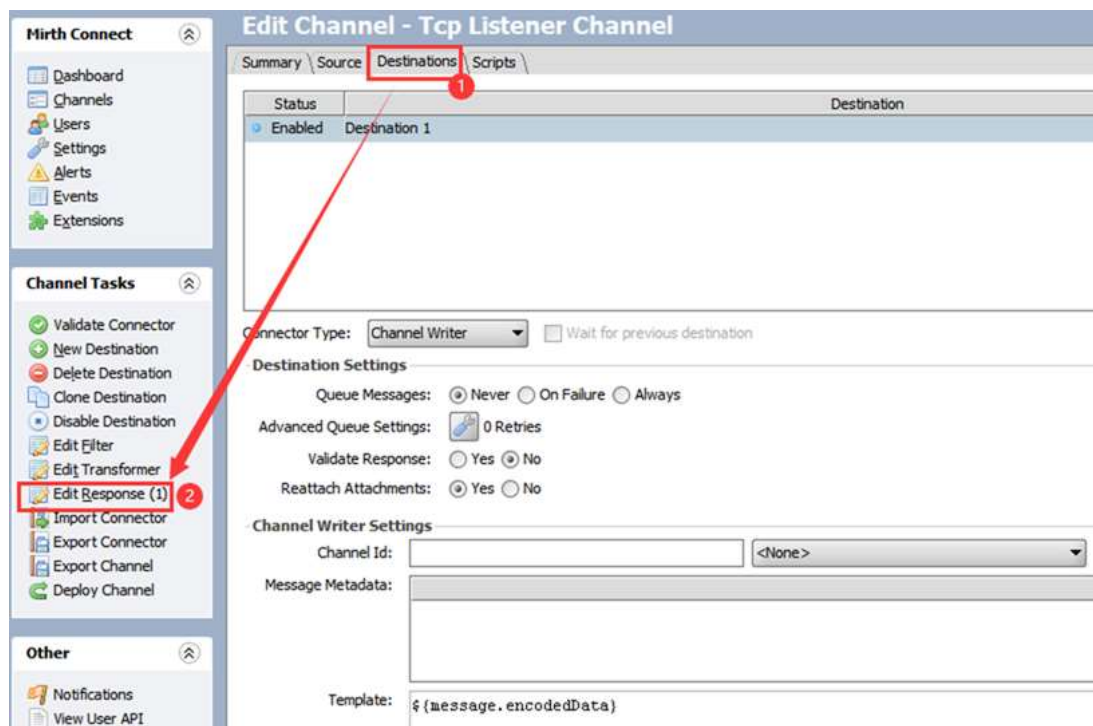
设置源端转换规则



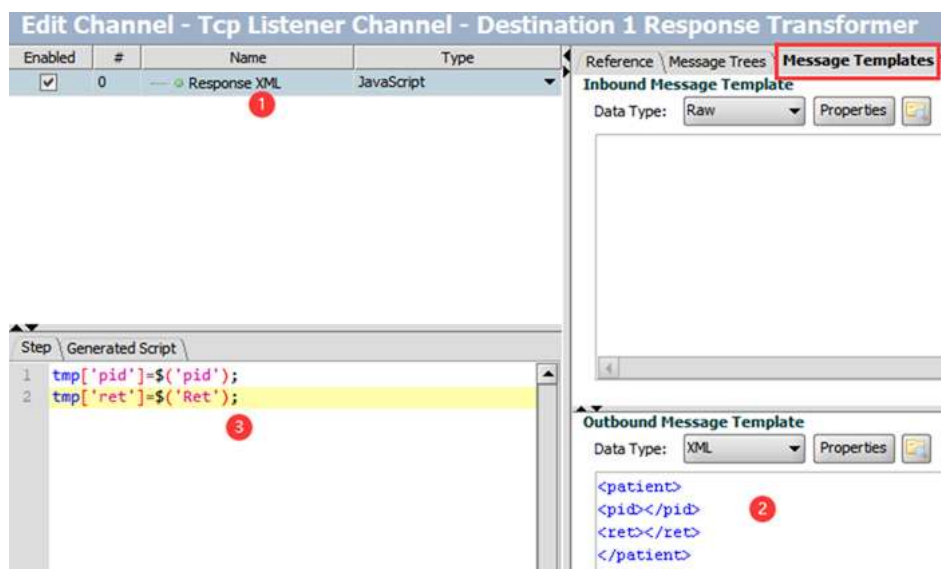
源端新建JS转换步骤，在“输入消息模板”录入如下HL7消息模板并添加如下js脚本

```
MSH|^~\&|SMS|IAH|CERNER|PATHNT|200201291848||ADT^A01|CHPFADIT|P|2.3|||AL|NE|
EVN|A01|200201291848|||RE|KB1
PID||53820452|00664524|220675537|AHMED^AYALNE^^^|19781218|M|E|5718 SEMINARY RD #B5^^FALLS CHURCH^VA^22041|
(703)379-8374||||0053820452|220675537|
PV1||I|2324^2302~B |||1111111^PINA||MED|||||1111111^PINA|S|S|P|||||||||IAH|||||200201291848|
PV2|||^SEIZURE, FEVER, RLQ PAIN
```

```
var dbConn;
try {
    var pid=msg['PID']['PID.2']['PID.2.1'].toString();
    var name=msg['PID']['PID.5']['PID.5.2'].toString();
    var sex=msg['PID']['PID.8']['PID.8.1'].toString();
    var dob=msg['PID']['PID.7']['PID.7.1'].toString();
    var addr=msg['PID']['PID.11']['PID.11.3'].toString();
    var ssn=msg['PID']['PID.19']['PID.19.1'].toString();
    channelMap.put('pid',pid);
    dbConn=DatabaseConnectionFactory.createDatabaseConnection('org.mariadb.jdbc.Driver',
'jdbc:mariadb://127.0.0.1:3306/mirthtestdb', 'root', '125353Ct');
    var result=dbConn.executeCachedQuery("select count(*) from patient WHERE pid = '"+pid);
    result.next();
    var icount = result.getInt(1);
    if(icount==0)
    {
        var expression = "INSERT INTO patient (pid,name,sex,dob,addr,ssn) VALUES (?, ?, ?, ?, ?, ?)";
        var paramList=new Array(pid,name,sex,dob,addr,ssn);
        icount = dbConn.executeUpdate(expression, paramList);
        channelMap.put('Ret','1');
    }else{
        channelMap.put('Ret','0');
    }
    result.close();
} finally {
    if (dbConn) {
        dbConn.close();
    }
}
```



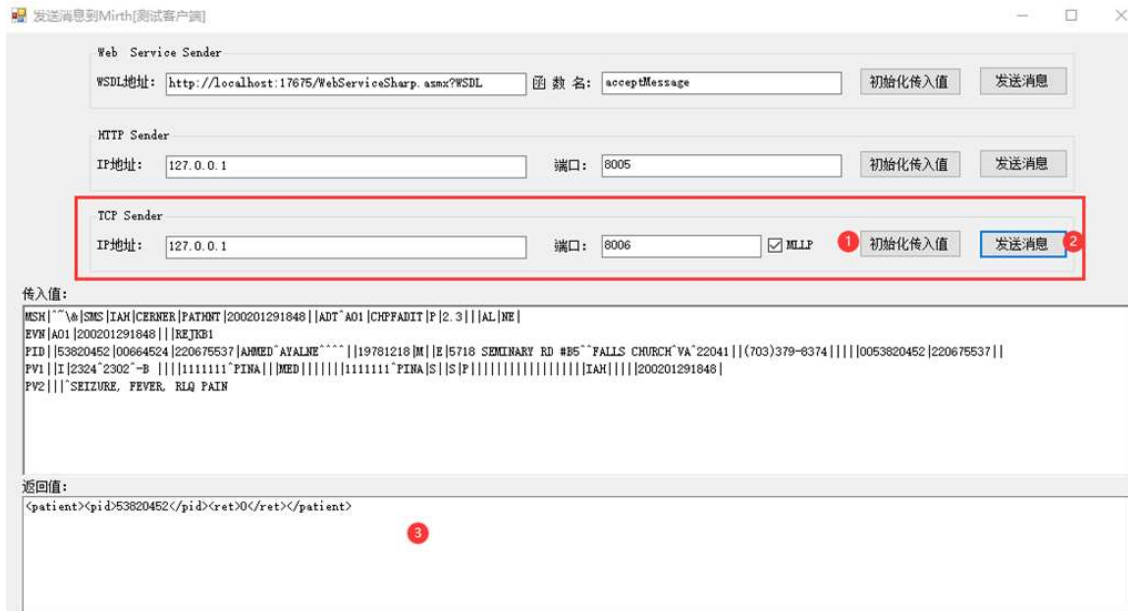
目的端响应转换规则



在'输出消息模板'中录入如下xml格式模板并添加如下js脚本

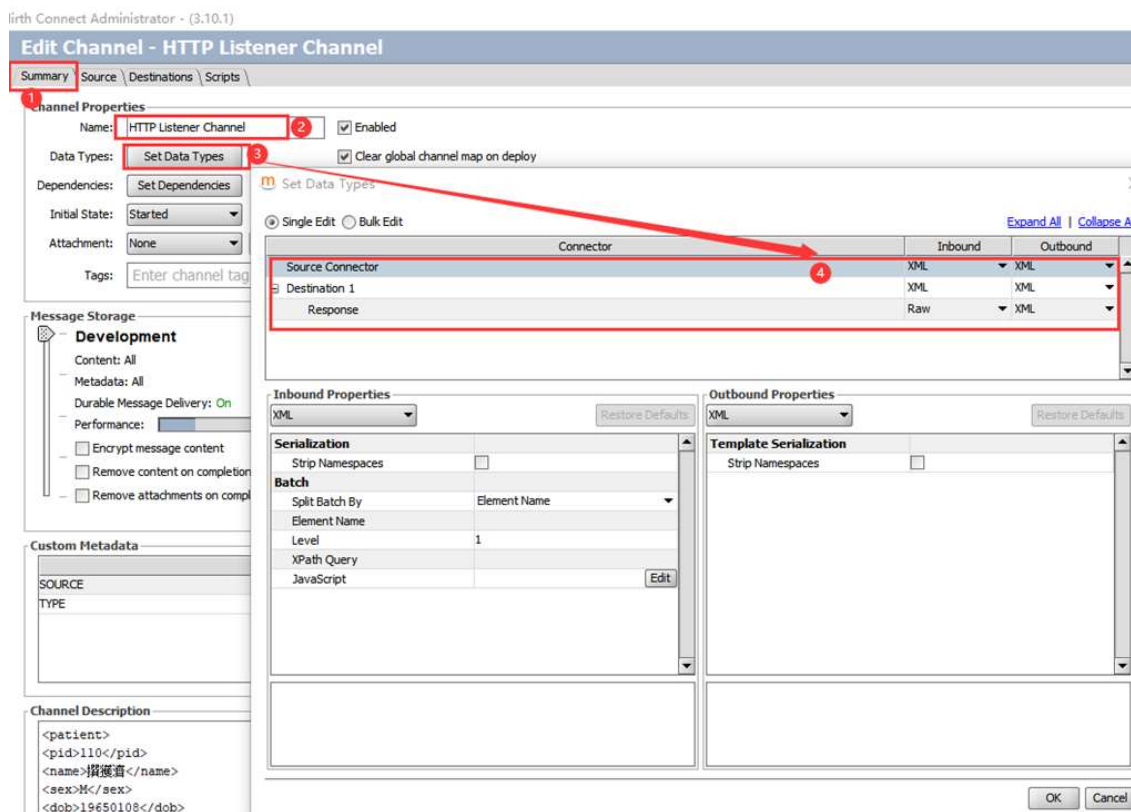
```
<patient>
<pid></pid>
<ret></ret>
</patient>

tmp['pid']='${pid}';
tmp['ret']='${Ret}';
```



运行C#版TCP Sender，进行如上测试，成功获取响应，验证Mirth Connect 的TCP Listener信道OK!

HTTP Listener 信道示例:



新建信道并设置源端输入输出参数格式、目的端输入输出格式、响应输入输出参数格式

https://localhost:8443 - Mirth Connect Administrator - (3.10.1)

Edit Channel - HTTP Listener Channel

Summary | **Source** | Destinations | Scripts

Connector Type: **HTTP Listener**

Listener Settings

Local Address: ☐ All interfaces ☐ Specific interface: 0.0.0.0

Local Port: **8005**

Source Settings

Source Queue: OFF (Respond after processing)

Queue Buffer Size: 1000

Response: **Destination 1**

Process Batch: ☐ Yes ☒ No

Batch Response: ☐ First ☒ Last

Max Processing Threads: 1

HTTP Authentication

Authentication Type: None

HTTP Listener Settings

Base Context Path:

Receive Timeout (ms): 30000

Message Content: ☒ Plain Body ☐ XML Body

Parse Multipart: ☒ Yes ☐ No

Include Metadata: ☐ Yes ☒ No

Binary MIME Types: application/.*(?!(json|xml)\$)image/..*video/..*audio/..* ☒ Regular Expression

HTTP URL: http://localhost:8005/

Response Content Type: text/plain

Response Data Type: ☐ Binary ☒ Text

Charset Encoding: UTF-8

Response Status Code:

Response Headers: ☒ Use Table ☐ Use Map:

設置源端HTTP Listener參數

Edit Channel - HTTP Listener Channel - Source Transformer

Enabled: ☒ # 0 Name: Insert One Patient To Mariadb Type: JavaScript

Message Templates

Inbound Message Template

Data Type: XML

Properties

<patient><pid>110</pid><name>撰獲濟</name><sex>M</sex><dob>19650108</dob><addr>臺灣大學</addr><ssn>33661489</ssn><status>0</status></patient>

Outbound Message Template

Data Type: XML

Properties

Generated Script

```

1 var dbConn;
2 try {
3   var pid=msg['pid'].toString();
4   var name=msg['name'].toString();
5   var sex=msg['sex'].toString();
6   var dob=msg['dob'].toString();
7   var addr=msg['addr'].toString();
8   var ssn=msg['ssn'].toString();
9   channelMap.put('pid',pid);
10  dbConn=DatabaseConnectionFactory.createDatabaseConnection('org.mariadb.jdbc.Driver', 'jdbc:mariadb://127.0.0.1:3306/mirthtestdb', 'root', '125353Ct');
11  var result=dbConn.executeCachedQuery("select count(*) from patient WHERE pid = '"+pid+"'");
12  result.next();
13  var icount = result.getInt(1);
14  if(icount==0)
15  {
16    var expression = "INSERT INTO patient (pid,name,sex,dob,addr,ssn) VALUES (?, ?, ?, ?, ?, ?)";
17    var paramlist=new Array(pid,name,sex,dob,addr,ssn);
18    icount = dbConn.executeUpdate(expression, paramlist);
19    channelMap.put("Ret","1");
20  }
21  else{
22    channelMap.put("Ret","0");
23  }
24  result.close();
25  finally {
26    if (dbConn) {
27      dbConn.close();
28    }
29  }
30 }

```

設置源端轉換規則;源端新建JS轉換步驟,在'輸入消息模板'录入如下XML消息模板并添加如下js脚本

```

<patient><pid>110</pid><name>撰獲濟</name><sex>M</sex><dob>19650108</dob><addr>臺灣大學</addr>
<ssn>33661489</ssn><status>0</status></patient>

```

```

var dbConn;
try {
  var pid=msg['pid'].toString();
  var name=msg['name'].toString();
  var sex=msg['sex'].toString();
  var dob=msg['dob'].toString();
  var addr=msg['addr'].toString();
  var ssn=msg['ssn'].toString();
  channelMap.put('pid',pid);
  dbConn=DatabaseConnectionFactory.createDatabaseConnection('org.mariadb.jdbc.Driver',
'jdbc:mariadb://127.0.0.1:3306/mirthtestdb', 'root', '125353Ct');
  var result=dbConn.executeCachedQuery("select count(*) from patient WHERE pid = '"+pid+"'");
  result.next();
}

```

```
var icount = result.getInt(1);
if(icount==0)
{
    var expression = "INSERT INTO patient (pid,name,sex,dob,addr,ssn) VALUES (?, ?, ?, ?, ?, ?)";
    var paramList=new Array(pid,name,sex,dob,addr,ssn);
    icount = dbConn.executeUpdate(expression, paramList);
    channelMap.put('Ret','1');
}else{
    channelMap.put('Ret','0');
}
result.close();
} finally {
    if (dbConn) {
        dbConn.close();
    }
}
```

https://localhost:8443 - Mirth Connect Administrator - (3.10.1)

Mirth Connect

Dashboard

Channels

Users

Settings

Alerts

Events

Extensions

Channel Tasks

Save Changes

Validate Connector

New Destination

Delete Destination

Clone Destination

Disable Destination

Edit Filter

Edit Transformer

Edit Response (1)

Import Connector

Export Connector

Export Channel

Deploy Channel

Other

Notifications

Edit Channel - HTTP Listener Channel

Summary | Source | Destinations | Scripts

Status	Destination
Enabled	Destination 1

Connector Type: Channel Writer ☐ Wait for previous destination

Destination Settings

Queue Messages: ☒ Never ☐ On Failure ☐ Always

Advanced Queue Settings: 0 Retries

Validate Response: ☐ Yes ☒ No

Reattach Attachments: ☒ Yes ☐ No

Channel Writer Settings

Channel Id: <None>

Message Metadata:

Template: \${message.encodedData}

设置目的端Channel Writer参数

Edit Channel - HTTP Listener Channel - Destination 1 Response Transformer

Enabled	#	Name	Type
<input checked="" type="checkbox"/>	0	Response XML	JavaScript

Step

Generated Script

1

tmp['pid']=\${'pid'};

2

tmp['ret']=\${'Ret'};

Reference | Message Trees | Message Templates

Inbound Message Template

Data Type: Raw

Outbound Message Template

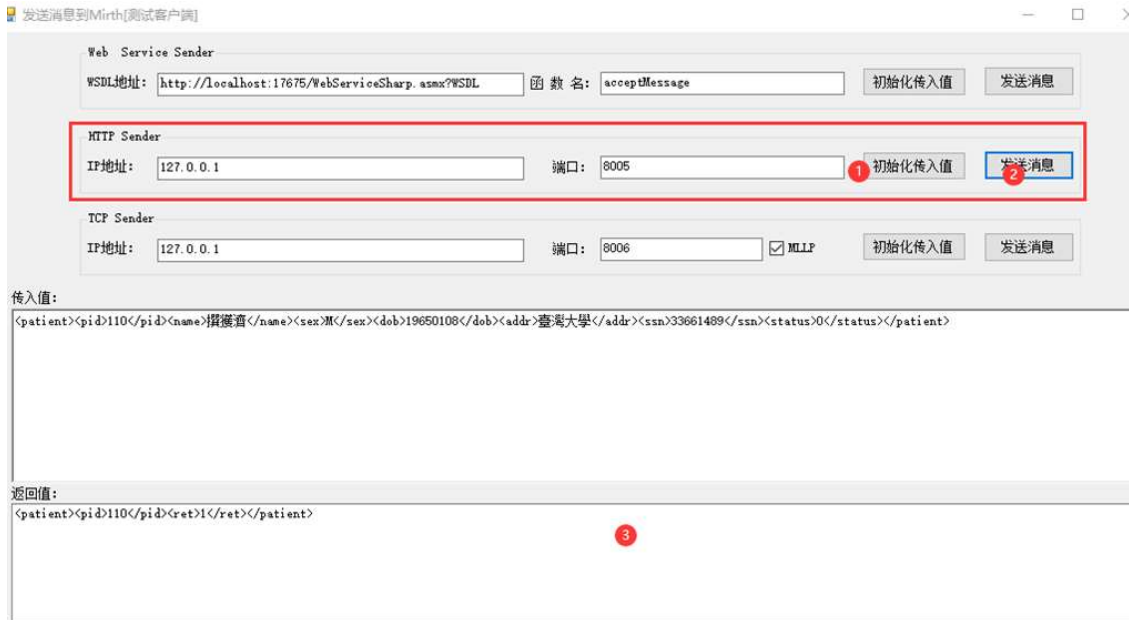
Data Type: XML

<patient>
<pid></pid>
<ret></ret>
</patient>

目的端响应转换规则; 在'输出消息模板'中录入如下xml格式模板并添加如下js脚本

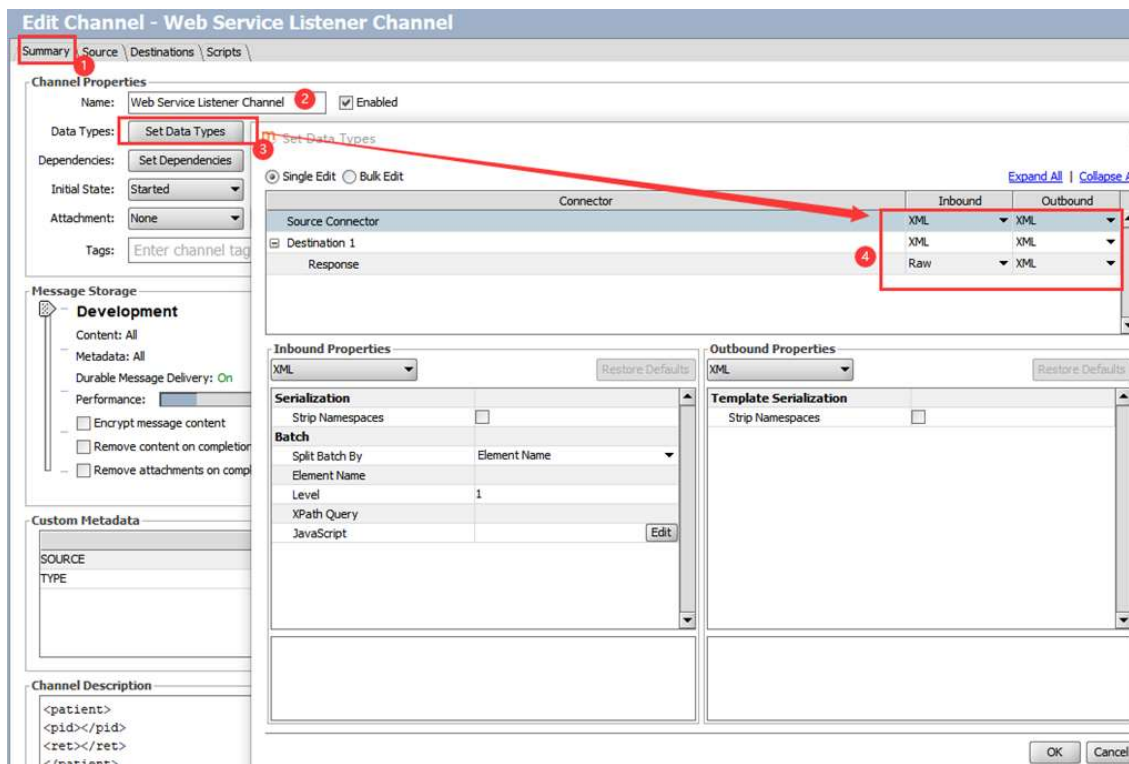
```
<patient>
<pid></pid>
<ret></ret>
</patient>

tmp['pid']=${'pid'};
tmp['ret']=${'Ret'};
```

运行C#版HTTP Sender，进行如上测试，成功获取响应，验证Mirth Connect 的HTTP Listener信道OK!

Web Service Listener 信道示例:



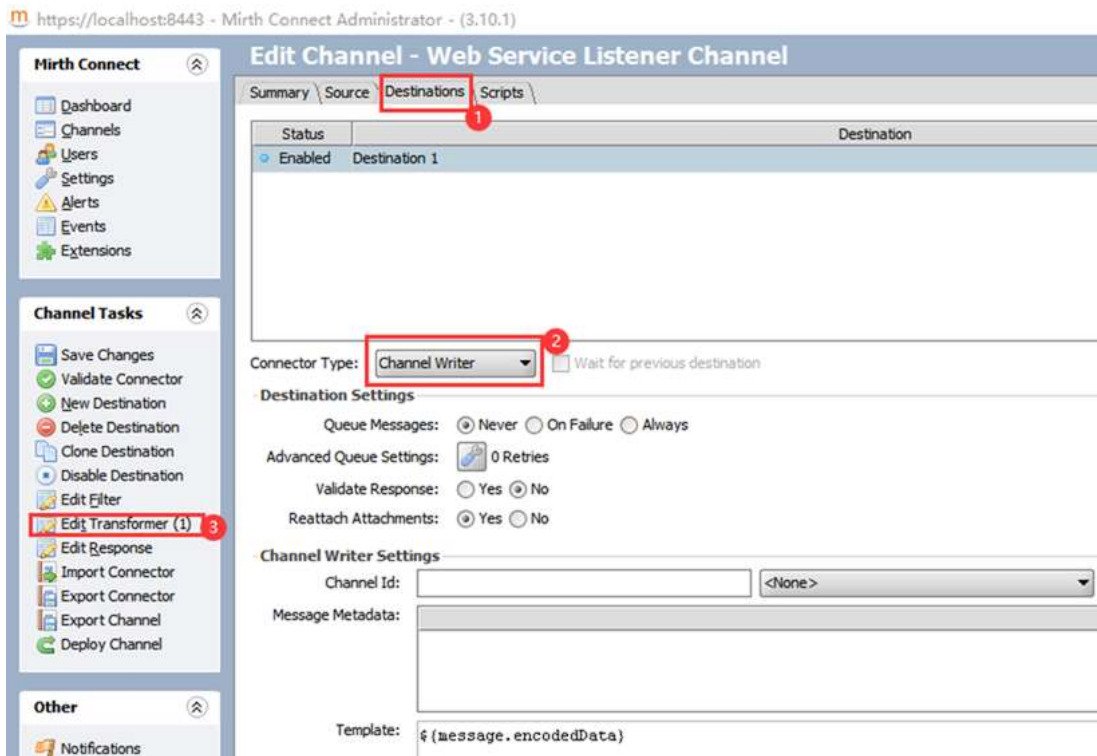
新建信道并设置源端输入输出参数格式、目的端输入输出格式、响应输入输出参数格式

https://localhost:8443 - Mirth Connect Administrator - (3.10.1)

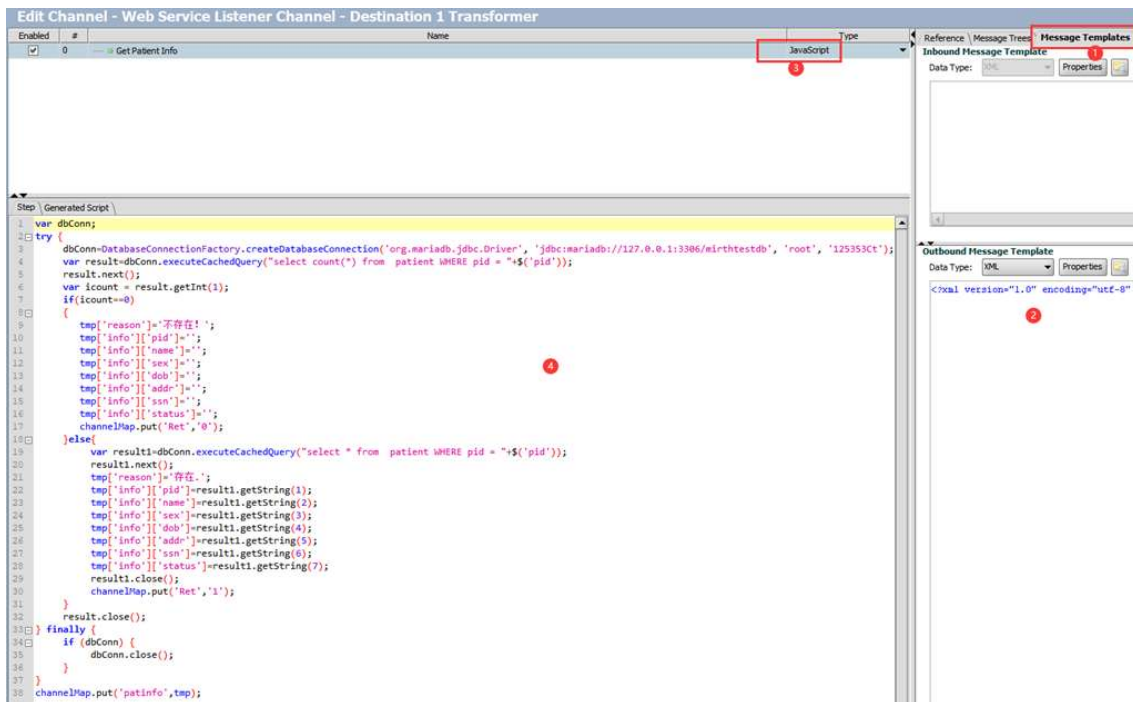
設置源端Web Service Listener參數

設置源端轉換規則;源端新建Mapper映射轉換步驟, 在'輸入消息模板'录入如下XML消息模板

```
<patient>
<pid></pid>
<ret></ret>
</patient>
```



設置目的端Channel Writer參數



目的端轉換規則：在'輸出消息模板'中輸入如下xml格式模板並添加如下js腳本

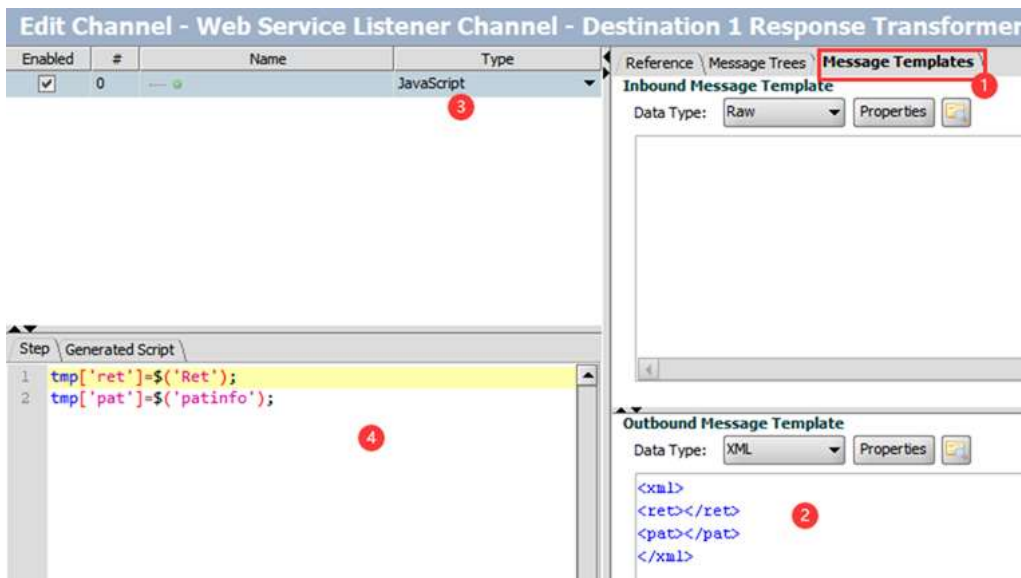
```
<?xml version="1.0" encoding="utf-8" ?><patient><reason></reason><info><pid>103</pid><name>Amarnath Zhao</name>
<sex>M</sex><dob>19910518</dob><addr>臺灣&apos;&lt;&quot;台中心中</addr><ssn>886</ssn><status>0</status></info>
</patient>
```

```
var dbConn;
try {
    dbConn=DatabaseConnectionFactory.createDatabaseConnection('org.mariadb.jdbc.Driver',
'jdbc:mariadb://127.0.0.1:3306/mirthtestdb', 'root', '125353ct');
    var result=dbConn.executeCachedQuery("select count(*) from patient WHERE pid = '"+${'pid'}");
    result.next();
    var icount = result.getInt(1);
    if(icount==0)
    {
        tmp['reason']='不存在! ';
        tmp['info']['pid']='';
        tmp['info']['name']='';
        tmp['info']['sex']='';
        tmp['info']['dob']='';
        tmp['info']['addr']='';
    }
}
```

```

tmp['info']['ssn']="";
tmp['info']['status']="";
channelMap.put('Ret','0');
}else{
    var result1=dbConn.executeCachedQuery("select * from patient WHERE pid = '"+$('pid')");
    result1.next();
    tmp['reason']='存在.';
    tmp['info']['pid']=result1.getString(1);
    tmp['info']['name']=result1.getString(2);
    tmp['info']['sex']=result1.getString(3);
    tmp['info']['dob']=result1.getString(4);
    tmp['info']['addr']=result1.getString(5);
    tmp['info']['ssn']=result1.getString(6);
    tmp['info']['status']=result1.getString(7);
    result1.close();
    channelMap.put('Ret','1');
}
result.close();
} finally {
    if (dbConn) {
        dbConn.close();
    }
}
}
channelMap.put('patinfo',tmp);

```



目的端响应转换规则；在‘输出消息模板’中录入如下xml格式模板并添加如下js脚本

```

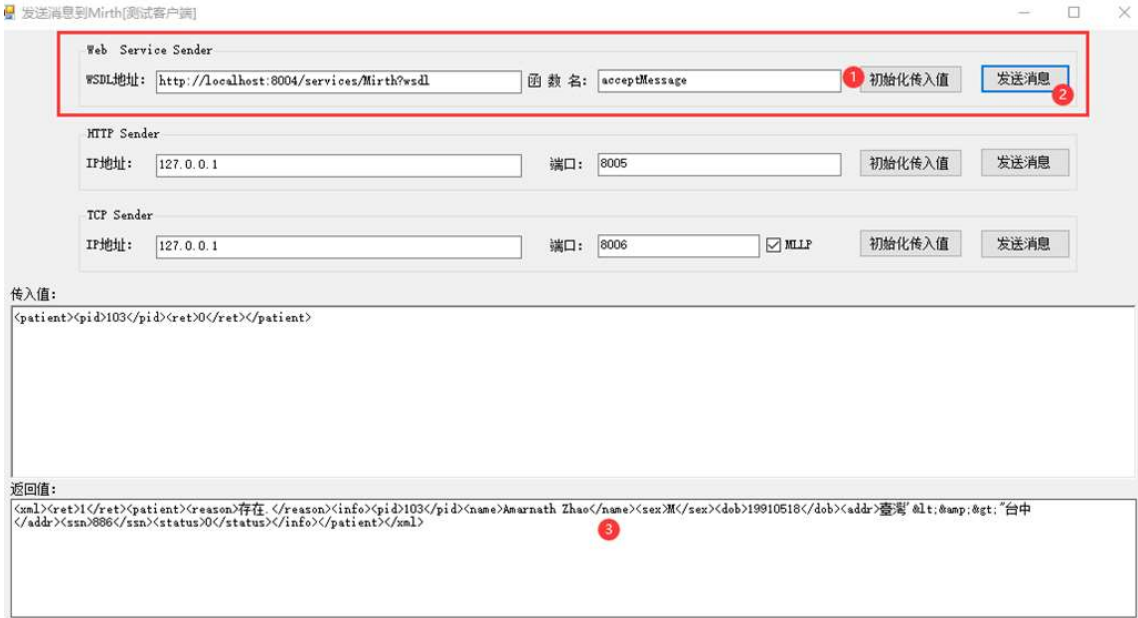
<xml>
<ret></ret>
<pat></pat>
</xml>

```

```

tmp['ret']=${'Ret'};
tmp['pat']=${'patinfo'};

```



运行C#版Web Service Sender，进行如上测试，成功获取响应，验证Mirth Connect 的Web Service Listener信道OK!

本课知识点总结：

- 1.TCP\HTTP\Web Service三种Listener的响应消息处理
- 2.Response均以目的端作为响应
- 3.数据库JS查询与插入编程

本课扩展思考：

请在目的端响应转换中写如下JS脚本responseMap.put('responseAck', ack.toString());然后在源端Response参数指定此作为响应。

潤沁網路大學

分类: [Mirth](#)

标签: [Mirth](#)

好文要顶

关注我

收藏该文

潤沁網路大學

关注 - 0

粉丝 - 2

0

0

- « 上一篇: [第五課-Channel Study TCP Listener & Web Service Listener](#)
- » 下一篇: [第七課-Channel Study For HTTP Listener & Web Service Sender Intercommunicates Response Handler](#)

posted @ 2021-01-24 15:31 潤沁網路大學 阅读(44) 评论(0) 编辑 收藏

[刷新评论](#) [刷新页面](#) [返回顶部](#)

发表评论

编辑 预览

B

支持 Markdown

自动补全

提交评论 退出

[Ctrl+Enter]快捷提交

- 【推荐】大型组态、工控、仿真、CAD\GIS 50万行VC++源码免费下载！
- 【推荐】亚马逊科技在线研讨会：借助图神经网络实现实时欺诈检测
- 【推荐】华为开发者联盟--邀友同注册，解锁阶梯“豪”礼
- 【推荐】限时秒杀！国云大数据魔镜，企业级云分析平台

- 园子动态：**
- 发起一个开源项目：博客引擎 fluss
 - 云计算之路-新篇章-出海记：开篇
 - 博客园2005年6月1日首页截图

- 最新新闻：**
- 黄峥勇退：一年之内卸任CEO和董事长 想去“寻找幸福”
 - 百度二次上市，三重价值
 - 快手三年游戏路，路在何处？
 - 谷歌涂鸦庆祝爱尔兰圣帕特里克节
 - NASA的SMA轮胎技术即将商用 30倍于钢的可恢复应变
- » 更多新闻...

历史上的今天：
2021-01-24 第六課-Channel Study For TCP Listener & HTTP Listener & Web Service Listener Abou...

Copyright © 2021 潤沁網路大學
Powered by .NET 5.0 on Kubernetes

潤沁網路大學