



AMQP との統合

概要

Advanced Message Queuing Protocol (AMQP) は、アプリケーション間や組織間でビジネスメッセージを渡すためのオープン標準です。Service Designer モジュールを使用して、AMQP タスクを定義することができます。AMQP タスクはメッセージブローカを介して外部システム (Process Orchestrator) にサービス要求をパブリッシュします。

AMQP はサービス要求を交換に送信し、交換は Service Catalog からのメッセージを受け入れて、それをメッセージキューにルーティングします。RabbitMQ は AMQP 標準を実装したオープンソースメッセージブローカソフトウェアです。外部システム (Process Orchestrator) は、メッセージを取得し、それら进行处理するために、必要な API を使用して RabbitMQ にアクセスします。

キュー サービス要求と呼ばれる新しいタスクタイプは Service Designer モジュールの [計画 (Plan)] タブで利用でき、サービス要求データをメッセージブローカにパブリッシュします。AMQP タスクを使用してデータをパブリッシュする方法の詳細については、『[Cisco Prime Service Catalog Designer Guide](#)』を参照してください。

メッセージキュー

交換はプロデューサアプリケーションからのメッセージを受け入れ、それをメッセージキューにルーティングします。事前、事後、および主要 AMQP タスクのために作成される交換はすべて「ファンアウト」タイプで、各タスクについてデフォルト キューが作成され、それらのキューが各タスクにバインドされます。デフォルト キューの名前は <topic-name>_queue です。トピック名は「キュー サービス要求」タスクに対してユーザが [Service Designer] > [計画 (Plan)] タブで入力した名前です。



(注)

プードル攻撃を防ぐために、RabbitMQ サーバとの Prime Service Catalog 統合は SSL プロトコル TLSv1.2 バージョンのみをサポートしています。TLSv 1.2 が指定されていない場合、クライアントは RabbitMQ サーバに接続してメッセージを利用することができず、次の例外が発生して接続に失敗します。javax.net.ssl.SSLException: Received fatal alert: protocol_version

交換とキューを作成し、メッセージを利用するためのサンプルコードは次のようになります。

```
package amqpProject;

import com.rabbitmq.client.ConnectionFactory;
import com.rabbitmq.client.Connection;
import com.rabbitmq.client.Channel;
import com.rabbitmq.client.QueueingConsumer;

public class SampleProcess {

    public static void main(String[] argv) throws Exception {
        String exchangeName = "testExchange";
        String queueName = exchangeName+"_queue";
        String brokerIpAddress = "10.142.10.77";
        String userName = "admin";
        String password = "cisco123";
        ConnectionFactory factory = new ConnectionFactory();
        factory.useSslProtocol("TLSv1.2");!--Included to prevent Poodle attack
        factory.setHost(brokerIpAddress);
        factory.setUsername(userName);
        factory.setPassword(password);

        Connection connection = factory.newConnection();

        Channel channel = connection.createChannel();
        channel.exchangeDeclare(exchangeName, "fanout", true, false, null);

        channel.queueDeclare(queueName, true, false, false, null);
        channel.queueBind(queueName, exchangeName, "");

        QueueingConsumer consumer = new QueueingConsumer(channel);
        channel.basicConsume(queueName, true, consumer);
        while (true) {

            QueueingConsumer.Delivery delivery = consumer.nextDelivery();
            String message = new String(delivery.getBody());
            System.out.println(" [x] Received from "+queueName+":-");
            System.out.println(message);

        }
    }
}
```

REST-based nsAPIs

サービスおよびそのタスクに関するメッセージブローカの詳細を返すための REST API セットがあります。これにより返される署名および応答を以下に示します。必要な入力値はパスパラメータとして受け入れられるサービス名です。交換の詳細を含む、そのサービスに関するすべての AMQP タスクが発信者に返されます。

サービス固有の API のサンプル応答は次のようになります。

http://localhost:8088/RequestCenter/nsapi/messagebroker/service/<service name>

```
{ "tasks": [ { "name": "PRE:amqpTask1", "connectionIdentifier": "am1", "messageFormat": "JSON", "exchangeName": "AQAB_Pre", "payloadType": "All Message Details (large)", "isAutoComplete": "true", "transformationType": "JOLT", "outboundTransformationName": "", "inboundTransformationName": "" }, { "name": "amqpTask1", "connectionIdentifier": "am1", "messageFormat": "JSON", "exchangeName": "post_exchange_0307", "payloadType": "All Message Details (large)", "isAutoComplete": "true", "transformationType": "JOLT", "outboundTransformationName": "", "inboundTransformationName": "" }, { "name": "POST:amqpTask1", "connectionIdentifier": "am1", "messageFormat": "JSON", "exchangeName": "AQAB_Post", "payloadType": "All Message Details (large)", "isAutoComplete": "false", "transformationType": "JOLT", "outboundTransformationName": "", "inboundTransformationName": "" } ] }
```

Overview API

Overview API は、Service Designer モジュール内で設定されたすべての交換を収集し、RabbitMQ サーバにすでに作成されている交換とキューを検出するためのコールを行います。Overview API は、AMQP サーバに対してルックアップを実行し、キューに関する情報を取得します。出力は JSON 形式で送信されます。

Overview API と noCache パラメータの使用:

- UI を使用して RabbitMQ 情報を指定し、noCache パラメータなしで Overview API を使用すると、情報がキャッシュとデータベースの両方から取得されます。
- ただし、データベースに直接アクセスして RabbitMQ 情報を挿入する場合は、確実に最新情報をフェッチするために、noCache =1 を指定して Overview API を使用する必要があります。

RabbitMQ サーバからの応答例は次のようになります。

http://localhost:8088/RequestCenter/nsapi/messagebroker/overview

```
{ "connections": [ { "connectionIdentifier": "am1", "host": "10.78.0.247", "port": 5671 }, { "connectionIdentifier": "am2", "host": "10.78.0.247", "port": 5671 }, { "connectionIdentifier": "am3", "host": "10.78.0.247", "port": 5671 }, { "connectionIdentifier": "AM4", "host": "10.78.0.247", "port": 5671 }, { "connectionIdentifier": "am5", "host": "10.78.0.247", "port": 5671 } ] }
```

http://localhost:8080/RequestCenter/nsapi/messagebroker/overview?connectionIdentifier=am1

```
{ "rabbitmq_version": "3.6.1", "username": "admin", "password": "UhP0zDGRoC2R7isv9qCQ6A==", "ipAddress": "10.78.0.247", "recoveryInterval": 300000, "vhost": "/", "inboundQueue": "psc_inbound_queue", "authorization_key": "owL7ViRfE4Sce0aG1jSzZInkIVw5CsM7acQ5r1swpXzF/kForoUj1frVUUouA+CqSFSTuJLlQ5GRUMmzzxbV2xtMvhGed6x1WU08MbDLKyDSSY5UQvoAS7aZ0dR0dXvF+G4uD4nA1Q6HSIc6dBct3M+dDJcm02z9OshvMacmKvZa380B8/MbbBhu5Q3FntzWkAVY/FobU9gvlidoDt1Ty0CmAgfvPbP6joLXYaTPHjaqjYaBaX2Y4m+1V7Wm3Rb+oLpHZkCVm7Pr1z1LByPs6d+qaQShIfr0yeXXbZrQNh3s8qH1+YbYFRtNin/JEeLa6pY5J19m6zUV78n2BstRQ==", "message-time-to-live": "300000", "useSSL": true, "port": "5671", "exchanges": [ { "name": "exchange2", "vhost": "/", "type": "fanout", "created": "false" }, { "name": "HeatStack", "vhost": "/", "type": "fanout", "created": "true" }, { "name": "exchange3", "vhost": "/" }
```

```

": "/" , "type": "fanout", "created": "false"}], "queues": [{"name": "exchange2_queue", "vhost": "/" , "created": "false"}, {"name": "HeatStack_queue", "vhost": "/" , "created": "true"}, {"name": "exchange3_queue", "vhost": "/" , "created": "false"}], "certificate": "-----BEGIN CERTIFICATE-----\nMIIC5DCCAcygAwIBAgIBATANBgkqhkiG9w0BAQsFADATMREwDwYDVQDDAhNeVr1c3RDQTAeFw0x\nnNjA2MTEwMTUwMTJhFw0xNzA2MTEwMTUwMTJhMCCxG9zdg5hbWUpMQ8wDQYD\nVQKDAZzZXJ2ZXIwggEiMA0GCSqGSIb3DQEBAQUAA4IBDwAwggEKAoIBAQcQ8v2yJT7tv+nOwFSo\n\nEE1c0oVg3skd86JN7jVJaz/mMOyJjDmf1147iUwZPMTTCB34ovYUXFkw+a+0ext2WRHgQLTMPvVO\n\nA86jwuPd/bhUxXg8jeEfe4V/1Seci9Xz+5VxqCybCNOzJQ12/vLXvIJK43U/+1GdXnpWxFaf0yd0\n\nnht3iUy6mfUAHFNM12SofJwbbdUaMyD0/Krsiu+X+vFQBUDmM7Y0priItiDvDq7rxug2UOPACzZMG\n\nn5yJ5aJjNLSlJRwKKst/jjvesqHIGWNo0qKvkTET3tIVsKDi1Fn9IdKQuoI1n225+58cWSANmZ5M\n\nn4BdSIf4z6QRuKliBri55AgMBAAGjLzAtMAKGA1UdEwQCMAAwCwYDVR0PBAQDAGUgMBMGA1UdJQM\n\nnMAoGCCsGAQUFBwMBMA0GCSqGSIb3DQEBwUAA4IBAQC4L9fk4kuu/dpLeFWNuhb5Uyk6AqbTRAYD\n\nn1q1ml1E09EhmTH7cnoFsz0ELBdpASIUADsb9jmUeNkKJtjW94gq81uSem1Z81QzUCOCE6rsaznrw\n\nnm9jJO7gXA5SSmy7PgdokdhbeTz1SYA3kkkR5ZE8M403Qv5cEPREqnsHs6f6bQSm8tzSNETy3OyHL\n\n\nnpUo7YVTBCfJMq/e2nZxCJSduEL6QaIL4kmEYeu8j/1RplBAofMkDfDe+yMx2MJYl+mopVggGexpa\n\n\nmQybCchrDTJ/8sI/R18Ld80TQ2Km70sQqvNVenCpGctgGicupRwA0psQH0PNaUK+1cwYRivf0VS\n\n\nn09QE\n\n\n-----END CERTIFICATE--"}

```

パブリック キー GUID を使用してクレデンシャルを暗号化する

Prime Service Catalog は、nsAPI に渡されるパブリック キーの GUID を使用することにより、安全に AMQP クレデンシャルを返す方法をサポートしています。

外部システムのパブリック キー GUID がクエリ パラメータとして渡された場合、応答内のクレデンシャルは GUID に関連付けられているパブリック キーにより暗号化されます (PO のセキュリティ ストリング形式を使用)。

外部システムのパブリック キー GUID が渡されない場合は、DB で暗号化された文字列がそのまま返されます (注: 外部システムではこれを復号化できません)。

暗号化された機密データは、以下に示す AMQP Overview API で返されます。

```
http://localhost:8088/RequestCenter/nsapi/messagebroker/overview?publicKeyGUID=...
```

暗号化は、接続ごとに設定された公開キーを使用して実行されます。

認証キー生成 API

create、update、および delete などのサービス項目操作では channel-id よりも認証キーのほうが重要です。しかし、外部のタスクを一意に識別するため、メッセージ内には channel-id も必要です。以下の nsAPI を使用して作成される認証キーは、ユーザに固有です。認証キーは、上述の nsAPI が呼び出されるたびに値が変更されるように見えますが、特定のユーザのすべての AMQP 接続に対して同じです。このキーは、ユーザを認証するため、また、各ユーザに与えられた許可に従ってさまざまなモジュールにアクセスするために使用されます。

認証キーは、以下の nsAPI から取得できます。

```
http://<ServerURL>/RequestCenter/nsapi/messagebroker/overview?connectionIdentifier=<particular amqp connection identifier>
```

```

{"rabbitmq_version": "3.6.1", "username": "admin", "password": "UhP0zDGRoC2R7isv9qCQ6A==", "ipAddress": "10.78.0.247", "recoveryInterval": 300000, "vhost": "/", "inboundQueue": "psc_inbound_queue", "authorization_key": "owL7ViRfE4Sce0aG1jSzZInkIVw5CsM7acQ5rlswpXzF/kForoUj1frVVUOuA+CqSFStuJ1LQ5GRUMmzxbV2xtrMvhGed6x1WU08MbDLKyDSSY5UQvoAS7az0dR0dXvF+G4u4nAlQ6HSIc6dBct3M+ddJcm02z9OshvMacmkvZa380B8/MbbBhu5Q3FntzWkAVY/FobU9gvlidoDt1Ty0CmAgfvPbP6joLXYaTPHjaqjYaBaX2Y4m+1v7Wm3Rb+oLpHZkCvm7Pr1z1LByPs6d+qaQShIfroYeXXbZrQNh3s8qH1+YbYFRtNin/JEeLa6pY5J19m6zUV78n2BstRQ=", "message-time-to-live": "300000", "useSSL": true, "port": "5671", "exchanges": [{"name": "exchange2", "vhost": "/", "type": "fanout", "created": "false"}, {"name": "HeatStack", "vhost": "

```


JOLT を使用した JSON 変換の例

JSON 着信 AMQP の入力:

```
{
  "message": {
    "updateData": {
      "dataValue": [
        {
          "namePO": "TextField.Text",
          "valuePO": [
            "QoE"
          ],
          "multiValuedPO": false
        }
      ]
    },
    "addCommentsPO": {
      "comment": [
        "test comment 1",
        "test comment 2",
        "test comment 3"
      ]
    },
    "takeAction": {
      "actionPO": "DONE"
    },
    "channelId": "51515F28-E36A-478C-8773-E35B215CF36C"
  }
}
```

JOLT 変換の仕様:

```
[
  {
    "operation": "shift",
    "spec": {
      "message": {
        "updateData": {
          "dataValue": {
            "**": {
              "namePO": "message.updateData.dataValue.[&1].name",
              "valuePO": "message.updateData.dataValue.[&1].value",
              "multiValuedPO": "message.updateData.dataValue.[&1].multiValued"
            }
          }
        },
        "**": "&1.&0",
        "channelId": "message.channelId"
      }
    }
  },
  {
    "operation": "shift",
    "spec": {
      "message": {
        "takeAction": {
          "actionPO": "message.&1.action"
        },
        "**": "&1.&0",
      }
    }
  }
]
```

```

        "channelId": "message.channelId"
      }
    },
    {
      "operation": "shift",
      "spec": {
        "message": {
          "addCommentsPO": "message.addComments",
          "**": "&1.&0",
          "channelId": "message.channelId"
        }
      }
    }
  ]
}

```

JSON 出力:

次は、上記の変換および入力に基づいて Prime Service Catalog 着信 AMQP コードによって生成された出力で、最終的に処理されたものです。

```

{
  "message" : {
    "addComments" : {
      "comment" : [ "test comment 1", "test comment 2", "test comment 3" ]
    },
    "channelId" : "51515F28-E36A-478C-8773-E35B215CF36C",
    "takeAction" : {
      "action" : "DONE"
    },
    "updateData" : {
      "dataValue" : [ {
        "multiValued" : false,
        "name" : "TextField.Text",
        "value" : [ "QoE" ]
      } ]
    }
  }
}

```

AMQP 着信 XML の例

```

<message channel-id="30FCC75D-83A0-4DD9-E050-007F01019778">
  <update-data>
    <data-value multi-valued="false">
      <name>amqpDict1.field1</name>
      <value>a_newer</value>
    </data-value>
  </update-data>
</message>

```

```

<message channel-id="30FCC75D-83A0-4DD9-E050-007F01019778">
  <add-comments>
    <comment>testing new comments</comment>
  </add-comments>
</message>

```

```

<message channel-id="30FCC75D-83A0-4DD9-E050-007F01019778">
  <take-action action="done">
  </take-action>
</message>

```

AMQP 着信 JSON の例

```
{
  "message" : {
    "addComments" : {
      "comment" : [ "test comment 1", "test comment 2", "test comment 3" ]
    },
    "channelId" : "51515F28-E36A-478C-8773-E35B215CF36C",
    "takeAction" : {
      "action" : "DONE"
    },
    "updateData" : {
      "dataValue" : [ {
        "multiValued" : false,
        "name" : "TextField.Text",
        "value" : [ "QoE" ]
      } ]
    }
  }
}
```

着信メッセージ

他社製システムの要求操作およびサービス項目操作からの着信メッセージについては、2種類の操作がサポートされています。

nsXML において最も重要な要素は **channel-id** です。これは、外部タスクを一意に識別する ID です。この ID は他社製システムに提供され、対応するデータアップデートがビジネス エンジンによって正常に適用されるためには、応答に表示される必要があります。

AMQP での要求操作

AMQP でサポートされている要求操作は、Service Link の要求操作と同様です。ただし、AMQP 着信メッセージでは、XML および JSON 形式がサポートされています。各操作の詳細については、[着信 nsXML メッセージ \(5-27 ページ\)](#) を参照してください。

take-action

take-action 操作は、配信タスクを完了としてマークします。

```
<message channel-id="30FCC75D-83A0-4DD9-E050-007F01019778">
  <take-action action="done">
</take-action>
</message>
```

update-data

要求のディクショナリ フィールドのデータは、新しい値に更新されます。

```
<message channel-id="30FCC75D-83A0-4DD9-E050-007F01019778">
  <update-data>
  <data-value multi-valued="false">
  <name>amqpDict1.field1</name>
  <value>a_newer</value>
  </data-value>
  </update-data>
</message>
```


add-comment

add-comments メッセージは、要求の [System Comments] セクションにコメントを追加するために使用します。

```
<message channel-id="30FCC75D-83A0-4DD9-E050-007F01019778">
<add-comments>
<comment>testing new comments</comment>
</add-comments>
</message>
```

AMQP でのサービス項目操作

(注) 変換は、サービス項目操作ではサポートされていません。

サービス項目操作では、認証キーを生成してメッセージに含める必要があります。認証キーを生成するには、次の nsAPI を使用します。

```
http://<ServerURL>
/RequestCenter/nsapi/messagebroker/overview?connectionIdentifier=<particular amqp
connection identifier>
```

この API の詳細については、[認証キー生成 API \(4-4 ページ\)](#) を参照してください。次は、AMQP でのサービス項目操作のサンプルです。

作成

create メッセージでは、サービス項目に属性値が追加されます。

```
<message channel-id="A984F860-DBE7-48EB-B3A2-F6A0159B093F"
authorization-key="liDBm1NeRVbVmwbBmAd+cca0/Z8jf863aYOK13QXSWcYwn5aGuPxn09CjDzv5EKR7/CcJx5
+2gulMhUopAuiX3WwjX/DP1Xw3nmFLMz6dmai iq0+5v4XOKmNqf1J3GvBJYjTxAkH1VbCQ2y1fNxE8/cP5wyd3mU6M
LlD9tjc/Iro950gLlTq+9C2/QMis6ya5202D8F652jnHnWbZrDf6zPdOS1FpCKkg05YSVpqi fGgqjEh3RTyPs9w0Qc
NcxWcEqD1vvBMNW0VAL/YaW+MNmoUnpg1R0cUeOJ3WFr8/11/uyNyhfVAYfDUznVCNfVDtkMzCWhA4eH25XQutmKwG
w==">
<create>
<serviceitem>
<name>AnandServiceItem</name>
<serviceItemData>
<serviceItemAttribute name="Name">Hannah</serviceItemAttribute>
<serviceItemAttribute name="DOB">1995-01-19</serviceItemAttribute>
<serviceItemAttribute name="Age">25</serviceItemAttribute>
</serviceItemData>
</serviceitem>
</create>
</message>
```

update

更新メッセージでは、サービス項目属性を省略すると、属性値は変更されません。メッセージで属性が明示的に指定されているものの、値が含まれない場合、そのサービス項目の属性値はテキストフィールドでは空白に設定され、数値フィールドでは 0 に設定されます。

```
<message channel-id="A984F860-DBE7-48EB-B3A2-F6A0159B093F"
authorization-key="liDBm1NeRVbVmwbBmAd+cca0/Z8jf863aYOK13QXSWcYwn5aGuPxn09CjDzv5EKR7/CcJx5
+2gulMhUopAuiX3WwjX/DP1Xw3nmFLMz6dmai iq0+5v4XOKmNqf1J3GvBJYjTxAkH1VbCQ2y1fNxE8/cP5wyd3mU6M
LlD9tjc/Iro950gLlTq+9C2/QMis6ya5202D8F652jnHnWbZrDf6zPdOS1FpCKkg05YSVpqi fGgqjEh3RTyPs9w0Qc
NcxWcEqD1vvBMNW0VAL/YaW+MNmoUnpg1R0cUeOJ3WFr8/11/uyNyhfVAYfDUznVCNfVDtkMzCWhA4eH25XQutmKwG
w==">
<update>
<serviceitem>
<name>AnandServiceItem</name>
```

```

<serviceItemData>
<serviceItemAttribute name="Name">Anand2update</serviceItemAttribute>
<serviceItemAttribute name="RAM">Primarymemory3</serviceItemAttribute>
<serviceItemAttribute name="MemoryInt">759001</serviceItemAttribute>
<serviceItemAttribute name="Model512">Lenovo T440 updated</serviceItemAttribute>
<serviceItemAttribute name="Money">80000</serviceItemAttribute>
<serviceItemAttribute name="ManufTime">2016-05-01 16:45</serviceItemAttribute>
</serviceItemData>
</serviceitem>
</update>

```

削除

delete サービス項目要求に必要なのは、サービス項目タイプおよびインスタンスの名前だけです。その他のサービス項目の属性およびサブスクリプション情報は無視されます。

```

<message channel-id="A984F860-DBE7-48EB-B3A2-F6A0159B093F"
authorization-key="liDBm1NeRVbVmwbBmAd+cca0/Z8jf863aYOK13QXSWcYwn5aGuPxn09CjDzv5EKR7/CcJx5
+2gulMhUopAuiX3WwjX/DP1Xw3nmFLMz6dmaiIQ0+5v4XOKmNqf1J3GvBJYjTxAKh1VbCQ2y1fNxE8/cP5wyd3mU6M
Lld9tjc/Iro950gLlTq+9C2/QMis6ya5202D8F652jnHnWbZrDf6zPdOS1FpCKkg05YSVpqi fGgqjEh3RTyPs9w0Qc
NcxWcEqD1vvBMNW0VAL/YaW+MNmoUnpglR0cUeOJ3WFr8/1l/uyNyhfVAYfDUznVCnfVdtkMzCWhA4eH25XQutmKwG
w==">
  <delete>
    <serviceitem>
      <name>ServiceItem</name>
      <serviceItemData>
        <serviceItemAttribute name="Name">Anand3</serviceItemAttribute>
      </serviceItemData>
    </serviceitem>
  </delete>
</message>

```

階層

hierarchy メッセージでは、属性値がサービス項目にネストされます。

発信メッセージ

パブリッシュされるメッセージ形式は、Service Link で現在使用されている形式と同様な、nsXML、JSON、または FTL 形式です。

```

{
  "message": {
    "create": {
      "hierarchy": true,
      "serviceitem": [ {
        "name": "Computer",
        "serviceItemData": [ {
          "Name": "Dell-HP Computer",
          "OS": "Linux 7",
          "RAM": "7878",
          "HD": "520",
          "Mouse": {

```

```

        "Name": "HPE"
    },
    "Network": {
        "Name": "My Mac Network",
        "IPAddress": "3.3.3.3",
        "MACAddress": "12.12.12.12",
        "AdapterType": "MAC"
    }
}
    ]]
}
    ],
    "channelId": "6A55AF3D-4EEC-402C-9055-CB79081C138F"
}
}

```

XML 発信メッセージの例

nsXml データ構造の例を以下に示します。

- データのメッセージ構造例、サービス詳細なし(デフォルト、小)

```

<?xml version="1.0" encoding="UTF-8"?>
<message channel-id="F203ACC7-E6CA-A2EA-E040-007F0101140E"
xmlns:fo="http://www.w3.org/1999/XSL/Format">
<task-started task-type="task">
<task>
<actual-duration>0.0</actual-duration>
<completed-date/>
<context-id>69</context-id>
<context-type>Requisition Entry</context-type>
<due-date>2014-04-25 20:00:00</due-date>
<effort>10.0</effort>
<estimated-date/>
<expected-duration>10.0</expected-duration>
<flag-id>0</flag-id>
<is-sharable>true</is-sharable>
<is-shared>true</is-shared>
<next-action-id>2</next-action-id>
<performer-actual-duration>0.0</performer-actual-duration>
<priority>2</priority>
<scheduled-start-date>2014-04-24 18:00:00</scheduled-start-date>
<start-date>2014-04-23 14:11:13</start-date>
<state-id>2</state-id>
<subject>queueServiceRequest1</subject>
<task-id>266</task-id>
</task>
<requisition>
<services>0</services>
<actual-cost>0.0</actual-cost>
<actual-duration>0.0</actual-duration>
<closed-on/>
<customer>

```

```

<company-address/>
<email>internal@newscale.com</email>
<fax/>
<first-name>admin</first-name>
<home-ou>
<name>&lt;s ID=&quot;847&quot;/></name>
<organizational-unit-id>1</organizational-unit-id>
</home-ou>
<home-phone/>
<last-name>admin</last-name>
<login-name>admin</login-name>
<person-id>1</person-id>
<personal-address/>
<supervisor-name/>
<timezone>Pacific Standard Time</timezone>
<work-phone/>
</customer>
<due-on>2014-04-25 20:00:00</due-on>
<expected-cost>0.0</expected-cost>
<expected-duration>0.0</expected-duration>
<external>>false</external>
<initiator>
<company-address/>
<email>internal@newscale.com</email>
<fax/>
<first-name>admin</first-name>
<home-ou>
<name>&lt;s ID=&quot;847&quot;/></name>
<organizational-unit-id>1</organizational-unit-id>
</home-ou>
<home-phone/>
<last-name>admin</last-name>
<login-name>admin</login-name>
<person-id>1</person-id>
<personal-address/>
<supervisor-name/>
<timezone>Pacific Standard Time</timezone>
<work-phone/>
</initiator>
<organizational-unit>
<name>&lt;s ID=&quot;847&quot;/></name>
<organizational-unit-id>1</organizational-unit-id>
</organizational-unit>
<requisition-entry>
<closed-date/>
<data-values>
<data-value multi-valued="false">
<name>amqpDict1.field1</name>
<value>a</value>
</data-value>
<data-value multi-valued="false">
<name>amqpDict1.field2</name>
<value>b</value>
</data-value>
<data-value is-secure="true" multi-valued="false">
<name>amqpDict1.field3</name>
<value>ut3u4RC699wz7Jd20+4cix23m9XXgAC/EsDiHp1ERsOpdKSKdrtgMjFWcVo096aCFSkCgwa2tkBo
vKoOHzaillNiJ47+aY8zCWKwd17tCjmMLEkeQ1TvrDvIHR/DT1iWSmN08DI9+N57hY3A/g6ijUoM
gcuMczH+5F/pGtgupLZF/L7FwOwu4VcKVWM/2N5tuXGz+1aHTRAAAABYQXr/yD/75Ysy57LnQLxc
</value>
</data-value>
<data-value is-secure="true" multi-valued="false">
<name>amqpDict1.field4</name>
<value>ut3u4RC699wz7Jd20+4cix23m9XXgABReynDp5AdBgRi5ivhS05Unv98BgWgxc5YNcZrCihhhH/1

```

```

rzZqhjiIjAoRelIjADDb3IQP72armXsLRTvh0/fusd4jLdVIm4q1s+GaStt6F3oqQeZ4RLhVUopo
p2zNAXmjGj629C8gWREes3Z8EjDvXg5K1YVi90fXJV1jDoADhAAAABPlhct5TNV2d8R1cN/qDtK
</value>
</data-value>
</data-values>
<due-date>2014-04-25 20:00:00</due-date>
<item-number>1</item-number>
<price-per-unit>0.0</price-per-unit>
<priced>true</priced>
<quantity>1</quantity>
<rejected>false</rejected>
<rejected-date/>
<requisition-entry-id>69</requisition-entry-id>
<revision-number>105</revision-number>
<service>
<estimated-cost>0.0</estimated-cost>
<name>queueService1</name>
<parameters>
<default-duration>0.0</default-duration>
<priority>2</priority>
<start-date/>
<start-mode>0</start-mode>
</parameters>
<pricing-schema>0</pricing-schema>
<quantity>1</quantity>
<service-id>2</service-id>
<version>105</version>
<standard-duration>0.0</standard-duration>
</service>
<start-after/>
<start-date>2014-04-23 14:10:48</start-date>
<start-mode>0</start-mode>
<status>1</status>
</requisition-entry>
<requisition-id>64</requisition-id>
<started-on>2014-04-23 14:10:47</started-on>
<status>1</status>
</requisition>
<context>
<requisitionentryref itemnumber="1"/>
</context>
</task-started>
</message>

```

- セキュリティストリングを持つメッセージ構造例を次に示します。

```

<?xml version="1.0" encoding="UTF-8"?>
<message channel-id="F203ACC7-E6CA-A2EA-E040-007F0101140E"
xmlns:fo="http://www.w3.org/1999/XSL/Format">
<task-started task-type="task">
<task>
<actual-duration>0.0</actual-duration>
<completed-date/>
<context-id>69</context-id>
<context-type>Requisition Entry</context-type>
<due-date>2014-04-25 20:00:00</due-date>
<effort>10.0</effort>
<estimated-date/>
<expected-duration>10.0</expected-duration>
<flag-id>0</flag-id>
<is-sharable>true</is-sharable>
<is-shared>true</is-shared>
<next-action-id>2</next-action-id>
<performer-actual-duration>0.0</performer-actual-duration>

```

```

<priority>2</priority>
<scheduled-start-date>2014-04-24 18:00:00</scheduled-start-date>
<start-date>2014-04-23 14:11:13</start-date>
<state-id>2</state-id>
<subject>queueServiceRequest1</subject>
<task-id>266</task-id>
</task>
<requisition>
<services>0</services>
<actual-cost>0.0</actual-cost>
<actual-duration>0.0</actual-duration>
<closed-on/>
<customer>
<company-address/>
<email>internal@newscale.com</email>
<fax/>
<first-name>admin</first-name>
<home-ou>
<name>&lt;s ID=&quot;847&quot;/></name>
<organizational-unit-id>1</organizational-unit-id>
</home-ou>
<home-phone/>
<last-name>admin</last-name>
<login-name>admin</login-name>
<person-id>1</person-id>
<personal-address/>
<supervisor-name/>
<timezone>Pacific Standard Time</timezone>
<work-phone/>
</customer>
<due-on>2014-04-25 20:00:00</due-on>
<expected-cost>0.0</expected-cost>
<expected-duration>0.0</expected-duration>
<external>>false</external>
<initiator>
<company-address/>
<email>internal@newscale.com</email>
<fax/>
<first-name>admin</first-name>
<home-ou>
<name>&lt;s ID=&quot;847&quot;/></name>
<organizational-unit-id>1</organizational-unit-id>
</home-ou>
<home-phone/>
<last-name>admin</last-name>
<login-name>admin</login-name>
<person-id>1</person-id>
<personal-address/>
<supervisor-name/>
<timezone>Pacific Standard Time</timezone>
<work-phone/>
</initiator>
<organizational-unit>
<name>&lt;s ID=&quot;847&quot;/></name>
<organizational-unit-id>1</organizational-unit-id>
</organizational-unit>
<requisition-entry>
<closed-date/>
<data-values>
<data-value multi-valued="false">
<name>amqpDict1.field1</name>
<value>a</value>
</data-value>
<data-value multi-valued="false">

```

```

<name>amqpDict1.field2</name>
<value>b</value>
</data-value>
<data-value is-secure="true" multi-valued="false">
<name>amqpDict1.field3</name>
<value>ut3u4RC699wz7Jd20+4cix23m9XXgAC/EsDiHp1ERsOpdKSKdrtgMjFwcVo096aCFSkCgwa2tkBo
vKoOHzaillNiJ47+aY8zCWKwd17tCjmMLEkeQ1TvrDvIHR/DT1iWSmN08DI9+Ns7hY3A/g6ijUoM
gcuMczH+5F/pGtgupLZF/L7FwOwu4VcKVWM/2N5tuXGz+1aHTRAAAByQXr/yD/75Ysy57LnQLxc
</value>
</data-value>
<data-value is-secure="true" multi-valued="false">
<name>amqpDict1.field4</name>
<value>ut3u4RC699wz7Jd20+4cix23m9XXgABReynDp5AdBgRi5ivhS05Unv98BgWgxc5YNcZrCihhhH/1
rzZqhjiIjAoRelIjADDb3IQP72armXsLRTvh0/fusd4jLdVIm4q1s+GaSTt6F3oqQeZ4RlhVUopo
p2zNAXmjaGj629C8gWREes3Z8EjDvXg5K1YVi90fXJV1jDoADhAAAABP1hct5TNV2d8R1cN/qDtK
</value>
</data-value>
</data-values>
<due-date>2014-04-25 20:00:00</due-date>
<item-number>1</item-number>
<price-per-unit>0.0</price-per-unit>
<priced>true</priced>
<quantity>1</quantity>
<rejected>>false</rejected>
<rejected-date/>
<requisition-entry-id>69</requisition-entry-id>
<revision-number>105</revision-number>
<service>
<estimated-cost>0.0</estimated-cost>
<name>queueService1</name>
<parameters>
<default-duration>0.0</default-duration>
<priority>2</priority>
<start-date/>
<start-mode>0</start-mode>
</parameters>
<pricing-schema>0</pricing-schema>
<quantity>1</quantity>
<service-id>2</service-id>
<version>105</version>
<standard-duration>0.0</standard-duration>
</service>
<start-after/>
<start-date>2014-04-23 14:10:48</start-date>
<start-mode>0</start-mode>
<status>1</status>
</requisition-entry>
<requisition-id>64</requisition-id>
<started-on>2014-04-23 14:10:47</started-on>
<status>1</status>
</requisition>
<context>
<requisitionentryref itemnumber="1"/>
</context>
</task-started>
</message>

```

JSON 発信メッセージの例

```

{
  "message": {
    "taskStarted": {
      "task": {
        "actualDuration": 0.0,
        "calendarEntries": {},
        "checkLists": {},
        "completedDate": "",
        "contextId": 17,
        "contextType": "Requisition Entry",
        "dueDate": "2016-11-29 19:00:00",
        "effort": 10.0,
        "estimatedDate": "",
        "expectedDuration": 10.0,
        "flagId": 0,
        "isSharable": true,
        "isShared": true,
        "nextActionId": 2,
        "performer": {
          "companyAddress": "",
          "email": "",
          "fax": "",
          "firstName": "Default Service Delivery",
          "homeOu": {
            "name": "<s ID=\"847\"/>",
            "organizationalUnitId": 1
          },
          "homePhone": "",
          "lastName": "Queue",
          "loginName": "",
          "personId": 2,
          "personalAddress": "",
          "supervisorName": "",
          "timezone": "Pacific Standard Time",
          "workPhone": ""
        },
        "performerRole": {
          "name": ""
        },
        "performerActualDuration": 0.0,
        "priority": 2,
        "queue": {
          "companyAddress": "",
          "email": "",
          "fax": "",
          "firstName": "Default Service Delivery",
          "homeOu": {
            "name": "<s ID=\"847\"/>",
            "organizationalUnitId": 1
          },
          "homePhone": "",
          "lastName": "Queue",
          "loginName": "",
          "personId": 2,
          "personalAddress": "",
          "supervisorName": "",
          "timezone": "Pacific Standard Time",
          "workPhone": ""
        },
        "scheduledStartDate": "2016-11-28 17:00:00",
        "startDate": "2016-11-26 12:32:36",

```



```

"stateId": 2,
"subject": "amqpTask1",
"supervisor": {
  "companyAddress": "",
  "email": "",
  "fax": "",
  "firstName": "Default Service Delivery",
  "homeOu": {
    "name": "<s ID=\"847\"/>",
    "organizationalUnitId": 1
  },
  "homePhone": "",
  "lastName": "Queue",
  "loginName": "",
  "personId": 2,
  "personalAddress": "",
  "supervisorName": "",
  "timezone": "Pacific Standard Time",
  "workPhone": ""
},
"supervisorRole": {
  "name": ""
},
"taskId": 67,
"waiting": false
},
"requisition": {
  "services": 0,
  "actualCost": 0.0,
  "actualDuration": 0.0,
  "closedOn": "",
  "comments": {},
  "costCenterCode": "",
  "customer": {
    "companyAddress": "",
    "email": "internal@newscale.com",
    "fax": "",
    "firstName": "admin",
    "homeOu": {
      "name": "<s ID=\"847\"/>",
      "organizationalUnitId": 1
    },
    "homePhone": "",
    "lastName": "admin",
    "loginName": "admin",
    "personId": 1,
    "personalAddress": "",
    "supervisorName": "",
    "timezone": "Pacific Standard Time",
    "workPhone": ""
  },
  "dueOn": "2016-11-26 12:32:35",
  "expectedCost": 0.0,
  "expectedDuration": 0.0,
  "external": false,
  "initiator": {
    "companyAddress": "",
    "email": "internal@newscale.com",
    "fax": "",
    "firstName": "admin",
    "homeOu": {
      "name": "<s ID=\"847\"/>",
      "organizationalUnitId": 1
    },
  },

```

```

    "homePhone": "",
    "lastName": "admin",
    "loginName": "admin",
    "personId": 1,
    "personalAddress": "",
    "supervisorName": "",
    "timezone": "Pacific Standard Time",
    "workPhone": ""
  },
  "invocations": {},
  "organizationalUnit": {
    "name": "<s ID=\"847\"/>",
    "organizationalUnitId": 1
  },
  "requisitionEntry": [
    {
      "closedDate": "",
      "dataValues": {
        "dataValue": [
          {
            "name": "amqpDict1.field1",
            "value": [
              "a"
            ],
            "multiValued": false
          },
          {
            "name": "amqpDict1.field2",
            "value": [
              "b"
            ],
            "multiValued": false
          }
        ]
      },
      "dueDate": "2016-11-29 19:00:00",
      "itemNumber": 1,
      "pricePerUnit": 0.0,
      "priced": true,
      "quantity": 1,
      "rejected": false,
      "rejectedDate": "",
      "rejector": {
        "companyAddress": "",
        "email": "",
        "fax": "",
        "firstName": "",
        "homeOu": {},
        "homePhone": "",
        "lastName": "",
        "loginName": "",
        "personId": 0,
        "personalAddress": "",
        "supervisorName": "",
        "timezone": "",
        "workPhone": ""
      },
      "requisitionEntryId": 17,
      "revisionNumber": 3,
      "service": {
        "estimatedCost": 0.0,
        "form": {
          "fields": {}
        }
      }
    },
  ],

```

```
        "name": "amqpService1",
        "parameters": {
            "defaultDuration": 0.0,
            "priority": 2,
            "startDate": "",
            "startMode": 0
        },
        "pricingSchema": 0,
        "quantity": 1,
        "serviceId": 142,
        "version": 3,
        "standardDuration": 0.0
    },
    "startAfter": "",
    "startDate": "2016-11-26 12:32:36",
    "startMode": "0",
    "status": 1,
    "bundle": false
}
],
"requisitionId": 17,
"startedOn": "2016-11-26 12:32:35",
"status": 1
},
"context": {
    "requisitionentryref": {
        "itemnumber": 1
    }
}
},
"taskType": "task"
},
"channelId": "42346EF5-D2A7-EC03-E050-11AC02003EE1"
}
}
```

