
文章

[Lilian Huang](#) · 六月 14, 2022 阅读大约需 3 分钟

[首次使用InterSystems IRIS] 互操作性: Message消息

这篇文章是上一篇文章的延续<https://cn.community.intersystems.com/post/首次使用intersystems-iris-互操作性-一个production是什么？>

在上一篇文章：<https://cn.community.intersystems.com/post/首次使用intersystems-iris-互操作性-一个production是什么？> 我们查验了什么是Production. 我们运行了示例代码，并在Visual Trace 页面查验了如何将流动消息的内容引入进Production中。

本文将回顾消息messages

的概念和定义，和消息如何在系统集成所需的开发内容中用于组件之间发送和接收数据。

- Production生产<https://cn.community.intersystems.com/post/首次使用intersystems-iris-互操作性-一个production是什么？>
- **Message消息**
- Components组件
 - Business Services业务服务
 - Business Processes业务流程
 - Business Operations业务操作

在创建一个Message消息之前，让我们回顾一个案例研究。

一家公司经营着一个购物网站，并且正在为了适应季节而改变显示的产品信息顺序。

然而，有些商品无论季节如何都卖得很好，而有些商品在想不到的时间卖的很好，这与当前要更改顺序的显示规则不符。

/>因此，我们研究了更改显示商品顺序规则的可能性是匹配当天的温度而不是季节。

这就使得在购买商品时测量当时的温度变的很有必要。

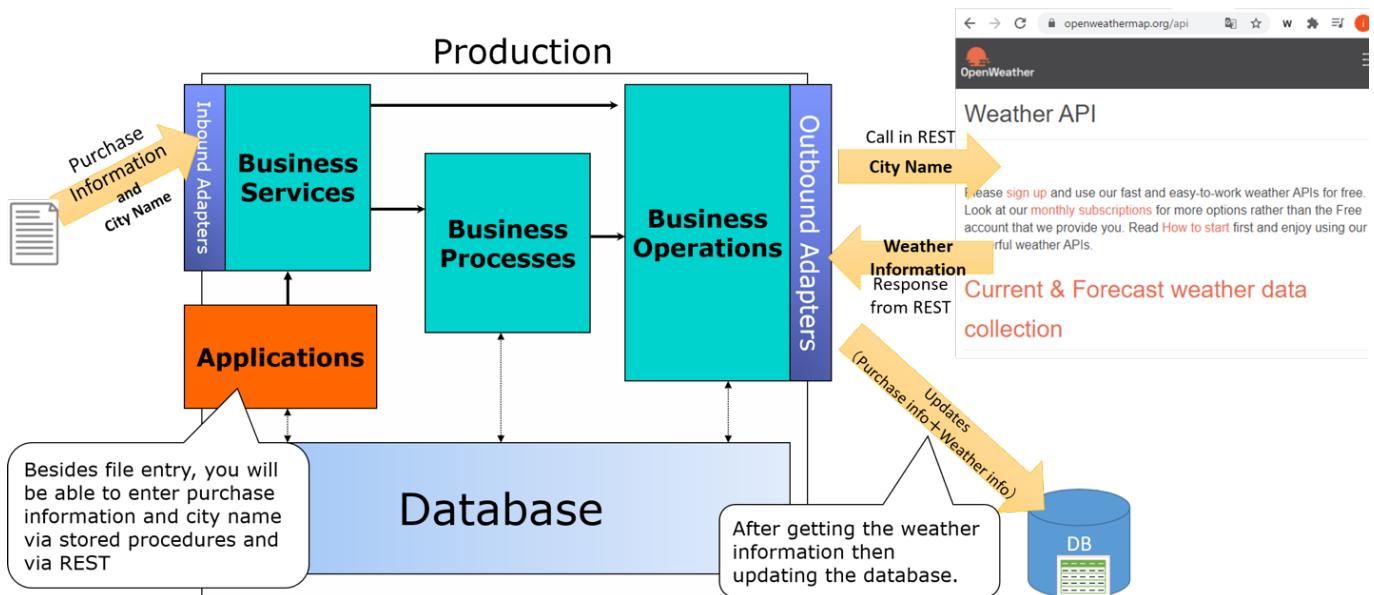
由于一个外部的Web

API可以查看天气信息，我们计划在商品购买时收集当时的天气信息并在将其录入数据库。

从本案例中，您可以看到以下内容：

从外部收到的信息是“购买的产品名称和城市名称”

从IRIS发送到外部系统以请求处理的信息是“城市名称”。该过程的结果是输入进的城市“天气信息”



通过本案例研究

, 我们将执行系统

集成所需的组件，在此之前，要运行

这些组件，需要发送和接收**messages消息**，

这些消息是中继转发数据，要使用这些**messages消息**，需要定义消息类。

一个消息类的设计需要考虑应该发送和接收哪些信息（即，**messages消息**），以使组件运行。

在此过程中，我们需要以下两类信息：

A) 要获取**天气信息**时发送给外部Web API 的**城市名称**。

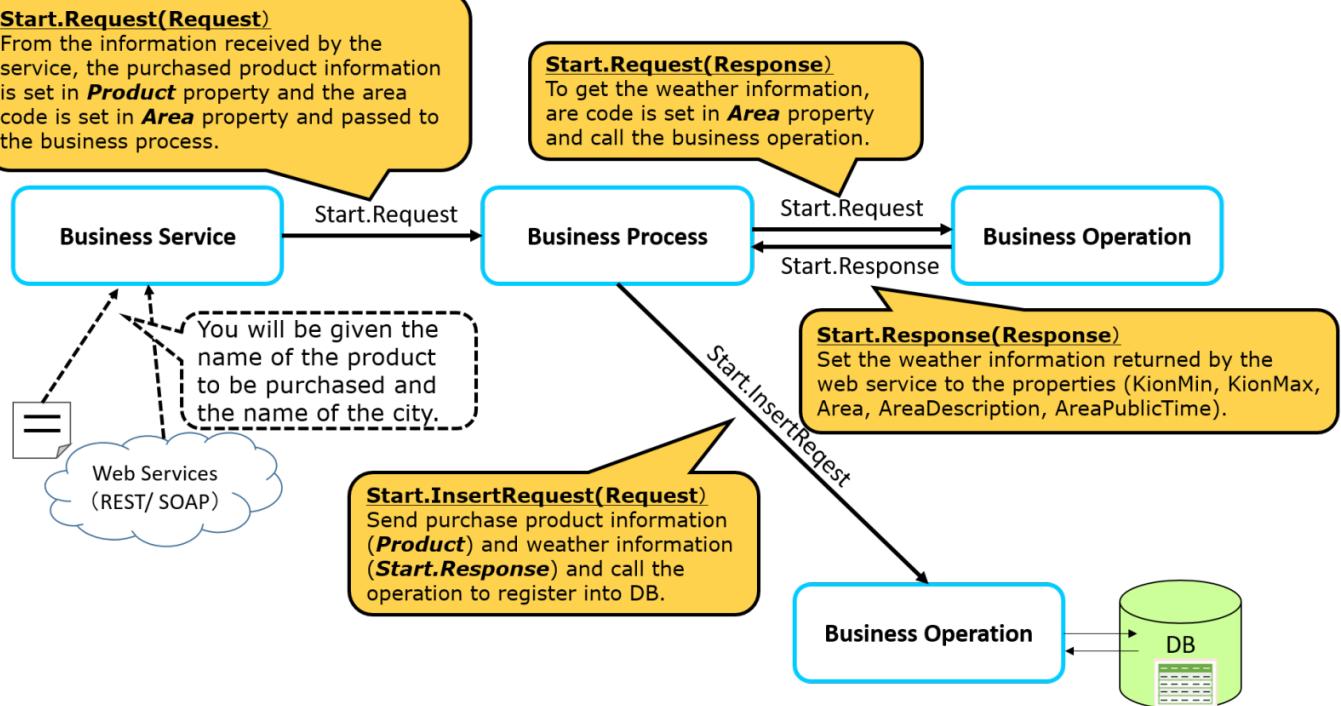
B) 记录在数据库中的**天气信息和购买的产品名称**。

A) 中的**城市名称**和B) 中的**购买产品名称**可以被包含进输入给IRIS的信息中。

B) 中的**天气信息**可以从**外部Web API**的响应信息中检索。

下面的图表就是考虑了从可用数据中发送和接收每个组件所需的信息。

黄色方框的第一行描述消息类别名，第二行之后说明为属性设置。



在示例代码中，我们有以下三种类型的messages消息：

Start.Request (请求消息)

它用于发送**购买的产品名称**和**城市**，以获取**天气信息**。

Start.Response (响应消息)

它们用于返回操作的结果 (**天气信息**)。

Start.InsertRequest (请求信息)

用于发送**天气信息**和**购买产品的名称**，以便在数据库系统注册。

The **messages** are specified in a superclass, Request message, and Response message are derived from Ens.Request and Ens.Response, respectively.

The following is an example of the definition of the Request message Start.Request.

messages 消息在

超类别中指定，请求消息和响应消息派生自Ens.Request and Ens.Response，分别作出回应。

以下是Start.Request的定义示例：

src > Start > Request.cls > Start.Request

```
1 Class Start.Request Extends Ens.Request
2 {
3
4 Property Product As %String;
5
6 Property Area As %String;
7
8 Storage Default
```

以下是响应消息的一个示例，

src > Start > Response.cls > Start.Response

```
1 Class Start.Response Extends Ens.Response
2 {
3
4 Property KionMax As %Numeric;
5
6 Property KionMin As %Numeric;
7
8 Property Area As %String;
9
10 Property AreaDescription As %String(MAXLEN = 1000);
11
12 Property AreaPublicTime As %String;
13
14 Storage Default
```

请求消息, 将请求输入和数据库注册的请求信息发送至如下：

(我们计划将天气信息属性设置在Start.Response中，获取天气信息后再返回。)

src > Start > InsertRequest.cls > Start.InsertRequest

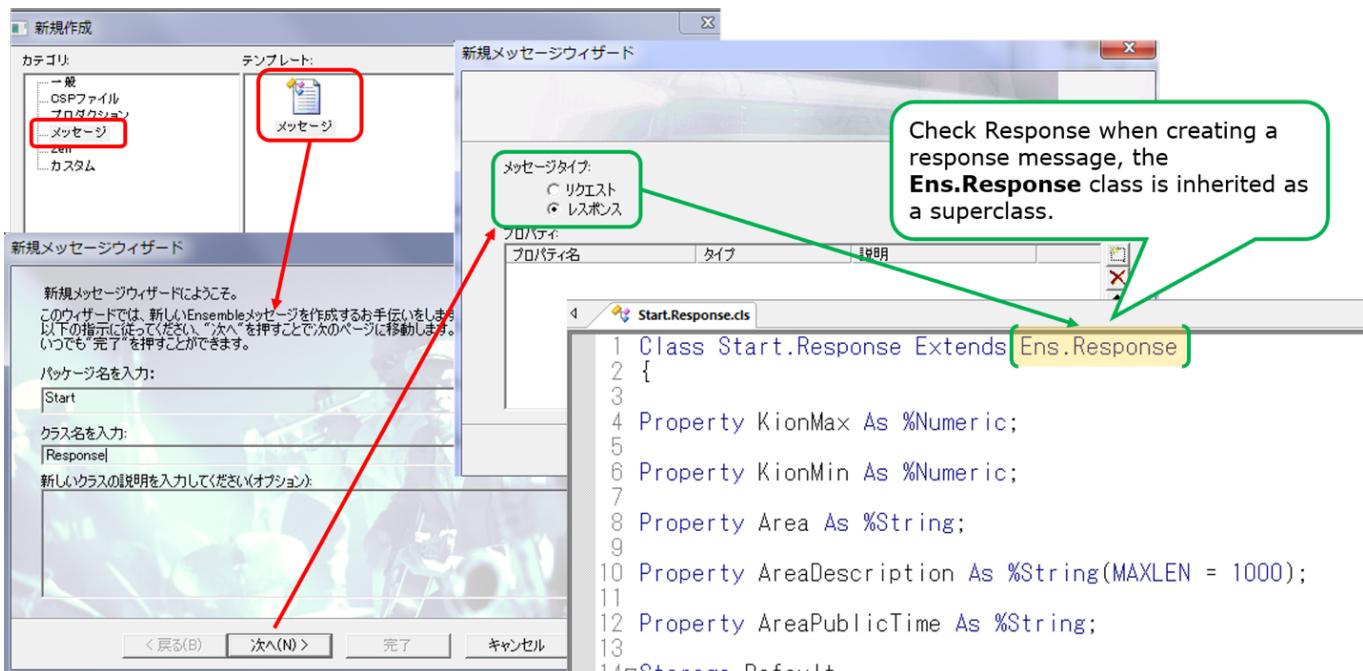
```

1 class Start.InsertRequest Extends Ens.Request
2 {
3
4 Property Product As %String;
5
6 Property WeatherInfo As Start.Response;
7

```

如果要在Studio中创建，您也可以使用Wizard创建消息。

参考) 在Studio中创建响应类的步骤。



总结关键的点在于消息设计的思路“应该发送和接收哪些信息.e., messages) ”以使组建运行。

一旦消息类实现后（驱动每个组件的信息是什么），下一步即是为组件创建类别。

原文请参考：<https://community.intersystems.com/post/intersystems-iris-first-time-int...> written by Mihoko Iijima

#API #HL7 #InterSystems 业务解决方案和架构 #Caché #HealthShare #InterSystems IRIS #InterSystems IRIS for Health #学习门户

源

URL:

<https://cn.community.intersystems.com/post/%E9%A6%96%E6%AC%A1%E4%BD%BF%E7%94%A8intersystems-iris-%E4%BA%92%E6%93%8D%E4%BD%9C%E6%80%A7-message%E6%B6%88%E6%81%AF>