
Artículo

[Ricardo Paiva](#) · 18 feb, 2022 Lectura de 4 min

[InterSystems IRIS por primera vez] Interoperabilidad: Mensaje

Este artículo es una continuación de [esta publicación](#).

En ese artículo revisamos los contenidos de la producción. Ejecutamos el código de muestra y revisamos el contenido de los mensajes que fluyen en la producción en la página de Visual Trace.

En este artículo revisaremos el concepto y la definición de los **mensajes** que se utilizan para enviar y recibir datos entre componentes desde el contenido de desarrollo necesario para la integración del sistema.

- [Producción \[publicación anterior \]](#)
- **Mensaje**
- Componentes
 - Business Services
 - Business Processes
 - Business Operations

Antes de crear un **mensaje**, vamos a revisar el caso de estudio.

Una empresa gestiona una tienda online y está cambiando el orden en que se muestra la información de los productos, para que coincida con las estaciones del año.

Sin embargo, algunos artículos se venden bien independientemente de la estación del año, mientras otros se venden en momentos inesperados, lo que no coincide con la regla actual de cambiar el orden.

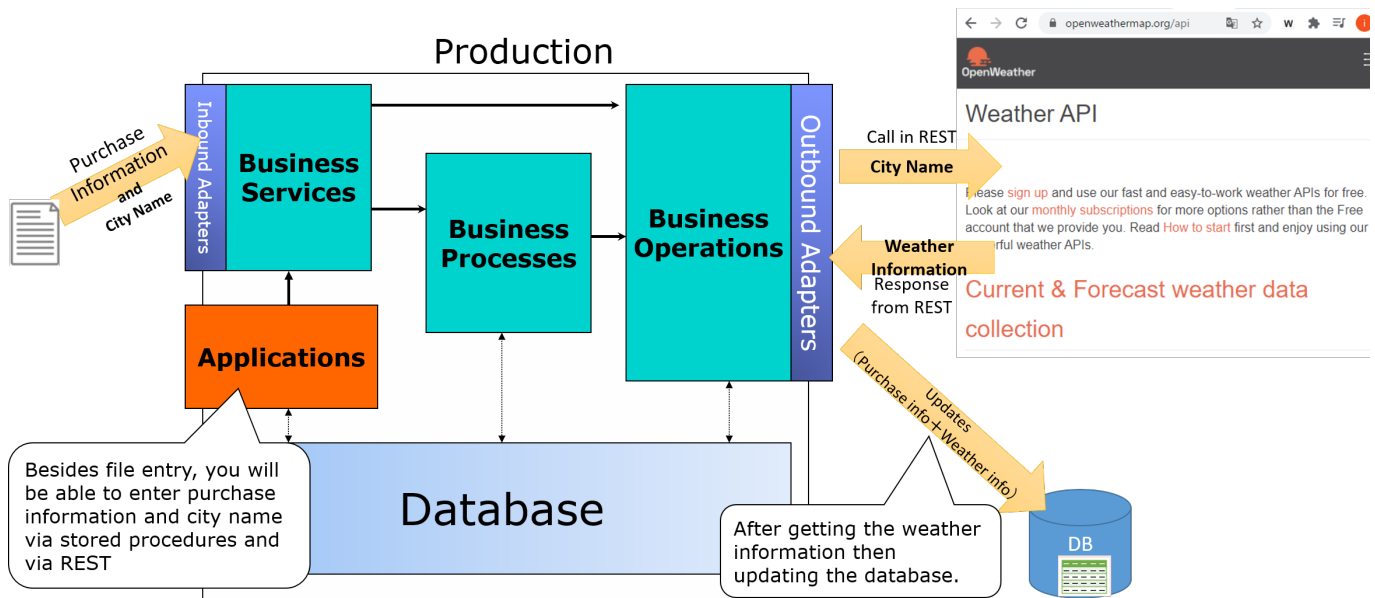
Por lo tanto, estudiamos la posibilidad de cambiar el orden para que coincida con la temperatura del día en lugar de con la estación del año. Era necesario saber la temperatura de los productos adquiridos en ese momento.

Como se dispone de una API web externa para consultar la información meteorológica, planeamos recopilar la información meteorológica en el momento de la compra y registrarla en la base de datos.

En este caso, puedes observar lo siguiente:

La información que se recibe del exterior es el "nombre del producto comprado y el nombre de la ciudad".

La información que se envía desde IRIS a un sistema externo para solicitar el procesamiento es el "nombre de la ciudad". El resultado de este proceso será la "información meteorológica" para la ciudad que se utilizará como entrada.



A partir de este caso de estudio, implementaremos los componentes necesarios para la integración del sistema. Pero antes, para ejecutar los componentes, es necesario enviar y recibir **mensajes**, que son datos transmitidos; y para utilizar los **mensajes**, es necesario definir la clase de mensajes.

Una clase de mensajes se diseña para considerar **qué información (es decir, mensajes) debe enviarse y recibirse para que el componente se ejecute.**

En este proceso, necesitamos estos dos tipos de información:

- A) El **nombre de la ciudad** para enviar a una API web externa y así obtener la **información meteorológica**.
- B) **La información meteorológica** y **el nombre del producto adquirido** para su registro en la base de datos.

El **nombre de la ciudad** en A) y el **nombre del producto comprado** en B) pueden incluirse en la información de entrada en IRIS.

La **información meteorológica** en B) puede recuperarse de la información de respuesta de una **API web externa**.

A continuación se muestra un diagrama que considera qué información sería necesaria para enviar y recibir cada componente a partir de los datos disponibles.

La **primera línea** en los **globos amarillos** describe el **nombre de la clase de mensajes**, y a partir de la segunda línea se indica lo que se debe establecer en las propiedades.

En el código de ejemplo, hay tres tipos de **mensajes**:

Start.Request (Mensaje de solicitud)

Se utiliza para enviar el nombre del **producto comprado** y la **ciudad** para adquirir la información meteorológica.

Start.Response (Mensaje de respuesta)

Se utiliza para devolver los resultados de las operaciones (**información meteorológica**) para obtener información meteorológica.

Start.InsertRequest (Mensaje de solicitud)

Se utiliza para enviar la **información meteorológica** y el **nombre de los productos comprados** para el registro en la base de datos.

Los **mensajes** se especifican en una superclase, Request message, y los Response messages derivan de **Ens.Request** y **Ens.Response**, respectivamente.

El siguiente es un ejemplo de la definición del mensaje de solicitud Start.Request.

Este es un ejemplo para la definición del mensaje de respuesta.

El mensaje de solicitud, Start.InsertRequest que se enviará con la solicitud de registro en la base de datos es el siguiente:

(Planeamos establecer la propiedad WeatherInfo con la información de Start.Response, que se devolverá después de obtener la información meteorológica).

Si quieres crearlo en Studio, también puedes utilizar el Asistente para Crear Mensajes.

Referencia) Pasos para crear una clase de respuesta en Studio.

La clave hasta ahora es

Una clase de mensajes se diseña con la idea de "qué información (es decir, **mensajes**) debe enviarse y recibirse" para que el componente FUNCIONE.

Cuando la clase de mensajes (que es la información que dirige cada componente) se implementa, el siguiente paso es crear una clase para el componente.

[#Interoperabilidad](#) [#Principiante](#) [#InterSystems IRIS](#) [#InterSystems IRIS for Health](#)

URL de
fuelle: <https://es.community.intersystems.com/post/intersystems-iris-por-primera-vez-interoperabilidad-mensaje>