

El desarrollo de servicios es una de las dos opciones principales para la creación de servicios. Si bien es más complicado que su contraparte, la generación de servicios, también es más flexible.

6 Desarrollo de servicios

En este capítulo se explica el proceso de desarrollo de servicios, que es el desarrollo del lado back-end de los servicios OData mediante ABAP. Después de leer este capítulo, podrá desarrollar servicios con la herramienta de desarrollo principal, SAP Gateway Service Builder, mientras utiliza el canal OData como paradigma de desarrollo.

Los pasos descritos en este capítulo están en línea con los tres pasos principales del desarrollo del servicio (definición del modelo de datos, implementación del servicio y mantenimiento del servicio) como se explica en el [Capítulo 5](#). Como recordará, la definición del modelo de datos, la implementación del servicio y el mantenimiento del servicio

Definición del modelo de datos, implementación del servicio, mantenimiento del servicio.

No es necesario que se realicen necesariamente con un enfoque en cascada; la definición del modelo de datos y la implementación del servicio son tareas iterativas que se revisarán con frecuencia para permitir que el servicio OData y sus capacidades crezcan iteración tras iteración. Este enfoque es muy similar a otros lenguajes de programación, donde se comienza con un primer elemento de interfaz de usuario (UI) y se implementa su funcionalidad, y luego se agrega el siguiente elemento de UI e implementa su funcionalidad, y así sucesivamente.

En este capítulo, explicamos los pasos del desarrollo del servicio OData tal como se llevarían a cabo en un escenario real. La Figura 6.1 [muestra los pasos](#) de este proceso de desarrollo.

6 Desarrollo de servicios

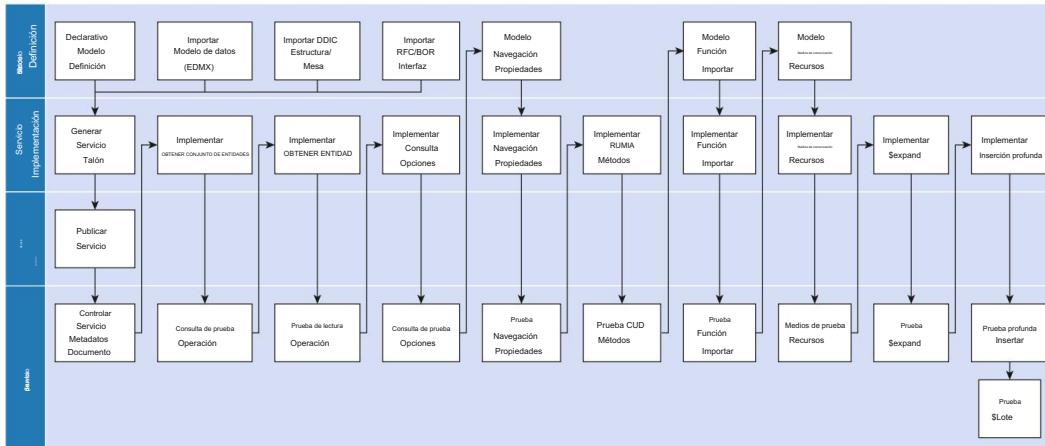


Figura 6.1 Ejemplo de proceso de desarrollo

6.1 Definición del modelo de datos

Service Builder Como aprendió en los capítulos anteriores, cada servicio OData consta de una definición de modelo de datos y una implementación de servicio. En tiempo de ejecución, la definición de modelo da como resultado el documento de servicio y el documento de metadatos de servicio, y la implementación de servicio da como resultado la funcionalidad real del servicio SAP Gateway. Para un servicio OData de SAP Gateway que utiliza el paradigma de desarrollo de canal OData, la definición de modelo se proporciona a través de la clase de proveedor de modelos (MPC). La implementación de servicio se proporciona a través de la clase de proveedor de datos (DPC).

La herramienta que realiza ambas tareas es Service Builder.

Creación manual de servicios

Puede implementar el MPC y el DPC manualmente mediante el uso del generador de clases ABAP (Transacción SE24) en ABAP. Sin embargo, este proceso puede llevar bastante tiempo y, especialmente en el caso del MPC, puede ser un poco propenso a errores.

También en el lado de DPC, hay algunas tareas recurrentes (por ejemplo, mapear una fuente de datos y crear métodos CRUD-Q para los diferentes conjuntos de entidades) que son más fáciles con una herramienta.

De las posibles opciones de implementación, has visto que también es posible realizar una llamada remota a otro sistema desde el componente SAP_GWFND/IW_BEP.

En este caso, hay que tener en cuenta que, por ejemplo, es posible que un determinado elemento de datos no esté disponible localmente en el sistema SAP_GWFND/IW_BEP.

Hay dos pasos principales en la definición del modelo de datos: primero, crear un proyecto y, segundo, definir el modelo de datos real. En esta sección, lo guiaremos a través de estos dos pasos principales.

Dos pasos en la definición del modelo de datos

6.1.1 Creación de un proyecto

Antes de poder comenzar con la definición del modelo, primero debe iniciar la Transacción SEGW Service Builder (Transaction SEGW) y crear un nuevo proyecto. Un proyecto define los corchetes que rodean todos los artefactos que se crean durante el desarrollo de su servicio OData. Service Builder le permite abrir y trabajar en varios proyectos en paralelo. La Figura 6.2 muestra un ejemplo de un solo proyecto de Service Builder.

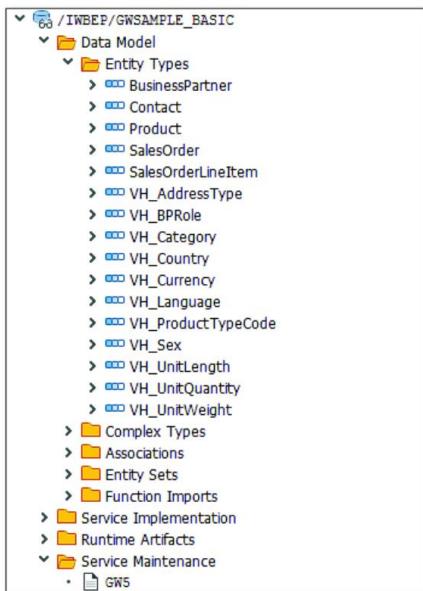


Figura 6.2 Ejemplo de un proyecto de Service Builder

El nodo raíz es el proyecto en sí, que, en este caso, es /IWBEPE/GWSAMPLE_BASIC (/IWBEPE/GWSAMPLE_BASIC es un servicio de muestra que forma parte del envío estándar a partir de SAP_GWFND 7.40 SP 08). El primer subnodo es el Modelo de datos. Este nodo se encarga de la definición completa del modelo. El subnodo Implementación del servicio realmente inyecta vida a su servicio al asignar fuentes de datos (por ejemplo, llamadas a funciones remotas).

6 Desarrollo de servicios

[RFC] o Business Application Programming Interfaces [BAPI]) a su modelo. También le permite implementar manualmente los métodos respectivos mediante ABAP. Runtime Artifacts resume todos los artefactos (o los más importantes) que se generan durante el ciclo de vida de su proyecto. Finalmente, Service Maintenance le permite registrar y activar su servicio en el hub SAP Gateway.

Al iniciar Service Builder por primera vez, verá una ventana vacía (consulte la Figura 6.3).

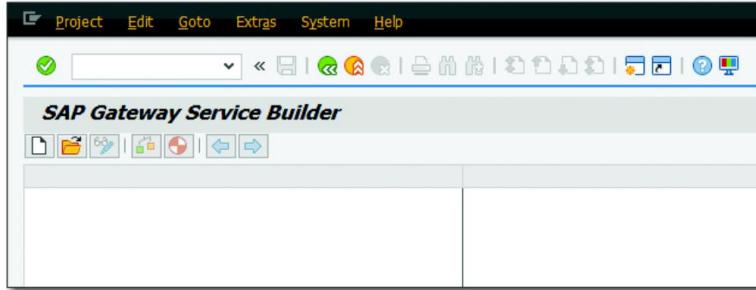


Figura 6.3 Pantalla de inicio de Service Builder

Crear proyecto Elija el botón Crear proyecto para crear un nuevo proyecto de Service Builder.

etc. (ver [Figura 6.4](#)).

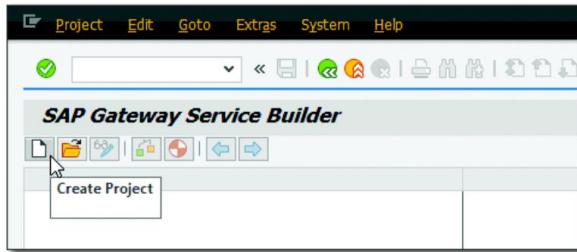


Figura 6.4 Botón Crear proyecto de Service Builder

Detalles del proyecto Esto abre el cuadro de diálogo Crear proyecto donde deberá proporcionar lo siguiente (consulte la Figura 6.5):

Proyecto

No se permiten caracteres especiales ni espacios en los nombres de los proyectos, y no pueden comenzar con números. Las palabras individuales pueden estar separadas por

guiones bajos. La longitud máxima del nombre del proyecto es de 30 caracteres. actores.

Descripción

Un campo de texto libre de 60 caracteres que describe su proyecto.

Tipo de proyecto

Elija entre las siguientes opciones:

Servicio con anotaciones de SAP: este es el tipo de proyecto predeterminado. En este contexto, las anotaciones hacen referencia a anotaciones SAP estándar basadas en atributos básicos que se pueden usar para anotar el modelo de datos (por ejemplo, sap:creatable).

Servicio con anotaciones basadas en vocabulario: este tipo de proyecto le permite asignar uno o más vocabularios a su proyecto y, de esa manera, hacer uso de términos basados en vocabulario. Puede elegir entre los vocabularios estándar que se proporcionan de manera predeterminada o cargar sus propios vocabularios.

Modelo de anotación para servicio referenciado: este tipo de proyecto se utiliza para anotar un servicio existente. Las anotaciones no aparecen con los metadatos del servicio en sí, pero se puede acceder a ellas a través de /IWFND/CatalogService;v=2. Estas anotaciones están pensadas para que las lea el consumidor (por ejemplo, la aplicación HTML5) para influir en el comportamiento de la interfaz de usuario. yo.

El tipo de proyecto predeterminado es Servicio con anotaciones SAP, que hace referencia a un proyecto con anotaciones regulares (consulte la Tabla 6.1). _____

Estrategia de generación

Esta opción está predeterminada como Estándar. Actualmente, solo se puede elegir una estrategia de generación. Según el tipo de proyecto, es posible que existan estrategias de generación adicionales en el futuro.

Paquete y Persona Responsable

Estos son los elementos predeterminados que ya conoce de otros objetos del repositorio (por ejemplo, informes ABAP, tablas de diccionario de datos [DDIC], etc.). \$tmp se utiliza para desarrollos locales que no desea transportar a ningún sistema de destino. Si desea transportar el proyecto y su contenido a cualquier sistema de destino, debe proporcionar un paquete transportable. Dicho paquete se puede crear utilizando ABAP Development Workbench (Transacción SE80).

6 Desarrollo de servicios

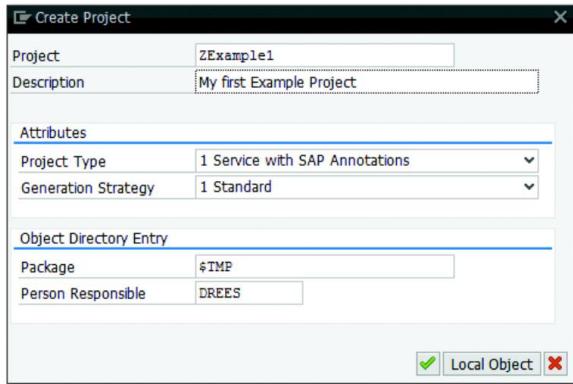


Figura 6.5 Pantalla de diálogo Crear proyecto

Entrada del repositorio ABAP El nombre del proyecto se convierte en una entrada del repositorio ABAP, por lo que el nombre del proyecto debe cumplir con las convenciones de nomenclatura típicas de SAP (comenzar con una "Z" o una "Y" para separar los objetos del cliente de los objetos de SAP). Además de eso, también puede utilizar un espacio de nombres registrado rodeado de barras (por ejemplo, /MyNamespace/project) tal como lo hace cuando, por ejemplo, crea su propio elemento de datos o módulo de función utilizando las otras herramientas de desarrollo estándar de SAP. La entrada del repositorio de objetos en este ejemplo es R3TR IWPR ZEXAMPLE1. Aquí, IWPR hace referencia a Gateway Business

Suite Enablement – Service Builder Project y, por supuesto, se puede transportar a través de una solicitud de transporte de ABAP Workbench normal. IWPR solo transporta el proyecto como tal, no los artefactos generados (por ejemplo, MPC o DPC).

Éstas serán entradas separadas en la solicitud de transporte con sus respectivos tipos de objetos de repositorio.

6.1.2 Creación de un modelo de datos

Modelo de datos Opciones Tan pronto como tenga un proyecto, puede comenzar a crear su servicio SAP OData. Lo primero que debe hacer es crear el modelo de datos. Como se presentó en el Capítulo 5, existen múltiples opciones para crear un modelo de datos:

Crear un modelo de datos declarativo.

Importar un modelo de datos a través del modelo de datos de entidad XML (EDMX).

Importe un modelo de datos a través de DDIC.

Importar un modelo de datos a través de RFC/Business Object Repository (BOR).

La definición del modelo declarativo se suele elegir siempre que se empieza desde cero con la definición del modelo de datos, lo que significa que no se dispone de ninguna fuente de la que derivar el modelo de datos. De lo contrario, se utilizará el método que sea adecuado para la fuente de datos en cuestión.

Analicemos cada una de estas opciones con más detalle.

Creación de un modelo de datos declarativo

Un modelo de datos declarativo se crea manualmente. Para crear su primera entidad Creación manual. Para escribir a través de este método, haga clic derecho en Modelo de datos y elija Crear tipo de entidad (consulte la [Figura 6.6](#)).

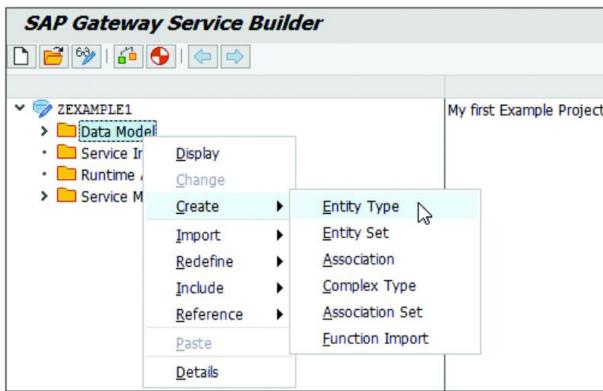


Figura 6.6 Creación de un nuevo tipo de entidad

En el cuadro de diálogo Crear tipo de entidad, puede especificar el nombre de su tipo de entidad. El nombre debe ser único dentro de su proyecto.

Opcionalmente, también puede marcar la casilla de verificación Crear conjunto de entidades relacionadas para crear un conjunto de entidades correspondiente con una asignación a este tipo de entidad en un solo paso. Normalmente, existe una relación 1:1 entre un tipo de entidad y un conjunto de entidades, pero esto no es obligatorio; un tipo de entidad se puede utilizar en muchos conjuntos de entidades. Por ejemplo, un tipo de entidad BusinessPartner se puede utilizar en conjuntos de entidades con el nombre ShipToPartnerSet, SoldToPartner-Set, etc.

[Crear un relacionado
conjunto de entidades](#)

En el primer ejemplo, creará un tipo de entidad de producto (consulte la [Figura 6.7](#)) con un [conjunto de propiedades](#). Despues de ingresar el nombre del tipo de entidad, seleccione el

6 Desarrollo de servicios

Casilla de verificación Crear conjunto de entidades relacionadas para que el nombre del conjunto de entidades predeterminado y el conjunto de entidades creado con el tipo de entidad.

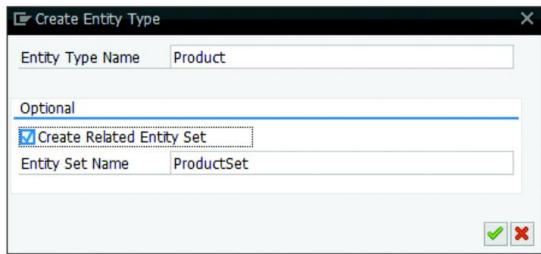


Figura 6.7 Creación de un nuevo tipo de entidad

Una vez creado el tipo de entidad, el árbol del proyecto se actualiza automáticamente y se resalta y selecciona el nuevo tipo de entidad. En la sección de detalles, se ven todos los atributos del tipo de entidad (consulte la Figura 6.8).

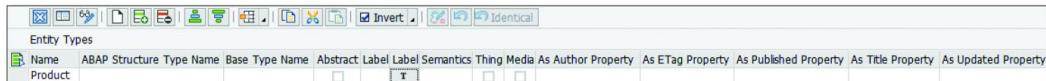


Figura 6.8 Atributos del tipo de entidad

Uno de los atributos importantes es el nombre del tipo de estructura ABAP. A través de este atributo opcional, puede definir un vínculo de su tipo de entidad a una estructura o tabla DDIC. Esto es muy importante porque le permite heredar la definición de su propiedad (por ejemplo, longitud y precisión) del elemento DDIC subyacente. Además, hereda todas las etiquetas, incluidas sus traducciones, del DDIC. Esto es beneficioso si ofrece su aplicación en varios idiomas y desea que la aplicación seleccione las etiquetas de los metadatos expuestos (ya sea una vez durante el tiempo de diseño o dinámicamente durante el tiempo de ejecución a través del documento de metadatos) en el idioma respectivo. Incluso si está ejecutando un solo idioma, este vínculo significa que no tiene que mantener sus etiquetas en varias áreas. (Por otro lado, si la aplicación obtiene dinámicamente sus etiquetas del documento de metadatos, no puede predecir cómo se verán las cosas en la interfaz de usuario final. Depende completamente de usted qué enfoque prefiere).

Hasta el momento, el tipo de entidad no tiene propiedades. Esto se puede cambiar expandiendo el árbol del proyecto y haciendo doble clic en Propiedades (consulte la Figura 6.9).



Figura 6.9 Nodo Propiedades en el Árbol del Proyecto

En el lado derecho de la ventana del proyecto, se muestra la cuadrícula de propiedades del tipo de entidad actual. Debido a que no hay ninguna propiedad hasta el momento, la cuadrícula está vacía después de crear un nuevo tipo de entidad. Al hacer clic en el botón Anexar Los botones Fila o Insertar fila le permiten crear una nueva propiedad para el tipo de entidad actual.

Como información obligatoria, deberás facilitar el Nombre y el EDM Tipo de núcleo de la propiedad (incluso si proporciona una referencia DDIC). Cada tipo de entidad también debe tener al menos una propiedad marcada como clave propiedad.

En el ejemplo actual, creará tres propiedades para el producto Tres propiedades

Tipo de entidad: ProductID, Category y Name. Todos ellos se definen como Edm.String con longitudes individuales (consulte la Figura 6.10).

The screenshot shows the 'Properties' grid in SAP Gateway Service Builder. The grid has columns: Name, Key, Edm Core Type, Precision, Scale, and Max Length. There are three rows: 'ProductID' (marked as Key, Edm.String type, precision 0, scale 0, max length 10), 'Category' (not marked as key, Edm.String type, precision 0, scale 0, max length 40), and 'Name' (not marked as key, Edm.String type, precision 0, scale 0, max length 255). The 'Key' column has a checked checkbox for ProductID and an unchecked checkbox for the other two. The 'Edm Core Type' column shows 'Edm.String' for all. The 'Precision' and 'Scale' columns are mostly 0, except for Category which is 40. The 'Max Length' column shows 10 for ProductID, 40 for Category, and 255 for Name. The top row of the grid has icons for copy, paste, invert, and other operations. The word 'Properties' is visible above the grid.

Name	Key	Edm Core Type	Precision	Scale	Max Length
ProductID	<input checked="" type="checkbox"/>	Edm.String	0	0	10
Category	<input type="checkbox"/>	Edm.String	0	0	40
Name	<input type="checkbox"/>	Edm.String	0	0	255

Figura 6.10 Propiedades del tipo de entidad

La propiedad ProductID está marcada como una propiedad clave. Por lo tanto, la propiedad clave principal La clave del tipo de entidad del producto consta de un solo campo. El valor máximo

6 Desarrollo de servicios

El atributo de longitud se define como 10 para el ID del producto, 40 para la categoría y 255 para el nombre en este ejemplo.

Los indicadores de atributos (ver Figura 6.11) se describen en la Tabla 6.1.

Creatable	Updatable	Deletable	Sortable	Nullable	Filt.
<input type="checkbox"/>					
<input type="checkbox"/>					
<input type="checkbox"/>					
<input type="checkbox"/>					

Figura 6.11 Indicadores de atributos de propiedad

Descripción/ejemplo de la bandera	Aplicación
sap:creable :creable Esta propiedad puede ser proporcionada por una aplicación al crear un recurso de entrada de este tipo de entidad.	La mayoría de las propiedades suelen tener esta bandera establecida porque se supone que deben completarse cuando se crea una entrada (por ejemplo, Custom-erName).
sap:actualizable :updatable Esta propiedad puede ser proporcionada por una aplicación al actualizar un recurso de entrada de este tipo de entidad.	Este atributo actualizable suele estar en línea con el atributo creable. Normalmente se establece siempre que se pueda cambiar una propiedad de una entrada existente (por ejemplo, el número de teléfono). Generalmente no se establece cuando el backend calcula una propiedad (por ejemplo, ChangeTime-Stamp o NetSum).
Eliminable N/A	Este atributo no tiene relevancia porque no se pueden eliminar propiedades individuales (esto no tiene efecto al cambiar propiedades individuales a su estado inicial).

Tabla 6.1 Atributos del tipo de entidad

Bandera	Anotación	Descripción/Ejemplo
	Sap ordenable :sortable	La anotación ordenable indica a un consumidor si la propiedad se puede utilizar en una opción de consulta \$orderby (por ejemplo, \$orderby=CustomerName).
	Anulable Anulable	La anotación que acepta valores nulos define si una propiedad puede contener un valor nulo o no. Esto es muy importante para los campos de fecha u hora , por ejemplo, si no contienen un valor en el lado ABAP; por lo tanto, un campo vacío no se puede representar en una respuesta OData si no se establece la anotación que acepta valores nulos. Es un buen consejo establecer el atributo que acepta valores nulos para todas las propiedades que no son clave para asegurarse de que puedan contener incluso valores ABAP vacíos. Por otro lado, un consumidor considera que los atributos que aceptan valores nulos son opcionales, lo que podría no estar en línea con su lógica comercial. Tenga en cuenta que las propiedades clave no pueden contener propiedades que acepten valores nulos, es decir, no pueden tener el valor Nulo.
	Sap filtrable :filterable	La anotación filtrable indica a un consumidor si la propiedad se puede usar en una opción de consulta \$filter (por ejemplo, \$filter=OrderNumber eq '1234').

Tabla 6.1 Atributos del tipo de entidad (cont.)

Las anotaciones se representan en el documento de metadatos (ver Figura 6.12).

```
- <EntityType Name="Product" sap:content-version="1">
  - <Key>
    <PropertyRef Name="ProductID" />
  </Key>
  <Property Name="ProductID" Type="Edm.String" Nullable="false" MaxLength="10" sap:creatable="false" sap:updatable="false"
    sap:sortable="false" sap:filterable="false" />
</EntityType>
```

Figura 6.12 Metadatos de propiedad con todos los indicadores configurados (como ejemplo)

Los atributos creables, actualizables, ordenables y filtrables solo se utilizan en el documento de metadatos para informar a la aplicación del consumidor sobre el comportamiento previsto. No existe ningún control a través del marco SAP Gateway

Atributos que se pueden crear, actualizar, ordenar y filtrar

6 Desarrollo de servicios

Si una determinada propiedad se está utilizando de la forma en que se anota a través de los metadatos. Incluso si una propiedad no está marcada como actualizable, puede proporcionarla junto con una solicitud de actualización y el valor de la propiedad llegará a la implementación del proveedor de datos. Por lo tanto, no se filtrará ni se manejará mediante un error, por ejemplo, si realiza una operación que no está permitida según las anotaciones en el documento de metadatos.

Como recomendación, la implementación del proveedor de datos (DPC) siempre debe comportarse de la misma manera porque los metadatos definen el modelo. Pero esto no es obligatorio y, lamentablemente, en muchas implementaciones de servicios no se tiene en cuenta, ya que la mayoría de los servicios se desarrollan para casos de uso individuales (por ejemplo, para una aplicación de consumidor dedicada y conocida que, por lo general, los convierte en un servicio no reutilizable). Siempre que desarrolle un servicio OData para su reutilización, se recomienda encarecidamente definir los metadatos adecuados; de lo contrario, los consumidores del servicio deben identificar las capacidades del servicio mediante prueba y error, lo que obviamente no es una buena práctica.

Como ejemplo ([consulte la Figura 6.13](#)), marque todos los atributos Creables de las propiedades ProductID, Category y Name. Más adelante (en las siguientes secciones), implementaremos la operación Create para que se proporcionen todas las propiedades al realizar una creación. El atributo Updatable no se marcará para ProductID porque la clave principal no se puede cambiar en nuestro ejemplo.

La opción Sortable no está marcada en absoluto porque no implementaremos ninguna capacidad de clasificación. Lo mismo se aplica a Nullable, ya que ninguna de las propiedades es opcional. Filterable solo está configurado para ProductID y Category, porque sólo proporcionaremos capacidades de filtrado para estas dos propiedades.

Properties													
	Name	Key	Edm Core Type	Prec.	Scale	Max Lngth	U	Creatable	Updatable	Sortable	Nullable	Filterable	Label
	ProductID	<input checked="" type="checkbox"/>	Edm.String	0	0	10		<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	Product Identifier
	Category	<input type="checkbox"/>	Edm.String	0	0	40		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	Product Category
	Name	<input type="checkbox"/>	Edm.String	0	0	255		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	Product Name

Figura 6.13 Atributos y etiquetas mantenidos

El siguiente paso es crear un conjunto de entidades, lo cual es necesario porque el suministro y el consumo de servicios se realizan en función de conjuntos de entidades. Se puede crear un conjunto de entidades haciendo doble clic en el nodo del conjunto de entidades en el Explorador de proyectos. Esto abre la sección Conjuntos de entidades en el lado derecho. Puede crear nuevos conjuntos de entidades utilizando el botón Anexar. Botones Fila o Insertar fila.

Como parte de nuestro ejemplo, hemos creado un conjunto de entidades con el nombre ProductSet durante la creación del tipo de entidad (ver Figura 6.14).

Name	Entity Type Name	L	L...	S...	Creatable	Updatable	Deletable	Pageable	Addressable	Searchable	Subscribable	Requires Filter
ProductSet	Product	T			<input type="checkbox"/>							

Figura 6.14 Conjunto de entidades ProductSet (estado inicial)

Existen diferentes enfoques para nombrar un conjunto de entidades para un conjunto de entidades determinado Tipo de entidad. Algunas personas prefieren agregar “Colección” al nombre del tipo de entidad (en este ejemplo, el resultado sería ProductCollection). Otras prefieren usar el plural (Products). En este ejemplo, agregará “Conjunto” para definir el nombre del conjunto de entidades resultante. Depende completamente de usted qué enfoque prefiere y cómo manejarlo en sus propios servicios OData; solo asegúrese de no mezclar los enfoques dentro de un solo servicio OData.

Para cada conjunto de entidades, también debe especificar el nombre del tipo de entidad subyacente. Esta es una información obligatoria. El nombre del tipo de entidad se puede escribir manualmente o elegir desde un cuadro de diálogo de ayuda (F4) . Si ha creado el conjunto de entidades durante la creación del tipo de entidad, la asignación ya está hecha.

Opcionalmente, también puede proporcionar una etiqueta. Esta etiqueta se representa en el documento de metadatos y permite que una aplicación de consumidor, por ejemplo, muestre el título de una página.

En el nivel de conjunto de entidades, hay nuevamente ciertos atributos que anotan las capacidades del conjunto de entidades relacionado. De manera similar a las banderas en el nivel de tipo de entidad, esas banderas son solo parte de la metainformación (en este caso, tanto en los metadatos como en el documento de servicio). No hay

Atributos del
conjunto de entidades

6 Desarrollo de servicios

Manejo o acción realizada a través del marco, lo que significa que, por ejemplo, si un conjunto de entidades no está anotado como creable, aún puede realizar una solicitud POST y crear el recurso de entrada relacionado. Nuevamente, asegúrese de que los metadatos del servicio OData estén en línea con el aprovisionamiento del servicio real.

En la Tabla 6.2 se describe el significado de los atributos del conjunto de entidades.

Bandera	Anotación	Descripción/Ejemplo
	Creable sap:creatable	<p>Creación de entradas de la entidad relacionada</p> <p>El tipo es compatible.</p> <p>Las creaciones se gestionan mediante HTTP POST solicitudes.</p> <p>Por ejemplo, puede crear una entrada de socio comercial.</p>
	sap:updatable	<p>Se admiten actualizaciones de entradas del tipo de entidad relacionada.</p> <p>Las actualizaciones se gestionan mediante HTTP PUT solicitudes dirigidas al recurso de entrada que desea actualizar.</p> <p>Por ejemplo, puede actualizar un producto con el ID de producto "123".</p>
	Eliminable sap:deletable	<p>Eliminaciones de entradas de la entidad relacionada</p> <p>Se admiten tipos.</p> <p>Las eliminaciones se gestionan mediante HTTP DELETE solicitudes dirigidas al recurso de entrada que desea eliminar.</p> <p>Por ejemplo, puede eliminar una línea de pedido de venta con la clave OrderId='50000',Item='10'.</p>
Paginable	sap:paginable	<p>Se admite la paginación del lado del cliente de este conjunto de entidades. La paginación del lado del cliente funciona a través del parámetro de consulta. El cliente puede definir el tamaño de la página y la cantidad de entradas que se omitirán.</p>

Tabla 6.2 Atributos del conjunto de entidades

Bandera	Anotación	Descripción/Ejemplo
direccional sap:direccionable	Esto indica si el conjunto de entidades relacionadas se puede direccionar directamente a través de la URI (por ejemplo, ProductSet) o si debe usar una propiedad de navegación para direccionarlo a través de una entrada diferente, por ejemplo, SalesOrderSet('1')/Items.	SalesOrderItemSet no se podría abordar directamente porque no tiene sentido Leer todos los artículos del pedido en todos los pedidos de venta. sin el encabezado de pedido respectivo.
sap:searchable	que permite búsquedas	Indica si la implementación del servicio (método QUERY) admite una cadena de búsqueda. Una cadena de búsqueda no filtra ninguna propiedad como tal, sino que proporciona la cadena de búsqueda a través del parámetro de búsqueda URI, por ejemplo, ProductSet?search='box'.
Suscriptible	N/A	Esto indica si un consumidor puede suscribirse a los cambios de las entidades subyacentes. El manejo de suscripciones y notificaciones requiere una configuración de sistema correspondiente (por ejemplo, bgRFC), así como implementaciones de salida de usuario o vínculos de eventos para enviar una notificación basada en un evento de backend.
Requiere Filtrar	sap:requiere filtro	Esto indica si una consulta requiere un filtro o no. Si se requiere un filtro, no se puede ejecutar una consulta sin un filtro. Por ejemplo, BusinessPartnerSet Obtiene todos los socios comerciales disponibles en el sistema, que pueden ser bastantes. Por lo tanto, tiene sentido obligar al consumidor a proporcionar un filtro.

Tabla 6.2 Atributos del conjunto de entidades (cont.)

En nuestro ejemplo, estableceremos los atributos Creable, Actualizable, Eliminable, Paginable y Direccionable (consulte la Figura 6.15).

6 Desarrollo de servicios

Entity Sets											
Name	Entity Type Name	L...	S...	Creatable	Updatable	Deletable	Pageable	Addressable	Searchable	Subscribable	Requires Filter
ProductSet	Product	T		<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>				

Figura 6.15 Conjunto de entidades ProductSet con atributos marcados

Una vez generado y activado el servicio (lo que se tratará en la Sección 6.2 y la Sección 6.4), puede ver cómo se representan los atributos en el documento de metadatos (consulte la Figura 6.16).

```
<?xml version="1.0" encoding="UTF-8"?>
<edmx:Edmx xmlns:sap="http://www.sap.com/Protocols/SAPData"
  xmlns:m="http://schemas.microsoft.com/ado/2007/08/dataservices/metadata"
  xmlns:edmx="http://schemas.microsoft.com/ado/2007/06/edmx" Version="1.0">
  - <edmx:DataService m:DataServiceVersion="2.0">
    - <Schema xml:lang="en" xmlns="http://schemas.microsoft.com/ado/2008/09/edm" sap:schema-version="1" Namespace="ZEXAMPLE1_SRV">
      - <EntityType sap:content-version="1" Name="Product">
        - <Key>
          <PropertyRef Name="ProductID"/>
        </Key>
        <Property Name="ProductID" sap:sortable="false" sap:updatable="false" sap:label="Product Identifier" MaxLength="10" Nullable="false" Type="Edm.String"/>
        <Property Name="Category" sap:sortable="false" sap:label="Product Category" MaxLength="40" Nullable="false" Type="Edm.String"/>
        <Property Name="Name" sap:sortable="false" sap:label="Product Name" MaxLength="255" Nullable="false" Type="Edm.String" sap:filterable="false"/>
      </EntityType>
      - <EntityContainer Name="ZEXAMPLE1_SRV_Entities" sap:supported-formats="atom json xlsx" m:IsDefaultEntityContainer="true">
        <EntitySet sap:content-version="1" Name="ProductSet" EntityType="ZEXAMPLE1_SRV.Product"/>
      </EntityContainer>
      <atom:link xmlns:atom="http://www.w3.org/2005/Atom"
        href="http://SAPGateway:50000/sap/opu/odata/sap/ZEXAMPLE1_SRV/$metadata"
        rel="self"/>
      <atom:link xmlns:atom="http://www.w3.org/2005/Atom"
        href="http://SAPGateway:50000/sap/opu/odata/sap/ZEXAMPLE1_SRV/$metadata"
        rel="latest-version"/>
    </Schema>
  </edmx:DataService>
</edmx:Edmx>
```

Figura 6.16 Ejemplo de documento de metadatos con atributos de conjunto de entidades

Este ejemplo fue relativamente fácil, ya que tenemos un solo tipo de entidad asignado a otro conjunto de entidades único. Como no hay varios tipos de entidad en este ejemplo, tampoco es necesario definir ninguna asociación todavía (esto cambiará más adelante).

Continúe leyendo para ver cómo importar un archivo EDMX más sofisticado con múltiples tipos de entidades, conjuntos de entidades, asociaciones y propiedades de navegación relacionadas.

Importación de un modelo de datos mediante XML de modelo de datos de entidad

También es posible importar un archivo EDMX ya existente. Esto se puede crear mediante cualquier conjunto de herramientas que sea capaz de modelar un modelo de datos de entidad (EDM) y exportarlo como un archivo EDMX. Un ejemplo de este tipo de conjunto de herramientas es el complemento OData Model Editor en SAP Web IDE. Otra fuente de archivos EDMX es cualquier servicio OData al que tenga acceso, ya que el documento de metadatos (\$metadata) proporciona un archivo que se puede utilizar como entrada.

Modelo de datos de entidad
XML

En este ejemplo, utilizará el documento de metadatos del servicio de muestra de SAP Gateway /IWBEPE/GWSAMPLE_BASIC. Para ello, debe almacenar temporalmente el archivo de metadatos en su sistema de archivos. Para importar un modelo de datos desde un archivo, primero debe crear un nuevo proyecto.

Comience creando un nuevo proyecto para importar un modelo de datos desde un archivo Crear nuevo proyecto (ver [Figura 6.17](#)).

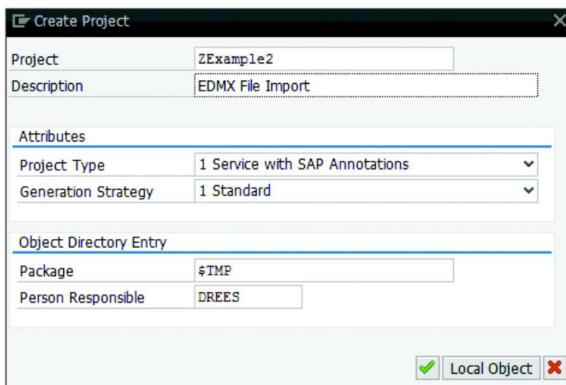


Figura 6.17 Creación de un nuevo proyecto para la importación de archivos

Después de haber creado un nuevo proyecto, haga clic derecho en Modelo de datos y elija Importar modelo de datos desde archivo (ver [Figura 6.18](#)).

Esto abre el paso 1 del asistente de importación de archivos ([consulte la Figura 6.19](#)). En SAP Gateway 2.0 SP 06 y versiones anteriores, la importación de archivos era una función de todo o nada. Solo se podía reemplazar el modelo de datos completo con el archivo importado. A partir de SAP Gateway 2.0 SP 07, esta función ha mejorado y ahora puede realizar una importación personalizada que le permite fusionar el modelo importado con su modelo existente.

6 Desarrollo de servicios

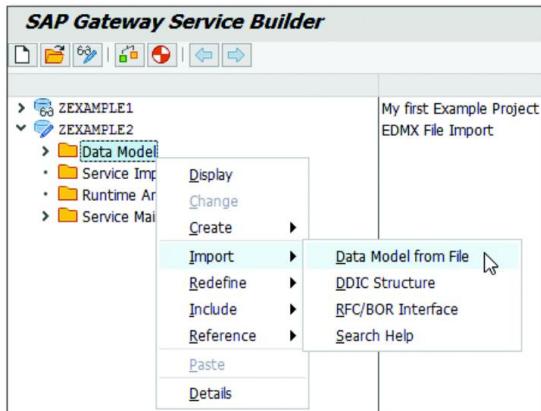


Figura 6.18 Importación de un modelo de datos desde un archivo

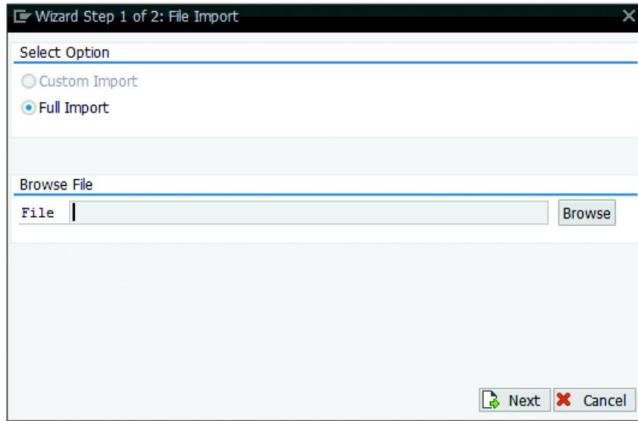


Figura 6.19 Paso 1 del Asistente de importación de archivos

Al seleccionar el botón Explorar se abre la opción Seleccionar un archivo de modelo para Cuadro de diálogo de importación donde selecciona el documento de metadatos o el archivo EDMX que desea importar (ver Figura 6.20).

Después de elegir el archivo, haga clic en Siguiente en el asistente de importación de archivos. Según la configuración de SAP GUI local, es posible que aparezca una pantalla emergente de seguridad de SAP GUI que le solicite que permita el acceso al archivo especificado. Debe permitirlo; de lo contrario, se cancelará la importación.

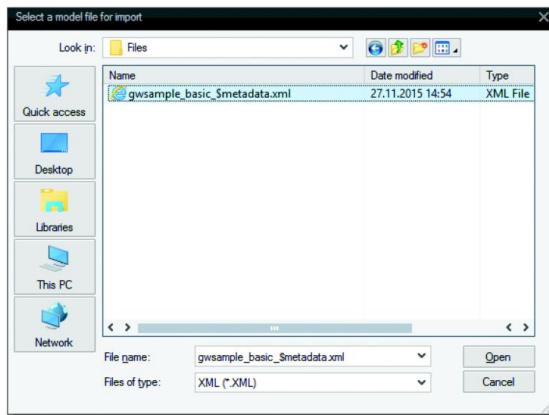


Figura 6.20 Cuadro de diálogo Seleccionar un archivo de modelo para importar

Se lee y verifica el archivo para garantizar que se proporcionó un archivo válido. En algunos casos, la validación puede fallar si el archivo del modelo contiene, por ejemplo, etiquetas o tipos de EDM no compatibles o inesperados. En este caso, debe ajustar el archivo del modelo de datos o elegir uno diferente.

En el paso 2 del asistente de importación de archivos (para una importación completa), verá una vista previa de cómo se verá el modelo de datos en su proyecto una vez que se complete el asistente ([consulte la Figura 6.21](#)). Haga clic en Finalizar para completar el asistente.

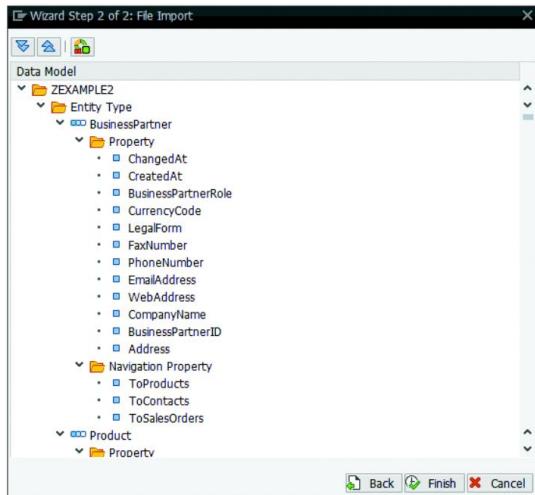


Figura 6.21 Paso 2 del Asistente de importación de archivos

6 Desarrollo de servicios

Importación de archivo exitosa

Una vez finalizada la importación del archivo, recibirá un mensaje de éxito: Archivo de modelo Importado correctamente. De forma predeterminada, se activa una comprobación del modelo que, en el caso actual (importación de los metadatos de /IWBE/P/GWSAM-PLE_BASIC), genera una serie de advertencias. Esto se debe a que la validación de metadatos de Service Builder se mejoró con el tiempo para garantizar que todo lo que desarrolle durante el tiempo de diseño se ejecute correctamente en tiempo de ejecución.

Dado que el archivo EDMX importado solo contiene información EDM pero no vínculo DDIC, es posible que los artefactos de tipo de datos ABAP derivados no encajen en el modelo importado. Si se producen advertencias o errores debido a esto, puede navegar desde la sección Mensajes haciendo doble clic directamente en el elemento erróneo para corregirlo.

Ahora puede expandir el árbol del proyecto y explorar los elementos importados (consulte la Figura 6.22). Para el servicio de ejemplo SAP Gateway, se han importado una gran cantidad de tipos de entidades, conjuntos de entidades, asociaciones, etc.

The screenshot shows the SAP Gateway Service Builder interface. On the left, the project tree displays a folder named 'EXAMPLE1' containing 'EXAMPLE2' and a 'Data Model' folder. The 'Data Model' folder is expanded, showing 'Entity Types' and 'Associations'. Under 'Entity Types', there are numerous entries such as 'BusinessPartner', 'Contact', 'Product', 'Navigation Properties', 'SalesOrder', 'SalesOrderLineItem', 'VH_AddressType', 'VH_BRole', 'VH_CRole', 'VH_Country', 'VH_Contry', 'VH_Language', 'VH_ProductTypeCode', 'VH_Sex', 'VH_UnitLength', 'VH_UnitQuantity', 'VH_UnitWeight', 'Complex Types', and 'Associations'. Under 'Associations', there are sets like 'BusinessPartnerSet', 'ContactSet', 'ProductSet', 'SalesOrderLineItemSet', 'SalesOrderSet', and 'VH_AddressTypeSet'. On the right, a large table titled 'Properties' lists various entity properties with columns for Name, Key, Edm Core Type, Prec., Scale, Max., Unit Prop., Cre., Upd., Sort., Null., Fit., Label, L., Comp. Type, ABAP Field Name, A., and Semantics. Many fields are marked with checkmarks in the first few columns, indicating successful import.

Figura 6.22 Modelo de datos después de la importación del archivo

Nota

El modelo de datos de la importación de archivos solo importa la metainformación en el servicio. No se importa ningún servicio. Esto debe hacerse después, ya sea a través del generador de servicios o implementando los métodos DPC relacionados en ABAP.

Una vez que se haya importado el modelo de datos, puede ajustarlo a sus necesidades. Por ejemplo, puede eliminar los elementos que no desea tener en el modelo final. También puede agregar nuevos elementos (por ejemplo, propiedades) al modelo si es necesario. Como se mencionó anteriormente, también puede volver a ejecutar la importación del modelo de datos si la fuente ha cambiado y desea aplicar esos cambios a su proyecto de Service Builder.

Importación de un modelo de datos a través del diccionario de datos

La tercera opción para crear una definición de modelo OData es importar una estructura o tabla DDIC. Para ello, se crea un nuevo proyecto (ZExample3), se hace clic con el botón derecho en Modelo de datos y se elige Importar estructura DDIC (consulte la Figura 6.23).

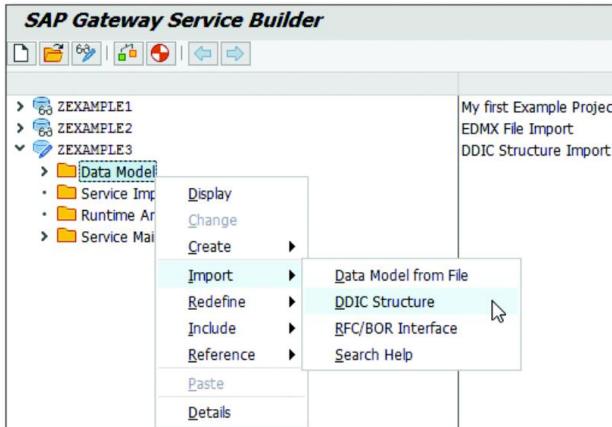


Figura 6.23 Inicio del Asistente de importación de estructura DDIC

Esto abre el paso 1 del asistente Importar desde la estructura DDIC (consulte la Figura 6.24). Aunque el asistente habla sobre las estructuras DDIC, también le permite importar la definición de datos desde las tablas DDIC y las vistas DDIC, incluidas las vistas CDS.

De manera predeterminada, la definición de DDIC se importa a un tipo de entidad. Como alternativa, puede seleccionar el botón de opción Tipo complejo para importar a un tipo complejo.

6 Desarrollo de servicios

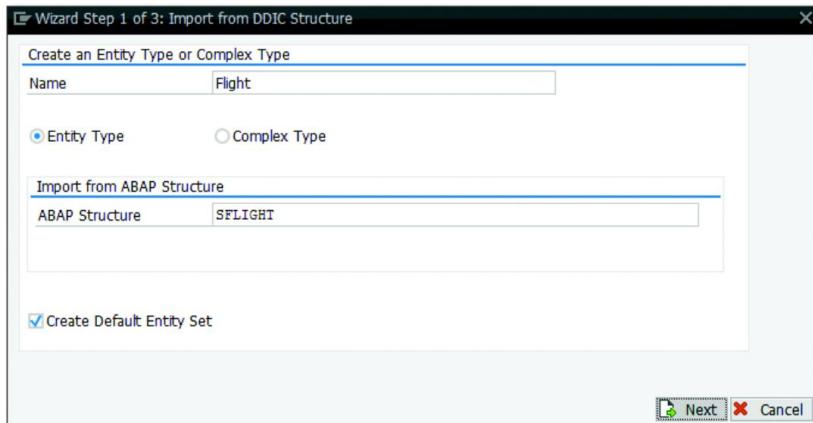


Figura 6.24 Paso 1 del Asistente para importar desde la estructura DDIC

En este ejemplo, ingresamos el nombre del tipo de entidad "VUELO". El ABAP La estructura sigue siendo la tabla de base de datos SFLIGHT. De forma predeterminada, la opción Crear La casilla de verificación Conjunto de entidades predeterminado está marcada para crear un conjunto de entidades junto con el tipo de entidad para vincularlos de inmediato.

Después de hacer clic en Siguiente, la definición del objeto DDIC proporcionado se muestra en el paso 2 del asistente (consulte la Figura 6.25).

The screenshot shows the second step of the wizard, listing parameters for the SFLIGHT structure. The 'Assign Structure' column contains checkboxes. The 'Description' and 'Type' columns provide details for each parameter. At the bottom are 'Back', 'Next', and 'Cancel' buttons.

Data Source Parameter	Assign Structure	Description	Type	Length	Decimals	Import	Search	Help	Search Help	Output L...
• <input checked="" type="checkbox"/> SFLIGHT	<input type="checkbox"/>	SFLIGHT								
• <input type="checkbox"/> MANDT	<input type="checkbox"/>	Client	CLNT	3						H_T000
• <input checked="" type="checkbox"/> CARRID	<input type="checkbox"/>	Airline	CHAR	3						H_SCARR
• <input checked="" type="checkbox"/> CONNID	<input type="checkbox"/>	Flight Number	NUMC	4						H_SPFLI
• <input checked="" type="checkbox"/> FLDATE	<input type="checkbox"/>	Date	DATS	8						
• <input checked="" type="checkbox"/> PRICE	<input type="checkbox"/>	Airfare	CURR	15	2					
• <input checked="" type="checkbox"/> CURRENCY	<input type="checkbox"/>	Airline Currency	CUKY	5						
• <input checked="" type="checkbox"/> PLANETYPE	<input type="checkbox"/>	Plane Type	CHAR	10						
• <input checked="" type="checkbox"/> SEATSMAX	<input type="checkbox"/>	Max. capacity econ.	INT4	10						
• <input checked="" type="checkbox"/> SEATSOC	<input type="checkbox"/>	Occupied econ.	INT4	10						
• <input checked="" type="checkbox"/> PAYMENTSUM	<input type="checkbox"/>	Total	CURR	17	2					
• <input checked="" type="checkbox"/> SEATSMAX_B	<input type="checkbox"/>	Max. capacity bus.	INT4	10						
• <input checked="" type="checkbox"/> SEATSOC_B	<input type="checkbox"/>	Occupied bus.	INT4	10						
• <input checked="" type="checkbox"/> SEATSMAX_F	<input type="checkbox"/>	Max. capacity 1st	INT4	10						
• <input checked="" type="checkbox"/> SEATSOC_F	<input type="checkbox"/>	Occupied 1st	INT4	10						

Figura 6.25 Paso 2 del Asistente para importar desde la estructura DDIC

En este paso, debe seleccionar los campos del objeto DDIC que desea importar como propiedades en su modelo de datos. Por lo general, primero debe marcar todos los campos y luego desmarcar los que no desea importar.

Por ejemplo, el campo de cliente (a veces llamado MANDT) no es necesario en su modelo porque, de manera predeterminada, su inicio de sesión se realiza en un cliente específico y el tiempo de ejecución ABAP agrega la información del cliente correspondiente a cada operación de base de datos.

También puede importar ayudas de búsqueda de DDIC relacionadas junto con la importación del propio objeto DDIC. Se ofrecen en función de la definición en el DDIC ABAP.

Después de seleccionar los campos del objeto DDIC que se importará, debe hacer clic en Siguiente. Esto conduce al paso 3 del asistente Importar desde la estructura DDIC (consulte la [Figura 6.26](#)), que le muestra un resumen del tipo de entidad/

Se creará un tipo complejo junto con las propiedades relacionadas. Primero, debe definir las propiedades clave de su tipo de entidad (los tipos complejos no tienen propiedades clave, por lo que todas las casillas de verificación están desmarcadas y la columna está en gris).

IsEn.. Complex/Entity Type	ABAP Name	Is K..	Type	Name	Label
✓ Flight	CARRID	<input checked="" type="checkbox"/>	CHAR	CarrierID	Airline
✓ Flight	CONNID	<input checked="" type="checkbox"/>	NUMC	ConnectionID	Flight Number
✓ Flight	FLDATE	<input checked="" type="checkbox"/>	DATS	FlightDate	Date
✓ Flight	PRICE	<input type="checkbox"/>	CURR	Price	Airfare
✓ Flight	CURRENCY	<input type="checkbox"/>	CUKY	Currency	Airline Currency
✓ Flight	PLANETYPE	<input type="checkbox"/>	CHAR	Planetype	Plane Type
✓ Flight	SEATSMAX	<input type="checkbox"/>	INT4	Seatsmax	Max. capacity econ.
✓ Flight	SEATSOCC	<input type="checkbox"/>	INT4	Seatsocc	Occupied econ.
✓ Flight	PAYMENTSUM	<input type="checkbox"/>	CURR	Paymentsum	Total
✓ Flight	SEATSMAX_B	<input type="checkbox"/>	INT4	SeatsmaxB	Max. capacity bus.
✓ Flight	SEATSOCC_B	<input type="checkbox"/>	INT4	SeatsoccB	Occupied bus.
✓ Flight	SEATSMAX_F	<input type="checkbox"/>	INT4	SeatsmaxF	Max. capacity 1st
✓ Flight	SEATSOCC_F	<input type="checkbox"/>	INT4	SeatsoccF	Occupied 1st

Figura 6.26 Paso 3 del Asistente para importar desde la estructura DDIC

Además, puede ajustar los nombres de propiedad precargados derivados del nombre de campo DDIC individual. Esto le permite cambiar el nombre rápidamente

6 Desarrollo de servicios

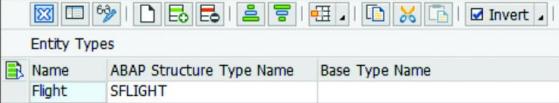
Las propiedades proporcionan nombres más significativos para el desarrollador de una aplicación de consumidor. Por ejemplo, los nombres de campo como CARRID (o Carrid) son difíciles de entender para cualquier desarrollador que no sea ABAP. En su lugar, es posible que desee proporcionar CarrierID. Además, cambie el nombre de Connid a ConnectionID y Fldate a FlightDate.

La última columna Etiqueta también le permite ajustar las etiquetas de propiedad. Están predeterminados con las etiquetas derivadas de los campos DDIC.

Tipo de entidad creado Cuando haya terminado, simplemente haga clic en Finalizar y se creará el tipo de entidad correspondiente en su proyecto.

Puede repetir el asistente tantas veces como desee para agregar más tipos de entidad y conjuntos de entidades a su proyecto.

El tipo de entidad creado se asocia automáticamente con la estructura/tabla DDIC relacionada que se utilizó en el asistente ([consulte la Figura 6.27](#)).

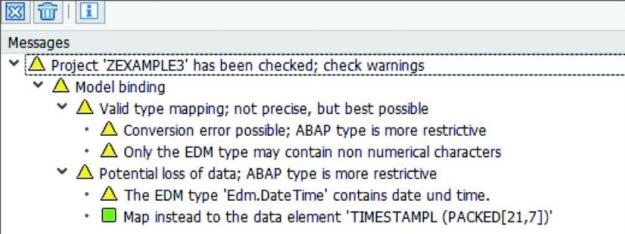


Entity Types		
Name	ABAP Structure Type Name	Base Type Name
Flight	SFLIGHT	

Figura 6.27 Tipo de entidad con una tabla DDIC asociada

Como se mencionó anteriormente, la referencia de tabla/estructura DDIC permite que el marco elija la definición de propiedad del campo de tabla/estructura subyacente. Esto incluye el tipo de datos, la longitud y la precisión, así como las etiquetas para todos los idiomas instalados.

Una vez finalizada la importación, se ejecutará una comprobación del proyecto. Al importar la estructura DDIC SFLIGHT, es posible que aparezcan algunos mensajes de advertencia ([consulte la Figura 6.28](#)).



Messages	
▼	Project 'ZEXAMPLE3' has been checked; check warnings
▼	Model binding
▼	Valid type mapping; not precise, but best possible
•	Conversion error possible; ABAP type is more restrictive
•	Only the EDM type may contain non numerical characters
▼	Potential loss of data; ABAP type is more restrictive
•	The EDM type 'Edm.DateTime' contains date and time.
•	Map instead to the data element 'TIMESTAMPL (PACKED[21,7])'

Figura 6.28 Advertencias después de la importación de DDIC

Al hacer doble clic en el mensaje, se accede al elemento relacionado. Por ejemplo, el mensaje Solo el tipo EDM puede contener valores no numéricos

Los caracteres indican que se utiliza un elemento de datos de tipo NUMC durante el tiempo de ejecución, mientras que el tipo de datos EDM es una cadena regular que no se limita únicamente a números.

Como consecuencia, el tipo de datos EDM puede contener caracteres no numéricos que no se pueden convertir en un campo NUMC en

tiempo de ejecución.

La otra advertencia indica que un campo de fecha ABAP, que solo almacena la fecha en sí pero no información de hora, está asignado a un Edm.DateTime. propiedad porque no hay ningún tipo de datos de solo fecha en los tipos EDM de OData 2.0.

Aparentemente, no hay una coincidencia del 100 % entre los tipos de datos ABAP y los tipos de datos EDM. Sin embargo, Service Builder intenta mapear las propiedades lo más cerca posible al importar desde cualquier fuente de datos. Si la asignación no se pudo ejecutar sin la posible pérdida de datos o si existe la posibilidad de que surjan problemas durante el tiempo de ejecución, le muestra advertencias o errores relacionados.

En nuestro ejemplo, es posible conservar las advertencias informadas y generar los artefactos de tiempo de ejecución del servicio OData (consulte la Sección 6.2). El enfoque recomendado es corregir las advertencias/errores informados y garantizar que los informes de verificación del proyecto tengan una luz verde.

Importación de un modelo de datos mediante RFC/BOR

La cuarta opción para importar un modelo de datos es la importación RFC/BOR. BOR permite importar la interfaz de un método BOR (por ejemplo, BusinessPartner.CreateFromData). RFC es un término más popular y permite la importación de una interfaz RFC. Los métodos BOR suelen apuntar a módulos de funciones remotas, por lo que las fuentes de datos BOR y RFC son bastante similares desde la perspectiva de Service Builder.

Creamos un nuevo proyecto y realicemos la importación del módulo de función remota BAPI_EPM_PRODUCT_GET_LIST. Inicie el asistente RFC haciendo clic derecho en Modelo de datos y eligiendo Importar interfaz RFC/BOR (consulte la Figura 6.29).

BAPI_EPM_
PRODUCTO_OBTENER_
LISTA

6 Desarrollo de servicios

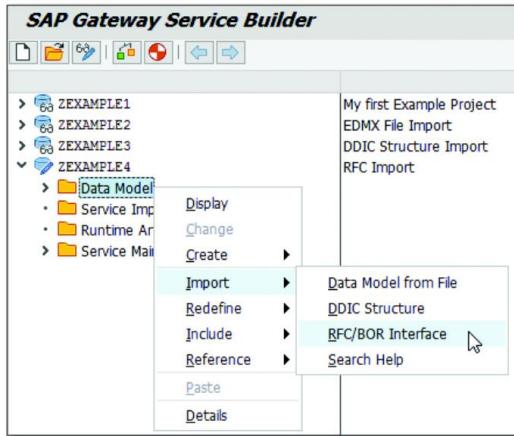


Figura 6.29 Inicio del asistente de importación de RFC/BOR

En la pantalla inicial, especifique el nombre del tipo de entidad. En este ejemplo, introduzca "Producto". En segundo lugar, especifique si la interfaz BOR/RFC que desea importar reside en el sistema local (local desde la perspectiva del componente SAP_GWFND/IW_BEP, que normalmente es el sistema backend de SAP ERP) o se encuentra en un sistema remoto. En este último caso, también debe proporcionar el destino RFC al sistema correspondiente.

BOR o RFC A continuación, debe indicar al asistente si desea importar la interfaz de un objeto BOR o de un módulo RFC. Por último, debe proporcionar el nombre del objeto que desea importar. En función del tipo especificado, obtendrá cuadros de diálogo de ayuda individuales (F4) que le permitirán buscar el objeto que está buscando. Esto también funciona en un escenario remoto.

En este ejemplo, cree el tipo de entidad "Producto" desde el sistema local importando la interfaz del RFC con el nombre "BAPI_EPM_PRO-DUCT_GET_LIST" (ver Figura 6.30).

De manera similar a otros asistentes de importación, está disponible una casilla de verificación Crear conjunto de entidades predeterminado que le permite crear un conjunto de entidades junto con el tipo de entidad.

Seleccione Siguiente cuando haya terminado con la primera página del asistente. En la segunda página (consulte la Figura 6.31), verá inicialmente la interfaz contraída del módulo RFC o el método BOR.

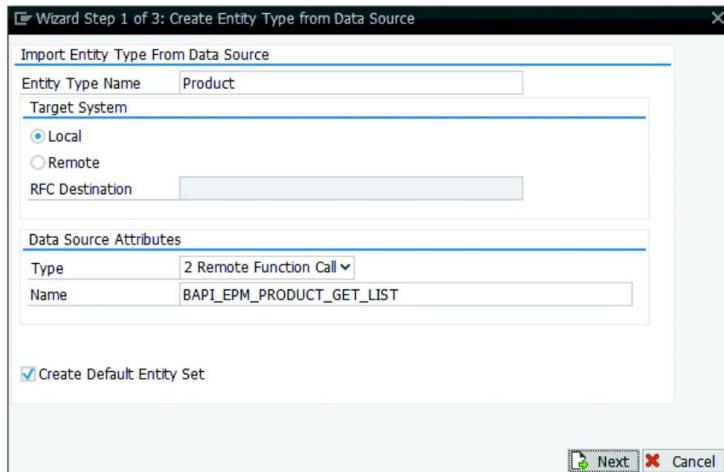


Figura 6.30 Asistente para crear un tipo de entidad a partir de una fuente de datos: Paso 1

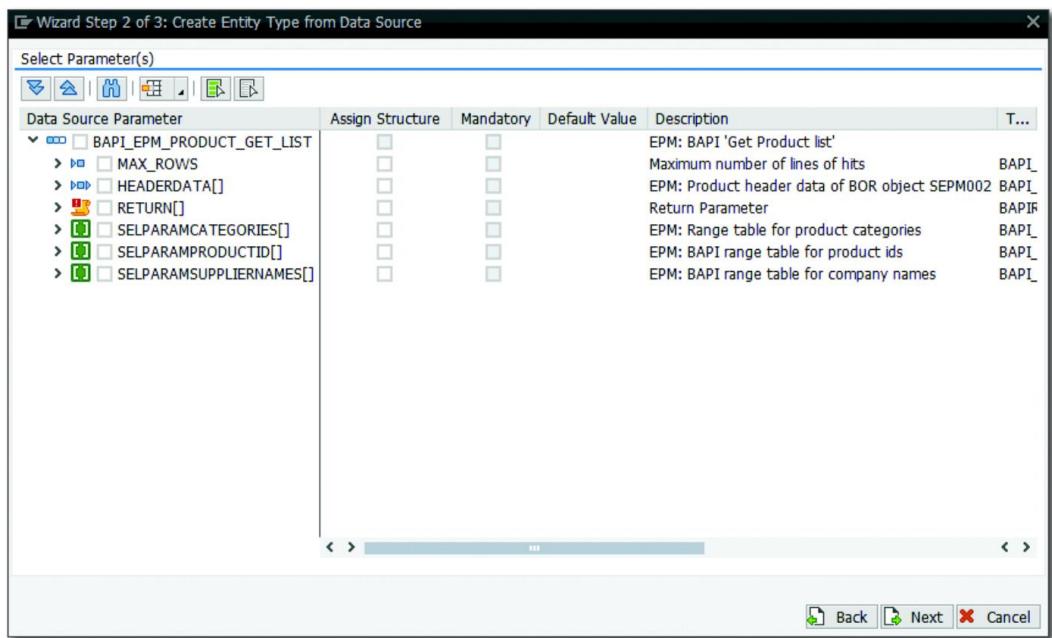


Figura 6.31 Asistente para crear un tipo de entidad a partir de una fuente de datos: paso 2

6 Desarrollo de servicios

Parámetros Cada parámetro de la interfaz tiene un ícono individual que describe el tipo de los siguientes parámetros:

- Parámetro de importación
- Parámetro de exportación
- Cambiar parámetro
- Parámetros de tablas
- Parámetro de retorno de BAPI
- Seleccionar parámetro de opciones

Los primeros cuatro parámetros (importación, exportación, cambio, tablas) son los parámetros típicos que pueden tener los módulos de función. El parámetro de retorno BAPI es un parámetro de tabla especial basado en el tipo BAPIRET2. Este parámetro tiene un ícono individual porque el Service Builder maneja automáticamente la tabla de retorno si modela también el aprovisionamiento del servicio con el Service Builder (por ejemplo, la operación GetEntitySet (Query)). No es necesario definir un tipo de entidad de retorno en su modelo de datos para exponer la información de error al consumidor; esto se hace automáticamente por usted leyendo la tabla de retorno y lanzando la excepción relacionada (por ejemplo, se lanzará una excepción de error comercial en caso de un mensaje "E" en la tabla de retorno). Este error da como resultado un código de retorno HTTP correspondiente, junto con el texto largo de error en la respuesta.

Nota

Este manejo automático de errores solo se lleva a cabo si modela la prestación del servicio mediante la asignación RFC/BOR. Si implementa usted mismo los métodos DPC correspondientes en ABAP, deberá verificar usted mismo cualquier código de retorno o tabla de retorno y generar la excepción relacionada.

Las opciones de selección o los parámetros de rangos se resaltan con un ícono verde entre corchetes. Tampoco es necesario agregar tablas de rangos como tipos de entidad en su modelo de datos porque deben asignarse con los parámetros de filtro correspondientes en la implementación del servicio y no a través de tipos de entidad individuales. Las tablas de rangos se identifican en función de su estructura DDIC con los campos típicos SIGN y OPTION , así como un campo LOW y un campo HIGH para el parámetro de filtro relacionado.

El asistente Crear tipo de entidad a partir de fuente de datos le permite expandir la interfaz contraída para explorar los parámetros, subestructuras o tablas anidadas subyacentes. Este ejemplo utiliza una BAPI que, por definición, no puede tener una interfaz anidada. Por lo tanto, solo encontrará campos “normales” en las estructuras de importación, exportación y tabla. Para definir un tipo de entidad para productos, marque el parámetro de tabla HEADERDATA[] (consulte la Figura 6.32).

Después de seleccionar el parámetro de tabla, la casilla de verificación Asignar estructura correspondiente se marca automáticamente. Esta casilla de verificación es importante porque asociará la estructura DDIC subyacente con el tipo de entidad creado. Si desmarca la casilla de verificación, no se realizará ninguna asignación.

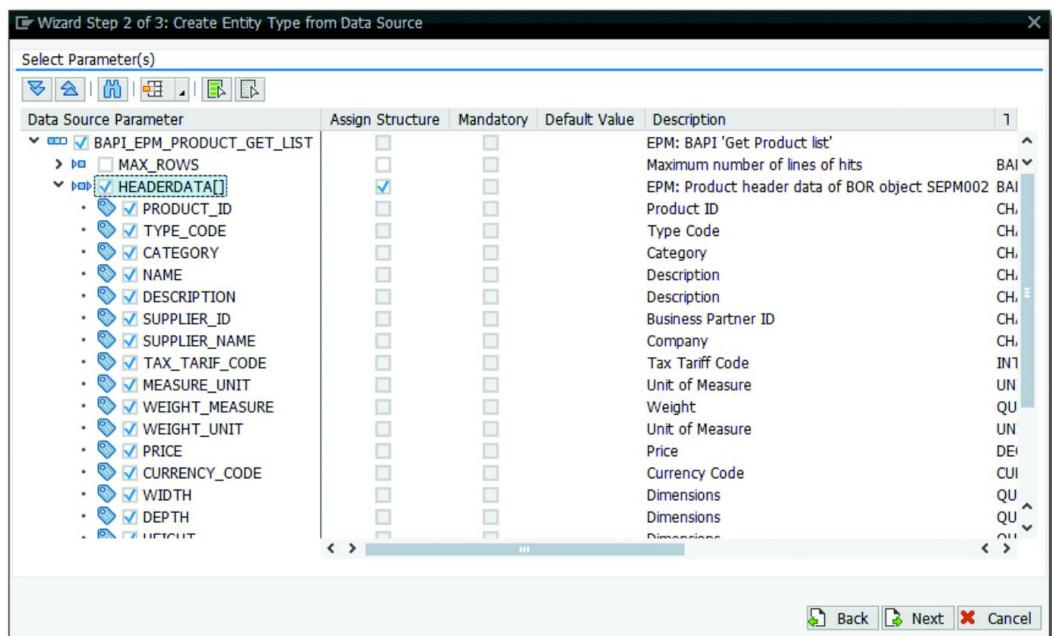


Figura 6.32 Crear un tipo de entidad a partir de una fuente de datos: selección de un parámetro de tabla

Una vez que haya terminado con la selección de los parámetros de la interfaz, puede acceder a la pantalla Resumen. Seleccione Siguiente. Esto muestra la pantalla final de Crear tipo de entidad.

desde el asistente de origen de datos (consulte la Figura 6.33). Esta es la pantalla de resumen donde puede ver qué conjuntos de entidades y/o tipos complejos se crearán.

6 Desarrollo de servicios

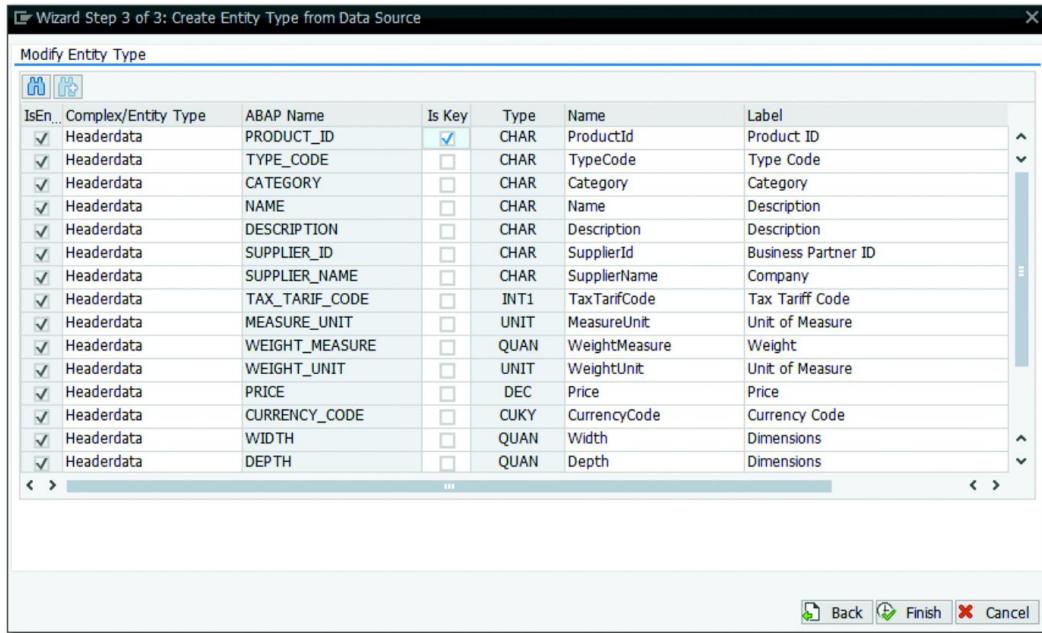


Figura 6.33 Asistente para crear un tipo de entidad a partir de una fuente de datos: paso 3

En este ejemplo, solo está marcado el parámetro de tablas Headerdata, que selecciona automáticamente todos los campos que se encuentran debajo. Por lo tanto, el asistente solo genera un único tipo de entidad con el nombre HEADERDATA y una propiedad para cada parámetro de la tabla de interfaz. El nombre del tipo de entidad se deriva del nodo que ha seleccionado.

HEADERDATA El nombre “Producto” que definió en la primera pantalla del asistente se ignora en este caso porque se seleccionó el nodo Headerdata. Si selecciona todos los parámetros (o un subconjunto) bajo el nodo Headerdata pero no el nodo Headerdata en sí, el asistente utiliza el nombre del tipo de entidad definido en la primera pantalla. La razón de este comportamiento es la capacidad del asistente RFC/BOR de crear múltiples tipos de entidad a partir de múltiples tablas de interfaz en un solo paso. Es por eso que se utiliza el nombre del elemento de interfaz para el nombre del tipo de entidad.

Verificar la primera columna Antes de hacer clic en el botón Finalizar, siempre debe verificar que la primera columna (IsEntity) esté marcada como se esperaba. En este ejemplo, esta

El campo está marcado para todas las entradas. Esto es correcto e indica que todas las propiedades terminarán en un tipo de entidad y no en un tipo complejo. El nombre del tipo de entidad resultante se muestra en la siguiente columna (editable), Nombre del tipo de entidad/complejo. Si lo desea, puede cambiar el nombre del tipo de entidad resultante modificando todas las columnas; sin embargo, este no es el enfoque recomendado porque lleva tiempo ajustar todas las propiedades una por una. Es mucho más fácil dejar que el asistente genere el tipo de entidad y luego simplemente cambiar el nombre a través del árbol del proyecto.

Si la columna IsEntity no está marcada, las propiedades terminarán en un tipo complejo (o en varios tipos complejos). Este es el caso, por ejemplo, si selecciona estructuras de importación o exportación (no tablas) de interfaces RFC/BOR.

Además, debes echar un vistazo a la columna Es clave y marcar esas columnas Clave propiedades que definen la clave principal del tipo de entidad resultante (los tipos complejos no tienen campos de clave). Si no marca ninguna propiedad como Es Clave: recibirás un mensaje correspondiente porque se activa automáticamente una verificación de proyecto cuando finalizas el asistente y un tipo de entidad sin una clave principal se considera un tipo de entidad erróneo.

La siguiente columna editable, Nombre, define el nombre de cada propiedad generada en su modelo. Esto es lo que el mundo exterior verá al consumir su servicio OData. Editar esta lista le permite cambiar rápidamente el nombre de todas las propiedades. Cada propiedad también tiene un nombre ABAP (consulte la columna Nombre ABAP), que no se puede editar en esta vista. El nombre ABAP es el nombre del campo que ve en el lado ABAP al procesar una consulta o cualquier otra operación. Es importante distinguir esto porque el nombre ABAP, especialmente cuando se deriva de una interfaz RFC o una estructura DDIC, a veces parece un poco críptico, mientras que el nombre de la propiedad se puede definir de una manera más fácil de usar para el consumidor (por ejemplo, Sales-OrderID en lugar de VBELN).

En la última columna, Etiqueta, puedes definir la etiqueta de tu propiedad. Si Definir etiqueta Un objeto RFC/BOR utiliza una estructura DDIC subyacente, el tipo de entidad está vinculado al elemento DDIC relacionado y, por lo tanto, no es necesario cambiar la etiqueta aquí. Si desea anular una etiqueta que viene a través del vínculo DDIC, puede hacerlo editando las propiedades en la descripción general de propiedades a través del árbol del proyecto (después de haber completado el asistente).

6 Desarrollo de servicios

Después de hacer clic en el botón Finalizar, se crean los tipos de entidad, los conjuntos de entidades y/o los tipos complejos. Se ejecuta una comprobación del proyecto para verificar si hay algún problema con la definición del proyecto actual (consulte la Figura 6.34).

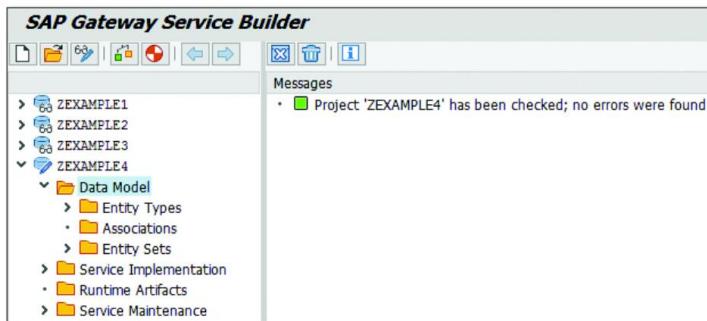


Figura 6.34 Comprobación del proyecto después de la importación de RFC/BOR

Un error típico que se muestra después de esta comprobación es el olvido de la definición de la clave principal. Si esto sucede, simplemente haga doble clic en el error y accederá al elemento erróneo.

En este ejemplo, se creó el tipo de propiedad HEADERDATA aunque especificamos "Producto" en la primera página del asistente. Esto se debe a que seleccionamos toda la tabla HEADERDATA desde la interfaz y no campos individuales. Puede cambiar fácilmente el nombre del tipo de entidad haciendo doble clic en él en el árbol del proyecto y cambiándolo a "Producto" en la cuadrícula de SAP List Viewer (ALV). De manera similar, puede navegar hasta el conjunto de entidades creado y cambiarle el nombre a, por ejemplo, "Conjunto de productos".

6.2 Registro de servicios en el sistema SAP Business Suite

Cuatro proyectos Si has seguido los ejemplos anteriores, ahora tienes cuatro proyectos en Su generador de servicios:

Ejemplo ZEjemplo1

Tipo de entidad de producto creado manualmente.

Ejemplo ZEjemplo2

Modelo de datos importado del servicio /IWBEPE/GWSAMPLE_BASIC.

Ejemplo ZEjemplo3

Modelo de datos importado de la estructura DDIC de SFLIGHT.

Ejemplo 4

Modelo de datos importado del RFC BAPI_EPM_PRODUCT_GET_LIST .

Ahora debe generar los artefactos de tiempo de ejecución y registrarlos en el sistema backend. Esto se hace generando el proyecto, que se activa al hacer clic en el botón Generar objetos de tiempo de ejecución (consulte la Figura 6.35).



Figura 6.35 Botón Generar objetos de tiempo de ejecución

La primera vez que haga clic en el botón Generar objetos de tiempo de ejecución, aparecerá la pantalla emergente de definición de modelo y servicio (consulte la Figura 6.36). Por lo general, se conservan los valores predeterminados y solo se hace clic en el botón Aceptar. Pero es importante comprender los valores predeterminados, así como el significado de los campos respectivos.

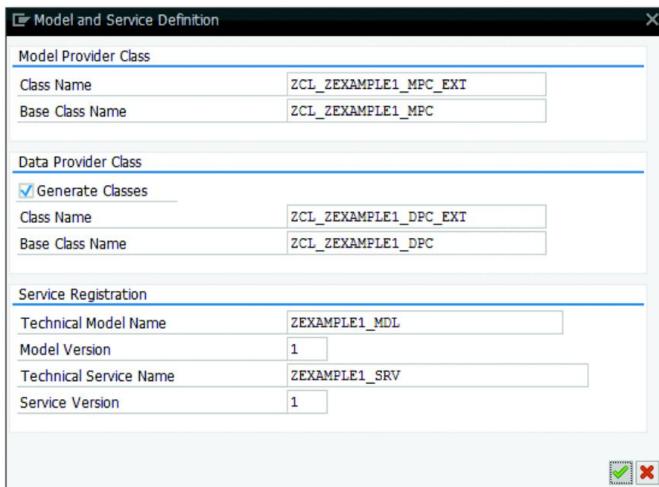


Figura 6.36 Pantalla emergente de definición de modelo y servicio

6 Desarrollo de servicios

Todas las clases ABAP que se generan se nombran con el patrón <name-space>CL_<project_name>_<suffix>. Normalmente, el espacio de nombres es solo uno de los espacios de nombres predeterminados "Z". Por lo tanto, en el proyecto ZExample1, los nombres de las clases generadas comenzarán con ZCL_ZEXAMPLE1.

MPC Como explicamos en el [Capítulo 5](#), el MPC es una clase ABAP que proporciona la representación en tiempo de ejecución de la definición del modelo. Como tal, toda la información del modelo que ha definido en su proyecto se genera en el MPC. En consecuencia, debe regenerar el MPC cada vez que cambie la definición del modelo en su proyecto.

Clase de extensión y clase base	<p>La definición del modelo se genera en dos clases: la clase de extensión (con el sufijo _MPC_EXT) y la clase base (con el sufijo _MPC). Técnicamente, la clase base se deriva de /IWBEPEP/CL_MGW_PUSH_ABS_Superclase MODELO . La clase de extensión tiene como superclase a la clase base. La clase de extensión se registra a través del nombre del modelo técnico, lo que significa que la clase de extensión puede definir los métodos a redefinir y los métodos a heredar de la clase base.</p>
---------------------------------	---

Lo mismo se aplica al DPC. Allí también encontrará una clase de extensión (sufijo _DPC_EXT) y una clase base (sufijo _DPC). Los DPC son responsables de la representación en tiempo de ejecución de la implementación de su servicio. Por ejemplo, los DPC ejecutan CREATE, READ, UPDATE, DELETE, QUERY y muchas más operaciones.

La clase base DPC hereda de /IWBEPEP/CL_MGW_PUSH_ABS_DATA superclase. La clase de extensión DPC hereda de la clase base DPC. La clase de extensión DPC se registra a través del nombre del servicio técnico, por lo que la clase de extensión se ejecuta en su servicio OData.

Las clases de extensión (MPC_EXT y DPC_EXT) le permiten agregar su propia codificación a la definición del modelo y a la implementación del servicio. Esto es necesario porque ciertos casos de uso (por ejemplo, el consumo de interfaces RFC complejas o el manejo de inserciones profundas) no se pueden definir completamente mediante el generador de servicios.

Las clases de extensión (MPC_EXT y DPC_EXT) se pueden cambiar a través de las herramientas de desarrollo típicas (por ejemplo, ABAP Class Builder), y Service Builder no sobrescribirá esas clases de extensión. La regeneración solo agrega nuevos métodos a las clases si es necesario (por ejemplo, si agrega una nueva

entidad establecida en su proyecto, los métodos CRUD-Q correspondientes se generan en la clase base DPC y, por lo tanto, se heredan en la clase de extensión DPC). Por lo tanto, puede realizar su propio desarrollo en las clases de extensión de forma segura.

Tenga en cuenta que los respectivos métodos CRUD-Q, incluido cualquier CRUD-Q manual Las redefiniciones creadas se eliminan tanto de la clase base DPC como de la clase de extensión si elimina el conjunto de entidades relacionadas de su proyecto.

Por lo tanto, debe manejar las eliminaciones de conjuntos de entidades con cuidado porque pueden provocar la pérdida de sus redefiniciones y, por lo tanto, la pérdida de su propia codificación ABAP.

Cada servicio de canal OData requiere un nombre de modelo técnico y un nombre de servicio técnico tanto en el backend como en el servidor central de SAP Gateway. Las definiciones del backend del nombre del modelo técnico también como nombre del servicio técnico se realizan junto con la generación de los MPC y DPC.

El nombre del modelo técnico básicamente solo apunta a la clase de extensión MPC. La definición del nombre del modelo técnico es igual al registro del modelo en el backend, que también se puede realizar a través de IMG (SAP Habilitación del servicio de puerta de enlace SAP de NetWeaver Backend OData Desarrollo de servicios de canal para el backend OData Channel Main- Modelos de contenedores). De forma predeterminada, cada modelo se registra con la versión 1.

El nombre del servicio técnico apunta a la clase de extensión DPC y es igual al nombre de su servicio externo (el nombre con el que consumirá su servicio OData). Además, el nombre del servicio técnico también apunta al nombre del modelo técnico porque la implementación del servicio debe tener una definición de modelo. La definición del nombre del servicio técnico es igual al registro del servicio, que también se puede realizar a través de IMG (SAP NetWeaver SAP Gateway Service Enablement Backend OData Channel Service Development for Backend

Servicios de mantenimiento del canal OData). De forma predeterminada, cada servicio se registra con la versión 1.

Después de hacer clic en Aceptar en la pantalla emergente Definición de modelo y servicio, Nombre del paquete Se le solicitará que proporcione el nombre del paquete para las clases generadas (consulte la Figura 6.37). Por lo general, debe asignar sus clases al mismo paquete que el proyecto.

6 Desarrollo de servicios

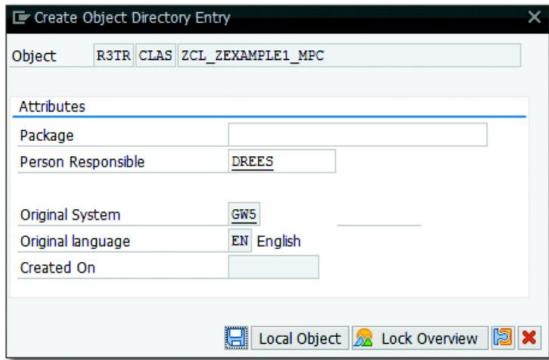


Figura 6.37 Entrada de directorio de objetos para la clase base MPC

Una vez que haya proporcionado toda la información, comienza la generación y el registro del backend. Según la cantidad de interfaces RFC diferentes y, lo que es más importante, el tamaño de las interfaces que haya definido en la implementación del servicio, la generación puede demorar un tiempo. En los ejemplos que se muestran aquí, actualmente no hay implementación del servicio, por lo que la generación se realiza rápidamente.

Ventana de mensajes Una vez finalizada la generación, aparece la ventana de mensajes (Figura 6.38). Debería mostrar solo semáforos verdes, lo que es su verificación de que la generación se realizó correctamente. Si la verificación o la generación del proyecto no se realizó correctamente, encontrará la información correspondiente en la ventana de mensajes.

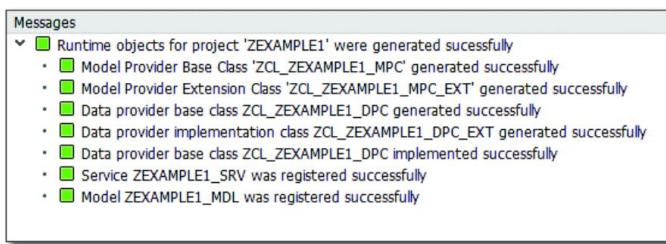


Figura 6.38 Ventana de mensajes después de la primera generación del proyecto

Una vez generado el proyecto, también verá que el tiempo de ejecución El nodo Artefactos de su proyecto está lleno. Si expande o hace doble clic en el nodo Artefactos en tiempo de ejecución, verá todos los artefactos que se han creado.

generado. En este ejemplo, hay dos DPC, dos MPC, el objeto de modelo y el objeto de servicio (consulte la Figura 6.39).

Runtime Artifacts					
Name	Generated Artifact Type	Program ID	Object Type	Object Name	
ZCL_ZEXAMPLE1_DPC	Data Provider Base Class	R3TR	CLAS	ZCL_ZEXAMPLE1_DPC	
ZCL_ZEXAMPLE1_DPC_EXT	Data Provider Extension Class	R3TR	CLAS	ZCL_ZEXAMPLE1_DPC_EXT	
ZCL_ZEXAMPLE1_MPC	Model Provider Base Class	R3TR	CLAS	ZCL_ZEXAMPLE1_MPC	
ZCL_ZEXAMPLE1_MPC_EXT	Model Provider Extension Class	R3TR	CLAS	ZCL_ZEXAMPLE1_MPC_EXT	
ZEXAMPLE1_MDL	Registered Model	R3TR	IWMO	ZEXAMPLE1_MDL	
ZEXAMPLE1_SRV	Registered Service	R3TR	IWSV	ZEXAMPLE1_SRV	

Figura 6.39 Artefactos de tiempo de ejecución de ZEXAMPLE1

Si tu proyecto crece, también encontrarás nuevas entradas en Runtime Arti- Artefactos en tiempo de ejecución Nodo de hechos. Por ejemplo, habrá una interfaz ABAP para cada módulo RFC/ método BOR que utilice en la implementación del servicio (por ejemplo, un módulo RFC que proporcione para la ejecución de la consulta). Como se mencionó anteriormente, el registro del modelo resultante es accesible a través de la IMG (consulte la Figura 6.40).

Display Model

Model

Model Information

Technical Model Name	ZEXAMPLE1_MDL
Model Version	1
Model Provider Class	ZCL_ZEXAMPLE1_MPC_EXT
Description	My first Example Project

Figura 6.40 Definición del modelo de ZEXAMPLE1_MDL

Lo mismo se aplica al registro del servicio (consulte la Figura 6.41). El registro del modelo y el registro del servicio juntos se denominan registro de back-end de un servicio de canal OData.

La generación de los MPC y DPC, así como el registro backend del modelo y del servicio concluye el registro del servicio OData en el backend.

6 Desarrollo de servicios

Display Service

Service Information

Technical Service Name	ZEXAMPLE1_SRV
Service Version	1
Description	My first Example Project
External Service Name	ZEXAMPLE1_SRV
Namespace	
Data Provider Class	ZCL_ZEXAMPLE1_DPC_EXT
Created By	DREES
Changed By	
Package	\$TMP

Extension for Service

Technical Service Name	
Service Version	0

Model Information

Technical Model Name	ZEXAMPLE1_MDL
Model Version	1
Description	My first Example Project
Model Provider Class	ZCL_ZEXAMPLE1_MPC_EXT
Created By	DREES
Changed By	
Package	\$TMP

[Create Model](#) [Unassign Model](#) [Assign Model](#) [Annotation Model](#)

Figura 6.41 Definición de servicio de ZEXAMPLE1_SRV

6.3 Generación de stub de servicio

La implementación del servicio realmente inyecta vida al servicio OData.

Hasta ahora, acaba de desarrollar el modelo de datos del servicio OData. La implementación del servicio conecta el modelo de datos con la lógica empresarial subyacente del backend (por ejemplo, proporcionada por un módulo de función RFC o cualquier otra lógica empresarial). Esto se explicará en detalle [en la Sección 6.5](#).

Según la Figura 6.1, mostrada anteriormente, en este punto, solo veremos el código auxiliar de implementación del servicio que generó el Generador de servicios como parte del paso de generación del proyecto [en la Sección 6.2](#).

Ahora, observemos brevemente la implementación del servicio para consumir el servicio OData y ver el resultado de los metadatos que se definieron en la Sección

6.1. Cada servicio de canal OData requiere un MPC y un DPC para funcionar, por lo que también debemos considerar la implementación del servicio antes de ejecutarlo.

Para ver el DPC generado, expanda el nodo Artefactos en tiempo de ejecución, a la derecha: DPC Haga clic en ZCL_ZEXAMPLE1_DPC y seleccione Ir a ABAP Workbench (ver Figura 6.42).

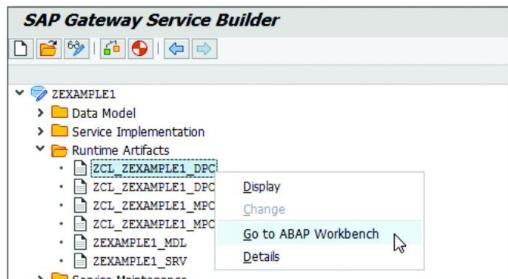


Figura 6.42 Visualización de la implementación de la clase base DPC generada

Esto abre el generador de clases ABAP con la clase base DPC. El generador de clases ABAP Los métodos que se muestran en negro se han redefinido; esto ocurre cada vez que genera su proyecto (ver Figura 6.43).

Class Interface		ZCL_ZEXAMPLE1_DPC	Implemented / Active							
			Properties	Interfaces	Friends	Attributes	Methods	Events	Types	Aliases
<input type="checkbox"/>	Parameter	<input type="checkbox"/>	Exception	<input type="checkbox"/>						
	Method						Level	Vis...	M...	Description
	/IWBEPIF_MGW_CONV_SRV_RUNTIME-SET_ETAG						Insta...	Pub...		Sets the etag to the HTTP response
	/IWBEPIF_MGW_CONV_SRV_RUNTIME-GET_DP_FACADE						Insta...	Pub...		Gets the Data Provider Facade
	/IWBEPIF_MGW_APPL_SRV_RUNTIME-GET_ENTITY						Insta...	Pub...		Execute a READ request (CReadUD) - ENTRY -
	/IWBEPIF_MGW_APPL_SRV_RUNTIME-GET_STREAM						Insta...	Pub...		Execute a READ request (CReadUD) - ENTRY -
	/IWBEPIF_MGW_APPL_SRV_RUNTIME-EXECUTE_ACTION						Insta...	Pub...		executes a function import
	/IWBEPIF_MGW_APPL_SRV_RUNTIME-UPDATE_ENTITY						Insta...	Pub...		Execute an UPDATE request (CRUpdated)
	/IWBEPIF_MGW_APPL_SRV_RUNTIME-PATCH_ENTITY						Insta...	Pub...		Execute a PATCH request (CRUpdated)
	/IWBEPIF_MGW_APPL_SRV_RUNTIME-CREATE_ENTITY						Insta...	Pub...		Execute a CREATE request (CreateRUD)
	/IWBEPIF_MGW_APPL_SRV_RUNTIME-CREATE_DEEP_ENTITY						Insta...	Pub...		Execute a deep insert CREATE request (CreateRUD)
	/IWBEPIF_MGW_APPL_SRV_RUNTIME-DELETE_ENTITY						Insta...	Pub...		Execute a DELETE request (CRUDelete)
	/IWBEPIF_MGW_APPL_SRV_RUNTIME-GET_ENTITYSET						Insta...	Pub...		Execute a READ request (CReadUD) - FEED -
	/IWBEPIF_MGW_APPL_SRV_RUNTIME-GET_EXPANDED_ENTITYSET						Insta...	Pub...		Execute a READ request (CReadUD) - FEED with inlines -

Figura 6.43 Clase base DPC

6 Desarrollo de servicios

A modo de ejemplo, haga doble clic en el archivo /IWBEPIF_MGW_APPL_SRV redefinido.

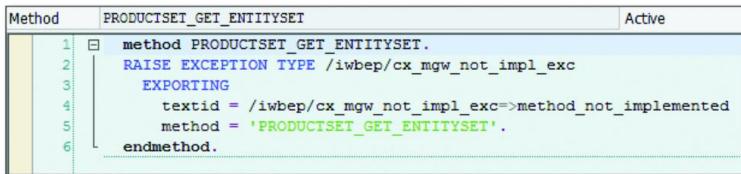
Nombre del método RUNTIME~GET_ENTITYSET (ver Figura 6.44).

Method	/IWBEPIF_MGW_APPL_SRV_RUNTIME~GET_ENTITYSET	Active
1	method /IWBEPIF_MGW_APPL_SRV_RUNTIME~GET_ENTITYSET.	
2	+	
3	*&--	
4	*& Include /IWBEPIF_MGW_APPL_SRV_RUNTIME~GET_ENTITYSET_BASE	
5	*& This class has been generated on 02.12.2015 16:27:26 in client 001	
6	*&*	
7	*& WARNING--> NEVER MODIFY THIS CLASS <--WARNING	
8	*& If you want to change the DPC implementation, use the	
9	*& generated methods inside the DPC provider subclass - ZCL_ZEXAMPLE1_DPC_EXT	
10	*&--	
11	DATA productset_get_entityset TYPE zcl_zexample1_mpc>tt_product.	
12	DATA lv_entityset_name TYPE string.	
13	lv_entityset_name = io_tech_request_context->get_entity_set_name().	
14		
15	CASE lv_entityset_name.	
16	+	
17	* EntitySet - ProductSet	
18	*	
19	WHEN 'ProductSet'.	
20	* Call the entity set generated method	
21	productset_get_entityset(
22	EXPORTING	
23	iv_entity_name = iv_entity_name	
24	iv_entity_set_name = iv_entity_set_name	
25	iv_source_name = iv_source_name	
26	it_filter_select_options = it_filter_select_options	
27	it_order = it_order	
28	is.paging = is.paging	
29	it_navigation_path = it_navigation_path	
30	it_key_tab = it_key_tab	
31	iv_filter_string = iv_filter_string	
32	iv_search_string = iv_search_string	
33	io_tech_request_context = io_tech_request_context	
34	IMPORTING	
35	et_entityset = productset_get_entityset	
36	es_response_context = es_response_context	
37)	

Figura 6.44 Método GET_ENTITYSET generado de la clase base DPC

Si observamos la codificación, podemos ver que hay una gran estructura de control de casos. Para cada conjunto de entidades, se llama a un submétodo independiente, que también se crea como parte del paso de generación. Esto sucede para las cinco operaciones CRUD-Q de cada conjunto de entidades.

- | | |
|---------------------|---|
| Negocio excepciones | De forma predeterminada, esos métodos generan excepciones comerciales (consulte la Figura 6.45). Esto ocurre porque la implementación real del servicio aún no ha tenido lugar, por lo que se lanza una excepción para manejar cualquier llamada de servicio a la operación respectiva. |
|---------------------|---|



```

Method PRODUCTSET_GET_ENTITYSET Active
1 method PRODUCTSET_GET_ENTITYSET.
2 RAISE EXCEPTION TYPE /iwbep/cx_mgw_not_implemented_exc
3 EXPORTING
4   textid = /iwbep/cx_mgw_not_implemented_exc=>method_not_implemented
5   method = 'PRODUCTSET_GET_ENTITYSET'.
6 endmethod.

```

Figura 6.45 Método PRODUCTSET_GET_ENTITYSET generado de la clase base DPC

6.4 Mantenimiento del servicio

El siguiente paso es el registro y activación del servicio OData en el hub SAP Gateway, también conocido como mantenimiento del servicio.

Registro y activación en el hub

fase. Esto se puede hacer manualmente ejecutando la transacción /IWFND/MAINT_SERVICE y utilizando el botón Agregar servicio en el sistema central SAP Gateway; sin embargo, es más fácil realizar el registro y la activación del servicio en el centro SAP Gateway directamente desde Service Builder en el backend.

Como requisito previo, debe mantener un sistema SAP Gateway Hub en Mantener en IMG

la IMG del sistema backend eligiendo SAP NetWeaver SAP

Habilitación del servicio de puerta de enlace Backend OData Canal Conne-

Configuración de la puerta de enlace SAP Configuración de la puerta de enlace SAP

En esta actividad de IMG, se especifica un nombre de sistema de destino, un cliente, un alias de sistema y un destino RFC a través del cual se puede acceder al sistema central.

Para una implementación integrada (cuando el backend y el hub SAP Gateway están en el mismo cuadro), aún es necesario mantener una entrada correspondiente.

En este caso, ingrese “NINGUNO” como destino RFC (consulte la Figura 6.46). _____



Figura 6.46 Sistema SAP Gateway registrado en el backend

6 Desarrollo de servicios

Asegúrese de ingresar un destino RFC para todas las entradas que mantenga.

De lo contrario, la entrada correspondiente no aparecerá en la interfaz de usuario de Service Builder.

Una vez realizado el mantenimiento del sistema SAP Gateway Hub en el IMG, encontrará una entrada correspondiente en cada proyecto en Mantenimiento de servicio nodo. (Asegúrese de reiniciar la transacción de Service Builder cada vez que cambie la configuración de SAP Gateway en la IMG porque Service Builder almacena en caché estos valores). Service Builder muestra información detallada en el lado derecho si hace doble clic en la entrada (consulte la Figura 6.47).

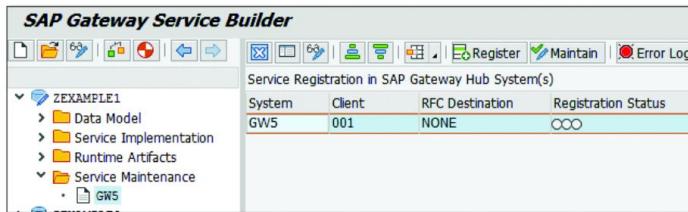


Figura 6.47 Estado del registro del servicio en SAP Gateway Hub

Ahora es el momento de registrar y activar el servicio en el hub SAP Gateway. Esto se realiza marcando el sistema del hub SAP Gateway y haciendo clic en el botón Registrar.

La primera vez que haga esto, aparecerá una pantalla emergente que le informará que la actividad se llevará a cabo en el sistema seleccionado (consulte la [Figura 6.48](#)). En otras palabras, el destino RFC proporcionado se utilizará para saltar al sistema central SAP Gateway de destino para registrar y activar su servicio allí, por lo que cada entrada debe tener un destino RFC.

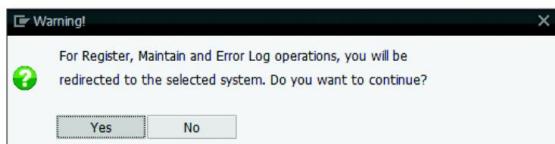


Figura 6.48 Advertencia de redirecciónamiento

Alias del sistema Si confirma la pantalla emergente, aparecerá la siguiente pantalla emergente para que proporcione el alias del sistema. Este alias del sistema se define en SAP

Sistema central de puerta de enlace que apunta al sistema de backend del que proviene. En una implementación integrada, normalmente se define un alias en la IMG de SAP Gateway que se denomina LOCAL. Este también utiliza RFC Destination NONE y se asegura de que la llamada permanezca en el sistema local. Después de haber seleccionado el alias del sistema ([consulte la Figura 6.49](#)), haga clic en el botón Aceptar.

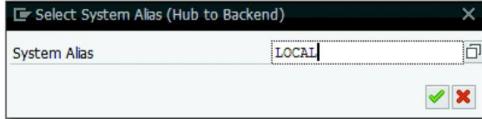


Figura 6.49 Definición de alias para la activación del servicio

La siguiente pantalla emergente proviene nuevamente del sistema central SAP Gateway y es la misma que ejecutar la Transacción /IWFND/MAINT_SERVICE en el sistema central SAP Gateway y hacer clic en el botón Agregar servicio (ver Figura 6.50).

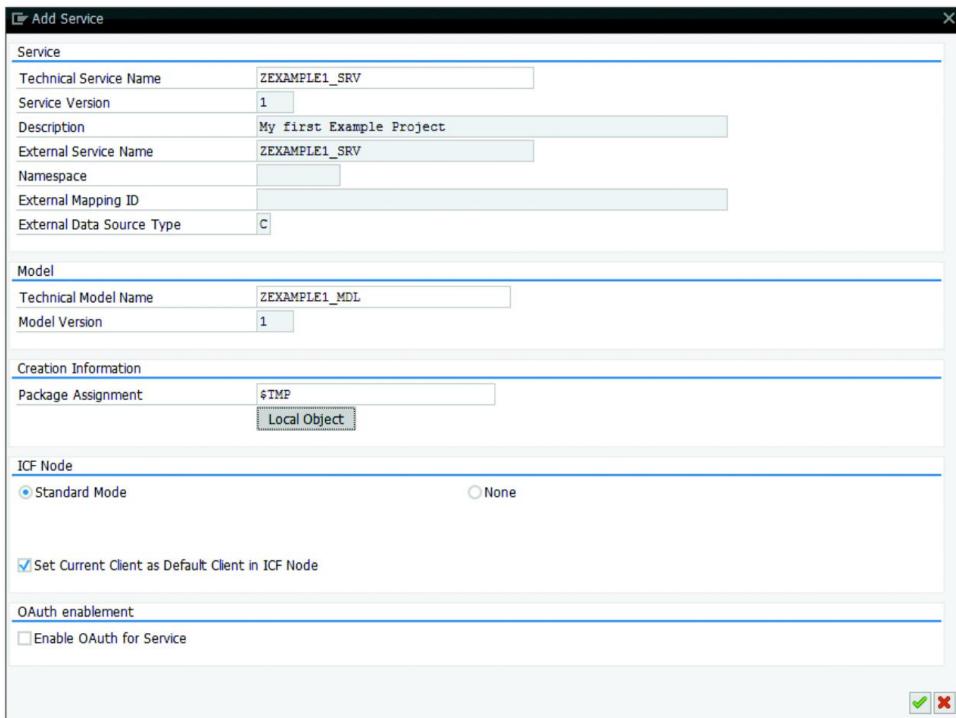


Figura 6.50 Cuadro de diálogo Agregar servicio en SAP Gateway Hub

6 Desarrollo de servicios

- Servicio técnico y nombre del modelo
- En esta pantalla, se define un nombre de servicio técnico y un nombre de modelo técnico en el lado del concentrador SAP Gateway. Estos se asignarán 1:1 al nombre del servicio y al nombre del modelo en el sistema backend definido durante el registro backend. El servicio externo
El nombre se deriva del nombre del servicio técnico en el backend que eligió anteriormente y no se puede cambiar en este cuadro de diálogo.
- Por lo general, se dejan los valores predeterminados como están y solo se proporciona un nombre de paquete para los objetos del repositorio que se crean. Después de hacer clic en el botón Aceptar, se lleva a cabo el registro y la activación del servicio en el sistema central SAP Gateway y se lo lleva de regreso al generador de servicios en el sistema backend.
- Si el registro y la activación del servicio fueron exitosos, la pantalla de descripción general del sistema que se muestra en la Figura 6.51 mostrará una luz verde.

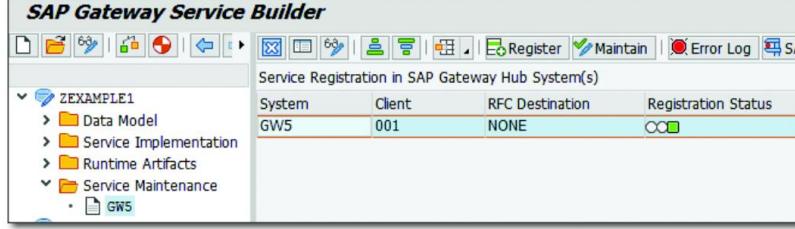
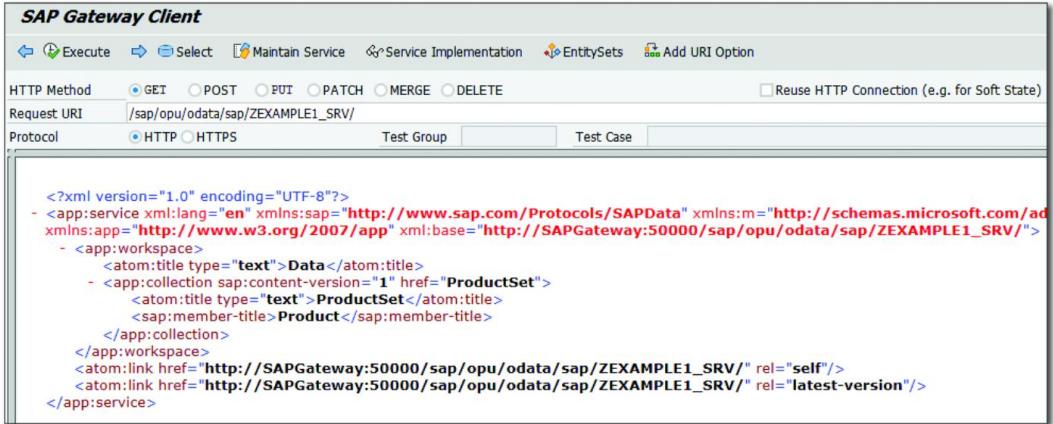


Figura 6.51 Cambio de estado después de la activación exitosa del servicio

El servicio OData ahora está registrado y activo en el sistema central de SAP Gateway, lo que significa que se puede utilizar. El nombre del servicio en la URI es el nombre del servicio externo que ha visto en Agregar servicio diálogo y se deriva del nombre del servicio técnico que especificó durante la primera generación del proyecto en Service Builder.

Transacción /IWFND/GW_CLIENTE
Puede utilizar el cliente SAP Gateway (Transacción /IWFND/GW_CLIENT) en el sistema central SAP Gateway para consumir el servicio. La URI es /sap/opu/odata/sap/ZEXAMPLE1_SRV. Si ejecuta esta URI en el cliente SAP Gateway, obtendrá el documento de servicio (consulte la Figura 6.52).

Si agrega /\$metadata, obtendrá el documento de metadatos del servicio de ejemplo (consulte la Figura 6.53).



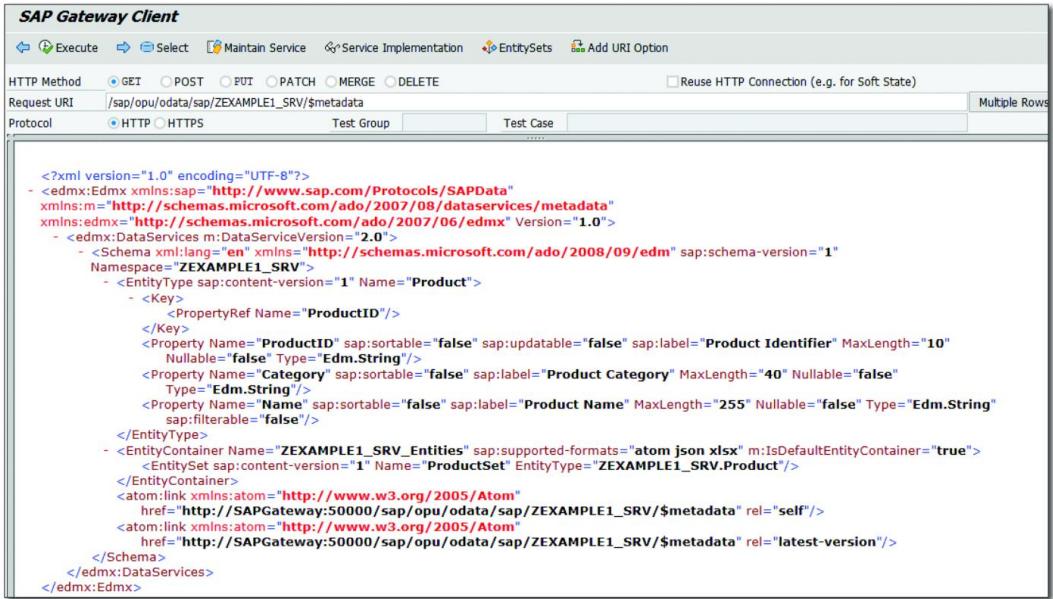
The screenshot shows the SAP Gateway Client interface. The top navigation bar includes 'Execute', 'Select', 'Maintain Service', 'Service Implementation', 'EntitySets', and 'Add URI Option'. Below this, the 'HTTP Method' is set to 'GET', 'Request URI' is '/sap/opu/odata/sap/ZEXAMPLE1_SRV/', and 'Protocol' is 'HTTP'. A checkbox for 'Reuse HTTP Connection (e.g. for Soft State)' is unchecked. The main content area displays the XML representation of the service document:

```

<?xml version="1.0" encoding="UTF-8"?>
- <app:service xml:lang="en" xmlns:sap="http://www.sap.com/Protocols/SAPData" xmlns:m="http://schemas.microsoft.com/ado/2007/08/dataservices/metadata" xmlns:app="http://www.w3.org/2007/app" xml:base="http://SAPGateway:50000/sap/opu/odata/sap/ZEXAMPLE1_SRV/">
  - <app:workspace>
    <atom:title type="text">Data</atom:title>
    - <app:collection sap:content-version="1" href="ProductSet">
      <atom:title type="text">ProductSet</atom:title>
      <sap:member-title>Product</sap:member-title>
    </app:collection>
  </app:workspace>
  <atom:link href="http://SAPGateway:50000/sap/opu/odata/sap/ZEXAMPLE1_SRV/" rel="self"/>
  <atom:link href="http://SAPGateway:50000/sap/opu/odata/sap/ZEXAMPLE1_SRV/" rel="latest-version"/>
</app:service>

```

Figura 6.52 Documento de servicio de ZEXAMPLE1_SRV



The screenshot shows the SAP Gateway Client interface. The top navigation bar includes 'Execute', 'Select', 'Maintain Service', 'Service Implementation', 'EntitySets', and 'Add URI Option'. Below this, the 'HTTP Method' is set to 'GET', 'Request URI' is '/sap/opu/odata/sap/ZEXAMPLE1_SRV/\$metadata', and 'Protocol' is 'HTTP'. A checkbox for 'Reuse HTTP Connection (e.g. for Soft State)' is unchecked. The 'Multiple Rows' button is selected. The main content area displays the XML representation of the metadata document:

```

<?xml version="1.0" encoding="UTF-8"?>
- <edmx:Edmx xmlns:sap="http://www.sap.com/Protocols/SAPData"
  xmlns:m="http://schemas.microsoft.com/ado/2007/08/dataservices/metadata"
  xmlns:edmx="http://schemas.microsoft.com/ado/2007/06/edmx" Version="1.0">
  - <edmx:DataServices m:DataServiceVersion="2.0">
    - <Schema xml:lang="en" xmlns="http://schemas.microsoft.com/ado/2008/09/edm" sap:schema-version="1">
      Namespace="ZEXAMPLE1_SRV"
      - <EntityType sap:content-version="1" Name="Product">
        - <Key>
          <PropertyRef Name="ProductID"/>
        </Key>
        <Property Name="ProductID" sap:sortable="false" sap:updatable="false" sap:label="Product Identifier" MaxLength="10" Nullable="false" Type="Edm.String"/>
        <Property Name="Category" sap:sortable="false" sap:label="Product Category" MaxLength="40" Nullable="false" Type="Edm.String"/>
        <Property Name="Name" sap:sortable="false" sap:label="Product Name" MaxLength="255" Nullable="false" Type="Edm.String" sap:hittable="false"/>
      </EntityType>
      - <EntityTypeContainer Name="ZEXAMPLE1_SRV_Entities" sap:supported-formats="atom json xlsx" m:IsDefaultEntityContainer="true">
        <EntitySet sap:content-version="1" Name="ProductSet" EntityType="ZEXAMPLE1_SRV.Product"/>
      </EntityTypeContainer>
      <atom:link xmlns:atom="http://www.w3.org/2005/Atom" href="http://SAPGateway:50000/sap/opu/odata/sap/ZEXAMPLE1_SRV/$metadata#self"/>
      <atom:link xmlns:atom="http://www.w3.org/2005/Atom" href="http://SAPGateway:50000/sap/opu/odata/sap/ZEXAMPLE1_SRV/$metadata#latest-version"/>
    </Schema>
  </edmx:DataServices>
</edmx:Edmx>

```

Figura 6.53 Documento de metadatos de ZEXAMPLE1_SRV

El servicio tiene un único conjunto de entidades (ProductSet), que también puedes consumir, pero como no has realizado ninguna implementación de servicio hasta el momento, aparecerá un mensaje de error. La implementación del servicio se explica a continuación.

6 Desarrollo de servicios

6.5 Implementación de servicios incrementales y mejora del modelo

Enfoque iterativo Ahora es el momento de implementar el servicio con un enfoque iterativo, como se muestra anteriormente en la Figura 6.1. Siempre que se consume un servicio OData, un punto de entrada típico es la ejecución de una consulta para obtener la colección/ Por ejemplo, el feed de socios comerciales o productos. Por eso, el get_ El método entityset se suele implementar primero en la mayoría de los casos. Esto le permite obtener un feed de un tipo de entidad y se explicará en detalle en la Sección 6.5.1. Desde el feed, recibe enlaces a cada recurso de entrada contenido. Es por eso que la lectura única (get_entity) se implementa típicamente después de get_entityset. La implementación de una lectura única se explica en detalle en la Sección 6.5.2.

Opciones de consulta Enriquecer un servicio OData con opciones de consulta potentes puede ser bastante importante porque permite a los consumidores, por ejemplo, solicitar solo las propiedades requeridas (\$select), filtrar el resultado según diferentes criterios (\$filter), realizar paginación del lado del cliente (\$top/\$skip), calcular la cantidad de entradas en un feed junto con el feed en sí (\$inlinecount) u ordenar el feed según sus necesidades (\$orderby). Todas estas opciones de consulta se explican en detalle con ejemplos de código en la Sección 6.5.3.

Las propiedades de navegación (también conocidas como enlaces de navegación) son elementos importantes de los servicios OData porque definen las navegaciones permitidas/ previstas entre elementos OData. Esto puede ser, por ejemplo, la navegación desde el encabezado de un pedido de venta (entrada) hasta la lista correspondiente de artículos de línea del pedido de venta (feed). La definición e implementación de las propiedades de navegación se explica en detalle en la Sección 6.5.4.

Si desea permitir el acceso de escritura a su conjunto de entidades, también debe implementar create_entity, update_entity y delete_entity. Métodos (métodos CUD). La implementación detallada de los métodos CUD se explica en la Sección 6.5.5.

En la Sección 6.5.6, explicamos y proporcionamos un ejemplo de implementación para las importaciones de funciones. Las importaciones de funciones se utilizan siempre que las operaciones CRUD-Q conocidas no sean suficientes y se requiera una función dedicada (por ejemplo, confirmar un pedido o rechazar una entrega) en un servicio OData.

Los recursos multimedia se pueden utilizar para exponer datos binarios, como por ejemplo gráficos, a los consumidores. En la Sección 6.5.7, explicamos cómo se puede definir un tipo de entidad como un tipo de entidad de enlace multimedia y qué se debe tener en cuenta en este caso.

\$expand es otra característica poderosa de los servicios OData que le permite recuperar múltiples entradas y/o feeds a lo largo de las propiedades de navegación definidas mediante una única llamada de servicio en lugar de ejecutar varias llamadas de servicio. Esto generalmente mejora el rendimiento significativamente porque, por ejemplo, se necesitan determinar menos datos en el backend.

La implementación detallada se explica en la Sección 6.5.8.

Una inserción profunda es básicamente lo opuesto a \$expand. En lugar de recuperar una estructura anidada, puede escribir una estructura anidada en el servidor SAP Gateway. Esto normalmente sustituye la necesidad de realizar varias llamadas de creación individuales. Una inserción profunda se puede utilizar, por ejemplo, si desea crear un encabezado de pedido de venta junto con una lista (feed) de artículos de línea de pedido de venta en una única llamada de creación. La Sección 6.5.9 explica en detalle cómo se puede implementar una inserción profunda.

Y, finalmente, la Sección 6.5.10 proporciona una introducción a \$batch, que se utiliza para agrupar múltiples llamadas de servicio independientes en una sola llamada por lotes.

Se puede utilizar, por ejemplo, para obtener datos de personalización que se proporcionan a través de múltiples conjuntos de entidades independientes.

Antes de comenzar a implementar su propia codificación, debe regenerar su proyecto para asegurarse de que los MPC y los DPC estén actualizados y coincidan con el estado actual de la definición de su proyecto. Esto también garantiza que no haya errores de definición en su proyecto (consulte la Figura 6.54).

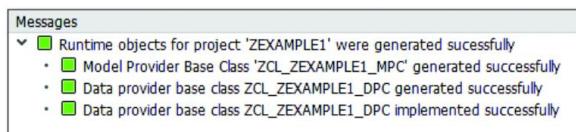


Figura 6.54 Regeneración exitosa del proyecto de muestra

6.5.1 Fuente (GET_ENTITYSET)

El get_entityset del conjunto de entidades ProductSet se implementa a través de Service Builder expandiendo el nodo Implementación de servicio y el

6 Desarrollo de servicios

Nodo ProductSet. A continuación, haga clic con el botón derecho en GetEntitySet (Consulta) y seleccione Ir a ABAP Workbench ([consulte la Figura 6.55](#)).

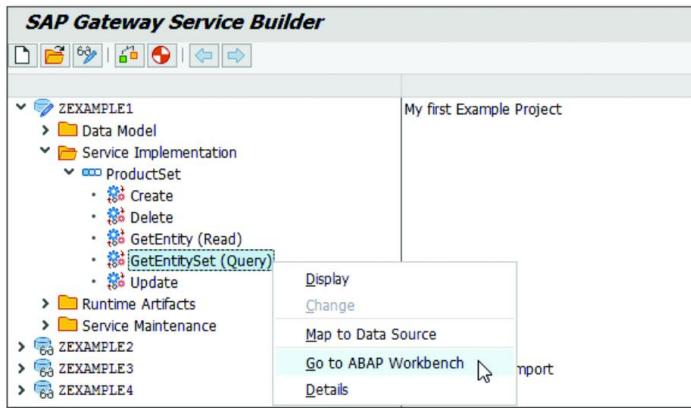


Figura 6.55 Navegación a la implementación de consultas en el Workbench ABAP

Esto abre un cuadro de diálogo que le informa que el método respectivo aún no se ha implementado (ver Figura 6.56).

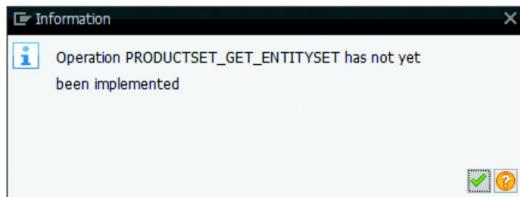


Figura 6.56 Pantalla de información: método aún no implementado

Al confirmar esta pantalla emergente, se abre la clase de extensión del proveedor de datos ZCL_ZEXAMPLE1_DPC_EXT en el generador de clases ABAP (Transacción SE24). Como todavía no se ha redefinido ningún método, todos los métodos se muestran con una fuente azul. La fuente azul es un indicador de que un método se hereda de la superclase (en este caso, la clase base DPC) y se ejecutará allí, si no se redefine.

Métodos derivados de interfaces

Encontrará una gran cantidad de métodos derivados de varias interfaces en la clase de extensión del proveedor de datos. El que observamos primero de forma indirecta es GET_ENTITYSET de la interfaz /IWBEPIF_MGW_APPL_SRV_RUNTIME .

Este método maneja todas las llamadas de consulta (Get Entityset) de todos los conjuntos de entidades de este servicio OData. Por eso, este método primero debe distinguir qué conjunto de entidades se está solicitando. Aquí es donde entra en juego nuevamente el Service Builder; genera una estructura de caso en cada uno de los cinco métodos CRUD-Q (CREATE_ENTITY, GET_ENTITY, UPDATE_ENTITY, DELETE_

ENTITY, GET_ENTITYSET) para manejar el conjunto de entidades individual solicitado llamando a un método independiente. Además de las estructuras de caso, el Service Builder genera cinco métodos CRUD-Q independientes para cada conjunto de entidades en la clase base de DPC, que también termina en la clase de extensión a través de la herencia.

tancia.

La separación de los cinco métodos CRUD-Q genéricos en métodos de instancia individuales (métodos protegidos) por conjunto de entidades proporciona una distinción clara de los métodos necesarios para gestionar las solicitudes OData. El nombre se concatena a partir del nombre del conjunto de entidades, así como del método CRUD-Q correspondiente. Si el nombre del método resultante supera los 30 caracteres, el nombre del conjunto de entidades se truncará para garantizar un nombre único en todo el DPC.

Los cinco métodos generados en el proyecto de ejemplo se enumeran aquí:

Cinco métodos

CONJUNTO_PRODUCTOS_CREAR_ENTIDAD

ENTIDAD_ELIMINADA_DE_CONJUNTO_PRODUCTOS

ENTIDAD_OBTENIDA_DEL_CONJUNTO_DE_PRODUCTOS

CONJUNTO_PRODUCTOS_OBTENER_CONJUNTO_ENTIDADES

ENTIDAD_DE_ACTUALIZACION_DEL_CONJUNTO_DE_PRODUCTOS

Para implementar el método Get_Entityset en el ejemplo, debe redefinir el método PRODUCTSET_GET_ENTITYSET .

Establezca la clase de extensión DPC en modo de edición presionando (Ctrl)+(F1) y desplácese hacia abajo hasta PRODUCTSET_GET_ENTITYSET (asegúrese de que esta línea esté seleccionada). Ahora haga clic en el botón Redefinir. Esto abre el editor ABAP y redefine el método PRODUCTSET_GET_ENTITYSET . De manera predeterminada, el generador de clases ABAP genera un fragmento de código que consta de una llamada comentada al mismo supermétodo. Reemplace la codificación existente con las líneas del Listado 6.1.

6 Desarrollo de servicios

MÉTODO productset_get_entityset.

```
DATOS: ls_headerdata TIPO bapi_epm_product_header,
       lt_headerdata TIPO TABLA ESTÁNDAR DE bapi_epm_product_
encabezamiento,
       ls_producto           COMO LÍNEA DE et_entityset.
```

FUNCTION DE LLAMADA 'BAPI_EPM_PRODUCT_GET_LIST'

* EXPORTACIÓN

* MAXIMO_FILAS =

Tablas

datos de encabezado	= lt_datos de encabezado
*	=
ID DE PRODUCTO SELPARAM	=
*	=
NOMBRESDEPROVEEDORESDESELPARAMS	=
*	=
CATEGORÍAS DE SELPARAMA	=
*	=
DEVOLVER	=

BUCLE EN lt_headerdata EN ls_headerdata.

ls_product-id_producto = ls_headerdata-id_producto.

ls_categoria-producto = ls_categoria-headerdata.

ls_nombre-producto = ls_nombre-data-header.

AÑADIR ls_product A et_entityset.

FIN DEL BUCLE.

MÉTODO FINAL.

[Listado 6.1 Conjunto de productos: método Get_Entityset](#)

Función de llamadas
módulo Como puede ver, el código de muestra realiza una llamada a la lógica empresarial subyacente (módulo de función BAPI_EPM_PRODUCT_GET_LIST) para recuperar la lista de productos. Este sencillo ejemplo no admite ningún criterio de filtro ni limita la cantidad de artículos leídos; simplemente obtiene todos los productos del Modelo de Adquisiciones Empresariales (EPM) existentes de la base de datos.

Después de recuperar los datos, debe asignarlos a la tabla de exportación. Para el método Get_EntitySet , esta tabla siempre se llama ET_ENTITYSET. Si observa la firma, verá que el tipo de esta tabla de exportación es ZCL_ZEX-AMPLE1_MPC=>TT_PRODUCT. Este es un tipo generado en la clase base MPC. Es importante saber esto porque cada vez que agrega o cambia alguna propiedad, debe asegurarse de regenerar su proyecto para ver este cambio en el MPC generado.

La asignación de la lista de productos (LT_HEADERDATA) a la tabla de exportación (ET_ENTITYSET) es bastante sencillo. Podría haber utilizado la instrucción ABAP move-corresponding si hubiera nombrado la propiedad productid con un guión bajo. Move-corresponding también podría haberse utilizado si hubiera implementado los proyectos que se han generado utilizando una importación DDIC o una importación de interfaz RFC porque, en esos casos, ET_ENTITYSET habría sido del mismo tipo de datos que LT_HEADERDATA. También se necesita mapeo si se utiliza la opción de importación EDMX porque entonces la tabla de exportación también habría sido un tipo generado como en el caso descrito que funciona con el tipo de entidad creado manualmente.

Siempre que encuentre algún problema al determinar el resultado, puede generar una excepción comercial /IWBEPCX_MGW_BUSI_EXCEPTION . Esto finaliza el procesamiento y arroja el código de error HTTP relacionado al consumidor junto con las declaraciones de error que proporcione.

Aumentar el negocio
excepción

Cuando haya terminado con la codificación, active los cambios haciendo clic en el botón Activación o presionando (Ctrl)+(F3) en el Generador de clases ABAP.

Asegúrese de activar todos los cambios que ha implementado (ver Figura 6.57).

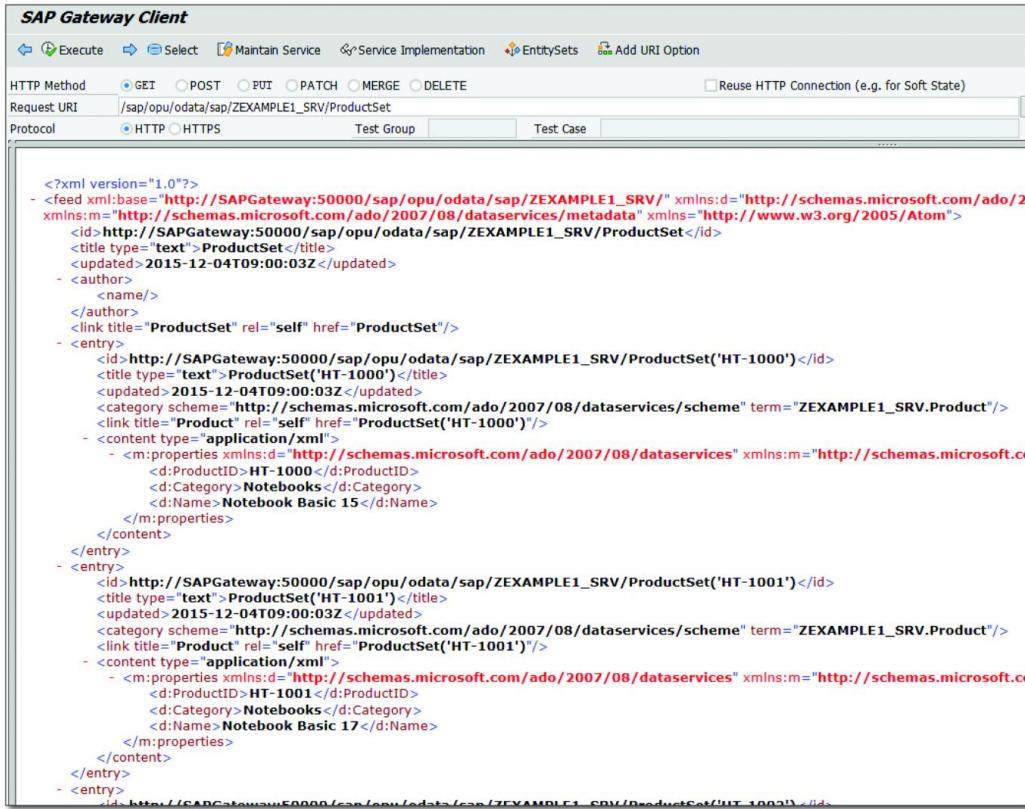
D..	Obj. ...	Object Name	User
CPRO	ZCL_ZEXAMPLE1_DPC_EXT	ZCL_ZEXAMPLE1_DPC_EXT	DREES
CPUB	ZCL_ZEXAMPLE1_DPC_EXT	ZCL_ZEXAMPLE1_DPC_EXT	DREES
METH	ZCL_ZEXAMPLE1_DPC_EXT	PRODUCTSET_GET_ENTITYSET	DREES

Figura 6.57 Activación de todos los cambios de su implementación

El proyecto Service Builder ya se generó anteriormente, por lo que no es necesario volver a generarlo.

A través del cliente SAP Gateway, ahora puede ejecutar la consulta ejecutando el URI /sap/opu/odata/sap/ZEXAMPLE1_SRV/ProductSet. Como se esperaba, obtiene el conjunto de entidades de producto (consulte la Figura 6.58).

6 Desarrollo de servicios



The screenshot shows the SAP Gateway Client interface. The top navigation bar includes 'Execute', 'Select', 'Maintain Service', 'Service Implementation', 'EntitySets', and 'Add URI Option'. Below this, the 'HTTP Method' dropdown is set to 'GET', and the 'Request URI' field contains '/sap/opu/odata/sap/ZEXAMPLE1_SRV/ProductSet'. The 'Protocol' dropdown is set to 'HTTP'. On the right, there is a checkbox for 'Reuse HTTP Connection (e.g. for Soft State)'.

The main content area displays the XML response for the ProductSet entity set. The XML structure is as follows:

```

<?xml version="1.0"?>
<feed xml:base="http://SAPGateway:50000/sap/opu/odata/sap/ZEXAMPLE1_SRV/" xmlns:d="http://schemas.microsoft.com/ado/2007/08/dataservices" xmlns:m="http://schemas.microsoft.com/ado/2007/08/dataservices/metadata" xmlns="http://www.w3.org/2005/Atom">
  <id>http://SAPGateway:50000/sap/opu/odata/sap/ZEXAMPLE1_SRV/ProductSet</id>
  <title type="text">ProductSet</title>
  <updated>2015-12-04T09:00:03Z</updated>
  <author>
    <name/>
  </author>
  <link title="ProductSet" rel="self" href="ProductSet"/>
  <entry>
    <id>http://SAPGateway:50000/sap/opu/odata/sap/ZEXAMPLE1_SRV/ProductSet('HT-1000')</id>
    <title type="text">ProductSet('HT-1000')</title>
    <updated>2015-12-04T09:00:03Z</updated>
    <category scheme="http://schemas.microsoft.com/ado/2007/08/dataservices/scheme" term="ZEXAMPLE1_SRV.Product"/>
    <link title="Product" rel="self" href="ProductSet('HT-1000')"/>
    <content type="application/xml">
      <m:properties xmlns:d="http://schemas.microsoft.com/ado/2007/08/dataservices" xmlns:m="http://schemas.microsoft.com/ado/2007/08/dataservices/metadata">
        <d:ProductID>HT-1000</d:ProductID>
        <d:Category>Notebooks</d:Category>
        <d:Name>Notebook Basic 15</d:Name>
      </m:properties>
    </content>
  </entry>
  <entry>
    <id>http://SAPGateway:50000/sap/opu/odata/sap/ZEXAMPLE1_SRV/ProductSet('HT-1001')</id>
    <title type="text">ProductSet('HT-1001')</title>
    <updated>2015-12-04T09:00:03Z</updated>
    <category scheme="http://schemas.microsoft.com/ado/2007/08/dataservices/scheme" term="ZEXAMPLE1_SRV.Product"/>
    <link title="Product" rel="self" href="ProductSet('HT-1001')"/>
    <content type="application/xml">
      <m:properties xmlns:d="http://schemas.microsoft.com/ado/2007/08/dataservices" xmlns:m="http://schemas.microsoft.com/ado/2007/08/dataservices/metadata">
        <d:ProductID>HT-1001</d:ProductID>
        <d:Category>Notebooks</d:Category>
        <d:Name>Notebook Basic 17</d:Name>
      </m:properties>
    </content>
  </entry>
  <entry>
    <id>http://SAPGateway:50000/sap/opu/odata/sap/ZEXAMPLE1_SRV/ProductSet('HT-1002')</id>
  
```

Figura 6.58 Conjunto de productos de ZExample1

Como puede ver en el resultado, la fuente OData proporciona un enlace para cada entrada que le permite navegar al recurso de entrada correspondiente (consulte la etiqueta entry-id). Debido a que aún no ha implementado el método de lectura única (get_entity), esos enlaces generarían un error si los ejecuta ahora.

6.5.2 Lectura única (GET_ENTITY)

De manera similar al método QUERY, puede desarrollar fácilmente la lectura única a través del generador de clases ABAP. Esta vez, expanda el nodo Service Implementation ProductSet, haga clic con el botón derecho en GetEntity (Read) y elija Ir a ABAP Workbench. Aparece una pantalla emergente de información porque el método aún no se ha implementado.

Esta vez, desplácese hacia abajo hasta el método PRODUCTSET_GET_ENTITY , coloque el cursor sobre él, cambie al modo de edición y haga clic en el botón Redefinir.

Reemplace el código generado con las líneas del Listado 6.2. _____

MÉTODO producto_obtener_entidad.

```
DATOS: ls_entity COMO er_entity,
       ls_product_id TIPO bapi_epm_product_id,
       ls_headerdata TIPO bapi_epm_product_header.
```

```
io_tech_request_context->obtener_claves_convertidas(
  IMPORTADOR
    es_key_values = ls_entity ).
```

ls_product_id-product_id = ls_entity-productid.

FUNCTION DE LLAMADA 'BAPI_EPM_PRODUCT_GET_DETAIL'

EXPORTADOR

id del producto	= ls_id_producto
-----------------	------------------

IMPORTADOR

datos de encabezado	= datos del encabezado ls
---------------------	---------------------------

*TABLA DE CUADROS

*	FACTORES DE CONVERSIÓN	=
---	------------------------	---

*	DEVOLVER	=
---	----------	---

er_entity-productid = ls_headerdata-product_id.

er_entity-categoría = ls_headerdata-categoría.

nombre_entidad_er = nombre-ls_headerdata.

MÉTODO FINAL.

Listado 6.2 Conjunto de productos: método Get_Entity

Lo primero que debe hacer en la codificación es determinar los valores de los campos clave para saber qué entrada proporcionar. El método Get_Entity direcciona un recurso de entrada única de la colección. Esto se hace proporcionando las propiedades clave como parte del URI entre paréntesis justo después del nombre del conjunto de entidades (por ejemplo, ProductSet('HT-1000')).

Determinar
campos clave

Los nombres de las propiedades clave deben proporcionarse como parte del URI si el tipo de entidad tiene más de una propiedad clave definida en sus metadatos.

Pero no es necesario concatenar manualmente ninguna URI con nombres de propiedades y valores clave porque el marco se encarga de ello generando los enlaces respectivos automáticamente.

6 Desarrollo de servicios

Propiedades de las teclas de acceso Puede acceder a las propiedades clave recorriendo la tabla de entrada IT_KEY_TAB. Sin embargo, no se recomienda utilizar esta tabla porque proporciona las propiedades clave en función del nombre de la propiedad externa. Este es el nombre que ve la aplicación del consumidor. El nombre interno (ABAP) de la propiedad puede ser diferente; esto depende de la definición de la propiedad en los metadatos. En nuestro ejemplo actual, los nombres de las propiedades interna y externa son iguales, por lo que no importa. Sin embargo, en general, es una buena práctica utilizar solo el nombre interno; por ejemplo, un cambio de nombre de una propiedad externa también requiere cambiar el código ABAP relacionado.

Uso de nombres de propiedades ABAP internas

Debe evitar utilizar los nombres de propiedades externas y, en su lugar, utilizar los nombres de propiedades ABAP internas. Esto se puede lograr utilizando io_referencia del objeto tech_request_context y los métodos GET proporcionados en lugar de los parámetros de la interfaz de importación de estructura o tabla simple (por ejemplo, IT_KEY_TAB o IV_SOURCE_NAME). Es por eso que esos parámetros de interfaz están marcados como "obsoletos" en la descripción del parámetro.

Esto es válido para todos los métodos (por ejemplo, GET_ENTITY, GET_ENTITYSET, CREATE_ENTIDAD, etc.) de la interfaz de aplicación /IWBEPIF_MGW_APPL_SRV_RUNTIME .

Las salidas de conversión también son consideraciones importantes. De forma predeterminada, las conversiones están habilitadas, lo que significa que los valores se convierten a su representación externa (por ejemplo, un número de artículo de línea "10" sin ceros iniciales en lugar de "0000000010") cuando se exponen. Esto también ocurre con los campos clave, que, por lo tanto, deben convertirse nuevamente para usarlos al determinar, por ejemplo, una determinada entrada como en este caso.

Si desea recuperar los valores clave en su representación externa, puede utilizar el método get_keys() de io_tech_request_context objeto. También puedes utilizar otro método llamado getConvertedKeys().

Como indica su nombre, este método proporciona las claves convertidas, lo que significa que se ejecutó la conversión de entrada y, por lo tanto, recibimos la representación interna (por ejemplo, con ceros a la izquierda).

Tenga en cuenta que las salidas de conversión se asignan en el nivel de propiedad en el método de definición del MPC. En nuestro ejemplo actual, no hemos asignado ninguna salida de conversión hasta el momento.

El método `get_CONVERTED_KEYS()` le proporciona los valores de clave en el formato de su tipo de entidad, que es una estructura. Por eso, define `ls_ENTITY` de tipo `ZCL_ZEXAMPLE1_MPC=>TS_PRODUCT` o simplemente como `er_ENTITY`. Por el contrario, `get_KEYS()` le proporciona las claves en una tabla de pares nombre/valor.

Una vez que haya determinado los valores clave necesarios, puede llamar a la función correspondiente para obtener los datos. En este caso, utilice `BAPI_EPM_PRODUCT_` Módulo de función `GET_DETAIL` para leer los detalles del producto. El resultado (`HEADERDATA`) se asigna luego a la estructura de retorno `ER_ENTITY` campo por campo, ya que el nombre de la propiedad interna `productid` y el nombre del campo en la estructura `BAPI product_id` son diferentes en nuestro ejemplo. Si los nombres de los campos son idénticos, por supuesto, puede utilizar la instrucción ABAP correspondiente para completar la estructura de retorno como se describió anteriormente al implementar el método `GET_ENTITYSET`.

La estructura de retorno es del tipo `ZCL_ZEXAMPLE1_MPC=>TS_PRODUCT`, que se genera en la clase base MPC.

Después de activar la clase de extensión DPC modificada, puede leer los detalles de, por ejemplo, el producto HT-1000 ejecutando la siguiente URI a través del cliente SAP Gateway: /sap/opu/odata/sap/ZEXAMPLE1_SRV/ProductSet('HT-1000').

Activar modificado
Clase DPC

Como era de esperar, ahora obtendrá una única entrada en lugar de una fuente OData (consulte la Figura 6.59).

The screenshot shows the SAP Gateway Client interface. In the top navigation bar, the 'Execute' button is highlighted. Below it, the 'HTTP Method' dropdown is set to 'GET'. The 'Request URI' field contains '/sap/opu/odata/sap/ZEXAMPLE1_SRV/ProductSet('HT-1000')'. The 'Protocol' dropdown shows 'HTTP' selected. On the right side of the interface, there is a large text area displaying the XML response:

```

<?xml version="1.0" encoding="UTF-8"?>
<entry xmlns:d="http://schemas.microsoft.com/ado/2007/08/dataservices" xmlns:m="http://schemas.microsoft.com/ado/2007/08/dataservices/metadata" xml:base="http://SAPGateway:50000/sap/opu/odata/sap/ZEXAMPLE1_SRV/">
  <id>http://SAPGateway:50000/sap/opu/odata/sap/ZEXAMPLE1_SRV/ProductSet('HT-1000')</id>
  <title type="text">ProductSet('HT-1000')</title>
  <updated>2015-12-04T09:27:56Z</updated>
  <category scheme="http://schemas.microsoft.com/ado/2007/08/dataservices/scheme" term="ZEXAMPLE1_SRV.Product"/>
  <link title="Product" rel="self" href="ProductSet('HT-1000')"/>
  <content type="application/xml">
    <m:properties>
      <d:ProductID>HT-1000</d:ProductID>
      <d:Category>Notebooks</d:Category>
      <d:Name>Notebook Basic 15</d:Name>
    </m:properties>
  </content>
</entry>

```

Figura 6.59 Lectura única del producto 'HT-1000'

6 Desarrollo de servicios

6.5.3 Opciones de consulta

Actualmente, el servicio de ejemplo puede proporcionar una lista sin filtrar de productos (feed/colección) así como las entradas relacionadas (obtener detalles, lectura única). En esta sección, analizaremos en detalle las diferentes opciones de consulta, como \$select, \$filter, \$top, \$skip, \$inlinecount y \$orderby. Le permiten seleccionar solo las propiedades que desea (\$select), filtrar el resultado (\$filter), realizar una paginación del lado del cliente (\$top y \$skip), cuentan el número de registros (\$inlinecount) y/o los ordenan (\$orderby).

\$seleccionar

Manejado automáticamente La opción de consulta \$select le permite seleccionar las propiedades que desea recibir en su fuente OData y en su lectura única. El marco maneja \$select automáticamente al reducir la lista de propiedades proporcionadas al consumidor OData, independientemente de los datos que haya determinado el proveedor de datos.

Si desea reaccionar ante una opción de consulta \$select de forma que, por ejemplo, solo lea y determine las propiedades que se han solicitado, debe hacerlo en la clase de extensión DPC (Get_Entity/

Métodos Get_EntitySet). Puede acceder a las propiedades \$select mediante el método io_tech_request_context->get_select() y, en versiones más nuevas, el método get_select_with_mandtry_fields() . Ambos métodos le proporcionan una tabla para todas las propiedades seleccionadas. El último se asegura de que también se incluyan todas las propiedades clave, incluso si no se han seleccionado en la URI.

Actuación mejora Reaccionar a una opción de consulta \$select puede tener una mejora significativa en el rendimiento si puede, por ejemplo, evitar el cálculo o la determinación de ciertas propiedades costosas (es decir, propiedades cuyos valores calculados requieren un esfuerzo significativo, como cálculos de precios, disponibilidad, etc.).

De todos modos , este ejemplo no se beneficiará al considerar la declaración \$select porque todas las propiedades están determinadas por el módulo de función subyacente.

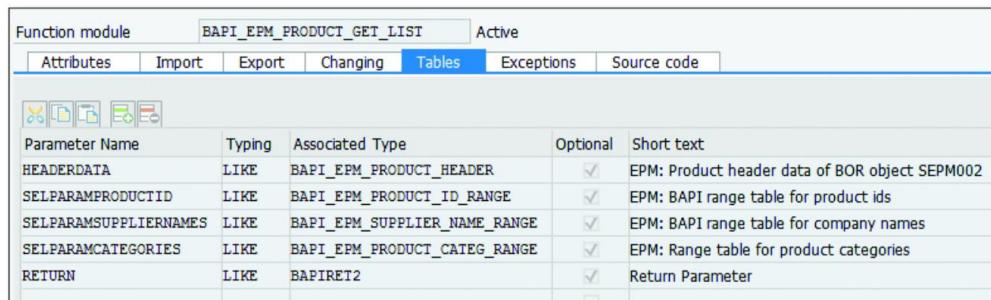
\$filtro

La opción de consulta \$filter es la opción de consulta más utilizada. Le permite filtrar el conjunto de resultados de su implementación de Get_EntitySet . De esta manera, puede evitar proporcionar una lista sin filtrar de todos los socios comerciales o pedidos de venta al consumidor (que puede ser una colección bastante grande con muchas entradas).

Usado con mayor frecuencia

También se puede indicar que un conjunto de entidades requiere un filtro. En este caso, debe proporcionar una opción de consulta \$filter en la URI. Como se mencionó anteriormente, esto es solo una anotación y ninguna característica del marco de trabajo la verifica.

El servicio de ejemplo aún no puede gestionar ningún criterio de filtro porque no ha hecho nada al respecto hasta ahora. Si observa la interfaz del módulo de función BAPI_EPM_PRODUCT_GET_LIST , verá que tiene parámetros de filtro de rango para categorías, ID de productos y nombres de proveedores (consulte la Figura 6.60).



The screenshot shows the SAP Function Module interface for BAPI_EPM_PRODUCT_GET_LIST. The top navigation bar has tabs for Attributes, Import, Export, Changing, Tables (which is selected), Exceptions, and Source code. Below the tabs is a toolbar with icons for search, refresh, and other functions. The main area displays a table of parameters:

Parameter Name	Typing	Associated Type	Optional	Short text
HEADERDATA	LIKE	BAPI_EPM_PRODUCT_HEADER	<input checked="" type="checkbox"/>	EPM: Product header data of BOR object SEPM002
SELPARAMPRODUCTID	LIKE	BAPI_EPM_PRODUCT_ID_RANGE	<input checked="" type="checkbox"/>	EPM: BAPI range table for product ids
SELPARAMSUPPLIERNAMES	LIKE	BAPI_EPM_SUPPLIER_NAME_RANGE	<input checked="" type="checkbox"/>	EPM: BAPI range table for company names
SELPARAMCATEGORIES	LIKE	BAPI_EPM_PRODUCT_CATEG_RANGE	<input checked="" type="checkbox"/>	EPM: Range table for product categories
RETURN	LIKE	BAPIRET2	<input checked="" type="checkbox"/>	Return Parameter

Figura 6.60 Interfaz de BAPI_EPM_PRODUCT_GET_LIST

Para utilizar un parámetro de filtro, debe mejorar la codificación en Mejorar codificación Clase de extensión DPC Get_EntitySet . Reemplace el método PRODUCTSET_GET_ENTITYSET con las líneas del Listado 6.3.

MÉTODO productset_get_entityset.

DATOS: filtro lr_filter, TIPO REF A /wbep/if_mgw_req_

```
lt_filter_select_options TIPO /wbep/t_mgw_select_
opción,
ls_filter_select_options TIPO /wbep/s_mgw_select_
opción,
opción ls_select TIPO /wbep/s_cod_select_
```

6 Desarrollo de servicios

opción,

<code>It_selparamproductid rango de id_del_producto_epm, ls_selparamproductid rango.</code>	TABLA ESTÁNDAR DE TIPO DE bapi_ TIPO bapi_epm_product_id_
---	--

DATOS: `It_headerdata
encabezado_producto_epm,
 ls_headerdata
 ls_producto`

<code>It_headerdata encabezado_producto_epm, ls_headerdata ls_producto</code>	TABLA ESTÁNDAR DE TIPO DE bapi_ TIPO bapi_epm_product_header, COMO LÍNEA DE et_entityset.
---	--

`lr_filter = contexto_solicitud_io_tech->obtener_filtro().
It_filter_select_options = lr_filter->obtener_selección_de_filtro_`
`opciones().`

BUCLE EN `It_filter_select_options` EN `ls_filter_select_`
 Opciones.

SI `ls_filter_select_options-property EQ 'PRODUCTID'.`

BUCLE EN `ls_filter_select_options-select_options` EN `ls_`
 seleccionar_opcion.

`signo-ls_selparamproductid = signo-ls_select_option.
ls_selparamproductid-option = ls_select_option-opción.`

El parámetro `ls_selparamproductid-low` es el parámetro `ls_select_option-low`.
`ls_selparamproductid-high = ls_select_option-high.`

ANEXAR `ls_selparamproductid` A `It_selparamproductid`.

FIN DEL BUCLE.

FIN.SI.

FIN DEL BUCLE.

FUNCTION DE LLAMADA 'BAPI_EPM_PRODUCT_GET_LIST'

* EXPORTACIÓN

<code>*</code> <code>MAXIMO_FILAS</code> <code>Tablas</code> <code> datos de</code> <code> encabezado selparamproductid</code> <code> NOMBRESDEPROVEEDORESDESELPARAMS</code> <code> CATEGORIAS DE SELPARAMA</code> <code> DEVOLVER</code>	<code>=</code> <code>= It_datos de encabezado</code> <code>= It_selparamproductid</code> <code>=</code> <code>=</code> <code>=</code>
---	--

BUCLE EN `It_headerdata` EN `ls_headerdata`.

`ls_product-id_producto = ls_headerdata-id_producto.
ls_categoria_producto = ls_categoria-headerdata.
ls_product-name AÑADIR = nombre-ls_headerdata.
ls_product A et_entityset.`

FIN DEL BUCLE.

MÉTODO FINAL.

Listado 6.3 Conjunto de productos: método Get_Entityset con filtro ProductID

Los cambios introducidos en este ejemplo se llevan a cabo al principio de la codificación. A través de `io_tech_request_context->get_filter()`, primero se recupera una referencia al objeto de filtro central, que proporciona una tabla de una manera compatible con las tablas de rangos a través de `get_filter_select_options()`, de modo que se puede asignar fácilmente a una tabla de rangos correspondiente. Esto solo funciona si el parámetro `$filter` de la URI solicitada se puede asignar a ella. Para ello, recorra la tabla de opciones de selección, verifique el nombre de la propiedad interna y copie las entradas en una tabla de rangos local que se define en función de la interfaz del módulo de función. Luego, también debe descomentar el parámetro `selparamproductid` y entregar la tabla de rangos local al llamar a la BAPI.

Ahora debe activar la clase de extensión DPC para utilizar los nuevos criterios de filtrado. Por ejemplo, puede ejecutar la siguiente URI para filtrar productos con el ID de producto `>= 'HT-1000'` y el ID de producto `<= 'HT-1020'`:

```
/sap/opu/odata/sap/ZEXAMPLE1_SRV/ProductSet?$filter=ID de producto  
'HT-1000' y el ID del producto es 'HT-1020'
```

Además de esto, puedes enriquecer la codificación para considerar también un filtro en `Filtrar por categorías Categorías`. Además de la declaración de datos y la asignación para el módulo de función, agregue el código que se muestra en el Listado 6.4 al bucle.

```
ELSEIF ls_filter_select_options-property EQ 'CATEGORÍA'.  
    BUCLE EN ls_filter_select_options-select_options EN ls_  
seleccionar_opcion.  
        ls_selparamcategories-sign = ls_select_option-sign.  
        ls_selparamcategories-option = opción ls_select_option.  
        ls_selparamcategories-low = ls_select_option-low.  
        ls_selparamcategories-high = ls_select_option-high.  
    AÑADIR ls_selparamcategories A lt_selparamcategories.  
FIN DEL BUCLE.
```

Listado 6.4 Conjunto de productos: método Get_Entityset Filtro adicional en CATEGORÍA

Además del filtro anterior, la URI para filtrar por Categoría = 'Cuadernos' tiene el siguiente aspecto:

```
/sap/opu/odata/sap/ZEXAMPLE1_SRV/ProductSet?$filter=ID de producto  
'HT-1000' y ProductID 'HT-1020' y Categoría eq 'Notebooks'
```

Como puedes ver, los criterios de filtro pueden volverse bastante complejos.

6 Desarrollo de servicios

Tablas de rangos El marco de trabajo asigna los criterios de filtro a tablas de rangos.

Esto le permite procesarlos fácilmente y asignarlos a la tabla de entrada de rangos correspondiente de su módulo de función en su `Get_EntitySet` método. El marco OData los proporciona en una tabla genérica en la que cada entrada representa una propiedad que se utiliza en la opción de consulta de filtro.

Téngase en cuenta que la preparación de las tablas de rangos tiene ciertas limitaciones.

De manera similar a la definición de un informe ABAP con campos de opciones de selección , no se pueden cubrir todos los casos de uso. Por ejemplo, un filtro como `CampoA eq 'A' o El campo B eq 'B'` no se puede colocar en tablas de rangos debido al operador or . Funcionaría si utilizaras " y " en lugar de "or", pero esa es, por supuesto, una declaración diferente con un resultado diferente.

Para estos casos, el objeto de filtro proporciona la función `get_filter_string()`. método que devuelve la cadena de filtro real proporcionada a través de `$filter` Opción de consulta (pero no en un estilo de rangos-tablas).

Otra opción es utilizar el método `get_osql_where_clause` o `get_Método osql_where_clause_convert` del contexto de solicitud `io_tech` parámetro de importación, que le proporciona una cadena que puede usarse en una declaración `SELECT` de Open SQL dinámica .

\$top, \$skip y \$inlinecount

Paginación del lado del cliente Las opciones de consulta `$top` y `$skip` se denominan paginación del lado del cliente, ya que el cliente (consumidor) define cuántas entradas recibirá (`$top`) y cuántas entradas omitirá (`$skip`) el servidor SAP Gateway. Esto permite que el consumidor implemente una funcionalidad de paginación.

Supongamos que el consumidor puede mostrar tres productos por página.

La siguiente URI obtiene la primera página (productos 1 – 3):

`/sap/opu/odata/sap/ZEXAMPLE1_SRV/Conjunto de productos?$top=3&$skip=0`

La siguiente URI obtiene la segunda página (productos 4 a 6):

`/sap/opu/odata/sap/ZEXAMPLE1_SRV/Conjunto de productos?$top=3&$skip=3`

Como puede ver, el valor \$top (tamaño de página) permanece estable si el tamaño de la página no cambia, mientras que el valor \$skip proporciona la cantidad de entradas. lo que debe omitirse, no el número de páginas.

Esto le permite implementar accesos a datos bastante eficientes a la base de datos subyacente porque solo necesita leer los registros que son relevantes para la página solicitada. Cuando obtiene sus datos a través de cualquier módulo de función estándar (como en este ejemplo), no puede beneficiarse realmente de ello porque la mayoría de los módulos de función no admiten parámetros de entrada para top y skip. Lo que normalmente encontrará es un MAX_ROWS parámetro de entrada que al menos limita la cantidad de registros devueltos. Pero solo puede utilizarlo si no se debe aplicar ninguna clasificación a la lista devuelta.

Considere un ejemplo con ordenación predeterminada por el módulo de función y tres productos por página. Para leer las entradas de productos de la página 5, por ejemplo, puede calcular el parámetro MAX_ROWS mediante la siguiente fórmula:

$$\text{MAX_ROWS} = \langle\text{tamaño_de_página}\rangle \times \langle\text{página}\rangle = 3 \times 5 = 15$$

Por lo tanto, es necesario leer los primeros 15 registros de la base de datos para proporcionar las entradas de la página 5 (productos 13 a 15). El módulo RFC lee las entradas de todas las demás páginas (productos 1 a 12), pero deben ignorarse en el procesamiento de Get_EntitySet . Esto empeora si el cliente sigue llamando al servidor SAP Gateway para la página 6, la página 7, la página 8, etc., porque es necesario obtener cada vez más entradas que finalmente se descartan.

Como puede ver, la paginación del lado del cliente puede causar una carga significativa en la carga del servidor. Sin embargo, en última instancia, todo depende de cómo se haya implementado el acceso a las tablas de la base de datos subyacente. Si accede a sus propias tablas Z, puede mejorar significativamente sus instrucciones SELECT para reducir la carga del servidor.

Los valores de \$top y \$skip se pueden recuperar mediante los métodos get_top() y get_skip() de io_tech_request_context. El cálculo y el filtrado de la tabla de resultados se deben realizar mediante su propia codificación. Antes de llamar al módulo de función BAPI_EPM_PRODUCT_GET_LIST , puede calcular el parámetro max-rows como se muestra en el Listado 6.5.

6 Desarrollo de servicios

```

lv_maxrows-bapimaxrow = 0.
lv_top = contexto_solicitud_io_tech->obtener_top( ).
lv_skip = contexto_solicitud_io_tech->get_skip( ).
SI (lv_top NO ES INICIAL).
    lv_maxrows-bapimaxrow = lv_top + lv_skip.
FIN.SI.

```

Listado 6.5 Calcular parámetros de fila máxima

Lv_maxrows se declara como tipo bapi_epm_max_rows. Inmediatamente después del módulo de función, puede aplicar las opciones \$top y \$skip . Para ello, debe calcular el inicio y el final de las entradas de la tabla que desea copiar en la tabla de exportación get_entityset , como se muestra en el Listado 6.6.

```

lv_inicio = 1.
SI lv_skip NO ES INICIAL.
    lv_inicio = lv_salto + 1.
FIN.SI.

SI lv_top NO ES INICIAL.
    lv_fin = lv_arriba + lv_inicio - 1.
DEMÁS.
    lv_end = líneas(lt_headerdata).
FIN.SI.

BUCLE EN lt_headerdata EN ls_headerdata
    DESDE lv_start HASTA lv_end.
        ls_product-id_producto = ls_headerdata-id_producto.
        ls_categoria-producto = ls_categoria-headerdata.
        ls_product-name          = nombre-ls_headerdata.
        AÑADIR ls_product A et_entityset.
    FIN DEL BUCLE.

```

Listado 6.6 Conjunto de productos: método Get_Entityset con paginación externa

Máximo de filas parámetro	Como se mencionó anteriormente, y como se puede ver en la codificación, el único beneficio proviene de proporcionar el parámetro max-rows , lo que podría reducir la cantidad de registros obtenidos de la base de datos (no lo sabemos a menos que investiguemos cómo el módulo de función determina sus datos). Aparte de eso, \$top y \$skip solo garantizan que un subconjunto de los datos determinados se devuelva al consumidor. La opción de consulta \$inlinecount normalmente entra en juego en combinación con \$top y \$skip. De manera similar a \$count, se supone que cuenta los
------------------------------	---

Número de entradas. La diferencia es que el valor del recuento en línea es se proporciona junto con la fuente OData, por eso se llama "en línea". La opción de consulta \$inlinecount debe incluir el valor all-pages para que funcione. Si proporciona \$inlinecount=none, se maneja como si no se hubiera proporcionado ningún recuento en línea.

El recuento en línea siempre cuenta la cantidad total de entradas, incluso si la colección resultante contiene menos debido a \$top y/o \$skip. Por lo tanto, en este caso, destruye la opción de usar el parámetro de entrada max-rows del módulo de función porque necesita recibir todas las entradas para poder contarlas. Puede usar el método has_inlinecount() de io_

Número entero de entradas

objeto de entrada tech_request_context para averiguar si \$inlinecount

Se proporcionó la opción de consulta. La codificación ajustada justo antes de la llamada al módulo de función se ve así:

```
lv_maxrows-bapimaxrow = 0.  
lv_top = contexto_solicitud_io_tech->obtener_top( ).  
lv_skip = contexto_solicitud_io_tech->get_skip( ).  
lv_has_inlinecount = contexto_de_solicitud_de_io_tech->has_  
recuento en línea().  
SI (lv_top NO ES INICIAL) Y  
    (lv_has_inlinecount EQ abap_false).  
    lv_maxrows-bapimaxrow = lv_top + lv_skip.  
FIN.SI.
```

Y justo después de la llamada al módulo de función, puede proporcionar el recuento en línea configurando el valor inlinecount de ES_RESPONSE_CONTEXT

Estructura de exportación:

```
SI lv_has_inlinecount EQ abap_true.  
    es_response_context-inlinecount = líneas(lt_headerdata).  
FIN.SI.
```

A modo de ejemplo, la siguiente URI solo proporciona las primeras 3 entradas (primera página) pero cuenta la colección completa (en este caso, 115 entidades):

Proporciona tres entradas, cuenta toda la colección

```
/sap/opu/odata/sap/ZEXAMPLE1_SRV/Conjunto de productos?$top=3&$skip=0&  
$inlinecount=todas las páginas
```

El recuento en línea está contenido en la respuesta OData a través de la etiqueta m:count (ver Figura 6.61).

6 Desarrollo de servicios

```

<?xml version="1.0"?>
- <feed xmlns:d="http://schemas.microsoft.com/ado/2007/08/dataservices" xmlns:m="http://schemas.microsoft.com/ado/2007/08/dataservices/metadata" xml:base="http://SAPGateway:50000/sap/opu/odata/sap/ZEXAMPLE1_SRV/">
  <id>http://SAPGateway:50000/sap/opu/odata/sap/ZEXAMPLE1_SRV/ProductSet</id>
  <title type="text">ProductSet</title>
  <updated>2015-12-04T14:00:50Z</updated>
  - <author>
    <name/>
  </author>
  <link title="ProductSet" rel="self" href="ProductSet"/>
  <m:count>115 </m:count>
  - <entry>
    <id>http://SAPGateway:50000/sap/opu/odata/sap/ZEXAMPLE1_SRV/ProductSet('HT-1000')</id>
    <title type="text">ProductSet('HT-1000')</title>
    <updated>2015-12-04T14:00:50Z</updated>
    <category scheme="http://schemas.microsoft.com/ado/2007/08/dataservices/scheme" term="ZEXAMPLE1_SRV">
      <link title="Product" rel="self" href="ProductSet('HT-1000')"/>
    </category>
    - <content type="application/xml">
      - <m:properties xmlns:d="http://schemas.microsoft.com/ado/2007/08/dataservices" xmlns:m="http://schemas.microsoft.com/ado/2007/08/dataservices/metadata" xmlns:dct="http://schemas.microsoft.com/ado/2007/08/dataservices/atom">
        <d:ProductID>HT-1000 </d:ProductID>
        <d:Category>CATEGORÍA </d:Category>
        <d:Name>NOMBRE </d:Name>
      </m:properties>
    </content>
  </entry>

```

Figura 6.61 Resultado de la consulta con \$inlinecount

\$ ordenar por

La opción de consulta \$orderby le permite definir la clasificación de su conjunto de resultados. Los parámetros de ordenación los proporciona el marco en su método Get_EntitySet a través del método get_orderby() de io_tech_request_context. Proporciona una tabla que contiene una entrada para cada propiedad proporcionada en la cláusula \$orderby con la propiedad interna nombre.

Si desea que la colección se ordene por categoría (primera) y nombre (segundo), la URI se vería así:

/sap/opu/odata/sap/ZEXAMPLE1_SRV/ProductSet?\$orderby=Categoría, Nombre

En este caso, la tabla recibida por get_orderby() contiene las dos entradas que se muestran en la Tabla 6.3. La columna Property_Path entra en juego si, por ejemplo, desea ordenar por una propiedad de un tipo complejo.

Propiedad	Ruta de la propiedad	Orden
CATEGORÍA	CATEGORÍA	ascendente
NOMBRE	NOMBRE	ascendente

Tabla 6.3 Contenido de la tabla Orderby

En consecuencia, obtendrá order = desc si proporciona la propiedad de consulta relacionada \$orderby con la opción descendente desc. No es necesario proporcionar la opción ascendente asc en la URI porque es el orden de clasificación predeterminado.

En combinación con \$top y \$skip, siempre debe realizar primero la ordenación y después la paginación. De lo contrario, el resultado podría ser incorrecto.

Si combina esas opciones de consulta y tampoco ejecuta su propia paginación del lado del servidor Declaración SELECT : es aún más difícil realizar un acceso eficiente a la base de datos porque los módulos RFC generalmente no ofrecen ninguna capacidad de clasificación. En otras palabras, para cada página, siempre hay que leer todos los registros, ordenarlos y luego elegir la página solicitada del resultado. Estas solicitudes deben manejarse con cuidado porque pueden afectar significativamente el rendimiento del servidor si hay muchas llamadas simultáneas. En este caso, es más eficiente optar por la paginación del lado del servidor, que normalmente requiere que el cliente almacene en caché las entradas que ya se leyeron.

6.5.4 Propiedades de navegación

Las propiedades de navegación le permiten navegar de una entrada o colección a otra entrada o colección. Un ejemplo típico es la navegación desde el encabezado de un pedido de venta (entrada) a la lista de artículos de línea del pedido de venta (colección). Las propiedades de navegación siempre se basan en asociaciones (relaciones) que también definen la cardinalidad. Además, puede definir opcionalmente una restricción referencial para especificar la relación de clave externa (si es posible).

Puede ver varios ejemplos de propiedades de navegación con asociaciones y restricciones referenciales en el proyecto ZExample2 en el que se han importado los metadatos del servicio de muestra /WBEP/GWSAMPLE_BASIC. Para agregar una propiedad de navegación al proyecto ZExample1, primero debe ingresar el nombre de la propiedad "SupplierID" del tipo de núcleo Edm "Edm.String" con la longitud máxima "10" para el tipo de entidad del producto (consulte la Figura 6.62). Marque los atributos Creable y Actualizable.

Ejemplos de propiedades de navegación

Luego debes regenerar el proyecto y agregar una línea de mapeo en los métodos Get_EntitySet y Get_Entity .

6 Desarrollo de servicios

Properties												
Name	Key	Edm Core Type	Prec.	Scale	Max...	U	Cre...	Upd...	Sort...	Null...	Filt...	Label
ProductID	<input checked="" type="checkbox"/>	Edm.String	0	0	10	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	Product Identifier
Category	<input type="checkbox"/>	Edm.String	0	0	40	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	Product Category
Name	<input type="checkbox"/>	Edm.String	0	0	255	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	Product Name
SupplierID	<input type="checkbox"/>	Edm.String	0	0	10	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	Supplier ID

Figura 6.62 Tipo de entidad de producto con nueva propiedad SupplierID

El ID del proveedor se proporciona a través de la tabla/estructura de salida HEADERDATA.

Para el método Get_EntitySet (PRODUCTSET_GET_ENTITYSET), se debe agregar la

línea requerida al bucle:

```
ls_product-supplierid = ls_headerdata-supplier_id.
```

Para el método Get_Entity (PRODUCTSET_GET_ENTITY), la línea requerida se parece a la siguiente:

```
er_entity-supplierid = ls_headerdata-supplier_id.
```

Después de activar la clase de extensión DPC, puede verificar si el SupplierID está correctamente expuesto en el servicio OData (consulte la Figura 6.63).

The screenshot shows the SAP Gateway Client interface. In the top navigation bar, 'Execute' is selected. Below it, the 'HTTP Method' dropdown is set to 'GET'. The 'Request URI' field contains '/sap/opu/odata/sap/ZEXAMPLE1_SRV/ProductSet'. The 'Protocol' dropdown shows 'HTTP' is selected. The main area displays an XML response:

```
<?xml version="1.0"?>
<feed xmlns:d="http://schemas.microsoft.com/ado/2007/08/dataservices" xmlns:m="http://schemas.microsoft.com/ado/2007/08/dataservices/metadata" xml:base="http://SAPGateway:50000/sap/opu/odata/sap/ZEXAMPLE1_SRV/">
  <id>http://SAPGateway:50000/sap/opu/odata/sap/ZEXAMPLE1_SRV/ProductSet</id>
  <title type="text">ProductSet</title>
  <updated>2015-12-04T15:18:14Z</updated>
  - <author>
    <name/>
  </author>
  <link title="ProductSet" rel="self" href="ProductSet"/>
  - <entry>
    <id>http://SAPGateway:50000/sap/opu/odata/sap/ZEXAMPLE1_SRV/ProductSet('HT-1000')</id>
    <title type="text">ProductSet('HT-1000')</title>
    <updated>2015-12-04T15:18:14Z</updated>
    <category scheme="http://schemas.microsoft.com/ado/2007/08/dataservices/scheme" term="ZEXAMPLE1_SRV.Product"/>
    <link title="Product" rel="self" href="ProductSet('HT-1000')"/>
    - <content type="application/xml">
      - <m:properties xmlns:d="http://schemas.microsoft.com/ado/2007/08/dataservices" xmlns:m="http://schemas.microsoft.com/ado/2007/08/dataservices/metadata">
        <d:ProductID>HT-1000</d:ProductID>
        <d:Category>Notebooks</d:Category>
        <d:Name>Notebook Basic 15</d:Name>
        <d:SupplierID>0100000000</d:SupplierID>
      </m:properties>
    </content>
  </entry>
```

Figura 6.63 Conjunto de productos con ID de proveedor

Para navegar de una entidad a otra, se necesita un nuevo tipo de entidad en el modelo. Para ello, se crea un nuevo tipo de entidad llamado "Proveedor" en el proyecto ZExample1 (ver Figura 6.64).

Tipo de entidad proveedora

Properties													
	Name	Key	Edm Core Type	Prec.	Scale	Max Lngth	U	Crea...	Upd...	Sort...	Null...	Filt.	Label
	SupplierID	<input checked="" type="checkbox"/>	Edm.String	0	0	10						<input type="checkbox"/>	Supplier ID
	SupplierName	<input type="checkbox"/>	Edm.String	0	0	80						<input type="checkbox"/>	Supplier Name

Figura 6.64 Nuevo tipo de entidad: Proveedor

Para simplificar, simplemente defina dos propiedades SupplierID

(Edm.String, Longitud máxima=10) y NombreDeProveedor (Edm.String, Longitud máxima=10)

length=80) para el tipo de entidad del proveedor. Los atributos se pueden dejar sin marcar, ya que solo admitiremos acceso de lectura sin filtrado ni clasificación.

Además, crea un nuevo conjunto de entidades llamado SupplierSet (ver Figura 6.65), donde solo se marcará el atributo Addressable, ya que no implementaremos ninguna otra capacidad.

Entity Sets													
	Name	Entity Type	Name	L...	L...	S...	Crea...	Upd...	Del...	Pag...	Add...	Sub...	Req...
	ProductSet	Product		T			<input checked="" type="checkbox"/>	<input type="checkbox"/>					
	SupplierSet	Supplier		T			<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	

Figura 6.65 Nuevo conjunto de entidades: SupplierSet

El siguiente paso es regenerar el proyecto y redefinir el nuevo método SUPPLIERSET_GET_ENTITYSET de la clase de extensión DPC con la codificación del Listado 6.7.

Generar y
redefinir

MÉTODO proveedoreset_get_entityset.

```
DATOS: ls_bpheaderdata TIPO bapi_epm_bp_header,
       lt_bpheaderdata TIPO TABLA ESTÁNDAR DE bapi_epm_bp_
encabezamiento,
       proveedor_ls           COMO LÍNEA DE et_entityset.
```

FUNCIÓN DE LLAMADA 'BAPI_EPM_BP_GET_LIST'
Tablas

El encabezado de encabezado es lt_bpheaderdata.

6 Desarrollo de servicios

```
BUCLE EN lt_bpheaderdata EN ls_bpheaderdata.
ls_supplier-supplierid = ls_bpheaderdata-bp_id.
ls_supplier-nombredelproveedor = ls_bpheaderdata-nombredelaempresa.
ANEXAR ls_supplier A et_entityset.
FIN DEL BUCLE.
```

MÉTODO FINAL.

[Listado 6.7 Conjunto de proveedores: método Get_Entityset](#)

Es necesario implementar SUPPLIERSET_GET_ENTITY con la codificación del Listado 6.8 para obtener una entrada basada en la clave principal proporcionada.

MÉTODO proveedoreset_get_entity.

```
DATOS: ls_entity COMO er_entity,
ls_bp_id TIPO bapi_epm_bp_id,
ls_headerdata TIPO bapi_epm_bp_header.

io_tech_request_context->obtener_claves_convertidas(
IMPORTADOR
es_key_values = ls_entity ).
```

ls_bp_id-bp_id = ls_entity-id-proveedor.

FUNCIÓN DE LLAMADA 'BAPI_EPM_BP_GET_DETAIL'

EXPORTADOR

bp_id = id_bp_ls
IMPORTANDO

encabezadodata = ls_headerdata.

er_entity-supplierid = ls_headerdata-bp_id.

er_entity-suppliername = ls_headerdata-nombre_empresa.

MÉTODO FINAL.

[Listado 6.8 Conjunto de proveedores: método Get_Entity](#)

Ahora debe activar los cambios en la clase de extensión DPC. A través del cliente SAP Gateway, puede ejecutar la siguiente URI para obtener la lista de proveedores (consulte la Figura 6.66):

/sap/opu/odata/sap/ZEXAMPLE1_SRV/Conjunto de proveedores

```

<?xml version="1.0"?>
<feed xmlns:d="http://schemas.microsoft.com/ado/2007/08/dataservices" xmlns:m="http://schemas.microsoft.com/ado/2007/08/dataservices/metadata" xml:base="http://SAPGateway:50000/sap/opu/odata/sap/ZEXAMPLE1_SRV/">
  <id>http://SAPGateway:50000/sap/opu/odata/sap/ZEXAMPLE1_SRV/SupplierSet</id>
  <title type="text">SupplierSet</title>
  <updated>2015-12-04T15:33:23Z</updated>
  - <author>
    <name/>
  </author>
  <link title="SupplierSet" rel="self" href="SupplierSet"/>
  - <entry>
    <id>http://SAPGateway:50000/sap/opu/odata/sap/ZEXAMPLE1_SRV/SupplierSet('0100000000')</id>
    <title type="text">SupplierSet('0100000000')</title>
    <updated>2015-12-04T15:33:23Z</updated>
    <category scheme="http://schemas.microsoft.com/ado/2007/08/dataservices/scheme" term="ZEXAMPLE1_SRV.Supplier"/>
    <link title="Supplier" rel="self" href="SupplierSet('0100000000')"/>
    - <content type="application/xml">
      - <m:properties xmlns:d="http://schemas.microsoft.com/ado/2007/08/dataservices" xmlns:m="http://schemas.microsoft.com/ado/2007/08/dataservices/metadata" xmlns:d="http://schemas.microsoft.com/ado/2007/08/dataservices">
        <d:SupplierID>0100000000</d:SupplierID>
        <d:SupplierName>SAP</d:SupplierName>
      </m:properties>
    </content>
  </entry>
  - <entry>
    <id>http://SAPGateway:50000/sap/opu/odata/sap/ZEXAMPLE1_SRV/SupplierSet('0100000001')</id>
    <title type="text">SupplierSet('0100000001')</title>
    <updated>2015-12-04T15:33:23Z</updated>
    <category scheme="http://schemas.microsoft.com/ado/2007/08/dataservices/scheme" term="ZEXAMPLE1_SRV.Supplier"/>
    <link title="Supplier" rel="self" href="SupplierSet('0100000001')"/>
    - <content type="application/xml">
      - <m:properties xmlns:d="http://schemas.microsoft.com/ado/2007/08/dataservices" xmlns:m="http://schemas.microsoft.com/ado/2007/08/dataservices/metadata" xmlns:d="http://schemas.microsoft.com/ado/2007/08/dataservices">
        <d:SupplierID>0100000001</d:SupplierID>
        <d:SupplierName>Becker Berlin</d:SupplierName>
      </m:properties>
    </content>
  </entry>
  - <entry>
    <id>http://SAPGateway:50000/sap/opu/odata/sap/ZEXAMPLE1_SRV/SupplierSet('0100000002')</id>
  
```

Figura 6.66 Conjunto de proveedores del servicio ZEXAMPLE1_SRV

Verifique también que la implementación de Get_Entity esté funcionando ejecutando cualquiera de los URI proporcionados en la colección.

Definir propiedad de navegación

Hasta ahora, ha agregado un nuevo tipo de entidad y un nuevo conjunto de entidades con una implementación para los métodos Get_Entity y Get_EntitySet. Ahora definamos la propiedad de navegación. Para ello, primero debe crear una asociación haciendo doble clic en Asociaciones del modelo de datos en el árbol del proyecto y haciendo clic en el botón Crear en la cuadrícula. Cada asociación debe tener un Nombre, un Tipo de entidad principal, una Cardinalidad de la entidad principal, un Tipo de entidad dependiente y una Cardinalidad de la entidad dependiente (consulte la Figura 6.67).

6 Desarrollo de servicios

Associations					
	Name	E...	Principal Entity	Principal Entity Cardinality	Dependent Entity
	Assoc_Product_Supplier		Product	M	Supplier
					1

Figura 6.67 Definición de la asociación Assoc_Product_Supplier

Como cada proveedor puede entregar 0...n productos, configure el Dependiente Cardinalidad de entidad (Proveedor) a 1 y Cardinalidad de entidad principal (Producto) a M. Asegúrese de utilizar la ayuda de valor (F4) para ver los posibles valores de cardinalidad.

Ahora puedes definir la propiedad de navegación navegando a Datos Propiedades de navegación del producto de los tipos de entidad del modelo y hacer clic en el botón Crear para definir una propiedad de navegación para el tipo de entidad del producto. La propiedad de navegación requiere un Nombre y una Relación Nombre (ver Figura 6.68). El nombre de la relación es la asociación que definió anteriormente.

Navigation Properties			
	Name	Relationship Name	Label
	ToSupplier	Assoc_Product_Supplier	L...

Figura 6.68 Definición de propiedad de navegación

Para la propiedad de navegación Nombre, especifique "ToSupplier" para utilizar esta propiedad de navegación para navegar desde el producto hasta el proveedor relacionado.

Comprobar y regenerado Ahora puedes comprobar y regenerar el proyecto. Después de la regeneración, verá que los metadatos han aumentado. Esto no se debe únicamente al nuevo tipo de entidad de proveedor añadido, sino también a la asociación y a la propiedad de navegación que ha definido.

La propiedad de navegación también es visible en el conjunto de productos (consulte la Figura 6.69), así como en cada entrada de producto. Cada recurso de entrada ahora contiene un nuevo vínculo que termina con /ToSupplier, que le permite navegar hasta la entrada de proveedor relacionada.

```

<entry>
  <id>http://SAPGateway:50000/sap/opu/odata/sap/ZEXAMPLE1_SRV/ProductSet('HT-1000')</id>
  <title type="text">ProductSet('HT-1000')</title>
  <updated>2015-12-04T15:39:12Z</updated>
  <category scheme="http://schemas.microsoft.com/ado/2007/08/dataservices/scheme" term="ZEXAMPLE1_SRV.Product"/>
  <link title="Product" rel="edit" href="ProductSet('HT-1000')/"/>
  <link title="ToSupplier" type="application/atom+xml;type=entry" rel="http://schemas.microsoft.com/ado/2007/08/dataservices/related/ToSupplier" href="ProductSet('HT-1000')/ToSupplier"/>
  <content type="application/xml">
    - <m:properties xmlns:m="http://schemas.microsoft.com/ado/2007/08/dataservices" xmlns:d="http://schemas.microsoft.com/ado/2007/08/dataservices/metadata">
      <d:ProductID>HT-1000</d:ProductID>
      <d:Category>Notebooks</d:Category>
      <d:Name>Notebook Basic 15</d:Name>
      <d:SupplierID>0100000000</d:SupplierID>
    </m:properties>
  </content>
</entry>

```

Figura 6.69 Propiedad de navegación en el conjunto de productos

Esta propiedad de navegación en particular lleva a una entrada de proveedor (no a la colección) debido a la definición de la cardinalidad. En el ejemplo de EPM, un producto siempre tiene un proveedor único, mientras que un proveedor puede entregar muchos productos.

Si ejecuta este enlace de navegación para cualquier producto, se encontrará con un Error en la ejecución Error. Esto se debe a que la URI con el enlace de navegación solo contiene la clave principal del producto, pero el método Get_Entity del proveedor (que se invoca al ejecutar el enlace de navegación) requiere la clave principal del proveedor. Debe mejorar un poco la codificación anterior para solucionar este problema.

El cambio que necesitas implementar ahora es derivar el SupplierID del ProductID en el método Get_Entity del proveedor SUPPLIERSET_GET_ENTITY, como se muestra en [el Listado 6.9](#).

MÉTODO proveedoreset_get_entity.

DATOS: ls_entity	ME GUSTA er_entity,
ls_bp_id	TIPO bapi_epm_bp_id,
ls_headerdata	TIPO bapi_epm_bp_header.

DATOS: lv_source_entity_set_name TIPO /wbep/mgw_tech_name,	
ls_product	TIPO zcl_zexample1_mpc=>ts_producto,
ls_id_producto	TIPO bapi_epm_product_id,
ls_prd_headerdata	TIPO bapi_epm_product_header.

lv_nombre_conjunto_entidad_origen = contexto_solicitud_io_tech->obtener_nombre_conjunto_entidad_origen().

6 Desarrollo de servicios

```

SI lv_source_entity_set_name EQ 'Conjunto de productos'.
    io_tech_request_context->obtener_claves_de_origen_convertidas(
        IMPORTADOR
        es_key_values = ls_product ).

ls_product_id-product_id = ls_product-productid.

FUNCIÓN DE LLAMADA 'BAPI_EPM_PRODUCT_GET_DETAIL'
    EXPORTADOR
        id del producto = ls_id_producto
    IMPORTADOR
        datos de encabezado = datos del encabezado ls_prd_

ls_bp_id-bp_id = ls_prd_headerdata-id_del_proveedor.
DEMÁS.
    io_tech_request_context->obtener_claves_convertidas(
        IMPORTADOR
        es_key_values = ls_entity ).

ls_bp_id-bp_id = ls_entity-id-proveedor.
FIN.SI.

FUNCIÓN DE LLAMADA 'BAPI_EPM_BP_GET_DETAIL'
    EXPORTADOR
        bp_id = id_bp_ls
    IMPORTANDO
        encabezadodata = ls_headerdata.

er_entity-supplierid = ls_headerdata-bp_id.
er_entity-suppliername = ls_headerdata-nombre_empresa.

MÉTODO FINAL.

```

[Listado 6.9 Conjunto de proveedores: método Get_Entity con determinación de proveedores a través de la BAPI del producto](#)

Como puede ver, el detalle del producto se lee llamando a la misma BAPI (BAPI_EPM_PRODUCT_GET_DETAIL) que en PRODUCTSET_GET_ENTITY

Método. Esto proporciona los detalles del producto que corresponden a la clave del producto que se entrega a través del enlace de navegación. La clave del producto se determina mediante el método getConvertedSourceKeys de io_tech_request_context en contraste con el uso de getConvertedKeys cuando se accede a los detalles del proveedor directamente sin un paso de navegación.

La declaración IF garantiza que este fragmento de código solo se ejecute en el caso de la propiedad de navegación. En el caso habitual de Get_Entity del proveedor , no es necesario ejecutar esta codificación.

Active los cambios en la clase de extensión DPC y ejecute lo siguiente: Activar cambios

Ejemplo de URI a través del cliente SAP Gateway:

/sap/opu/odata/sap/ZEXAMPLE1_SRV/ConjuntoDeProductos('HT-1000')/AlProveedor

Como resultado, verá la entrada del proveedor (con su clave y nombre) que pertenece al producto 'HT-1000' (ver Figura 6.70).

The screenshot shows the SAP Gateway Client interface. In the top navigation bar, the 'Execute' button is highlighted. Below it, the 'HTTP Method' dropdown is set to 'GET'. The 'Request URI' field contains '/sap/opu/odata/sap/ZEXAMPLE1_SRV/Products('HT-1000')/ToSupplier'. The 'Protocol' dropdown is set to 'HTTP'. On the right side of the header, there is a checkbox labeled 'Reuse HTTP Connection (e.g. for Soft State)' which is unchecked.

The main content area displays an XML response. The XML structure is as follows:

```

<?xml version="1.0" encoding="UTF-8"?>
- <entry xmlns:d="http://schemas.microsoft.com/ado/2007/08/dataservices" xmlns:m="http://schemas.microsoft.com/ado/2007/08/dataservices/metadata" xml:base="http://SAPGateway:50000/sap/opu/odata/sap/ZEXAMPLE1_SRV/"> 
  <id>http://SAPGateway:50000/sap/opu/odata/sap/ZEXAMPLE1_SRV/SupplierSet('0100000000')</id>
  <title type="text">SupplierSet('0100000000')</title>
  <updated>2015-12-04T16:00:51Z</updated>
  <category scheme="http://schemas.microsoft.com/ado/2007/08/dataservices/scheme" term="ZEXAMPLE1_SRV.Supplier"/>
  <link title="Supplier" rel="self" href="SupplierSet('0100000000')"/>
  - <content type="application/xml">
    - <m:properties>
      <d:SupplierID>0100000000</d:SupplierID>
      <d:SupplierName>SAP</d:SupplierName>
    </m:properties>
  </content>
</entry>

```

Figura 6.70 Resultado de navegación de ToSupplier del producto HT-1000

Según el estándar OData, encontrará un enlace propio a la entrada del proveedor recuperada (.../SupplierSet('0100000000')) en el resultado de la navegación.

6.5.5 Crear, actualizar y eliminar métodos

Hasta ahora, solo hemos leído datos de la fuente de datos subyacente ejecutando los módulos de función GET_LIST y GET_DETAIL respectivos a través de los métodos Get_EntitySet y Get_Entity . A diferencia del acceso de lectura, los métodos CREATE, UPDATE y DELETE (CUD) son operaciones de acceso de escritura porque, por lo general, modifican los datos subyacentes.

Crear

CREATE se utiliza siempre que se desea crear una nueva entrada en la colección correspondiente. La operación CREATE para el tipo de entidad de producto se gestiona mediante el método PRODUCTSET_CREATE_ENTITY en el DPC.

6 Desarrollo de servicios

Redefinir en el
Extensión DPC
clase

De manera similar a los métodos de lectura, debe redefinir este método en la clase de extensión del proveedor de datos. Como ejemplo, utilice la codificación en la [Lista 6.10.](#)

MÉTODO productset_create_entity.

```
DATOS: ls_headerdata TIPO bapi_epm_product_header,
       ls_producto      ME GUSTA er_entity,
       lt_retorno        TABLA ESTANDAR TIPO bapiret2.
```

```
io_data_provider->read_entry_data(IMPORTANDO es_data = ls_
producto ).
```

```
ls_headerdata-product_id = ls_product-productid.
ls_headerdata-category = ls_product-category.
ls_headerdata-name = ls_product-name.
ls_headerdata-supplier_id = ls_product-supplierid.
ls_headerdata-measure_unit = 'EA'.
ls_headerdata-currency_code = 'EUR'.
ls_headerdata-tax_tarif_code = '1'.
ls_headerdata-type_code = 'PR'.
```

FUNCTION DE LLAMADA 'BAPI_EPM_PRODUCT_CREATE'
EXPORTADOR

*	datos de encabezado	= datos del encabezado ls
	PERSISTIR EN LA BASE DE DATOS	= ABAP_VERDADERO
Tablas		
*	FACTORES DE CONVERSIÓN =	
	devolver	= lt_retorno.

```
SI lt_return NO ES INICIAL.
mo_context->obtener_contenedor_de_mensajes()->agregar_mensajes_de_
bapi(
    it_bapi_messages = lt_return
    iv_determinar_mensaje_principal = /wbep/si_mensaje_
contenedor=>gcs_leading_msg_search_option-first ).
```

GENERA EL TIPO DE EXCEPCIÓN /wbep/cx_mgw_busi_exception
EXPORTADOR

```
id de texto = /wbep/cx_mgw_busi_
excepción=>error_comercial
contenedor_de_mensajes = mo_context->obtener_mensaje_
contenedor( ).
FIN.SI.
```

er_entidad = ls_producto.

MÉTODO FINAL.

[Listado 6.10 Conjunto de productos: método de creación de entidad](#)

Con el método `read_entry_data` de la referencia del objeto de importación `io_data_provider`, puede recuperar los datos que se pasaron a lo largo de la solicitud POST en el cuerpo HTTP. La entrada se recupera en el formato de la definición del tipo de entidad. Por eso, la estructura `ls_product` es del tipo `ER_ENTITY`.

Para llamar correctamente al módulo de función `BAPI_EPM_PRODUCT_CREATE`, algunos valores que no forman parte del modelo de datos deben tener valores predeterminados (por ejemplo, `measure_unit`). Esta es la lógica específica de la función empresarial subyacente.

En este ejemplo, también hemos introducido algunos métodos de manejo de errores. Para el método de manejo de errores `Para simplificar`, no hemos hecho esto en `Get_EntitySet` y `Get_Casos de entidad`, aunque definitivamente también tiene sentido en este caso. El manejo de errores consiste en recuperar la tabla de retorno de la llamada al módulo de función.

Suponiendo que solo los errores serán parte de la tabla de retorno, si la tabla no está vacía, primero se utiliza la referencia del objeto miembro `mo_context` para determinar el contenedor de mensajes (a través de `get_message_container()`) y se utiliza el método `add_messages_from_bapi` para registrar los mensajes BAPI. En segundo lugar, se genera una excepción comercial `/wbep/cx_mgw_busi_exception` que abortará el procesamiento, enviará un código de retorno HTTP 400 al consumidor y proporcionará los mensajes como parte del cuerpo de la respuesta.

Al final del método, se devuelve la entrada recién creada al marco (`ER_ENTITY`). Normalmente, hay que llamar a la BAPI `Get-Detail` correspondiente para obtener los detalles de la entrada recién creada, ya que es posible que se hayan modificado o calculado algunas propiedades. Este no es el caso de este ejemplo, por lo que se puede devolver la entrada recuperada directamente al marco.

Después de haber realizado la implementación del método `CREATE`, vuelva a activar la clase de extensión DPC. Después de eso, utilice el cliente SAP Gateway para verificar si la operación `CREATE` funciona como se espera. El método `CREATE` requiere un cuerpo HTTP adecuado con la entrada que se creará. En lugar de armar manualmente un cuerpo HTTP, puede hacer clic en el botón `Usar como solicitud` en el cliente SAP Gateway. Para esto, primero debe ejecutar una operación GET para recibir una entrada de origen adecuada. Luego, haga clic en el botón `Usar como solicitud` para copiarla en el lado izquierdo de

Activar el DPC
clase de extensión

6 Desarrollo de servicios

El cliente SAP Gateway (ver Figura 6.71). El lado izquierdo es el HTTP
El lado de la solicitud y el lado derecho es el lado de la respuesta HTTP.

The screenshot shows the SAP Gateway Client interface. On the left, under 'HTTP Request', there is an XML representation of an HTTP POST request. This request is directed to the URI /sap/opu/odata/sap/ZEXAMPLE1_SRV/ProductSet('HT-1000'). The XML body contains product information, including a category and supplier details. On the right, under 'HTTP Response - Processing Time = 65 ms', the response is shown. It is also an XML document, indicating a successful operation. The response includes the same product information as the request, confirming the creation of a new entry.

Figura 6.71 La función Usar como solicitud en el cliente SAP Gateway

Antes de poder ejecutar la operación CREATE , también debe cambiar el método HTTP de GET a POST. Además, debe ajustar la URI porque la operación CREATE debe ejecutarse en una colección y no en una sola entrada (asegúrese de quitar los corchetes con la clave principal). Por último, es posible que desee ajustar el cuerpo HTTP, que contiene los datos de la entrada que se creará (por ejemplo, asegúrese de cambiar el ID y el nombre del producto).

Según la lógica empresarial subyacente, es posible que no sea necesario proporcionar la clave principal en el cuerpo de la solicitud HTTP si la calcula el módulo de función que se asignó en la operación CREATE . En este ejemplo, debe proporcionar un ID de producto nuevo/único porque no existe un rango de números que calcule un ID nuevo para usted.

HTTP 201 Si la operación CREATE fue exitosa (ver Figura 6.72), obtendrá un HTTP Respuesta 201. Además, obtienes el registro recién creado porque el servidor backend podría haber determinado la clave principal de la nueva entrada (no en este caso de ejemplo, sin embargo). Por lo tanto, el servidor tiene que proporcionar el registro, o el cliente no podrá acceder a él sin la clave.

Además, el cliente se beneficia de esto si el servidor determina o calcula propiedades distintas a la clave principal, porque esos valores se envían junto con la respuesta y el cliente no necesita ejecutar otra solicitud GET . Este patrón es diferente en el caso de actualización, como verás

ver siguiente.

The screenshot shows the SAP Gateway Client interface. In the 'HTTP Request' section, there is an XML payload for creating a product:

```

<?xml version="1.0" encoding="UTF-8"?>
<entry xmlns="http://SAPGateway:50000/sap/opu/odata/sap/ZEXAMPLE1_SRV/" xml:lang="en-US">
  <id href="http://SAPGateway:50000/sap/opu/odata/sap/ZEXAMPLE1_SRV/ProductSet('HT-1000')"/>
  <title type="text">ProductSet('HT-1000')</title>
  <updated>2015-12-07T10:27:24Z</updated>
  <category term="ZEXAMPLE1_SRV:Product" scheme="http://schemas.microsoft.com/ado/2007/08/dataservices/scheme"><id href="http://SAPGateway:50000/sap/opu/odata/sap/ZEXAMPLE1_SRV/ProductSet('HT-1000')"/><rel="edit" title="Product"/><link href="http://SAPGateway:50000/sap/opu/odata/sap/ZEXAMPLE1_SRV/Products('VD-1000')/ToSupplier" rel="http://schemas.microsoft.com/ado/2007/08/dataservices/related/ToSupplier" type="application/atom+xml;type=entry; href="http://SAPGateway:50000/sap/opu/odata/sap/ZEXAMPLE1_SRV/Products('VD-1000')/ToSupplier"/><content type="application/xml"><m:properties><d:ProductID>VD-1000</d:ProductID><d:Category>Notebooks</d:Category><d:Name>My Notebook Basic 15</d:Name><d:SupplierID>0100000000</d:SupplierID></m:properties></content></entry>
```

In the 'HTTP Response' section, the status is 201 (Created) and the processing time is 162 ms. The response XML includes the newly created product details:

```

<?xml version="1.0" encoding="UTF-8"?>
<entry xmlns:m="http://schemas.microsoft.com/ado/2007/08/dataservices/metadata" xmlns="http://SAPGateway:50000/sap/opu/odata/sap/ZEXAMPLE1_SRV/" xml:base="http://SAPGateway:50000/sap/opu/odata/sap/ZEXAMPLE1_SRV/" xmlns:d="http://schemas.microsoft.com/ado/2007/08/dataservices" xmlns:i="http://www.w3.org/2005/08/semantics/~" xmlns:m="http://schemas.microsoft.com/ado/2007/08/dataservices/metadata" xmlns="http://SAPGateway:50000/sap/opu/odata/sap/ZEXAMPLE1_SRV/">
  <id href="http://SAPGateway:50000/sap/opu/odata/sap/ZEXAMPLE1_SRV/ProductSet('HT-1000')"/>
  <title type="text">ProductSet('HT-1000')</title>
  <updated>2015-12-07T10:27:24Z</updated>
  <category scheme="http://schemas.microsoft.com/ado/2007/08/dataservices/scheme"><id href="http://SAPGateway:50000/sap/opu/odata/sap/ZEXAMPLE1_SRV/Products('VD-1000')/ToSupplier" rel="http://schemas.microsoft.com/ado/2007/08/dataservices/related/ToSupplier" type="application/atom+xml;type=entry; href="http://SAPGateway:50000/sap/opu/odata/sap/ZEXAMPLE1_SRV/Products('VD-1000')/ToSupplier"/><link href="http://SAPGateway:50000/sap/opu/odata/sap/ZEXAMPLE1_SRV/Products('VD-1000')/ToSupplier" rel="edit" title="ToSupplier"/><content type="application/atom+xml;type=entry; href="http://SAPGateway:50000/sap/opu/odata/sap/ZEXAMPLE1_SRV/Products('VD-1000')/ToSupplier"/><m:properties><d:ProductID>VD-1000</d:ProductID><d:Category>Notebooks</d:Category><d:Name>My Notebook Basic 15</d:Name><d:SupplierID>0100000000</d:SupplierID></m:properties></content></entry>
```

Figura 6.72 Operación CREATE exitosa

Actualizar

La operación UPDATE entra en juego siempre que se necesita cambiar un recurso de entrada existente. Por lo tanto, el URI debe abordar la entrada como en un caso de Get-Detail . La operación UPDATE para el tipo de entidad de producto se maneja mediante el método PRODUCTSET_UPDATE_ENTITY en el DPC. De manera similar a los otros métodos que ha implementado, redefina este método en la clase de extensión DPC y aplique la codificación del Listado 6.11.

MÉTODO productset_update_entity.

```

DATOS: ls_product COMO er_entity,
       ls_entity COMO er_entity,
       ls_product_id TIPO bapi_epm_product_id,
       ls_headerdata TIPO bapi_epm_product_header,
       ls_headerdatax TIPO bapi_epm_product_headerx,
       lt_return TIPO TABLA ESTÁNDAR DE bapiret2.
```

6 Desarrollo de servicios

```

io_data_provider->read_entry_data(IMPORTANDO es_data = ls_
producto).

io_tech_request_context->obtener_claves_convertidas(
    IMPORTADOR
    es_key_values = ls_entity).

ls_product_id-product_id = ls_entity-productid.

FUNCIÓN DE LLAMADA 'BAPI_EPM_PRODUCT_GET_DETAIL'
    EXPORTADOR
        id del producto = ls_id_producto
        IMPORTADOR
            datos de encabezado = datos del encabezado ls
        Tablas
        *
            Factores de conversión de =
            retorno = lt_retorno.

SI lt_return NO ES INICIAL.
    mo_context->obtener_contenedor_de_mensajes()->agregar_mensajes_de_
bapi(
    it_bapi_messages = lt_return
    iv_determinar_mensaje_principal = /iwbep/si_mensaje_
contenedor=>gcs_leading_msg_search_option-first).

GENERA EL TIPO DE EXCEPCIÓN /iwbep/cx_mgw_busi_exception
    EXPORTADOR
        id de texto = /iwbep/cx_mgw_busi_
excepción=>error_comercial
        contenedor_de_mensajes = mo_context->obtener_mensaje_
contenedor().
    FIN.SI.

    ls_headerdata-categoría ls_headerdata- = ls_categoría-producto.
    nombre ls_headerdata- = ls_nombre-producto.
    supplier_id = ls_product-supplierid.

    ls_headerdatax-id_producto = ls_headerdata-id_producto.
    ls_headerdatax-categoría = 'X'.
    ls_headerdatax-nombre = 'X'.
    ls_headerdatax-supplier_id = 'X'.

FUNCIÓN DE LLAMADA 'BAPI_EPM_PRODUCT_CHANGE'
    EXPORTADOR
        id del producto = ls_id_producto
        datos del = datos del encabezado ls
        *
            encabezado datos del encabezado = ls_headerdatax
            PERSISTIR EN LA BASE DE DATOS = ABAP_VERDADERO
        Tablas

```

```

*      FACTORES DE CONVERSIÓN =
*      FACTORES_DE_CONVERSIÓN =
*          devolver           = lt_retorno.

SI lt_return NO ES INICIAL.
    mo_context->obtener_contenedor_de_mensajes()->agregar_mensajes_de_
bapi(
    it_bapi_messages = lt_return
    iv_determinar_mensaje_principal = /wbep/si_mensaje_
contenedor=>gcs_leading_msg_search_option-first ).

GENERA EL TIPO DE EXCEPCIÓN /wbep/cx_mgw_busi_exception
EXPORTADOR
    excepción           = /wbep/cx_mgw_busi_
de texto =>error_comercial
    contenedor_de_mensajes = mo_context->obtener_mensaje_
contenedor( ).
FIN.SI.

er_entidad = ls_producto.

```

MÉTODO FINAL.

Listado 6.11 Conjunto de productos: método de actualización de entidad

Esta implementación reutiliza ciertos elementos de los otros métodos.

Primero, use la referencia del objeto de entrada io_data_provider para obtener los datos entrantes del cuerpo HTTP. Luego, seleccione la clave principal convertida del método getConvertedKeys de io_tech_request_context. Con la clave principal, lea la instancia del producto con BAPI_

Función EPM_PRODUCT_GET_DETAIL para llenar la estructura ls_headerdata .

Esta vez, verifica el retorno de la tabla en busca de entradas (asumiendo que solo se informarán errores en esa tabla y no se proporcionará ningún mensaje de éxito) para evitar que se haya utilizado una clave principal no válida en el URI.

Después de leer la instancia del producto, se aplican las propiedades entrantes. También debe configurar los indicadores X en la estructura X correspondiente para informar a la BAPI qué atributos actualizar. A continuación, llame a BAPI_EPM_PRODUCT_CHANGE para realizar la actualización. De manera similar a CREATE Método, se comprueba si la tabla devuelve mensajes existentes y se genera una excepción en ese caso. Por último, también hay un parámetro de exportación ER_ENTITY que se debe completar.

Propiedades
entrantes

6 Desarrollo de servicios

Una vez que haya terminado con la implementación del método UPDATE , debe activar la clase de extensión DPC. Ahora debe usar el cliente SAP Gateway para verificar si la operación UPDATE funciona como se espera. Para ello, primero realice una sola lectura para obtener un cuerpo de respuesta HTTP adecuado que pueda copiar en el lado de la solicitud haciendo clic en el botón Usar como solicitud. El URI permanece sin cambios porque el método UPDATE

La operación siempre se realiza en un recurso de una sola entrada. Solo necesita cambiar el método HTTP de GET a PUT. Si la actualización se realizó correctamente ([consulte la Figura 6.73](#)), solo obtendrá una respuesta HTTP 204 (sin contenido).

The screenshot shows the SAP Gateway Client interface with the following details:

- HTTP Method:** PUT
- Request URI:** /sap/opu/odata/sap/ZEXAMPLE1_SRV/ProductSet('VD-1000')
- Protocol:** HTTPS
- HTTP Request Headers:**

Header Name	Value
Content-Type	application/atom+xml;type=entry; charset=utf-8
X-CSRF-Token	JBPXh_n13gKmkzHGrpnA==
- HTTP Response Headers:**

Header Name	Value
status_code	204
status_reason	No Content
- HTTP Response Body:** <RESPONSE_BODY/>
- XML Request Body (Line Numbers):**

```

1  <?xml version="1.0" encoding="utf-8"?>
2  <entry xml:base="http://SAPGateway:50000/sap/opu/odata/sap/ZEXAMPLE1_SRV/" xmlns="http://www
3  <id href="http://SAPGateway:50000/sap/opu/odata/sap/ZEXAMPLE1_SRV/ProductSet('VD-1000')"/><id>
4  <title type="text">ProductSet('VD-1000')</title>
5  <updated>2015-12-07T12:17:50Z</updated>
6  <category term="ZEXAMPLE1_SRV.Product" schema="http://schemas.microsoft.com/ado/2007/08/d
7  <link href="ProductSet('VD-1000')/edit" rel="edit" title="Product"/>
8  <link href="ProductSet('VD-1000')/ToSupplier" rel="http://schemas.microsoft.com/ado/2007/0
9  <content type="application/xml">
10 <properties>
11   <d:ProductID>VD-1000</d:ProductID>
12   <d:Category>Notebooks</d:Category>
13   <d:Name>My new Notebook Basic 15</d:Name>
14   <d:SupplierID>0100000000</d:SupplierID>
15   <m:properties>
16   </m:properties>
17 </content>
18 </entry>

```

Figura 6.73 Operación de actualización exitosa

El código HTTP indica que el cuerpo de la respuesta está vacío. Esto sucede a propósito porque se supone que el cliente tiene toda la información y, por lo tanto, enviar la entrada junto con la respuesta genera una sobrecarga innecesaria. El cliente siempre puede realizar una única lectura para recuperar la entidad nuevamente si el servidor backend ha calculado valores que el cliente no tiene.

Borrar

La operación DELETE suele ser muy sencilla de implementar. Requiere redefinir el método PRODUCTSET_DELETE_ENTITY de la clase de extensión DPC. Puede utilizar la codificación del Listado 6.12 [como ejemplo](#).

MÉTODO productset_delete_entity.

DATOS: ls_entity TIPO zcl_zexample1_mpc=>ts_product,
 ls_product_id TIPO bapi_epm_product_id,
 lt_return TIPO TABLA ESTÁNDAR DE bapiret2.

io_tech_request_context->obtener_claves_convertidas(
 IMPORTADOR
 es_key_values = ls_entity).

ls_product_id-product_id = ls_entity-productid.

FUNCIÓN DE LLAMADA 'BAPI_EPM_PRODUCT_DELETE'
 EXPORTADOR

* id_producto = ls_id_producto
 PERSISTIR EN LA BD = ABAP VERDADERO
 Tablas
 devolver = lt_retorno.

SI lt_return NO ES INICIAL.

 mo_context->obtener_contenedor_de_mensajes()->agregar_mensajes_de_
 bapi(
 it_bapi_messages = lt_return
 iv_determinar_mensaje_principal = /wbep/si_mensaje_
 contenedor=>gcs_leading_msg_search_option-first).

GENERA EL TIPO DE EXCEPCIÓN /wbep/cx_mgw_busi_exception
 EXPORTADOR

 id de texto = /wbep/cx_mgw_busi_
 excepción=>error_comercial
 contenedor_de_mensajes = mo_context->obtener_mensaje_
 contenedor().
 FIN.SI.

MÉTODO FINAL.

[Listado 6.12 Conjunto de productos: método de eliminación de entidad](#)

En primer lugar, vuelva a manejar la clave principal de la entrada que se va a eliminar. Con esta clave principal información, llame a la función DELETE BAPI_EPM_PRODUCT_DELETE de inmediato. Se verifica nuevamente la tabla de retorno para ver si hay entradas y se genera una excepción si existe alguna entrada, suponiendo que no reciba ningún mensaje de éxito o advertencia en la tabla de retorno.

Después de activar la clase de extensión DPC, puede verificar si la operación DELETE implementada funciona correctamente. Las operaciones DELETE siempre se ejecutan en una sola entrada. Por lo tanto, debe proporcionar una

6 Desarrollo de servicios

URI que direcciona un recurso de entrada única. El método HTTP debe ser BORRAR.

Si la eliminación fue exitosa, obtendrá una respuesta HTTP 204 (sin contenido) (ver Figura 6.74).

Header Name	Value
Content-Type	application/atom+xml;type=entry; charset=utf-8
X-CSRF-Token	pSwnj809ju5_NL9m6ERzeg==

Header Name	Value
~status_code	204
~status_reason	No Content
~server_protocol	HTTP/1.0

Figura 6.74 Operación DELETE exitosa

6.5.6 Importaciones de funciones

Las importaciones o acciones de funciones se utilizan siempre que se necesita ejecutar una función en un objeto comercial que no encaja en las operaciones CRUD-Q predeterminadas. Un ejemplo es la liberación de un pedido de venta que puede influir en una serie de propiedades y, además, también activar alguna funcionalidad de backend (por ejemplo, iniciar un flujo de trabajo). Las importaciones de funciones se definen en el nivel de servicio, por lo que una función no se ejecuta en una colección dedicada o

entrada.

- Definir función
importar
- Se puede definir una importación de función haciendo clic derecho en el Modelo de datos y seleccionando Crear importación de función en el Generador de servicios. Esto abre la ventana Crear importación de función, donde primero debe proporcionar un nombre para la importación de función (consulte la Figura 6.75).



Figura 6.75 Cuadro de diálogo Crear importación de función

Para cada importación de función, puede definir un tipo de retorno de la importación de función. Se admiten los valores que se indican en la Tabla 6.4.

Tipo de devolución	Descripción
Tipo complejo	Datos devueltos como tipo complejo
Tipo de entidad	Datos devueltos como tipo de entidad
Sin retorno	No se devolvieron datos

Tabla 6.4 Tipos de retorno de importación de funciones

En este ejemplo, desea implementar una importación de función que determine los productos más pesados. Para ello, primero debe ingresar la propiedad WeightMeasure y seleccionar Edm.Decimal, precisión 13 y Scale 3 para el tipo de entidad del producto. Además, agregue la propiedad WeightUnit y seleccione Edm.String y MaxLength 3 (consulte la Figura 6.76). Necesita esto para ver el peso y la unidad de peso de cada producto. La importación de función en sí no necesita estas dos propiedades adicionales.

Properties													
Name	Key	Edm Core Type	Prec.	Scale	Max...	Unit	Property Name	Crea...	Upd...	Sort...	Null...	Filt.	Label
ProductID	<input checked="" type="checkbox"/>	Edm.String	0	0	10			<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	Product Identifier
Category	<input type="checkbox"/>	Edm.String	0	0	40			<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	Product Category
Name	<input type="checkbox"/>	Edm.String	0	0	255			<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	Product Name
SupplierID	<input type="checkbox"/>	Edm.String	0	0	10			<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	Supplier ID
WeightMeasure	<input type="checkbox"/>	Edm.Decimal	13	3	0	WeightUnit		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	Weight Measure
WeightUnit	<input type="checkbox"/>	Edm.String	0	0	3			<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	Weight Unit

Figura 6.76 WeightMeasure y WeightUnit del tipo de entidad Producto

Asegúrese de ajustar PRODUCTSET_GET_ENTITYSET, PRODUCTSET_GET_ENTITY, PRODUCTSET_CREATE_ENTITY y PRODUCTSET_UPDATE_ENTITY también consideran ambas propiedades nuevas (agregan asignaciones relacionadas).

El tipo de retorno de la función import es un conjunto de entidades porque se desea recibir la lista de productos más pesados. Se utiliza una lista porque podría no haber ningún producto según el parámetro o podría haber un conjunto de productos que sean todos los más pesados.

Si establece el tipo de devolución (campo Tipo de devolución) en Tipo de entidad, también puede especificar el tipo de devolución. En este caso, se trata del tipo de entidad Producto.

6 Desarrollo de servicios

La cardinalidad de retorno especifica cuántos tipos de retorno definidos pueden aparecer. Se admiten los valores de la [Tabla 6.5.](#)

Cardinalidad	Descripción
0..1	No se produce más de una instancia. También se permiten cero instancias. Tenga en cuenta que no se recomienda utilizar esta cardinalidad de retorno (consulte la nota SAP 2009874 para obtener más detalles).
1	Ocurrencia de exactamente una instancia.
0..n	Ocurrencia de cero o más instancias.
1..n	Ocurrencia de una o más instancias.

Tabla 6.5 Cardinalidades de retorno

En este ejemplo, elegimos 0..n para el campo Cardinalidad de retorno, ya que puede haber una lista (incluida una lista con una sola entrada) o ninguna entrada en el conjunto de resultados.

Después de elegir una cardinalidad en la que se puedan devolver muchas (n) entradas (colección), también es necesario definir el conjunto de entidades de retorno. En este ejemplo, ingresamos "ProductSet" para el conjunto de entidades de retorno.

El método HTTP define básicamente si la función de importación solo lee datos (HTTP GET) o también manipula datos (HTTP POST). Esto es importante porque un GET no puede cambiar ningún dato en el sistema. Un método POST siempre está protegido mediante un token de falsificación de solicitud entre sitios (XSRF), lo que no ocurre con los accesos GET .

Por último, configure la Acción para el Tipo de entidad en Producto para indicar que esta importación de función está relacionada con el tipo de entidad del producto.

Una vez especificados todos los atributos, tendrá la definición de importación de la función como se muestra [en la Figura 6.77.](#)

Function Imports							
Name	Return Type Kind	Return Type	Return Cardinality	Return Entity Set	HTTP	Action for Entity Type	
DetermineHeaviestProduct	Entity Type	Product	0..n	ProductSet	GET	Product	

Figura 6.77 Definición de una función de muestra de importación

En nuestro ejemplo, también debe proporcionar un parámetro de importación para la función de ejemplo import. Este parámetro define la categoría de producto de la que desea determinar el producto más pesado.

Proporcionar parámetro de importación

Para definir los parámetros de importación para las importaciones de funciones, debe expandir el nuevo nodo Importaciones de funciones DetermineHeaviestProduct en el árbol del proyecto y luego hacer doble clic en Parámetros de importación de funciones.

Puede agregar entradas a la cuadrícula utilizando los botones Agregar/Crear.

En este ejemplo, introduzca un parámetro de importación de función denominado Categoría. Tiene los mismos atributos que la propiedad Categoría en el tipo de entidad de producto (consulte la Figura 6.78).

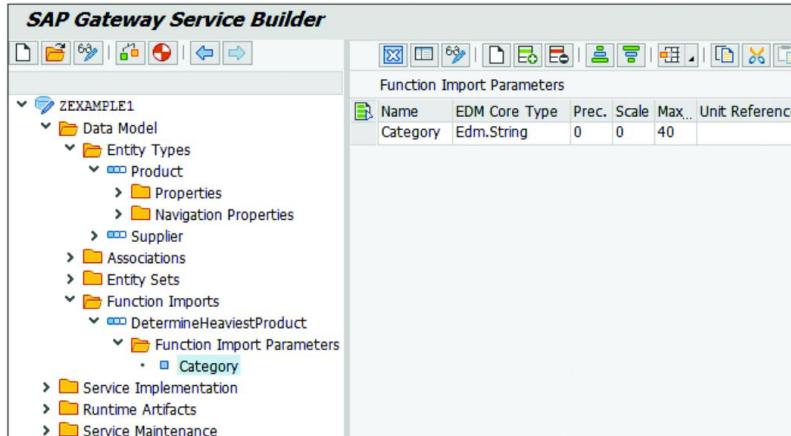


Figura 6.78 Definición del parámetro de importación de función

Eso es todo lo que necesita hacer en el lado de los metadatos. Ahora puede verificar y regenerar el proyecto. El aprovisionamiento del servicio (proveedor de datos) debe realizarse manualmente redefiniendo el método respectivo en la clase de extensión DPC. Para esto, abra ZCL_ZEXAMPLE1_DPC_EXT en el generador de clases ABAP y póngalo en modo de edición. El método que necesita redefinir es EXECUTE_ACTION (interfaz /WBEP/IF_MGW_APPL_SRV_RUNTIME).

El listado 6.13 muestra la codificación que estamos utilizando.

MÉTODO /wbep/if_mgw_appl_srv_runtime~ejecutar_acción.

DATOS: lv_action_name	TIPO /wbep/mgw_nombre_tecnico,
lt_producto	TIPO zcl_zexample1_mpc=>tt_producto,

6 Desarrollo de servicios

```

ls_producto          TIPO zcl_zexample1_mpc=>ts_product,
lt_headerdata        TABLA DE TIPOS DE bapi_epm_product_
encabezamiento,
lt_headerdata_orig  TABLA DE TIPO DE bapi_epm_product_
encabezamiento,
ls_headerdata        TIPO bapi_epm_product_header,
lt_categorías_rango TABLA DE TIPOS DE bapi_epm_product_
de categorías,
ls_categorías       TIPO bapi_epm_product_categ_range,
lv_máximo_peso      TIPO snwd_weight_measure,
lv_salida           TIPO snwd_weight_measure.

CONSTANTES: cv_weight_unit_kg TIPO snwd_weight_
unidad VALOR 'KG'.

lv_action_name = contexto_de_solicitud_io_tech->función_obtener_
importar_nombre().

CASO lv_nombre_acción.
CUANDO 'DeterminarProductoMásPesado'.
*
Coloque el filtro de categorías junto
io_tech_request_context->obtener_parámetros_convertidos(
IMPORTADOR
es_valores_parámetro = ls_producto ).

SI NO ls_product-category ES INICIAL.
ls_categories-low = ls_categoria-producto.
ls_categories-option = 'EQ'.           "Igual
ls_categories-sign = 'Yo'.            "Incluido
AÑadir ls_categories A lt_categories.
FIN.SI.

*
Obtener productos filtrados
FUNCIÓN DE LLAMADA 'BAPI_EPM_PRODUCT_GET_LIST'
Tablas
    datos de encabezado = lt_headerdata_orig
    selparamcategories = lt_categorías.

*
Convierte todos los pesos a KG y determina el peso más alto
valor t
BORRAR lv_max_weight.
BUCLE EN lt_headerdata_orig EN ls_headerdata.
FUNCIÓN DE LLAMADA 'UNIT_CONVERSION_SIMPLE'
EXPORTADOR
    medida             aporte          = ls_headerdata-peso_
    unidad_en          unidad_en      = ls_headerdata-peso_
    unidad             unidad_fuera   = cv_unidad_de_peso_kg

```

IMPORTADOR

producción	= salida lv
EXCEPCIONES	
conversión_no_encontrada	= 1
división_por_cero	= 2
entrada_inválida	= 3
salida_inválida	= 4
	= 5
desbordamiento	= 6
tipo_inválido	= 7
unidades_faltantes	= 8
unidad_de_entrada_no_encontrada	= 9
OTROS	= 10.

SI sy-subrc NE 0. CONTINUAR. FINSI.

ls_headerdata-peso_medida = lv_salida.

ls_headerdata-peso_unidad = cv_peso_unidad_kg.

AÑadir ls_headerdata A lt_headerdata.

SI lv_salida > lv_peso_máximo. lv_peso_máximo = lv_salida.ENDIF.

FIN DEL BUCLE.

* Coloque el o los productos con el mayor peso en el resultado.
mesa t

ORDENAR lt_headerdata_orig POR id_del_producto.

BUCLE EN lt_headerdata EN ls_headerdata DONDE peso_medida = lv_max_weight.

LEER TABLA lt_headerdata_orig EN ls_headerdata CON CLAVE product_id = ls_headerdata-product_id BÚSQUEDA BINARIA.

BORRAR ls_product.
ls_product-id_producto = ls_headerdata-id_producto.
ls_nombre_producto = ls_nombre-data-header.
ls_categoria_producto = ls_categoria-headerdata.
ls_product-supplierid = ls_headerdata-supplier_id.
ls_medida-peso-producto = ls_peso-datos-encabezado_medida.

ls_unidad-peso-producto = ls_unidad_peso-data-header.

AÑadir ls_product A lt_product.

FIN DEL BUCLE.

copiar_datos_a_referencia(
EXPORTADOR

es_datos = lt_producto
CAMBIO
cr_data = is_data).

CUANDO OTROS.

6 Desarrollo de servicios

```

MÉTODO DE LLAMADA super->/wbep/if_mgw_appl_srv_
tiempo de ejecución~acción de ejecución
    EXPORTADOR
        iv_nombre_acción = iv_nombre_acción
        it_parameter = it_parametro
        contexto_de_solicitud_de_io_tech = contexto_de_solicitud_de_io_tech
    IMPORTADOR
        son_datos
                    = son_datos.

CAJA FINAL.

```

MÉTODO FINAL.

[Listado 6.13 Importación de funciones para DetermineHeaviestProduct](#)

En este método, primero se utiliza una estructura de control de casos, ya que cada importación de función del servicio OData se gestiona a través de este único método ABAP (no se genera ningún submétodo) y es necesario distinguir qué importación de función se supone que se debe gestionar. Para ello, se utiliza el método `get_func tion_import_name()` de `io_tech_request_context`.

El siguiente paso es determinar el parámetro de importación (categoría) utilizando el método `getConvertedParameters` de `io_tech_request_context`.

Este parámetro de filtro se agrega a la tabla de rangos de categorías del módulo de función que utiliza.

Función de llamada
módulo

Luego, llame al módulo de función `BAPI_EPM_PRODUCT_GET_LIST` para proporcionar la lista filtrada de productos. Debido a que un resultado puede contener productos que usan diferentes unidades de peso (por ejemplo, g y kg), primero debe convertir todas las medidas de peso a la misma unidad. Para simplificar, omitiremos aquellos productos en los que la conversión de unidades no fue exitosa. Al recorrer la tabla de resultados, también identifica el valor de peso más alto (`iv_peso_máximo`).

Una vez que haya determinado el peso más alto, seleccione aquellos productos (ya que puede haber varios) con su peso original (sin convertir). Luego, colóquelos en la tabla de resultados `lt_product`. Finalmente, esta tabla se copia en la referencia de datos `er_data` mediante el útil `copy_data_to_ref`

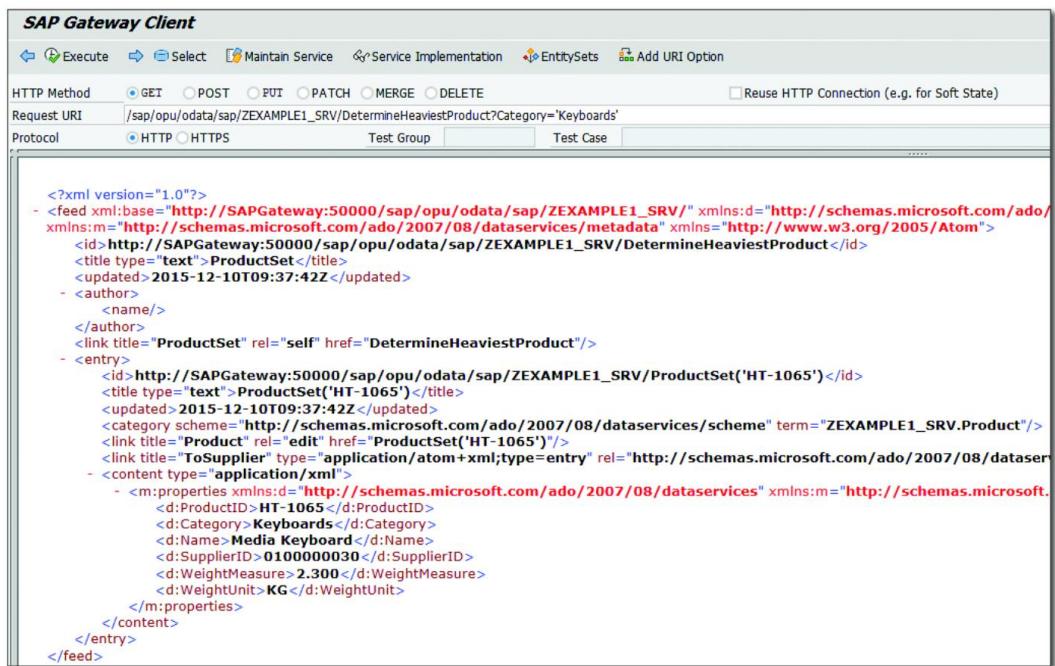
método del marco.

Una vez que haya activado la clase de extensión DPC, puede ejecutar la función de importación. La URI en el cliente SAP Gateway podría ser la siguiente:

/sap/opu/odata/sap/ZEXAMPLE1_SRV/DetermineHeaviestProduct?Categoría='Teclados'

A partir de la estructura de la URI, se puede ver que cada importación de función se ejecuta en función del servicio en sí y no de una colección o una entrada individual. Los parámetros se transfieren de la forma habitual mediante parámetros de URL. Los parámetros múltiples se separan con &. No se pueden utilizar parámetros de tabla (aunque esta funcionalidad está prevista para versiones futuras).

Del resultado de la importación de funciones (ver Figura 6.79), se puede ver que se encontró un solo producto. El filtro de categorías también se utilizó correctamente.



```

<?xml version="1.0"?>
<feed xml:base="http://SAPGateway:50000/sap/opu/odata/sap/ZEXAMPLE1_SRV/" xmlns:d="http://schemas.microsoft.com/ado/2007/08/dataservices/metadata" xmlns:m="http://schemas.microsoft.com/ado/2007/08/dataservices/metadata" xmlns="http://www.w3.org/2005/Atom">
  <id>http://SAPGateway:50000/sap/opu/odata/sap/ZEXAMPLE1_SRV/DetermineHeaviestProduct</id>
  <title type="text">ProductSet</title>
  <updated>2015-12-10T09:37:42Z</updated>
  - <author>
    <name/>
  </author>
  <link title="ProductSet" rel="self" href="DetermineHeaviestProduct"/>
  - <entry>
    <id>http://SAPGateway:50000/sap/opu/odata/sap/ZEXAMPLE1_SRV/ProductSet('HT-1065')</id>
    <title type="text">ProductSet('HT-1065')</title>
    <updated>2015-12-10T09:37:42Z</updated>
    <category scheme="http://schemas.microsoft.com/ado/2007/08/dataservices/scheme" term="ZEXAMPLE1_SRV.Product"/>
    <link title="Product" rel="edit" href="ProductSet('HT-1065')"/>
    <link title="ToSupplier" type="application/atom+xml;type=entry" rel="http://schemas.microsoft.com/ado/2007/08/dataservices/links/related">
      <content type="application/xml">
        - <m:properties xmlns:d="http://schemas.microsoft.com/ado/2007/08/dataservices" xmlns:m="http://schemas.microsoft.com/ado/2007/08/dataservices/metadata">
          <d:ProductID>HT-1065</d:ProductID>
          <d:Category>Keyboards</d:Category>
          <d:Name>Media Keyboard</d:Name>
          <d:SupplierID>0100000030</d:SupplierID>
          <d:WeightMeasure>2.300</d:WeightMeasure>
          <d:WeightUnit>KG</d:WeightUnit>
        </m:properties>
      </content>
    </link>
  </entry>
</feed>

```

Figura 6.79 Resultado de la ejecución de la importación de funciones

6.5.7 Recursos de medios

Los recursos multimedia se utilizan para exponer distintos datos binarios, como gráficos y vídeos, a los consumidores. Se accede a un recurso multimedia a través de una entrada de enlace multimedia.

6 Desarrollo de servicios

Marcar como medio Para permitir que un tipo de entidad regular se convierta en un tipo de entidad de enlace de medios, debe marcarlo como medio a través del Service Builder utilizando la casilla de verificación M en el nivel de tipo de entidad. El mismo atributo también se puede configurar manualmente a través de la codificación en la clase de extensión MPC utilizando el método `set_is_media()` en el nivel de tipo de entidad. Como resultado, el atributo HasStream en el documento de metadatos del tipo de entidad se establece en verdadero.

Tan pronto como se define un tipo de entidad como tipo de medio, también hay que tener en cuenta dos propiedades:

URI de origen

La propiedad URI de origen puede ser utilizada por la aplicación consumidora para recuperar el recurso multimedia asociado desde la ubicación respectiva.

Se recomienda anotar la propiedad que contiene la URI de origen durante el tiempo de ejecución mediante el método `set_as_content_source()`, pero esto es opcional. El marco utiliza esta información durante el tiempo de ejecución para proporcionar una etiqueta de contenido que consta del tipo MIME (consulte el siguiente punto) y la URI.

Si no se define ninguna propiedad como fuente de contenido o si el valor de la propiedad está vacío en tiempo de ejecución, el marco genera automáticamente una etiqueta de origen que utiliza la opción `$value`. Esto permite que el consumidor acceda al valor sin procesar de la entrada.

Tipo MIME El

tipo MIME describe el tipo de recurso multimedia (por ejemplo, imagen/jpeg). La propiedad que contiene el tipo MIME debe anotarse mediante el método `set_as_content_type()`. Es obligatorio definirlo. Si no lo hace, obtendrá un error de tiempo de ejecución al acceder a cualquier dato.

Si la URI de origen contiene una ruta absoluta al recurso multimedia, el consumidor puede acceder al recurso directamente desde su ubicación (por ejemplo, <content-server>:<port>/path/subpath/.../picture.jpg). Pero esto requiere que la aplicación que consume tenga acceso directo a esa ubicación multimedia, lo que podría no estar disponible de manera predeterminada debido, por ejemplo, a la autenticación requerida o al acceso a través de un firewall.

`$value` En lugar de acceder al recurso a través de una URI absoluta, también puede acceder a él a través del servidor SAP Gateway utilizando la opción `$value` para obtener el valor sin procesar.

valor de la entrada correspondiente. Para ello es necesario implementar Get_Método Stream en la clase de extensión DPC, que debe devolver el flujo correspondiente. Esto es muy beneficioso porque una aplicación de consumidor continúa comunicándose con el mismo servidor (SAP Gateway) en lugar de conectarse o autenticarse con múltiples destinos o puntos finales.

Para mejorar el servicio ZExample1_SRV, primero debe agregar dos propiedades nuevas: una para la URI de origen y otra para el tipo MIME. Como se mencionó anteriormente, la propiedad de la URI de origen es opcional.

Para nuestro ejemplo, ingrese los nombres de propiedad "PictureURI" (Edm.String, Longitud máxima 255) y "PictureMIMEType" (Edm.String, Longitud máxima 255 128) al tipo de entidad del producto (ver [Figura 6.80](#)).

Properties													
Name	Key	Edm Core Type	Prec.	Scale	Max...	Unit	Property Name	Crea...	Upd...	Sort...	Null...	Filt...	Label
ProductID	<input checked="" type="checkbox"/>	Edm.String	0	0	10			<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	Product Identifier
Category	<input type="checkbox"/>	Edm.String	0	0	40			<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	Product Category
Name	<input type="checkbox"/>	Edm.String	0	0	255			<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	Product Name
SupplierID	<input type="checkbox"/>	Edm.String	0	0	10			<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	Supplier ID
WeightMeasure	<input type="checkbox"/>	Edm.Decimal	13	3	0	WeightUnit		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	Weight Measure
WeightUnit	<input type="checkbox"/>	Edm.String	0	0	3			<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	Weight Unit
PictureURI	<input type="checkbox"/>	Edm.String	0	0	255			<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	Picture URI
PictureMIMEType	<input type="checkbox"/>	Edm.String	0	0	128			<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	Picture MIME Type

Figura 6.80 Propiedades de los recursos multimedia

En la clase de extensión MPC, debe redefinir el método define() con el código que se muestra en el [Listado 6.14](#).

MÉTODO definir.

```
DATOS: lo_entity_type TIPO REF A /iwbep/if_mgw_odata_entity_
tipo,
propiedad           TIPO REF A /iwbep/if_mgw_odata_
lo_property.

super->define().

lo_entity_type = modelo->get_entity_type( 'Producto' ).
lo_tipo_entidad->establecer_es_media().
lo_propiedad = lo_tipo_entidad->obtener_propiedad( 'PictureURI' ).
lo_property->establecer_como_fuente_de_contenido().
lo_propiedad = lo_tipo_entidad->obtener_
propiedad('PictureMIMEType').
```

6 Desarrollo de servicios

lo_property->establecer_como_tipo_de_contenido().

MÉTODO FINAL.

Listado 6.14 Redefinición del método Define

Clase base de MPC Este código primero llama al método define de la superclase (clase base de MPC) donde se llevan a cabo todas las definiciones definidas a través del Service Builder. Luego, convierte el tipo de entidad en un tipo de medio llamando a set_is_media() método. La propiedad PictureURI está configurada para proporcionar la fuente del contenido y la propiedad PictureMIMEType está configurada para proporcionar el tipo de contenido.

Después de haber regenerado el proyecto y activado la clase de extensión MPC, encontrará una nueva etiqueta en el documento de metadatos de su servicio: m:HasStream (ver Figura 6.81).

```
<?xml version="1.0" encoding="UTF-8"?>
<edmx:Edmx xmlns:sap="http://www.sap.com/Protocols/SAPData"
  xmlns:m="http://schemas.microsoft.com/ado/2007/08/dataservices/metadata"
  xmlns:edmx="http://schemas.microsoft.com/ado/2007/06/edmx" Version="1.0">
  <edmx:DataServices m:DataServiceVersion="2.0">
    - <Schema xml:lang="en" xmlns="http://schemas.microsoft.com/ado/2008/09/edmx" sap:schema-version="1"
      Namespace="ZEXAMPLE1_SRV">
      - <EntityType sap:content-version="1" m:HasStream="true" Name="Product">
        - <Key>
          <PropertyRef Name="ProductID"/>
        </Key>
        <Property Name="ProductID" sap:label="Product Identifier" MaxLength="10" Nullable="false" Type="Edm.String"/>
        <Property Name="Category" sap:label="Product Category" MaxLength="40" Type="Edm.String"/>
        <Property Name="Name" sap:label="Product Name" MaxLength="255" Type="Edm.String"/>
        <Property Name="SupplierID" sap:label="Supplier ID" MaxLength="10" Type="Edm.String"/>
        <Property Name="WeightMeasure" sap:label="Weight Measure" Type="Edm.Decimal" sap:unit="WeightUnit" Scale="3"
          Precision="13"/>
        <Property Name="WeightUnit" sap:label="Weight Unit" MaxLength="3" Type="Edm.String"/>
        <Property Name="PictureURI" sap:label="Picture URL" MaxLength="255" Type="Edm.String"/>
        <Property Name="PictureMIMEType" sap:label="Picture MIME Type" MaxLength="128" Type="Edm.String"/>
        <NavigationProperty Name="ToSupplier" ToRole="ToRole_Assoc_Product_Supplier" FromRole="FromRole_Assoc_Prod
          Relationship="ZEXAMPLE1_SRV.Assoc_Product_Supplier"/>
      </EntityType>
```

Figura 6.81 Tipo de entidad de medios

A continuación, debe ocuparse del suministro de datos. Para ello, vuelve a ajustar PRODUCTSET_GET_ENTITYSET, PRODUCTSET_GET_ENTITY, PRODUCTSET_CREATE_ENTITY, PRODUCTSET_UPDATE_ENTITY y EXECUTE_ACTION. métodos para mapear también las nuevas propiedades.

Tipo MIME Lamentablemente, el tipo MIME no forma parte de la estructura HEADERDATA y, por lo tanto, debe calcularse manualmente. Esto suele ser necesario porque no siempre hay un campo de origen adecuado que se pueda tomar de la estructura/tabla de retorno de una determinada función. En este ejemplo,

Agregue esta determinación al método Get_EntitySet del producto de la clase de extensión DPC.

Junto con los ejemplos anteriores, PRODUCTSET_GET_ENTITYSET

El método ha crecido hasta lo que se muestra en [el Listado 6.15](#).

MÉTODO productset_get_entityset.

DATOS: filtro lr_filter, TIPO REF A /iwbeplif_mgw_req_

lt_filter_select_options TIPO /iwbeplmgw_select_	
opción,	
ls_filter_select_options TIPO /iwbeplmgw_select_	
opción,	
opción ls_select	TIPO /iwbeplcod_select_
opción,	
lt_selparamproductid	TABLA ESTÁNDAR DE TIPO DE bapi_
rango de id_del_producto_epm,	
ls_selparamproductid	TIPO bapi_epm_product_id_
rango,	
lt_selparamcategories	TABLA ESTÁNDAR DE TIPO DE bapi_
epm_rango_categoria_producto,	
categorías ls_selparam	TIPO bapi_epm_categoria_producto_
rango.	

DATOS: lt_headerdata	TABLA ESTÁNDAR DE TIPO DE bapi_
encabezado_producto_epm,	
ls_headerdata	TIPO bapi_epm_product_header,
ls_producto	COMO LÍNEA DE et_entityset.

DATOS: lv_maxrows	TIPO bapi_epm_max_rows,
lv_top	TIPO int4,
lv_skip	TIPO int4,
lv_start	TIPO int4,
lv_end	TIPO int4,
lv_has_inlinecount	TIPO abap_bool.

DATOS: lr_mr_api TIPO REF A if_mr_api.

```
lr_mr_api = cl_mime_repository_api->if_mr_api->get_api( ).
```

```
lr_filter = contexto_solicitud_io_tech->obtener_filtro().
lt_filter_select_options = lr_filter->obtener_selección_de_filtro_
opciones( ).
```

BUCLE EN lt_filter_select_options EN ls_filter_select_
Opciones.

SI ls_filter_select_options-property EQ 'PRODUCTID'.

6 Desarrollo de servicios

BUCLE EN ls_filter_select_options-select_options EN ls_select_option.

ls_selparamproductid-signo = ls_select_option-signo. ls_selparamproductid-opción = ls_select_option-opción. ls_selparamproductid-bajo = ls_select_option-bajo.
ls_selparamproductid-alto = ls_select_option-alto.

ANEXAR ls_selparamproductid A lt_selparamproductid.

FIN DEL BUCLE.

ELSEIF ls_filter_select_options-property EQ 'CATEGORÍA'.

BUCLE EN ls_filter_select_options-select_options EN ls_select_option.

ls_selparamcategories-sign = ls_select_option-sign. ls_selparamcategories-option = ls_select_option-option. ls_selparamcategories-low = ls_select_option-low.
ls_selparamcategories-high = ls_select_option-high.

AÑADIR ls_selparamcategories A lt_selparamcategories.

FIN DEL BUCLE.

FIN.SI.

FIN DEL BUCLE.

lv_maxrows-bapimaxrow = 0. lv_top =
contexto_solicitud_tecnología_io->obtener_top(). lv_skip =
contexto_solicitud_tecnología_io->obtener_skip(). lv_tiene_conteo_en_línea =
contexto_solicitud_tecnología_io->tiene_conteo_en_línea().

SI (lv_top NO ES INICIAL) Y

(lv_has_inlinecount EQ abap_false).lv_maxrows-bapimaxrow
= lv_top + lv_skip.

FIN.SI.

FUNCTION DE LLAMADA 'BAPI_EPM_PRODUCT_GET_LIST'

EXPORTADOR

filas máximas = lv_maxfilas

TABLAS

headerdata = lt_headerdata =

* selparamproductid = lt_selparamproductid =

* NOMBRESDEPROVEEDORESDESELPARAMSelparamcategories
* DEVOLVER =

SI lv_has_inlinecount EQ abap_true.

es_response_context-inlinecount = líneas(lt_headerdata).

FIN.SI.

lv_inicio = 1.

SI lv_skip NO ES INICIAL. lv_start =

lv_skip + 1.

FIN.SI.

```

SI lv_top NO ES INICIAL.
    lv_fin = lv_arriba + lv_inicio - 1.
DEMÁS.
    lv_end = líneas(lt_headerdata).
FIN.SI.

BUCLE EN lt_headerdata EN ls_headerdata
DESDE lv_start HASTA lv_end.
ls_product-id-producto ls_categoria-          = ls_headerdata-id_del_producto.
producto ls_nombre-producto                   = categoría ls_headerdata.
ls_id-proveedor-producto                  = nombre-ls_headerdata.
ls_medida-peso-producto =                 = ls_headerdata-id_del_proveedor.
ls_headerdata-medida_peso.                 = unidad_peso_ls_headerdata.
ls_unidad-peso-producto ls_uri-           = ls_headerdata-url_imagen_producto.
imagen-producto                         = ls_headerdata-url_imagen_producto.

```

```

SI ls_product-pictureuri NO ES INICIAL.
    MÉTODO DE LLAMADA lr_mr_api->get
        EXPORTADOR
            i_url                      = ls_producto-imagenuri
        IMPORTANDO
            EXCEPCIONES                = ls_tipo-mime-imagen-producto
            de e_mime_type
                parámetro_faltante = 1
                error_ocurrido = 2
                no_encontrado = 3
                error_de_permiso = 4
                OTROS = 5.
FIN.SI.

```

AÑADIR ls_product A et_entityset.
FIN DEL BUCLE.

MÉTODO FINAL.

Listado 6.15 Conjunto de productos: método Get_Entityset con filtros, paginación del lado del cliente, recuento en línea y determinación de tipo MIME

En este ejemplo específico, el PictureURI contiene una ruta absoluta al repositorio MIME. Esta es una característica especial de los módulos RFC del producto EPM. Por lo tanto, puede utilizar la API del repositorio MIME para determinar el tipo MIME de la imagen a través del método GET .

Como resultado, verá que la propiedad PictureMIMEType está completa para aquellas entidades de la colección que tienen un PictureURI válido ([consulte la Figura 6.82](#)). Además, verá que una etiqueta de contenido con los valores type y src está

6 Desarrollo de servicios

se representa en el resultado. También hay un nuevo vínculo que apunta a los datos sin procesar de la entidad (\$value).

```
-<entry>
<id>http://SAPGateway:50000/sap/opu/odata/sap/ZEXAMPLE1_SRV/ProductSet('HT-1000')</id>
<title type="text">ProductSet('HT-1000')</title>
<updated>2015-12-10T14:22:00Z</updated>
<category scheme="http://schemas.microsoft.com/ado/2007/08/dataservices/scheme" term="ZEXAMPLE1_SRV.Product"/>
<link title="Product" href="ProductSet('HT-1000')/$value" rel="edit"/>
<link href="ProductSet('HT-1000')/$value" rel="edit-media" type="image/jpeg"/>
<link title="ToSupplier" href="ProductSet('HT-1000')/ToSupplier" rel="http://schemas.microsoft.com/ado/2007/08/dataservices/content-type=image/jpeg src=/sap/public/bc/NWDEMO_MODEL/IMAGES/HT-1000.jpg"/>
<m:properties xmlns:m="http://schemas.microsoft.com/ado/2007/08/dataservices/metadata" xmlns:d="http://schemas.microsoft.com/ado/2007/08/dataservices/typemedia">
<d:ProductID>HT-1000</d:ProductID>
<d:Category>Notebooks</d:Category>
<d:Name>Notebook Basic 15</d:Name>
<d:SupplierID>0100000000</d:SupplierID>
<d:WeightMeasure>4.200</d:WeightMeasure>
<d:WeightUnits>KG</d:WeightUnits>
<d:PictureURI>/sap/public/bc/NWDEMO_MODEL/IMAGES/HT-1000.jpg</d:PictureURI>
<d:PictureMIMEType>image/jpeg</d:PictureMIMEType>
</m:properties>
</entry>
```

Figura 6.82 Entidad con información de recursos multimedia

Asegúrese de ajustar también el método Get_Entity del producto (PRODUCTSET_GET_ENTITY) de la clase de extensión DPC con la misma lógica para determinar también la propiedad PictureMIMEType . Esto garantiza que Get_Entity y Get_EntitySet proporciona las mismas propiedades. La codificación final se muestra en el listado 6.16.

MÉTODO producto_obtener_entidad.

DATOS: ls_entity COMO er_entity,
 ls_product_id TIPO bapi_epm_product_id,
 ls_headerdata TIPO bapi_epm_product_header.

DATOS: lr_mr_api TIPO REF A if_mr_api.

lr_mr_api = cl_mime_repository_api=>if_mr_api~get_api().
 io_tech_request_context->obtener_claves_convertidas(
 IMPORTADOR
 es_key_values = ls_entity).

ls_product_id-product_id = ls_entity-productid.

FUNCTION DE LLAMADA 'BAPI_EPM_PRODUCT_GET_DETAIL'
 EXPORTADOR

id del producto IMPORTADOR	= ls_id_producto
datos de encabezado	= datos del encabezado ls
*TABLA DE CUADROS	=
*	
FACTORES DE CONVERSIÓN	

```

*          DEVOLVER      =
.

er_entity-productid er_entity-      = ls_headerdata-id_del_producto.
category er_entity-name      = categoría ls_headerdata.
er_entity-supplierid      = nombre_ls_headerdata.
er_entity-weightmeasure =      = ls_headerdata-id_del_proveedor.
ls_headerdata-weight_measure.
unidad de peso de la entidad er_entity      = unidad_peso_ls_headerdata.
uri de la imagen de la entidad er_entity      = ls_headerdata-url_imagen_producto.

```

SI er_entity-pictureuri NO ES INICIAL.

```

MÉTODO DE LLAMADA lr_mr_api->get
EXPORTADOR
    i_url      = er_entidad-imagenuri
IMPORTANDO
    EXCEPCIONES      = er_entity-imagenmimetype
    de e_mime_type
        parámetro_faltante = 1
        error_ocurrido = 2
        no_encontrado = 3
        error de permiso = 4
        OTROS = 5.

```

FIN.SI.

MÉTODO FINAL.

[Listado 6.16 Conjunto de productos: método Get_Entity con determinación de tipo MIME](#)

La etiqueta content-src ya proporciona una ruta absoluta a la imagen Ruta absoluta En sí mismo. Un consumidor puede llamar a este para recuperar el recurso multimedia.

Pero en este ejemplo, desea proporcionar los datos sin procesar de la entidad que se recupera cuando se utiliza la opción /\$value .

Para ello, debes implementar (redefinir) /IWBEPIF_MGW_APPL_ Método SRV_RUNTIME~GET_STREAM en la clase de extensión DPC. Proporcione la codificación que se muestra en el Listado 6.17.

MÉTODO /iwbeplif_mgw_appl_srv_runtime~obtener_transmisión.

```

DATOS: lv_entityset_name TIPO /iwbeplmgw_tech_name,
       ls_producto TIPO zcl_zexample1_mpc=>ts_producto,
       lr_mr_api TIPO REF A if_mr_api,
       ls_stream TIPO ty_s_media_resource.

```

```

lv_entityset_name = contexto_de_solicitud_io_tech->obtener_conjunto_de_entidades
nombre( ).
```

6 Desarrollo de servicios

CASO lv_entityset_name.
 CUANDO 'ProductSet'.

MÉTODO DE LLAMADA productset_get_entity
 EXPORTADOR

```

iv_nombre_entidad = iv_nombre_entidad
iv_nombre_del_conjunto_de_entidades = iv_nombre_del_conjunto_de_entidades
iv_nombre_fuente = iv_nombre_fuente
es_tecla_tab = es_tecla_tab
contexto_de_solicitud_de_io_tech = contexto_de_solicitud_de_io_tech
it_navigation_path = ruta_de_navegación_it
  
```

IMPORTADOR

```

es_entidad = ls_producto.
  
```

SI NO ls_product-pictureuri ES INICIAL.

```

lr_mr_api = cl_mime_repository_api->if_mr_api-obtener_
api( ).
```

MÉTODO DE LLAMADA lr_mr_api->get
 EXPORTADOR

```

yo_url = ls_producto-imagenuri
  
```

IMPORTADOR

```

e_content = valor-ls_stream
e_mime_type = tipo mime de ls_stream
  
```

EXCEPCIONES

```

parámetro_faltante = 1
error_ocurrido = 2
no_encontrado = 3
error de permiso = 4
OTROS = 5.
  
```

copiar_datos_a_referencia(
 EXPORTADOR

```

es_datos = ls_stream
  
```

CAMBIO

```

cr_data = er_stream).
  
```

FIN.SI.

CUANDO OTROS.

MÉTODO DE LLAMADA super->/wbeplif_mgw_appl_srv_runtime~get_arreglo
 EXPORTADOR

```

iv_nombre_entidad = iv_nombre_entidad
iv_nombre_del_conjunto_de_entidades = iv_nombre_del_conjunto_de_entidades
iv_nombre_fuente = iv_nombre_fuente
es_tecla_tab = es_tecla_tab
it_navigation_path = ruta_de_navegación_it
contexto_de_solicitud_de_io_tech = contexto_de_solicitud_de_io_tech
  
```

IMPORTADOR

```

er_stream = es_corriente
es_respuesta_contexto = es_response_context.
  
```

CAJA FINAL.

MÉTODO FINAL.

Listado 6.17 Método de obtención de transmisión del servicio

La codificación primero verifica si la entidad actual que se va a procesar es un producto (similar a la función de importación, no se genera ningún método individual, por lo que estos métodos generalmente comienzan con una estructura de caso).

A continuación, ejecute el método Get_Entity del producto para obtener el método Execute del producto. detalles. Porque has implementado el manejo de PictureURI Antes, el campo ls_product-pictureuri ahora contiene el valor que necesita para determinar la secuencia, al menos para aquellos productos que tienen una URI. Para esto, vuelva a utilizar la API del repositorio MIME GET Método. Proporciona el contenido real y el tipo MIME. Ambos deben colocarse en una estructura específica de tipo ty_s_media_resource. Al final, se utiliza el método copy_data_to_ref ya conocido para copiar la estructura en un contenedor de datos genérico.

Después de activar la clase de extensión DPC, puede ejecutar la siguiente URI en el cliente SAP Gateway:

/sap/opu/odata/sap/ZEXAMPLE1_SRV/Conjunto de productos('HT-1000')/\$valor

Como resultado, obtendrá la imagen del producto HT-1000 (ver Figura 6.83).



Figura 6.83 Valor bruto del producto HT-1000

6 Desarrollo de servicios

6.5.8 Expandir/Autoexpandir

La opción de consulta \$expand es muy poderosa y le permite proporcionar múltiples entidades y/o conjuntos de entidades en una sola llamada de servicio, en lugar de realizar varias llamadas posteriormente.

Manejado por Marco de trabajo de SAP Gateway	La expansión se lleva a cabo a lo largo de las propiedades de navegación definidas. El marco SAP Gateway maneja esto llamando al Get_EntitySet del DPC relacionado y agrupación del resultado en una tabla o estructura anidada. El marco SAP Gateway conoce las dependencias (asociaciones) entre cada tipo de entidad, tal como se define en los metadatos, y, por lo tanto, sabe qué métodos llamar y cómo agrupar el resultado.
--	---

La única desventaja que tiene al dejar que el marco SAP Gateway haga el trabajo es que ciertas RFC pueden ser llamadas innecesariamente varias veces. Esto se debe a que el marco conoce las dependencias técnicas pero no el contexto comercial de los datos manejados y, por lo tanto, no puede saber que una RFC específica podría proporcionar los datos de todo el árbol en una sola llamada. Para evitar esto, puede redefinir el método GET_EXPANDED_ENTITY del marco DPC (/IWBEPEP/IF_MGW_APPL_SRV_Interfaz_RUNTIME) o el método GET_EXPANDED_ENTITYSET .

En este ejemplo, se ha definido una propiedad de navegación entre el producto y el proveedor. En lugar de realizar varias llamadas de servicio para obtener primero el producto y luego el proveedor, puede utilizar la opción de consulta \$expand para proporcionar ambos servicios con una única llamada de servicio. Para ello, puede, por ejemplo, utilizar la siguiente URI en el cliente SAP Gateway para recuperar los detalles del producto HT-1000 junto con la información del proveedor:

```
/sap/opu/odata/sap/ZEXAMPLE1_SRV/ProductSet('HT-1000')?
$expand=AlProveedor
```

En la respuesta, puede ver que se le proporciona al proveedor un m:inline etiqueta ([ver Figura 6.84](#)).

```

<?xml version="1.0" encoding="UTF-8"?>
<entry xmlns="http://schemas.microsoft.com/ado/2007/08/dataservices" xmlns:m="http://schemas.microsoft.com/ado/2007/08/dataservices/metadata"
      xmlns:d="http://schemas.microsoft.com/ado/2007/08/dataservices/atom">
  <id>http://SAPGateway:50000/sap/opu/odata/sap/ZEXAMPLE1_SRV/ProductSet('HT-1000')</id>
  <title type="text">ProductSet('HT-1000')</title>
  <updated>2015-12-10T14:46:08Z</updated>
  <category scheme="http://schemas.microsoft.com/ado/2007/08/dataservices/scheme" term="ZEXAMPLE1_SRV.Product"/>
  <link title="Product" rel="edit" href="ProductSet('HT-1000')/Value"/>
  <link type="Image/jpeg" rel="edit-media" href="ProductSet('HT-1000')/Value"/>
  <link title="ToSupplier" type="application/atom+xml;type=entry" rel="http://schemas.microsoft.com/ado/2007/08/dataservices/related/ToSupplier" href="ProductSet('HT-1000')/ToSupplier">
    <m:inline>
      <entry m:base="http://SAPGateway:50000/sap/opu/odata/sap/ZEXAMPLE1_SRV/">
        <id>http://SAPGateway:50000/sap/opu/odata/sap/ZEXAMPLE1_SRV/SupplierSet('0100000000')</id>
        <title type="text">SupplierSet('0100000000')</title>
        <updated>2015-12-10T14:46:08Z</updated>
        <category scheme="http://schemas.microsoft.com/ado/2007/08/dataservices/scheme" term="ZEXAMPLE1_SRV.Supplier"/>
        <link title="Supplier" rel="self" href="SupplierSet('0100000000')"/>
        <content type="application/xml">
          <m:properties>
            <d:SupplierID>0100000000</d:SupplierID>
            <d:SupplierName>SAP</d:SupplierName>
          </m:properties>
        </content>
      </entry>
    </m:inline>
  </link>
  <content type="image/jpeg" src="/sap/public/bc/NWDEMO_MODEL/IMAGES/HT-1000.jpg"/>
</m:properties>
<d:ProductID>HT-1000</d:ProductID>
<d:Category>Notebooks</d:Category>
<d>Name>Notebook Basic 15</d>Name>
<d:SupplierID>0100000000</d:SupplierID>
<d:WeightMeasure>4.200</d:WeightMeasure>
<d:WeightUnit>KG</d:WeightUnit>
<d:PictureURI>/sap/public/bc/NWDEMO_MODEL/IMAGES/HT-1000.jpg</d:PictureURI>
<d:PictureMIMEType>image/jpeg</d:PictureMIMEType>
</m:properties>

```

Figura 6.84 Detalles del producto con \$expand a Proveedor

Para demostrar la redefinición del método GetExpandedEntity , primero debe mejorar el modelo de datos. Agregue otro tipo de entidad con el nombre ProductConvFactor al modelo. Sus propiedades se muestran en la Figura 6.85.

Mejorar los datos modelo

Properties													Filt.	Label
	Name	Key	Edm Core Type	Prec.	Scale	Max...	Unit	Property Name	Cre...	Upd...	Sort...	Null...		
	ProductID	<input checked="" type="checkbox"/>	Edm.String	0	0	10			<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	Product Identifier
	SourceUnit	<input checked="" type="checkbox"/>	Edm.String	0	0	3			<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	Source Unit
	TargetUnit	<input checked="" type="checkbox"/>	Edm.String	0	0	3			<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	Target Unit
	Numerator	<input type="checkbox"/>	Edm.Int32	0	0	0			<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	Numerator
	Denominator	<input type="checkbox"/>	Edm.Int32	0	0	0			<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	Denominator

Figura 6.85 Propiedades del tipo de entidad ProductConvFactor

Cree un conjunto de entidades con el nombre ProductConvFactorSet que utilice el nuevo tipo de entidad. Cree una asociación con el nombre Assoc_Product_ProdConvFactor que define una relación 1:M entre el Producto y los tipos de entidad ProductConvFactor (ver Figura 6.86).

6 Desarrollo de servicios

Associations					
Name	E...	Principal Entity	Principal Entity Cardinality	Dependent Entity	Dependent Entity Cardinality
Assoc_Product_Supplier		Product	M	Supplier	1
Assoc_Product_ProdConvFactor		Product	1	ProductConvFactor	M

Figura 6.86 Nueva asociación Assoc_Product_ProdConvFactor

Haga doble clic en Restricciones referenciales , y agregue una nueva entrada para el ProductID (consulte la Figura 6.87). Tenga en cuenta que los campos de solo lectura se completan tan pronto como ingresa datos en los otros campos y presiona (Entrar).

Referential Constraints			
Principal Entity	Principal Key	Dependent Entity	Dependent Property
Product	ProductID	ProductConvFactor	ProductID

Figura 6.87 Restricción referencial para ProductID

- Crear propiedad de navegación Cree una propiedad de navegación para el tipo de entidad de producto con el nombre ToConvFactors que utiliza la asociación recién creada (consulte la Figura 6.88).
-

Navigation Properties		
Name	Relationship Name	ABAP Field Name
ToSupplier	Assoc_Product_Supplier	TOSUPPLIER
ToConvFactors	Assoc_Product_ProdConvFactor	TOCONVFACORS

Figura 6.88 Propiedad de navegación ToConvFactors

Genere el proyecto para actualizar los MPC y los DPC. Debido a que agregó un nuevo conjunto de entidades al modelo, se generaron cinco métodos nuevos (CRUD-Q) en la clase base DPC, que se heredan a la clase de extensión.

Redefina el factor de conversión del método Get_EntitySet PRODUCTCONVFAC-TO_GET_ENTITYSET en la clase de extensión DPC y proporcione la codificación que se muestra en el [Listado 6.18](#).

MÉTODO productoconvfacto_get_entityset.

```
DATOS: lv_source_entity_set_name TIPO /wbep/mgw_tech_name,
       ls_product          TIPO zcl_zexample1_mpc=>ts_
       producto,
       ls_product_id        TIPO bapi_epm_product_id,
       ls_conversion_factors factores,      TIPO bapi_epm_product_conv_
```

```

factores_de_conversión_lt          TABLA ESTÁNDAR DE TIPO DE bapi_
factores_de_conversión_del_producto_epm,
    factor de conversión ls          COMO LA LÍNEA DE et_entityset,
    lt_
devolver                           TABLA ESTANDAR TIPO bapiret2.

```

```

iv_nombre_conjunto_entidad_origen = contexto_solicitud_io_tech->obtener_
nombre_conjunto_entidad_origen().

```

* ¿Es esta una navegación desde el ProductSet?

```

SI lv_source_entity_set_name EQ 'Conjunto de productos'.
    io_tech_request_context->obtener_claves_de_origen_convertidas(
        IMPORTADOR
            es_key_values = ls_product.
    )
    ls_product_id-product_id = ls_product-productid.

```

FUNCTION DE LLAMADA 'BAPI_EPM_PRODUCT_GET_DETAIL'
EXPORTADOR

*	id del producto	= ls_id_producto
*	IMPORTADOR	=
*	DATOS DEL ENCABEZADO	

Tablas

Factores de conversión de	= factores_de_conversión_lt
retorno	= lt_retorno.

```

SI lt_return NO ES INICIAL.
    mo_context->obtener_contenedor_de_mensajes()->agregar_mensajes_de_
bapi(
    it_bapi_messages = lt_return
    iv_determinar_mensaje_principal = /wbep/si_mensaje_
contenedor=>gcs_leading_msg_search_option-first).

```

```

GENERA EL TIPO DE EXCEPCIÓN /wbep/cx_mgw_busi_exception
EXPORTADOR
    excepción = /wbep/cx_mgw_busi_
de texto =>error_comercial
    contenedor_de_mensajes = mo_context->obtener_mensaje_
contenedor( ).
```

FIN.SI.

BUCLE EN lt_conversion_factors EN ls_conversion_factors.

```
ls_conv_factor-productid = ls_conversion_factors-product_id.
```

```
ls_conv_factor-sourceunit = ls_conversion_factors-source_unit.
```

```
ls_conv_factor-unidad_objetivo = ls_conversion_factors-unidad_objetivo.
```

```
ls_conv_factor-numerador = ls_conversion_factors-
```

6 Desarrollo de servicios

numerador.

```
ls_conv_factor-denominador = ls_conversion_factors-denominador.
```

AÑADIR ls_conv_factor A et_entityset.

FIN DEL BUCLE.

DEMÁS.

MÉTODO DE LLAMADA super->productconvfacto_get_entityset
EXPORTADOR

```
iv_nombre_entidad = iv_nombre_entidad
iv_nombre_del_conjunto_de_entidades = iv_nombre_del_conjunto_de_entidades
iv_nombre_fuente = iv_nombre_fuente
it_filter_select_options = opciones_seleccionadas_de_filtro
es_paginación = es_paginación
es_tecla_tab = es_tecla_tab
it_navigation_path = ruta_de_navegación_it
es_orden = es_orden
cadena_filtro_iv = cadena_filtro_iv
iv_cadena_de_búsqueda = iv_cadena_de_búsqueda
contexto_de_solicitud_de_io_tech = contexto_de_solicitud_de_io_tech
IMPORTADOR
et_entityset = et_conjunto de entidades
es_response_context ENDIF. = es_response_context.
```

MÉTODO FINAL.

[Listado 6.18 Conjunto de factores de conversión de productos: método Get_Entityset](#)

Activar DPC
clase de extensión Active la clase de extensión DPC. Ahora puede ejecutar la siguiente URL para enumerar todos los factores de conversión del producto HT-1000:

```
/sap/opu/odata/sap/ZEXAMPLE1_SRV/Conjunto de productos('HT-1010')/  
Para factores de conversión
```

La declaración de expansión correspondiente para obtener el producto junto con los factores de conversión es la siguiente:

```
/sap/opu/odata/sap/ZEXAMPLE1_SRV/Conjunto de productos('HT-1010')?$expand=  
Para factores de conversión
```

Como puede ver, esto es bastante sencillo y funciona bien. El único problema es que la instrucción \$expand anterior ejecuta BAPI_EPM_

El módulo de función PRODUCT_GET_DETAIL se ejecuta dos veces: primero para obtener el detalle del producto y segundo para obtener la lista de factores de conversión. En este pequeño ejemplo, esto podría no ser un problema, pero en un escenario real de acceso,

por ejemplo, datos de ventas de SAP CRM sobre múltiples tipos de entidades, que pueden tener un impacto significativo en el rendimiento.

Por lo tanto, es posible redefinir los métodos del marco GET_EXPANDED_ENTITY y/o GET_EXPANDED_ENTITYSET como ya se mencionó anteriormente. En estos métodos, puede expandir los datos usted mismo y devolver el resultado en una tabla o estructura anidada. Esto le brinda total flexibilidad sobre qué expansiones desea manejar usted mismo y cuáles desea que maneje el marco. También puede manejar \$expand parcialmente si

Redefinir
estructura
métodos

desear.

En el siguiente ejemplo, manejará el \$expand de los factores de conversión, así que navegue a la clase de extensión DPC y redefina GET_METHODO EXPANDED_ENTITY de la interfaz /IWBEPEP/IF_MGW_APPL_SRV_RUNTIME .

Proporcione la codificación que se muestra en el [Listado 6.19](#).

MÉTODO /iwbepep/if_mgw_appl_srv_runtime~obtener_entidad_expandida.

* Tipo de resultado anidado

DATOS: INICIO DE ls_prod_convfactors.

 INCLUYE TIPO zcl_zexample1_mpc=>ts_product.

 DATOS: toconvfactors TIPO TABLA ESTÁNDAR DE zcl_zexample1_mpc=>ts_productconvfactor CON CLAVE PREDETERMINADA,

 FIN DE ls_prod_convfactors.

 DATOS: ls_product TIPO zcl_zexample1_mpc=>ts_product,
 ls_product_id TIPO bapi_epm_product_id,
 ls_headerdata TIPO bapi_epm_product_header,
 ls_conv_factors TIPO bapi_epm_product_conv_factors,
 lt_conv_factors TABLA DE TIPO DE bapi_epm_product_conv_factores,
 ls_conv_factor TIPO zcl_zexample1_mpc=>ts_factorconvproducto,
 lt_retorno TABLA ESTANDAR TIPO bapiret2.

 DATOS: lv_entityset_name TIPO /iwbepep/mgw_tech_nombre,
 lv_comparar_resultado_prod_convfac TIPO io_expand->ty_e_comparar_resultado,
 cláusula_expandida_ls ME GUSTA LA LINEA DE et_cláusulas_técnicas_expandidas.

 DATOS: lr_mr_api TIPO REF A if_mr_api.

6 Desarrollo de servicios

```

lr_mr_api = cl_mime_repository_api->if_mr_api->get_api( ).

lv_nombre_conjunto_entidad           = contexto_solicitud_io_tech-
                                         >obtener_nombre_conjunto_entidad() .lv_comparar_resultado_prod_convfac = io_expand->comparar_con_tecnología_
                                         nombres( 'TOCONVFACTORS' ).

* ¿Ampliar producto/factores de conversión?
  SI lv_entityset_name EQ 'Conjunto de productos' Y
    (lv_compare_result_prod_convfac EQ io_expand->gcs_compare_
    resultado-coincidencia_subconjunto
    O lv_comparar_resultado_prod_convfac EQ io_expand->gcs_comparar_resultado-
    coincidencia igual ).

  io_tech_request_context->obtener_claves_convertidas( IMPORTANDO
  es_key_values
  = ls_product ).

ls_product_id-product_id = ls_product-productid.

FUNCIÓN DE LLAMADA 'BAPI_EPM_PRODUCT_GET_DETAIL'
EXPORTANDO
  product_id          = ls_id_producto
  IMPORTANDO
  headerdata          = datos del encabezado ls
  TABLAS
  conversion_factors = lt_conv_factors return = lt_return.

SI lt_return NO ES INICIAL.
  mo_context->obtener_contenedor_de_mensajes()->agregar_mensajes_de_
  bapi(
    it_bapi_messages = lt_return iv_determinar_mensaje_principal
    = /wbep/if_message_
  contenedor=>gcs_leading_msg_search_option-first ).

GENERA EL TIPO DE EXCEPCIÓN /wbep/cx_mgw_busi_exception
EXPORTANDO
  textid              = /wbep/cx_mgw_busi_
  excepción=>error_comercial
  contenedor_de_mensajes = mo_context->obtener_mensaje_contenedor( ).

FIN.SI.

  ls_prod_convfactors-id del producto          = ls_headerdata-producto_
  identificación.
  ls_prod_convfactors-categoría ls_prod_convfactors-  = ls_headerdata-categoría. = ls_headerdata-
  nombre ls_prod_convfactors-id del          nombre. = ls_headerdata-proveedor_
  proveedor

```

```

identificación.

ls_prod_convfactors-peso-medida = ls_headerdata-peso_
medida.

ls_prod_convfactors-weightunit unidad. = ls_headerdata-peso_

ls_prod_convfactors-pictureuri url de imagen. = ls_headerdata-producto_


SI ls_prod_convfactors-pictureuri NO ES INICIAL.
MÉTODO DE LLAMADA lr_mr_api->get
EXPORTADOR
    yo_url = ls_prod_convfactors-imagenuri
IMPORTADOR
    tipo_mime_e = ls_prod_convfactors-
tipo_mime_imagen
EXCEPCIONES
    parámetro_faltante = 1
    error_ocurrido = 2
    no_encontrado = 3
    error_de_permiso = 4
    OTROS = 5.
FIN.SI.

BUCLE EN lt_conv_factors EN ls_conv_factors.
ls_conv_factor-productid = ls_conv_factors-product_id.
ls_conv_factor-unidad_origen = ls_conv_factors-unidad_origen.
ls_conv_factor-unidad_objetivo = ls_conv_factors-unidad_objetivo.
ls_conv_factor-numerador = ls_conv_factors-numerador.
ls_conv_factor-denominador = ls_conv_factors-denominador.
AGREGAR ls_conv_factor A ls_prod_convfactors-
paraconvfactores.
FIN DEL BUCLE.

copiar_datos_a_referencia(
EXPORTADOR
    es_datos = ls_prod_convfactors
CAMBIO
    cr_data = er_entidad ).

cláusula ls_expanded = 'TOCONVFACTORS'.
AGREGAR ls_expanded_clause A et_expanded_tech_clauses.

DEMÁS.
super->/wbep/if_mgw_appl_srv_runtime~obtener_entidad_expandida(
EXPORTADOR
    iv_nombre_entidad = iv_nombre_entidad
    iv_nombre_conjunto_entidad = iv_nombre_del_conjunto_de_entidades
    iv_nombre_origen = iv_nombre_fuente
    io_expand = io_expand

```

6 Desarrollo de servicios

```

es_tecla_tab = es_tecla_tab
it_navigation_path = ruta_de_navegación_it
contexto_de_solicitud_de_io_tech = contexto_de_solicitud_de_io_tech
IMPORTADOR
es_entidad = es_entidad
et_cláusulas_expandidas = et_cláusulas_expandidas
et_expanded_tech_clauses = et_expanded_tech_clauses ).
FIN.SI.

```

MÉTODO FINAL.

[Listado 6.19](#) Método de obtención de entidad expandida para la expansión de factores de conversión

Esta codificación parece un poco larga, pero en realidad no es demasiado compleja. La parte de declaración de datos define una estructura anidada que puede contener el producto, así como la lista de factores de conversión. Los factores de conversión van en un campo de tabla con el nombre ToConvFactors, que siempre debe ser igual al nombre de la propiedad de navegación definida. Luego, necesita algunas tablas/estructuras para llamar al módulo RFC, así como algunos campos para algunos datos del marco.

El campo lv_compare_result_prod_convfac basado en io_expand->ty_e_compare_result es muy importante porque se utilizará en io_llamada expand->compare_to_tech_names para determinar dónde se encuentra en el árbol de expansión. Como se trata de un árbol bastante pequeño, la comparación dará como resultado resultados de coincidencia. Dentro de la declaración IF , se determina la clave principal del producto y se llama al módulo de función BAPI_EPM_PRODUCT_GET_DETAIL solo una vez, recuperando los HEADERDATA del producto junto con la tabla de factores de conversión de una sola vez.

Relleno anidado estructura A continuación, debe completar la estructura anidada que se definió al principio con los resultados de la llamada al módulo de función. La estructura anidada se copia a la referencia de datos que devuelve. Por último, debe indicarle al marco que realmente se ha ocupado de \$expand agregando una línea a et_expanded_tech_clauses que especifique la propiedad de navegación (ToConvFactors) que ha manejado.

La rama else es para todas las demás expansiones que desea que el marco SAP Gateway gestione de forma genérica.

6.5.9 Inserción profunda

Una inserción profunda es la inversión de \$expand. En lugar de recibir una estructura anidada de entradas o colecciones, un consumidor puede enviar una estructura anidada por POST al servidor SAP Gateway. Al hacer esto, puede, por ejemplo, crear un encabezado de pedido de venta junto con la colección de artículos de línea con una única llamada de servicio.

Inversión de
\$expand

En este ejemplo, creará un producto junto con la lista de factores de conversión mediante una inserción profunda. Un producto es una entrada única, mientras que la lista de factores de conversión es una colección.

De manera similar a \$expand, debe implementar el manejo usted mismo en la clase de extensión DPC. La inserción profunda se maneja mediante CREATE_DEEP_ENTITY Método ENTITY de la interfaz del marco /WBEP/IF_MGW_APPL_SRV_RUNTIME .

Si ya ha seguido los pasos de implementación de recursos multimedia, cambie el tipo de entidad de nuevo de un tipo de entidad multimedia a un tipo de entidad normal. De lo contrario, el método CREATE_STREAM debe gestionar la inserción profunda (que, por supuesto, también funciona, pero no es el objetivo de esta discusión). La forma más sencilla de cambiar el tipo de entidad de nuevo a uno normal es comentar la línea `lo_entity_type->set_is_media()` en la clase de extensión MPC. Después de haber cambiado la clase de extensión MPC, asegúrese de activarla.

En la clase de extensión DPC, redefina el método CREATE_DEEP_ENTITY de la interfaz /WBEP/IF_MGW_APPL_SRV_RUNTIME y proporcione la codificación que se muestra en el Listado 6.20.

MÉTODO /wbep/if_mgw_appl_srv_runtime~crear_entidad_profunda.

* Tipo de entrada/resultado anidado
 DATOS: INICIO DE ls_prod_convfactors.
 INCLUYE TIPO zcl_zexample1_mpc=>ts_product.
 DATOS: toconvfactors TIPO TABLA ESTÁNDAR DE zcl_zexample1_mpc=>ts_productconvfactor CON CLAVE PREDETERMINADA,
 FIN DE ls_prod_convfactors.
 DATOS: lv_entityset_name TIPO /wbep/mgw_tech_name,
 Datos del encabezado ls TIPO bapi_epm_product_header,
 ls_conv_factors TIPO bapi_epm_product_conv_factors,

6 Desarrollo de servicios

```

It_conv_factors TABLA DE TIPO DE bapi_epm_product_conv_
factores,
    ls_conv_factor          TIPO zcl_zexample1_mpc=>ts_
factorconvproducto,
    lt_retorno             TABLA DE TIPOS DE bapiret2.

lv_entityset_name = contexto_de_solicitud_io_tech->obtener_conjunto_de_entidades
nombre( ).

CASO lv_entityset_name.
CUANDO 'ProductSet'.
    io_data_provider->read_entry_data(IMPORTANDO es_
datos = ls_prod_convfactors).

    ls_headerdata-product_id id del producto.           = ls_prod_convfactors-
    ls_headerdata-category categoría.                  = ls_prod_convfactors-
    ls_headerdata-name                      = nombre-ls_prod_convfactors.
    ls_headerdata-supplier_id id del proveedor.      = ls_prod_convfactors-
    ls_headerdata-unidad_de_medida          = 'EA'.
    ls_headerdata-código_de_moneda        = 'EUR'.
    ls_headerdata-tax_tarif_code = '1'.
    ls_headerdata-type_code = 'PR'.
    ls_headerdata-peso_medida = ls_prod_convfactors-peso_medida.

    ls_headerdata-weight_unit unidad de peso.         = ls_prod_convfactors-
    ls_headerdata-product_pic_url = ls_prod_convfactors-pictureuri.

BUCLE EN ls_prod_convfactors-toconvfactors EN ls_conv_
factor.
    ls_conv_factors-id_producto = ls_conv_factor-id_producto.
    ls_conv_factors-unidad_origen = ls_conv_factor-unidad_origen.

    ls_conv_factors-unidad_objetivo = ls_conv_factor-unidad_objetivo.

    ls_conv_factors-numerador = ls_conv_factor-numerador.
    ls_conv_factors-denominador = ls_conv_factor-denominador.

AÑADIR ls_conv_factors A lt_conv_factors.
FIN DEL BUCLE.

FUNCIÓN DE LLAMADA 'BAPI_EPM_PRODUCT_CREATE'
EXPORTADOR
    datos de encabezado          = datos del encabezado ls
Tablas

```

```

factores_de_conversión = lt_factores_conv
devolver = lt_retorno.

SI lt_return NO ES INICIAL.
mo_context->obtener_contenedor_de_mensajes()->agregar_mensajes_
desde_bapi(
    it_bapi_messages = lt_return
    iv_determinar_mensaje_principal = /wbep/si_mensaje_
contenedor=>gcs_leading_msg_search_option-first ).

GENERA EL TIPO DE EXCEPCIÓN /wbep/cx_mgw_busi_exception
EXPORTADOR
excepción = /wbep/cx_mgw_busi_
de texto =>error_comercial
contenedor_de_mensajes = mo_context->obtener_mensaje_
contenedor( ).  

FIN.SI.

copiar_datos_a_referencia(
EXPORTADOR
es_datos = ls_prod_convfactors
CAMBIO
or_data = er_deep_entity ).  

CUANDO OTROS.

MÉTODO DE LLAMADA super->/wbep/if_mgw_appl_srv_runtime~create_
entidad profunda
EXPORTADOR
iv_nombre_entidad = iv_nombre_entidad
iv_nombre_del_conjunto_de_entidades = iv_nombre_del_conjunto_de_entidades
iv_nombre_fuente = iv_nombre_fuente
io_data_provider = io_data_provider
es_tecla_tab = es_tecla_tab
it_navigation_path = ruta_de_navegación_it
io_expandir = io_expandir
contexto_de_solicitud_de_io_tech = contexto_de_solicitud_de_io_tech
IMPORTADOR
is_deep_entity = es_entidad_profunda.
ENDCASE.  


```

MÉTODO FINAL.

Listado 6.20 Método de creación de entidad profunda para la creación de productos y factores de conversión con manejo de errores

En primer lugar, defina nuevamente un tipo de datos complejo ls_prod_convfactors que pueda transportar los datos anidados recuperados con la llamada POST . En el caso de la estructura, asegúrese de que se trate de una POST en el ProductSet y no en ningún otro

Definir tipo de datos complejos

6 Desarrollo de servicios

Otra colección. La rama Otros terminará en una excepción técnica porque una inserción profunda no es manejada genéricamente por el marco y por lo tanto debe ser manejada a través de la CREATE_DEEP_ENTITY redefinida. método.

Además de las propiedades del producto, la estructura anidada ls_prod_convfactors tiene un campo de tabla (toconvfactors) que contiene la tabla de factores de conversión. Todas estas propiedades se procesan y se colocan en los campos de datos de la estructura/tabla correspondientes para poder llamar al módulo de función BAPI_EPM_PRO-DUCT_CREATE . El manejo de errores se realiza comprobando si la tabla de retorno está vacía. Si la tabla de retorno tiene entradas, se guarda el registro y se genera una excepción comercial.

Por último, la estructura anidada se copia a una referencia de datos que la devuelve al marco de trabajo a través del parámetro er_deep_entity . Esto es necesario porque un POST requiere la devolución de la entidad creada para al menos informar al consumidor sobre la clave principal de la entidad creada (y tal vez algunos campos calculados). En este ejemplo, no hay un rango de números interno que defina la siguiente clave principal ni una propiedad de entidad que sea predeterminada. Por lo tanto, puede devolver la misma estructura anidada que ha recibido (si la BAPI CREATE se procesó correctamente).

A través del cliente SAP Gateway, puede probar fácilmente la inserción profunda de su servicio. Se recomienda realizar primero una \$expand en cualquier entidad de origen que desee utilizar como patrón y hacer clic en Usar como solicitud Botón para copiar el cuerpo de la respuesta al cuerpo de la solicitud.

Ajustar el ID del producto A continuación, ajusta los datos, ya que el ID del producto debe ser único.

También debes ajustar el ID del producto para los factores de conversión. Luego, debes cambiar el método HTTP a POST y cambiar el URI a ProductSet. Finalmente, puede ejecutar la llamada; si todo es exitoso, obtendrá una estructura anidada de los datos creados (consulte la Figura 6.89). Como verificación, puede realizar una operación GET en el producto recién creado con sus factores de conversión para ver si los datos anidados se conservaron correctamente.

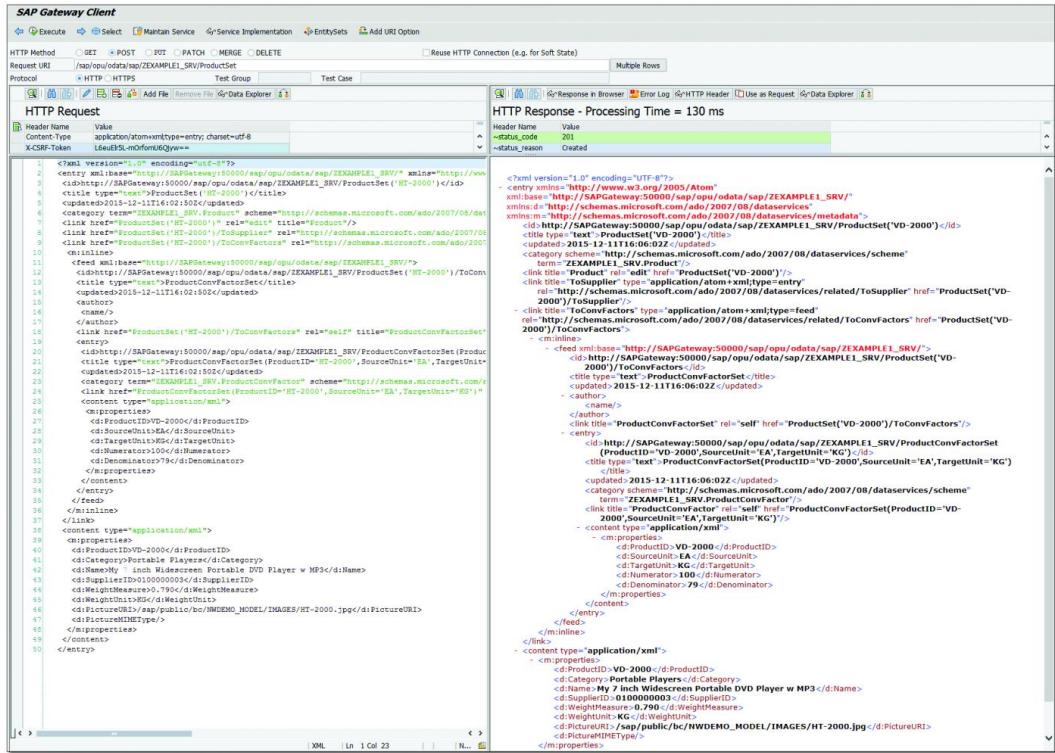


Figura 6.89 Inserción profunda exitosa de productos y factores de conversión

6.5.10 Lote

Una solicitud por lotes de OData se utiliza normalmente cuando un consumidor desea realizar varias llamadas HTTP independientes y quiere evitar múltiples viajes de ida y vuelta al servidor.

Un ejemplo común es la obtención de datos personalizados al iniciar la aplicación, donde normalmente se solicitan los datos de varios conjuntos de entidades independientes.

El procesamiento por lotes simplemente agrupa varias llamadas de servicio OData independientes en una sola llamada (grande). Esas llamadas pueden combinar accesos de lectura y escritura.

5 Combinar lectura y acceso de escritura

Para obtener acceso de escritura, debe definir conjuntos de cambios relacionados en el cuerpo de la solicitud para definir unidades lógicas de trabajo (LUW) para todo lo que debe ejecutarse en su totalidad o revertirse en caso de que surja algún problema.

6 Desarrollo de servicios

Nota

No se puede utilizar el procesamiento por lotes si las llamadas individuales dependen unas de otras (por ejemplo, una llamada requiere los resultados de una llamada anterior).

A partir de SAP_GWFND 7.40 SP 13 y SAP Gateway 2.0 SP 12, se introdujo una nueva característica llamada Referencia de ID de contenido que permite realizar llamadas que dependen entre sí.

Una llamada por lotes siempre se realiza a través del método HTTP POST . El URI para todas las solicitudes \$batch de un servicio OData siempre es el mismo; por ejemplo, el URI para este ejemplo en el cliente SAP Gateway se ve así: /sap/opu/odata/sap/ZEXAMPLE1_SRV/\$batch.

Los pasos individuales que se deben ejecutar deben incluirse en el cuerpo de la solicitud. Esto tiene el efecto secundario positivo de que las URL también están protegidas si se utiliza el protocolo HTTPS.

La solicitud por lotes de OData es un mensaje MIME v1.0 de varias partes en el que cada parte puede tener un tipo de contenido diferente. El lote significativo más sencillo es probablemente la ejecución de dos llamadas get_entity . El cuerpo de una solicitud de este tipo se parece al código que se muestra en el [Listado 6.21](#).

--batch_zmybatch

Tipo de contenido: aplicación/http

Codificación de transferencia de contenido: binario

GET Conjunto de productos ('HT-1000') HTTP/1.1

--batch_zmybatch

Tipo de contenido: aplicación/http

Codificación de transferencia de contenido: binario

GET Conjunto de productos ('HT-1001') HTTP/1.1

--batch_zmybatch--

[Listado 6.21 Ejemplo de cuerpo de solicitud \\$batch](#)

Una solicitud por lotes de OData también requiere configurar el encabezado de solicitud Content-Type según el límite definido. En el caso anterior, el Content-Type debe ser multipart/mixed;boundary=batch_zmybatch.

Si ejecuta esta solicitud por lotes a través del cliente SAP Gateway, obtendrá los detalles de los productos HT-1000 y HT-1001 (consulte la Figura 6.90).

The screenshot shows the SAP Gateway Client interface. The top navigation bar includes 'Execute', 'Select', 'Maintain Service', 'Service Implementation', 'EntitySets', 'Add URI Option', 'Reuse HTTP Connection (e.g. for Soft State)', and 'Multiple Rows'. Below this, the 'HTTP Method' dropdown is set to 'GET'. The 'Request URI' field contains '/sap/opu/odata/sap/ZEXAMPLE1_SRV/\$batch'. The 'Protocol' dropdown is set to 'HTTP / HTTPS'. The 'Test Group' and 'Test Case' tabs are selected. The 'HTTP Request' pane on the left shows the following content:

```

HTTP Request
Header Name Value
Content-Type multipart/mixed;boundary=batch_zmybatch
X-CSRF-Token -mfMDWwrfYK0kHw==

1 --batch_zmybatch
2 Content-Type: application/http
3 Content-Transfer-Encoding: binary
4
5 GET ProductSet('HT-1000') HTTP/1.1
6
7
8 --batch_zmybatch
9 Content-Type: application/http
10 Content-Transfer-Encoding: binary
11
12 GET ProductSet('HT-1001') HTTP/1.1
13
14 --batch_zmybatch--

```

The 'HTTP Response - Processing Time = 919 ms' pane on the right shows the following response:

```

HTTP Response - Processing Time = 919 ms
Header Name Value
~status_code 202
~status_reason Accepted

1 --8334256010C65D18C85D5497FFA590
2 Content-Type: application/http
3 Content-Length: 1536
4 content-transfer-encoding: binary
5
6 HTTP/1.1 200 OK
7 Content-Type: application/atom+xml;type=entry
8 Content-Length: 1423
9 dataserviceversion: 2.0
10
11 <?xml version="1.0" encoding="utf-8"?>
12 <entry xmlns="http://SAPGateway:50000/sap/opu/odata/sap/ZEXAMPLE1_SRV/" xmlns="http://www.w3.org/2005/08/odata"
13 <id href="http://SAPGateway:50000/sap/opu/odata/sap/ZEXAMPLE1_SRV/ProductSet('HT-1000')"/><id>
14 <title type="text">ProductSet('HT-1000')</title>
15 <updated>2015-12-11T16:27:54Z</updated>
16 <category term="ZEXAMPLE1_SRV:Product" scheme="http://schemas.microsoft.com/ado/2007/08/dataservice"
17 <link href="ProductSet('HT-1000')" rel="edit" title="Product"/>
18 <link href="ProductSet('HT-1000')/ToSupplier" rel="http://schemas.microsoft.com/ado/2007/08/dataservice"
19 <link href="ProductSet('HT-1000')/ToConvFactors" rel="http://schemas.microsoft.com/ado/2007/08/dataservice"
20 <content type="application/xml">
21 <mp:properties>
22 <d:ProductID>HT-1000</d:ProductID>
23 <d:Category>Notebooks</d:Category>
24 <d:Name>Notebook Basic 15</d:Name>
25 <d:SupplierID>0100000000</d:SupplierID>
26 <d:WeightMeasure>1.000</d:WeightMeasure>
27 <d:WeightUnit>KG</d:WeightUnit>
28 <d:PictureURI>/sap/public/bc/RWDemo_MODEL/IMAGES/HT-1000.jpg</d:PictureURI>
29 <d:PictureMIMEType>/jpeg</d:PictureMIMEType>
30 </mp:properties>
31 </content>
32 </entry>
33 --8334256010C65D18C85D5497FFA590
34 Content-Type: application/http
35 Content-Length: 1536
36 content-transfer-encoding: binary
37
38 HTTP/1.1 200 OK
39 Content-Type: application/atom+xml;type=entry
40 Content-Length: 1423
41 dataserviceversion: 2.0
42
43 <?xml version="1.0" encoding="utf-8"?>
44 <entry xmlns="http://SAPGateway:50000/sap/opu/odata/sap/ZEXAMPLE1_SRV/" xmlns="http://www.w3.org/2005/08/odata"
45 <id href="http://SAPGateway:50000/sap/opu/odata/sap/ZEXAMPLE1_SRV/ProductSet('HT-1001')"/><id>
46 <title type="text">ProductSet('HT-1001')</title>
47 <updated>2015-12-11T16:27:54Z</updated>

```

Figura 6.90 Solicitud de lote simple con dos llamadas de lectura

Tenga en cuenta que el marcador de límite de batch_zmybatch se establece en el marcador de límite de la solicitud. encabezado, así como delante de cada operación de recuperación precedida por -- en el cuerpo de la solicitud. El tipo de contenido en el nivel de encabezado se establece en multi-part/mixed. Cada operación debe definir su tipo de contenido, que en este ejemplo es application/http.

Asegúrese de proporcionar al menos dos líneas en blanco entre cada operación por lotes. De lo contrario, obtendrá un error de solicitud de servicio mal formada. No olvide concluir el cuerpo de la solicitud HTTP con el marcador de límite de cierre con guiones dobles anteriores (--) así como guiones dobles posteriores (--). La operación por lotes en sí (GET ProductSet('HT-1000')

6 Desarrollo de servicios

HTTP/1.1) necesita definir la operación HTTP (en este caso, GET), así como el recurso y el parámetro adicional, como en la URI.

Dado que este ejemplo de lote simple solo consta de operaciones de recuperación, no es necesario un conjunto de cambios . Siempre que una solicitud de lote contenga operaciones de escritura, debe proporcionar un conjunto de cambios.

Puede utilizar esto, por ejemplo, en una operación UPDATE normal que, según el estándar OData, no devuelve el recurso de entrada actualizado. Pero puede combinar una operación UPDATE junto con una operación GET para obtener los detalles. Una solicitud \$batch correspondiente puede parecerse al código del [Listado 6.22](#).

--batch_zmybatch

Tipo de contenido: multipart/mixed; límite=changeset_zmychangeset

--changeset_zmychangeset

Tipo de contenido: aplicación/http

Codificación de transferencia de contenido: binario

PUT ProductSet('HT-1001') HTTP/1.1 Tipo de

contenido: aplicación/atom+xml

```
<atom:entry xmlns:atom="http://www.w3.org/2005/Atom"> <atom:content
type="application/xml"> <m:properties xmlns:m="http://
schemas.microsoft.com/ado/2007/08/dataservices/metadata" xmlns:d="http://schemas.microsoft.com/
ado/2007/08/dataservices"> <d:IDDeProducto>HT-1001</
d:IDDeProducto> <d:Categoría>Cuadernos</d:Categoría> <d:Nombre>Cuadernos
Básicos 42</d:Nombre> <d:IDDeProveedor>0100000001</
d:IDDeProveedor> <d:MedidaDePeso>4.500</
d:MedidaDePeso> <d:UnidadDePeso>KG</
d:UnidadDePeso> <d:PictureURI>/sap/public/bc/
NWDEMO_MODEL/IMAGES/HT-
```

1001.jpg

```
 <d:PictureMIMEType>imagen/jpeg</d:PictureMIMEType> </m:propiedades> </
atom:contenido> </
atom:entrada>
```

--changeset_zmychangeset--

--batch_zmybatch

Tipo de contenido: aplicación/http

Codificación de transferencia de contenido: binario

GET Conjunto de productos ('HT-1001') HTTP/1.1

--batch_zmybatch-- [Listado](#)

6.22 Ejemplo de llamada por lotes con operación de escritura

Esta solicitud por lotes de OData primero actualiza el producto HT-1001 y luego realiza una solicitud GET para leer el recurso de entrada actualizado. Si hay múltiples operaciones de escritura (CREATE, UPDATE, DELETE) contenidas en un único conjunto de cambios, ninguna operación puede realizar un trabajo de confirmación porque, de lo contrario, no se puede cumplir el paradigma de todo o nada. El marco de trabajo comprueba esto y activa un volcado breve si se infringe esta regla. Esta comprobación se desactiva si el conjunto de cambios solo consta de una única operación de escritura.

El marco SAP Gateway ejecuta un trabajo de confirmación al final de cada trabajo de confirmación. conjunto de cambios. Al comienzo de cada conjunto de cambios, el marco SAP Gateway ejecuta el método CHANGESET_BEGIN (/IWBEPIF_MGW_APPL_Interfaz SRV_RUNTIME). Este método se puede redefinir si desea implementar su propio manejo. Debe redefinirse si el conjunto de cambios contiene cambios en más de un tipo de entidad.

En el método change-set-begin , puede, por ejemplo, establecer una variable miembro para indicar que se está procesando un conjunto de cambios . Esta información se puede utilizar dentro de un método UPDATE para almacenar únicamente los cambios en la memoria o para evitar un trabajo de confirmación.

El método change-set-begin también se puede utilizar para verificar si el conjunto de cambios contiene combinaciones no admitidas de tipos de entidad. El método tiene una tabla de entrada it_operation_info, que enumera todos los tipos de entidad contenidos en el conjunto de cambios actual. Esto le permite reaccionar ante combinaciones que no desea admitir lanzando la excepción correspondiente.

Al final de cada conjunto de cambios, se llama al método CHANGESET_END (misma interfaz). Puede redefinirlo para realizar su propio manejo del final del conjunto de cambios , donde restablece la variable miembro y realiza su trabajo de actualización y/o confirmación de la base de datos.

6 Desarrollo de servicios

6.6 Resumen

En este capítulo se presentó una introducción al desarrollo del servicio OData mediante el conjunto de herramientas Service Builder. Siguiendo un enfoque paso a paso, modelamos, implementamos y ejecutamos el servicio OData.

La definición del modelo se realizó de manera declarativa mediante la creación de tipos de entidad, conjuntos de entidades, propiedades, asociaciones, restricciones referenciales, propiedades de navegación e importaciones de funciones. También analizamos las opciones para importar una definición de modelo desde un archivo EDMX, desde una estructura DDIC/tabla y desde una interfaz BOR/RFC.

La parte de implementación del servicio se realizó en su totalidad redefiniendo o sobrescribiendo los métodos relacionados en la clase de extensión DPC mediante ABAP. Este método funciona bastante bien, aunque a veces puede parecer un poco largo. Es por eso que Service Builder tiene la capacidad de generar la implementación del servicio de los métodos CRUD-Q asignando el método respectivo a una fuente de datos correspondiente (por ejemplo, un módulo RFC).

Esto se explicará más detalladamente en el siguiente capítulo.