

Versión 9 Release 1.2
23 de septiembre de 2015

*IBM Marketing Operations Módulo de
integración*



Note

Before using this information and the product it supports, read the information in "Notices" en la página 33.

Contenido

Capítulo 1. ¿Qué es Servicios de integración de IBM Marketing Operations?

¿Cuáles son los requisitos para los servicios de integración de Marketing Operations?	2
Conceptos básicos de los Servicios de integración de IBM Marketing Operations	3
Instalación de Servicios de integración.	6
Contenido del kit de desarrollador de software..	6
JavaDocs alojados.	7
Documentación y ayuda de Marketing Operations..	7

Capítulo 2. Servicio web de integración de Marketing Operations

WDSL de servicios de integración de Marketing Operations	9
executeProcedure.	9
Tipos de datos del servicio web de integración de Marketing Operations	10

Capítulo 3. Procedimientos de IBM Marketing Operations

Se presupone lo siguiente.	15
Parámetros de configuración.	17
Diseño	17

Ciclo de vida de procedimiento.	18
Clases Java clave	19
Bloqueo de datos	20
Transacciones de procedimiento	20
Comunicación de procedimientos	21
Registro del procedimiento	21
Archivo de definiciones del plugin de procedimiento	21

Capítulo 4. API de IBM Marketing Operations

Contenido de la API de IBM Marketing Operations	23
Interfaces de API	23
Excepciones comunes	24
Manejadores	24
Correlación de atributos	26
Tipos de datos enumerados	27

Antes de contactar con el servicio de soporte técnico de IBM

Notices

Trademarks	35
Privacy Policy and Terms of Use Considerations	35

Capítulo 1. ¿Qué es Servicios de integración de IBM Marketing Operations?

Servicios de integración de IBM®Marketing Operations combina el servicio web de integración de Marketing Operations, los procedimientos de API y los desencadenantes para ampliar las prestaciones empresariales.

Servicios de integración de IBM Marketing Operations es un compuesto de lo siguiente.

- **Servicio web de integración de Marketing Operations**

Los Servicios de integración proporcionan una forma para que los clientes de Marketing Operations y los servicios profesionales de IBM integren Marketing Operations con otras aplicaciones que se ejecuten en su entorno.

- **Procedimientos y API de Marketing Operations**

Los procedimientos personalizados se pueden definir dentro de Marketing Operations para ampliar la lógica empresarial de Marketing Operations de formas arbitrarias. Una vez definidos, estos procedimientos pueden ser los destinos para las llamadas de servicio web de Servicios de integración desde otras aplicaciones. Los procedimientos también se pueden definir para enviar mensajes a otras aplicaciones.

- **Desencadenantes de Marketing Operations**

Los desencadenantes se pueden asociar a eventos y procedimientos en Marketing Operations. Cuando se produce uno de estos eventos, se ejecuta el desencadenante asociado.

Versiones y compatibilidad con versiones anteriores

Las futuras versiones de los servicios de integración serán compatibles con versiones anteriores con todos los releases secundarios y de mantenimiento que compartan el número de versión principal. Sin embargo, IBM se reserva el derecho de romper la compatibilidad con una versión anterior para releases principales de punto cero (x.0) si el caso empresarial o técnico lo garantiza.

El número de versión principal de esta API se incrementará si se realiza cualquiera de los siguientes cambios.

- Cambia la interpretación de datos
- Cambia la lógica empresarial (por ejemplo, cambios de las funciones de método de servicio)
- Cambian los parámetros de método, los tipos de retorno, o ambos

El número de versión secundario de la API se incrementará si se realiza cualquiera de los cambios siguientes. Por definición, estos cambios son compatibles con una versión anterior.

- Se ha añadido un método nuevo
- Se ha añadido un nuevo tipo de datos y su uso está restringido a un nuevo método
- Se ha añadido un elemento nuevo a un tipo enumerado
- Se ha definido una nueva versión con un sufijo de versión

Autenticación

La autenticación no es necesaria; todos los clientes están asociados a un usuario conocido de IBM Marketing Operations llamado PlanAPIUser. Un administrador del sistema configura las prestaciones de seguridad de este usuario especial para cumplir las necesidades de todos los clientes de servicio web.

Entorno local

El único entorno local soportado es el entorno local configurado actualmente para la instancia del sistema de IBM Marketing Operations. Se supone que todos los datos dependientes del entorno local, como mensajes y moneda, están en el mismo entorno local.

Gestión de estado

La API y el servicio web son *sin estado*; la implementación del servicio no guarda información por cliente entre las llamadas a la API. Esta característica proporciona una implementación de servicio eficaz y simplifica el soporte a clúster.

Transacciones de base de datos

Integración de servicios de Marketing Operations no muestra transacciones de base de datos al cliente, pero utiliza esta información si se incluye en el contexto de ejecución. Si se ha iniciado una transacción, el efecto de todas las llamadas a la API dentro de un procedimiento concreto será atómico. En otras palabras, una llamada fallida a la API deja la base de datos en el mismo estado que si no se hubiera llamado nunca a la API. Otros usuarios de Marketing Operations no verán los cambios hasta que el procedimiento confirme satisfactoriamente la transacción.

Las llamadas a la API que actualizan la base de datos deben, en primer lugar, adquirir un bloqueo de edición para evitar que otros usuarios modifiquen los datos subyacentes durante las llamadas a la API. Otros usuarios no podrán actualizar los componentes bloqueados hasta que se complete la llamada a la API. De la misma forma, el siguiente usuario o cliente de la API de Marketing Operations deberá adquirir el bloqueo en los datos antes de que se envíe otra llamada a la API.

Proceso de eventos

Las operaciones en los componentes de IBM Marketing Operations mediante la API generan los mismos eventos que si un usuario de Marketing Operations hubiera realizado la operación. A los usuarios suscritos a determinadas notificaciones, como cambios de estado de proyecto, se les notifican de los cambios de estado resultantes de llamadas a la API y de acciones de los usuarios.

¿Cuáles son los requisitos para los servicios de integración de Marketing Operations?

Los servicios de integración de Marketing Operations tienen los requisitos siguientes.

Los servicios de integración de Marketing Operations deben:

- Acoplar ligeramente la integración del sistema.
- Proporcionar un mecanismo para las aplicaciones de cliente para afectar a Marketing Operations a través de llamadas de servicio web.

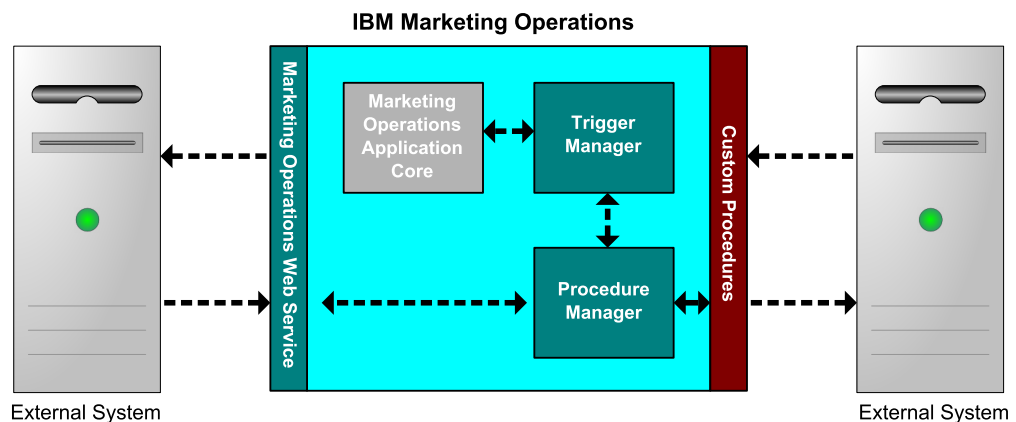
- Proporcionar un mecanismo para que se notifique a las aplicaciones de cliente sobre determinados eventos en Marketing Operations.
- Proporcionar un modelo de programación que sea fácil de comprender y utilizar.
- Ser sólidos al recuperarse de un error.
- Garantizar la integridad de los datos.
- Integrarse con y minimizar el efecto en los clientes basados en la GUI de Marketing Operations existentes.
- Proporcionar un acceso preciso a los componentes de Marketing Operations mientras se aísla a los programadores de los detalles de la implementación subyacente.

Conceptos básicos de los Servicios de integración de IBM Marketing Operations

Puede utilizar los Servicios de integración de IBM Marketing Operations para crear procedimientos personalizados. Puede utilizar estos procedimientos para desencadenar eventos externos cuando se producen determinados eventos dentro de Marketing Operations. Puede utilizar estos procedimientos para ejecutar las funciones de Marketing Operations de programas o sistemas externos.

La interfaz de la API interactúa con IBM Marketing Operations en el nivel programático, de la misma forma que las interfaces de la GUI con Marketing Operations a nivel de usuario. Mediante el uso de la API, puede construir procedimientos. Mediante estos procedimientos se puede comunicar entre Marketing Operations y sistemas externos. El servicio web de Marketing Operations es el objeto de contenedor para los procedimientos, la API y los desencadenantes.

La arquitectura de los servicios de integración de Marketing Operations se muestra aquí.

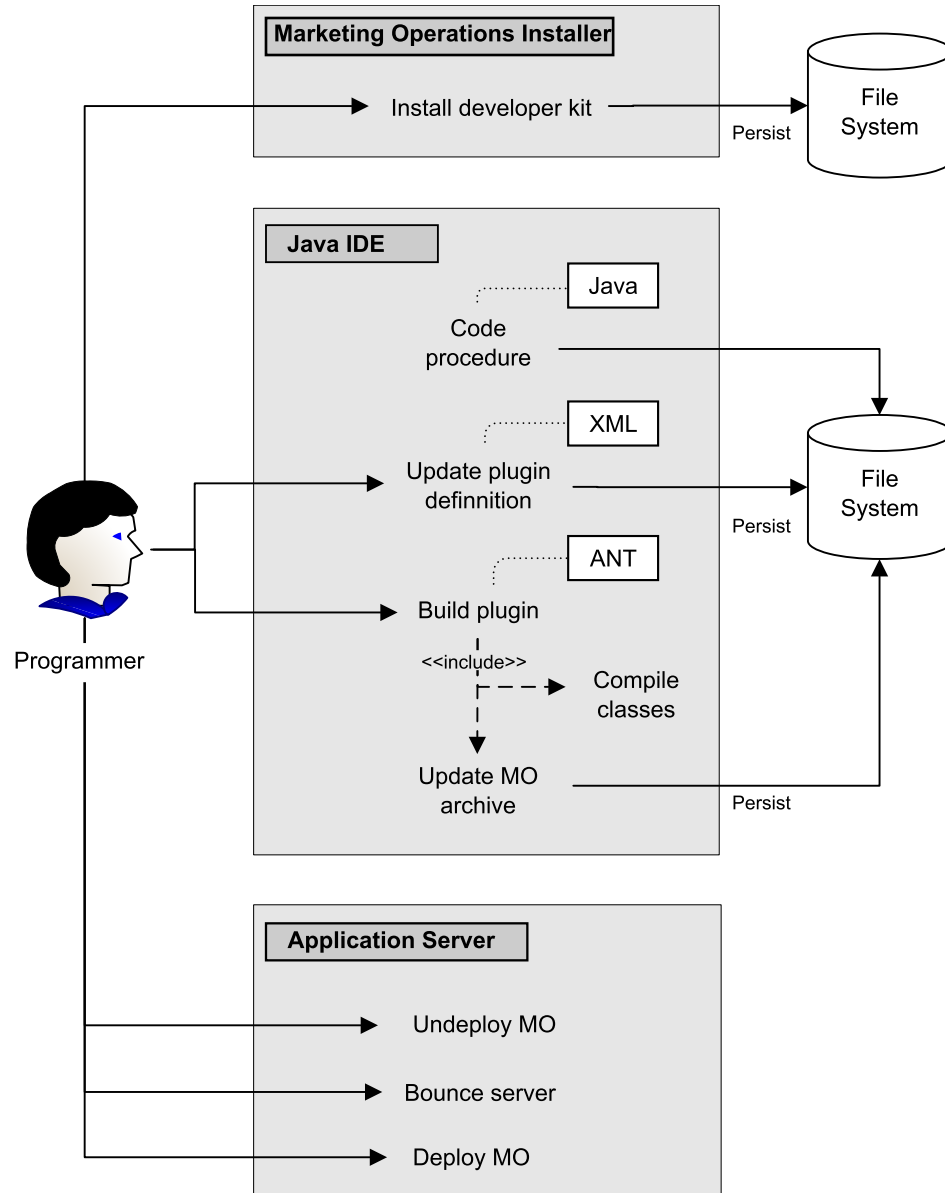


Lo que aparece a continuación son componentes clave de los Servicios de integración.

- Gestor de procedimiento Marketing Operations: amplía la lógica empresarial interactuando con Marketing Operations a través de la API.
- Gestor de desencadenante de Marketing Operations: asocia una condición (por ejemplo, el cambio de estado de un objeto de marketing) con una acción (un procedimiento para ejecutar cuando se cumpla la condición para el desencadenante).

Métodos

Puede utilizar los componentes de los Servicios de integración de IBM Marketing Operations para desarrollar procedimientos personalizados, tal como se muestra en el diagrama siguiente.



Una vez instalado el kit del desarrollador, siga estos pasos básicos:

1. Codifique el procedimiento personalizado. Actualmente, debe utilizar Java™.
2. Actualice la definición de plugin en el archivo de definición XML.
3. Cree el plugin:
 - a. Compile las clases necesarias.
 - b. Si utiliza una biblioteca de terceros que no se encuentra en el archivado de Marketing Operations, empaque la biblioteca dentro del archivo `plan.war` y vuelva a realizar el despliegue.

4. Reinicie Marketing Operations. Los cambios en las clases de procedimiento se aplican al reiniciar el servidor de aplicaciones.

Nota: Si modifica el archivo **plan.war**, debe anular el despliegue y volver a desplegar Marketing Operations con el nuevo archivo **plan.war**. Anule el despliegue y vuelva a desplegar Marketing Operations si utiliza una biblioteca de terceros que no está en el archivado de Marketing Operations y edita el archivo **plan.war**.

Ejemplos básicos para comunicarse entre IBM Marketing Operations y la API

El siguiente ejemplo básico describe el establecimiento de la comunicación entre la API y Marketing Operations. Si esto no realiza ningún trabajo útil; realiza un viaje de ida y vuelta entre Marketing Operations y los Servicios de integración.

Este ejemplo utiliza partes de los procedimientos de ejemplo incluidos con el kit del desarrollador de Servicios de integración de Marketing Operations. Específicamente, puede encontrar el código referenciado aquí en los archivos siguientes.

- PlanClientFacade.java
- PlanWSNOOPTestCase.java

El método noop es una llamada de servicio web a Marketing Operations. Está definido en la clase PlanClientFacade y pasa valores nulos en una matriz.

```
public ProcedureResponse noop(String jobId)
    throws RemoteException, ServiceException {
    NameValueArrays parameters =
        new NameValueArrays(null, null, null, null, null, null, null, null);
    return _serviceBinding.executeProcedure("uapNOOPProcedure", jobId, parameters);
}
```

El procedimiento testExecuteProcedure llama al método noop desde PlanClientFacade para establecer un viaje de ida y vuelta con la aplicación Marketing Operations.

```
public void testExecuteProcedure() throws Exception {
    // Tiempo de espera excedido después de un minuto
    int timeout = 60000;
    PlanClientFacade clientFacade = new PlanClientFacade(urlWebService, timeout);
    System.out.println("noop w/no parameters");
    long startTime = new Date().getTime();
    ProcedureResponse response = clientFacade.noop("junit-jobid");
    long duration = new Date().getTime() - startTime;

    // cero o estado positivo => satisfactorio
    System.out.println("Status: " + response.getStatus());
    System.out.println("Duration: " + duration + " ms");
    assertTrue(response.getStatus() >= 0);
    System.out.println("Done.");
}
```

Si desea ver detalles sobre NameValueArrays, ProcedureResponse y otros métodos y tipos de datos listados, consulte el apartado *Módulo de integración de Marketing Operations* y los JavaDocs.

Instalación de Servicios de integración

El módulo de Servicios de integración de IBM Marketing Operations es un componente aparte de pago. Si adquiere el módulo de Servicios de integración, debe instalarlo.

1. Descargue los instaladores de Servicios de integración de IBM Marketing Operations.
2. Los instaladores de IBM EMM detectan el módulo de Servicios de integración.
3. El instalador establece las propiedades de configuración en **Marketing Operations | umoConfiguration | integrationServices | enableIntegrationServices**. Puede personalizar la instalación modificando los parámetros de configuración. Para obtener más información, consulte "Parámetros de configuración" en la página 17.

Contenido del kit de desarrollador de software

Normalmente, todos los componentes de Servicios de integración de Marketing Operations se instalan en una carpeta denominada "devkits".

El kit de desarrollador de software contiene documentación, Javadocs que contienen todas las clases e interfaces publicapi, y código de ejemplo.

El código de ejemplo se instala en las carpetas siguientes.

- La carpeta **build** contiene scripts para crear y desplegar procedimientos personalizados.
- La carpeta **Classes** contiene las clases de procedimientos compilados.

Los usuarios deben desplegar las clases compiladas de sus procedimientos personalizados en la ruta especificada por el parámetro de configuración **integrationProcedureClasspathURL**. A continuación, el gestor de procedimiento de IBM Marketing Operations los carga tal como se especifica en el archivo de configuración de `procedure-plugins.xml`.

- La carpeta **lib** contiene las bibliotecas necesarias para desarrollar y compilar procedimientos personalizados.
- La carpeta **src** contiene archivos de origen para los procedimientos personalizados. Los usuarios pueden colocar aquí los procedimientos personalizados para que se inicien como desencadenante o servicios web.
 - La carpeta **src/procedure** contiene el archivo de configuración `procedure-plugins.xml`. Cada procedimiento personalizado que se ejecuta como desencadenante basado en un evento o mediante un servicio web externo debe tener una entrada en este archivo. Las entradas deben contener una ruta de clases completa del procedimiento y los parámetros de inicialización necesarios.
 - La carpeta **src/procedure** contiene también algunos procedimientos de ejemplo incluidos con IBM Marketing Operations. Estos procedimientos se pueden utilizar para comprender y desarrollar los procedimientos personalizados.

Coloque los procedimientos personalizados en el directorio **src** en una nueva estructura de carpeta, por ejemplo, `com/<mycompany>/<mypackage>`. No coloque procedimientos personalizados en la carpeta de procedimientos de ejemplo.

- La carpeta **src/soap** contiene clientes de servicio web de ejemplo desarrollados en Java. Utilice estos ejemplos como punto de partida para desarrollar clientes basados en servicio web para Servicios de integración. Esta carpeta contiene también scripts binarios para iniciar clientes de ejemplo mediante la línea de comandos.

JavaDocs alojados

Si desea información específica sobre los métodos de API públicos, consulte la clase iPlanAPI en los archivos de documentación de API JavaDocs.

Estos archivos están disponibles de las siguientes maneras:

- Mediante los archivos en el directorio `<IBM_EMM>/<Inicio_MarketingOperations>/devkits/integration/javadocs` en el servidor que aloja Marketing Operations.
- Iniciando sesión en Marketing Operations y seleccionando **Ayuda > Documentación del producto** en cualquier página y, a continuación, descargando el archivo IBM `<versión>PublicAPI.zip`.

Documentación y ayuda de Marketing Operations

Distintas personas en su organización utilizan IBM Marketing Operations para realizar distintas tareas. La información sobre Marketing Operations está disponible en un conjunto de guías, cada una diseñada para que la utilicen los miembros del equipo con objetivos específicos y conjuntos de habilidades.

La tabla siguiente describe la información disponible en cada guía.

Tabla 1. Guías en el conjunto de documentación de Marketing Operations.

La tabla siguiente de tres columnas describe las tareas en una columna, los nombres de guía en la segunda columna y la audiencia en la tercera columna.

Si	Consulte	Audiencia
• Planifica y gestiona proyectos • Establece tareas de flujo de trabajo, hitos y personas • Realiza un seguimiento de los gastos del proyecto • Obtiene revisiones y aprobaciones del contenido • Genera informes	<i>IBM Marketing Operations</i> Guía del usuario	• Gestores de proyectos • Diseñadores creativos • Directores de marketing de correo directo
• Diseña plantillas, formularios, atributos y medidas • Personaliza la interfaz de usuario • Define seguridad y niveles de acceso de usuario • Implementa características opcionales • Configura y ajusta Marketing Operations	<i>IBM Marketing Operations</i> Guía del administrador	• Gestores de proyectos • Administradores de TI • Asesores de implementación
• Crea campañas de marketing • Planifica ofertas • Implementa integración entre Marketing Operations y Campaign • Implementa integración entre Marketing Operations y IBM Digital Recommendations	<i>IBM Marketing Operations</i> y <i>IBM</i> Guía de integración	• Gestores de proyectos • Especialistas de ejecución de marketing • Directores de marketing directo

Tabla 1. Guías en el conjunto de documentación de Marketing Operations (continuación).

La tabla siguiente de tres columnas describe las tareas en una columna, los nombres de guía en la segunda columna y la audiencia en la tercera columna.

Si	Consulte	Audiencia
• Conozca las nuevas características del sistema • Investigue problemas conocidos y las soluciones	<i>IBM Marketing Operations - Notas del release</i>	Todos los usuarios de Marketing Operations
• Instale la Marketing Operations • Configure Marketing Operations • Actualice a una nueva versión de Marketing Operations	<i>Guía de instalación de IBM Marketing Operations</i>	• Asesores de implementación de software • Administradores de TI • Administradores de bases de datos
Crea procedimientos personalizados para integrar Marketing Operations con otras aplicaciones	<i>IBM Marketing Operations Módulo de integración</i> y los JavaDocs de API disponibles cuando pulsa Ayuda > Documentación del producto en Marketing Operations y, a continuación, descargue el archivo IBM<versión>PublicAPI.zip	• Administradores de TI • Administradores de bases de datos • Asesores de implementación
Conozca la estructura de la base de datos de Marketing Operations	<i>IBM Marketing OperationsEsquema del sistema de origen</i>	Administradores de bases de datos
Necesita más información mientras trabaja	• Puede obtener ayuda y buscar o examinar las <i>Guías del usuario o administrador de Marketing Operations</i>, o la <i>Guía de instalación de Marketing Operations</i>: Pulse Ayuda > Ayuda para esta página • Acceda a todas las guías de Marketing Operations: Pulse Ayuda > Documentación del producto • Acceda a las guías de todos los productos de IBM Enterprise Marketing Management (EMM):Pulse Ayuda > Toda la documentación de IBM EMM	Todos los usuarios de Marketing Operations

Capítulo 2. Servicio web de integración de Marketing Operations

El servicio web proporciona una vista de cliente de los servicios de integración de Marketing Operations, que forman parte del despliegue del servidor IBM Marketing Operations. El servicio web se utiliza simultáneamente con los usuarios web de Marketing Operations.

El servicio web soporta una llamada a la API, `executeProcedure`.

Un cliente realiza esta llamada al servicio web directamente.

WSDL de servicios de integración de Marketing Operations

El WSDL (Web Services Definition Language) se ha definido de forma manual y es la palabra final sobre la definición del servicio web.

Axis

Esta versión del servicio web utiliza Axis2 1.5.2 para generar clases del servidor que conforman la implementación del servicio web desde el archivo WSDL. Los usuarios pueden utilizar cualquier versión de Axis o una técnica no Axis para crear una implementación del lado del cliente para integrarse con la API desde el WSDL proporcionado.

Versión de protocolo

La versión del protocolo se enlaza explícitamente al WSDL del modo siguiente:

- como parte del nombre de WSDL, por ejemplo, `PlanIntegrationService1.0.wsdl`
- Como parte del `targetNamespace` de WSDL, por ejemplo, `xmlns:tns="http://webservices.unica.com /MktOps/services/PlanIntegrationServices1.0?wsdl"`

WSDL

Se proporciona un archivo WSDL con los servicios de integración de IBM Marketing Operations: `PlanIntegrationServices1.0.wsdl`. El WSDL se entrega en el directorio `integration/examples/soap/plan`. El script de creación de muestra utiliza este archivo para generar los stubs del lado del cliente apropiados para conectarse al servicio web.

`executeProcedure`

`executeProcedure` es la llamada a la API que soporta el servicio web.

Sintaxis

```
executeProcedure(string key, string jobid, NameValueArrays paramArray)
```

Devuelve

```
int: estado  
Message[]: mensajes
```

Descripción

Este método invoca al procedimiento especificado con una matriz de parámetros opcionales. La llamada se ejecuta de forma síncrona; es decir, bloquea el cliente y devuelve el resultado después de la finalización.

Parámetros

Tabla 2. Parámetros de *executeProcedure*

Nombre	Descripción
key	La clave exclusiva del procedimiento para ejecutar. Se devuelve un error <i>RemoteException</i> si no hay ningún procedimiento enlazado a la clave.
jobid	Cadena opcional que identifica el trabajo asociado con esta ejecución de procedimiento. Esta cadena es un elemento de paso, pero se puede utilizar para unir los trabajos del cliente con la ejecución de un procedimiento concreto.
paramArray	Una matriz de parámetros para pasar al procedimiento. Se devuelve un estado y mensaje de error si uno o más de los parámetros no es válido (por ejemplo, el tipo es incorrecto o un valor no está permitido). Es decisión del cliente determinar los parámetros, sus tipos y el número de valores necesarios para el procedimiento.

Parámetros de devolución

Tabla 3. Parámetros de devolución de *executeProcedure*

Nombre	Descripción
estado	Un código entero: • 0 indica que el procedimiento se ha ejecutado correctamente • un entero indica un error Los procedimientos pueden utilizar el estado para indicar distintos niveles de errores.
messages	Una matriz de cero o más estructuras de datos de mensaje. Si el estado es 0, esta matriz no contiene mensajes de ERROR, pero podría contener mensajes INFORMATIVOS o de ADVERTENCIA. Si el estado no es cero, los mensajes pueden contener cualquier combinación de mensajes de ERROR, INFORMATIVOS y de ADVERTENCIA.

Tipos de datos del servicio web de integración de Marketing Operations

Los tipos de datos que utiliza el servicio web son independientes de cualquier implementación de programación o enlace de servicio determinado.

Se utiliza la siguiente anotación.

- *<tipo>*: *<definición tipo>* define un tipo de datos sencillo. Por ejemplo:
Manejador: cadena
- *<tipo>*: [*<definición tipo>*] define un tipo de datos o una estructura de datos complejo.

- `<tipo>: { <definición tipo> }` define un tipo de datos o una estructura de datos complejo.

Los elementos de tipo complejos y los parámetros de API pueden utilizar estos tipos para declarar matrices. Por ejemplo:

```
Handle [] handles
```

El tipo, manejadores, es una matriz de tipos de manejador.

Tipos primitivos

Los tipos primitivos están limitados a los tipos definidos en la tabla que sigue para simplificar el soporte para enlaces SOAP 1.1. Todos los tipos se pueden declarar como matrices, por ejemplo, **Cadena** []. De forma inherente, los tipos de datos binarios como, por ejemplo, **long**, se pueden representar como cadenas mediante un enlace de protocolo (por ejemplo, SOAP). Sin embargo, esta representación no tiene ninguna repercusión en la semántica del tipo, los valores permitidos, etc. tal como lo ve el cliente.

Tabla 4. Tipos primitivos

Tipo de API	Descripción	Tipo SOAP	Tipo Java.
Boolean	Valor booleano: true o false	xsd:Boolean	Boolean
dateTime	Un valor de fecha y hora	xsd:datetime	Date
decimal	Un valor decimal de precisión arbitraria	xsd:decimal	java.math.BigDecimal
double	Un valor decimal firmado de doble precisión	xsd:double	double
int	Un valor entero de 32 bits firmado	xsd:int	int
integer	Un valor entero firmado de precisión arbitraria	xsd:integer	java.math.BigInteger
long	Un valor entero de 64 bits firmado	xsd:long	long
string	Una cadena de caracteres Unicode	xsd:string	java.lang.String

MessageTypeEnum

```
MessageTypeEnum: { INFORMATION, WARNING, ERROR }
```

MessageTypeEnum es un tipo enumerado que define todos los tipos de mensaje posibles.

- INFORMATION: un mensaje informativo
- WARNING: un mensaje de advertencia
- ERROR: un mensaje de error

Mensaje

```
Mensaje: [tipo MessageTypeEnum, código de cadena, cadena localizedText, cadena logDetail]
```

El mensaje es una estructura de datos que define el resultado de una llamada a la API de servicio web. Proporciona campos opcionales para un código no localizado,

texto localizado y detalle de registro. Actualmente, todo el texto localizado utiliza el entorno local establecido para la instancia del servidor de IBM Marketing Operations.

Tabla 5. Parámetros de mensaje

Parámetro	Descripción
type	Un MessageTypeEnum, que define el tipo del mensaje.
code	Un código opcional, en formato de cadena, para el mensaje.
localizedText	Una cadena de texto opcional para asociar con el mensaje.
logDetail	Un mensaje de seguimiento de pila opcional.

NameValue

NameValue: [nombre cadena, secuencia int]

NameValue es un tipo complejo base que define un par nombre-valor. También define una secuencia opcional que utiliza el servicio para construir matrices de valores según sea necesario (las secuencias se basan en cero).

Todos los NameValues con el mismo nombre, pero distintos números de secuencia, se convierten a una matriz de valores y se asocian al nombre común.

El tamaño de matriz se determina mediante el número de secuencia máximo; los elementos de matriz sin especificar tienen valores nulos. Los números de secuencia de matriz deben ser exclusivos. El tipo ampliado proporciona el valor y su tipo.

Tabla 6. Parámetros NameValue

Parámetro	Descripción
name	Una cadena que define el nombre de un tipo NameValue.
secuencia	Un entero basado en cero que define el número de secuencia para el valor implícito deNameValue.

Los tipos ampliados NameValue se definen para cada tipo primitivo, del modo siguiente:

Tabla 7. Tipos NameValue ampliados

Tipo ampliado	Descripción
BigDecimalNameValue: NameValue [valor decimal]	Un tipo NameValue cuyo valor es un número decimal de precisión arbitraria.
BigIntegerNameValue: NameValue [valor entero]	Un tipo NameValue cuyo valor es un entero de un tamaño arbitrario.
BooleanNameValue: NameValue [valor booleano]	Un tipo NameValue cuyo valor es un booleano.
CurrencyNameValue: NameValue [valor decimal de entorno local de cadena]	Un tipo NameValue apto para representar la moneda en algún entorno local. El entorno local es un código de idioma ISO, es decir, los códigos de dos letras minúsculas tal como define ISO-639. Actualmente, el entorno local debe coincidir con el entorno local establecido en la instancia del servidor IBM Marketing Operations.

Tabla 7. Tipos NameValue ampliados (continuación)

Tipo ampliado	Descripción
DateNameValue: NameValue [valor de fecha y hora]	Un tipo NameValue cuyo valor es una fecha.
DecimalNameValue: NameValue [valor doble]	Un tipo NameValue cuyo valor es un número decimal de doble precisión.
IntegerNameValue: NameValue [valor largo]	Un tipo NameValue cuyo valor es un entero de 64 bits.
String NameValue: NameValue [valor de cadena]	Un tipo NameValue cuyo valor es una cadena.

Finalmente, se define una matriz de tipos NameValue ampliados para su uso cuando se debe definir un conjunto de NameValues de distintos tipos.

```

    NameValueArrays: [
BooleanNameValue[]    booleanValues,
StringNameValue[]     stringValues,
IntegerNameValue[]    integerValues,
BigIntegerNameValue[] bigIntegoooleanNameValue,
DecimalNameValue[]    decimalValues,
BigDecimalNameValue[] bigDecimalValues
DateNameValue[]       dateNameValues
CurrencyNameValue[]   currencyValues
    ]

```

Capítulo 3. Procedimientos de IBM Marketing Operations

Un "procedimiento" es una clase Java estándar o personalizada alojada por IBM Marketing Operations que realiza alguna unidad de trabajo. Los procedimientos proporcionan una manera para que los clientes y servicios profesionales de IBM amplíen la lógica empresarial de Marketing Operations de formas arbitrarias.

Los procedimientos siguen un modelo de programación sencillo con una API bien definida para afectar a los componentes gestionados por Marketing Operations. Los procedimientos se "descubren" a través de un mecanismo de búsqueda sencilla y un archivo de definiciones basado en XML. Marketing Operations ejecuta los procedimientos según las necesidades de sus "clientes". Por ejemplo, en respuesta a una solicitud de integración (de entrada) o la activación de un desencadenante (interna o de salida).

Los procedimientos se ejecutan de forma síncrona con su cliente; los resultados pasan a estar disponibles directamente al cliente y mediante un mecanismo de auditoría persistente. La ejecución de un procedimiento también puede causar que otros eventos y desencadenantes se activen en Marketing Operations.

Los procedimientos se deben grabar en Java.

Se presupone lo siguiente

Las clases de implementación del procedimiento se empaquetan en un árbol de clases independiente o un archivo JAR y se ponen a disposición general en IBM Marketing Operations a través de una ruta de URL.

Implementación de procedimientos

El gestor de ejecución del procedimiento utiliza un cargador de clases independiente para cargar estas clases, según sea necesario. De forma predeterminada, Marketing Operations busca en el directorio siguiente.

```
<Inicio_MarketingOperations>/devkits/integration/examples/classes
```

Para cambiar este valor predeterminado, establezca el parámetro **integrationProcedureClasspathURL** bajo **Valores > Configuración > Marketing Operations > umoConfiguration > integrationServices**.

El nombre de clase de implementación del procedimiento sigue los convenios de denominación de Java aceptados para evitar conflictos con "unica" y clases de otros proveedores. En concreto, los clientes no deben colocar procedimientos en el árbol de paquetes "com.unica" o "com.unicacorp".

La implementación del procedimiento se codifica en la versión de Java Runtime utilizada por IBM Marketing Operations en el servidor de aplicaciones (como mínimo, JRE 1.5.10).

La clase de implementación del procedimiento se carga mediante la política de carga de clases utilizada normalmente por IBM Marketing Operations (por regla general, **padre-último**). El servidor de aplicaciones puede proporcionar opciones y herramientas de desarrollo para volver a cargar las clases que se aplicarán a los

procedimientos de Marketing Operations, pero esto no es necesario.

Bibliotecas

IBM Marketing Operations proporciona algunas bibliotecas de código abierto y de terceros; los servidores de aplicaciones también utilizan distintas versiones de estas bibliotecas.

Por regla general, esta lista cambia de release a release. Para la versión 9.1.0 se da soporte a las siguientes bibliotecas de terceros.

- Ant 1.6.5 (ant.jar)
- Axis2 1.5.2 y dependencias
 - axiom-api-1.2.9.jar
 - axiom-impl-1.2.9.jar
 - axis2-adb-codegen-1.5.2.jar
 - axis2-codegen-1.5.2.jar
 - axis2-adb-1.5.2.jar
 - axis2-kernel-1.5.2.jar
 - axis2-transport-http-1.5.2.jar
 - axis2-transport-local-1.5.2.jar
 - commons-codec.jar
 - commons-httpclient-3.1.jar
 - commons-logging.jar
 - httpcore-4.0.jar
 - neethi-2.0.4.jar
 - geronimo-stax-api_1.0_spec-1.0.1.jar
 - jaxrpc.jar
 - xlsxScanner.jar
 - xlsxScannerUtils.jar
 - xlsxWASParasers.jar
 - wsdl4j-1.6.2.jar
 - XmlSchema-1.4.3.jar
- JavaMail 1.4.3 (activation.jar, mail.jar)
- JUnit 4.4 (junit-4.4.jar)
- Las API de IBM Marketing Operations (affinium_plan.jar)
- Las API de IBM Marketing Platform (unica-common.jar)

Si un procedimiento, o las clases secundarias importadas por el procedimiento, utilizan estos paquetes, su uso debe coincidir exactamente con los paquetes proporcionados por Marketing Operations o el servidor de aplicaciones. En este caso, deberá volver a trabajar en el código del procedimiento si una versión posterior de Marketing Operations actualiza o abandona una biblioteca.

Procedimientos y subprocessos

El procedimiento debe tener los subprocessos protegidos con respecto a su propio estado; es decir, su método de ejecución no puede depender de los cambios de estado interno entre llamadas. Un procedimiento no puede crear subprocessos por su cuenta.

Parámetros de configuración

Cuando se instala el módulo de integración de Marketing Operations, el instalador establece tres propiedades de configuración. Puede modificar las propiedades de configuración para personalizar el comportamiento del módulo de integración.

Las propiedades de configuración del módulo de integración se encuentran en **Marketing Operations | umoConfiguration | integrationServices**.

- La propiedad de configuración **enableIntegrationServices** activa y desactiva el módulo de Servicios de integración.
- El parámetro **integrationProcedureDefinitionPath** contiene la ruta de archivo completa del archivo XML de definiciones de procedimiento personalizado.
El valor predeterminado es `<Inicio_IBM_EMM><MarketingOperations_Home>/devkits/integration/examples/src/procedure/procedure-plugins.xml/`.
- El parámetro **integrationProcedureClasspathURL** contiene el URL de la ruta de clases de los procedimientos personalizados.
El valor predeterminado es `file:/// <Inicio_IBM_EMM><MarketingOperations_Home>/devkits/integration/examples/classes/`.

Nota: La '/' al final de la ruta `integrationProcedureClasspathURL` es necesaria para cargar correctamente las clases de procedimiento.

Diseño

La clase de implementación del procedimiento utiliza la API de IBM Marketing Operations para leer y actualizar los componentes de Marketing Operations, invocar servicios, etc. Otros paquetes de Java se pueden utilizar para realizar otras tareas.

El diseño se debe centrar en la producción de una sola unidad de trabajo que funcione de forma atómica. De forma ideal, un procedimiento realiza algunas series de tareas que se pueden planificar para ejecutarse de forma asíncrona más adelante. Este modelo de integración de "activar y olvidar" genera la mínima carga en ambos sistemas.

Nota: Sólo las clases y los métodos documentados estarán soportados en futuros releases de Marketing Operations. Considere que todas las demás clases y métodos en Marketing Operations se encuentran fuera de los límites.

Después de codificar y compilar las clases de implementación de procedimiento, debe ponerlas a disposición de Marketing Operations. Los scripts de creación proporcionados con los servicios de integración de Marketing Operations colocan los procedimientos compilados en la ubicación predeterminada. El paso final de desarrollo es actualizar el archivo de definiciones del plugin del procedimiento personalizado utilizado por Marketing Operations para descubrir los procedimientos personalizados.

El procedimiento debe implementar la interfaz **com.unica.publicapi.plan.plugin.procedure.IProcedure** y tener un constructor sin parámetros (modelo JavaBean usual). La codificación y compilación de cada procedimiento se realiza en un IDE Java de elección del cliente como, por ejemplo, Eclipse, Borland JBuilder o Idea. El código de muestra se proporciona con IBM Marketing Operations como kits de herramientas de desarrollador en la siguiente ubicación:

Ciclo de vida de procedimiento

Cada procedimiento se ejecuta durante un ciclo de vida completo.

El ciclo de vida de tiempo de ejecución de un procedimiento incluye los pasos siguientes:

1. Descubrimiento e inicialización
2. Selección (opcional)
3. Ejecución
4. Destrucción

Descubrimiento e inicialización

IBM Marketing Operations deben conocer todos los procedimientos personalizados y estándar disponibles para una instancia de instalación particular. Este proceso se llama descubrimiento.

Nota: Los procedimientos estándar (procedimientos definidos por el equipo de ingeniería de Marketing Operations) son conocidos de forma implícita y, por lo tanto, no necesitan ninguna acción para ser descubiertos.

Los procedimientos personalizados están definidos en el archivo de definición del plugin de procedimiento. El gestor de plugin de Marketing Operations lee este archivo durante la inicialización. Para cada procedimiento encontrado, el gestor de plugin realiza los pasos siguientes:

1. Crea una instancia del procedimiento; se cambia a su estado a INSTANTIATED.
2. Crea un registro de auditoría de procedimiento.
3. Si se ha creado una instancia del procedimiento, se llama a su método **initialize()** con los parámetros de inicialización que se encuentran en su archivo de descripción de plugin. Si este método lanza una excepción, se registra el estado y se abandona el procedimiento. De lo contrario, el estado de procedimiento cambia al estado INITIALIZED. Ahora está listo para su ejecución.
4. Crea un registro de auditoría de procedimiento.
5. Si se ha creado una instancia del procedimiento, se llama a su método **getKey()** para determinar la clave que utilizan los clientes para hacer referencia al procedimiento. Esta clave se asocia a la instancia y se guarda para una búsqueda posterior.

Selección

De vez en cuando, es posible que IBM Marketing Operations presente una lista de procedimientos disponibles a los usuarios, por ejemplo, para permitir que los usuarios configuren un desencadenante. Marketing Operations solo presenta esta lista una vez que se ha inicializado el procedimiento, utilizando los métodos **getDisplayName()** y **getDescription()** del procedimiento.

Ejecución

En algún momento después de que se haya inicializado el procedimiento, IBM Marketing Operations recibirá una solicitud para ejecutar el procedimiento. Esta

solicitud podría producirse simultáneamente con otros procedimientos (o el mismo procedimiento) en ejecución en otras hebras.

Durante la ejecución, el gestor de ejecución de procedimiento realiza los pasos siguientes.

1. Inicia una transacción de base de datos.
2. Establece el estado del procedimiento en EXECUTING.
3. Crea un registro de auditoría de procedimiento.
4. Llama al método **execute()** del procedimiento con un contexto de ejecución y los parámetros de ejecución proporcionados por el cliente. La implementación del método utiliza la API de Marketing Operations según sea necesario, adquiriendo bloqueos de edición y pasando el contexto de ejecución. Si el método de ejecución lanza una excepción, el gestor de ejecución marca la transacción para la retroacción.
5. Confirma o retrotrae la transacción de acuerdo con los resultados de ejecución; establece el estado del procedimiento en EXECUTED.
6. Libera los bloqueos de edición pendientes.
7. Crea un registro de auditoría de procedimiento.

Nota: El método **execute()** no debe modificar los datos de la instancia del procedimiento.

Destrucción

Cuando IBM Marketing Operations concluye, el gestor de plugin de procedimiento pasa por todos los procedimientos cargados. Para cada procedimiento encontrado, completa los pasos siguientes.

1. Llama al método **destroy()** del procedimiento para permitir al procedimiento limpiar antes de que se destruya la instancia.
2. Cambia el estado del procedimiento a FINALIZED (no se puede ejecutar).
3. Crea un registro de auditoría de procedimiento.
4. Destruye la instancia del procedimiento.

Clases Java clave

El kit del desarrollador de integración proporcionado contiene un conjunto de Javadoc de la API pública de IBM Marketing Operations y las clases soportadas.

A continuación se listan las clases Java más importantes.

- **IProcedure** (com.unica.publicapi.plan.plugin.procedure.IProcedure): interfaz que todos los procedimientos deben implementar. Los procedimientos pasan a través de un ciclo de vida bien definido y acceden a la API de Marketing Operations para realizar el trabajo.
- **ITriggerProcedure** (com.unica.publicapi.plan.plugin.procedure.ITriggerProcedure): interfaz que todos los procedimientos de desencadenante deben implementar (interfaz de marcador).
- **IExecutionContext** (com.unica.publicapi.plan.plugin.procedure.IExecutionContext): interfaz de objeto de contexto opaco transferido al procedimiento mediante el gestor de ejecución. Este objeto tiene métodos públicos para registrar y editar la gestión de bloqueos. El procedimiento también pasa este objeto a todas las llamadas a PlanAPI.

- IPlanAPI (com.unica.publicapi.plan.api.IPlanAPI): interfaz en la API de Marketing Operations. El contexto de ejecución proporciona un método **getPlanAPI()** para recuperar la implementación correcta.

Bloqueo de datos

IBM Marketing Operations utiliza un esquema de bloqueo de edición pesimista; es decir, sólo un usuario está autorizado para un acceso de actualización a instancias de componente a la vez. Para el usuario de la GUI, este bloqueo se realiza en el nivel de pestaña visual. En algunos casos, los datos se bloquean para un subconjunto de una instancia, por ejemplo, la pestaña Resumen de un proyecto. En otros casos, los datos se bloquean en varias instancias, por ejemplo, la pestaña Flujo de trabajo. Una vez que un usuario adquiere un bloqueo, todos los demás usuarios estarán restringidos a acceso de sólo lectura a los datos relacionados.

Para garantizar que otro usuario no sobrescriba accidentalmente los cambios que ha realizado un procedimiento en una instancia de componente o un grupo de instancias, un procedimiento debe adquirir los bloqueos apropiados antes de actualizar los datos de componente. Se utiliza el objeto de contexto de ejecución que se pasa al método **execute()** del procedimiento para realizar el bloqueo de los datos.

Antes de que el procedimiento actualice ningún dato, debe llamar al método **acquireLock()** del contexto para cada bloqueo que necesite. Por ejemplo, si un procedimiento va a actualizar un proyecto y el flujo de trabajo asociado, el procedimiento debe adquirir bloqueos para ambos.

Si otro usuario ya tiene un bloqueo, el método **acquireLock()** lanza una excepción **LockInUseException** inmediatamente. Para minimizar las colisiones, el procedimiento debe liberar el bloqueo tan pronto como actualice el objeto.

El gestor de ejecución libera automáticamente los bloqueos pendientes cuando vuelve el método de ejecución. En cualquier caso, los bloqueos sólo se conservan durante la vida de la transacción de base de datos. Es decir, los bloqueos caducan si se excede el tiempo de espera de la transacción específica de la base de datos.

Nota: Los bloqueos de edición no son lo mismo que las transacciones de base de datos.

Transacciones de procedimiento

El gestor de ejecución de procedimiento envuelve automáticamente la ejecución de un procedimiento con una transacción de base de datos, confirmándola o retrotrayéndola en función del resultado de la ejecución del procedimiento.

Envolver la ejecución del procedimiento y la transacción de base de datos garantiza que las actualizaciones en la base de datos de IBM Marketing Operations no serán visibles para otros usuarios hasta que se confirmen. También hace que las actualizaciones sean atómicas.

El grabador del procedimiento debe seguir adquiriendo los bloqueos de edición necesarios para asegurarse de que los otros usuarios no pueden grabar cambios en la base de datos antes de que se complete la ejecución del procedimiento.

Comunicación de procedimientos

El método **execute()** de un procedimiento devuelve un código de estado de entero a la tabla de auditoría de procedimiento de IBM Marketing Operations. El método **execute()** de un procedimiento puede devolver cero o más mensajes a la tabla de auditoría de procedimiento, que se registran y son persistentes.

El cliente podría comunicar también la información de estado de alguna otra forma.

Registro del procedimiento

IBM Marketing Operations tiene un archivo de registro aparte para procedimientos: `<Inicio_MarketingOperations>\logs\procedure.log`

El gestor de ejecución de procedimiento registra el ciclo de vida de cada procedimiento y crea registros de auditoría.

- **logInfo()**: graba un mensaje informativo en el registro de procedimiento.
- **logWarning()**: graba un mensaje de aviso en el registro de procedimiento.
- **logError()**: graba un mensaje de error en el registro de procedimiento.
- **logException()**: vuelca el seguimiento de pila para la excepción en el registro de procedimiento.

Archivo de definiciones del plugin de procedimiento

El archivo de definiciones del plugin de procedimiento define la clase de implementación, los metadatos y demás información sobre los procedimientos personalizados que se van a alojar en IBM Marketing Operations.

De forma predeterminada, se da por sentado que la definición del plugin del procedimiento está en la siguiente ruta:

```
<Inicio_MarketingOperations>/devkits/integration/examples/src/procedures/  
procedure-plugins.xml
```

Este archivo es un documento XML que contiene la siguiente información.

Procedimientos: una lista de cero o más elementos **Procedure**.

Procedimiento: un elemento que define un procedimiento. Cada procedimiento contiene los siguientes elementos.

- **key** (opcional): una cadena que define la clave de búsqueda para el procedimiento. Esta clave debe ser exclusiva con respecto a todos los procedimientos personalizados y estándar (proporcionados por IBM) alojados por una instancia de Marketing Operations determinada. Si no se ha definido, adopta de forma predeterminada el valor de la versión completa del elemento **className**. Los nombres que empiezan con la cadena "uap" se reservan para ser utilizados por IBM Marketing Operations.
- **className** (obligatorio): nombre de paquete completo de la clase del procedimiento. Esta clase debe implementar la clase `IProcedure` (`com.unica.public.plan.plugin.procedure.IProcedure`).
- **initParameters** (opcional): una lista de cero o más elementos `initParameter`.

initParameter(opcional): parámetro que debe transferirse al método de inicializar() procedimiento. Este elemento incluye el nombre de parámetro anidado, el tipo y los elementos de valor.

- name: cadena que define el nombre del parámetro
- type: nombre de clase opcional de la clase de envoltorio Java que define el tipo del valor de parámetro. Debe adoptar uno de los siguientes tipos:
 - java.lang.String (el valor predeterminado)
 - java.lang.Integer
 - java.lang.Double
 - java.lang.Calendar
 - java.lang.Boolean
- valor: forma de cadena del valor del atributo de acuerdo con su tipo

Capítulo 4. API de IBM Marketing Operations

La API de IBM Marketing Operations es una fachada que proporciona una vista de cliente de una instancia de Marketing Operations en ejecución.

Sólo se muestra a los usuarios un subconjunto de las prestaciones de Marketing Operations. La API la utilizan simultáneamente los usuarios web de Marketing Operations y las solicitudes SOAP y los desencadenantes del servicio web de Servicios de integración de Marketing Operations. La API soporta los siguientes tipos de operaciones.

- Creación y supresión de componente
- Descubrimiento (por tipo de componente, valor de atributo, etc.)
- Inspección de componente (a través de sus atributos, enlaces especializados, etc.)
- Modificación de componente

Contenido de la API de IBM Marketing Operations

El paquete `com.unica.publicapi.plan.api` suministra la API de IBM Marketing Operations.

Este paquete ofrece interfaces y excepciones, y contiene los siguientes tipos de clases:

- Tipos de datos enumerados.
- Manejadores para identificar instancias de componentes y objetos.
- Una correlación Java, `AttributeMap`.

Hay disponible documentación completa sobre la API, incluidos todos los métodos y valores posibles, pulsando **Ayuda > Documentación del producto** en una instancia de Marketing Operations y, a continuación, descargando el archivo IBM `<versión>PublicAPI.zip`.

Interfaces de API

La API (interfaz de programación de aplicaciones) de IBM Marketing Operations incluye **IPlanAPI** e **IExecutionContext**.

La API de Marketing Operations incluye las interfaces siguientes.

IPlanAPI

Define la API pública para Marketing Operations. Proporciona métodos para crear, descubrir y modificar objetos, incluidos carpetas, proyectos, programas, tareas de flujo de trabajo y miembros de equipo.

Para los sistemas que tienen habilitada la integración opcional con IBM Campaign, también proporciona métodos para crear, descubrir y modificar ofertas.

IExecutionContext

Define los desencadenantes y bloqueos que ejecutan métodos en la API.

Método de API

Si desea información específica sobre los métodos de API públicos, consulte la clase `iPlanAPI` en los archivos de documentación de API JavaDocs.

Estos archivos están disponibles iniciando sesión en Marketing Operations y seleccionando **Ayuda y documentación del producto** desde cualquier página y descargando el archivo `<version>PublicAPI.zip`.

Excepciones comunes

Las excepciones comunes que emite la API incluyen `NotFoundException`, `AuthorizationException`, `DataException`, `InvalidExecutionContextException` y `NotLockedException`.

La lista siguiente explica por qué se podrían producir estas excepciones comunes.

- *<tipo objeto>* `NotFoundException`: El sistema no puede devolver el elemento u objeto especificado.
- `AuthorizationException`: El usuario asociado al contexto de ejecución no está autorizado para realizar la operación solicitada. Esta excepción puede ser generada por cualquier método de API, de modo que no se declara.
- `DataException`: Se ha producido una excepción en la capa de base de datos subyacente en IBM Marketing Operations. Consulte el registro SQL para ver detalles.
- `InvalidExecutionContextException`: Hay un problema con un contexto de ejecución pasado a un método de API (por ejemplo, el método no se inicializó correctamente). Esta excepción puede ser generada por cualquier API, de modo que no se declara.
- `NotLockedException`: Se ha intentado actualizar los datos de componente sin adquirir primero el bloqueo necesario. Consulte el método `acquireLock()` de la interfaz `IExecutionContext`.

Manejadores

Un Manejador es un objeto de URL especial que hace referencia a una instancia de objeto concreta en una instancia de IBM Marketing Operations. Los Manejadores incluyen el tipo de componente, el identificador de datos internos y un URL base de instancia.

Los manejadores utilizados o generados por la API pueden externalizarse en un URL completo. Puede utilizar el URL resultante de distintas formas. Puede utilizar el URL para abrir una vista del componente en la GUI de Marketing Operations, enviar mensajes de correo electrónico o utilizar en otro procedimiento como un parámetro.

Los Manejadores sólo son válidos para una instancia de servicio de Marketing Operations concreta, o una instancia en clúster, pero son válidos durante todo el ciclo de vida del servicio desplegado. Como resultado, los Manejadores se pueden guardar en un archivo para una referencia posterior, pero no se pueden utilizar para acceder a componentes en otra instancia de Marketing Operations. Esta restricción también se aplica a instancias en el mismo servidor de host físico. Sin embargo, Marketing Operations proporciona un mecanismo para correlacionar distintos URL base con la instancia actual para dar cabida a la reubicación de una instancia en otro servidor (por ejemplo, en el caso de que el equipo no funcione correctamente).

Los Manejadores dependen del cliente. Por ejemplo, un desencadenante puede pasar un Manejador a un procedimiento, que lo utiliza como parámetro en una llamada SOAP a un sistema de otro proveedor. A continuación, este proveedor puede emitir una solicitud SOAP de vuelta a Marketing Operations para iniciar un procedimiento que actualice un atributo.

Los miembros de la clase de Manejador tienen métodos de fábrica para crear manejadores a partir de distintos tipos de URL. A continuación se muestran algunos ejemplos.

Aprobación

`http://mymachine:7001/plan/affiniumplan.jsp?cat=approvaldetail&approvalid=101`

Activo

`http://mymachine:7001/plan/affiniumplan.jsp?cat=asset&assetMode=VIEW_ASSET&assetid=101`

Carpeta de activos

`http://mymachine:7001/plan/affiniumplan.jsp?cat=folder&id=101`

Biblioteca de activos

`http://mymachine:7001/plan/affiniumplan.jsp?cat=library&id=101`

Archivo adjunto

`http://mymachine:7001/plan/affiniumplan.jsp?cat=attachmentview&attachid=101&parentObjectId=101&parentObjectType=project`

Cuenta financiera

`http://mymachine:7001/plan/affiniumplan.jsp?cat=accountdetails&accountid=101`

Carpeta

`http://mymachine:7001/plan/affiniumplan.jsp?cat=grouping_folder&folderid=1234`

Factura

`http://mymachine:7001/plan/affiniumplan.jsp?cat=invoicedetails&invoiceid=134`

Elemento de línea de factura

`http://mymachine:7001/plan/affiniumplan.jsp?cat=invoicedetails&invoiceid=134&line_item_id=101`

Objeto de marketing

`http://mymachine:7001/plan/affiniumplan.jsp?cat=componenttabs&componentid=creatives&componentinstid=1234`

Cuadrícula de objeto de marketing

`http://mymachine:7001/plan/affiniumplan.jsp?cat=componenttabs&componentid=creatives&componentinstid=1234&gridid=grid`

Fila de cuadrícula de objeto de marketing

`http://mymachine:7001/plan/affiniumplan.jsp?cat=componenttabs&componentid=creatives&componentinstid=1234&gridid=grid&gridrowid=101`

Equipo de plan

`http://mymachine:7001/plan/affiniumplan.jsp?cat=teamdetails&func=edit&teamid=100001`

Usuario de plan

`http://mymachine:7001/plan/affiniumplan.jsp?cat=adminuserpermissions&func=edit&userid=101`

Programa

`http://mymachine:7001/plan/affiniumplan.jsp?cat=programtabs&programid=125`

Cuadrícula de programa

`http://mymachine:7001/plan/affiniumplan.jsp?cat=programtabs&programid=1234&gridid=grid`

Fila de cuadrícula de programa

`http://mymachine:7001/plan/affiniumplan.jsp?cat=programtabs&programid=1234&gridid=grid&gridrowid=101`

Proyecto

```
http://mymachine:7001/plan/affiniumplan.jsp?cat=projecttabs&
projectid=1234
```

Cuadrícula de proyecto

```
http://mymachine:7001/plan/affiniumplan.jsp?cat=projecttabs&
projectid=1234&gridid=grid
```

Fila de cuadrícula de proyecto

```
http://mymachine:7001/plan/affiniumplan.jsp?cat=projecttabs&
projectid=1234&gridid=grid&gridrowid=101
```

Elemento de línea de proyecto

```
http://mymachine:7001/plan/affiniumplan.jsp?cat=projecttabs&
projectid=1234&projectlineitemid=123&projectlineitemisversionfinal=false
```

Etapas de flujo de trabajo

```
http://mymachine:7001/plan/affiniumplan.jsp?cat=projectworkflow&
projectid=1234&taskid=5678
```

Tarea de flujo de trabajo

```
http://mymachine:7001/plan/affiniumplan.jsp?cat=projectworkflow&
projectid=1234&taskid=5678
```

Correlación de atributos

La clase `AttributeMap` es una correlación Java que solo contiene atributos. El atributo `<Nombre>` es la clave de entrada de correlación y el atributo de matriz de `<valores>` (tener en cuenta que es plural) es el valor de entrada de la correlación.

La clase `AttributeMap` incluye los campos siguientes.

- `<Nombre>`: el nombre programático del atributo. Este nombre sirve como una clave exclusiva para acceder al atributo dentro de la instancia de componente en la que se produce.

Nota: El `<Nombre>` no es necesariamente el nombre de visualización presentado al usuario en la GUI. Para los componentes creados a partir de plantillas (como, por ejemplo, proyectos o tareas de flujo de trabajo), el nombre de atributo se especifica mediante la definición del elemento de plantilla. El nombre de atributo debe ser exclusivo. Para otros componentes, el nombre del atributo normalmente se suele derivar mediante programa de la instancia del componente del lado del servidor (por ejemplo, a través de una introspección Java).

Nota: Por convención, los atributos personalizados incluyen el nombre del formulario en el cual está definida la versión editable:

`<nombre_formulario>.<nombre_atributo>`.

- **Valores:** una matriz de objetos Java que contiene cero o más valores de atributo. El tipo de cada valor debe ser el mismo y estar conforme al tipo de atributo tal como está definido en Marketing Operations. Sólo están soportados el envoltorio Java y los tipos Marketing Operations:
 - `AssetLibraryStateEnum`: un valor de tipo enumerado `AssetLibraryStateEnum`.
 - `AssetStateEnum`: un valor de tipo enumerado `AssetStateEnum`.
 - `AttachmentTypeEnum`: un valor de tipo enumerado `AttachmentTypeEnum`.
 - `AttributeMap`: una correlación que contiene atributos.
 - `BudgetPeriodEnum`: un valor de tipo enumerado `BudgetPeriodEnum`.
 - `BudgetTypeEnum`: un valor de tipo enumerado `BudgetTypeEnum`.

- Handle: una referencia a una instancia de componente, fila de cuadrícula, atributo, etc.
- InvoiceStateEnum: un valor de tipo enumerado InvoiceStateEnum.
- java.io.File: representación de un archivo.
- java.lang.Boolean: un valor booleano, ya sea True o False
- java.lang.Double: un valor de número decimal de doble precisión.
- java.lang.Float: un valor de número decimal de precisión única
- java.lang.Integer: un valor entero de 32 bits
- java.lang.Long: un valor entero de 64 bits
- java.lang.Object: objeto Java genérico
- java.lang.String: una cadena de cero o más caracteres Unicode
- java.math.BigDecimal: valor de número decimal firmado de precisión arbitraria. Apto para monedas; la interpretación del valor depende del entorno local de moneda del cliente.
- java.math.BigInteger: valor entero de precisión arbitraria.
- java.net.URL: un objeto URL (Universal Resource Locator).
- java.util.ArrayList: lista de objetos.
- java.util.Calendar: una valor de fecha y hora para un entorno local particular.
- java.util.Date: un valor de fecha y hora. Este tipo está en desuso. Utilice en su lugar java.util.Calendar o java.util.GregorianCalendar.

Nota: Para implementar la fecha, los usuarios puede seleccionar java.util.Calendar o java.util.GregorianCalendar.

- java.util.GregorianCalendar: GregorianCalendar es una subclase concreta de java.util.Calendar y proporciona el sistema de calendario estándar más utilizado en el mundo.
- MonthEnum: un valor de tipo enumerado MonthEnum.
- ProjectStateEnum: un valor de tipo enumerado ProjectStateEnum.
- QuarterEnum: un valor de tipo enumerado QuarterEnum.
- TaskStateEnum: un valor de tipo enumerado TaskStateEnum.
- WeekEnum: un valor de tipo enumerado WeekEnum.

Los metadatos de un atributo (por ejemplo, la descripción y el nombre de visualización traducidos) se definen mediante la plantilla asociada al atributo y su instancia de objeto padre. Los atributos proporcionan un mecanismo sencillo pero ampliable para mostrar ambos atributos de instancia de objeto, los obligatorios y los opcionales como, por ejemplo, nombre de proyecto, código y fecha de inicio.

Tipos de datos enumerados

La API de IBM Marketing Operations soporta los siguientes tipos de datos y valores.

ApprovalMethodEnum

ApprovalMethodEnum define métodos válidos de aprobación. Los valores posibles son:

- SEQUENTIAL
- SIMULTANEOUS

ApprovalStateEnum

ApprovalStateEnum define estados válidos de aprobación. Los valores posibles son:

- CANCELLED
- COMPLETED
- IN_PROGRESS
- NOT_STATED
- ON_HOLD

AssetLibraryStateEnum

AssetLibraryStateEnum define estados válidos de bibliotecas de activos. Los valores posibles son:

- DISABLED
- ENABLED

AssetStateEnum

AssetStateEnum define estados válidos de activos. Los valores posibles son:

- ARCHIVE
- DRAFT
- FINALIZE
- LOCK

AttachmentTypeEnum

AttachmentTypeEnum define tipos válidos de adjuntos. Los valores posibles son:

- ASSET
- FILE
- URL

BudgetPeriodEnum

BudgetPeriodEnum define los periodos posibles de presupuesto. Los valores posibles son:

- ALL
- MONTHLY
- QUARTERLY
- WEEKLY
- YEARLY

BudgetTypeEnum

BudgetTypeEnum define tipos válidos de presupuestos. Los valores posibles son:

- ACTUAL
- ALLOCATED
- COMMITTED
- FORECAST
- TOTAL

ComponentTypeEnum

ComponentTypeEnum identifica los tipos de componente de Marketing Operations accesibles. Los valores posibles son:

- APPROVAL
- ASSET
- ASSET_FOLDER
- ASSET_LIBRARY
- ARCHIVO ADJUNTO

- FINANCIAL_ACCOUNT
- GROUPING_FOLDER
- INVOICE
- MARKETING_OBJECT
- PLAN_TEAM
- PLAN_USER
- PROGRAMA
- PROJECT
- PROJECT_REQUEST
- TASK
-

InvoiceStateEnum

InvoiceStateEnum define estados válidos de facturas. Los valores posibles son:

- CANCELLED
- DRAFT
- PAID
- PAYABLE

MonthEnum

MonthEnum define valores válidos para el mes.

OfferStateEnum

OfferStateEnum define estados válidos de ofertas. Los valores posibles son:

- STATE_OFFER_DRAFT
- STATE_OFFER_PUBLISHED
- STATE_OFFER_RETIRED

ProjectCopyTypeEnum

ProjectCopyTypeEnum define métodos válidos para copiar un proyecto. Los valores posibles son:

- COPY_USING_PROJECT_METRICS
- COPY_USING_TEMPLATES_METRICS

ProjectParticipantLevelEnum

ProjectParticipantLevelEnum identifica los roles que pueden tener los usuarios en un proyecto. Los valores posibles son:

- OWNER
- PARTICIPANT
- REQUESTER

ProjectStateEnum

ProjectStateEnum define estados válidos de proyectos y solicitudes. Los valores posibles son:

- ACCEPTED
- CANCELLED
- COMPLETED
- DRAFT
- IN_PROGRESS
- IN_RECONCILIATION
- LATE: el proyecto no se ha iniciado en la fecha de inicio planificada.

- NO_INICIADO
- ON_HOLD
- OVERDUE: el proyecto no se ha completado antes de la fecha de finalización planificada.
- RETURNED
- SUBMITTED

Para obtener más información sobre estados de proyectos y tareas, consulte la *Guía del usuario de IBM Marketing Operations*.

QuarterEnum

QuarterEnum define los valores válidos para trimestres: Q1, Q2, Q3 y Q4.

TaskStateEnum

TaskStateEnum define estados válidos de tareas de flujo de trabajo. Los valores posibles son:

- ACTIVE
- DISABLED
- FINISHED
- PENDING
- SKIPPED

WeekEnum

WeekEnum define valores válidos para las semanas de un año, de WEEK_1 a WEEK_53.

Antes de contactar con el servicio de soporte técnico de IBM

Si encuentra un problema que no puede resolver consultando la documentación, la persona responsable del contacto con el servicio de soporte técnico de la empresa puede registrar una llamada con el soporte técnico de IBM. Utilice estas directrices para asegurarse de que el problema se resuelva de forma eficaz y satisfactoria.

Si usted no es una de las personas responsables del contacto con el servicio de soporte técnico en su empresa, póngase en contacto con su administrador de IBM para obtener información.

Nota: El soporte técnico no escribe ni crea scripts de API. Para obtener ayuda en la implementación de nuestras ofertas de API, póngase en contacto con IBM Professional Services.

Información que se debe recopilar

Antes de ponerse en contacto con el soporte técnico de IBM, recopile la información siguiente:

- Una breve descripción del tipo de problema.
- Los mensajes de error detallados que aparecen cuando se produce el problema.
- Los pasos detallados que se deben seguir para reproducir el problema.
- Los archivos de registro, archivos de sesión, archivos de configuración y archivos de datos relacionados.
- Información sobre el producto y el entorno del sistema, que puede obtener tal como se describe en "Información del sistema".

Información del sistema

Cuando llame al soporte técnico de IBM, es posible que se le pida información sobre su entorno.

Si el problema no le impide conectarse, gran parte de esta información está disponible en la página Acerca de, que proporciona información sobre las aplicaciones IBM instaladas.

Puede acceder a la página Acerca de seleccionando **Ayuda > Acerca de**. Si la página Acerca de no es accesible, localice un archivo `version.txt` que se encuentra en el directorio de instalación de la aplicación.

Información de contacto del servicio de soporte técnico de IBM

Para conocer las formas de contacto con el soporte técnico de IBM, consulte el sitio web del soporte técnico del producto IBM: (http://www.ibm.com/support/entry/portal/open_service_request).

Nota: Para especificar una solicitud de soporte, debe iniciar sesión con una cuenta de IBM. Esta cuenta debe estar enlazada al número de cliente de IBM. Para obtener más información sobre cómo asociar la cuenta a su número de cliente de IBM, consulte **Support Resources>Entitled Software Support** en el portal de soporte.

Notices

This information was developed for products and services offered in the U.S.A.

IBM may not offer the products, services, or features discussed in this document in other countries. Consult your local IBM representative for information on the products and services currently available in your area. Any reference to an IBM product, program, or service is not intended to state or imply that only that IBM product, program, or service may be used. Any functionally equivalent product, program, or service that does not infringe any IBM intellectual property right may be used instead. However, it is the user's responsibility to evaluate and verify the operation of any non-IBM product, program, or service.

IBM may have patents or pending patent applications covering subject matter described in this document. The furnishing of this document does not grant you any license to these patents. You can send license inquiries, in writing, to:

IBM Director of Licensing
IBM Corporation
North Castle Drive
Armonk, NY 10504-1785
U.S.A.

For license inquiries regarding double-byte (DBCS) information, contact the IBM Intellectual Property Department in your country or send inquiries, in writing, to:

Intellectual Property Licensing
Legal and Intellectual Property Law
IBM Japan, Ltd.
19-21, Nihonbashi-Hakozakicho, Chuo-ku
Tokyo 103-8510, Japan

The following paragraph does not apply to the United Kingdom or any other country where such provisions are inconsistent with local law: INTERNATIONAL BUSINESS MACHINES CORPORATION PROVIDES THIS PUBLICATION "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF NON-INFRINGEMENT, MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. Some states do not allow disclaimer of express or implied warranties in certain transactions, therefore, this statement may not apply to you.

This information could include technical inaccuracies or typographical errors. Changes are periodically made to the information herein; these changes will be incorporated in new editions of the publication. IBM may make improvements and/or changes in the product(s) and/or the program(s) described in this publication at any time without notice.

Any references in this information to non-IBM Web sites are provided for convenience only and do not in any manner serve as an endorsement of those Web sites. The materials at those Web sites are not part of the materials for this IBM product and use of those Web sites is at your own risk.

IBM may use or distribute any of the information you supply in any way it believes appropriate without incurring any obligation to you.

Licensees of this program who wish to have information about it for the purpose of enabling: (i) the exchange of information between independently created programs and other programs (including this one) and (ii) the mutual use of the information which has been exchanged, should contact:

IBM Corporation
B1WA LKG1
550 King Street
Littleton, MA 01460-1250
U.S.A.

Such information may be available, subject to appropriate terms and conditions, including in some cases, payment of a fee.

The licensed program described in this document and all licensed material available for it are provided by IBM under terms of the IBM Customer Agreement, IBM International Program License Agreement or any equivalent agreement between us.

Any performance data contained herein was determined in a controlled environment. Therefore, the results obtained in other operating environments may vary significantly. Some measurements may have been made on development-level systems and there is no guarantee that these measurements will be the same on generally available systems. Furthermore, some measurements may have been estimated through extrapolation. Actual results may vary. Users of this document should verify the applicable data for their specific environment.

Information concerning non-IBM products was obtained from the suppliers of those products, their published announcements or other publicly available sources. IBM has not tested those products and cannot confirm the accuracy of performance, compatibility or any other claims related to non-IBM products. Questions on the capabilities of non-IBM products should be addressed to the suppliers of those products.

All statements regarding IBM's future direction or intent are subject to change or withdrawal without notice, and represent goals and objectives only.

All IBM prices shown are IBM's suggested retail prices, are current and are subject to change without notice. Dealer prices may vary.

This information contains examples of data and reports used in daily business operations. To illustrate them as completely as possible, the examples include the names of individuals, companies, brands, and products. All of these names are fictitious and any similarity to the names and addresses used by an actual business enterprise is entirely coincidental.

COPYRIGHT LICENSE:

This information contains sample application programs in source language, which illustrate programming techniques on various operating platforms. You may copy, modify, and distribute these sample programs in any form without payment to IBM, for the purposes of developing, using, marketing or distributing application programs conforming to the application programming interface for the operating

platform for which the sample programs are written. These examples have not been thoroughly tested under all conditions. IBM, therefore, cannot guarantee or imply reliability, serviceability, or function of these programs. The sample programs are provided "AS IS", without warranty of any kind. IBM shall not be liable for any damages arising out of your use of the sample programs.

If you are viewing this information softcopy, the photographs and color illustrations may not appear.

Trademarks

IBM, the IBM logo, and ibm.com are trademarks or registered trademarks of International Business Machines Corp., registered in many jurisdictions worldwide. Other product and service names might be trademarks of IBM or other companies. A current list of IBM trademarks is available on the Web at "Copyright and trademark information" at www.ibm.com/legal/copytrade.shtml.

Privacy Policy and Terms of Use Considerations

IBM Software products, including software as a service solutions, ("Software Offerings") may use cookies or other technologies to collect product usage information, to help improve the end user experience, to tailor interactions with the end user or for other purposes. A cookie is a piece of data that a web site can send to your browser, which may then be stored on your computer as a tag that identifies your computer. In many cases, no personal information is collected by these cookies. If a Software Offering you are using enables you to collect personal information through cookies and similar technologies, we inform you about the specifics below.

Depending upon the configurations deployed, this Software Offering may use session and persistent cookies that collect each user's user name, and other personal information for purposes of session management, enhanced user usability, or other usage tracking or functional purposes. These cookies can be disabled, but disabling them will also eliminate the functionality they enable.

Various jurisdictions regulate the collection of personal information through cookies and similar technologies. If the configurations deployed for this Software Offering provide you as customer the ability to collect personal information from end users via cookies and other technologies, you should seek your own legal advice about any laws applicable to such data collection, including any requirements for providing notice and consent where appropriate.

IBM requires that Clients (1) provide a clear and conspicuous link to Customer's website terms of use (e.g. privacy policy) which includes a link to IBM's and Client's data collection and use practices, (2) notify that cookies and clear gifs/web beacons are being placed on the visitor's computer by IBM on the Client's behalf along with an explanation of the purpose of such technology, and (3) to the extent required by law, obtain consent from website visitors prior to the placement of cookies and clear gifs/web beacons placed by Client or IBM on Client's behalf on website visitor's devices

For more information about the use of various technologies, including cookies, for these purposes, See IBM's Online Privacy Statement at: <http://www.ibm.com/privacy/details/us/en> section entitled "Cookies, Web Beacons and Other Technologies."



Impreso en España