Versión 10 Release 0
Junio de 2016

IBM Campaign Guía de la API SOAP

Nota

Antes de utilizar esta información y el producto al que da soporte, lea la información de “Avisos” en la página 63.

Esta edición se aplica a la versión 10, release 0, modificación 0 de IBM Campaign y a todos los releases y modificaciones subsiguientes hasta que no se indique lo contrario en nuevas ediciones.

© Copyright IBM Corporation 1998, 2017.

Contenido

Capítulo 1. Visión general de la API SOAP de servicios de IBM Campaign . . 1
Visión general del diseño de la API SOAP 1
Cambios de la API SOAP por versión 3
Implementaciones existentes utilizando el archivo jar de la API Client 4
Implementaciones existentes utilizando directamente el WSDL 5
Referencias de la API SOAP 7

Capítulo 2. Utilización de la API SOAP de IBM Campaign 9
Utilización del archivo .jar de la API de cliente para llamar a los servicios de Campaign 9
OfferAPI.java 9
Utilización del WSDL para llamar a los servicios de Campaign 11
Consideraciones sobre el rendimiento. 11

Capítulo 3. Tipos de datos de la API SOAP 13
WSReference 13
WSVersion. 13
WSServiceInfo 13
WSAttributeTypeEnum 14
WSAttributeStatusEnum 14
WSAccessTypeEnum 14
WSSelectTypeEnum. 14
WSRunStatusEnum 14
WSRunTypeEnum 15
WSAttribute 15
WSAttributeMetadata 16
WSCampaignInfo 18
WSComponentOrFolderInfo 18
WSTargetCellInfo 18
WSMetricsInfo 18
WSRunResults 19
WSOfferInfo 19
WSOfferCodeOrName 19
WSOfferValidationInfo. 20
WSOfferTemplateInfo 20
WSBulkOfferInfo 20
WSOfferInfoStatus 20

Capítulo 4. Métodos de la API SOAP . . 21
Métodos de la API SOAP: Servicio. 21
getServiceInfo 21
Métodos de la API SOAP: Atributos 21
getAttributesByName 21
updateAttributes 22
getAttributeMetadataByName 23
createAttributeMetadata 24
updateAttributeMetadata 25
deleteAttributeMetadata 26
Métodos de la API SOAP: Campañas y diagramas de flujo 27
generateCampaignCode 28
deleteCampaigns 29
createCampaign 29
listCampaignsByPage 30
stopFlowchart 31
Métodos de la API SOAP: celdas objetivo 31
createTargetCell 32
bulkCreateTargetCells 33
listTargetCells 34
bulkUpdateTargetCells. 35
getRunResultsByCell 36
bulkDeleteTargetCells 37
updateTemplateAttributes 37
listBottomUpTargetCells 38
Métodos de la API SOAP: Análisis 39
getCampaignMetrics 39
Métodos de la API SOAP: ofertas, listas de ofertas, plantillas de ofertas. 39
listOffersAndFolders 40
searchOffersBasic 41
listOffersByPage 42
createSmartOfferList 43
createStaticOfferList 44
getOffers 44
validateOffers 45
editOfferList 46
createOffer. 46
retireOffers 47
deleteOffers 48
deleteOffersAndLists 48
listOfferTemplates 49
createTemplate 49
getOfferTemplate 50
retireOfferTemplates 51
getOffersAndListsByPage 51
bulkCreateOffers 52
getOfferListDetails 52
getOfferListMembers 53
getOffersByQuery 53
retireOfferLists 54
createFolder 55
editFolder 55
getSubFoldersList 56
moveFolders 57
deleteFolders 57

© Copyright IBM Corp. 1998, 2017

Capítulo 5. Excepciones comunes de la API SOAP 59

Antes de ponerse en contacto con el equipo de soporte técnico de IBM . . . 61

Avisos 63
Marcas registradas 65
Consideraciones sobre la política de privacidad y los términos de uso 65

Capítulo 1. Visión general de la API SOAP de servicios de IBM Campaign

La especificación de la API SOAP de servicios de IBM® Campaign define la versión 3.0 de la interfaz de programación de aplicaciones, también conocida como Servicios de Campaign. Esta especificación se ofrece como parte de CampaignServices Software Developer's Tookit (devkits) instalado con IBM Campaign.

El directorio `<CAMPAIGN_HOME>/devkits/CampaignServicesAPI` que proporciona el instalador incluye ejemplos, scripts de construcción y de texto, Javadoc para clases e interfaces públicas y notas del release.

La API SOAP de servicios de Campaign se ha diseñado para lo siguiente:

- Proporcionar el acceso de creación, descubrimiento, lectura y actualización preciso para los componentes de IBM Campaign, a la vez que se aísla a los clientes de los detalles de implementación subyacente.
- Funcionar con la interfaz de usuario existente de IBM Campaign con efectos mínimos.
- Garantizar la validez de los datos.
- Satisfacer los servicios de seguridad necesarios de IBM Campaign.
- Soporte del estándar del sector SOAP (Simple Object Access Protocol), incluida la autenticación segura.

Visión general del diseño de la API SOAP

La API SOAP de Campaign Services es una fachada que proporciona una vista de cliente de una instancia de aplicación de IBM Campaign en ejecución. Solamente se expone un subconjunto de las prestaciones de IBM Campaign pero basta para manejar los aspectos clave de la funcionalidad de Campaign.

Características y diagrama

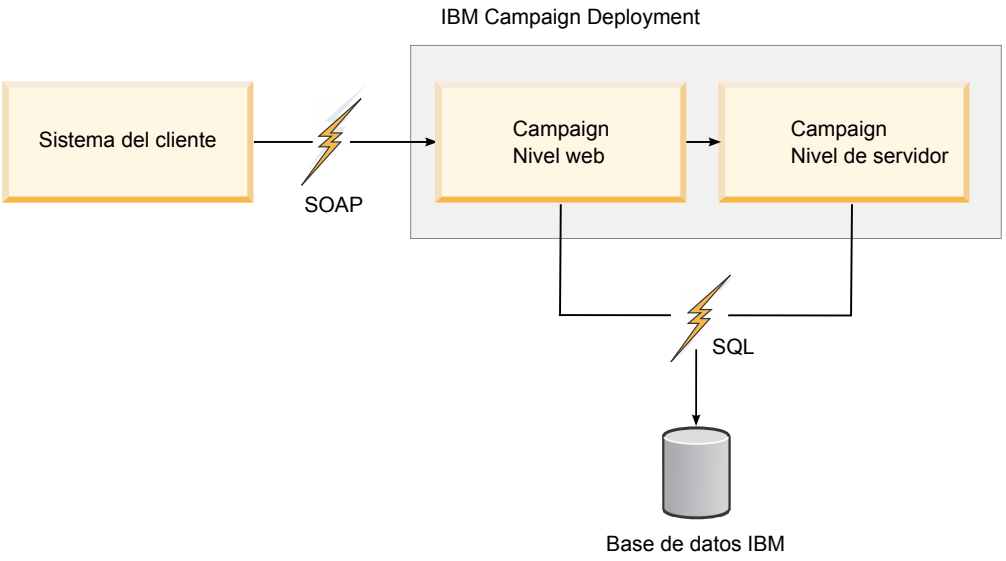
La API está diseñada para utilizarse simultáneamente con usuarios web de IBM Campaign y otros subprocesos de API.

Generalmente, la API soporta los siguientes tipos de operaciones en componentes de campañas, ofertas y celdas objetivo:

- Creación de componentes
- Descubrimiento de componentes
- Supresión de componentes
- Creación del atributo de componente y de los metadatos de atributos, inspección y modificación
- Captación de resultados de la ejecución de diagrama de flujo

El siguiente diagrama presenta una muestra del despliegue de CampaignServices 3.0.

© Copyright IBM Corp. 1998, 2017



Autentificación de usuario

La autenticación se encarga de establecer una identidad de usuario. La autenticación de usuario es responsabilidad de la aplicación cliente.

Autorización de usuario

La autorización se encarga de los permisos que un usuario autenticado tiene en relación a los componentes y las operaciones expuestos por la API.

Es posible que un usuario se autentique satisfactoriamente, pero no tenga suficientes permisos para realizar algunas operaciones, como por ejemplo editar la información de resumen de una campaña. En este caso, el método de la API emite `AuthorizationException`.

Entorno local

Las solicitudes de API proporcionan un parámetro opcional **`requestedLocale`**, que define el entorno local que se debe utilizar para llevar a cabo esa solicitud en particular. Si no se ha definido, la API toma de forma predeterminada el entorno local preferido del usuario de IBM . Se utiliza el algoritmo normal de máxima coincidencia Java™ para devolver mensajes y otros textos localizados en el entorno local solicitado.

Este parámetro es de tipo de clase `java.util.Locale`.

Nota: Cierto texto especificado por el usuario como, por ejemplo, las descripciones de campañas, se hallan en el entorno local del usuario que ha especificado el texto. IBM Campaign no intenta localizar este tipo de datos. La API solamente localiza mensajes de información, aviso y error.

Gestión de estado

La API CampaignServices es sin estado, lo que significa que la API no guarda información por cliente entre llamadas.

Obviamente, llamadas específicas a la API pueden cambiar el estado de instancias de componentes subyacentes gestionadas por Campaign, y estos cambios de estado pueden ser persistentes en la base de datos.

Cambios de la API SOAP por versión

La finalidad de este tema es identificar los cambios en la API SOAP de IBM Campaign para los clientes que están actualmente utilizando la API. Si ha actualizado desde una versión anterior, revise la siguiente información para determinar si necesita realizar cambios en el código de la aplicación.

Versiones y compatibilidad con versiones anteriores

Las versiones futuras de la API CampaignServices son compatibles con versiones anteriores, con todos los releases menores y de mantenimiento que comparten un mismo número de versión principal. Sin embargo, IBM se reserva el derecho de interrumpir la compatibilidad con la versión anterior para versiones importantes "punto cero" (x.0).

El número de versión principal de la API se incrementa si se realiza alguno de los cambios siguientes:

- Se cambia la interpretación de datos
- Se cambia la lógica empresarial, es decir, la funcionalidad de método de servicio
- Se cambian los parámetros de método y/o tipos de retorno

El número de versión menor de la API se incrementa si se realiza alguno de los cambios siguientes.

- Se añade un nuevo método
- Se añade un tipo de datos nuevo y se restringe su uso a nuevos métodos
- Se añade un nuevo tipo a un tipo enumerado
- Se define una nueva versión de una interfaz

IBM continúa dando soporte el cliente SOAP WSDL publicado y la versión de Apache Axis que se utiliza para implementar la oferta SOAP hasta al menos la próxima versión principal de IBM . En la práctica, esto se lleva a cabo dando soporte simultáneamente a varios servicios web específicos de versión. (IBM ya da soporte a varias versiones de este servicio internamente.)

Cambios en v9.1.2

No han habido cambios de la API en la versión 9.1.2.

Nota: En la versión v9.1.2 se ha introducido una API REST, además de la API SOAP descrita en esta guía. La API REST se describe por separado.

Cambios en v9.1.1

`stopFlowchart` incluye un parámetro de entrada nuevo, **`runId`**, para dar soporte a un entorno de escucha en clúster.

Cambios en v9.1

No hubo cambios de API en la versión 9.1.

Cambios en v9.0

No hubo cambios de API en la versión 9.0.

Cambios en v8.6

Cambios de la API implementados en v8.6:

- El motor SOAP se ha actualizado de la versión 1.4.1 a AXIS2 AXIS 1.5.2.
- El WSDL se ha reestructurado para abordar temas que manejan los parámetros necesarios y opcionales.
- El archivo `.jar` de la API de cliente ha cambiado debido a los cambios de WSDL. Por lo tanto, los apéndices y las clases generados han cambiado. Los parámetros del método de la API de cliente no han cambiado, pero los constructores de los objetos de valor a los que se da soporte se han modificado para el uso del conversor AXIS2 WSDL2Java.
- El URL del servicio web apunta a:

  ```
  http://<host>:<puerto>/Campaign/services/CampaignServices30Service
  ```

  y el WSDL correspondiente se recupera en:

  ```
  http://<host>:<puerto>/Campaign/services/CampaignServices30Service?wsdl
  ```

Si está actualizando a IBM Campaign versión 8.6 o superior y actualmente utiliza la API de servicios de Campaign, debe modificar el código de aplicación.

En función de si se utiliza la API de cliente o el WSDL, consulte las secciones siguientes para obtener detalles:

- "Implementaciones existentes utilizando el archivo jar de la API Client"
- "Implementaciones existentes utilizando directamente el WSDL" en la página 5

Implementaciones existentes utilizando el archivo jar de la API Client

Esta información se aplica si está realizando una actualización a la versión 8.6 o posterior de IBM Campaign y si utiliza actualmente el archivo `.jar` de la API Client para interactuar con la aplicación web de Campaign.

archivo .jar de la API Client

La aplicación Java debe utilizar el archivo `.jar` que se encuentra en:

```
<CAMPAIGN_HOME>/devkits/CampaignServicesAPI/lib/
CampaignServicesClient30.jar
```

Para ver un ejemplo de Java que muestre la creación de una nueva oferta, consulte "OfferAPI.java" en la página 9. El mismo ejemplo se puede encontrar en la instalación de Campaign en:

```
<CAMPAIGN_HOME>/devkits/CampaignServicesAPI/samples/OfferAPI.java
```

Archivos .jar dependientes

Como resultado de la actualización a AXIS2 versión 1.5.2, la aplicación Java también debe actualizarse para utilizar los archivos `.jar` de la distribución de

AXIS2 1.5.2, ya que `CampaignServicesClient30.jar` depende de estos archivos `.jar`. Todos los archivos `.jar` dependientes se deben incluir en la ruta de clases Java de la aplicación y se pueden encontrar en el archivo `Campaign.war` ubicado en `<INICIO_CAMPAIGN>/Campaign.war`.

Extraiga los archivos `.jar` de `Campaign.war`, e inclúyalos en la ruta de clases Java.

constructor de la API Client

Al construir el objeto de la API Client, cambie el URL del servicio web y la firma de excepción, según se muestra en este ejemplo.

```
try {
URL serviceURL = new URL(PROTOCOL, HOST, PORT,
"/Campaign/services/CampaignServices30Service");
CampaignServices30SoapClient client = new
CampaignServices30SoapClient(serviceURL, TIMEOUT);
} catch (RemoteException exception) {
exception.printStackTrace();
}
```

constructores parametrizados de clases de soporte

Con el motor AXIS2, las clases y los apéndices generados no tienen constructores parametrizados. En su lugar, estas clases solo tienen el constructor sin argumentos predeterminado con métodos set y get para los miembros.

```
WSReference wsRef = new WSReference();
wsRef.setComponentTypeEnum(typeEnum);
wsRef.setId(id);
```

Implementaciones existentes utilizando directamente el WSDL

Esta información se aplica si está realizando una actualización a la versión 8.6 o posterior de IBM Campaign y si utiliza actualmente el WSDL para interactuar con la aplicación web de Campaign. El WSDL del servicio web Campaign se utiliza para generar apéndices del lado de cliente y clases de soporte utilizando una herramienta de convertidor de otra empresa. Aquí se proporcionan ejemplos de la utilización de la herramienta WSDL2Java de Apache AXIS2 1.5.2.

URL de servicio y ubicación de WSDL

El servicio web de Campaign para IBM Campaign se despliega en:

`http://host:port/Campaign/services/CampaignServices30Service`

El WSDL correspondiente se puede recuperar de:

`http://host:port/Campaign/services/CampaignServices30Service?wsdl`

Generación de apéndices y clases

La herramienta WSDL2Java de Apache AXIS2 1.5.2 se puede utilizar para generar los apéndices y las clases Java de soporte del WSDL. Un ejemplo de tarea Ant se muestra a continuación.

La herramienta también se puede utilizar desde la línea de comandos con un conjunto de argumentos similar. Los valores de los argumentos se pueden modificar para ajustarlos al entorno.

Nota: El vínculo ADB predeterminado se utiliza para el siguiente ejemplo de convertidor WSDL2Java.

```
<java classname="org.apache.axis2.wsdl.WSDL2Java" fork="true">
    <classpath refid="axis2.class.path"/> <!-Ruta de clases que tiene
bibliotecas AXIS2 -->
    <arg value="-uri"/>
    <arg file="CampaignServices30.wsdl"/> <!-Ubicación real de
WSDL -->
    <arg value="-s"/> <!-- Generar código de estilo de sincronización -->
    <arg value="-Euwc"/> <!-- Se encarga de la generación de tipos java de derivador
para elementos nillable = true. -->
    <arg value="-uw"/> <!-- Parámetros para anular recorte -->
    <arg value="-u"/> <!-- Clases de desempaquetado -->
    <arg value="-ns2p"/> <!-- Espacio de nombre para empaquetar correlación. El cliente
puede tener sus propios nombres de paquete. -->
    <arg value="http://webservices.unica.com/campaign/CampaignServices/
3.0=com.unica.publicapi.campaign.campaignservices.soap.v30"/>
    <arg value="-o"/> <!-- Directorio de salida -->
    <arg file="${autogen.java.dir}"/>
</java>
```

Utilización de apéndices y clases de soporte generados

El apéndice se puede utilizar de la siguiente manera:

```
CampaignServices30ServiceStub serviceStub = new
CampaignServices30ServiceStub(serviceURL);

serviceStub._getServiceClient().getOptions().setTimeOutInMilliSeconds
(webServiceTimeout); //Tiempo de espera en milisegundos.
```

La oferta se puede crear de la siguiente manera:

```
try{
    //Cambie el puerto y host para que coincida con su entorno.
    String serviceURL = "http://host:port/Campaign/services/CampaignServices30Service";
    CampaignServices30ServiceStub serviceStub = new CampaignServices30ServiceStub(serviceURL);
    long webServiceTimeout = 2*60*1000; // 2 minutos
    serviceStub._getServiceClient().getOptions().setTimeOutInMilliSeconds(webServiceTimeout); //Tiempo de espera en milisegundos.

    WSTextAttribute nameAttirbute = new WSTextAttribute();
    nameAttirbute.setMetadata(null);
    nameAttirbute.setName("uacOfferDescription");
    nameAttirbute.setValues(new String[]{"description " + System.currentTimeMillis()});

    WSTextAttribute[] wsAttributes = {nameAttirbute};
    // convert to WSAttributeArrays
    WSAttributeArrays obj = new WSAttributeArrays();
    obj.setTextAttributes(wsAttributes);
    //Cambie los valores de las variables siguientes para que coincidan con su entorno.
    String authorizationLoginName = "asm_admin"; //nombre de usuario de inicio de sesión
    String partitionName = "partition1"; //Usar su política de seguridad de Campaign
    String securityPolicyName = "Política global"; //Usar su política de seguridad de Campaign

    String offerName = "Primera oferta"; //Nombre de la oferta que se va a crear.
    String templateName = "Plantilla de oferta"; //Nombre de la plantilla de oferta existente.
    long folderID = 100; //ID real de la carpeta donde se creará esta oferta.
    //Para folderID <=0, la oferta se creará a nivel raíz.

    CreateOffer createOfferObject = new CreateOffer();
    createOfferObject.setAuthorizationLoginName(authorizationLoginName);
    createOfferObject.setPartitionName(partitionName);
    createOfferObject.setRequestedLocale(Locale.US.toString());
    createOfferObject.setSecurityPolicyName(securityPolicyName);
    createOfferObject.setName(offerName);
    createOfferObject.setFolderID(folderID);
    createOfferObject.setTemplateName(templateName);
    createOfferObject.setAttributes(obj);
    // realizar llamada a WebService de campaña
    WSCreateOfferResponse wsResponse = serviceStub.createOffer(createOfferObject);
    // estado de proceso
    WSRequestStatus status = wsResponse.getStatus();
    // fin
```

```
    WSOfferInfo offerInfo = wsResponse.getOfferInfo();
    System.out.println("status = "+status.getStatusType());
    System.out.println("offerInfo = "+offerInfo.getName());
} catch (Exception exception) {
    //Manejar la excepción aquí.
    exception.printStackTrace();
}
```

En este ejemplo, createOffer() solo acepta un parámetro de tipo CreateOffer.

Con el motor AXIS2, las clases y los apéndices generados ya no tienen constructores parametrizados.

Referencias de la API SOAP

Las referencias siguientes se han utilizado para preparar esta especificación de la API SOAP de IBM Campaign.

- "Basic Profile Version 1.1", Web Service Interoperability Organization (WS-I), 10 de abril de 2006. (`http://www.ws-i.org/Profiles/BasicProfile-1.1-2006-0310.html`)
- "SOAP 1.2 (draft)", W3C Soap working group, 24 de junio de 2003 (`http://www.w3.org/TR/soap/`)
- "JAX-RPC 1.1", Sun Microsystems, 14 de octubre de 2003 (`http://java.sun.com/webservices/jaxrpc/index.jsp`)
- Apache Web services working group (`http://ws.apache.org/axis2`)

Capítulo 2. Utilización de la API SOAP de IBM Campaign

Para utilizar la API SOAP de servicios web de IBM Campaign, puede utilizar el archivo .jar de la API de cliente o utilizar directamente el WSDL. Un ejemplo muestra cómo se utiliza el método de archivo .jar para crear una oferta.

Utilización del archivo .jar de la API de cliente para llamar a los servicios de Campaign

IBM Campaign proporciona una API de cliente que utiliza los servicios web de SOAP para interactuar con la aplicación web de Campaign. Este derivador está empaquetado en un archivo `.jar` que la aplicación cliente puede utilizar para llamar a la API de Campaign.

El archivo `.jar` se puede hallar en:

```
<CAMPAIGN_HOME>/devkits/CampaignServicesAPI/lib/
CampaignServicesClient30.jar
```

El siguiente ejemplo muestra la creación de una nueva oferta en el nivel de carpeta de oferta raíz en Campaign. La misma muestra se puede encontrar en:

```
<CAMPAIGN_HOME>/devkits/CampaignServicesAPI/samples/OfferAPI.java
```

Nota: El ejemplo utiliza algunos valores ficticios para los parámetros; los valores reales pueden ser diferentes.

Además, el URL para los servicios web de Campaign es `http://host:puerto/Campaign/services/CampaignServices30Service`, donde host y puerto hacen referencia al nombre de host y al número de puerto de la máquina donde se despliega la aplicación web de Campaign.

Si utiliza una muestra que le haya sido proporcionada, asegúrese de modificarla para ajustarla a su entorno de cliente.

OfferAPI.java

Para compilar y ejecutar este ejemplo Java, debe incluir todos los archivos `.jar` dependientes en la vía de acceso de clases Java. El archivo `CampaignServicesClient30.jar` depende de los archivos `.jar` del motor Apache AXIS2 SOAP y de otros archivos `.jar` de Apache comunes, que se pueden encontrar en `<CAMPAIGN_HOME>/Campaign.war`. Extraiga los archivos `.jar` de `Campaign.war`, e inclúyalos en la vía de acceso de clases Java.

```
import java.net.URL;
import java.util.Locale;
import com.unica.publicapi.campaign.campaignservices.CampaignServicesException;
import com.unica.publicapi.campaign.campaignservices.attribute.metadata.
    IAttribute Metadata;
import com.unica.publicapi.campaign.campaignservices.soap.v30.
    CampaignServices30SoapClient;
import com.unica.publicapi.campaign.campaignservices.soap.v30.WSAttribute;
import com.unica.publicapi.campaign.campaignservices.soap.v30.WSOfferInfo;
import com.unica.publicapi.campaign.campaignservices.utils.WSAttributeUtils;
```

© Copyright IBM Corp. 1998, 2017

```
/**
 * Esta es la clase de cliente java de ejemplo que muestra el uso de la API de
servicios Campaign SOAP.
 * Este ejemplo utiliza la fachada de CampaignServices30SoapClient para interactuar
con el servicio web de Campaign.
 * Aquí se muestra la creación de ofertas. Consulte la guía de la API para
obtener más detalles.
 *
 * @author AGijare
 *
 */
public class OfferAPI {

    /**
     * @param args
     */
    protected static CampaignServices30SoapClient CLIENT = null;

    private static void setup(){
        try {
           String protocol = "http"; //http o https
           String host = "localhost"; //Nombre de host de la campaña desplegada.
Utilizar el nombre de host apropiado.
           int port = 7001; //número de puerto de la campaña desplegada
           long timeOut = 2*60*1000; // 2 minutos
           String servicesURI = "/Campaign/services/CampaignServices30Service";
           CLIENT = new CampaignServices30SoapClient(
               new URL(protocol, host, port, servicesURI),
               timeOut);
      } catch (Exception exception) {
          exception.printStackTrace();
          System.exit(-1);
      }
    }

    public static void main(String[] args) {
        //Cambie los valores de las variables siguientes para que coincidan con el
entorno.
        String userName = "nombre_usuario"; // Nombre de usuario de inicio de sesión
        String partitionName = "partición1"; //Usar el nombre de partición adecuado de
Campaign        Locale loc = Locale.US;
        String securityPolicy = "Global"; //Usar la política de seguridad de
Campaign
        String offerName = "Oferta1";
        String offerTemplate = "Plantilla de oferta"; // Plantilla a partir de la cual
se creará la oferta.
        long folderID = 1002; //ID real de la carpeta donde esta oferta
se creará. Cuando folderID <=0, la oferta se creará a nivel raíz.
        //Atributos de oferta
        WSAttribute[] wsAttributes = {
            WSAttributeUtils.getWSTextAttribute(IAttributeMeta
data.AC_OFFER_DESCRIPTION_ATTRIBUTE_NAME, null, new String[]{"descripción "
+ System.currentTimeMillis()})
        };

        setup();

        try {
            WSOfferInfo wsOfferInfo = CLIENT.createOffer(userName,
partitionName, loc, securityPolicy,
                offerName, folderID, offerTemplate, wsAttributes);
            System.out.println("Created offer: " + wsOfferInfo.getName());
        } catch (CampaignServicesException e) {
            e.printStackTrace();
        }
    }
}
```

Utilización del WSDL para llamar a los servicios de Campaign

Los servicios de Campaign se pueden llamar utilizando el archivo de WSDL de servicios web de can Campaign: `CampaignServices30.wsdl`.

El archivo `CampaignServices30.wsdl` se puede hallar en:

```
http://host:port/Campaign/services/CampaignServices30Service?wsdl
```

o en la distribución de Campaign en:

```
<CAMPAIGN_HOME>/devkits/CampaignServicesAPI/lib/
```

La aplicación Java cliente necesita utilizar las clases y los apéndices generados de WSDL utilizando cualquier herramienta de conversión de WSDL a Java de terceros. IBM recomienda el uso de Apache AXIS.

El javadocs creado a partir de apéndices y clases generados de WSDL utilizando Apache AXIS2 se puede encontrar en:

```
<CAMPAIGN_HOME>/devkits/CampaignServicesAPI/javadocs/index.html
```

Nota: Todos los archivos `.jar` dependientes deben incluirse en la ruta de clases Java. El archivo `CampaignServicesClient30.jar` depende de los archivos `.jar` del motor SOAP de Apache AXIS2 y otros archivos `.jar` comunes de Apache, que se pueden encontrar en el archivo `Campaign.war` ubicado en `<INICIO_CAMPAIGN>/Campaign.war`. Extraiga los archivos `.jar` de `Campaign.war`, e inclúyalos en la ruta de clases Java.

Consideraciones sobre el rendimiento

El perfil actual de rendimiento de la implementación de la API CampaignServices es similar al que la aplicación ha experimentado a través de la GUI. Algunas API están diseñadas explícitamente para el rendimiento. En particular, la API `listCampaignsByPage()` permite la paginación relativamente eficiente.

La interfaz SOAP, debido a su naturaleza, introduce latencia y sobrecarga porque todos los datos se convierten al formato XML, que en algunos casos es bastante detallado. Por ejemplo, una simple llamada de SOAP de bucle de retorno puede tardar 100 ms en una red típica (Java 1.4.x era aún más lenta). La API se ha optimizado para los casos de uso del portal típico y otras empresas de aplicación cliente, como see `listOffersByPage()`, por lo que el rendimiento de SOAP debería ser adecuado.

Sin embargo, el cliente debe tener cuidado de no poner demasiada carga en los servicios normales que CampaignServices presta a las solicitudes de usuario web. En general, se espera que las necesidades del proceso de un usuario de API no excedan de las de un usuario web típico de IBM Campaign.

Capítulo 3. Tipos de datos de la API SOAP

La API SOAP de servicios de IBM Campaign utiliza los siguientes tipos de datos públicos.

WSReference

Un derivador simple alrededor de un ID de base de datos:

- **`componentTypeEnum`**: tipo enumerado que indica el tipo de componente para el que es el ID. Uno de los valores siguientes:
  - FOLDER
  - CAMPAIGN
  - FLOWCHART
  - TCS_CELL
  - OFFER
  - OFFER_LIST
  - OFFER_TEMPLATE
- *id*: valor *Long*, que define un identificador numérico específico de base de datos para la referencia.

WSVersion

Un tipo de derivador que captura los diversos componentes de una versión, incluidos los siguientes:

- *major*: entero que define el número de versión principal, como '8' de la versión completa 8.1.2.3.
- *minor*: entero que define el número de versión menor, como '1' de la versión completa 8.1.2.3.
- *maintenance*: entero opcional que define el número de mantenimiento de la versión, si es aplicable, como '2' de la versión completa 8.1.2.3. Nunca se suministra con una versión de la API.
- *patch*: entero opcional que define el número de release del parche, si es aplicable, como '3' de la versión completa 8.1.2.3. Nunca se suministra con una versión de la API.

WSServiceInfo

Un tipo de derivador simple alrededor de información sobre el servicio. Contiene los campos siguientes:

- *apiVersion*: una instancia de *WSVersion* que define la versión más actual de la API soportada por el servicio. (*apiVersion* incluye información sólo versión principal y menor.)
- *campaignVersion*: una instancia de *WSVersion* que define la versión de la instancia de IBM Campaign subyacente.
- *name*: nombre interno del servicio, como "CampaignServices30Service".

© Copyright IBM Corp. 1998, 2017

WSAttributeTypeEnum

Un tipo enumerado que define todos los tipos de atributos posibles, uno de los siguientes:

- STANDARD: atributo estándar o base definido por Campaign.
- CUSTOM: atributo definido por otra aplicación de IBM , el cliente o un tercero.
- INPUT_PARAMETER: parámetro de entrada, como un atributo que se utiliza para ejecutar un diagrama de flujo de IBM Campaign.
- OUTPUT_PARAMETER: parámetro de salida, como por ejemplo un atributo cuyo valor se completa como resultado de una ejecución de diagrama de flujo en IBM Campaign.

WSAttributeStatusEnum

Una enumeración de todos los posibles códigos de estado de atributo, uno de los siguientes:

- ACTIVE: el atributo está activo y se puede utilizar a voluntad.
- RETIRED: el atributo se ha eliminado del servicio y no debe utilizarse.

WSAccessTypeEnum

Un tipo enumerado que define todos los tipos de acceso posibles a los valores de atributos, uno de los siguientes:

- READ_ONLY: el valor del atributo se puede leer y visualizar, pero no modificar.
- READ_WRITE: el valor del atributo se puede leer, visualizar y modificar.

El acceso al atributo es adicional a los permisos de seguridad. Por ejemplo, si la política de seguridad para el usuario cliente deniega el acceso de lectura a un atributo en particular, el acceso al atributo no puede alterar temporalmente ese valor de seguridad. De hecho, la API nunca devolverá el atributo al cliente.

WSSelectTypeEnum

Define todos los tipos de selección posibles para un valor de atributo en particular, uno de los siguientes:

- NONE: ninguna selección (*hasOptions* es false).
- SINGLE_SELECT: solo se puede elegir una opción de atributo de la lista de opciones posibles cada vez (solo es válido si es un atributo *hasOptions*).
- MULTIPLE_SELECT: similar a SINGLE_SELECT, excepto que se puede seleccionar una o varias opciones a la vez.

WSRunStatusEnum

Un tipo enumerado de todos los estados posibles de ejecución de diagrama de flujo, rama o celda, uno de los siguientes:

- NOT_STARTED: la ejecución se ha planificado, pero todavía no se ha iniciado aún.
- RUNNING: ejecución en curso.
- CANCELLED: la ejecución se ha cancelado, por un usuario de Campaign o mediante esta API.
- SUCCEEDED: la ejecución se ha completado con éxito.

- FAILED: la ejecución ha fallado; los detalles del error se informan por separado. (Consulte “WSRunResults” en la página 19).

WSRunTypeEnum

Un tipo enumerado de todos los tipos de ejecución posibles, uno de los siguientes:

- NOT_RUN
- TEST_RUN
- PRODUCTION_RUN
- RUN_SKIPPED
- TEST_FLOWCHART
- PRODUCTION_FLOWCHART
- TEST_BRANCH
- PRODUCTION_BRANCH
- TEST_PROCESS
- PRODUCTION_PROCESS

WSAttribute

Los atributos proporcionan un mecanismo simple y ampliable para conectar datos arbitrarios a instancias de componentes accesibles a través de la API, ya sean datos estándar como un *nombre* para una campaña, los parámetros de entrada de ejecución de diagrama de flujo como *género* o datos personalizados arbitrarios especificados por otra aplicación de IBM o cliente de IBM .

Nota: En esta API, los atributos se utilizan para modelar la mayoría de los datos de componente, no solo los atributos personalizados de Campaign.

Por lo general, los componentes tienen muchos atributos asociados, que la API CampaignServices expone como una correlación de tipo especial denominada *AttributeMap*. Los datos de atributo se representan como una clase concreta de tipo en toda la API, como *WSDecimalAttribute*, para los atributos que contienen datos decimales (numéricos de precisión doble).

Cada atributo incluye lo siguiente:

- *Nombre*: nombre exclusivo del atributo. Este nombre sirve como clave para acceder al atributo y sus metadatos dentro de la instancia de componente en la que aparece. El formato del nombre está sin definir; en algunos casos lo asigna el servicio, el cliente o un usuario de IBM Campaign.

  Generalmente, este nombre no es el nombre de visualización que se presenta a Campaign o a un usuario cliente. La API lo puede estandarizar, como *uacDescription*, IBM Campaign lo puede asignar al publicar diagramas de flujo o bien la aplicación o cliente de IBM lo pueden asignar al definir atributos personalizados. Sin embargo, en todos los casos, se garantiza que el nombre será exclusivo.
- *Metadatos*: (opcional) información sobre los datos del atributo, como el tipo de datos del valor, el nombre de visualización, la descripción, las solicitudes, el valor predeterminado, el tipo de selección, la longitud (texto), la precisión (decimales), las opciones (de selección única o múltiple), etc. Consulte “WSAttributeMetadata” en la página 16.
- *Valores*: matriz de cero o más objetos de valor de tipo firme. La clase de atributo concreta suministra el campo de valores; el tipo de cada valor debe ser el mismo

y ajustarse a la definición de tipo en el campo de metadatos del atributo. Sin embargo, no todos los atributos soportan varios valores.

Están soportados los siguientes tipos de atributos concretos:

– **WSBooleanAttribute**: atributo cuyo valor es booleano, es decir, *true* o *false*.
– **WSIntegerAttribute**: valor entero (*java.lang.Long*).
– **WSDecimalAttribute**: valor de número decimal de precisión doble (*java.lang.Double*).
– **WSCurrencyAttribute**: valor compuesto de moneda, que incluye un código de moneda ISO 4217 opcional del valor de moneda, como "*USD*" para dólar americano y los valores de moneda capturados como *Doble*. Si el código de moneda no se proporciona, se presupone el valor predeterminado utilizado por IBM Campaign.

  Consulte `http://www.xe.com/symbols.php` para obtener una lista de países, símbolos de moneda y códigos. El entorno local utilizado para un valor de moneda puede ser diferente del entorno local preferido de un usuario.
– **WSCalendarAttribute**: cuyos valores son fechas de calendario, o fechas y horas, en algún huso horario y configuración regional.
– **WSTextAttribute**: una cadena de texto Unicode (posiblemente nulo o vacío).

Nota: La lista de atributos posibles normalmente es diferente para cada tipo de componente, pero las listas se pueden solapar.

WSAttributeMetadata

WSAttributeMetadata define la información sobre los datos de un atributo de tipo en particular, como un tipo de datos de valor, texto localizado (nombre de visualización, descripción, solicitudes) su valor predeterminado, el rango de valores permitidos, el tipo de selección, las opciones (de selección única o múltiple). Al igual que con los atributos, es especifican los metadatos de atributo. Por ejemplo, WSDecimalAttribute *miNúmero* debe disponer de un enlace *DecimalAttributeMetadata* de WS y todos los valores, incluidos los valores de atributo, el valor predeterminado de metadatos y los valores de opciones posibles, serán de tipo *Doble*.

Las descripciones, las etiquetas y otro texto de metadatos de atributos están generalmente localizados; sin embargo, el texto especificado por el usuario puede que solo esté disponible tal como lo escribió el usuario. Cada llamada a la API incluye un entorno local solicitado que el código de cliente puede utilizar para definir el entorno local en el que un usuario en particular desea que se visualicen los mensajes localizados. Se utilizan las políticas de reserva de entorno local de Java normales para satisfacer la solicitud.

WSAttributeMetadata contiene los campos siguientes:

- *name*: nombre del atributo, estándar o personalizado; también el nombre utilizado por el atributo que se vincula a estos metadatos. Los atributos estándar se definen por el sistema y tienen nombres estándar en un espacio de nombres reservados (es decir, utilizan un prefijo "*uac*"), los nombres personalizados pueden utilizar otro convenio de denominación.

  Nota: El nombre de atributo debe ser exclusivo, nunca se localiza, y tiene restricciones de longitud (que dependen del contenido de caracteres y base de datos). El nombre distingue entre mayúsculas y minúsculas, y puede estar compuesto de cualquier combinación de caracteres de letras o dígitos Unicode, más el carácter de subrayado '_', pero no puede empezar por un dígito.

- *description*: descripción opcional del atributo. Adecuado para una ayuda contextual u otra presentación de interfaz de usuario.
- Predicados: diversos predicados que describen el atributo:
  - *isRequired*: true si el atributo es obligatorio.
  - *isInternal*: true si el atributo lo define el sistema y es solo para uso interno (no se debe presentar a un usuario).
  - *isGenerated*: true si los valores del atributo los genera automáticamente IBM Campaign al crear el componente, como un código de celda objetivo. Normalmente, *accessTypeEnum* será READ_ONLY para los valores generados.
  - *hasOptions*: true si el atributo tiene opciones. Implica que hay opciones definidas para estos metadatos y que *selectTypeEnum* es SINGLE_SELECT o MULTIPLE_SELECT.
- *typeEnum*: *WSAttributeTypeEnum* que define el tipo de atributo, como STANDARD o CUSTOM.
- *statusEnum: WSAttributeStatusEnum* que define el estado del atributo, como ACTIVE.
- *accessTypeEnum*: *WSAccessTypeEnum* que define el tipo de acceso para el valor de atributo, como READ_ONLY.
- *selectTypeEnum*: *WSAccessTypeEnum* que define el tipo de selección que se utiliza para el atributo, como SINGLE. Debe ser NONE para los atributos de campaña y celda, o si no se proporcionan opciones.
- *componentTypeEnum*: *WSComponentTypeEnum* de todos los componentes posibles de Campaign expuestos por la API, como CAMPAIGN, FOLDER, etcétera.
- *defaultValue* (solo diagramas de flujo): valor predeterminado de tipo opcional para el atributo. Este valor lo proporciona la clase concreta de metadatos de atributo, como valor predeterminado de *WSTextAttributeMetadata* de tipo Cadena. (Consulte la descripción de los valores de atributo). Para otros componentes distintos de los diagramas de flujo, el valor predeterminado está sin definir.
- *options*: lista opcional de opciones para este atributo. Combinadas, las opciones de un atributo definen el conjunto exacto de valores permitidos para ese atributo; cada opción es de tipo firme, por lo que por ejemplo *WSTextAttributeMetadata* solo puede tener una *WSTextAttributeOption* vinculada.

Nota: Hay una restricción sobre las opciones; solo se soportan atributos de texto.

Cada opción define lo siguiente:

- *prompt*: solicitud de la opción adecuada para menús desplegables, como "Masculino", como opción de atributo de género. A diferencia de la solicitud de metadatos, los nombres de visualización de la opción no suelen incluir puntuación.
- *description:* descripción localizada de la opción, como "Una persona de convicción masculina". Adecuado para el texto de ayuda contextual.
- *isDefault*: true si esta opción en particular es el valor predeterminado. Para los tipos de selección MULTIPLE_SELECT, se puede marcar más de una opción como valor predeterminado.
- *value*: valor de opción de tipo. Igual que en defaultValue de los metadatos de atributos, este valor lo proporciona la subclase de opción concreta, por ejemplo, un valor de *WSDecimalAttributeOption* es de tipo Decimal. (Consulte la descripción de los valores de atributo). Continuando con el ejemplo de *género* anterior, el valor puede declararse como una cadena (*WSTextAttributeOption*) o como código numérico, 123 (*WSDecimalAttributeOption*).

WSCampaignInfo

Un tipo de derivador simple alrededor de los datos de atributos de la campaña.

Contiene los campos siguientes:

- *reference*: referencia de la campaña.
- *name*: nombre de la campaña (*uacName*); no se garantiza que sea exclusivo.
- *description*: descripción opcional de la campaña (*uacDescription*).
- *campaignCode*: código exclusivo de la campaña (*uacCampaignCode*); asignado por el cliente o Campaign.

WSComponentOrFolderInfo

Contiene una combinación de la campaña derivada o los datos de atributos de carpeta, como el nombre de visualización, su referencia, etc.

Contiene los campos siguientes:

- *reference*: referencia del componente o carpeta.
- *name*: nombre de componente o carpeta (*uacName*); no se garantiza que sea exclusivo.
- *description*: descripción opcional del componente o carpeta (*uacDescription*).
- *componentCode*: código exclusivo del componente, o nulo si es una carpeta.

WSTargetCellInfo

Un derivador simple alrededor de datos de atributos de filas de celda objetivo.

Contiene los campos siguientes:

- *reference*: referencia de celda.
- *name*: nombre de celda (*uacName*); no se garantiza que sea exclusivo.
- *description*: descripción opcional de celda (*uacDescription*).
- *cellCode*: código de celda (*uacCellCode*); asignando por el cliente o Campaign. Tenga en cuenta que se puede forzar que los códigos de celda sean exclusivos definiendo el parámetro de configuración IBM Campaign *DuplicateCellCodesAllowed* como false.
- *flowchartName*: nombre opcional del diagrama de flujo al que está vinculada la celda.

WSMetricsInfo

Un tipo de derivador simple alrededor de los datos analíticos de la campaña, incluido el número de contactos. Contiene los campos siguientes:

- *totalContacts*: valor largo que proporciona el número total de contactos.
- *respuestas*: lista de tipo de instancias de *WSMetricsResponse*, cada una de las instancias define la información de contacto para una respuesta:
  - *typeCode*: cadena que define el código de tipo de respuesta, como *PHC* para un contacto telefónico.
  - *count*: valor largo que proporciona el número de veces que ha ocurrido este contacto.

WSRunResults

Un tipo de derivador alrededor de los resultados de la ejecución de un diagrama de flujo, un cuadro de proceso o una celda, posiblemente todavía en curso, que incluye el estado de la ejecución, la fecha y la hora de inicio y finalización de la ejecución del diagrama de flujo y recuentos.

Contiene los campos siguientes:

- *sourceReference*: referencia opcional del origen del resultado de la ejecución. Según el contexto en el que se capten los resultados de la ejecución, puede hacer referencia a un diagrama de flujo, a un cuadro de proceso de diagrama de flujo o a una celda objetivo. En cualquier caso, los datos del resultado de la ejecución restantes hacen referencia a este origen.
- *flowchartName*: nombre del diagrama de flujo que se ha ejecutado.
- *flowchartId*: ID de base de datos para el diagrama de flujo.
- *runId*: ID de base de datos de la ejecución.
- *typeEnum*: tipo enumerado que define la clase de ejecución que ha generado los resultados, como PRODUCTION_PROCESS (consulte *WSRunTypeEnum*).
- *statusEnum*: tipo enumerado que define el estado de ejecución, como RUNNING (consulte *WSRunStatusEnum*).
- *statusCode*: código de estado entero opcional.
- *statusMessage*: mensaje de estado opcional.
- *startDate*: fecha y hora opcional del calendario de cuando se ha iniciado la ejecución; será nulo si la ejecución no se ha iniciado.
- *endDate*: igual que *startDate*, pero la fecha y hora en que ha finalizado la ejecución (éxito o error); será nulo si la ejecución no se ha iniciado o aún no ha finalizado.
- *count*: recuento total opcional de contactos seleccionados por la ejecución; puede ser cero o nulo si la ejecución no se ha completado.

WSOfferInfo

Un tipo de derivador simple alrededor de los datos de atributos de ofertas o listas de ofertas.

Contiene los campos siguientes:

- *reference*: referencia de la oferta o lista de ofertas.
- *name*: nombre de oferta o lista de ofertas (*uacName*); no se garantiza que sea exclusivo.
- *description*: descripción opcional (*uacDescription*).
- *offerCode*: código de oferta (*uacOfferCode*) si es una oferta, o nulo si es una lista de ofertas. (No se garantiza que sea exclusivo.)

WSOfferCodeOrName

Un tipo de derivador simple alrededor de los datos de códigos de oferta o nombres de listas de códigos de oferta.

Contiene los campos siguientes:

- *isCode:* booleano que indica si el campo *codeOrName* es un presunto código de oferta (true) o el nombre de una lista de ofertas (false).

- *codeOrName*: código de oferta exclusivo (*uacOfferCode*) si es una oferta, o el nombre de la lista de ofertas.

WSOfferValidationInfo

Un tipo de derivador simple alrededor de la información de validación de la oferta.

Contiene los campos siguientes:

- *errorCode*: si no es nulo, define el código de error de validación alfanumérica. Consulte la clase IStandardDefinitions para ver los códigos de error.
- *errorMessage*: mensaje localizado opcional que describe el error (si se ha producido uno).
- *codeOrName*: código de oferta o nombre de lista de ofertas validados.
- *reference*: referencia de oferta o lista de ofertas, si es válida.

WSOfferTemplateInfo

Un tipo de derivador simple alrededor de los datos de plantilla de oferta.

Contiene los campos siguientes:

- *reference*: referencia de la plantilla de oferta.
- *name*: nombre de plantilla de oferta; se garantiza como exclusivo.
- *description*: descripción opcional (*uacDescription*).
- *offerTemplateID*: ID de base de datos exclusivo de plantillas de ofertas.

WSBulkOfferInfo

Se utiliza para crear ofertas colectivas.

Contiene los campos siguientes:

- *offerName*: nombre de la oferta que se está creando.
- *attributes*: matriz de tipos WSAttribute que indica los atributos de la oferta.

WSOfferInfoStatus

Un tipo de retorno para el método de API *bulkCreateOffers()* que indica el estado de la creación de ofertas colectivas.

Contiene los campos siguientes:

- *name*: nombre de la oferta.
- *code*: código de la oferta. Es nulo si falla la creación de la oferta.
- *description*: descripción de la oferta.
- *reference*: WSReference de la oferta creada. Es nulo si falla la creación de la oferta.
- *status*: una instancia de WSRequestStatus que indica el estado de la creación de la oferta.

Capítulo 4. Métodos de la API SOAP

La API SOAP de servicios de IBM Campaign utiliza los métodos siguientes.

Métodos de la API SOAP: Servicio

La API SOAP de servicios de IBM Campaign proporciona una forma de obtener información de identificación sobre el propio servicio.

getServiceInfo

```
WSServiceInfo getServiceInfo()
    throws CampaignServicesException;
```

Devuelve la información sobre el servicio, como la versión de API más reciente que soporta, la versión completa de la instancia de IBM Campaign subyacente, etc.

Nota: Esta llamada no necesita ninguna información de cliente y no se aplica ningún permiso de seguridad.

Parámetros

Ninguna.

Devoluciones

Devuelve una instancia de *WSServiceInfo*.

Errores

Ninguna.

Métodos de la API SOAP: Atributos

La mayoría de los datos de la instancia de componentes se pueden exponer mediante la API SOAP de IBM Campaign como atributos o metadatos de atributos.

En algunos casos, las definiciones de metadatos de atributos son globales para IBM Campaign (como, por ejemplo, los atributos personalizados de campañas). En otros casos, están restringidas a un componente en particular como, por ejemplo, las variables de usuario de diagramas de flujo. A menos que se indique lo contrario, todos los atributos se pueden leer si el cliente dispone de los permisos de seguridad suficientes para ello.

Nota: Esta API expone solamente los componentes que están activos y a los que el cliente puede acceder. El soporte público se limita a un subconjunto de las API disponibles.

getAttributesByName

```
Map<String, WSAttribute>
    getAttributesByName(String userCredential, String partitionName,
        Locale requestedLocale,
        WSReference reference,
        String[] names)
    throws CampaignServicesException;
```

© Copyright IBM Corp. 1998, 2017

Captura los atributos con nombre asociados a la instancia de componente especificada (puede estar vacía).

Parámetros

userCredential: la credencial de usuario cliente.

requestedLocale: entorno local opcional para utilizar para esta solicitud; si no se suministra, se utilizan las preferencias de entorno local del usuario de IBM . Se aplicará el algoritmo predeterminado del entorno local normal si es necesario.

partitionName: nombre opcional de la partición de campaña que se va a utilizar. Si no se ha definido, se utiliza la partición predeterminada.

reference: *reference* para la instancia del componente que contenga los atributos deseados. Se emite una *InvalidComponentException* si la referencia no es válida o el componente no existe.

names: matriz de nombres opcional de atributos que desea captar (no nombres de visualización): si no se suministra, se devuelven todos los atributos. Se emite *AttributeNotFoundException* si uno de los atributos denominados no existe.

Devoluciones

Una correlación de tipo de cero o más atributos; el nombre de atributo es la clave de entrada de correlación y la instancia de atributo es el valor de entrada.

Errores

InvalidComponentException, AttributeNotFoundException

AuthorizationException, DataException

Nota: Todas estas excepciones se recortan dentro de la CampaignServicesException.

updateAttributes

```
void updateAttributes(String userCredential, String partitionName,
        Locale requestedLocale, WSReference reference,
        boolean allowCreate,
        WSAttribute[] attributes)
    throws CampaignServicesException;
```

Actualice uno o varios atributos de la instancia de componente con los valores de atributos suministrados.

Lógica de la actualización

La lógica de la actualización es la siguiente.

Para cada atributo contenido en la correlación de atributos suministrada:

1. Si el nombre de atributo coincide con un atributo existente, intente sobrescribir su campo *values* con el campo de valores suministrado.
2. Si el atributo aún no existe, *allowCreate* es true y sus metadatos son conocidos, cree el atributo. Esto se aplica a los metadatos de atributos globales así como a los atributos de instancia (excepto a diagramas de flujo).
3. Si el tipo de valor o algún otro aspecto de la definición de metadatos del atributo no se cumple, o uno o varios de los valores suministrados no son válidos, están fuera de rango, etc., se emite *InvalidAttributeException*.

4. Si no, se emite *AttributeNotFoundException* si el atributo denominado no existe.

Nota: En caso de una excepción, no se confirmará ninguna de las actualizaciones.

Este método en particular no soporta la definición de nuevos atributos personalizados; utilice el método `createAttributeMetadata()` para esto.

En todos los casos, la operación de actualización de atributos está sujeta a las restricciones y la validación de seguridad normales. Es responsabilidad del cliente determinar qué atributos son necesarios para una instancia de componente en particular, los tipos correctos, etc.

Parámetros

userCredential: la credencial de usuario cliente.

requestedLocale: entorno local opcional para utilizar para esta solicitud.

partitionName: nombre opcional de partición de campaña para utilizar.

reference: referencia para la instancia de componente que contiene los atributos que desea actualizar.

allowCreate: indica si se debe crear un nuevo atributo si no existe aún para el componente. (Consulte "Lógica de la actualización" en la página 22).

attributes: matriz de atributos que se desea actualizar; el nombre de atributo se utiliza para localizar el atributo que se va a actualizar y los valores nuevos se utilizan para actualizar el valor del atributo existente como un único objeto del tipo apropiado o una matriz, si es aplicable. (Consulte Capítulo 5, "Excepciones comunes de la API SOAP", en la página 59).

Devoluciones

Ninguna.

Errores

InvalidComponentException, AttributeNotFoundException, InvalidAttributeException

AuthorizationException, DataException

getAttributeMetadataByName

```
Map<String, WSAttributeMetadata>
    getAttributeMetadataByName(String userCredential,
        StringpartitionName, Locale requestedLocale,
        WSReference reference, String[] names)
    throws CampaignServicesException;
```

Capta las definiciones de metadatos de atributos con nombre vinculadas a un componente o una plantilla en particular, o definidas globalmente.

Parámetros

userCredential: la credencial de usuario cliente.

requestedLocale: entorno local opcional para utilizar para esta solicitud.

partitionName: nombre opcional de partición de campaña para utilizar.

reference: referencia opcional para el componente o plantilla que contiene los metadatos de atributos deseados. Si sólo se proporciona ComponentTypeEnum, la captación se restringe a los componentes de ese tipo. Si la referencia no se suministra de ningún modo, la captación devuelve todas las definiciones de metadatos globales, para todos los tipos de componentes. Se emite *InvalidComponentException* si la referencia suministrada no es válida.

names: matriz de nombres opcional de los metadatos de atributo que desea captar. Si no se suministra, se devuelven todos los metadatos para el componente, o se definen globalmente si no se proporciona ninguna referencia. Se emite *AttributeNotFoundException* si una o varias de las definiciones de metadatos de atributos especificadas no existen.

Devoluciones

Una correlación de tipo de cero o más definiciones de metadatos de atributo; el nombre de atributo es la clave de entrada de correlación y los metadatos de atributos es el valor de entrada.

Errores

InvalidComponentException, AttributeNotFoundException

AuthorizationException, DataException

createAttributeMetadata

```
void createAttributeMetadata(String userCredential,
        String partitionName,
        Locale requestedLocale, WSReference reference,
        WSAttributeMetadata[] attributeMetadata)
    throws CampaignServicesException;
```

Cree una o varias definiciones de metadatos de atributos nuevas y, opcionalmente, enlácelas a un componente o plantilla en particular.

Parámetros

userCredential: la credencial de usuario cliente.

requestedLocale: entorno local opcional para utilizar para esta solicitud.

partitionName: nombre opcional de partición de campaña para utilizar.

reference: referencia opcional para el componente o la plantilla a los que desea vincular los metadatos. Si no se proporciona, la definición de metadatos creada será global. Si se proporciona la referencia, pero no es válida, se emite una *InvalidComponentException*.

attributeMetadata: matriz de definiciones de metadatos de atributos que se van a vincular. Si uno o varios de los metadatos especificados ya están vinculados al componente, es decir, el nombre no es exclusivo, se emite una *AttributeExistsException*. Se emite una *InvalidAttributeException* si hay un problema con uno o varios de los metadatos especificados, es decir, es internamente incoherente.

Devoluciones

Ninguna.

Errores

InvalidComponentException, AttributeExistsException, InvalidAttributeException

AuthorizationException, DataException

updateAttributeMetadata

```
void updateAttributeMetadata(String userCredential,
        String partitionName,
        Locale requestedLocale, WSReference reference,
        boolean allowCreate,
        WSAttributeMetadata[] attributeMetadata)
    throws CampaignServicesException;
```

Actualice una o varias definiciones de metadatos de atributos del componente o plantilla especificados, creando opcionalmente nuevas definiciones de metadatos si es necesario.

Lógica de la actualización

La lógica de la actualización es la siguiente.

Para cada definición de metadatos de atributos contenida en la matriz suministrada:

1. Si el nombre de atributo no coincide con metadatos existentes vinculados al componente, haga lo siguiente según el valor del parámetro *allowCreate*:
   a. *True*: cree una nueva definición de metadatos. Es funcionalmente idéntico a utilizar la solicitud `createAttributeMetadata()`.
   b. *False*: emita *AttributeNotFoundException*.
2. Si el tipo de datos de metadatos de atributos es diferente, se emite *InvalidAttributeException*.
3. Intente sobrescribir la definición de metadatos de atributos existente con los valores de campo de los metadatos suministrados, si no se emite *InvalidAttributeException*. Solo están soportadas las siguientes actualizaciones (si no se emite *InvalidAttributeException*):
   a. *name*: no se puede cambiar (el nombre es la clave) .
   b. *displayName*: aceptar el nuevo valor.
   c. *description*: aceptar el nuevo valor.
   d. *isRequired*: solo permite cambiar de *true* a *false*.
   e. *isInternal*: aceptar el nuevo valor.
   f. *isGenerated*: no se permite ningún cambio.
   g. *attributeTypeEnum*: no se permite ningún cambio.
   h. *accessTypeEnum*: aceptar el nuevo valor.
   i. *selectTypeEnum*: aceptar estas transiciones si se proporcionan opciones:
      1) NONE a SINGLE_SELECT o MULTIPLE_SELECT
      2) SINGLE_SELECT a MULTIPLE_SELECT
   j. *options*: se pueden añadir opciones, pero no suprimir. Solo los siguientes cambios de opciones están soportados (según la coincidencia de valores):
      1) *displayName*: aceptar el nuevo valor (sin ajuste)
      2) *description*: aceptar el nuevo valor (sin ajuste)
      3) *isDefault*: aceptar el nuevo valor; aunque debe coincidir con SelectTypeEnum.
      4) *valor*: no se permite ningún cambio (el valor es la clave) .

k. *defaultValue* (solo diagramas de flujo): aceptar el nuevo valor predeterminado.
l. *maximumLength* (solo texto): aceptar la nueva longitud si es mayor.

4. Si la definición de metadatos de atributo no es coherente internamente, se emite *InvalidAttributeException*.
5. Si es necesario, busque todas las instancias de componentes que hagan referencia a los metadatos de atributos actualizados y actualícelas según proceda.

Nota: En caso de una excepción, no se confirmará ninguna de las actualizaciones.

En todos los casos, la operación de actualización de atributos está sujeta a las restricciones y la validación de seguridad normales.

See `createAttributeMetadata()`, `deleteAttributeMetadata()`

Parámetros

userCredential: la credencial de usuario cliente.

requestedLocale: entorno local opcional para utilizar para esta solicitud.

partitionName: nombre opcional de partición de campaña para utilizar.

reference: referencia opcional para la instancia de componente que contiene los atributos deseados. Si no se suministra, la actualización estará restringida a las definiciones de metadatos globales. Se emite *InvalidComponentException* si la referencia suministrada no es válida.

allowCreate: si es true, se crearán las definiciones de metadatos que no existen actualmente (es funcionalmente equivalente a utilizar el método `createAttributeMetadata()`).

attributeMetadata: una matriz de definiciones de metadatos de atributos que se debe actualizar (y añadir si el indicador *allowCreate* es true). El nombre de atributo se utiliza para localizar la definición de metadatos que desea actualizar y los datos restantes se utilizan para actualizar la definición existente. (Consulte "Lógica de la actualización" en la página 25).

Devoluciones

Ninguna.

Errores

InvalidComponentException, InvalidAttributeException

AuthorizationException, DataException

deleteAttributeMetadata

```
void deleteAttributeMetadata(String userCredential,
        String partitionName,
        Locale requestedLocale, WSReference reference,
        String[] names)
    throws CampaignServicesException;
```

Suprime una o varias definiciones de metadatos de atributos con nombre del componente especificado, la plantilla (solo metadatos de atributos personalizados) o las definiciones de metadatos de atributos globales.

Como parte de esta tarea, el método encontrará todos los componentes que hacen referencia a los metadatos suprimidos y los actualizará según sea apropiado.

Nota: Sin embargo, en el caso de una excepción, ninguna de las supresiones se confirmará.

Parámetros

userCredential: la credencial de usuario cliente.

requestedLocale: entorno local opcional para utilizar para esta solicitud.

partitionName: nombre opcional de partición de campaña para utilizar.

reference: referencia opcional del componente o plantilla que contiene los atributos que se deben suprimir. Si no se suministra, la supresión estará restringida a las definiciones de metadatos globales. Se emite *InvalidComponentException* si la referencia suministrada no es válida.

Nota: Si la matriz de nombres opcional de los metadatos de atributo no se suministra, este método intentará suprimir todos los metadatos de atributo personalizados asociados con el componente, o todas las definiciones globales si no se ha proporcionado la referencia.

nombres: matriz de nombres opcional de los metadatos de atributo que desea suprimir. Se emite *AttributeNotFoundException* si uno o más de los metadatos de atributos no existen. Se emite una *InvalidAttributeException* si un atributo no se ha podido eliminar.

Devoluciones

Ninguna.

Errores

InvalidComponentException, AttributeNotFoundException, InvalidAttributeException

AuthorizationException, DataException

Métodos de la API SOAP: Campañas y diagramas de flujo

La API SOAP de IBM Campaign da soporte a las operaciones siguientes en campañas y diagramas de flujo (sujetas a los permisos de seguridad).

- creación de una campaña
- descubrimiento (lista campañas según diversos criterios)
- creación, lectura y actualización de atributos (mediante las API de atributos)
- detención de una ejecución de diagrama de flujo

Las campañas tienen un número de atributos estándar asociados que las API exponen. El cliente puede ampliar esta lista a voluntad añadiendo atributos personalizados (consulte las API de atributos).

Los atributos de campaña estándar se listan a continuación:

- *uacName*: nombre de campaña (no se garantiza que sea exclusivo).
- *uacDescription*: cadena opcional que describe la campaña.

- *uacCampaignCode*: código de cadena que identifica de forma exclusiva la campaña. Normalmente Campaign lo genera automáticamente, pero también lo puede proporcionar el cliente.
- *uacCreateDate*: un calendario que indica la fecha y la hora en que el servidor ha creado la campaña.
- *uacUpdateDate*: un calendario que indica la fecha y la hora en que el servidor ha actualizado la campaña.
- *uacInitiative*: cadena opcional que define la iniciativa de la campaña.
- *uacObjectives*: cadena opcional que identifica los objetivos de la campaña.
- *uacStartDate*: un calendario opcional que proporciona la fecha y la hora en que el servidor ha iniciado la campaña, o en que se ha planificado que se inicie.
- *uacEndDate*: igual que *uacStartDate*, pero define la fecha y hora en que la campaña se ha completado o está planificado que se complete. Debe ser posterior a *uacStartDate*.
- *uacLastRunDate*: calendario opcional que indica la fecha y hora de la última ejecución (o nulo) de cualquier diagrama de flujo vinculado a la campaña.
- *uacExternalLinkOwner*: una cadena opcional que define el nombre del propietario de un enlace externo (consulte el atributo uacExternalLinkReference). Solo para uso de IBM ; debe ser uno de los siguientes:
  - "Plan" (conocido ahora como IBM Marketing Operations)
  - "Collaborate" (conocido ahora como IBM Distributed Marketing)
- *uacExternalLinkId*: un identificador numérico de base de datos opcional que asigna otra aplicación de IBM a un objeto enlazado a esta campaña. Solo para uso de IBM : vea también el atributo *uacExternalLinkOwner*.

generateCampaignCode

```
String generateCampaignCode(String userCredential,
    String partitionName,
    Locale requestedLocale);
```

Genere un nuevo código de campaña.

Se garantiza que este código es exclusivo y diferente del valor devuelto por una llamada anterior o futura a este método, el método `createCampaign()` o el valor generado para una campaña creada mediante la GUI de IBM Campaign.

Nota: El uso de este método es opcional, ya que la API `createCampaign()` generará un código de campaña para el cliente si no se suministra uno.

Consulte `createCampaign()`.

Parámetros

userCredential: la credencial de usuario cliente.

requestedLocale: entorno local opcional que se utiliza para esta solicitud.

partitionName: nombre opcional de la partición de campaña que se va a utilizar. Si solo hay una partición en la instalación de Campaign, el argumento puede ser nulo.

Devoluciones

El código de campaña generado.

Errores

AuthorizationException, DataException

deleteCampaigns

```
public WSDeleteCampaignsResponse deleteCampaigns(String userCredential,
String partitionName, Locale requestedLocale, WSReference[] wsReferences)
 throws CampaignServicesException
```

Suprime las campañas especificadas del sistema.

Parámetros

userCredential: la credencial de usuario cliente.

partitionName: nombre opcional de la partición de campaña que se va a utilizar.

requestedLocale: entorno local opcional que se utiliza para esta solicitud.

wsReference: referencias de campañas para suprimir.

Devoluciones

Devuelve el objeto de tipo WSDeleteCampaignsResponse.

Errores

Se emite CampaignServicesException si la campaña no existe, la referencia no es válida o no se proporcionan referencias.

createCampaign

```
CampaignInfo createCampaign(String userCredential,
        String partitionName,
        Locale requestedLocale,
        String securityPolicyName,
        String name, Attribute[] attributes)
    throws InvalidFolderException, AttributeNotFoundException,
        InvalidAttributeException;
```

Cree una nueva campaña para el cliente, la partición y securityPolicyName, aplicando los atributos especificados. Todas las campañas creadas por esta API estarán ubicadas en la carpeta raíz.

Parámetros

userCredential: la credencial de usuario cliente.

requestedLocale: entorno local opcional que se utiliza para esta solicitud.

partitionName: nombre opcional de la partición de campaña que se va a utilizar.

securityPolicyName: nombre opcional de la política de seguridad de la campaña que desea utilizar para crear la campaña. Todas las operaciones subsiguientes de esta campaña utilizarán esta política. Si no se ha definido, se utilizará la política global.

nombre: el nombre que se va a asignar a la instancia de la campaña nueva (su atributo "*uacName*").

atributos: una matriz opcional de atributos de inicialización; los atributos suministrados sobrescriben los valores predeterminados de la campaña; el resto permanecerán intactos. Por ejemplo, si se proporciona un atributo *uacCampaignCode*, se utilizará en lugar de uno generado automáticamente. Es responsabilidad del cliente determinar los atributos necesarios por la campaña, sus tipos, etc.

Se emite una *AttributeNotFoundException* si uno o varios de los atributos denominados no existen o *InvalidAttributeException* si un valor de atributo no es válido (como un tipo de datos incorrecto).

Devoluciones

Una única instancia de CampaignInfo para la campaña creada.

Errores

InvalidAttributeException, AttributeNotFoundException

AuthorizationException, DataException

listCampaignsByPage

```
List<CampaignInfo>
    listCampaignsByPage(String userCredential, String partitionName,
        Locale requestedLocale, Attribute[] attributes,
        long pageOffset, int pageSize)
    throws AttributeNotFoundException, InvalidAttributeException,
        RangeException;
```

Enumere una "página" de campañas que coincida con los valores de atributos opcionales, empezando por el desplazamiento de página especificado. Las carpetas se ignoran.

Una vez recuperadas, las *CampaignInfo* devueltas se pueden utilizar tal cual, por ejemplo, para visualizar una lista de resumen, o los métodos de atributos se pueden utilizar para captar o actualizar los atributos de la campaña.

Esta API no mantiene ningún estado, por lo que se puede llamar en cualquier orden.

Parámetros

userCredential: la credencial de usuario cliente.

requestedLocale: entorno local opcional que se utiliza para esta solicitud.

partitionName: nombre opcional de partición de campaña para utilizar.

attributes: matriz de atributos opcionales que deben coincidir; el nombre del atributo, el tipo de datos y los valores se utilizan para determinar la coincidencia; si el atributo soporta matrices, todos los valores especificados deben coincidir. El operador de coincidencia implícito es Y, por lo que solo se devolverán las campañas que coincidan con todos los valores de atributos suministrados.

Se emite *AttributeNotFoundException* si un nombre de atributo no existe o *InvalidAttributeException* si uno o varios de los atributos suministrados no son válidos.

pageOffset: desplazamiento inicial de todas las campañas posibles en el que debe empezar la enumeración (con valor cero). Por ejemplo, si la enumeración coincide

con 1000 campañas y este valor se define como 10, la página empezaría en el componente número 11. Se emite una *RangeException* si el desplazamiento suministrado está fuera de rango.

pageSize: número máximo de campañas coincidentes que se deben devolver para la página (no puede exceder de 500).

Devoluciones

Una lista de tipo de cero o más instancias de derivador de datos *CampaignInfo*, una por cada campaña con coincidencia en la página.

Errores

AttributeNotFoundException, InvalidAttributeException, RangeException

InvalidExecutionContextException, AuthorizationException

stopFlowchart

```
stopFlowchart(int pid, int runid)
```

Esta API detiene la ejecución de un diagrama de flujo. Para una configuración de un solo escucha, una ejecución de diagrama de flujo se puede identificar de forma exclusiva mediante el PID que está asociado a la ejecución del diagrama de flujo. El PID indica el ID de proceso del proceso `unica_acsvr`. Si se han configurado varios escuchas de IBM Campaign, debe incluir el ID de ejecución que está asociado a la ejecución del diagrama de flujo y también el PID.

Parámetros

pid: ID de proceso del proceso `unica_acsvr` asociado a una ejecución de diagrama de flujo.

runid: ID de ejecución asociado a una ejecución de diagrama de flujo. Parámetro necesario para una configuración de escucha en clúster. El parámetro opcional si se ha configurado un único escucha.

Devoluciones

Ninguno

Errores

Ninguno

Métodos de la API SOAP: celdas objetivo

Las celdas objetivo son una abstracción para algunos subconjuntos conocidos de resultados de campañas que gestiona IBM Campaign como una hoja de cálculo de celdas objetivo (TCS). Las celdas objetivo pueden ser globales para una campaña o estar asociadas a un diagrama de flujo de campaña en particular.

La API SOAP de IBM Campaign da soporte a las siguientes operaciones en celdas objetivo:

- Creación de una o más celdas objetivo globales nuevas
- Actualización colectiva de una o varias celdas objetivo existentes
- Descubrimiento (listado de celdas objetivo)

- creación, lectura y actualización de atributos (mediante las API de atributos)
- Supresión de una celda objetivo existente
- Captación de resultados de ejecución asociados a una o varias celdas

Las celdas objetivo tienen un número de atributos estándar asociados que la API expone. El cliente puede ampliar esta lista a voluntad añadiendo definiciones de metadatos de atributos personalizados (consulte las API Attributes). Todos los metadatos de atributo se puede considerar como una columna de la TCS; el diseño de la hoja de cálculo depende del cliente.

Los atributos de celda objetivo estándar son:

- *uacName*: nombre de celda.
- *uacDescription*: cadena opcional que describe el diagrama de flujo.
- *uacCellCode*: cadena de código que identifica de forma exclusiva la celda. Normalmente Campaign lo genera automáticamente, pero también lo puede proporcionar el cliente.
- *uacCreateDate*: instancia de calendario que proporciona la fecha y la hora en que el servidor ha creado la celda.
- *uacUpdateDate*: instancia de calendario que define cuándo el servidor ha actualizado la celda por última vez.
- *uacIsControl*: booleano que indica si es una celda de control (true) o no (false). Otras celdas pueden hacer referencia a esta celda como celda de control (consulte *uacControlCell*).
- *uacControlCell*: referencia opcional de la celda de control (no permitido si es una celda de control). Consulte el atributo *uaclsControl*.
- *uacIsApproved*: booleano que indica si la celda se ha aprobado (true) o no (false).
- *uacIsReadOnly*: booleano que indica si la celda es de solo lectura (true) o no (false).
- *uacDisplayOrder*: entero que proporciona el orden de esta celda (fila) en relación a otras de la hoja de cálculo de celdas objetivo.
- *uacIsTopDown*: booleano que indica si la celda es de arriba a abajo.
- *uacAssignedOffers*: matriz opcional de una o más referencias de ofertas o listas de ofertas asignadas a esta celda (no permitido si es una celda de control).
- *uacFlowchartName*: nombre opcional de diagrama de flujo al que está enlazada esta celda (de sólo lectura, debe establecerse mediante la GUI de IBM Campaign; no está permitido si es una celda de control).
- *uacFlowchartId*: identificador de base de datos opcional para el diagrama de flujo al que está enlazada esta celda (de solo lectura como antes).

createTargetCell

```
TargetCellInfo
    createTargetCell(String userCredential, String partitionName,
        Locale requestedLocale,
        Reference campaignReference,
        Attribute[] attributes)
    throws InvalidComponentException, CompositeException;
```

Cree una nueva fila de celdas objetivo específica de la campaña, aplicando los atributos por celdas especificados y la información de usuario.

Los atributos especificados pueden ser estándar o personalizados; sin embargo, si son personalizados, deben existir las definiciones de metadatos de atributos globales correspondientes.

Una vez creada la celda objetivo, los valores de atributos se pueden cambiar utilizando las API de atributos.

Consulte `listTargetCells()`, `bulkCreateTargetCells()`.

Consulte `createAttributeMetadata()`, `listAttributeMetata()`, `getAttributesByName()`

Parámetros

userCredential: la credencial de usuario cliente.

requestedLocale: entorno local opcional que se utiliza para esta solicitud.

partitionName: nombre opcional de la partición de campaña que se va a utilizar.

campaignReference: la referencia de la campaña que contiene la hoja de cálculo de celdas objetivo que se deben actualizar. Se acumula una *InvalidComponentException* si la campaña no existe o si la referencia no es válida.

atributos: matriz opcional de atributos de TCS para la nueva celda. Cada elemento de atributo suministrado sobrescribirá a los valores predeterminados del correspondiente atributo de celda; el resto permanecen intactos. Es responsabilidad del cliente determinar qué atributos necesita la celda, sus tipos, etc. Se acumula una *InvalidAttributeException* si hay un problema con un atributo especificado.

Si se acumulan excepciones, este método emitirá una *CompositeException* y todas las creaciones se desharán. La lista de excepciones de causas incluye una excepción para cada atributo que ha causado el error e incluye un índice numérico en lugar de la *referencia*, el nombre del atributo y, normalmente, el valor problemático. La lista de causas estará ordenada como en la entrada *attributeList*.

Devoluciones

Un derivador de datos *TargetCellInfo* para la celda de TCS creada.

Errores

InvalidComponentException, CompositeException

AuthorizationException, DataException

bulkCreateTargetCells

```
List<TargetCellInfo>
        bulkCreateTargetCells(String userCredential,
        String partitionName,
        Locale requestedLocale,
        Reference campaignReference,
        List<Attribute[]> attributesList)
    throws InvalidComponentException, CompositeException;
```

Cree muchas filas de celdas objetivo específicas de la campaña a la vez, aplicando los atributos por celda especificados y la información de usuario.

Los atributos especificados pueden ser estándar o personalizados; sin embargo, si son personalizados, deben existir las definiciones de metadatos de atributos globales correspondientes.

Una vez creada la celda objetivo, los valores de atributos se pueden cambiar utilizando las API de atributos.

Consulte `listTargetCells()`.

Consulte `createAttributeMetadata()`, `listAttributeMetata()`, `getAttributesByName()`

Parámetros

userCredential: la credencial de usuario cliente.

requestedLocale: entorno local opcional que se utiliza para esta solicitud.

partitionName: nombre opcional de la partición de campaña que se va a utilizar.

campaignReference: la referencia de la campaña que contiene la hoja de cálculo de celdas objetivo que se deben actualizar. Se acumula una *InvalidComponentException* si la campaña no existe o si la referencia no es válida.

attributeList: lista opcional de matrices de atributos por celda, una para cada fila de celdas objetivo que desea crear. Los atributos suministrados para un elemento de lista en particular sobrescriben a los valores predeterminados del correspondiente atributo de celda; el resto permanecen intactos. Es responsabilidad del cliente determinar qué atributos necesita la celda, sus tipos, etc. Se acumula una *InvalidAttributeException* si hay un problema con un atributo especificado.

Si se acumulan excepciones, este método emitirá una *CompositeException* y todas las creaciones se desharán. La lista de excepciones de las causas incluye una excepción para cada atributo que ha generado el error e incluye un índice numérico en lugar de la *referencia* y el nombre del atributo, etc. La lista de causas se ordena igual que la *attributeList* de entrada.

Devoluciones

Una lista de derivadores de datos de *TargetCellInfo*, uno por cara instancia creada, ordenados según el orden de los elementos del parámetro *attributesList* de entrada.

Errores

InvalidComponentException, CompositeException

AuthorizationException, DataException

listTargetCells

```
List<TargetCellInfo>
        listTargetCells(String userCredential,
        Reference campaignReference, Locale requestedLocale,
        Attribute[] attributes)
    throws InvalidComponentException, InvalidAttributeException;
```

Lista información acerca de todas las celdas objetivo que existen actualmente que coinciden con los atributos especificados, ya sea para la campaña especificada o bien globalmente si no se especifica ninguna campaña.

Consulte `getAttributeMetadata()`, `getAttributesByName()`.

Parámetros

userCredential: la credencial de usuario cliente.

requestedLocale: entorno local opcional que se utiliza para esta solicitud.

partitionName: nombre opcional de la partición de campaña que se va a utilizar.

campaignReference: referencia de la campaña padre. Se emite *InvalidComponentException* si la campaña no existe o la referencia no es válida.

attributes: matriz opcional de atributos que deben coincidir. El operador de coincidencia implícito es Y, por lo que solo se devolverán las celdas que coincidan con todos los valores de atributos suministrados.

Se emite *InvalidAttributeException* si uno o varios de los atributos especificados no es válido.

Devoluciones

Devuelve una lista de cero o más instancias de *TargetCellInfo* para las celdas con coincidencia.

Errores

InvalidComponentException, InvalidAttributeException

AuthorizationException, DataException

bulkUpdateTargetCells

```
void bulkUpdateTargetCells(String userCredential,
        String partitionName,
        Locale requestedLocale,
        Map<Reference, Attribute[]> attributesMap)
    throws CompositeException;
```

Actualice los atributos de una o varias celdas objetivo.

La lógica de la actualización es la siguiente.

Para cada elemento en el *attributesMap* proporcionado, la clave de entrada es la referencia de la celda objetivo que se va a actualizar y el valor de entrada es una matriz de atributos de actualización para esa celda. Si la celda objetivo no existe, se acumula una *InvalidComponentException*.

Una vez localizada una celda objetivo, haga lo siguiente para cada atributo especificado:

1. Si el nombre de atributo coincide con un atributo existente, intente sobrescribir su campo de valores con el campo de valores suministrado.
2. Si el tipo de valor o algún otro aspecto de la definición de metadatos del atributo no se cumple, o uno o varios de los valores suministrados no son válidos, están fuera de rango, etc., se acumula una *InvalidAttributeException*.
3. Si no, se acumula una *AttributeNotFoundException* si el atributo denominado no existe.

Si se acumulan excepciones, este método emitirá una *CompositeException* y todas las actualizaciones se desharán. La lista de excepciones de las causas incluye las

excepciones que se han listado anteriormente. Para cada atributo que ha causado el error, se registrarán la referencia y el nombre de atributo.

En todos los casos, la operación de actualización de atributos está sujeta a las restricciones y la validación de seguridad normales. Es responsabilidad del cliente determinar qué atributos son necesarios para una instancia de componente en particular, los tipos correctos, etc.

Parámetros

userCredential: la credencial de usuario cliente.

requestedLocale: entorno local opcional que se utiliza para esta solicitud.

partitionName: nombre opcional de la partición de campaña que se va a utilizar.

attributesMap: correlación de celdas objetivo que se van a actualizar; la clave de entrada es la referencia de la celda que se va a actualizar y el valor de entrada es una matriz de los atributos de actualización. El nombre del atributo se utiliza para localizar el atributo que se debe actualizar y los valores nuevos de atributo se utilizan para actualizar el valor existente del atributo como un objeto único del tipo apropiado o una matriz, si es aplicable. Consulte las excepciones anteriores.

Devoluciones

Ninguna.

Errores

ComponentException

AuthorizationException, DataException

getRunResultsByCell

```
List<RunResults>
    getRunResultsByCell(String userCredential, String partitionName,
        Locale requestedLocale,
        Reference[] cellReferences)
    throws InvalidComponentException;
```

Obtenga los resultados de la ejecución de una a o varias celdas objetivo, posiblemente para un diagrama de flujo que no se ha iniciado nunca o que todavía está en curso.

Parámetros

userCredential: la credencial de usuario cliente.

requestedLocale: entorno local opcional para utilizar para esta solicitud.

partitionName: nombre opcional de la partición de campaña que se va a utilizar.

cellReferences: matriz de referencias de las celdas objetivo cuyos resultados de ejecución se desean. Se emite *InvalidComponentException* si una o varias referencias de celda no son válidas o hacen referencia a una celda que no existe.

Devoluciones

Devuelve una lista de tipo de resultados de ejecución para las celdas denominadas, ordenadas según la matriz de referencias de entrada.

Cada estado de ejecución será RUNNING si el cuadro de proceso de diagrama de flujo sigue en ejecución, FAILED si la ejecución ha fallado por alguna razón o NOT_STARTED si el cuadro de proceso no se ha iniciado. También se proporcionan los detalles de estado.

Errores

InvalidComponentException

AuthorizationException, DataException

bulkDeleteTargetCells

```
void bulkDeleteTargetCells(String userCredential,
        String partitionName,
        Locale requestedLocale,
        Reference[] cellReferences)
    throws CompositeException;
```

Suprime una o varias celdas objetivo existentes y todos los componentes dependientes (es decir, enlace de diagramas de flujo, atributos, etc.)

Parámetros

userCredential: la credencial de usuario cliente.

requestedLocale: entorno local opcional que se utiliza para esta solicitud.

partitionName: nombre opcional de la partición de campaña que se va a utilizar.

cellReferences: una matriz o una o más referencias de celdas que se van a suprimir. Se acumula una *InvalidComponentException* si hay un problema con una de las referencias especificadas o si no existe una celda.

Si las excepciones se acumulan, este método emitirá una *CompositeException* y todas las supresiones se desharán. La lista de excepciones de las causas incluye las excepciones que se han listado anteriormente. Se registrará la referencia para cada celda que ha causado el error.

Devoluciones

Ninguna.

Errores

CompositeException

AuthorizationException, DataException

updateTemplateAttributes

```
updateTemplateAttributes
(String userCredential, String partitionName, Locale requestedLocale,
WSReference wsReference, boolean allowCreate,
boolean clearExisting, WSAttribute[] wsStaticAttributes,
WSAttribute[] wsHiddenAttributes,WSAttribute[] wsParametricAttributes)
 throws CampaignServicesException
```

Actualiza los atributos de las plantillas especificadas

Parámetros

userCredential: la credencial de usuario cliente.

partitionName: nombre opcional de la partición de campaña que se va a utilizar.

requestedLocale: entorno local opcional que se utiliza para esta solicitud.

wsCampaignReference: referencia de la campaña padre.

allowCreate: no se utiliza ahora.

clearExisting: distintivo que, si se establece en true, hace que se borren todos los valores anteriores de la plantilla que no se envían en la solicitud.

wsStaticAttributes: lista de atributos estáticos de la plantilla.

wsHiddenAttributes: lista de atributos ocultos de la plantilla.

wsParametricAttributes: lista de atributos paramétricos de la plantilla.

Devoluciones

Ninguna.

Errores

Se emite CampaignServicesException si la plantilla de oferta no existe, la referencia no es válida o no se proporcionan referencias.

listBottomUpTargetCells

```
public List <WSTargetCellDetails>
listBottomUpTargetCells(String userCredential, String partitionName,
Locale requestedLocale, WSReference wsCampaignReference)
 throws CampaignServicesException
```

Lista información acerca de todas las celdas objetivo descendentes que existen actualmente para la campaña especificada.

Parámetros

userCredential: la credencial de usuario cliente.

partitionName: nombre opcional de la partición de campaña que se va a utilizar.

requestedLocale: entorno local opcional que se utiliza para esta solicitud.

wsCampaignReference: referencia de la campaña padre.

Devoluciones

Devuelve una lista de cero o más instancias de WSTargetCellDetail para las celdas coincidentes.

Errores

Se emite CampaignServicesException si la campaña no existe o la referencia no es válida.

Métodos de la API SOAP: Análisis

La API SOAP de IBM Campaign da soporte a la recuperación métricas simples desde IBM Campaign.

getCampaignMetrics

```
MetricsInfo getCampaignMetrics(String userCredential,
        String partitionName,
        Locale requestedLocale,
        Reference campaignReference)
    throws InvalidComponentException;
```

Captura las métricas de la campaña especificada.

Parámetros

userCredential: la credencial de usuario cliente.

requestedLocale: entorno local opcional que se utiliza para esta solicitud.

partitionName: nombre opcional de la partición de campaña que se va a utilizar.

campaignReference: referencia de la campaña padre. Se emite *InvalidComponentException* si hay un problema con una referencia de campaña o la campaña no existe.

Devoluciones

Devuelve una instancia de *MetricsInfo* para la campaña.

Errores

InvalidComponentException

AuthorizationException, DataException

Métodos de la API SOAP: ofertas, listas de ofertas, plantillas de ofertas

La API SOAP de IBM Campaign da soporte a las siguientes operaciones relacionadas con las ofertas.

- descubrimiento: listado por carpeta (ofertas, listas de ofertas y subcarpetas), atributo (ofertas y plantillas de ofertas), o valor de búsqueda (ofertas)
- validación
- recuperación de información (atributos de recuperación para una oferta o plantilla de oferta específica)
- creación, edición, retiro y supresión de ofertas

Las ofertas tienen varios atributos estándar asociados. El cliente puede ampliar esta lista añadiendo definiciones de metadatos de atributos personalizados (consulte las API de atributos).

Los atributos de ofertas estándar son:

- *uacName*: nombre de oferta.
- *uacDescription*: cadena opcional que describe la oferta.
- *uacOfferCode*: cadena de código que identifica de forma exclusiva la oferta. Normalmente lo genera IBM Campaign, pero también lo puede proporcionar el cliente.
- *uacCreateDate*: Una instancia de calendario que indica la fecha y hora en que el servidor ha creado la oferta.
- *uacUpdateDate*: Una instancia de calendario que indica la hora en que el servidor ha actualizado la oferta por última vez.

Las plantillas de ofertas también tienen atributos estándar y personalizados. Los atributos de plantillas de ofertas estándar son:

- *uacName*: nombre de plantilla de oferta.
- *uacDescription*: cadena opcional que describe la plantilla de oferta.
- *uacCreateDate*: Una instancia de calendario que indica la fecha y hora en que el servidor ha creado la plantilla de oferta.
- *uacUpdateDate*: Una instancia de calendario que indica la hora en que el servidor ha actualizado la plantilla de oferta por última vez.

listOffersAndFolders

```
List<WSComponentOrFolderInfo>
    listOffersAndFolders(String userCredential, String partitionName,
        Locale requestedLocale,
        WSReference parentReference)
    throws CampaignServicesException;
```

Liste todas las ofertas, las listas de ofertas y las carpetas bajo la carpeta padre opcional.

Una vez recuperadas, las instancias *WSComponentOrFolderInfo* devueltas se pueden utilizar tal cual, por ejemplo para visualizar el siguiente nivel de la carpeta de jerarquías, las API de atributos se pueden captar o actualizar las ofertas contenidas.

Parámetro

userCredential: la credencial de usuario cliente.

requestedLocale: entorno local opcional para utilizar para esta solicitud.

partitionName: nombre opcional de partición de campaña para utilizar.

parentReference: referencia opcional de la carpeta padre para listar. Solo se enumerarán las ofertas hijo, las listas de ofertas y las carpetas de esta carpeta padre, por lo que se necesitarán llamadas sucesivas a esta API para navegar por toda la jerarquía de la carpeta (sin embargo, normalmente será muy poco profunda). Si no se suministra ningún padre, se devuelven todos los componentes y las carpetas bajo la raíz.

Se emite *InvalidFolderException* si hay un problema con la referencia de carpeta padre especificada.

Una *Lista* de tipo de cero o más instancias de derivador de datos *WSComponentOrFolderInfo*, una por cada componente o carpeta con coincidencia.

Errores

InvalidFolderException

InvalidExecutionContextException, AuthorizationException

searchOffersBasic

```
List<WSOfferInfo>
    searchOffersBasic(String userCredential, Locale requestedLocale,
        String partitionName, long folderID,
        String searchCriteria, boolean includeRetired,
        int pageOffset, int pageSize)
    throws CampaignServicesException;
```

Enumere una "página" de ofertas que contengan los criterios de búsqueda proporcionados en los campos de nombre, descripción, creada por o código de oferta, empezando por el desplazamiento de página especificado. La búsqueda se basa en la entrada de carpeta opcional. (Si se proporciona un folderID de 0, la carpeta de ofertas raíz se utiliza de forma predeterminada). Se devuelven las coincidencias en función de una coincidencia "contiene" para la cadena de búsqueda.

Una vez recuperadas, las *WSOfferInfo* devueltas se pueden utilizar tal cual, por ejemplo para visualizar una lista de resumen, o los métodos de atributos se pueden utilizar para captar o actualizar los atributos de la oferta.

Esta API no mantiene ningún estado, por lo que se puede llamar en cualquier orden.

Parámetros

userCredential: la credencial de usuario cliente.

requestedLocale: entorno local opcional para utilizar para esta solicitud.

partitionName: nombre opcional de partición de campaña para utilizar.

folderID: ID de la carpeta de ofertas en la que se debe buscar; si se especifica un folderID de 0, se buscará en la carpeta raíz.

searchCriteria: frase de búsqueda.

includeRetired: valor booleano que especifica si los resultados de la búsqueda incluirán las ofertas retiradas. Los valores válidos son TRUE y FALSE, TRUE indica que se incluyen las ofertas retiradas y FALSE indica que las ofertas retiradas no se incluyen.

pageOffset: desplazamiento inicial de todos los componentes posibles en el que debe empezar la enumeración (con valor cero). Por ejemplo, si la enumeración coincide con 1000 ofertas y este valor se establece en 10, la página empezaría en el componente número 11. Se emite una RangeException si el desplazamiento suministrado está fuera de rango.

pageSize: número máximo de componentes coincidentes que se deben devolver para la página (no puede exceder de 500).

Devoluciones

Devuelve una lista de tipo de cero o más instancias de derivador de datos *Offer*, uno para cada oferta devuelta en la página.

Errores

RangeException

listOffersByPage

```
List<OfferInfo>
    listOffersByPage(String userCredential, String partitionName,
        Locale requestedLocale, Attribute[] attributes,
        long pageOffset, int pageSize)
    throws AttributeNotFoundException, InvalidAttributeException,
        RangeException;
```

Enumere una "página" de ofertas que coincidan con los valores de atributos opcionales, empezando por el desplazamiento de página especificado. Las carpetas se ignoran. Las coincidencias se devuelven en función de una coincidencia "similar" para las cadenas (donde la coincidencia se considera suficiente si una cadena contiene el valor consultado), y una coincidencia exacta para las fechas y números.

Una vez recuperadas, las *OfferInfo* devueltas se pueden utilizar tal cual, por ejemplo, para visualizar una lista de resumen, o los métodos de atributos se pueden utilizar para captar o actualizar los atributos de la campaña.

Esta API no mantiene ningún estado, por lo que se puede llamar en cualquier orden.

Parámetros

userCredential: la credencial de usuario cliente.

requestedLocale: entorno local opcional que se utiliza para esta solicitud.

partitionName: nombre opcional de la partición de campaña que se va a utilizar.

attributes: matriz de atributos opcionales que deben coincidir; el nombre del atributo, el tipo de datos y los valores se utilizan para determinar la coincidencia; si el atributo soporta matrices, todos los valores especificados deben coincidir. El operador de coincidencia implícito es O, por lo que se devolverán todos los componentes que coincidan con alguno de los valores de atributos suministrados.

Se emite *AttributeNotFoundException* si un nombre de atributo no existe o InvalidAttributeException si uno o varios de los atributos suministrados no es válido.

pageOffset: el desplazamiento inicial de todos los componentes posibles en que debe empezar la enumeración (con valor cero). Por ejemplo, si la enumeración coincide con 1000 ofertas y este valor se establece en 10, la página empezaría en el componente número 11. Se emite una *RangeException* si el desplazamiento suministrado está fuera de rango.

pageSize: número máximo de componentes coincidentes que se deben devolver para la página (no puede exceder de 500).

Devoluciones

Una lista de tipo de una o más instancias de derivador de datos *OfferInfo*, una por cada componente con coincidencia en la página.

Errores

AttributeNotFoundException, InvalidAttributeException, RangeException

InvalidExecutionContextException, AuthorizationException

createSmartOfferList

```
public WSCreateSmartOfferListResponse createSmartOfferList
(String userCredential, String partitionName, Locale requestedLocale,
String name, String description,String policyName,
WSReference parentFolder,WSSmartListInfo offerListInfo,
WSApplicationTypeEnum createdBy,long creatorObjectId)
 throws CampaignServicesException
```

Crea una nueva lista de ofertas inteligentes.

Parámetros

userCredential: la credencial de usuario cliente.

partitionName: nombre opcional de la partición de campaña que se va a utilizar.

requestedLocale: entorno local opcional que se utiliza para esta solicitud.

nombre: nombre de la carpeta que se creará.

descripción: descripción de la carpeta nueva.

securityPolicyName: nombre de la política de seguridad que se utilizará.

parentFolder: ID de carpeta donde se debe crear la lista de ofertas.

offerListInfo: objeto de tipo WSSmartListInfo.

createdBy: (opcional) el objeto de tipo WSApplicationTypeEnum indica la aplicación que ha creado la carpeta. Los valores posibles son Campaign/Plan/Collaborate/eMessage. Si no se proporciona, se utiliza Campaign.

creatorObjectId: (opcional) lo utiliza el Plan (IBM Marketing Operations) para enlazar una carpeta de IBM Campaign con una carpeta de Marketing Operations.

Devoluciones

Devuelve un objeto de tipo WSCreateSmartOfferListResponse.

Errores

Se emite CampaignServicesException si el ID de parentFolder no es válido u offerListInfo no se proporciona.

Se emite CampaignServicesException si el nombre de lista está duplicado.

createStaticOfferList

```
public WSCreateStaticOfferListResponse createStaticOfferList
(String userCredential, String partitionName, Locale requestedLocale, String name,
String description,String policyName,WSReference parentFolder,WSReference[]
listMembers,WSApplicationTypeEnum createdBy,long creatorObjectId)
 throws CampaignServicesException
```

Crea una nueva lista de ofertas estáticas.

Parámetros

userCredential: la credencial de usuario cliente.

partitionName: nombre opcional de la partición de campaña que se va a utilizar.

requestedLocale: entorno local opcional que se utiliza para esta solicitud.

nombre: nombre de la carpeta que se creará.

descripción: descripción de la carpeta nueva.

securityPolicyName: nombre de la política de seguridad que se utilizará.

parentFolder: ID de carpeta donde se debe crear la lista de ofertas.

listMembers: referencias a ofertas que se incluirán en la lita de ofertas.

createdBy: (opcional) el objeto de tipo WSApplicationTypeEnum indica la aplicación que ha creado la carpeta. Los valores posibles son Campaign/Plan/Collaborate/eMessage. Si no se proporciona, se utiliza Campaign.

creatorObjectId: (opcional) lo utiliza el Plan (IBM Marketing Operations) para enlazar una carpeta de IBM Campaign con una carpeta de Marketing Operations.

Devoluciones

Devuelve el objeto de tipo WSCreateStaticOfferListResponse.

Errores

Se emite CampaignServicesException si el ID de parentFolder no es válido o listMembers no son válidos.

Se emite CampaignServicesException si el nombre de lista está duplicado.

getOffers

```
public WSGetOffersResponse getOffers
(String userCredential, String partitionName,
Locale requestedLocale,WSReference[] wsReferences)
 throws CampaignServicesException
```

Lista detalles de ofertas tal como se especifica en la solicitud.

Parámetros

userCredential: la credencial de usuario cliente.

partitionName: nombre opcional de la partición de campaña que se va a utilizar.

requestedLocale: entorno local opcional que se utiliza para esta solicitud.

wsCampaignReference: referencia de la campaña padre.

Devoluciones

Devuelve el objeto de tipo WSGetOffersResponse.

Errores

Se emite CampaignServicesException si la oferta no existe, la referencia no es válida o no se proporcionan referencias.

validateOffers

```
List<OfferValidationInfo>
    validateOffers(String userCredential, String partitionName,
    Locale requestedLocale,
    OfferCodeOrName[] codeOrNames);
```

Valide los códigos de ofertas o los nombres de la lista de ofertas suministrados y la información de validación de retorno para cada uno de ellos. La "validación" consiste en la comprobación de si existe una y solo una oferta o lista de ofertas que coincide en la base de datos.

El objeto OfferValidationInfo contiene un mensaje de error en lugar de la información de oferta si se encuentran cero ofertas o listas de ofertas que coinciden con el código o nombre correspondiente. También se devuelve un error en lugar de una coincidencia si el código o el nombre dado coincide con varias ofertas y listas de ofertas. La lista se devuelve en el mismo orden dado. Los códigos de oferta y los nombres de la lista de ofertas se validan en función de la coincidencia exacta con ofertas.

Parámetros

userCredential: la credencial de usuario cliente.

requestedLocale: entorno local opcional que se utiliza para esta solicitud.

partitionName: nombre opcional de la partición de campaña que se va a utilizar.

codeOrNames: matriz de todos los códigos de oferta o nombres de la lista de ofertas que desea validar.

Nota: No se emiten excepciones por este método; en su lugar se devuelve información de validación para todos los códigos o nombres suministrados.

Devoluciones

Una lista de tipo de cero o más instancias de derivador de datos *OfferValidationInfo*.

Errores

Ninguna.

editOfferList

```
public WSEditOfferListResponse editOfferList(String userCredential,
String partitionName, Locale requestedLocale, WSReference listReference,
boolean isSmartList,String name, String description,
WSReference[] listMembers,WSSmartListInfo offerListInfo,
Long creatorObjectId, boolean clearExisting)
throws CampaignServicesException
```

Actualiza los detalles de la lista de ofertas estáticas e inteligentes.

Parámetros

userCredential: la credencial de usuario cliente.

partitionName: nombre opcional de la partición de campaña que se va a utilizar.

requestedLocale: entorno local opcional que se utiliza para esta solicitud.

listReference: referencia a lista de ofertas.

isSmartList: distintivo que indica si la lista es inteligente o estática.

nombre: nombre de la carpeta que se creará.

descripción: descripción de la carpeta nueva.

listMembers: referencias a ofertas que se incluirán en la lita de ofertas.

offerListInfo: objeto de tipo WSSmartListInfo.

creatorObjectId: (opcional) lo utiliza el Plan (IBM Marketing Operations) para enlazar una carpeta de IBM Campaign con una carpeta de Marketing Operations.

clearExisting: distintivo que indica si se debe borrar la información existente. Si es true, los miembros existentes de la lista se borrarán antes de añadir nuevos miembros. Si es false, los miembros nuevos se añadirán a los miembros existentes.

Devoluciones

Devuelve un objeto de tipo WSEditOfferListResponse.

Errores

Se emite CampaignServicesException si el ID de parentFolder no es válido, si no se proporciona siofferListInfo o si listMembers no son válidos.

Se emite CampaignServicesException si el nombre de lista está duplicado.

createOffer

```
OfferInfo createOffer(String userCredential, String partitionName,
        Locale requestedLocale,
        String securityPolicyName,
        String name, String templateName,
        Attribute[] attributes)
    throws InvalidFolderException, AttributeNotFoundException,
        InvalidAttributeException;
```

```
public WSOfferInfo createOffer(String authorizationLoginName, String
        partitionName, Locale requestedLocale, String
        securityPolicyName, String name, long folderID,
        String templateName, WSAttribute[] wsAttributes)
    throws CampaignServicesException;
```

Cree una nueva oferta para el cliente, aplicando los atributos especificados.

Parámetros

authorizationLoginName: nombre de usuario que crea la oferta. Se debe otorgar a los usuarios el permiso Añadir ofertas para utilizar este método.

partitionName: nombre opcional de partición de campaña para utilizar.

requestedLocale: entorno local opcional para utilizar para esta solicitud.

securityPolicyName: nombre opcional de la política de seguridad de campaña para utilizar para crear la oferta. Todas las operaciones subsiguientes de esta oferta utilizarán esta política. Si no se ha definido, se utilizará la política *Global*.

nombre: nombre que desea asignar a la nueva instancia de oferta (su atributo *uacName*).

folderID: ID de la carpeta de ofertas donde se creará la oferta. Se comprobará si este ID es correcto y se emitirá una excepción si el ID no es válido.

templateName: nombre necesario (exclusivo) de una plantilla de oferta existente que desea utilizar para la nueva oferta.

wsAttributes: una matriz de atributos de inicialización; los atributos suministrados sobrescribirán los valores predeterminados de la oferta; otros se dejarán intactos. Por ejemplo, si se proporciona un atributo *uacOfferCode*, se utilizará en lugar de uno generado automáticamente. Es responsabilidad del cliente determinar los atributos que necesita la oferta, sus tipos, etc.

Se emite una *CampaignServicesException* si se produce una de las condiciones siguientes:

- El parámetro folderID no es válido (no existe o no es del tipo de oferta).
- El usuario no está autorizado a realizar esta operación.
- Se suministran atributos no válidos en wsAttributes.
- Se producen otras excepciones de tiempo de ejecución.

Devoluciones

Una única instancia de *OfferInfo* para la oferta creada.

Errores

CampaignServicesException

retireOffers

```
void retireOffers(String userCredential, String partitionName,
        Locale requestedLocale, WSReference[] references)
    throws CampaignServicesException;
```

Retira una o varias ofertas existentes.

Parámetros

userCredential: la credencial de usuario cliente.

requestedLocale: entorno local opcional para utilizar para esta solicitud.

partitionName: nombre opcional de partición de campaña para utilizar.

references: matriz de referencias de las ofertas que desea retirar. Se emite *InvalidComponentException* si hay un problema con una referencia en particular o una oferta no existe.

Devoluciones

Ninguna.

Errores

InvalidComponentException

AuthorizationException, DataException

deleteOffers

```
void deleteOffers(String userCredential, String partitionName,
        Locale requestedLocale, WSReference[] references)
    throws CampaignServicesException;
```

Suprime una o varias ofertas existentes.

Parámetros

userCredential: la credencial de usuario cliente.

requestedLocale: entorno local opcional para utilizar para esta solicitud.

partitionName: nombre opcional de partición de campaña para utilizar.

referencia: una matriz de referencias de las ofertas que desea suprimir. Se emite una *InvalidComponentException* si hay un problema con una referencia especificada, o si una oferta no existe.

Devoluciones

Ninguna.

Errores

InvalidComponentException

AuthorizationException, DataException

deleteOffersAndLists

```
public WSDeleteOffersAndListsResponse deleteOffersAndLists
(String userCredential, String partitionName, Locale requestedLocale,
 WSReference[] offers)
 throws CampaignServicesException
```

Suprime las ofertas y listas especificadas.

Parámetros

userCredential: la credencial de usuario cliente.

partitionName: nombre opcional de la partición de campaña que se va a utilizar.

requestedLocale: entorno local opcional que se utiliza para esta solicitud.

offers: matriz de referencias de oferta o lista de oferta.

Devoluciones

Devuelve el objeto de tipo WSGetOfferListMembersResponse.

Errores

Se emite CampaignServicesException si el ID de oferta o ID de lista de ofertas no es válido.

listOfferTemplates

```
List<WSOfferTemplateInfo>
    listOfferTemplates(String userCredential, String partitionName,
        Locale requestedLocale)
    throws CampaignServicesException;
```

Lista de todas las plantillas de ofertas que el usuario tiene permisos para ver.

Una vez recuperadas, las instancias *WSOfferTemplateInfo* devueltas se pueden utilizar tal cual, o las API de atributos se pueden utilizar para captar o actualizar las ofertas contenidas.

Parámetro

userCredential: la credencial de usuario cliente.

requestedLocale: entorno local opcional para utilizar para esta solicitud.

partitionName: nombre opcional de partición de campaña para utilizar.

Devoluciones

Una lista de tipo de cero o varias instancias de derivador de datos *WSOfferTemplateInfo*, una para cada plantilla devuelta.

Errores

InvalidExecutionContextException, AuthorizationException

DataException

createTemplate

```
createTemplate(String userCredential, String partitionName, Locale requestedLocale,
 String name, String securityPolicyName ,WSAttribute[]
 wsStaticAttributes,WSAttribute[] wsHiddenAttributes,
WSAttribute[] wsParametricAttributes)
 throws CampaignServicesException
```

Crea una nueva plantilla de oferta.

Parámetros

userCredential: la credencial de usuario cliente.

partitionName: nombre opcional de la partición de campaña que se va a utilizar.

requestedLocale: entorno local opcional que se utiliza para esta solicitud.

nombre: nombre de la plantilla de oferta que se acaba de crear.

securityPolicyName: nombre de la política de seguridad que se utilizará.

wsStaticAttributes: lista de atributos estáticas en la plantilla.

wsHiddenAttributes: lista de atributos ocultos en la plantilla.

wsParametricAttributes: lista de atributos paramétricas en la plantilla.

Devoluciones

Devuelve el objeto de tipo WSCreateTemplateResponse.

Errores

Se emite CampaignServicesException si la plantilla de oferta no existe, la referencia no es válida o no se proporcionan referencias.

getOfferTemplate

```
public WSGetOfferTemplateResponse getOfferTemplate(String userCredential,
 String partitionName, Locale requestedLocale,WSReference[] wsReferences)
 throws CampaignServicesException
```

Lista detalles de plantillas de oferta tal como se especifica en referencias.

Parámetros

userCredential: la credencial de usuario cliente.

partitionName: nombre opcional de la partición de campaña que se va a utilizar.

requestedLocale: entorno local opcional que se utiliza para esta solicitud.

wsCampaignReference: referencia de la campaña padre.

Devoluciones

Devuelve el objeto de tipo WSGetOfferTemplateResponse.

Errores

Se emite CampaignServicesException si la plantilla de oferta no existe, la referencia no es válida o no se proporcionan referencias.

retireOfferTemplates

```
public WSGenerateOfferCodeResponse generateOfferCodes
(String userCredential, String partitionName,Locale requestedLocale,
String offerName, WSReference template)
 throws CampaignServicesException
```

Retira una o más plantillas de oferta especificadas.

Parámetros

userCredential: la credencial de usuario cliente.

partitionName: nombre opcional de la partición de campaña que se va a utilizar.

requestedLocale: entorno local opcional que se utiliza para esta solicitud.

wsCampaignReference: referencia de la campaña padre.

Devoluciones

Devuelve el objeto de tipo WSRetireOfferTemplatesResponse.

Errores

Se emite CampaignServicesException si la plantilla de oferta no existe, la referencia no es válida o no se proporcionan referencias.

getOffersAndListsByPage

```
public WSGetOffersAndListsByPageResponse getOffersAndListsByPage
(String userCredential, String partitionName, Locale requestedLocale,
 WSComponentTypeEnum type,
int pageSize, int pageOffset)
 throws CampaignServicesException
```

Lista ofertas o listas de ofertas por página.

Parámetros

userCredential: la credencial de usuario cliente.

partitionName: nombre opcional de la partición de campaña que se va a utilizar.

requestedLocale: entorno local opcional que se utiliza para esta solicitud.

type: indica si se solicitan ofertas o listas de ofertas.

pageSize: número máximo de componentes coincidentes que se deben devolver para la página.

pageOffset: el desplazamiento inicial de todos los componentes posibles en que debe empezar la enumeración (con valor cero). Por ejemplo, si la enumeración coincide con 1000 ofertas y este valor se establece en 10, la página empezaría en el componente número 11. Se emite una RangeException si el desplazamiento suministrado está fuera de rango.

Devoluciones

Devuelve el objeto de tipo WSGetOffersAndListsByPageResponse.

Errores

Ninguna.

bulkCreateOffers

```
WSOfferInfoStatus[] bulkCreateOffers(String authorizationLoginName,
        String partitionName, Locale requestedLocale,
        String securityPolicyName, String templateName, long folderID,
        WSBulkOfferInfo[] offers)
    throws CampaignServicesException;
```

Crea ofertas colectivas con los atributos para cada oferta que se especifica en el parámetro *offers*. Todas las ofertas se crean bajo el valor *folderID* especificado utilizando el valor de *templateName* especificado.

Parámetro

authorizationLoginName: la credencial de usuario cliente.

partitionName: nombre opcional de partición de campaña para utilizar.

requestedLocale: entorno local opcional para utilizar para esta solicitud.

securityPolicyName: nombre opcional de la política de seguridad de campaña para utilizar para crear la oferta. Si no está definido, se utiliza la política global.

templateName: nombre de laplantilla de oferta existente en el sistema. Todas las ofertas se crean utilizando esta plantilla.

folderID: el ID de la carpeta de ofertas donde se crean las ofertas. Este ID se valida y se lanza una excepción si el ID no es válido.

offers: una matriz de objetos *WSBulkOfferInfo* que define el nombre de la oferta y los atributos. Consulte el tipo de datos WSBulkOfferInfo si desea más detalles.

Devoluciones

Una matriz de las instancias *WSOfferInfoStatus* para cada oferta. Contiene información sobre el estado y la oferta. El estado indica si la creación de la oferta ha sido satisfactoria o no.

Errores

CampaignServicesException

getOfferListDetails

```
public WSGetOfferListDetailsResponse getOfferListDetails(String userCredential,
String partitionName, Locale requestedLocale, WSReference listReference)
 throws CampaignServicesException {
```

Lista detalles de la lista de ofertas especificada.

Parámetros

userCredential: la credencial de usuario cliente.

partitionName: nombre opcional de la partición de campaña que se va a utilizar.

requestedLocale: entorno local opcional que se utiliza para esta solicitud.

listReference: referencia a lista de ofertas.

Devoluciones

Devuelve el objeto de tipo WSGetOfferListDetailsResponse.

Errores

Se emite CampaignServicesException si las referencias de lista no son válidas.

getOfferListMembers

```
public WSGetOfferListMembersResponse getOfferListMembers
(String userCredential, String partitionName, Locale requestedLocale,
 WSReference listReference)
 throws CampaignServicesException {
```

Lista información de ofertas en la lista de ofertas especificada.

Parámetros

userCredential: la credencial de usuario cliente.

partitionName: nombre opcional de la partición de campaña que se va a utilizar.

requestedLocale: entorno local opcional que se utiliza para esta solicitud.

listReference: referencia a lista de ofertas.

Devoluciones

Devuelve el objeto de tipo WSDeleteOffersAndListsResponse.

Errores

Se emite CampaignServicesException si el ID de lista de ofertas no es válido.

getOffersByQuery

```
public WSGetOffersByQueryResponse getOffersByQuery(String user_credential,
 String partition_name, Locale locale, String query, Integer maxSize,
Boolean includeSubFolder, WSReference[] scopeFolders)
 throws CampaignServicesException
```

Lista ofertas que coinciden con las ofertas proporcionadas.

Parámetros

userCredential: la credencial de usuario cliente.

partitionName: nombre opcional de la partición de campaña que se va a utilizar.

requestedLocale: entorno local opcional que se utiliza para esta solicitud.

query: consulta para buscar ofertas. El formato de la consulta es el mismo que el utilizado en la lista de ofertas inteligentes.

maxSize: número máximo de registros para listar.

includeSubFolder: distintivo que indica si una subcarpeta debe estar incluida en la búsqueda.

scopeFolders: lista de referencias de carpeta en las que se buscarán ofertas.

Devoluciones

Devuelve el objeto de tipo WSGetOffersByQueryResponse.

Errores

Se emite CampaignServicesException si las referencias de carpeta no son válidas.

retireOfferLists

```
public void retireOfferLists(String credencial_usuario, String nombre_partición,
Locale entorno local, WSReference[] wsReferences)
 throws CampaignServicesException
```

Retira una o más listas de ofertas especificadas.

Parámetros

userCredential: la credencial de usuario cliente.

partitionName: nombre opcional de la partición de campaña que se va a utilizar.

requestedLocale: entorno local opcional que se utiliza para esta solicitud.

wsReferences: referencia para ofrecer listas.

Devoluciones

Ninguna.

Errores

Se emite CampaignServicesException si las referencias de lista no son válidas.

createFolder

```
public WSCreateFolderResponse createFolder(String userCredential,
String partitionName, Locale requestedLocale,String name,
String description,String securityPolicyName,
long parentFolderId,WSFolderTypeEnum folderType,
WSApplicationTypeEnum createdBy,long creatorObjectId)
 throws CampaignServicesException
```

Crea una nueva carpeta de tipo campaña/oferta/sesiones/segmentos.

Parámetros

userCredential: la credencial de usuario cliente.

partitionName: nombre opcional de la partición de campaña que se va a utilizar.

requestedLocale: entorno local opcional que se utiliza para esta solicitud.

nombre: nombre de la carpeta que se creará.

descripción: descripción de la carpeta nueva.

securityPolicyName: nombre de la política de seguridad que se utilizará.

parentFolderId: (opcional) ID de carpeta padre. Si no se proporciona, la carpeta se crea en la raíz.

folderType: tipo de carpeta (oferta, sesión, campaña, segmento).

createdBy: (opcional) el objeto de tipo WSApplicationTypeEnum indica la aplicación que ha creado la carpeta. Los valores posibles son Campaign, Plan, Collaborate y eMessage. Si no se proporciona, se utiliza Campaign.

Nota: Plan=IBM Marketing Operations. Collaborate=IBM Distributed Marketing.

creatorObjectId: (opcional) lo utiliza el Plan (IBM Marketing Operations) para enlazar una carpeta de IBM Campaign con una carpeta de Marketing Operations.

Devoluciones

Devuelve un objeto de tipo WSCreateFolderResponse.

Errores

Se emite CampaignServicesException si el tipo de carpeta no es válido o está duplicado.

editFolder

```
public WSEditFolderResponse editFolder(String userCredential,
String partitionName, Locale requestedLocale, long folderId,
String name,String description, WSFolderTypeEnum folderType,
Long creatorObjectId, boolean clearExisting)
 throws CampaignServicesException
```

Actualiza la carpeta especificada.

Parámetros

userCredential: la credencial de usuario cliente.

partitionName: nombre opcional de la partición de campaña que se va a utilizar.

requestedLocale: entorno local opcional que se utiliza para esta solicitud.

id: ID de carpeta que se actualizará.

nombre: nombre de la carpeta que se creará.

descripción: descripción de la carpeta nueva.

folderType: tipo de carpeta (oferta, sesión, campaña, segmento).

creatorObjectId: (opcional) lo utiliza el Plan (IBM Marketing Operations) para enlazar una carpeta de IBM Campaign con una carpeta de Marketing Operations.

clearExisting: ahora no se utiliza.

Devoluciones

Devuelve un objeto de tipo WSEditFolderResponse.

Errores

Se emite CampaignServicesException si el tipo de carpeta no es válido, está duplicado o si el ID de carpeta no es válido.

getSubFoldersList

```
public WSGetSubFolderListResponse getSubFoldersList(String user_credential,
String partition_name, Locale locale, WSReference parentFolder,
WSFolderTypeEnum folderType)
 throws CampaignServicesException
```

Lista todas las subcarpetas de la carpeta especificada.

Parámetros

userCredential: la credencial de usuario cliente.

partitionName: nombre opcional de la partición de campaña que se va a utilizar.

requestedLocale: entorno local opcional que se utiliza para esta solicitud.

parentFolder: referencia de carpeta a la que se solicitan todas las subcarpetas.

folderType: tipo de carpeta.

Devoluciones

Devuelve el objeto de tipo WSGetSubFolderListResponse.

Errores

Se emite CampaignServicesException si la referencia de carpeta no es válida.

moveFolders

```
public WSMoveFolderResponse moveFolders(String userCredential,
String partitionName, Locale requestedLocale, Long[] folderId,
long parentFolder,long destinationFolder,
 WSFolderTypeEnum folderType)
 throws CampaignServicesException
```

Mueve la carpeta especificada a otra carpeta padre.

Parámetros

userCredential: la credencial de usuario cliente.

partitionName: nombre opcional de la partición de campaña que se va a utilizar.

requestedLocale: entorno local opcional que se utiliza para esta solicitud.

folderId: ID de carpeta para suprimir.

parentFolder: ID de carpeta padre.

destinationFolder: ID de carpeta de destino donde se mueve la carpeta especificada.

folderType: tipo de carpeta (oferta, sesión, campaña, segmento).

Devoluciones

Devuelve el objeto de tipo WSMoveFolderResponse.

Errores

Se emite CampaignServicesException si el tipo de carpeta o ID de carpeta padre no es válido o si el ID de carpeta no es válido.

deleteFolders

```
public WSDeleteFolderResponse deleteFolders(String userCredential,
 String partitionName, Locale requestedLocale, Long[] folderId,
 long parentFolder,boolean deleteChilds,
WSFolderTypeEnum folderType)
 throws CampaignServicesException
```

Suprime las carpetas especificadas en el sistema junto con todos los elementos de la carpeta.

Parámetros

userCredential: la credencial de usuario cliente.

partitionName: nombre opcional de la partición de campaña que se va a utilizar.

requestedLocale: entorno local opcional que se utiliza para esta solicitud.

folderId: ID de carpeta para suprimir.

parentFolder: ID de carpeta padre.

deleteChilds: distintivo que indica si todas las dependencias de la carpeta se deben suprimir. Si se especifica false, no se suprime ninguna dependencia.

folderType: tipo de carpeta (oferta, sesión, campaña, segmento).

Devoluciones

Devuelve un objeto de tipo WSDeleteFolderResponse.

Errores

Se emite CampaignServicesException si el tipo de carpeta no es válido o si el ID de carpeta no es válido.

Capítulo 5. Excepciones comunes de la API SOAP

La API SOAP de IBM Campaign puede generar las siguientes excepciones comunes. Todos los mensajes localizados se hallan en el entorno local solicitado si está disponible para IBM Campaign. Se aplican las políticas de reserva del entorno local normal de Java.

RemoteException

Este elemento se aplica solamente ala interfaz SOAP.

Todas las llamadas SOAP a la API pueden emitir una RemoteException si se detecta un error a nivel del sistema como, por ejemplo, un problema en la capa de proceso SOAP (Axis), se ha infringido por algún motivo una restricción, que se define en el WSDL de servicio web.

Las excepciones comunes de API comprobadas y no comprobadas, como DataException, se devolverán como un estado de error, no como una RemoteException.

Consulte la sección de la interfaz SOAP para obtener detalles.

AuthenticationException

El usuario no puede autenticarse para la partición de Campaign especificada. Compruebe el rol de usuario que se ha definido en IBM Marketing Platform.

Excepción de autorización

El usuario no está autorizado a realizar la operación solicitada. Cualquier método de API puede emitir esta excepción, por lo que no se declara (no se selecciona). Compruebe los permisos que se han asignado al rol de usuario en IBM Marketing Platform.

Excepción de datos

Se ha producido una excepción muy grave en la capa de base de datos subyacente en Campaign (deseleccionado).

Compruebe los registros del escucha y del diagrama de flujo de IBM Campaign para obtener detalles.

Excepción de bloqueo

Una excepción temporal emitida cuando el cliente intenta actualizar un componente como, por ejemplo, un diagrama de flujo, mientras lo edita otro usuario. Generalmente, se puede realizar una recuperación a partir de esta excepción esperando y, a continuación, volviendo a intentar la operación. La lógica de reintento es responsabilidad del cliente.

© Copyright IBM Corp. 1998, 2017

InvalidComponentException

Se ha intentado hacer referencia a un componente no válido o desconocido (campaña, diagrama de flujo, celda objetivo). El método `getComponentReference()` de la excepción devuelve la referencia del componente problemático.

InvalidAttributeException

Una excepción que se emite cuando el cliente proporciona o hace referencia a un atributo no válido como, por ejemplo, si utiliza el tipo de datos incorrecto o si utiliza una matriz de valores donde no hay ninguna permitida. El método `getAttributeName()` de la excepción devuelve el nombre del atributo del problema, `getAttributeValue()` devuelve el valor y `getComponentReference()` identifica el componente (o índice colectivo).

AttributeExistsException

Se emite cuando el cliente intenta definir metadatos de un atributo duplicado para un componente. El método `getAttributeName()` de la excepción devuelve el nombre del atributo duplicado; `getComponentReference()` identifica el componente (o índice colectivo).

AttributeNotFoundException

Se genera cuando el cliente intenta hacer referencia a un atributo desconocido (campaña, diagrama de flujo, celda objetivo, etc.). El método `getAttributeName()` devuelve el nombre del atributo no coincidente; `getComponentReference()` identifica el componente (o índice colectivo).

CompositeException

Algunas API utilizan una CompositeException para informar de los diversos errores al llamante. Normalmente habrá más de una causa vinculada; todas las causas se capturan como una lista en el orden en que se producen. El método `getCauseList()` de la excepción devuelve esta lista, que se puede inspeccionar para obtener los detalles de cada error.

Nota: Generalmente, la API completa correctamente o retrotrae su propio trabajo antes de emitir una excepción compuesta. Consulte, por ejemplo, las API de hoja de cálculo de celdas objetivo colectivas descritas en “Métodos de la API SOAP: celdas objetivo” en la página 31.

Antes de ponerse en contacto con el equipo de soporte técnico de IBM

Si encuentra un problema que no puede resolver consultando la documentación, el contacto de soporte designado por la empresa puede realizar una llamada al soporte técnico de IBM. Siga estas directrices para asegurarse de que el problema se resuelve de forma eficiente y satisfactoria.

Si usted no es una de las personas designadas como 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 por recopilar

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

- Una descripción breve de la naturaleza del problema.
- Los mensajes de error detallados que ve cuando se produce el problema.
- Pasos detallados para reproducir el problema.
- Archivos de registro, archivos de sesión, archivos de configuración y archivos de datos relacionados con el problema.
- Información sobre el producto y el entorno del sistema del , 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 iniciar sesión, 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 está accesible, compruebe si hay un archivo `version.txt` en el directorio de instalación de la aplicación.

Información de contacto del soporte técnico de IBM

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

Nota: Para especificar una solicitud de soporte, hay que iniciar sesión con una cuenta de IBM. Esta cuenta tiene que estar vinculada con su número de cliente de

© Copyright IBM Corp. 1998, 2017

IBM Para obtener más información sobre cómo asociar la cuenta a su número de cliente de IBM, consulte **Recursos de soporte>Soporte de software contratado** en el portal de soporte.

Avisos

Esta información se ha desarrollado para productos y servicios ofrecidos en EE.UU.

Puede que IBM no ofrezca los productos, servicios o características tratados en este documento en otros países. Póngase en contacto con el representante local de IBM para obtener información sobre los productos y servicios que se pueden adquirir actualmente en su zona geográfica. Las referencias a programas, productos o servicios de IBM no pretenden dar a entender ni implicar que solo puedan utilizarse dichos productos, programas o servicios de IBM. En su lugar se puede utilizar cualquier producto, programa o servicio funcionalmente equivalente que no vulnere ningún derecho de propiedad intelectual de IBM. No obstante, es responsabilidad del usuario evaluar y verificar el funcionamiento de cualquier producto, programa o servicio no IBM.

Puede que IBM tenga patentes o solicitudes de patente pendientes que traten el tema descrito en este documento. La posesión de este documento no confiere ninguna licencia sobre dichas patentes. Puede enviar sus consultas sobre licencias, por escrito, a:

IBM Director of Licensing
IBM CorporationNorth Castle Drive
Armonk, NY 10504-1785
EE.UU.

Para realizar consultas sobre licencias relacionadas con la información de doble byte (DBCS), póngase en contacto con el departamento de propiedad intelectual de IBM en su país o envíe sus consultas, por escrito, a:

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

El párrafo siguiente no se aplica al Reino Unido ni a ningún otro país donde las disposiciones en él expuestas sean incompatibles con la legislación local: INTERNATIONAL BUSINESS MACHINES CORPORATION PROPORCIONA ESTA PUBLICACIÓN "TAL CUAL", SIN GARANTÍA DE NINGUNA CLASE, EXPRESA O IMPLÍCITA, INCLUYENDO, PERO SIN LIMITARSE A, LAS GARANTÍAS IMPLÍCITAS DE NO INFRACCIÓN, COMERCIALIZACIÓN O IDONEIDAD PARA UN FIN DETERMINADO. Algunos países no permiten la renuncia a garantías explícitas o implícitas en determinadas transacciones, por lo que puede que esta declaración no sea aplicable en su caso.

Esta información podría contener imprecisiones técnicas o errores tipográficos. Se realizan cambios periódicos en la información aquí contenida; estos cambios se incorporarán en nuevas ediciones de la publicación. Puede que IBM realice mejoras y/o modificaciones en los productos y/o programas descritos en esta publicación en cualquier momento sin previo aviso.

Las referencias en esta información a sitios web que no son de IBM se proporcionan solo para su comodidad y en modo alguno constituyen una

© Copyright IBM Corp. 1998, 2017

recomendación de dichos sitios web. Los materiales de dichos sitios web no forman parte de los materiales de este producto de IBM y el uso de dichos sitios web corre por cuenta y riesgo del usuario.

IBM se reserva el derecho a utilizar o distribuir, en la forma que considere más adecuada, la información que se le facilite sin contraer por ello ninguna obligación con usted.

Los licenciatarios de este programa que deseen obtener información sobre el mismo para poder: (i) intercambiar información entre programas creados de forma independiente y otros programas (incluidos este) y (ii) utilizar, de forma mutua, la información que se ha intercambiado, habrán de ponerse en contacto con:

IBM CorporationB1WA LKG1
550 King Street
Littleton, MA 01460-1250
EE.UU.

Dicha información puede estar disponible conforme a los correspondientes términos y condiciones, incluyendo en algunos casos el pago de una tarifa.

IBM proporciona el programa bajo licencia descrito en este documento, y todo el material bajo licencia disponible para el mismo, conforme a los términos del Contrato de cliente de IBM, del Acuerdo Internacional de Programas bajo Licencia de IBM o de cualquier acuerdo equivalente entre IBM y el cliente.

Los datos de rendimiento contenidos en esta documentación se han determinado en un entorno controlado. Por lo tanto, los resultados obtenidos en otros entornos operativos pueden variar de forma significativa. Es posible que algunas mediciones se hayan realizado en sistemas a nivel de desarrollo y no existe ninguna garantía de que dichas mediciones sean las mismas en los sistemas disponibles de manera general. Además, es posible que algunas mediciones se hayan estimado mediante extrapolación. Los resultados reales podrían ser distintos. Los usuarios de este documento deberán verificar los datos aplicables a su entorno específico.

La información relativa a productos que no son de IBM se ha obtenido de los proveedores de dichos productos, de los anuncios publicados por los mismos o de otras fuentes a disposición del público. IBM no ha probado dichos productos y no puede confirmar la precisión del rendimiento, compatibilidad o cualquier otra declaración relacionada con productos que no son de IBM. Las preguntas sobre las prestaciones de los productos que no son de IBM se deben dirigir a los proveedores de tales productos.

Todas las declaraciones relativas a la orientación o intención futura de IBM están sujetas a cambio o anulación sin previo aviso y representan solamente metas y objetivos.

Todos los precios de IBM indicados son precios de minorista recomendados por IBM, son actuales y están sujetos a cambio sin previo aviso. Los precios de los distribuidores pueden variar.

Esta información contiene ejemplos de datos e informes utilizados en operaciones de negocio cotidianas. Para ilustrarlas de la forma más completa posible, los ejemplos incluyen nombres de personas, empresas, marcas y productos. Todos estos nombres son ficticios y cualquier parecido con los nombres y direcciones utilizados por una empresa real es pura coincidencia.

LICENCIA DE COPYRIGHT:

Esta información contiene ejemplos de programas de aplicación en código fuente que ilustran las técnicas de programación en diversas plataformas operativas. Puede copiar, modificar y distribuir estos programas de ejemplo de cualquier manera sin realizar ningún pago a IBM, a fin de desarrollar, utilizar, comercializar y distribuir programas de aplicación que se adecúen a la interfaz de programación de aplicaciones de la plataforma operativa para la que se han escrito los programas de ejemplo. Estos ejemplos no se han probado exhaustivamente bajo todas las condiciones. Por lo tanto, IBM no puede garantizar ni implicar la fiabilidad, el buen servicio o el funcionamiento de estos programas. Los programas de ejemplo se proporcionan "TAL CUAL" sin garantías de ningún tipo. IBM no se responsabiliza de ningún daño derivado de la utilización de los programas de ejemplo.

Si está viendo esta información en copia software, es posible que las fotografías y las ilustraciones en color no aparezcan.

Marcas registradas

IBM, el logotipo de IBM e ibm.com son marcas comerciales o registradas de International Business Machines Corp. en muchas jurisdicciones de todo el mundo. Otros nombres de productos y servicios pueden ser marcas registradas de IBM o de otras compañías. Encontrará la lista actualizada de las marcas registradas de IBM en la web, en "Copyright and trademark information" en www.ibm.com/legal/copytrade.shtml.

Consideraciones sobre la política de privacidad y los términos de uso

Los productos software de IBM Software, incluido el software como una solución de servicio, ("Ofertas de Software") pueden utilizar cookies u otras tecnologías para recopilar información de utilización de producto, para ayudar a mejorar la experiencia del usuario final, para adaptar las interacciones con el usuario final o con otros fines. Una cookie contiene datos que un sitio web puede enviar al navegador y que luego se pueden almacenar en el sistema como una etiqueta que identifica al sistema. En muchos casos, estas cookies no recopilan información personal. Si utiliza una Oferta de software que le permite recopilar información personal mediante cookies y tecnologías similares, a continuación le ofrecemos información específica.

Dependiendo de las configuraciones desplegadas, esta Oferta de software puede utilizar cookies de sesión y persistentes que recopilen el nombre de cada usuario y otra información personal a efectos de gestión de sesiones, utilización de usuario mejorada u otros fines funcionales o de seguimiento de uso. Estas cookies pueden inhabilitarse, pero si se inhabilitan también se eliminará la funcionalidad que habilitan.

Diversas jurisdicciones regulan la recopilación de información personal mediante cookies y tecnologías similares. Si las configuraciones desplegadas para esta Oferta de Software le proporcionan como cliente la posibilidad de recopilar información personal sobre usuarios a través de cookies u otras tecnologías, deberá asesorarse sobre la legislación vigente relativa a dicha recopilación de datos, incluidos los requisitos para proporcionar avisos y el consentimiento cuando corresponda.

IBM requiere que los Clientes (1) proporcionen un enlace claro y visible a los términos de uso del sitio web del Cliente (por ejemplo, política de privacidad) que

incluya un enlace a las prácticas de uso y recopilación de datos de IBM y del cliente, (2) notifiquen que IBM coloca, en nombre del Cliente, cookies y balizas web/gifs claras en el sistema del visitante, junto con una explicación de la finalidad de dicha tecnología, y (3) en la medida requerida por ley, obtengan el consentimiento de los visitantes del sitio web antes de la colocación de cookies y balizas web/gifs por parte del Cliente o IBM en nombre del Cliente, en dispositivos del visitante del sitio web

Si desea obtener más información sobre la utilización de las distintas tecnologías, incluidas las cookies, a tales efectos, consulte la declaración de privacidad en línea de IBM en http://www.ibm.com/privacy/details/us/en, en la sección titulada "Cookies, Web Beacons and Other Technologies".

Impreso en España