

IBM Campaign  
Versione 9 Release 1.1  
26 novembre 2014

*Specifica API di servizi*

**IBM**

**Nota**

Prima di utilizzare queste informazioni ed il prodotto che supportano, leggere le informazioni contenute in "Informazioni particolari" a pagina 63.

Questa edizione si applica alla versione 9, release 1, modifica 1 di IBM Campaign e a tutte le release e le modifiche successive, se non diversamente indicato nelle nuove edizioni.

© Copyright IBM Corporation 1998, 2014.

# Indice

## Capitolo 1. Panoramica sull'API di servizi IBM Campaign . . . . . 1

Panoramica sulla progettazione API . . . . .	1
Modifiche all'API per versione . . . . .	3
Implementazioni esistenti che utilizzano il file jar dell'API client . . . . .	4
Implementazioni esistenti che utilizzano direttamente WSDL . . . . .	5
Riferimenti . . . . .	7

## Capitolo 2. Utilizzo dell'API di servizi IBM Campaign . . . . . 9

Utilizzo del file .jar dell'API client per chiamare i servizi Campaign . . . . .	9
OfferAPI.java . . . . .	9
Utilizzo di WSDL per chiamare i servizi Campaign . . . . .	11
Considerazioni sulle prestazioni . . . . .	11

## Capitolo 3. Tipi di dati . . . . . 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

## Capitolo 4. Metodi API . . . . . 21

Metodi API: Servizio . . . . .	21
getServiceInfo . . . . .	21
Metodi API: Attributi . . . . .	21
getAttributesByName . . . . .	21
updateAttributes . . . . .	22
getAttributeMetadataByName . . . . .	23
createAttributeMetadata . . . . .	24
updateAttributeMetadata . . . . .	25
deleteAttributeMetadata . . . . .	26
Metodi API: Campagne e diagrammi di flusso . . . . .	27
generateCampaignCode . . . . .	28

deleteCampaigns . . . . .	28
createCampaign . . . . .	29
listCampaignsByPage . . . . .	30
stopFlowchart . . . . .	31
Metodi API: Celle obiettivo . . . . .	31
createTargetCell . . . . .	32
bulkCreateTargetCells . . . . .	33
listTargetCells . . . . .	34
bulkUpdateTargetCells . . . . .	35
getRunResultsByCell . . . . .	36
bulkDeleteTargetCells . . . . .	37
updateTemplateAttributes . . . . .	37
listBottomUpTargetCells . . . . .	38
Metodi API: Analitiche . . . . .	38
getCampaignMetrics . . . . .	39
Metodi API: Offerte, elenchi offerte, modelli di offerte . . . . .	39
listOffersAndFolders . . . . .	40
searchOffersBasic . . . . .	40
listOffersByPage . . . . .	41
createSmartOfferList . . . . .	42
createStaticOfferList . . . . .	43
getOffers . . . . .	44
validateOffers . . . . .	44
editOfferList . . . . .	45
createOffer . . . . .	46
retireOffers . . . . .	47
deleteOffers . . . . .	47
deleteOffersAndLists . . . . .	48
listOfferTemplates . . . . .	48
createTemplate . . . . .	49
getOfferTemplate . . . . .	50
retireOfferTemplates . . . . .	50
getOffersAndListsByPage . . . . .	51
bulkCreateOffers . . . . .	51
getOfferListDetails . . . . .	52
getOfferListMembers . . . . .	52
getOffersByQuery . . . . .	53
retireOfferLists . . . . .	53
createFolder . . . . .	54
editFolder . . . . .	55
getSubFoldersList . . . . .	55
moveFolders . . . . .	56
deleteFolders . . . . .	57

## Capitolo 5. Eccezioni comuni . . . . . 59

### Prima di contattare il supporto tecnico di IBM . . . . . 61

### Informazioni particolari. . . . . 63

Marchi . . . . .	65
Normativa sulla privacy e termini di utilizzo - Considerazioni . . . . .	65



---

## Capitolo 1. Panoramica sull'API di servizi IBM Campaign

La specifica API di servizi IBM® Campaign definisce la versione 3.0 dell'API, indicata anche come Servizi Campaign. Questa specifica è fornita come parte di Campaign Services Software Developer's Toolkit (devkits), installato con IBM Campaign.

La directory <CAMPAIGN\_HOME>/devkits/CampaignServicesAPI fornita dal programma di installazione include esempi, build e script di testo, Javadoc per le classi pubbliche e le interfacce, e le note sulla release.

L'API di servizi Campaign è progettata per effettuare le operazioni seguenti.

- Fornire i dettagli di accesso per creazione, rilevamento, lettura e aggiornamento ai componenti IBM Campaign, mentre i client sono isolati dai dettagli di implementazione sottostanti.
- Operare insieme all'interfaccia utente di IBM Campaign con effetti minimi.
- Garantire la validità dei dati.
- Rispettare i servizi di sicurezza di IBM Campaign richiesti.
- Supportare lo standard SOAP, inclusa l'autenticazione sicura.

---

### Panoramica sulla progettazione API

L'API di servizi Campaign è un componente di facciata che fornisce una vista client di una istanza dell'applicazione IBM Campaign. Viene esposto solo un sottoinsieme delle funzioni di IBM Campaign, ma è sufficiente per rendere disponibili gli aspetti chiave delle funzionalità di Campaign.

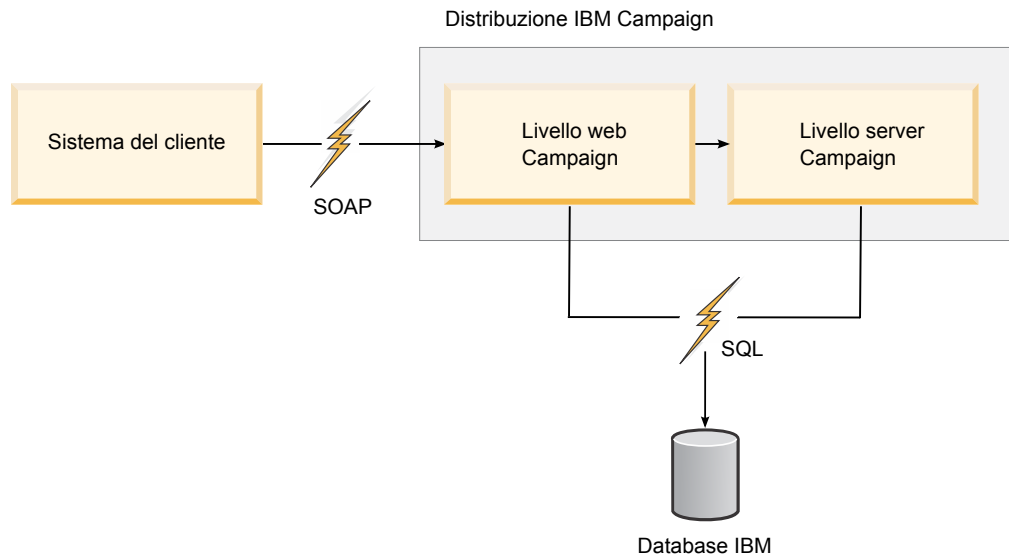
#### Funzioni e diagramma

L'API è utilizzata contemporaneamente agli utenti web di IBM Campaign e altri thread API.

Generalmente l'API supporta i seguenti tipi di operazioni sui componenti campagne, offerte e celle obiettivo:

- Creazione di componente
- Rilevamento di componente
- Eliminazione di componente
- Creazione, ispezione e modifica di attributi e metadati di attributi del componente
- Estrazione dei risultati dell'esecuzione del diagramma di flusso

Il diagramma seguente mostra una distribuzione di esempio dei servizi Campaign 3.0.



## Autenticazione utente

L'autenticazione ha lo scopo stabilire l'identità di un utente. L'autenticazione utente è responsabilità dell'applicazione client.

## Autorizzazione utente

L'autorizzazione riguarda le autorizzazioni di un utente autenticato rispetto ai componenti e alle operazioni esposte dall'API.

È possibile che un utente esegua correttamente l'autenticazione, ma non disponga di autorizzazioni sufficienti per eseguire alcune operazioni, ad esempio modificare le informazioni di riepilogo di una campagna. In questo caso, il metodo API genera una `AuthorizationException`.

## Locale

API richiede di fornire il parametro facoltativo **requestedLocale**, che definisce la locale da utilizzare per completare una particolare richiesta. Se non è definita, l'API assume come impostazione predefinita la locale preferita dell'utente IBM. Per restituire i messaggi ed altro testo localizzato nella locale richiesta viene utilizzato il normale algoritmo di corrispondenza Java™.

Questo parametro è di tipo classe `java.util.Locale`.

**Nota:** Alcuni testi specificati dall'utente, ad esempio le descrizioni della campagna, sono nella locale dell'utente che ha specificato il testo. IBM Campaign non cerca di localizzare tali dati. Solo i messaggi informativi, di avvertenza e di errore sono localizzati dall'API.

## Gestione stato

L'API di servizi Campaign è priva di stato, ovvero l'API non salva alcuna informazione per client tra le chiamate.

Naturalmente, chiamate API specifiche possono modificare lo stato delle istanze del componente sottostante che sono gestite da Campaign, e queste modifiche di

stato potrebbero essere persistenti per il database.

---

## Modifiche all'API per versione

Lo scopo di questa sezione è di identificare le modifiche all'API di servizi IBM Campaign per i clienti che attualmente utilizzano l'API. Se è stato eseguito l'aggiornamento da una versione precedente, esaminare le informazioni seguenti per determinare se è necessario apportare modifiche al codice dell'applicazione.

### Versioni e compatibilità all'indietro

Le versioni future dell'API di servizi Campaign saranno compatibili con le versioni precedenti, con tutte le release minori e di manutenzione che condividono un numero di versione principale. Tuttavia, IBM si riserva il diritto di interrompere la compatibilità con la versione precedente per le release principali (x.0).

Il numero di versione principale dell'API viene incrementato se è stata effettuata una delle seguenti modifiche:

- Interpretazione dei dati modificata
- Logica di business, ovvero funzionalità del metodo del servizio, modificata
- Parametri del metodo e/o tipi di ritorno modificati

Il numero di versione secondaria dell'API viene incrementato se è stata effettuata una delle seguenti modifiche.

- Nuovo metodo aggiunto
- Nuovo tipo di dati aggiunto e relativo utilizzo limitato a nuovi metodi
- Nuovo tipo aggiunto ad un tipo enumerato
- Nuova versione di interfaccia definita

IBM continua a supportare il WSDL pubblicato, il client SOAP e la versione di Apache Axis utilizzata per implementare l'offerta SOAP almeno fino alla successiva release principale IBM . In pratica, ciò si ottiene supportando contemporaneamente diversi servizi web specifici di versione. IBM già supporta internamente numerose versioni di questo servizio.

### Modifiche nella versione 9.1.1

**stopFlowchart** include un nuovo parametro di input, **runId**, per supportare un ambiente listener con cluster.

### Modifiche nella versione 9.1

Non erano presenti modifiche all'API nella versione 9.1.

### Modifiche nella versione 9.0

Non erano presenti modifiche all'API nella versione 9.0.

### Modifiche nella versione 8.6

Modifiche all'API implementate nella versione 8.6:

- Il motore SOAP era stato aggiornato da AXIS versione 1.4.1 a AXIS2 1.5.2.
- Il WSDL era stato ristrutturato per risolvere i problemi nella gestione di parametri obbligatori e facoltativi.

- Il file .jar dell'API client era stato modificato a causa delle modifiche di WSDL. Di conseguenza, le classi e gli stub generati erano stati modificati. I parametri del metodo dell'API client non erano stati modificati, ma i costruttori degli oggetti valore di supporto erano stati modificati per l'uso del convertitore AXIS2 WSDL2Java.
- L'URL del servizio web punta a:

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

e

il corrispondente WSDL viene recuperato all'indirizzo:

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

Se si sta eseguendo l'aggiornamento a IBM Campaign versione 8.6 o successiva e attualmente si sta utilizzando l'API di servizi Campaign, è necessario modificare il codice dell'applicazione.

A seconda che si utilizzi l'API client o il WSDL, consultare le seguenti sezioni per i dettagli:

- “Implementazioni esistenti che utilizzano il file jar dell'API client”
- “Implementazioni esistenti che utilizzano direttamente WSDL” a pagina 5

## Implementazioni esistenti che utilizzano il file jar dell'API client

Queste informazioni si applicano se si sta aggiornando a IBM Campaign versione 8.6 o successiva e attualmente viene utilizzato il file .jar dell'API client per interagire con l'applicazione web Campaign.

### File .jar dell'API client

L'applicazione Java dell'utente deve utilizzare il file .jar che si trova in:

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

Per un esempio Java che mostra la creazione di una nuova offerta, consultare “OfferAPI.java” a pagina 9. Lo stesso esempio si trova nell'installazione Campaign in:

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

### File .jar dipendenti

Come risultato dell'aggiornamento a AXIS2 versione 1.5.2, anche l'applicazione Java dell'utente deve essere aggiornata utilizzando i file .jar della distribuzione AXIS2 1.5.2, poiché CampaignServicesClient30.jar dipende da questi file .jar. Tutti i file .jar dipendenti devono essere inclusi nel percorso della classe Java dell'applicazione e si trovano nel file Campaign.war in `<CAMPAIGN_HOME>/Campaign.war`.

Estrarre i file .jar da Campaign.war e includerli nel percorso della classe Java.



## Costruttore di API client

Quando si costruisce l'oggetto API client, modificare l'URL del servizio web e la firma di eccezione come mostrato in questo esempio.

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

## Costruttori parametrizzati delle classi di supporto

Con il motore AXIS2, le classi e gli stub generati non presentano costruttori parametrizzati. Al contrario, queste classi hanno solo il costruttore predefinito senza argomenti con setter e getter per i membri.

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

## Implementazioni esistenti che utilizzano direttamente WSDL

Queste informazioni si applicano se si sta aggiornando a IBM Campaign versione 8.6 o o successiva e attualmente viene utilizzato WSDL per interagire con l'applicazione web Campaign. Il WSDL del servizio web di Campaign viene utilizzato per generare stub lato client e classi di supporto con un qualsiasi strumento di conversione di terze parti. Gli esempi forniti qui utilizzano lo strumento WSDL2Java da Apache AXIS2 1.5.2.

### Ubicazione WSDL e URL del servizio

Il servizio web di Campaign per IBM Campaign viene distribuito all'indirizzo:

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

Il corrispondente WSDL può essere recuperato all'indirizzo:

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

### Generazione di stub e classi

Lo strumento WSDL2Java da Apache AXIS2 1.5.2 può essere utilizzato per generare gli stub e le classi Java di supporto dal WSDL. Qui viene mostrato un esempio di attività Ant.

Lo strumento può essere utilizzato anche da riga comandi con una serie di argomenti simili. I valori degli argomenti possono essere modificati per adattarli al proprio ambiente.

**Nota:** Per il seguente esempio di convertitore WSDL2Java viene utilizzato il binding ADB.

```
<java classname="org.apache.axis2.wsdl.WSDL2Java" fork="true">
  <classpath refid="axis2.class.path"/> <!--Class path having
AXIS2 libraries -->
  <arg value="-uri"/>
  <arg file="CampaignServices30.wsdl"/> <!--Actual location of
```

```

WSDL -->
  <arg value="-s"/> <!-- Generate sync style code -->
  <arg value="-Euwc"/> <!-- Takes care of generating Wrapper
java types for nillable = true elements. -->
  <arg value="-uw"/> <!-- Unwrap params -->
  <arg value="-u"/> <!-- Unpack classes -->
  <arg value="-ns2p"/> <!-- Namespace to package mapping. Customer
can have their own package names. -->
  <arg value="http://webservices.unica.com/campaign/CampaignServices/
3.0=com.unica.publicapi.campaign.campaignservices.soap.v30"/>
  <arg value="-o"/> <!-- Output directory -->
  <arg file="{autogen.java.dir}"/>
</java>

```

## Utilizzo di stub e classi di supporto generati

Gli stub possono essere utilizzati nel modo seguente:

```

CampaignServices30ServiceStub serviceStub = new
CampaignServices30ServiceStub(serviceURL);

serviceStub._getServiceClient().getOptions().setTimeoutInMilliseconds
(webServiceTimeout); //Timeout in milliseconds.

```

L'offerta può essere creata nel modo seguente:

```

try{
  //Please change host and port to match your environment.
  String serviceURL = "http://host:port/Campaign/services/CampaignServices30Service";
  CampaignServices30ServiceStub serviceStub = new CampaignServices30ServiceStub(serviceURL);
  long webServiceTimeout = 2*60*1000; // 2 minutes
  serviceStub._getServiceClient().getOptions().setTimeoutInMilliseconds(webServiceTimeout); //Timeout in milliseconds.

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

  WSTextAttribute[] wsAttributes = {nameAttribute};
  // convert to WSAttributeArrays
  WSAttributeArrays obj = new WSAttributeArrays();
  obj.setTextAttributes(wsAttributes);
  //Please change the values of following variables to match your environment.
  String authorizationLoginName = "asm_admin"; //login user name
  String partitionName = "partition1"; //Use your security policy of Campaign
  String securityPolicyName = "Global Policy"; //Use your security policy of Campaign

  String offerName = "1st Offer"; //Name of the offer to be created.
  String templateName = "Offer Template"; //Existing offer template name.
  long folderID = 100; //Actual ID of the folder where this offer will be created.
  //For folderID <=0, offer will be created at root level.

  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);
  // make campaign Webservice call
  WSCreateOfferResponse wsResponse = serviceStub.createOffer(createOfferObject);
  // process status
  WSRequestStatus status = wsResponse.getStatus();
  // done
  WSOfferInfo offerInfo = wsResponse.getOfferInfo();
  System.out.println("status = "+status.getStatusType());
  System.out.println("offerInfo = "+offerInfo.getName());
} catch (Exception exception) {
  //Handle the Exception here.
  exception.printStackTrace();
}

```

In questo esempio, `createOffer()` accetta ora soltanto un parametro di tipo `CreateOffer`.

Con il motore AXIS2, le classi e gli stub generati non presentano più costruttori parametrizzati.

---

## Riferimenti

I riferimenti seguenti sono stati utilizzati per preparare questa specifica:

- "Basic Profile Version 1.1", Web Service Interoperability Organization (WS-I), 10 aprile, 2006. (<http://www.ws-i.org/Profiles/BasicProfile-1.1-2006-0310.html>)
- "SOAP 1.2 (draft)", W3C Soap working group, 24 giugno, 2003 (<http://www.w3.org/TR/soap/>)
- "JAX-RPC 1.1", Sun Microsystems, 14 ottobre, 2003 (<http://java.sun.com/webservices/jaxrpc/index.jsp>)
- Gruppo di lavoro servizi web Apache (<http://ws.apache.org/axis2>)



---

## Capitolo 2. Utilizzo dell'API di servizi IBM Campaign

Per utilizzare l'API di servizi web IBM Campaign, è possibile utilizzare il file .jar dell'API client o direttamente WSDL. Un esempio mostra come usare il metodo del file .jar per creare un'offerta.

---

### Utilizzo del file .jar dell'API client per chiamare i servizi Campaign

IBM Campaign fornisce un'API client che utilizza i servizi web SOAP per interagire con l'applicazione web Campaign. Questo wrapper è racchiuso in un file .jar che può essere utilizzato dall'applicazione client per chiamare l'API Campaign.

Il file .jar si trova in:

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

Il seguente esempio mostra la creazione di una nuova offerta a livello della cartella offerta principale in Campaign. Lo stesso esempio si trova in:

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

**Nota:** L'esempio utilizza alcuni valori fittizi per i parametri; i valori effettivi potrebbero essere diversi.

Inoltre, l'URL per i servizi web di Campaign è `http://host:porta/Campaign/services/CampaignServices30Service`, dove `host` e `porta` fanno riferimento al nome `host` e al numero di porta della macchina in cui è stata distribuita l'applicazione web Campaign.

Se si utilizza un esempio fornito, assicurarsi di modificarlo per adattarlo al proprio ambiente.

### OfferAPI.java

Per compilare ed eseguire questo esempio Java, è necessario includere tutti i file .jar dipendenti nel percorso classi Java. Il file `CampaignServicesClient30.jar` dipende dai file .jar del motore Apache AXIS2 SOAP e da altri file .jar Apache comuni, che si trovano in `<CAMPAIGN_HOME>/Campaign.war`. Estrarre i file .jar da `Campaign.war` e includerli nel percorso classi 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;
```

```
/**  
 * This is the sample java client class that shows the usage of Campaign SOAP  
 services API.
```

```

    * This sample uses CampaignServices30SoapClient facade to interact with Campaign
    web service.
    * Here the creation of Offer is shown. Please refer to the API guide for
    more details.
    *
    * @author AGijare
    *
    */
public class OfferAPI {

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

    private static void setup(){
        try {
            String protocol = "http"; //http or https
            String host = "localhost"; //Host name of deployed Campaign.
            Use proper host name here.
            int port = 7001; //port number of deployed Campaign
            long timeOut = 2*60*1000; // 2 minutes
            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) {
        //Please change the values of following variables to match your
        environment.
        String userName = "user_name"; //login user name
        String partitionName = "partition1"; //Use proper partition name of
        Campaign
        Locale loc = Locale.US;
        String securityPolicy = "Global"; //Use your security policy of
        Campaign

        String offerName = "Offer1";
        String offerTemplate = "Offer Template"; // Template from which
        Offer will be created.
        long folderID = 1002; //Actual ID of the folder where this offer
        will be created. For folderID <=0, offer will be created at root level.
        //Attributes of Offer
        WSAttribute[] wsAttributes = {
            WSAttributeUtils.getWSTextAttribute(IAttributeMeta
            data.AC_OFFER_DESCRIPTION_ATTRIBUTE_NAME, null, new String[]{"description "
            + 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();
        }
    }
}

```

---

## Utilizzo di WSDL per chiamare i servizi Campaign

È possibile chiamare i servizi Campaign utilizzando il file WSDL dei servizi web di Campaign: CampaignServices30.wsdl.

Il file CampaignServices30.wsdl si trova all'indirizzo:

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

oppure nella distribuzione Campaign in:

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

L'applicazione client Java deve utilizzare le classi e gli stub generati dal WSDL utilizzando un qualsiasi strumento di conversione WSDL-Java di terze parti. IBM consiglia l'utilizzo di Apache AXIS.

I javadoc che sono stati creati da stub e le classi generate da WSDL utilizzando Apache AXIS2 si trovano in:

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

**Nota:** Tutti i file .jar dipendenti devono essere inclusi nel percorso classi Java. Il file CampaignServicesClient30.jar dipende dai file .jar del motore Apache AXIS2 SOAP e da altri file .jar Apache comuni, che si trovano nel file Campaign.war in `<CAMPAIGN_HOME>/Campaign.war`. Estrarre i file .jar da Campaign.war e includerli nel percorso classi Java.

---

## Considerazioni sulle prestazioni

Il profilo delle prestazioni dell'attuale implementazione dell'API di servizi Campaign è simile a quello dell'applicazione tramite GUI. Alcune API sono progettate esplicitamente per le prestazioni. In particolare, l'API `listCampaignsByPage()` consente una paginazione relativamente efficiente.

L'interfaccia SOAP, per sua natura, introduce latenza e sovraccarico, poiché tutti i dati vengono convertiti in formato XML, che in alcuni casi è piuttosto ridondante. Ad esempio, una semplice chiamata di loopback SOAP può richiedere 100 ms su una normale rete (Java 1.4.x era anche più lento). L'API viene ottimizzata per un portale tipico e altri casi d'uso di business dell'applicazione client, ad esempio see `listOffersByPage()`, quindi le prestazioni SOAP dovrebbero essere adeguate.

Tuttavia, il client deve preoccuparsi di non imporre un onere troppo elevato sulla normale operatività dei servizi Campaign per soddisfare le richieste degli utenti web. In generale, si prevede che le esigenze di elaborazione di un utente API non superino quelle di un normale utente web IBM Campaign.





---

## Capitolo 3. Tipi di dati

L'API di servizi IBM Campaign utilizza i seguenti tipi di dati pubblici.

---

### WSReference

Un wrapper semplice che racchiude un identificativo di database:

- **componentTypeEnum**: è un tipo enumerato che indica il tipo di componente a cui si riferisce l'id. Uno dei seguenti:
  - FOLDER
  - CAMPAIGN
  - FLOWCHART
  - TCS\_CELL
  - OFFER
  - OFFER\_LIST
  - OFFER\_TEMPLATE
- *id*: un *Long*, che definisce un identificativo numerico univoco specifico del database per il riferimento.

---

### WSVersion

Un tipo di wrapper che raccoglie i diversi componenti di una versione, che includono i seguenti:

- *major*: un numero intero che definisce il numero di versione principale, ad esempio '8' per la versione 8.1.2.3.
- *minor*: un numero intero che definisce il numero di versione secondario, ad esempio '1' per la versione 8.1.2.3.
- *maintenance*: facoltativo, un numero intero che definisce il numero di manutenzione della versione, se applicabile, ad esempio '2' per la versione 8.1.2.3. Mai fornito con una versione API.
- *patch*: facoltativo, un numero intero che definisce il numero di release della patch, se applicabile, ad esempio '3' per la versione 8.1.2.3. Mai fornito con una versione API.

---

### WSServiceInfo

Un tipo di wrapper semplice che racchiude le informazioni sul servizio. Contiene i seguenti campi:

- *apiVersion*: un'istanza di *WSVersion*, che definisce la versione più recente dell'API supportata dal servizio. *apiVersion* include solo le informazioni sulla versione principale e secondaria
- *campaignVersion*: un'istanza di *WSVersion*, che definisce la versione completa dell'istanza di IBM Campaign di base.
- *name*: nome interno del servizio, ad esempio "CampaignServices30Service".

---

## WSAttributeTypeEnum

Un tipo enumerato che definisce tutti i possibili tipi di attributo, uno tra:

- STANDARD: attributo standard o di base definito da Campaign.
- CUSTOM: un attributo personalizzato definito da un'altra applicazione IBM , dal cliente o da terze parti.
- INPUT\_PARAMETER: un parametro di input, ad esempio un attributo utilizzato per eseguire un diagramma di flusso IBM Campaign.
- OUTPUT\_PARAMETER: un parametro di output, ad esempio un attributo il cui valore viene completato come risultato di un diagramma di flusso eseguito in IBM Campaign.

---

## WSAttributeStatusEnum

Un'enumerazione di tutti i possibili codici di stato dell'attributo, uno tra:

- ACTIVE: l'attributo è attivo e può essere utilizzato.
- RETIRED: l'attributo è stato rimosso dal servizio e non deve essere utilizzato.

---

## WSAccessTypeEnum

Un tipo enumerato che definisce tutti i possibili tipi di accesso al valore dell'attributo, uno tra:

- READ\_ONLY: il valore dell'attributo può essere letto e visualizzato, ma non modificato.
- READ\_WRITE: il valore dell'attributo può essere scritto, visualizzato e modificato.

L'accesso agli attributi è in aggiunta alle autorizzazioni di sicurezza. Ad esempio, se la politica di sicurezza per l'utente client nega l'accesso in lettura ad un particolare attributo, l'accesso all'attributo non può sovrascrivere tale impostazione di sicurezza. In pratica, l'API non potrà mai restituire l'attributo al client.

---

## WSSelectTypeEnum

Definisce tutti i possibili tipi di selezione per un valore di attributo particolare, uno tra:

- NONE: nessuna selezione (*hasOptions* è false).
- SINGLE\_SELECT: è possibile scegliere solo una opzione di attributo alla volta dall'elenco delle opzioni possibili (valido solo per un attributo con *hasOptions*).
- MULTIPLE\_SELECT: simile a SINGLE\_SELECT, tranne che è possibile selezionare una o più opzioni alla volta.

---

## WSRunStatusEnum

Un tipo enumerato di tutti i possibili stati di esecuzione del diagramma di flusso, ramo o cella, uno tra:

- NOT\_STARTED: l'esecuzione è stata pianificata, ma non è ancora iniziata.
- RUNNING: l'esecuzione è in corso.
- CANCELLED: l'esecuzione è stata annullata, da un utente Campaign o da questa API.
- SUCCEEDED: l'esecuzione è stata completata correttamente.

- **FAILED:** l'esecuzione non è riuscita; i dettagli dell'errore sono riportati separatamente. Consultare "WSRunResults" a pagina 19.

---

## WSRunTypeEnum

Un tipo enumerato di tutti i tipi di esecuzione possibili, uno tra:

- NOT\_RUN
- TEST\_RUN
- PRODUCTION\_RUN
- RUN\_SKIPPED
- TEST\_FLOWCHART
- PRODUCTION\_FLOWCHART
- TEST\_BRANCH
- PRODUCTION\_BRANCH
- TEST\_PROCESS
- PRODUCTION\_PROCESS

---

## WSAttribute

Gli attributi forniscono un meccanismo semplice e estensibile per collegare dati arbitrari alle istanze dei componenti accessibili tramite l'API, sia dati standard come il *nome* della campagna, che parametri di input dell'esecuzione del diagramma di flusso come il *genere* o dati personalizzati arbitrari specificati da un'altra applicazione IBM o da un cliente IBM .

**Nota:** In questa API, gli attributi vengono utilizzati per il modeling della maggior parte dei dati del componente, non solo gli attributi personalizzati di Campaign.

Normalmente ai componenti sono associati molti attributi, che sono esposti dall'API di servizi Campaign con una mappa speciale con tipo denominata *AttributeMap*. I dati dell'attributo sono rappresentati nell'API come una classe concreta con tipo, ad esempio *WSDecimalAttribute*, per gli attributi che contengono dati decimali (numerici a doppia precisione).

Ogni attributo contiene:

- *Name:* il nome univoco dell'attributo. Questo nome è utilizzato come chiave per accedere all'attributo e ai relativi metadati all'interno dell'istanza del componente. Il formato del nome non è definito; in alcuni casi è assegnato dal servizio, dal client o da un utente di IBM Campaign.

Normalmente questo nome non è il nome di visualizzazione che viene presentato ad un utente di Campaign o un utente client. Può essere standardizzato dall'API, come *uacDescription*, può essere assegnato da IBM Campaign quando vengono pubblicati i diagrammi di flusso o può essere assegnato dall'applicazione IBM o dal cliente che definisce gli attributi personalizzati. In tutti i casi, tuttavia, è garantita l'univocità del nome.

- *Metadata:* (facoltativo) informazioni sui dati dell'attributo, ad esempio tipo di dati del valore, nome di visualizzazione, descrizione, prompt, valore predefinito, tipo di selezione, lunghezza (testo), precisione (decimali), opzioni (se a selezione singola o multipla). Consultare "WSAttributeMetadata" a pagina 16.
- *Values:* un array di zero o più oggetti valore con tipo. Il campo dei valori è fornito dalla classe dell'attributo concreta; il tipo di ciascun valore deve essere

uguale e coincidere con la definizione del tipo nel campo metadati dell'attributo. Tuttavia, non tutti gli attributi supportano più valori.

Sono supportati i seguenti tipi di attributo concreto:

- **WSBooleanAttribute**: un attributo il cui valore è un booleano, ovvero *true* o *false*.
- **WSIntegerAttribute**: un valore intero (*java.lang.Long*).
- **WSDecimalAttribute**: un valore numerico decimale a doppia precisione (*java.lang.Double*).
- **WSCurrencyAttribute**: un valore di valuta composto, che include un codice valuta ISO 4217 facoltativo, ad esempio "USD" per dollaro USA, e i valori di valuta acquisiti come *Double*. Se non viene fornito il codice valuta, si assume il valore predefinito utilizzato da IBM Campaign.  
Per un elenco di paesi, simboli di valuta e codici, consultare <http://www.xe.com/symbols.php>. La locale che viene utilizzata per un valore di valuta potrebbe essere diversa dalla locale preferita dell'utente.
- **WSCalendarAttribute**: i cui valori sono date, o date e ore, di calendario in un determinato fuso orario e locale.
- **WSTextAttribute**: una stringa di testo Unicode (anche null o vuoto).

**Nota:** L'elenco dei possibili attributi generalmente è diverso per ciascun tipo di componente, ma gli elenchi possono sovrapporsi.

---

## WSAttributeMetadata

WSAttributeMetadata definisce le informazioni sui dati di un particolare attributo con tipo, ad esempio tipo di dati del valore, testo localizzato (nome di visualizzazione, descrizione, prompt), valore predefinito, intervallo di valori ammessi, tipo di selezione, opzioni (se a selezione singola o multipla). Analogamente agli attributi, i metadati degli attributi sono con tipo. Ad esempio, un *WSDecimalAttribute myNumber* deve avere un *DecimalAttributeMetadata* collegato, e tutti i valori, inclusi i valori dell'attributo, il valore predefinito dei metadati e i possibili valori di opzioni, sono tutti di tipo *Double*.

I testi di descrizioni, etichette e altri metadati di attributi sono localizzati; tuttavia, il testo specificato dall'utente può essere disponibile solo come è stato immesso dall'utente. Ogni chiamata API include una locale richiesta che il codice client può utilizzare per definire la locale in cui uno specifico utente desidera visualizzare i messaggi localizzati. Le normali politiche di fallback Java sono utilizzate per soddisfare la richiesta.

WSAttributeMetadata contiene i seguenti campi:

- *name*: il nome dell'attributo, standard o personalizzato; è anche il nome utilizzato dall'attributo che si collega a questi metadati. Gli attributi standard sono definiti dal sistema e presentano nomi standard in uno spazio nomi riservato (ovvero, utilizzano il prefisso "uac"), i nomi personalizzati possono utilizzare qualsiasi altra convenzione di denominazione.

**Nota:** Il nome dell'attributo deve essere univoco, non è mai localizzato ed ha restrizioni di lunghezza (che dipendono dal contenuto di caratteri e dal database). Il nome non effettua la distinzione tra maiuscole e minuscole e può essere composto da una combinazione di cifre e lettere Unicode, oltre che dal carattere di sottolineatura '\_', ma non può iniziare con un cifra.

- *description*: facoltativo, descrizione dell'attributo. Adatto per un suggerimento o altra presentazione dell'interfaccia utente.
- *Predicati*: diversi predicati che descrivono l'attributo:
  - *isRequired*: true se l'attributo è obbligatorio.
  - *isInternal*: true se l'attributo è definito dal sistema ed è solo per uso interno (non deve essere presentato ad un utente).
  - *isGenerated*: true se il valore o i valori dell'attributo sono generati automaticamente da IBM Campaign quando si crea il componente, ad esempio un codice cella obiettivo. Generalmente, *accessTypeEnum* sarà *READ\_ONLY* per i valori generati.
  - *hasOptions*: true se l'attributo presenta delle opzioni. Implica che per questi metadati sono state definite opzioni e che *selectTypeEnum* è *SINGLE\_SELECT* o *MULTIPLE\_SELECT*.
- *typeEnum*: un *WSAttributeTypeEnum* che definisce il tipo di attributo, ad esempio *STANDARD* o *CUSTOM*.
- *statusEnum*: un *WSAttributeStatusEnum* che definisce lo stato dell'attributo, ad esempio *ACTIVE*.
- *accessTypeEnum*: un *WSAccessTypeEnum* che definisce il tipo di accesso al valore dell'attributo, ad esempio *READ\_ONLY*.
- *selectTypeEnum*: un *WSAccessTypeEnum* che definisce il tipo di selezione utilizzato per l'attributo, ad esempio *SINGLE*. Deve essere *NONE* per attributi della campagna e della cella, o se non sono fornite opzioni.
- *componentTypeEnum*: un *WSComponentTypeEnum* di tutti i possibili componenti Campaign esposti dall'API, ad esempio *CAMPAIGN*, *FOLDER*.
- *defaultValue* (solo diagrammi di flusso): facoltativo, valore predefinito del tipo per l'attributo. Questo valore viene fornito dalla classe di metadati dell'attributo concreta, ad esempio il valore predefinito di *WSTextAttributeMetadata* è di tipo *String*. (Fare riferimento alla descrizione dei valori dell'attributo). Per componenti diversi dai diagrammi di flusso, il valore predefinito non è definito.
- *options*: facoltativo, elenco di opzioni di questo attributo. Nel loro insieme le opzioni dell'attributo definiscono l'insieme preciso dei valori ammessi per tale attributo; ogni opzione è con tipo, ad esempio un *WSTextAttributeMetadata* può avere solo una *WSTextAttributeOption* collegata.

**Nota:** Esiste una restrizione sulle opzioni; sono supportati solo gli attributi testo.

Per ogni opzione sono definiti:

- *prompt*: prompt per l'opzione adatto ai menu a discesa, ad esempio "Maschile" per un'opzione dell'attributo genere. A differenza dei prompt di metadati, i nomi di visualizzazione dell'opzione normalmente non includono la punteggiatura.
- *description*: descrizione localizzata dell'opzione, ad esempio "Una persona di genere maschile". Adatto per il testo di un suggerimento.
- *isDefault*: true se questa particolare opzione è il valore predefinito. Per i tipi di selezione *MULTIPLE\_SELECT*, più di una opzione può essere contrassegnata come predefinita.
- *value*: il valore dell'opzione con tipo. Come per il *defaultValue* dei metadati dell'attributo, questo valore viene fornito dalla sottoclasse concreta dell'opzione, ad esempio un valore di *WSDecimalAttributeOption* è di tipo *Decimal*. (Fare riferimento alla descrizione dei valori dell'attributo). Riprendendo l'esempio di *genere* precedente, il valore può essere dichiarato come stringa "m" (*WSTextAttributeOption*) o come codice numerico 123 (*WSDecimalAttributeOption*).

---

## WSCampaignInfo

Un tipo di wrapper semplice che racchiude i dati di attributo della campagna.

Contiene i seguenti campi:

- *reference*: il riferimento della campagna.
- *name*: il nome della campagna (*uacName*); non è garantita l'univocità.
- *description*: facoltativo, descrizione della campagna (*uacDescription*).
- *campaignCode*: il codice della campagna univoco (*uacCampaignCode*); assegnato dall'utente o da Campaign.

---

## WSComponentOrFolderInfo

Contiene una combinazione di dati di attributo della campagna o della cartella in un wrapper, ad esempio nome di visualizzazione, il relativo riferimento.

Contiene i seguenti campi:

- *reference*: il riferimento del componente o della cartella.
- *name*: nome del componente o della cartella (*uacName*); non è garantita l'univocità.
- *description*: facoltativo, descrizione del componente o della cartella (*uacDescription*).
- *componentCode*: codice univoco per il componente, o null se è una cartella.

---

## WSTargetCellInfo

Un wrapper semplice che racchiude i dati dell'attributo di riga della cella obiettivo:

Contiene i seguenti campi:

- *reference*: il riferimento alla cella.
- *name*: il nome della cella (*uacName*); non è garantita l'univocità.
- *description*: facoltativo, descrizione della cella (*uacDescription*).
- *cellCode*: il codice cella (*uacCellCode*); assegnato dall'utente o da Campaign. L'univocità dei codici cella può essere forzata impostando il parametro di configurazione *DuplicateCellCodesAllowed* di IBM Campaign su false.
- *flowchartName*: facoltativo, nome del diagramma di flusso a cui è collegata la cella.

---

## WSMetricsInfo

Un tipo di wrapper semplice che racchiude i dati analitici della campagna, incluso il numero di contatti. Contiene i seguenti campi:

- *totalContacts*: un long che indica il numero totale di contatti.
- *responses*: un elenco con tipi delle istanze di *WSMetricsResponse*, ogni istanza definisce le informazioni sul contatto per una risposta:
  - *typeCode*: una stringa che definisce il codice del tipo di risposta, ad esempio *PHC* per un contatto telefonico.
  - *count*: un long che indica il numero di volte per cui si è verificato il contatto.

---

## WSRunResults

Un tipo di wrapper che racchiude i risultati di un'esecuzione di diagramma di flusso, casella del processo o cella, eventualmente ancora in corso; include lo stato dell'esecuzione, la data e ora di inizio e di fine dell'esecuzione del diagramma di flusso e i conteggi.

Contiene i seguenti campi:

- *sourceReference*: facoltativo, riferimento all'origine dei risultati dell'esecuzione. In base al contesto in cui sono estratti i risultati dell'esecuzione, può fare riferimento a un diagramma di flusso, una casella del processo del diagramma di flusso o una cella obiettivo. In ogni caso, i rimanenti dati dei risultati dell'esecuzione fanno riferimento a questa origine.
- *flowchartName*: il nome del diagramma di flusso che è stato eseguito.
- *flowchartId*: l'identificativo di database per il diagramma di flusso.
- *runId*: l'identificativo di database dell'esecuzione.
- *typeEnum*: un tipo enumerato che definisce quale esecuzione ha generato i risultati, ad esempio PRODUCTION\_PROCESS (vedere *WSRunTypeEnum*).
- *statusEnum*: un tipo enumerato che definisce lo stato dell'esecuzione, ad esempio RUNNING (vedere *WSRunStatusEnum*).
- *statusCode*: facoltativo, codice di stato numerico.
- *statusMessage*: facoltativo, messaggio di stato.
- *startDate*: facoltativo, data e ora del calendario in cui è iniziata l'esecuzione; è null se l'esecuzione non è iniziata.
- *endDate*: associata a *startDate*, la data e ora in cui è terminata l'esecuzione (con esito positivo o negativo); è null se l'esecuzione non è iniziata o non è ancora terminata.
- *count*: facoltativo, conteggio totale dei contatti selezionati dall'esecuzione; può essere zero o null se l'esecuzione non è stata completata.

---

## WSOfferInfo

Un tipo di wrapper semplice che racchiude i dati di attributo dell'offerta o elenco offerte.

Contiene i seguenti campi:

- *reference*: il riferimento dell'offerta o dell'elenco offerte.
- *name*: nome dell'offerta o elenco offerte (*uacName*); non è garantita l'univocità.
- *description*: descrizione facoltativa (*uacDescription*).
- *offerCode*: il codice offerta (*uacOfferCode*) se è un'offerta, o null se è un elenco offerte. Non è garantita l'univocità.

---

## WSOfferCodeOrName

Un tipo di wrapper semplice che racchiude i dati dei codici offerta o i nomi dell'elenco offerte.

Contiene i seguenti campi:

- *isCode*: booleano che indica se il campo *codeOrName* è un codice offerta (true) o il nome di un elenco offerte (false).

- *codeOrName*: il codice offerta univoco (*uacOfferCode*) di un'offerta o il nome dell'elenco offerte.

---

## WSOfferValidationInfo

Un tipo di wrapper semplice che racchiude le informazioni sulla convalida dell'offerta.

Contiene i seguenti campi:

- *errorCode*: se non è null, definisce il codice di errore di convalida alfanumerico. Per i codici di errore fare riferimento alla classe *IStandardDefinitions*.
- *errorMessage*: facoltativo, messaggio localizzato che descrive l'errore (se si è verificato un errore).
- *codeOrName*: il codice offerta o il nome dell'elenco offerte convalidato.
- *reference*: il riferimento dell'offerta o dell'elenco offerte, se valido.

---

## WSOfferTemplateInfo

Un tipo di wrapper semplice che racchiude i dati sul modello di offerta.

Contiene i seguenti campi:

- *reference*: il riferimento al modello di offerta.
- *name*: nome del modello di offerta; è garantita l'univocità.
- *description*: descrizione facoltativa (*uacDescription*).
- *offerTemplateID*: l'identificativo di database del modello di offerta univoco.

---

## WSBulkOfferInfo

Utilizzato per creare offerte in massa.

Contiene i seguenti campi:

- *offerName*: il nome dell'offerta che viene creata.
- *attributes*: un array di tipi *WSAttribute* che indica gli attributi dell'offerta.

---

## WSOfferInfoStatus

Un tipo restituito dal metodo API *bulkCreateOffers()* che indica lo stato di creazione dell'offerta di massa.

Contiene i seguenti campi:

- *name*: il nome dell'offerta.
- *code*: il codice offerta. È null se la creazione dell'offerta non riesce.
- *description*: descrizione dell'offerta.
- *reference*: *WSReference* dell'offerta creata. È null se la creazione dell'offerta non riesce.
- *status*: un'istanza di *WSRequestStatus* che indica lo stato della creazione dell'offerta.



---

## Capitolo 4. Metodi API

L'API di servizi IBM Campaign utilizza i seguenti metodi API.

---

### Metodi API: Servizio

L'API di servizi IBM Campaign fornisce un modo per ottenere le informazioni di identificazione sul servizio stesso.

#### getServiceInfo

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

Restituisce le informazioni sul servizio, ad esempio la versione API più recente supportata, la versione completa dell'istanza di IBM Campaign di base.

**Nota:** Questa chiamata non richiede informazioni sul client e non sono applicate le autorizzazioni di sicurezza.

#### Parametri

Nessuno.

#### Restituzioni

Restituisce un'istanza di *WSServiceInfo*.

#### Errori

Nessuno.

---

### Metodi API: Attributi

La maggior parte dei dati di istanza del componente possono essere esposti dall'API IBM Campaign come attributi o metadati di attributi.

In alcuni casi, le definizioni dei metadati di attributi sono globali per IBM Campaign (ad esempio attributi personalizzati della campagna). In altri casi, sono limitate a un particolare componente, ad esempio le variabili utente del diagramma di flusso. Se non diversamente indicato, tutti gli attributi possono essere letti se il client dispone delle autorizzazioni di sicurezza sufficienti per farlo.

**Nota:** Solo i componenti attivi e a cui ha accesso il client vengono esposti da questa API. Il supporto pubblico è limitato ad un sottoinsieme delle API disponibili.

#### getAttributesByName

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

Estrae gli attributi indicati associati all'istanza di componente specificata (può essere vuota).

## Parametri

*userCredential*: le credenziali dell'utente client.

*requestedLocale*: facoltativo, locale da utilizzare per questa richiesta; se non fornita vengono utilizzate le preferenze della locale dell'utente IBM . Se necessario, si applica il normale algoritmo di impostazione della locale predefinita.

*partitionName*: facoltativo, nome della partizione della campagna da utilizzare. Se non definito, viene utilizzata la partizione predefinita.

*reference*: facoltativo, riferimento per l'istanza del componente che contiene gli attributi desiderati. Viene generata una *InvalidComponentException* se il riferimento non è valido o il componente non esiste.

*names*: facoltativo, array di nomi di attributi da estrarre (non i nomi di visualizzazione); se non fornito, vengono restituiti tutti gli attributi. Genera *AttributeNotFoundException* se uno degli attributi indicati non esiste.

## Restituzioni

Una mappa con tipo di zero o più attributi; il nome attributo è la chiave della voce di mappa e l'istanza dell'attributo è il valore della voce.

## Errori

*InvalidComponentException*, *AttributeNotFoundException*

*AuthorizationException*, *DataException*

**Nota:** Tutte queste eccezioni sono racchiuse in *CampaignServicesException*.

## updateAttributes

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

Aggiorna uno o più attributi dell'istanza del componente con i valori di attributo forniti.

## Logica di aggiornamento

La logica di aggiornamento è la seguente.

Per ciascun attributo contenuto nella mappa di attributi fornita:

1. Se il nome attributo corrisponde a un attributo esistente, cerca di sovrascrivere i relativi campi *values* con i campi di valori forniti.
2. Se l'attributo non esiste ancora, *allowCreate* è true e i relativi metadati sono noti, crea l'attributo. Questo si applica ai metadati di attributi globali oltre che agli attributi di istanza (tranne diagrammi di flusso).
3. Se il tipo del valore o qualche altro aspetto della definizione di metadati dell'attributo non corrisponde, o se uno o più dei valori forniti non sono validi o sono fuori intervallo, genera una *InvalidAttributeException*.
4. Altrimenti genera una *AttributeNotFoundException* se l'attributo indicato non esiste.

**Nota:** Se si verifica un'eccezione, non viene eseguito il commit di nessuno degli aggiornamenti.

Questo particolare metodo non supporta la definizione dei nuovi attributi personalizzati; utilizzare il metodo `createAttributeMetadata()` a questo scopo.

In tutti i casi, l'operazione di aggiornamento dell'attributo è soggetta ai normali vincoli di sicurezza e alla convalida. È responsabilità del client determinare quali attributi sono richiesti da una particolare istanza del componente, i tipi corretti, ecc.

### Parametri

*userCredential*: le credenziali dell'utente client.

*requestedLocale*: facoltativo, locale da utilizzare per questa richiesta.

*partitionName*: facoltativo, nome della partizione della campagna da utilizzare.

*reference*: il riferimento per l'istanza del componente che contiene gli attributi da aggiornare.

*allowCreate*: indica se deve essere creato un nuovo attributo se non esiste ancora per il componente. Consultare "Logica di aggiornamento" a pagina 22.

*attributes*: un array di attributi da aggiornare; il nome attributo viene utilizzato per individuare l'attributo da aggiornare e i nuovi valori vengono utilizzati per aggiornare il valore dell'attributo esistente come singolo oggetto del tipo appropriato o come array, se applicabile. Fare riferimento a Capitolo 5, "Eccezioni comuni", a pagina 59.

### Restituzioni

Nessuno.

### Errori

`InvalidComponentException`, `AttributeNotFoundException`,  
`InvalidAttributeException`

`AuthorizationException`, `DataException`

## getAttributeMetadataByName

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

Estrae le definizioni dei metadati di attributi indicati che sono collegati a un particolare componente, modello o definiti globalmente.

### Parametri

*userCredential*: le credenziali dell'utente client.

*requestedLocale*: facoltativo, locale da utilizzare per questa richiesta.

*partitionName*: facoltativo, nome della partizione della campagna da utilizzare.

*reference*: facoltativo, riferimento per il componente o il modello che contiene i metadati di attributi desiderati. Se viene fornito solo `ComponentTypeEnum`, l'estrazione è limitata ai componenti di quel tipo. Se il riferimento non è fornito affatto, l'estrazione restituisce tutte le definizioni di metadati globali, per tutti i tipi di componente. Genera una `InvalidComponentException` se il riferimento fornito non è valido.

*names*: facoltativo, array di nomi di metadati di attributi da estrarre. Se non fornito, vengono restituiti tutti i metadati per il componente, o quelli definiti globalmente se non viene fornito nessun riferimento. Genera una `AttributeNotFoundException` se una o più delle definizioni di metadati di attributi specificate non sono valide.

### Restituzioni

Una mappa con tipo di zero o più definizioni di metadati di attributi; il nome attributo è la chiave della voce di mappa e i metadati dell'attributo sono il valore della voce.

### Errori

`InvalidComponentException`, `AttributeNotFoundException`

`AuthorizationException`, `DataException`

## createAttributeMetadata

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

Crea una o più nuove definizioni di metadati di attributi e facoltativamente le collega a un particolare componente o modello.

### Parametri

*userCredential*: le credenziali dell'utente client.

*requestedLocale*: facoltativo, locale da utilizzare per questa richiesta.

*partitionName*: facoltativo, nome della partizione della campagna da utilizzare.

*reference*: facoltativo, riferimento per il componente o modello a cui devono essere collegati i metadati. Se non fornito, la definizione di metadati creata è globale. Se viene fornito un riferimento non valido, viene generata una `InvalidComponentException`.

*attributeMetadata*: un array di definizioni di metadati di attributi da collegare. Se uno o più dei metadati specificati sono già collegati al componente, ovvero se il nome non è univoco, viene generata una `AttributeExistsException`. Genera una `InvalidAttributeException` se si verifica un problema con uno o più dei metadati specificati, ovvero se c'è incongruenza interna.

### Restituzioni

Nessuno.

### Errori

`InvalidComponentException`, `AttributeExistsException`, `InvalidAttributeException`

`AuthorizationException`, `DataException`

## updateAttributeMetadata

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

Aggiorna una o più definizioni di metadati di attributo del componente o modello specificato, facoltativamente creando nuove definizioni dei metadati, se necessario.

### Logica di aggiornamento

La logica di aggiornamento è la seguente.

Per ciascuna definizione di metadati di attributi contenuti nell'array fornito:

1. Se il nome dell'attributo non corrisponde ai metadati esistenti collegati al componente, effettua le seguenti operazioni in base al valore del parametro *allowCreate*:
  - a. *True*: crea una nuova definizione di metadati. Funzionalmente uguale all'utilizzo della richiesta `createAttributeMetadata()`.
  - b. *False*: genera una *AttributeNotFoundException*.
2. Se il tipo di dati dei metadati di attributo è diverso, genera una *InvalidAttributeException*.
3. Cerca di sovrascrivere la definizione di metadati di attributo esistente con i valori dei campi dei metadati forniti, altrimenti genera una *InvalidAttributeException*. Sono supportati solo i seguenti aggiornamenti (altrimenti viene generata una *InvalidAttributeException*):
  - a. *name*: non può essere modificato (name è la chiave!).
  - b. *displayName*: accetta il nuovo valore.
  - c. *description*: accetta il nuovo valore.
  - d. *isRequired*: consente solo la modifica da *true* a *false*.
  - e. *isInternal*: accetta il nuovo valore.
  - f. *isGenerated*: nessuna modifica consentita.
  - g. *attributeTypeEnum*: nessuna modifica consentita.
  - h. *accessTypeEnum*: accetta il nuovo valore.
  - i. *selectTypeEnum*: accetta queste transizioni se vengono fornite le opzioni:
    - 1) Da *NONE* a *SINGLE\_SELECT* o *MULTIPLE\_SELECT*
    - 2) Da *SINGLE\_SELECT* a *MULTIPLE\_SELECT*
  - j. *options*: le opzioni possono essere aggiunte, ma non eliminate. Sono supportate solo le seguenti modifiche di opzione (corrispondenza per valore):
    - 1) *displayName*: accetta il nuovo valore (nessun effetto a catena).
    - 2) *description*: accetta il nuovo valore (nessun effetto a catena).
    - 3) *isDefault*: accetta il nuovo valore; tuttavia deve corrispondere a *SelectTypeEnum*.
    - 4) *value*: nessuna modifica consentita (value è la chiave!).
  - k. *defaultValue* (solo diagrammi di flusso): accetta il nuovo valore predefinito.
  - l. *maxLength* (solo testo): accetta la nuova lunghezza se maggiore.
4. Se la definizione dei metadati di attributo non è congruente internamente, genera una *InvalidAttributeException*.

5. Se necessario, individua tutte le istanze del componente che fanno riferimento ai metadati di attributo aggiornati e aggiorna come appropriato.

**Nota:** Se si verifica un'eccezione, non viene eseguito il commit di nessuno degli aggiornamenti.

In tutti i casi, l'operazione di aggiornamento dell'attributo è soggetta ai normali vincoli di sicurezza e alla convalida.

Vedere `createAttributeMetadata()`, `deleteAttributeMetadata()`

## Parametri

*userCredential*: le credenziali dell'utente client.

*requestedLocale*: facoltativo, locale da utilizzare per questa richiesta.

*partitionName*: facoltativo, nome della partizione della campagna da utilizzare.

*reference*: facoltativo, riferimento per l'istanza del componente che contiene gli attributi desiderati. Se non fornito, l'aggiornamento è limitato alle definizioni di metadati globali. Genera una *InvalidComponentException* se il riferimento fornito non è valido.

*allowCreate*: se true, le definizioni di metadati attualmente non esistenti vengono create (funzionalmente equivalente all'uso del metodo `createAttributeMetadata()`).

*attributeMetadata*: un array di definizioni di metadati di attributi da aggiornare (e aggiungere se l'indicatore *allowCreate* è true). Il nome attributo viene utilizzato per individuare la definizione di metadati da aggiornare e i rimanenti dati sono utilizzati per aggiornare la definizione esistente. Fare riferimento a "Logica di aggiornamento" a pagina 25.

## Restituzioni

Nessuno.

## Errori

*InvalidComponentException*, *InvalidAttributeException*

*AuthorizationException*, *DataException*

## deleteAttributeMetadata

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

Elimina una o più definizioni di metadati di attributi indicate dal componente specificato, dal modello specificato (solo metadati di attributi personalizzati), o le definizioni di metadati di attributi globali.

Come parte di questa attività, il metodo individua tutti i componenti che fanno riferimento ai metadati eliminati e li aggiorna, come richiesto.

**Nota:** Tuttavia, se si verifica un'eccezione, non viene eseguito il commit delle eliminazioni.

## Parametri

*userCredential*: le credenziali dell'utente client.

*requestedLocale*: facoltativo, locale da utilizzare per questa richiesta.

*partitionName*: facoltativo, nome della partizione della campagna da utilizzare.

*reference*: facoltativo, riferimento del componente o del modello che contiene gli attributi da eliminare. Se non fornito, l'eliminazione è limitata alle definizioni di metadati globali. Genera una *InvalidComponentException* se il riferimento fornito non è valido.

**Nota:** Se l'array facoltativa di nomi dei metadati di attributi non viene fornita, questo metodo cerca di eliminare tutti i metadati di attributi personalizzati associati al componente, oppure tutte le definizioni globali se non era stato fornito il riferimento.

*names*: facoltativo, array di nomi di metadati di attributi da eliminare. Genera una *AttributeNotFoundException* se uno o più dei metadati di attributi indicati non esistono. Genera *InvalidAttributeException* se non è stato possibile rimuovere un attributo.

## Restituzioni

Nessuno.

## Errori

*InvalidComponentException*, *AttributeNotFoundException*,  
*InvalidAttributeException*

*AuthorizationException*, *DataException*

---

## Metodi API: Campagne e diagrammi di flusso

L'API IBM Campaign supporta le seguenti operazioni sulle campagne e i diagrammi di flusso (soggette alle autorizzazioni di sicurezza):

- creazione di una campagna
- rilevamento (creazione elenchi di campagne in base a criteri diversi)
- creazione, lettura e aggiornamento degli attributi (tramite le API di attributi)
- arresto dell'esecuzione del diagramma di flusso

Le campagne dispongono di numerosi attributi standard ad esse associati, che sono esposti dall'API. Questo elenco può essere esteso a piacimento dal client aggiungendo attributi personalizzati (vedere le API di attributi).

Gli attributi standard della campagna sono elencati di seguito:

- *uacName*: nome della campagna (non è garantita l'univocità).
- *uacDescription*: facoltativo, stringa che descrive la campagna.
- *uacCampaignCode*: una stringa di codice che identifica in modo univoco la campagna. Generalmente generato da Campaign, ma può essere fornito dal client.
- *uacCreateDate*: un'istanza di calendario che indica la data e ora in cui la campagna è stata creata dal server.
- *uacUpdateDate*: un'istanza di calendario che indica la data e ora dell'ultimo aggiornamento della campagna da parte del server.

- *uacInitiative*: facoltativo, stringa che definisce l'iniziativa della campagna.
- *uacObjectives*: facoltativo, stringa che identifica gli obiettivi della campagna.
- *uacStartDate*: facoltativo, un'istanza di calendario che fornisce la data e ora in cui la campagna è stata avviata dal server, oppure in cui è stato pianificato l'avvio.
- *uacEndDate*: analogo a *uacStartDate*, ma definisce la data e ora in cui è stata completata la campagna oppure è stato pianificato il completamento. Deve essere successivo a *uacStartDate*.
- *uacLastRunDate*: facoltativo, un'istanza di calendario che indica la data e ora dell'ultima esecuzione di un diagramma di flusso collegato alla campagna (null se non eseguito).
- *uacExternalLinkOwner*: facoltativo, una stringa che definisce il nome del proprietario di un link esterno (vedere l'attributo *uacExternalLinkReference*). Solo uso IBM ; deve essere uno tra:
  - "Plan" (ora noto come IBM Marketing Operations)
  - "Collaborate" (ora noto come IBM Distributed Marketing)
- *uacExternalLinkId*: facoltativo, un identificativo di database numerico che viene assegnato da un'altra applicazione IBM a un oggetto collegato a questa campagna. Solo uso IBM : vedere anche l'attributo *uacExternalLinkOwner*.

## generateCampaignCode

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

Genera un nuovo codice della campagna.

L'univocità di questo codice è garantita ed è diverso dal valore restituito da una precedente o successiva chiamata a questo metodo, o al metodo `createCampaign()`, o dal valore generato per una campagna creata dalla GUI di IBM Campaign.

**Nota:** L'utilizzo di questo metodo è facoltativo, poiché l'API `createCampaign()` genera un codice della campagna per il client, se non ne è stato fornito uno.

Vedere `createCampaign()`.

### Parametri

*userCredential*: le credenziali dell'utente client.

*requestedLocale*: facoltativo, locale da utilizzare per questa richiesta.

*partitionName*: facoltativo, nome della partizione della campagna da utilizzare. Se nell'installazione di Campaign è presente solo una partizione, questo argomento può essere null.

### Restituzioni

Il codice della campagna generato.

### Errori

AuthorizationException, DataException

## deleteCampaigns

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



Elimina le campagne specificate dal sistema.

## Parametri

*userCredential*: le credenziali dell'utente client.

*partitionName*: facoltativo, nome della partizione della campagna da utilizzare.

*requestedLocale*: facoltativo, locale da utilizzare per questa richiesta.

*wsReference*: riferimenti delle campagne da eliminare.

## Restituzioni

Restituisce un oggetto di tipo `WSDeleteCampaignsResponse`.

## Errori

Genera una `CampaignServicesException` se la campagna non esiste oppure se il riferimento non è valido o non è stato fornito.

## createCampaign

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

Crea una nuova campagna per il client, partizione e `securityPolicyName`, applicando gli attributi specificati. Tutte le campagne create da questa API sono sotto la cartella root.

## Parametri

*userCredential*: le credenziali dell'utente client.

*requestedLocale*: facoltativo, locale da utilizzare per questa richiesta.

*partitionName*: facoltativo, nome della partizione della campagna da utilizzare.

*securityPolicyName*: facoltativo, nome della politica di sicurezza della campagna da utilizzare per creare la campagna. Tutte le successive operazioni su questa campagna utilizzeranno questa politica. Se non definito, viene utilizzata la politica globale.

*name*: il nome da assegnare alla nuova istanza della campagna (il relativo attributo `uacName`).

*attributes*: facoltativo, un array di attributi di inizializzazione; tutti gli attributi forniti sovrascrivono i valori predefiniti della campagna; gli altri restano invariati. Ad esempio, se viene fornito un attributo `uacCampaignCode`, questo viene utilizzato al posto di uno generato automaticamente. È il client a determinare gli attributi che sono richiesti dalla campagna, i relativi tipi, ecc.

Genera una *AttributeNotFoundException* se uno o più degli attributi indicati non esistono o *InvalidAttributeException* se un valore di attributo non è valido (ad esempio un tipo di dati non corretto).

### Restituzioni

Una singola istanza di *CampaignInfo* per la campagna creata.

### Errori

*InvalidAttributeException*, *AttributeNotFoundException*

*AuthorizationException*, *DataException*

## listCampaignsByPage

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

Elenca una "pagina" di campagne che corrispondono ai valori degli attributi facoltativi, a partire dall'offset di pagina specificato. Le cartelle vengono ignorare.

Una volta richiamato, ciascun *CampaignInfo* restituito può essere utilizzato direttamente, ad esempio per visualizzare un elenco di riepilogo, oppure è possibile utilizzare i metodi di attributo per estrarre o aggiornare gli attributi della campagna.

Questa API non gestisce alcuno stato, quindi è possibile effettuare chiamate in qualsiasi ordine.

### Parametri

*userCredential*: le credenziali dell'utente client.

*requestedLocale*: facoltativo, locale da utilizzare per questa richiesta.

*partitionName*: facoltativo, nome della partizione della campagna da utilizzare.

*attributes*: facoltativo, array di attributi da confrontare; il nome, il tipo di dati e i valori dell'attributo sono utilizzati per determinare la corrispondenza; se l'attributo supporta gli array, tutti i valori specificati devono corrispondere. L'operatore di corrispondenza implicito è un AND, quindi vengono restituite solo le campagne che corrispondono a tutti i valori di attributo forniti.

Genera una *AttributeNotFoundException* se un nome attributo non esiste o una *InvalidAttributeException* se uno o più degli attributi forniti non sono validi.

*pageOffset*: l'offset iniziale di tutte le possibili campagne da cui iniziare l'enumerazione (con valore da zero). Ad esempio, se l'enumerazione comprende 1000 campagne e questo valore è impostato su 10, la pagina inizierebbe dall'undicesimo componente. Viene generata una *RangeException* se l'offset fornito è fuori intervallo.

*pageSize*: il numero massimo di campagne corrispondenti da restituire per la pagina (non può superare 500).

## Restituzioni

Un elenco con tipo di zero o più istanze del wrapper dati *CampaignInfo*, una per ogni campagna corrispondente nella pagina.

## Errori

*AttributeNotFoundException*, *InvalidAttributeException*, *RangeException*

*InvalidExecutionContextException*, *AuthorizationException*

## stopFlowchart

`stopFlowchart(int pid, int runid)`

Questa API arresta un diagramma di flusso in esecuzione. Per una configurazione a listener singolo, una esecuzione del diagramma di flusso può essere identificata univocamente dal PID associato all'esecuzione del diagramma di flusso. Il PID indica l'ID processo del processo *unica\_acsvr*. Se sono configurati più listener di IBM Campaign, è necessario includere sia l'ID esecuzione associato all'esecuzione del diagramma di flusso che il PID.

## Parametri

*pid*: ID processo del processo *unica\_acsvr* associato a una esecuzione del diagramma di flusso.

*runid*: ID esecuzione associato a una esecuzione del diagramma di flusso. Parametro obbligatorio per una configurazione listener con cluster. Parametro facoltativo se è configurato un solo listener.

## Restituzioni

Nessuno

## Errori

Nessuno

---

## Metodi API: Celle obiettivo

Le celle obiettivo sono un'astrazione per alcuni sottoinsiemi noti dei risultati della campagna che sono gestiti da IBM Campaign come foglio di calcolo delle celle obiettivo (TCS). Le celle obiettivo possono essere globali per una campagna o possono essere associate a un particolare diagramma di flusso della campagna.

L'API IBM Campaign supporta le seguenti operazioni sulle celle obiettivo:

- creazione di una o più nuove celle obiettivo globali
- aggiornamento di massa di una o più celle obiettivo esistenti
- rilevamento (creazione elenco delle celle obiettivo)
- creazione, lettura e aggiornamento degli attributi (tramite le API di attributi)
- eliminazione di una cella obiettivo esistente
- estrazione dei risultati di esecuzione associati a una o più celle

Le celle obiettivo dispongono di numerosi attributi standard ad esse associati, che sono esposti dall'API. Questo elenco può essere esteso a piacimento dal client aggiungendo le definizioni dei metadati degli attributi personalizzati (vedere le

API di attributi). Si può pensare ai metadati di ogni attributo come a una colonna nel TCS; il layout del foglio di calcolo è definito dal client.

Gli attributi standard delle celle obiettivo sono:

- *uacName*: nome della cella.
- *uacDescription*: facoltativo, stringa che descrive il diagramma di flusso.
- *uacCellCode*: una stringa di codice che identifica in modo univoco la cella. Generalmente generato automaticamente da Campaign, ma può essere fornito dal client.
- *uacCreateDate*: un'istanza di calendario che indica la data e ora in cui la cella è stata creata dal server.
- *uacUpdateDate*: un'istanza di calendario che indica l'ultima volta in cui la cella è stata aggiornata dal server.
- *uacIsControl*: un booleano che indica se questa è una cella di controllo (true) o non lo è (false). Le altre celle possono fare riferimento a questa cella come cella di controllo (vedere *uacControlCell*).
- *uacControlCell*: facoltativo, riferimento della cella di controllo (non consentito per una cella di controllo). Vedere l'attributo *uacIsControl*.
- *uacIsApproved*: un booleano che indica se la cella è approvata (true) o non lo è (false).
- *uacIsReadOnly*: un booleano che indica se la cella è di sola lettura (true) o non lo è (false).
- *uacDisplayOrder*: un numero intero che indica l'ordine di questa cella (riga) relativamente ad altre nel foglio di calcolo delle celle obiettivo.
- *uacIsTopDown*: un booleano che indica se la cella è top-down.
- *uacAssignedOffers*: facoltativo, un array di uno o più riferimenti delle offerte o elenchi offerte assegnati a questa cella (non consentito per una cella obiettivo).
- *uacFlowchartName*: facoltativo, nome del diagramma di flusso a cui è collegata questa cella (è necessario impostarlo di sola lettura tramite la GUI di IBM Campaign; non consentito per una cella di controllo).
- *uacFlowchartId*: facoltativo, identificativo di database per il diagramma di flusso a cui è collegata questa cella (di sola lettura come il precedente).

## createTargetCell

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

Crea una nuova riga di cella obiettivo specifica della campagna, applicando gli attributi per cella specificati e le informazioni utente.

Gli attributi specificati possono essere standard o personalizzati; tuttavia, se personalizzati, le corrispondenti definizioni di metadati degli attributi globali devono esistere.

Dopo che la cella obiettivo è stata creata, i valori degli attributi possono essere modificati utilizzando le API di attributi.

Vedere `listTargetCells()`, `bulkCreateTargetCells()`.

Vedere `createAttributeMetadata()`, `listAttributeMetadata()`,  
`getAttributesByName()`

## Parametri

*userCredential*: le credenziali dell'utente client.

*requestedLocale*: facoltativo, locale da utilizzare per questa richiesta.

*partitionName*: facoltativo, nome della partizione della campagna da utilizzare.

*campaignReference*: il riferimento della campagna che contiene il foglio di calcolo delle celle obiettivo (TCS) da aggiornare. Accumula una *InvalidComponentException* se la campagna non esiste o il riferimento non è valido.

*attributes*: facoltativo, array di attributi TCS per la nuova cella. Ogni elemento di attributo fornito sovrascriverà i corrispondenti valori predefiniti dell'attributo della cella; gli altri restano invariati. È il client a determinare gli attributi che sono richiesti dalla cella, i relativi tipi, ecc. Accumula una *InvalidAttributeException* se si verifica un problema con l'attributo specificato.

Se vengono accumulate delle eccezioni, questo metodo genera una *CompositeException* e tutte le operazioni di creazione sono annullate. L'elenco di cause dell'eccezione include un'eccezione per ogni attributo che ha causato l'errore e include un indice numerico invece di *reference*, il nome dell'attributo, e normalmente il valore all'origine del problema. L'elenco di cause è ordinato come l'*attributeList* di input.

## Restituzioni

Un wrapper di dati *TargetCellInfo* per la cella di TCS creata.

## Errori

*InvalidComponentException*, *CompositeException*

*AuthorizationException*, *DataException*

## bulkCreateTargetCells

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

Crea molte nuove righe di celle obiettivo specifiche della campagna in una sola volta, applicando gli attributi per cella specificati e le informazioni utente.

Gli attributi specificati possono essere standard o personalizzati; tuttavia, se personalizzati, le corrispondenti definizioni di metadati degli attributi globali devono esistere.

Dopo che la cella obiettivo è stata creata, i valori degli attributi possono essere modificati utilizzando le API di attributi.

Vedere `listTargetCells()`.

Vedere `createAttributeMetadata()`, `listAttributeMetata()`,  
`getAttributesByName()`

### Parametri

*userCredential*: le credenziali dell'utente client.

*requestedLocale*: facoltativo, locale da utilizzare per questa richiesta.

*partitionName*: facoltativo, nome della partizione della campagna da utilizzare.

*campaignReference*: il riferimento della campagna che contiene il foglio di calcolo delle celle obiettivo (TCS) da aggiornare. Accumula una *InvalidComponentException* se la campagna non esiste o il riferimento non è valido.

*attributeList*: facoltativo, elenco di array di attributi per cella, uno per ciascuna cella obiettivo da creare. Tutti gli attributi forniti per un particolare elemento dell'elenco sovrascrivono i corrispondenti valori predefiniti dell'attributo della cella; gli altri restano invariati. È il client a determinare gli attributi che sono richiesti dalla cella, i relativi tipi, ecc. Accumula una *InvalidAttributeException* se si verifica un problema con l'attributo specificato.

Se vengono accumulate delle eccezioni, questo metodo genera una *CompositeException* e tutte le operazioni di creazione sono annullate. L'elenco di cause dell'eccezione include un'eccezione per ogni attributo che ha causato l'errore e include un indice numerico invece di *reference*, il nome dell'attributo, ecc. L'elenco di cause è ordinato come l'*attributeList* di input.

### Restituzioni

Un elenco di wrapper di dati *TargetCellInfo*, uno per ciascuna istanza creata, ordinati in base all'ordine dell'elemento del parametro di input *attributesList*.

### Errori

*InvalidComponentException*, *CompositeException*

*AuthorizationException*, *DataException*

## listTargetCells

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

Elenca le informazioni su tutte le celle obiettivo esistenti che corrispondono agli attributi specificati, per la campagna specificata o globalmente, se non è stata specificata nessuna campagna.

Vedere `getAttributeMetadata()`, `getAttributesByName()`.

### Parametri

*userCredential*: le credenziali dell'utente client.

*requestedLocale*: facoltativo, locale da utilizzare per questa richiesta.

*partitionName*: facoltativo, nome della partizione della campagna da utilizzare.

*campaignReference*: riferimento della campagna padre. Genera una *InvalidComponentException* se la campagna non esiste o il riferimento non è valido.

*attributes*: facoltativo, array di attributi da confrontare. L'operatore di corrispondenza implicito è un AND, quindi vengono restituite solo le celle che corrispondono a tutti i valori di attributo forniti.

Genera una *InvalidAttributeException* se uno o più degli attributi specificati non sono validi.

## Restituzioni

Restituisce un elenco di zero o più istanze di *TargetCellInfo* per le celle corrispondenti.

## Errori

*InvalidComponentException*, *InvalidAttributeException*

*AuthorizationException*, *DataException*

## bulkUpdateTargetCells

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

Aggiorna gli attributi di una o più celle obiettivo.

La logica di aggiornamento è la seguente.

Per ciascun elemento nella *attributesMap* fornita, la chiave della voce è il riferimento della cella obiettivo da aggiornare e il valore della voce è un array di attributi di aggiornamento per tale cella. Se la cella obiettivo non esiste, accumula una *InvalidComponentException*.

Dopo aver individuato la cella obiettivo, per ciascun attributo specificato, esegue le seguenti operazioni:

1. Se un nome attributo corrisponde a un attributo esistente, cerca di sovrascrivere i relativi campi di valori con i campi di valori forniti.
2. Se il tipo del valore o qualche altro aspetto della definizione di metadati dell'attributo non corrisponde, o se uno o più dei valori forniti non sono validi o sono fuori intervallo, accumula una *InvalidAttributeException*.
3. Altrimenti accumula una *AttributeNotFoundException* se l'attributo indicato non esiste.

Se vengono accumulate delle eccezioni, questo metodo genera una *CompositeException* e tutte le operazioni di aggiornamento sono annullate. L'elenco di cause dell'eccezione include le eccezioni elencate sopra. Per ciascun attributo che ha causato l'errore, vengono registrati il riferimento e il nome attributo.

In tutti i casi, l'operazione di aggiornamento dell'attributo è soggetta ai normali vincoli di sicurezza e alla convalida. È responsabilità del client determinare quali attributi sono richiesti da una particolare istanza del componente, i tipi corretti, ecc.

## Parametri

*userCredential*: le credenziali dell'utente client.

*requestedLocale*: facoltativo, locale da utilizzare per questa richiesta.

*partitionName*: facoltativo, nome della partizione della campagna da utilizzare.

*attributesMap*: una mappa delle celle obiettivo da aggiornare; la chiave della voce è il riferimento della cella da aggiornare e il valore della voce è un array di attributi di aggiornamento. Il nome attributo è utilizzato per individuare l'attributo da aggiornare e i nuovi valori degli attributi sono utilizzati per aggiornare il valore dell'attributo esistente, come singolo oggetto del tipo appropriato o come array, se applicabile. Vedere le eccezioni riportate in precedenza.

## Restituzioni

Nessuno.

## Errori

ComponentException

AuthorizationException, DataException

## getRunResultsByCell

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

Ottiene i risultati dell'esecuzione di una o più celle obiettivo, eventualmente per un diagramma di flusso che non è mai iniziato o è ancora in corso.

## Parametri

*userCredential*: le credenziali dell'utente client.

*requestedLocale*: facoltativo, locale da utilizzare per questa richiesta.

*partitionName*: facoltativo, nome della partizione della campagna da utilizzare.

*cellReferences*: un array di riferimenti delle celle obiettivo per le quali si richiedono i risultati dell'esecuzione. Genera una *InvalidComponentException* se uno o più riferimenti della cella non sono validi oppure se fanno riferimento a una cella inesistente.

## Restituzioni

Restituisce un elenco con tipo di risultati dell'esecuzione per le celle indicate, ordinato in base all'array di riferimento di input.

Lo stato dell'esecuzione è `RUNNING` se il processo del diagramma di flusso sottostante è ancora in esecuzione, `FAILED` se l'esecuzione non è riuscita per qualche motivo o `NOT_STARTED` se l'esecuzione della casella del processo non è iniziata. Vengono forniti anche i dettagli sullo stato.



## Errori

InvalidComponentException

AuthorizationException, DataException

## bulkDeleteTargetCells

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

Elimina uno o più celle obiettivo esistenti e tutti i relativi componenti dipendenti (ovvero collegamento al diagramma di flusso, attributi).

### Parametri

*userCredential*: le credenziali dell'utente client.

*requestedLocale*: facoltativo, locale da utilizzare per questa richiesta.

*partitionName*: facoltativo, nome della partizione della campagna da utilizzare.

*cellReferences*: un array o uno o più riferimenti di celle da eliminare. Viene accumulata una *InvalidComponentException* se si verifica un problema con uno dei riferimenti specificati e se una cella non esiste.

Se vengono accumulate delle eccezioni, questo metodo genera una *CompositeException* e tutte le operazioni di eliminazione sono annullate. L'elenco di cause dell'eccezione include le eccezioni elencate sopra. Per ogni cella che ha causato l'errore, viene registrato il riferimento.

### Restituzioni

Nessuno.

## Errori

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
```

Aggiorna gli attributi dei modelli specificati

### Parametri

*userCredential*: le credenziali dell'utente client.

*partitionName*: facoltativo, nome della partizione della campagna da utilizzare.

*requestedLocale*: facoltativo, locale da utilizzare per questa richiesta.

*wsCampaignReference*: riferimento della campagna padre.

*allowCreate*: attualmente non utilizzato.

*clearExisting*: indicatore, se impostato su true tutti i precedenti valori nel modello che non vengono inviati nella richiesta sono cancellati.

*wsStaticAttributes*: elenco di attributi statici nel modello.

*wsHiddenAttributes*: elenco di attributi nascosti nel modello.

*wsParametricAttributes*: elenco di attributi parametrici nel modello.

## Restituzioni

Nessuno.

## Errori

Genera una CampaignServicesException se il modello di offerta non esiste oppure se il riferimento non è valido o non è stato fornito.

## listBottomUpTargetCells

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

Elenca le informazioni su tutte le celle obiettivo bottom-up attualmente esistenti per la campagna specificata.

## Parametri

*userCredential*: le credenziali dell'utente client.

*partitionName*: facoltativo, nome della partizione della campagna da utilizzare.

*requestedLocale*: facoltativo, locale da utilizzare per questa richiesta.

*wsCampaignReference*: riferimento della campagna padre.

## Restituzioni

Restituisce un elenco di zero o più istanze di WSTargetCellDetails per le celle corrispondenti.

## Errori

Genera una CampaignServicesException se la campagna non esiste oppure se il riferimento non è valido.

---

## Metodi API: Analitiche

L'API IBM Campaign supporta il richiamo di metriche semplici da IBM Campaign.

## getCampaignMetrics

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

Estrae le metriche per la campagna specificata.

### Parametri

*userCredential*: le credenziali dell'utente client.

*requestedLocale*: facoltativo, locale da utilizzare per questa richiesta.

*partitionName*: facoltativo, nome della partizione della campagna da utilizzare.

*campaignReference*: il riferimento della campagna padre. Genera una *InvalidComponentException* se si verifica un problema con il riferimento della campagna o se la campagna non esiste.

### Restituzioni

Restituisce un'istanza di *MetricsInfo* per la campagna.

### Errori

*InvalidComponentException*

*AuthorizationException*, *DataException*

---

## Metodi API: Offerte, elenchi offerte, modelli di offerte

L'API IBM Campaign supporta le seguenti operazioni relative alle offerte.

- rilevamento: creazione elenchi per cartella (offerte, elenchi offerte e sottocartelle), attributo (offerte e modelli di offerta) o valore di ricerca (offerte)
- convalida
- recupero delle informazioni (recupero degli attributi per una specifica offerta o modello di offerta)
- creazione, modifica, ritiro e eliminazione delle offerte

Le offerte dispongono di numerosi attributi standard ad esse associati. Questo elenco può essere esteso dal client aggiungendo le definizioni dei metadati degli attributi personalizzati (vedere le API di attributi).

Gli attributi standard dell'offerta sono:

- *uacName*: nome dell'offerta.
- *uacDescription*: facoltativo, stringa che descrive l'offerta.
- *uacOfferCode*: una stringa di codice che identifica in modo univoco l'offerta. Generalmente generato da IBM Campaign, ma può essere fornito dal client.
- *uacCreateDate*: un'istanza di calendario che indica la data e ora in cui l'offerta è stata creata dal server.
- *uacUpdateDate*: un'istanza di calendario che indica l'ultima volta in cui l'offerta è stata aggiornata dal server.

Anche i modelli di offerta dispongono di attributi standard e personalizzati. Gli attributi standard del modello di offerta sono:

- *uacName*: nome del modello di offerta.
- *uacDescription*: facoltativo, stringa che descrive il modello di offerta.
- *uacCreateDate*: un'istanza di calendario che indica la data e ora in cui il modello di offerta è stato creato dal server.
- *uacUpdateDate*: un'istanza di calendario che indica l'ultima volta in cui il modello di offerta è stato aggiornato dal server.

## listOffersAndFolders

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

Elenca tutte le offerte, gli elenchi offerte e le cartelle al di sotto di una cartella padre facoltativa.

Una volta richiamata, ciascuna istanza di *WSComponentOrFolderInfo* restituita può essere utilizzata direttamente, ad esempio per visualizzare il successivo livello della gerarchia della cartella, oppure è possibile utilizzare le API di attributi per estrarre o aggiornare le offerte contenute.

### Parametro

*userCredential*: le credenziali dell'utente client.

*requestedLocale*: facoltativo, locale da utilizzare per questa richiesta.

*partitionName*: facoltativo, nome della partizione della campagna da utilizzare.

*parentReference*: facoltativo, riferimento della cartella padre da elencare. Vengono elencate solo le offerte, elenchi offerte e cartelle figlio di primo livello della cartella padre, quindi sono necessarie successive chiamate a questa API per navigare nell'intera gerarchia di cartelle (comunque, generalmente la gerarchia è poco profonda). Se non viene fornita nessuna cartella padre, vengono restituiti tutti i componenti e cartelle sotto root.

Genera una *InvalidFolderException* se si verifica un problema con il riferimento alla cartella padre specificata.

Un *elenco* con tipo di zero o più istanze del wrapper dati *WSComponentOrFolderInfo*, una per ogni componente o cartella corrispondente.

### Errori

*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;
```

Elenca una "pagina" di offerte che contiene i criteri di ricerca forniti nei campi nome, descrizione, creato da o codice offerta, a partire dall'offset di pagina specificato. La ricerca di basa sul valore di cartella di input facoltativo. (Se si fornisce un folderID uguale 0, per impostazione predefinita viene utilizzata la cartella root dell'offerta). Le corrispondenze sono restituite in base a una corrispondenza "contains" per la stringa di ricerca.

Una volta richiamato, ciascun *WSOfferInfo* restituito può essere utilizzato direttamente, ad esempio per visualizzare un elenco di riepilogo, oppure è possibile utilizzare i metodi di attributo per estrarre o aggiornare gli attributi dell'offerta.

Questa API non gestisce alcuno stato, quindi è possibile effettuare chiamate in qualsiasi ordine.

### Parametri

*userCredential*: le credenziali dell'utente client.

*requestedLocale*: facoltativo, locale da utilizzare per questa richiesta.

*partitionName*: facoltativo, nome della partizione della campagna da utilizzare.

*folderID*: l'ID della cartella offerta in cui ricercare; se si specifica un folderID uguale a 0, la ricerca viene effettuata nella cartella root.

*searchCriteria*: la frase di ricerca.

*includeRetired*: il valore booleano che specifica se i risultati della ricerca includono le offerte ritirate. I valori validi sono TRUE e FALSE, TRUE indica che le offerte ritirate sono incluse e FALSE indica che non sono incluse.

*pageOffset*: l'offset iniziale di tutti i possibili componenti da cui iniziare l'enumerazione (con valore da zero). Ad esempio, se l'enumerazione comprende 1000 offerte e questo valore è impostato su 10, la pagina inizierebbe dall'undicesimo componente. Viene generata una *RangeException* se l'offset fornito è fuori intervallo.

*pageSize*: il numero massimo di componenti corrispondenti da restituire per la pagina (non può superare 500).

### Restituzioni

Restituisce un elenco con tipo di zero o più istanze del wrapper dati *Offer*, una per ogni offerta restituita nella pagina.

### Errori

*RangeException*

## listOffersByPage

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

Elenca una "pagina" di offerte che corrispondono ai valori degli attributi facoltativi, a partire dall'offset di pagina specificato. Le cartelle vengono ignorate. Le

corrispondenze vengono restituite in base a una corrispondenza "like" per le stringhe (la corrispondenza è considerata sufficiente se una stringa contiene il valore indicato nella query) e a una corrispondenza esatta per le date e i numeri.

Una volta richiamato, ciascun *OfferInfo* restituito può essere utilizzato direttamente, ad esempio per visualizzare un elenco di riepilogo, oppure è possibile utilizzare i metodi di attributo per estrarre o aggiornare gli attributi dell'offerta.

Questa API non gestisce alcuno stato, quindi è possibile effettuare chiamate in qualsiasi ordine.

## Parametri

*userCredential*: le credenziali dell'utente client.

*requestedLocale*: facoltativo, locale da utilizzare per questa richiesta.

*partitionName*: facoltativo, nome della partizione della campagna da utilizzare.

*attributes*: facoltativo, array di attributi da confrontare; il nome, il tipo di dati e i valori dell'attributo sono utilizzati per determinare la corrispondenza; se l'attributo supporta gli array, tutti i valori specificati devono corrispondere. L'operatore di corrispondenza implicito è un OR, quindi vengono restituiti i componenti che corrispondono a uno qualsiasi dei valori di attributo forniti.

Genera una *AttributeNotFoundException* se un nome attributo non esiste o una *InvalidAttributeException* se uno o più degli attributi forniti non sono validi.

*pageOffset*: l'offset iniziale di tutti i possibili componenti da cui iniziare l'enumerazione (con valore da zero). Ad esempio, se l'enumerazione comprende 1000 offerte e questo valore è impostato su 10, la pagina inizierebbe dall'undicesimo componente. Viene generata una *RangeException* se l'offset fornito è fuori intervallo.

*pageSize*: il numero massimo di componenti corrispondenti da restituire per la pagina (non può superare 500).

## Restituzioni

Un elenco con tipo di zero o più istanze del wrapper dati *OfferInfo*, una per ogni componente corrispondente nella pagina.

## Errori

*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 un nuovo elenco offerte smart.

## Parametri

*userCredential*: le credenziali dell'utente client.

*partitionName*: facoltativo, nome della partizione della campagna da utilizzare.

*requestedLocale*: facoltativo, locale da utilizzare per questa richiesta.

*name*: nome della cartella da creare.

*description*: descrizione per la nuova cartella.

*securityPolicyName*: nome della politica di sicurezza da utilizzare.

*parentFolder*: ID della cartella in cui deve essere creato l'elenco offerte.

*offerListInfo*: oggetto di tipo WSSmartListInfo.

*createdBy*: (facoltativo) oggetto di tipo WSApplicationTypeEnum che indica quale applicazione ha creato la cartella. Valori possibili: Campaign/Plan/Collaborate/eMessage. Se non fornito, viene utilizzato Campaign.

*creatorObjectId*: (facoltativo) utilizzato dal piano per collegare la cartella nella campagna con la cartella nel piano.

## Restituzioni

Restituisce un oggetto di tipo WSCreateSmartOfferListResponse.

## Errori

Genera una CampaignServicesException se l'id parentFolder non è valido o se offerListInfo non è stato fornito.

Genera una CampaignServicesException se il nome elenco è duplicato.

## 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 un nuovo elenco offerte statico.

## Parametri

*userCredential*: le credenziali dell'utente client.

*partitionName*: facoltativo, nome della partizione della campagna da utilizzare.

*requestedLocale*: facoltativo, locale da utilizzare per questa richiesta.

*name*: nome della cartella da creare.

*description*: descrizione per la nuova cartella.

*securityPolicyName*: nome della politica di sicurezza da utilizzare.

*parentFolder*: ID della cartella in cui deve essere creato l'elenco offerte.

*listMembers*: riferimenti alle offerte da includere nell'elenco offerte.

*createdBy*: (facoltativo) oggetto di tipo `WSApplicationTypeEnum` che indica quale applicazione ha creato la cartella. Valori possibili: `Campaign/Plan/Collaborate/eMessage`. Se non fornito, viene utilizzato `Campaign`.

*creatorObjectId*: (facoltativo) utilizzato dal piano per collegare la cartella nella campagna con la cartella nel piano.

## Restituzioni

Restituisce un oggetto di tipo `WSCreateStaticOfferListResponse`

## Errori

Genera una `CampaignServicesException` se l'id `parentFolder` non è valido o se `listMembers` non è valido.

Genera una `CampaignServicesException` se il nome elenco è duplicato.

## getOffers

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

Elenca i dettagli delle offerte come specificato nella richiesta.

## Parametri

*userCredential*: le credenziali dell'utente client.

*partitionName*: facoltativo, nome della partizione della campagna da utilizzare.

*requestedLocale*: facoltativo, locale da utilizzare per questa richiesta.

*wsCampaignReference*: riferimento della campagna padre.

## Restituzioni

Restituisce un oggetto di tipo `WSGetOffersResponse`.

## Errori

Genera una `CampaignServicesException` se l'offerta non esiste oppure se il riferimento non è valido o non è stato fornito.

## validateOffers

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



Convalida i codici offerta o i nomi di elenchi offerte forniti e restituisce le informazioni di convalida per ognuno. La "convalida" consiste nel verificare se esiste una sola offerta o elenco offerte corrispondente nel database.

L'oggetto OfferValidationInfo contiene un messaggio di errore invece delle informazioni sull'offerta se non è stata trovata nessuna offerta o elenco offerte corrispondente al codice o nome fornito. Viene restituito un errore anche se a un determinato codice o nome corrispondono più offerte o elenchi offerte. L'elenco viene restituito nello stesso ordine con cui viene fornito. I codici offerta e i nomi di elenchi offerte sono convalidati se esiste una corrispondenza esatta con le offerte.

### Parametri

*userCredential*: le credenziali dell'utente client.

*requestedLocale*: facoltativo, locale da utilizzare per questa richiesta.

*partitionName*: facoltativo, nome della partizione della campagna da utilizzare.

*codeOrNames*: un array di tutti i codici offerta o nomi di elenchi offerte da convalidare.

**Nota:** Questo metodo non genera eccezioni; le informazioni di convalida sono restituite per tutti i codici o nomi forniti.

### Restituzioni

Un elenco con tipo di zero o più istanze del wrapper dati *OfferValidationInfo*.

### Errori

Nessuno.

## 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
```

Aggiorna i dettagli degli elenchi offerte smart e statici.

### Parametri

*userCredential*: le credenziali dell'utente client.

*partitionName*: facoltativo, nome della partizione della campagna da utilizzare.

*requestedLocale*: facoltativo, locale da utilizzare per questa richiesta.

*listReference*: riferimento all'elenco offerte.

*isSmartList*: indicatore che indica se l'elenco è smart o statico.

*name*: nome della cartella da creare.

*description*: descrizione per la nuova cartella.

*listMembers*: riferimenti alle offerte da includere nell'elenco offerte.

*offerListInfo*: oggetto di tipo `WSSmartListInfo`.

*creatorObjectId*: (facoltativo) utilizzato dal piano per collegare la cartella nella campagna con la cartella nel piano.

*clearExisting*: indicatore che indica se le informazioni esistenti devono essere cancellate. Se `true`, i membri dell'elenco esistenti vengono cancellati prima di aggiungere i nuovi membri. Se `false`, i nuovi membri vengono aggiunti ai membri esistenti.

## Restituzioni

Restituisce un oggetto di tipo `WSEditOfferListResponse`.

## Errori

Genera una `CampaignServicesException` se l'id `parentFolder` non è valido, se `offerListInfo` non è stato fornito o se `listMembers` non è valido.

Genera una `CampaignServicesException` se il nome elenco è duplicato.

## 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, WSAAttribute[] wsAttributes)  
throws CampaignServicesException;
```

Crea una nuova offerta per il client, applicando gli attributi specificati.

## Parametri

*authorizationLoginName*: il nome utente dell'utente che crea l'offerta. Agli utenti devono essere concesse le autorizzazioni Aggiungi offerte per utilizzare questo metodo.

*partitionName*: facoltativo, nome della partizione della campagna da utilizzare.

*requestedLocale*: facoltativo, locale da utilizzare per questa richiesta.

*securityPolicyName*: facoltativo, nome della politica di sicurezza della campagna da utilizzare per creare l'offerta. Tutte le successive operazioni su questa offerta utilizzeranno questa politica. Se non definito, viene utilizzata la politica *Global*.

*name*: il nome da assegnare alla nuova istanza dell'offerta (il relativo attributo *uacName*).

*folderID*: l'ID della cartella in cui viene creata l'offerta. Questo ID viene convalidato per la correttezza e viene generata un'eccezione se l'ID non è valido.

*templateName*: obbligatorio, nome (univoco) di un modello di offerta esistente che deve essere utilizzato per la nuova offerta.

*wsAttributes*: un array di attributi di inizializzazione; tutti gli attributi forniti sovrascrivono i valori predefiniti dell'offerta; gli altri restano invariati. Ad esempio, se viene fornito un attributo *uacOfferCode*, questo viene utilizzato al posto di uno generato automaticamente. È il client a determinare gli attributi che sono richiesti dall'offerta, i relativi tipi, ecc.

Genera una *CampaignServicesException* se si verifica una delle seguenti condizioni:

- Il parametro *folderID* non è valido (non esiste o non è di tipo offerta).
- L'utente non è autorizzato a eseguire questa operazione.
- Vengono forniti attributi non validi in *wsAttributes*.
- Si verificano altre eccezione al runtime.

### Restituzioni

Una singola istanza di *OfferInfo* per l'offerta creata.

### Errori

*CampaignServicesException*

## retireOffers

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

Ritira una o più offerte esistenti.

### Parametri

*userCredential*: le credenziali dell'utente client.

*requestedLocale*: facoltativo, locale da utilizzare per questa richiesta.

*partitionName*: facoltativo, nome della partizione della campagna da utilizzare.

*references*: un array di riferimenti delle offerte da ritirare. Viene generata una *InvalidComponentException* se si verifica un problema con un particolare riferimento o se un'offerta non esiste.

### Restituzioni

Nessuno.

### Errori

*InvalidComponentException*

*AuthorizationException*, *DataException*

## deleteOffers

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

Elimina una o più offerte esistenti.

## Parametri

*userCredential*: le credenziali dell'utente client.

*requestedLocale*: facoltativo, locale da utilizzare per questa richiesta.

*partitionName*: facoltativo, nome della partizione della campagna da utilizzare.

*reference*: un array di riferimenti delle offerte da eliminare. Viene generata una *InvalidComponentException* se si verifica un problema con un riferimento specificato o se un'offerta non esiste.

## Restituzioni

Nessuno.

## Errori

*InvalidComponentException*

*AuthorizationException*, *DataException*

## deleteOffersAndLists

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

Elimina le offerte e gli elenchi specificati.

## Parametri

*userCredential*: le credenziali dell'utente client.

*partitionName*: facoltativo, nome della partizione della campagna da utilizzare.

*requestedLocale*: facoltativo, locale da utilizzare per questa richiesta.

*offers*: array di riferimenti dell'offerta o dell'elenco offerte.

## Restituzioni

Restituisce un oggetto di tipo *WSGetOfferListMembersResponse*.

## Errori

Genera una *CampaignServicesException* se l'ID offerta o l'ID elenco offerte non è valido.

## listOfferTemplates

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

Elenca tutti i modelli di offerta per i quali l'utente dispone dell'autorizzazione per la visualizzazione.

Una volta richiamata, ciascuna istanza di *WSOfferTemplateInfo* restituita può essere utilizzata direttamente, oppure è possibile utilizzare una o più API di attributi per estrarre o aggiornare un modello elencato.

### Parametro

*userCredential*: le credenziali dell'utente client.

*requestedLocale*: facoltativo, locale da utilizzare per questa richiesta.

*partitionName*: facoltativo, nome della partizione della campagna da utilizzare.

### Restituzioni

Un elenco con tipo di zero o più istanze del wrapper dati *WSOfferTemplateInfo*, una per ogni offerta restituita nella pagina.

### Errori

`InvalidExecutionContextException`, `AuthorizationException`

`DataException`

## createTemplate

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

Crea un nuovo modello di offerta.

### Parametri

*userCredential*: le credenziali dell'utente client.

*partitionName*: facoltativo, nome della partizione della campagna da utilizzare.

*requestedLocale*: facoltativo, locale da utilizzare per questa richiesta.

*name*: nome del modello di offerta appena creato.

*securityPolicyName*: nome della politica di sicurezza da utilizzare.

*wsStaticAttributes*: elenco di attributi statici nel modello.

*wsHiddenAttributes*: elenco di attributi nascosti nel modello.

*wsParametricAttributes*: elenco di attributi parametrici nel modello.

### Restituzioni

Restituisce un oggetto di tipo `WSCreateTemplateResponse`.

### Errori

Genera una `CampaignServicesException` se il modello di offerta non esiste oppure se il riferimento non è valido o non è stato fornito.

## getOfferTemplate

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

Elenca i dettagli dei modelli di offerta come specificato nei riferimenti.

### Parametri

*userCredential*: le credenziali dell'utente client.

*partitionName*: facoltativo, nome della partizione della campagna da utilizzare.

*requestedLocale*: facoltativo, locale da utilizzare per questa richiesta.

*wsCampaignReference*: riferimento della campagna padre.

### Restituzioni

Restituisce un oggetto di tipo `WSGetOfferTemplateResponse`.

### Errori

Genera una `CampaignServicesException` se il modello di offerta non esiste oppure se il riferimento non è valido o non è stato fornito.

## retireOfferTemplates

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

Ritira uno o più modelli di offerta specificati.

### Parametri

*userCredential*: le credenziali dell'utente client.

*partitionName*: facoltativo, nome della partizione della campagna da utilizzare.

*requestedLocale*: facoltativo, locale da utilizzare per questa richiesta.

*wsCampaignReference*: riferimento della campagna padre.

### Restituzioni

Restituisce un oggetto di tipo `WSRetireOfferTemplatesResponse`.

### Errori

Genera una `CampaignServicesException` se il modello di offerta non esiste oppure se il riferimento non è valido o non è stato fornito.

## getOffersAndListsByPage

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

Elenca le offerte o gli elenchi offerte per pagina.

### Parametri

*userCredential*: le credenziali dell'utente client.

*partitionName*: facoltativo, nome della partizione della campagna da utilizzare.

*requestedLocale*: facoltativo, locale da utilizzare per questa richiesta.

*type*: tipo che indica se sono richieste le offerte o gli elenchi offerte.

*pageSize*: il numero massimo di componenti corrispondenti da restituire per la pagina.

*pageOffset*: l'offset iniziale di tutti i possibili componenti da cui iniziare l'enumerazione (con valore da zero). Ad esempio, se l'enumerazione comprende 1000 offerte e questo valore è impostato su 10, la pagina inizierebbe dall'undicesimo componente. Viene generata una RangeException se l'offset fornito è fuori intervallo.

### Restituzioni

Restituisce un oggetto di tipo WSGetOffersAndListsByPageResponse.

### Errori

Nessuno.

## bulkCreateOffers

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

Crea offerte in massa con gli attributi per ciascuna offerta specificata nel parametro *offers* . Tutte le offerte sono create nel *folderID* specificato utilizzando il *templateName* specificato.

### Parametro

*authorizationLoginName*: le credenziali dell'utente client.

*partitionName*: facoltativo, nome della partizione della campagna da utilizzare.

*requestedLocale*: facoltativo, locale da utilizzare per questa richiesta.

*securityPolicyName*: facoltativo, nome della politica di sicurezza della campagna da utilizzare per creare l'offerta. Se non definito, viene utilizzata la politica globale.

*templateName*: nome del modello di offerta esistente nel sistema. Tutte le offerte sono create utilizzando questo modello.

*folderID*: l'ID della cartella offerta in cui sono create le offerte. Questo ID viene convalidato e viene generata un'eccezione se l'ID non è valido.

*offers*: un array di oggetti *WSBulkOfferInfo* che definiscono il nome e gli attributi dell'offerta. Per ulteriori dettagli, vedere il tipo di dati *WSBulkOfferInfo*.

### **Restituzioni**

Un array di istanze di *WSOfferInfoStatus* per ogni offerta. Contiene lo stato e le informazioni sull'offerta. Lo stato indica se la creazione dell'offerta è riuscita o non riuscita.

### **Errori**

*CampaignServicesException*

## **getOfferListDetails**

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

Elenca i dettagli dell'elenco offerte specificato.

### **Parametri**

*userCredential*: le credenziali dell'utente client.

*partitionName*: facoltativo, nome della partizione della campagna da utilizzare.

*requestedLocale*: facoltativo, locale da utilizzare per questa richiesta.

*listReference*: riferimento all'elenco offerte.

### **Restituzioni**

Restituisce un oggetto di tipo *WSGetOfferListDetailsResponse*.

### **Errori**

Genera una *CampaignServicesException* se i riferimenti agli elenchi non sono validi.

## **getOfferListMembers**

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

Elenca le informazioni sulle offerte nell'elenco offerte specificato.

### **Parametri**

*userCredential*: le credenziali dell'utente client.

*partitionName*: facoltativo, nome della partizione della campagna da utilizzare.



*requestedLocale*: facoltativo, locale da utilizzare per questa richiesta.

*listReference*: riferimento all'elenco offerte.

## Restituzioni

Restituisce un oggetto di tipo `WSDeleteOffersAndListsResponse`.

## Errori

Genera una `CampaignServicesException` se l'ID elenco offerte non è valido.

## getOffersByQuery

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

Elenca le offerte che corrispondono alle offerte fornite.

## Parametri

*userCredential*: le credenziali dell'utente client.

*partitionName*: facoltativo, nome della partizione della campagna da utilizzare.

*requestedLocale*: facoltativo, locale da utilizzare per questa richiesta.

*query*: query per individuare le offerte. Il formato della query è lo stesso utilizzato per l'elenco offerte smart.

*maxSize*: numero massimo di record da elencare.

*includeSubFolder*: indicatore che indica se una sottocartella deve essere inclusa nella ricerca.

*scopeFolders*: elenco di riferimenti della cartella in cui ricercare le offerte.

## Restituzioni

Restituisce un oggetto di tipo `WSGetOffersByQueryResponse`.

## Errori

Genera una `CampaignServicesException` se i riferimenti della cartella non sono validi.

## retireOfferLists

```
public void retireOfferLists(String user_credential, String partition_name,  
Locale locale, WSReference[] wsReferences)  
throws CampaignServicesException
```

Ritira uno o più elenchi offerte specificati.

## Parametri

*userCredential*: le credenziali dell'utente client.

*partitionName*: facoltativo, nome della partizione della campagna da utilizzare.

*requestedLocale*: facoltativo, locale da utilizzare per questa richiesta.

*wsReferences*: riferimento agli elenchi offerte.

## Restituzioni

Nessuno.

## Errori

Genera una CampaignServicesException se i riferimenti agli elenchi non sono validi.

## 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 nuova cartella di tipo campagna/offerta/sessione/segmento.

## Parametri

*userCredential*: le credenziali dell'utente client.

*partitionName*: facoltativo, nome della partizione della campagna da utilizzare.

*requestedLocale*: facoltativo, locale da utilizzare per questa richiesta.

*name*: nome della cartella da creare.

*description*: descrizione per la nuova cartella.

*securityPolicyName*: nome della politica di sicurezza da utilizzare.

*parentFolderId*: (facoltativo) ID della cartella padre. Se non fornito, la cartella viene creata in root.

*folderType*: tipo di cartella (offerta/sessione/campagna/segmento).

*createdBy*: (facoltativo) oggetto di tipo WSApplicationTypeEnum che indica quale applicazione ha creato la cartella. I valori possibili sono Campaign, Plan, Collaborate e eMessage. Se non fornito, viene utilizzato Campaign.

*creatorObjectId*: (facoltativo) utilizzato dal piano per collegare la cartella nella campagna con la cartella nel piano.

## Restituzioni

Restituisce un oggetto di tipo `WSCreateFolderResponse`.

## Errori

Genera una `CampaignServicesException` se il tipo della cartella non è valido o è duplicato.

## editFolder

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

Aggiorna la cartella specificata.

## Parametri

*userCredential*: le credenziali dell'utente client.

*partitionName*: facoltativo, nome della partizione della campagna da utilizzare.

*requestedLocale*: facoltativo, locale da utilizzare per questa richiesta.

*id*: ID della cartella da aggiornare.

*name*: nome della cartella da creare.

*description*: descrizione per la nuova cartella.

*folderType*: tipo di cartella (offerta/sessione/campagna/segmento).

*creatorObjectId*: (facoltativo) utilizzato dal piano per collegare la cartella nella campagna con la cartella nel piano.

*clearExisting*: attualmente non utilizzato.

## Restituzioni

Restituisce un oggetto di tipo `WSEditFolderResponse`.

## Errori

Genera una `CampaignServicesException` se il tipo della cartella non è valido o duplicato o se l'ID della cartella non è valido.

## getSubFoldersList

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

Elenca tutte le sottocartelle nella cartella specificata.

## Parametri

*userCredential*: le credenziali dell'utente client.

*partitionName*: facoltativo, nome della partizione della campagna da utilizzare.

*requestedLocale*: facoltativo, locale da utilizzare per questa richiesta.

*parentFolder*: riferimento della cartella di cui sono richieste le sottocartelle.

*folderType*: tipo della cartella.

## Restituzioni

Restituisce un oggetto di tipo `WSGetSubFolderListResponse`.

## Errori

Genera una `CampaignServicesException` se il riferimento della cartella non è valido.

## moveFolders

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

Sposta la cartella specificata in un'altra cartella padre.

## Parametri

*userCredential*: le credenziali dell'utente client.

*partitionName*: facoltativo, nome della partizione della campagna da utilizzare.

*requestedLocale*: facoltativo, locale da utilizzare per questa richiesta.

*folderId*: ID della cartella da eliminare.

*parentFolder*: ID della cartella padre.

*destinationFolder*: ID della cartella di destinazione in cui viene spostata la cartella specificata.

*folderType*: tipo di cartella (offerta/sessione/campagna/segmento).

## Restituzioni

Restituisce un oggetto di tipo `WSMoveFolderResponse`.

## Errori

Genera una `CampaignServicesException` se il tipo della cartella o l'ID della cartella padre non è valido oppure se l'ID della cartella non è valido.

## deleteFolders

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

Elimina le cartelle specificate nel sistema con tutti gli elementi nella cartella.

### Parametri

*userCredential*: le credenziali dell'utente client.

*partitionName*: facoltativo, nome della partizione della campagna da utilizzare.

*requestedLocale*: facoltativo, locale da utilizzare per questa richiesta.

*folderId*: ID della cartella da eliminare.

*parentFolder*: ID della cartella padre.

*deleteChilds*: indicatore che indica se tutte le dipendenze della cartella devono essere eliminate. Se false, non viene eliminata nessuna dipendenza.

*folderType*: tipo di cartella (offerta/sessione/campagna/segmento).

### Restituzioni

Restituisce un oggetto di tipo WSDeleteFolderResponse.

### Errori

Genera una CampaignServicesException se il tipo della cartella non è valido o se l'ID della cartella non è valido.



---

## Capitolo 5. Eccezioni comuni

L'API di servizi IBM Campaign può generare le seguenti eccezioni comuni. Tutti i messaggi di eccezione localizzati sono nella locale richiesta, se disponibili per IBM Campaign. Si applicano le normali politiche di fallback della locale Java.

### **RemoteException**

Questo elemento si applica solo all'interfaccia SOAP.

Tutte le chiamate SOAP all'API possono generare una RemoteException se si verifica un errore a livello di sistema, ad esempio un problema nel livello di elaborazione envelope SOAP (Axis), un vincolo definito nel WSDL del servizio web è stato violato per qualche motivo.

Le normali eccezioni API, controllate e non controllate, ad esempio DataException, vengono restituite come uno stato di errore, non come RemoteException.

Per i dettagli, consultare la sezione relativa all'interfaccia SOAP.

### **AuthenticationException**

Non è stato possibile autenticare l'utente per la partizione Campaign specificata. Verificare il ruolo dell'utente definito in IBM Marketing Platform.

### **AuthorizationException**

L'utente non è autorizzato a eseguire l'operazione richiesta. Questa eccezione può essere generata da qualsiasi metodo API, quindi è non dichiarata (non controllata). Verificare le autorizzazioni assegnate al ruolo dell'utente in IBM Marketing Platform.

### **DataException**

Si è verificata un'eccezione irreversibile nel livello di database sottostante in Campaign (non controllato).

Per dettagli, verificare i log del diagramma di flusso e del listener IBM Campaign.

### **LockException**

Un'eccezione temporanea che viene generata quando il client cerca di aggiornare un componente, ad esempio un diagramma di flusso, mentre viene modificato da un altro utente. Normalmente, questa eccezione può essere ripristinata restando in attesa e poi riprovando ad eseguire l'operazione. La logica di ripetizione è responsabilità del client.

### **InvalidComponentException**

Si è tentato di fare riferimento a un componente non valido o sconosciuto (campagna, diagramma di flusso, cella obiettivo). Il metodo `getReference()` dell'eccezione restituisce il riferimento del componente che ha causato l'errore.

## InvalidAttributeException

Un'eccezione che viene generata quando il client fornisce o fa riferimento a un attributo non valido, ad esempio se utilizza il tipo di dati errato, oppure utilizza un array di valori dove non è consentito. Il metodo `getAttributeName()` dell'eccezione restituisce il nome dell'attributo con il problema, `getAttributeValue()` restituisce il valore e `getComponentReference()` identifica il componente (o indice di massa).

## AttributeExistsException

Generata quando il client cerca di definire metadati di attributi duplicati per un componente. Il metodo `getAttributeName()` dell'eccezione restituisce il nome dell'attributo duplicato; `getComponentReference()` identifica il componente (o indice di massa).

## AttributeNotFoundException

Generata quando il client cerca di fare riferimento a un attributo sconosciuto (campagna, diagramma di flusso, cella obiettivo, ecc.). Il metodo `getAttributeName()` dell'eccezione restituisce il nome dell'attributo senza corrispondenza; `getComponentReference()` identifica il componente (o indice di massa).

## CompositeException

Una `CompositeException` viene utilizzata da alcune API per restituire più errori al chiamante. Generalmente esistono più cause collegate a questa eccezione; tutte le cause sono raccolte come elenco, nell'ordine con cui si sono verificate. Il metodo `getCauseList()` dell'eccezione restituisce questo elenco, che può essere ulteriormente esaminato per ottenere i dettagli di ciascun errore.

**Nota:** Normalmente l'API viene completata correttamente oppure esegue il rollback prima di generare un'eccezione composta. Consultare, ad esempio, l'API di massa del foglio di calcolo delle celle obiettivo descritta in "Metodi API: Celle obiettivo" a pagina 31.



---

## Prima di contattare il supporto tecnico di IBM

Se si riscontra un problema che non è possibile risolvere consultando la documentazione, il contatto di supporto designato dell'azienda può registrare una chiamata al supporto tecnico di IBM. Utilizzare queste linee guida per garantire una risoluzione efficiente e valida del problema.

Se il proprio ruolo non è quello di contatto di supporto designato, rivolgersi all'amministratore di IBM per informazioni.

**Nota:** Il supporto tecnico non scrive o crea script API. Per supporto nell'implementazione delle offerte API, contattare i Servizi professionali IBM.

### Informazioni da raccogliere

Prima di contattare il supporto tecnico di IBM, raccogliere le seguenti informazioni:

- Una breve descrizione della natura del problema.
- I messaggi di errore dettagliati generati quando si verifica il problema.
- La procedura dettagliata per riprodurre il problema.
- File di log, file di sessione, file di configurazione e file di dati correlati.
- Informazioni sull'ambiente di sistema e del prodotto , che è possibile ottenere come descritto in "Informazioni sul sistema".

### Informazioni sul sistema

Quando si contatta il supporto tecnico di IBM, è possibile che venga richiesto di fornire delle informazioni sull'ambiente.

Se il problema non impedisce di eseguire l'accesso, la maggior parte di queste informazioni è disponibile nella pagina Informazioni su, che fornisce informazioni sulle applicazioni IBM installate.

È possibile accedere alla pagina Informazioni su selezionando **Guida > Informazioni su**. Se la pagina Informazioni su non è accessibile, controllare il file `version.txt` che si trova nella directory di installazione dell'applicazione.

### Informazioni di contatto per il supporto tecnico di IBM

Per informazioni sui modi in cui è possibile contattare il supporto tecnico di IBM, consultare il sito Web del supporto tecnico del prodotto IBM: ([http://www.ibm.com/support/entry/portal/open\\_service\\_request](http://www.ibm.com/support/entry/portal/open_service_request)).

**Nota:** Per inoltrare una richiesta di supporto, è necessario eseguire l'accesso utilizzando un account IBM. Questo account deve essere collegato al proprio numero cliente IBM. Per ulteriori informazioni relative all'associazione del proprio account al numero cliente IBM, consultare **Risorse di supporto > Supporto software autorizzato** nel portale di supporto.



---

## Informazioni particolari

Queste informazioni sono state sviluppate per prodotti e servizi offerti negli Stati Uniti.

IBM può non offrire i prodotti, i servizi o le funzioni presentati in questo documento in altri paesi. Consultare il proprio rappresentante locale IBM per informazioni sui prodotti ed i servizi attualmente disponibili nella propria zona. Qualsiasi riferimento ad un prodotto, programma o servizio IBM non implica o intende dichiarare che solo quel prodotto, programma o servizio IBM può essere utilizzato. Qualsiasi prodotto funzionalmente equivalente al prodotto, programma o servizio che non violi alcun diritto di proprietà intellettuale IBM può essere utilizzato. È tuttavia responsabilità dell'utente valutare e verificare il funzionamento di eventuali prodotti, programmi o servizi non IBM.

IBM può avere applicazioni di brevetti o brevetti in corso relativi all'argomento descritto in questo documento. La fornitura di tale documentazione non garantisce alcuna licenza su tali brevetti. Chi desiderasse ricevere informazioni relative alle licenze può rivolgersi per iscritto a:

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

Per richieste di licenze relative ad informazioni double-byte (DBCS), contattare il Dipartimento di Proprietà Intellettuale IBM nel proprio paese o inviare richieste per iscritto a:

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

Il seguente paragrafo non è valido per il Regno Unito o per tutti i paesi le cui leggi nazionali siano in contrasto con le disposizioni in esso contenute:

L'INTERNATIONAL BUSINESS MACHINES CORPORATION FORNISCE QUESTA PUBBLICAZIONE "NELLO STATO IN CUI SI TROVA", SENZA ALCUNA GARANZIA, ESPLICITA O IMPLICITA, IVI INCLUSE EVENTUALI GARANZIE DI COMMERCIALIZZABILITÀ ED IDONEITÀ AD UNO SCOPO PARTICOLARE. Alcuni stati non consentono la rinuncia ad alcune garanzie espresse o implicite in determinate transazioni, pertanto, la presente dichiarazione può non essere applicabile.

Queste informazioni potrebbero includere inesattezze tecniche o errori tipografici. Le modifiche alle presenti informazioni vengono effettuate periodicamente; tali modifiche saranno incorporate nelle nuove pubblicazioni della pubblicazione. IBM può effettuare miglioramenti e/o modifiche ai prodotti e/o ai programmi descritti nella presente pubblicazione in qualsiasi momento senza preavviso.

Qualsiasi riferimento in queste informazioni a siti Web non IBM sono fornite solo per convenienza e non servono in alcun modo da approvazione di tali siti Web. I materiali presenti in tali siti Web non sono parte dei materiali per questo prodotto IBM e l'utilizzo di tali siti Web è a proprio rischio.

IBM può utilizzare o distribuire qualsiasi informazione fornita in qualsiasi modo ritenga appropriato senza incorrere in alcun obbligo verso l'utente.

I licenziatari di questo programma che desiderano avere informazioni allo scopo di abilitare: (i) lo scambio di informazioni tra i programmi creati indipendentemente e gli altri programmi (incluso il presente) e (ii) il reciproco utilizzo di informazioni che sono state scambiate, dovrebbero contattare:

IBM Corporation  
170 Tracer Lane  
Waltham, MA 02451  
U.S.A.

Tali informazioni possono essere disponibili, in base ad appropriate clausole e condizioni, includendo in alcuni casi, il pagamento di una tassa.

Il programma concesso in licenza descritto nel presente documento e tutto il materiale concesso in licenza disponibile sono forniti da IBM in base alle clausole dell'Accordo per Clienti IBM (IBM Customer Agreement), dell'IBM IPLA (IBM International Program License Agreement) o qualsiasi altro accordo equivalente tra le parti.

Qualsiasi dato sulle prestazioni qui contenuto è stato determinato in un ambiente controllato. Di conseguenza, i risultati ottenuti in altri ambienti operativi potrebbero variare in modo significativo. Alcune misurazioni possono essere state effettuate su sistemi del livello di sviluppo e non vi è alcuna garanzia che tali misurazioni resteranno invariate sui sistemi generalmente disponibili. Inoltre, alcune misurazioni possono essere state stimate tramite estrapolazione. I risultati effettivi possono quindi variare. Gli utenti del presente documento dovranno verificare i dati applicabili per i propri ambienti specifici.

Le informazioni relative a prodotti non IBM sono ottenute dai fornitori di quei prodotti, dagli annunci pubblicati o da altre fonti disponibili al pubblico. IBM non ha testato quei prodotti e non può confermarne l'accuratezza della prestazione, la compatibilità o qualsiasi altro reclamo relativo ai prodotti non IBM. Le domande sulle capacità dei prodotti non IBM dovranno essere indirizzate ai fornitori di tali prodotti.

Tutte le dichiarazioni relative all'orientamento o alle intenzioni future di IBM sono soggette a modifica o a ritiro senza preavviso e rappresentano unicamente mete ed obiettivi.

Tutti i prezzi IBM mostrati sono i prezzi al dettaglio suggeriti da IBM, sono attuali e soggetti a modifica senza preavviso. I prezzi al fornitore possono variare.

Queste informazioni contengono esempi di dati e report utilizzati in quotidiane operazioni aziendali. Per illustrarle nel modo più completo possibile, gli esempi includono i nomi di individui, società, marchi e prodotti. Tutti questi nomi sono fittizi e qualsiasi somiglianza con nomi ed indirizzi utilizzati da organizzazioni di impresa realmente esistenti è puramente casuale.

## LICENZA SUL DIRITTO D'AUTORE:

Queste informazioni contengono programmi applicativi di esempio in linguaggio sorgente, che illustrano tecniche di programmazione su varie piattaforme operative. È possibile copiare, modificare e distribuire questi programmi di esempio sotto qualsiasi forma senza alcun pagamento alla IBM, allo scopo di sviluppare, utilizzare, commercializzare o distribuire i programmi applicativi in conformità alle API (application programming interface) a seconda della piattaforma operativa per cui i programmi di esempio sono stati scritti. Questi esempi non sono stati testati in maniera approfondita e tenendo conto di tutte le possibili condizioni. La IBM, quindi, non può garantire o sottintendere l'affidabilità, l'utilità o il funzionamento di questi programmi. I programmi di esempio vengono forniti "NELLO STATO IN CUI SI TROVANO" e senza alcun tipo di garanzia. IBM declina ogni responsabilità per eventuali danni derivanti dall'uso degli stessi.

Se si visualizzano tali informazioni come softcopy, non potranno apparire le fotografie e le illustrazioni a colori.

---

## Marchi

IBM, il logo IBM e [ibm.com](http://ibm.com) sono marchi o marchi registrati di International Business Machines Corp., registrati in molte giurisdizioni nel mondo. Altri nomi di prodotti e servizi potrebbero essere marchi di IBM o di altre società. Un elenco attuale di marchi IBM è disponibile sul Web all'indirizzo su Informazioni relative a "Copyright and trademark information" all'indirizzo [www.ibm.com/legal/copytrade.shtml](http://www.ibm.com/legal/copytrade.shtml).

---

## Normativa sulla privacy e termini di utilizzo - Considerazioni

I prodotti software IBM, incluso il software come le soluzioni di servizi ("Offerta software") possono utilizzare cookie o altre tecnologie per raccogliere informazioni sull'utilizzo del prodotto, per migliorare l'esperienza dell'utente finale, per personalizzare le interazioni con l'utente o per altri scopi. Un cookie è un insieme di dati che un sito web può inviare al browser che viene memorizzato nel computer come un tag che lo identifica. In molti casi questi cookie non raccolgono informazioni personali. Se un'offerta software consente di raccogliere informazioni personali mediante i cookie e tecnologie simili, saranno fornite le seguenti specifiche.

In base alle configurazioni distribuite questa offerta software può utilizzare cookie di sessione e cookie persistenti che identificano il nome utente di ciascun utente e altre informazioni personali per la gestione della sessione, per fornire una migliorata usabilità all'utente o per tenere traccia dell'utilizzo o per scopi funzionali. Questi cookie possono essere disabilitati, ma disabilitando si eliminerà anche la funzione che essi abilitano.

Varie autorità regolano la raccolta di informazioni personali attraverso cookie e tecnologie simili. Se le configurazioni distribuite per questa offerta software forniscono la possibilità di raccogliere informazioni personali degli utenti finali mediante cookie e altre tecnologie, è necessario richiedere un parere legale relativo alle leggi in materia di tale raccolta di dati compreso qualsiasi requisito per ottenere il consenso quando necessario.

IBM richiede di (1) fornire un chiaro e visibile collegamento alle condizioni di utilizzo del sito web del cliente (ad esempio, normative sulla privacy) compreso un collegamento alla raccolta dei dati e alle pratiche d'uso dei clienti e di IBM, (2)

notificare che i cookie e i beacon gif/web trasparenti siano collocati sul computer del visitatore da IBM per conto del cliente unitamente ad una descrizione dello scopo di tali tecnologia e (3) per estensione richiesta dalla legge, ottenere il consenso dai visitatori del sito web prima del posizionamento di cookie e beacon gif/web trasparenti collocati dal cliente o da IBM per conto del cliente sui dispositivi dei visitatori del sito web.

Per informazioni sull'utilizzo di varie tecnologie compreso i cookie, a tale scopo consultare la normativa sulla privacy online IBM all'indirizzo:  
<http://www.ibm.com/privacy/details/us/en> e fare riferimento alla sezione "Cookies, Web Beacons and Other Technologies."





Stampato in Italia