

Version 10 Release 0  
Juni 2016

*IBM Campaign  
Handbuch zur SOAP-API*

**IBM**

**Hinweis**

Vor Verwendung dieser Informationen und des darin beschriebenen Produkts sollten die Informationen unter „Bemerkungen“ auf Seite 67 gelesen werden.

Diese Ausgabe bezieht sich auf Version 10, Release 0, Modifikation 0 von IBM Campaign und alle nachfolgenden Releases und Modifikationen, bis dieser Hinweis in einer Neuauflage geändert wird.

Diese Veröffentlichung ist eine Übersetzung des Handbuchs  
*IBM Campaign SOAP API Guide, Version 10 Release 0*,  
herausgegeben von International Business Machines Corporation, USA

© Copyright International Business Machines Corporation 1998, 2016

Informationen, die nur für bestimmte Länder Gültigkeit haben und für Deutschland, Österreich und die Schweiz nicht zutreffen, wurden in dieser Veröffentlichung im Originaltext übernommen.

Möglicherweise sind nicht alle in dieser Übersetzung aufgeführten Produkte in Deutschland angekündigt und verfügbar; vor Entscheidungen empfiehlt sich der Kontakt mit der zuständigen IBM Geschäftsstelle.

Änderung des Textes bleibt vorbehalten.

Herausgegeben von:  
TSC Germany  
Kst. 2877  
Juni 2016

© Copyright IBM Corporation 1998, 2017.

---

# Inhaltsverzeichnis

## Kapitel 1. Übersicht über die SOAP-API von IBM Campaign . . . . . 1

Übersicht über den SOAP-API-Entwurf . . . . .	1
SOAP-API-Änderungen nach Version . . . . .	3
Bestehende Implementierungen, die die JAR-Datei der Client-API verwenden . . . . .	4
Bestehende Implementierungen, die WSDL direkt verwenden . . . . .	5
SOAP-API-Referenzen . . . . .	7

## Kapitel 2. Verwenden der SOAP-API von IBM Campaign . . . . . 9

Verwenden der JAR-Datei der Client-API zum Aufrufen der Campaign-Services . . . . .	9
OfferAPI.java . . . . .	9
Verwenden der WSDL zum Aufrufen der Campaign-Services. . . . .	11
Leistungsaspekte . . . . .	11

## Kapitel 3. SOAP-API-Datentypen. . . . . 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 . . . . .	19
WSRunResults . . . . .	19
WSOfferInfo . . . . .	20
WSOfferCodeOrName. . . . .	20
WSOfferValidationInfo. . . . .	20
WSOfferTemplateInfo . . . . .	20
WSBulkOfferInfo . . . . .	20
WSOfferInfoStatus . . . . .	21

## Kapitel 4. SOAP-API-Methoden . . . . . 23

SOAP-API-Methoden: Service . . . . .	23
getServiceInfo . . . . .	23
SOAP-API-Methoden: Attribute. . . . .	23
getAttributesByName . . . . .	23
updateAttributes . . . . .	24
getAttributeMetadataByName . . . . .	25
createAttributeMetadata . . . . .	26
updateAttributeMetadata . . . . .	27
deleteAttributeMetadata . . . . .	29

SOAP-API-Methoden: Kampagnen und Ablaufdiagramme . . . . .	30
generateCampaignCode . . . . .	30
deleteCampaigns . . . . .	31
createCampaign . . . . .	32
listCampaignsByPage . . . . .	32
stopFlowchart . . . . .	33
SOAP-API-Methoden: Zielzellen . . . . .	34
createTargetCell . . . . .	35
bulkCreateTargetCells . . . . .	36
listTargetCells . . . . .	37
bulkUpdateTargetCells. . . . .	38
getRunResultsByCell . . . . .	39
bulkDeleteTargetCells . . . . .	40
updateTemplateAttributes . . . . .	40
listBottomUpTargetCells . . . . .	41
SOAP-API-Methoden: Analysen . . . . .	41
getCampaignMetrics . . . . .	42
SOAP-API-Methoden: Angebote, Angebotslisten, Angebotsvorlagen . . . . .	42
listOffersAndFolders . . . . .	43
searchOffersBasic . . . . .	44
listOffersByPage . . . . .	45
createSmartOfferList . . . . .	46
createStaticOfferList . . . . .	47
getOffers . . . . .	47
validateOffers . . . . .	48
editOfferList . . . . .	49
createOffer. . . . .	50
retireOffers . . . . .	51
deleteOffers . . . . .	51
deleteOffersAndLists . . . . .	52
listOfferTemplates . . . . .	52
createTemplate . . . . .	53
getOfferTemplate . . . . .	53
retireOfferTemplates . . . . .	54
getOffersAndListsByPage. . . . .	54
bulkCreateOffers . . . . .	55
getOfferListDetails . . . . .	56
getOfferListMembers . . . . .	56
getOffersByQuery . . . . .	57
retireOfferLists . . . . .	57
createFolder . . . . .	58
editFolder . . . . .	59
getSubFoldersList . . . . .	60
moveFolders . . . . .	60
deleteFolders . . . . .	61

**Kapitel 5. SOAP-API - Allgemeine Ausnahmen . . . . . 63**

**Bevor Sie sich an den technischen Support von IBM wenden. . . . . 65**

**Bemerkungen. . . . . 67**  
Marken. . . . . 69

Hinweise zu Datenschutzrichtlinien und Nutzungsbedingungen . . . . . 69

---

# Kapitel 1. Übersicht über die SOAP-API von IBM Campaign

In der Spezifikation der SOAP-API von IBM® Campaign wird Version 3.0 der Anwendungsprogrammierschnittstelle (Application Programming Interface, kurz API), auch als Campaign-Services bekannt, definiert. Diese Spezifikation wird als Bestandteil des Software Developer's Toolkit (devkits) für die Campaign-Services bereitgestellt, das mit IBM Campaign installiert wurde.

Das vom Installationsprogramm bereitgestellte Verzeichnis <CAMPAIGN\_HOME>/devkits/CampaignServicesAPI enthält Beispiele, Erstellungs- und Textscripts, Javadoc für öffentliche Klassen und Schnittstellen sowie Releaseinformationen.

Die SOAP-API der Campaign-Services wurde für folgende Aktivitäten entworfen:

- Bereitstellen eines differenzierten Erstellungs-, Erkennungs-, Lese- und Aktualisierungszugriffs auf IBM Campaign-Komponenten während der Isolierung der Clients von zugrunde liegenden Einzelheiten zur Implementierung.
- Arbeiten mit der bestehenden IBM Campaign-Benutzerschnittstelle mit minimalen Auswirkungen.
- Garantieren der Datenintegrität.
- Erfüllen der erforderlichen Sicherheitservices von IBM Campaign.
- Unterstützen des standardisierten SOAP (Simple Object Access Protocol), einschließlich sicherer Authentifizierung.

---

## Übersicht über den SOAP-API-Entwurf

Die Services-SOAP-API von Campaign ist eine Fassade, in der die Clientansicht einer aktiven IBM Campaign-Anwendungsinstanz bereitgestellt wird. Es wird zwar nur ein Subset der Funktionalitäten von IBM Campaign angezeigt, dies reicht jedoch aus, um Schlüsselaspekte der Campaign-Funktionalität zu steuern.

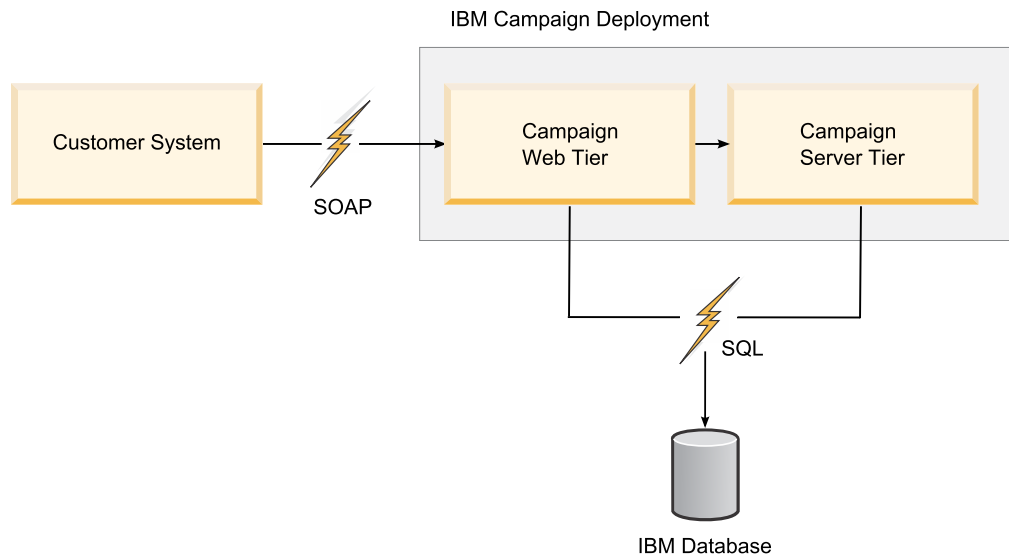
### Funktionen und Diagramm

Die API wird gleichzeitig mit IBM Campaign-Webbenutzern und anderen API-Threads verwendet.

Im Allgemeinen unterstützt die API in Kampagnen, Angeboten und Zielzellenkomponenten die folgenden Operationstypen:

- Komponentenerstellung
- Komponentenermittlung
- Komponentenlöschung
- Erstellung, Überprüfung und Änderung von Komponentenattributen und Metadaten für Attribute
- Abrufen von Ergebnissen der Ablaufdiagrammausführung

Im folgenden Diagramm wird eine Beispielimplementierung von CampaignServices 3.0 dargestellt.



## Benutzerauthentifizierung

Bei der Authentifizierung geht es um das Herstellen einer Benutzeridentität. Für die Benutzerauthentifizierung ist die Clientanwendung zuständig.

## Benutzerberechtigung

Bei der Berechtigung geht es um Berechtigungen eines Benutzers, über die er relativ zu von der API angezeigten Komponenten und Operationen verfügt.

Es ist möglich, dass ein Benutzer sich zwar erfolgreich authentifizieren kann, jedoch nicht über ausreichend Berechtigungen verfügt, um einige Operationen durchzuführen, wie z. B. das Bearbeiten der Übersichtsinformationen zu einer Kampagne. In diesem Fall löst die API-Methode `AuthorizationException` aus.

## Ländereinstellungen

API-Anforderungen werden für einen optionalen Parameter **requestedLocale** bereitgestellt. Dieser definiert die Ländereinstellung, die bei der Durchführung dieser bestimmten Anforderung verwendet werden soll. Ist keine Ländereinstellung definiert, wird in der API standardmäßig die bevorzugte IBM Ländereinstellung des Benutzers verwendet. Der gewöhnliche Java™-Abgleichalgorithmus für die beste Leistung wird verwendet, um Nachrichten und andere in der angeforderten Ländereinstellung lokalisierten Texte zurückzugeben.

Dieser Parameter gehört zur Klasse vom Typ `java.util.Locale`.

**Anmerkung:** Einige benutzerdefinierte Text, wie z. B. Kampagnenbeschreibungen, befinden sich in der Ländereinstellung des Benutzers, der den Text angegeben hat. IBM Campaign versucht nicht, solche Daten zu lokalisieren. Die API lokalisiert lediglich die Informationen, Warnungshinweise und Fehlermeldungen.

## Statusverwaltung

Die CampaignServices-API ist statusunabhängig. Das bedeutet, dass bei Aufrufen keine Informationen pro Client von der API gespeichert werden.

Offensichtlich wird der Status bestimmter zugrunde liegender Komponenteninstanzen, die von Campaign verwaltet werden, durch bestimmte API-Aufrufe geändert. Diese Statusänderungen könnten in der Datenbank als dauerhaft definiert werden.

---

## SOAP-API-Änderungen nach Version

Der Zweck dieses Abschnitts besteht darin, Änderungen an der SOAP-API von IBM Campaign für Kunden zu identifizieren, die die API derzeit verwenden. Wenn Sie ein Upgrade von einer früheren Version durchgeführt haben, prüfen Sie die folgenden Informationen, um zu ermitteln, ob Sie Änderungen an Ihrem Anwendungscode vornehmen müssen.

### Versionen und Abwärtskompatibilität

Zukünftige Versionen der CampaignServices-API sind mit älteren Versionen kompatibel, mit allen Neben- und Wartungsreleases, die die gleiche Hauptversionsnummer haben. IBM behält sich jedoch das Recht vor, die Kompatibilität mit der älteren Version für Hauptreleases vom Typ "dot zero" (x.0) zu unterbrechen.

Die Hauptversionsnummer der API wird erhöht, wenn eine der folgenden Änderungen vorgenommen wird:

- Änderung der Dateninterpretation
- Änderung der Geschäftslogik, d. h. der Funktionalität der Servicemethode
- Änderung der Methodenparameter und/oder Rückgabetypen

Die Nebenversionsnummer der API wird erhöht, wenn eine der folgenden Änderungen vorgenommen wird.

- Hinzufügen einer neuen Methode
- Hinzufügen eines neuen Datentyps und Einschränkung dieses Datentyps auf neue Methoden
- Hinzufügen eines neuen Typs zu einem aufgezählten Typ
- Definieren einer neuen Version einer Schnittstelle

IBM unterstützt weiterhin die veröffentlichte WSDL, den SOAP-Client und die Apache Axis-Version, die zur Implementierung des SOAP-Angebots bis zum nächsten IBM Hauptrelease mindestens verwendet wird. Praktisch betrachtet wird dies dadurch erreicht, dass mehrere versionsspezifische Web-Services gleichzeitig unterstützt werden. (IBM unterstützt bereits intern mehrere Versionen dieses Service.)

### Änderungen in v9.1.2

In v9.1.2 wurden keine Änderungen an der API vorgenommen.

**Anmerkung:** In v9.1.2 wurde eine REST-API eingeführt, zusätzlich zu der in diesem Handbuch beschriebenen SOAP-API. Die REST-API wird separat beschrieben.

### Änderungen in v9.1.1

**stopFlowchart** umfasst einen neuen Eingabeparameter, **runId**, zur Unterstützung einer Umgebung mit in Gruppen zusammengefassten Listnern.

### Änderungen in v9.1

In Version 9.1 wurden keine Änderungen vorgenommen.

## Änderungen in v9.0

In Version 9.0 wurden keine Änderungen vorgenommen.

## Änderungen in v8.6

In v8.6 implementierte API-Änderungen:

- Für die SOAP-Engine wurde ein Upgrade von AXIS Version 1.4.1 auf AXIS2 1.5.2 durchgeführt.
- Die WSDL wurde restrukturiert, damit Probleme mit den erforderlichen und optionalen Parametern behandelt werden können.
- Die JAR-Datei der Client-API wurde aufgrund der WSDL-Änderungen ebenfalls geändert. Folglich wurden die generierten Stubs und Klassen geändert. Die Parameter für die Client-API-Methode wurden nicht geändert, die Konstruktoren der unterstützenden Wertobjekte wurden jedoch für die Nutzung des AXIS2 WSDL2Java-Converters geändert.
- Die Web-Service-URL verweist auf:

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

und die entsprechende WSDL wird abgerufen unter:

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

Wenn Sie ein Upgrade auf IBM Campaign ab Version 8.6 durchführen und Sie derzeit die Campaign-Services-API verwenden, müssen Sie Ihren Anwendungscode ändern.

Abhängig davon, ob Sie die Client-API oder die WSDL verwenden, finden Sie in den folgenden Abschnitten weitere Einzelheiten:

- „Bestehende Implementierungen, die die JAR-Datei der Client-API verwenden“
- „Bestehende Implementierungen, die WSDL direkt verwenden“ auf Seite 5

## Bestehende Implementierungen, die die JAR-Datei der Client-API verwenden

Diese Informationen gelten, wenn Sie ein Upgrade auf IBM Campaign ab Version 8.6 durchführen und derzeit die JAR-Datei der Client-API für die Interaktion mit der Campaign-Webanwendung verwenden.

### JAR-Datei der Client-API

In Ihrer Java-Anwendung muss die JAR-Datei verwendet werden, die sich an der folgenden Position befindet:

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

Ein Java-Beispiel zur Erstellung eines neuen Angebots finden Sie im Abschnitt „OfferAPI.java“ auf Seite 9. Das gleiche Beispiel kann in Ihrer Campaign-Installation an der folgenden Position gefunden werden:

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



## Abhängige JAR-Dateien

Durch das Upgrade auf AXIS2 Version 1.5.2 muss auch für Ihre Java-Anwendung ein Upgrade durchgeführt werden, damit die JAR-Dateien der AXIS2-Verteilung Version 1.5.2 verwendet werden können, da CampaignServicesClient30.jar von diesen JAR-Dateien abhängig ist. Alle abhängigen JAR-Dateien müssen im Java-Klassenpfad Ihrer Anwendung enthalten sein und können in der Datei Campaign.war gefunden werden, die sich unter <CAMPAIGN\_HOME>/Campaign.war befindet.

Extrahieren Sie die JAR-Dateien aus Campaign.war und fügen Sie sie in den Java-Klassenpfad ein.

## Konstruktor der Client-API

Ändern Sie während der Erstellung des Client-API-Objekts die Web-Service-URL und die Ausnahmesignatur wie im folgenden Beispiel dargestellt.

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

## Parametrisierte Konstruktoren der unterstützenden Klassen

Bei der AXIS2-Engine verfügen die generierten Klassen und Stubs nicht über parametrisierte Konstruktoren. Stattdessen verfügen diese Klassen nur über den Standardkonstruktor ohne Argumente mit Settern und Gettern.

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

## Bestehende Implementierungen, die WSDL direkt verwenden

Diese Informationen gelten, wenn Sie ein Upgrade auf IBM Campaign ab Version 8.6 durchführen und derzeit die WSDL für die Interaktion mit der Campaign-Webanwendung verwenden. Mithilfe der WSDL des Campaign-Web-Services werden durch Verwendung eines Converter-Tools eines anderen Anbieters clientseitige Stubs generiert und Klassen unterstützt. In hier angegebenen Beispielen wird das WSDL2Java-Tool von Apache AXIS2 1.5.2 verwendet.

## WSDL-Position und Service-URL

Die Campaign-Web-Services von IBM Campaign werden an der folgenden Position bereitgestellt:

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

Die entsprechende WSDL kann an der folgenden Position abgerufen werden:

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

## Generieren von Stubs und Klassen

Mit dem WSDL2Java-Tool von Apache AXIS2 1.5.2 können die Stubs generiert und Java-Klassen von der WSDL unterstützt werden. Im Folgenden wird eine Ant-Task als Beispiel dargestellt.

Das Tool kann mit vergleichbaren Argumenten auch über die Befehlszeile verwendet werden. Die Argumentwerte können so geändert werden, dass sie an Ihre Umgebung angepasst werden.

**Anmerkung:** Im folgenden Beispiel des WSDL2Java-Converters wird Die ADB-Standardbindung verwendet.

```
<java classname="org.apache.axis2.wsd1.WSDL2Java" fork="true">
  <classpath refid="axis2.class.path"/> <!--Klassenpfad mit
AXIS2-Bibliotheken -->
  <arg value="-uri"/>
  <arg file="CampaignServices30.wsdl"/> <!--Tatsächliche Position der
WSDL -->
  <arg value="-s"/> <!-- Generieren des Codes für den Stil der Synchronisation -->
  <arg value="-Euwc"/> <!-- Übernimmt die Generierung der Wrappertypen
von Java für den nullfähigen Typ = Elemente vom Typ "true". -->
  <arg value="-uw"/> <!-- Aufheben des Wrappings von Parametern -->
  <arg value="-u"/> <!-- Entpacken von Klassen -->
  <arg value="-ns2p"/> <!-- Namensbereich bei Paketzuordnung. Kunden
können über eigene Paketnamen verfügen. -->
  <arg value="http://webservices.unica.com/campaign/CampaignServices/
3.0=com.unica.publicapi.campaign.campaignservices.soap.v30"/>
  <arg value="-o"/> <!-- Ausgabeverzeichnis -->
  <arg file="\${autogen.java.dir}"/>
</java>
```

## Verwenden von generierten Stubs und Unterstützen von Klassen

Der Stub kann wie folgt verwendet werden:

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

serviceStub._getServiceClient().getOptions().setTimeoutInMilliseconds
(webServiceTimeout);
//Zeitlimitüberschreitung in Millisekunden.
```

Das Angebot kann wie folgt erstellt werden:

```
try{
//Ändern Sie Host und Port, um sie an Ihre Umgebung anzupassen.
String serviceURL = "http://host:port/Campaign/services/CampaignServices30Service";
CampaignServices30ServiceStub serviceStub = new CampaignServices30ServiceStub(serviceURL);
long webServiceTimeout = 2*60*1000; //2 Minuten
serviceStub._getServiceClient().getOptions().setTimeoutInMilliseconds(webServiceTimeout);
//Zeitlimitüberschreitung in Millisekunden.

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 diesem Beispiel akzeptiert `createOffer()` jetzt nur einen Parameter vom Typ `CreateOffer`.

Bei der AXIS2-Engine verfügen die generierten Klassen und Stubs nicht mehr über parametrisierte Konstruktoren.

---

## SOAP-API-Referenzen

Die folgenden Referenzen wurden bei der Vorbereitung der SOAP-API-Spezifikation von IBM Campaign verwendet.

- "Basic Profile Version 1.1", Web Service Interoperability Organization (WS-I), 10. April 2006. (<http://www.ws-i.org/Profiles/BasicProfile-1.1-2006-0310.html>)
- "SOAP 1.2 (Entwurf)", W3C Soap Working Group, 24. Juni 2003 (<http://www.w3.org/TR/soap/>)
- "JAX-RPC 1.1", Sun Microsystems, 14. Oktober 2003 (<http://java.sun.com/webservices/jaxrpc/index.jsp>)
- Apache Web Services Working Group (<http://ws.apache.org/axis2>)



---

## Kapitel 2. Verwenden der SOAP-API von IBM Campaign

Um die SOAP-API der IBM Campaign-Web-Services zu verwenden, können Sie die JAR-Datei der Client-API oder direkt die WSDL verwenden. In einem Beispiel wird dargestellt, wie die Methode mit der JAR-Datei zum Erstellen eines Angebots verwendet wird.

---

### Verwenden der JAR-Datei der Client-API zum Aufrufen der Campaign-Services

IBM Campaign stellt eine Client-API bereit, die für die Interaktion mit der Campaign-Webanwendung SOAP-Web-Services verwendet. Dieser Wrapper ist in einer JAR-Datei gebündelt, mit der die Clientanwendung die Campaign-API aufrufen kann.

Die JAR-Datei kann an der folgenden Position gefunden werden:

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

Im folgenden Beispiel wird die Erstellung eines neuen Angebots auf Stammverzeichnisebene im Angebotsordner in Campaign dargestellt. Das gleiche Beispiel kann an der folgenden Position gefunden werden:

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

**Anmerkung:** Im Beispiel werden einige Dummywerte für die Parameter verwendet; Ihre tatsächlichen Werte unterscheiden sich unter Umständen davon.

Zudem lautet die URL für die Campaign-Web-Services `http://host:port/Campaign/services/CampaignServices30Service`, wobei sich "host" und "port" auf den Hostnamen und die Portnummer des Systems beziehen, auf dem die Campaign-Webanwendung implementiert wurde.

Wenn Sie ein bereitgestelltes Beispiel verwenden, stellen Sie sicher, dass Sie es so ändern, dass es Ihrer Clientumgebung entspricht.

#### OfferAPI.java

Um dieses Java-Beispiel zu kompilieren und auszuführen, müssen Sie alle abhängigen JAR-Dateien in den Java-Klassenpfad einbeziehen. Die Datei `CampaignServicesClient30.jar` hängt von den JAR-Dateien der Apache AXIS2 SOAP-Engine und weiteren allgemeinen JAR-Dateien ab, die unter `<CAMPAIGN_HOME>/Campaign.war` gefunden werden können. Extrahieren Sie die JAR-Dateien aus `Campaign.war` und fügen Sie sie in den Java-Klassenpfad ein.

```
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(IAAttributeMeta
            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();
    }
}
```

---

## Verwenden der WSDL zum Aufrufen der Campaign-Services

Campaign-Services können über die WSDL-Datei der Campaign-Web-Services, CampaignServices30.wsdl, aufgerufen werden.

Die Datei CampaignServices30.wsdl kann an der folgenden Position gefunden werden:

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

oder in der Campaign-Verteilung unter:

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

In der Java-Clientanwendung müssen die Klassen und Stubs verwendet werden, die von der WSDL durch die Verwendung eines Converter-Tools von WSDL in Java eines anderen Anbieters generiert werden. IBM empfiehlt die Verwendung von Apache AXIS.

Die Javadocs, die mit den Stubs und Klassen erstellt werden, die von der WSDL mit Apache AXIS2 generiert wurden, können an der folgenden Position gefunden werden:

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

**Anmerkung:** Alle abhängigen JAR-Dateien müssen im Java-Klassenpfad enthalten sein. Die Datei CampaignServicesClient30.jar ist abhängig von den JAR-Dateien auf der Apache AXIS2 SOAP-Engine und anderen JAR-Dateien von Apache, die in der Datei Campaign.war unter `<CAMPAIGN_HOME>/Campaign.war` gefunden werden können. Extrahieren Sie die JAR-Dateien aus Campaign.war und fügen Sie sie in den Java-Klassenpfad ein.

---

## Leistungsaspekte

Das aktuelle Leistungsprofil für die Implementierung der CampaignServices-API ähnelt dem Leistungsprofil der Anwendung, das über die grafische Benutzerschnittstelle angezeigt wird. Einige APIs werden explizit für die Leistung entworfen. Insbesondere die API `listCampaignsByPage()` ermöglicht eine relativ effiziente Paginierung.

In der SOAP-Schnittstelle werden von Natur aus Latenzzeit und Mehraufwand eingeführt, da alle Daten in das XML-Format konvertiert werden, das in einigen Fällen ziemlich ausführlich ist. Beispiel: Ein einfacher SOAP-Aufruf für ein Loopback kann in einem typischen Netz 100 ms dauern (Java 1.4.x war sogar langsamer). Die API wird für typische Geschäftsanwendungsfälle in Portals- und anderen Clientanwendungen wie `listOffersByPage()` optimiert. Die SOAP-Leistung sollte demnach adäquat sein.

Der Client muss jedoch aufpassen, dass er die Gemeinkosten bei der normalen CampaignServices-Wartung bei Anforderungen von Webbenutzern nicht zu hoch

ansetzt. Im Allgemeinen wird erwartet, dass die Verarbeitungsanforderungen eines API-Benutzers die Anforderungen eines typischen IBM Campaign-Webbenutzers nicht überschreiten.



---

## Kapitel 3. SOAP-API-Datentypen

Die SOAP-API der IBM Campaign-Services verwendet die folgenden öffentlichen Datentypen.

---

### WSReference

Ein einfacher Wrapper in einer Datenbank-ID:

- **componentTypeEnum**: Ein aufgezählter Typ, der den Komponententyp angibt, für den die ID bestimmt ist. Dazu zählen die Folgenden:
  - FOLDER
  - CAMPAIGN
  - FLOWCHART
  - TCS\_CELL
  - OFFER
  - OFFER\_LIST
  - OFFER\_TEMPLATE
- *id*: Ein *Long*, der eine eindeutige, numerische, datenbankspezifische ID für die Referenz definiert.

---

### WSVersion

Ein Wrappertyp, der die verschiedenen Komponenten einer Version erfasst. Dazu zählen die folgenden:

- *major*: eine Ganzzahl, mit der die Hauptversionsnummer definiert wird, wie z. B. "8" von der Vollversion 8.1.2.3.
- *minor*: eine Ganzzahl, mit der die Nebenversionsnummer definiert wird, wie z. B. "1" von der Vollversion 8.1.2.3.
- *maintenance*: optionale Ganzzahl, mit der, falls zutreffend, die Wartungsnummer der Version definiert wird, wie z. B. "2" von der Vollversion 8.1.2.3. Wird niemals mit einer API-Version bereitgestellt.
- *patch*: optionale Ganzzahl, mit der, falls zutreffend, die Patch-Release-Nummer definiert wird, wie z. B. "3" von der Vollversion 8.1.2.3. Wird niemals mit einer API-Version bereitgestellt.

---

### WSServiceInfo

Ein einfacher Wrappertyp in Informationen zum Service. Er enthält die folgenden Felder:

- *apiVersion*: eine Instanz von *WSVersion*, die die aktuelle Version der vom Service unterstützten API definiert. (*apiVersion* umfasst nur Informationen zur Haupt- und Nebenversion.)
- *campaignVersion*: eine Instanz von *WSVersion*, die die Vollversion der zugrunde liegenden IBM Campaign-Instanz definiert.
- *name*: interner Name des Service, z. B. "CampaignServices30Service".

---

## WSAttributeTypeEnum

Ein aufgezählter Typ, der alle möglichen Attributtypen definiert. Dazu zählen:

- STANDARD: Standard- oder Basisattribut, das von Campaign definiert wurde.
- CUSTOM: Ein Attribut, das von einer anderen IBM Anwendung, dem Kunden oder einem anderen Anbieter definiert wurde.
- INPUT\_PARAMETER: Ein Eingabeparameter, wie z. B. ein Attribut, mit dem ein IBM Campaign-Ablaufdiagramm ausgeführt wird.
- OUTPUT\_PARAMETER: Ein Ausgabeparameter, wie z. B. ein Attribut, dessen Wert als Ergebnis eines Ablaufdiagramms entsteht, das in IBM Campaign ausgeführt wird.

---

## WSAttributeStatusEnum

Eine Aufzählung aller möglichen Attributstatuscodes. Dazu zählen:

- ACTIVE: Das Attribut ist aktiv und kann unter Umständen nach Belieben verwendet werden.
- RETIRED: Das Attribut wurde vom Service entfernt und sollte nicht verwendet werden.

---

## WSAccessTypeEnum

Ein aufgezählter Typ, der alle möglichen Zugriffstypen für Attributwerte definiert. Dazu zählen:

- READ\_ONLY: Der Attributwert kann gelesen und angezeigt, jedoch nicht geändert werden.
- READ\_WRITE: Der Attributwert kann gelesen, angezeigt und geändert werden.

Der Attributzugriff ist additiv zu Sicherheitsberechtigungen. Beispiel: Wenn die Sicherheitsrichtlinie für den Client den Lesezugriff auf ein bestimmtes Attribut verweigert, kann diese Sicherheitseinstellung nicht durch den Attributzugriff überschrieben werden. Die API würde das Attribut im Grunde niemals an den Client zurückgeben.

---

## WSSelectTypeEnum

Definiert alle möglichen Auswahltypen für einen bestimmten Attributwert. Dazu zählen:

- NONE: keine Auswahl (*hasOptions* ist auf "false" festgelegt).
- SINGLE\_SELECT: Es kann nur jeweils eine Attributoption aus der Liste der möglichen Optionen ausgewählt werden (nur gültig, wenn das Attribut über *hasOptions* verfügt).
- MULTIPLE\_SELECT: ähnelt SINGLE\_SELECT. Ausnahme ist, dass eine oder mehr Optionen gleichzeitig ausgewählt werden können.

---

## WSRunStatusEnum

Ein aufgezählter Typ aller möglichen Ausführungsstatus von Ablaufdiagrammen, Zweigen oder Zellen:

- NOT\_STARTED: die Ausführung wurde geplant, jedoch noch nicht gestartet.
- RUNNING: die Ausführung ist in Bearbeitung.

- CANCELLED: die Ausführung wurde abgebrochen, entweder von einem Campaign-Benutzer oder dieser API.
- SUCCEEDED: die Ausführung wurde erfolgreich abgeschlossen.
- FAILED: die Ausführung ist fehlgeschlagen; Fehlerdetails werden separat gemeldet. (Siehe „WSRunResults“ auf Seite 19.)

---

## WSRunTypeEnum

Ein aufgezählter Typ aller möglichen Ausführungstypen. Dazu zählen:

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

---

## WSAttribute

Attribute bieten einen einfachen, erweiterbaren Mechanismus zum Hinzufügen beliebiger Daten zu Komponenteninstanzen, auf die über die API zugegriffen werden kann. Bei den Daten handelt es sich entweder um Standarddaten, wie z. B. den *Namen* einer Kampagne oder Eingabeparameter für die Ablaufdiagrammausführung wie *Geschlecht*, oder um beliebige benutzerdefinierte Daten, die von einer anderen IBM Anwendung oder einem IBM Kunden angegeben werden.

**Anmerkung:** In dieser API werden mit den Attributen die meisten Komponentendaten abgebildet, nicht nur die benutzerdefinierten Campaign-Attribute.

Komponenten sind in der Regel viele Attribute zugeordnet, die von der Campaign-Services-API als eine besonders typisierte Zuordnung mit dem Namen *Attribute-Map* dargestellt werden. Attributdaten werden bei Attributen mit Dezimalzeichen (Numerik mit doppelter Genauigkeit) in der gesamten API als typisierte konkrete Klasse dargestellt, wie z. B. *WSDecimalAttribute*.

Jedes Attribut umfasst folgende Komponenten:

- *Name*: der eindeutige Name des Attributs. Dieser Name dient als Schlüssel für den Zugriff auf das Attribut und die zugehörigen Attributdaten innerhalb der auftretenden Komponenteninstanz. Das Format des Namens ist nicht definiert; in einigen Fällen wird es vom Service, vom Client oder von einem IBM Campaign-Benutzer zugeordnet.

Im Allgemeinen handelt es sich bei diesem Namen nicht um den Anzeigenamen, der einem Campaign-Benutzer oder einem Clientbenutzer angezeigt werden würde. Er kann von der API standardisiert werden, wie z. B. *uacDescription*, er kann von IBM Campaign bei der Veröffentlichung von Ablaufdiagrammen zugeordnet werden oder er wird unter Umständen von der IBM Anwendung oder dem -Kunden bei der Definition benutzerdefinierter Attribute zugeordnet. In allen Fällen ist der Name allerdings garantiert eindeutig.

- *Metadata*: (optional) Informationen zu den Attributdaten, wie z. B. Wertdatentyp, Anzeigename, Beschreibung, Eingabeaufforderungen, Standardwert, Auswahltyp, Länge (Text), Präzision (Dezimalstellen), Optionen (ob Einfach- oder Mehrfachauswahl). Siehe „WSAttributeMetadata“.
- *Values*: ein Array mit null oder mehr typisierten Wertobjekten. Das Wertfeld wird von der konkreten Attributklasse bereitgestellt; der Typ der einzelnen Werte muss mit der Typdefinition im Feld mit den Metadaten des Attributs identisch sein. Allerdings unterstützen nicht alle Attribute mehrere Werte.

Die folgenden konkreten Attributtypen werden unterstützt:

- **WSBooleanAttribute**: ein Attribut dessen Wert ein booleschen Wert ist, d. h., es ist *true* oder *false*.
- **WSIntegerAttribute**: ein ganzzahliger Wert (*java.lang.Long*).
- **WSDecimalAttribute**: ein Wert mit einer Dezimalzahl mit doppelter Genauigkeit (*java.lang.Double*).
- **WSCurrencyAttribute**: ein zusammengesetzter Währungswert, der den optionalen Währungscode ISO 4217 des Währungswerts umfasst, z. B. "USD" für den US-Dollar, und die als *Double* erfassten Währungswerte. Wenn der Währungscode nicht angegeben ist, wird der von IBM Campaign verwendete Standardwert angenommen.  
Eine Liste der Länder, Währungssymbole und Codes finden Sie unter <http://www.xe.com/symbols.php>. Die Ländereinstellung, die bei einem Währungswert verwendet wird, kann sich von der bevorzugten Ländereinstellung eines Benutzers unterscheiden.
- **WSCalendarAttribute**: Werte sind, in einigen Zeitzonen und Ländereinstellungen, Kalenderdaten oder Datumsangaben.
- **WSTextAttribute**: eine Zeichenfolge mit Unicodetext (wahrscheinlich null oder leer).

**Anmerkung:** die Liste der möglichen Attribute ist im Allgemeinen bei jedem Komponententyp unterschiedlich, in den Listen können jedoch Überschneidungen auftreten.

---

## WSAttributeMetadata

WSAttributeMetadata definiert Informationen zu den Daten eines bestimmten typisierten Attributs, wie z. B. zum Wertdatentyp, zu lokalisiertem Text (Anzeigename, Beschreibung, Eingabeaufforderungen), zum zugehörigen Standardwert, zum Bereich der gültigen Werte, zum Auswahltyp und zu Optionen (ob Einfach- oder Mehrfachauswahl). Metadaten für Attribute sind wie Attribute typisiert. Beispiel: An das *WSDecimalAttribute myNumber* muss ein WS vom Typ *DecimalAttributeMetadata* gebunden sein, und alle Werte, einschließlich der Attributwerte, Standardwerte für Metadaten und möglicher Optionswerte, werden als *Double* eingegeben.

Beschreibungen, Beschriftungen und weitere Metadatentexte für Attribute werden lokalisiert; benutzerdefinierter Text ist jedoch nur verfügbar, wenn er vom Benutzer eingegeben wurde. Jeder API-Aufruf umfasst eine angeforderte Ländereinstellung, die vom Clientcode für die Definition der Ländereinstellung verwenden kann, in der ein bestimmter Benutzer seine lokalisierten Nachrichten gerne anzeigen würde. Die gewöhnlichen Fallback-Richtlinien der Java-Ländereinstellung werden verwendet, um die Anforderung zu erfüllen.

WSAttributeMetadata umfasst die folgenden Felder:

- *name*: der Attributname, Standard oder benutzerdefiniert; dies ist auch der Name, der von dem an diese Metadaten gebundenen Attribut verwendet wird. Standardattribute werden vom System definiert und verfügen über Standardnamen in einem reservierten Namensbereich (d. h., sie verwenden das Präfix "uac"); benutzerdefinierte Namen verwenden eine beliebige andere Benennungskonvention.

**Anmerkung:** Der Attributname muss eindeutig sein, wird niemals lokalisiert und weist Längenbeschränkungen auf (die vom Zeicheninhalt und der Datenbank abhängen). Bei dem Namen muss die Groß-/Kleinschreibung nicht beachtet werden. Er kann aus einer Kombination aus Unicode-Buchstaben oder Ziffern und dem Unterstrich '\_' zusammengesetzt werden, er darf jedoch nicht mit einer Ziffer beginnen.

- *description*: optionale Beschreibung des Attributs. Geeignet für eine QuickInfo oder eine andere Darstellung in der Benutzerschnittstelle.
- Prädikate: diverse Prädikate, mit denen das Attribut beschrieben wird:
  - *isRequired*: trifft zu, wenn das Attribut obligatorisch ist.
  - *isInternal*: trifft zu, wenn das Attribut vom System definiert wurde und nur für interne Zwecke verwendet wird (sollte einem Benutzer nicht dargestellt werden).
  - *isGenerated*: trifft zu, wenn der Wert oder die Werte des Attributs bei der Erstellung der Komponente, z. B. ein Zielzellencode, automatisch von IBM Campaign generiert werden. Üblicherweise ist *accessTypeEnum* bei generierten Werten READ\_ONLY (schreibgeschützt).
  - *hasOptions*: trifft zu, wenn das Attribut Optionen hat. Impliziert, dass für diese Metadaten Optionen definiert wurden und *selectTypeEnum* entweder SINGLE\_SELECT (Einfachauswahl) oder MULTIPLE\_SELECT (Mehrfachauswahl) ist.
- *typeEnum*: eine *WSAttributeTypeEnum*, die den Attributtyp definiert, z. B. STANDARD oder CUSTOM (benutzerdefiniert).
- *statusEnum*: eine *WSAttributeStatusEnum*, die den Attributstatus definiert, z. B. ACTIVE (aktiv).
- *accessTypeEnum*: eine *WSAccessTypeEnum*, die die Art des Zugriffs auf den Attributwert definiert, z. B. READ\_ONLY (schreibgeschützt).
- *selectTypeEnum*: eine *WSAccessTypeEnum*, die die Art der für das Attribut verwendeten Auswahl definiert, z. B. SINGLE (einfach). Muss bei Kampagnen- und Zellenattributen, oder wenn keine Optionen angegeben sind, NONE sein.
- *componentTypeEnum*: eine *WSComponentTypeEnum* aller möglichen Campaign-Komponenten, die von der API angezeigt werden, wie z. B. CAMPAIGN (Kampagne), FOLDER (Ordner).
- *defaultValue* (nur bei Ablaufdiagrammen): optionaler Standardwerttyp für das Attribut. Dieser Wert wird von der Metadatenklasse für konkrete Attribute bereitgestellt, der Standardwert von *WSTextAttributeMetadata's* ist z. B. vom Typ "Zeichenfolge". (Weitere Informationen finden Sie in der Beschreibung der Attributwerte). Bei Komponenten, bei denen es sich nicht um Ablaufdiagramme handelt, ist der Standardwert undefiniert.
- *options*: optionale Liste mit Optionen für dieses Attribut. Insgesamt definieren die Optionen eines Attributs die genauen gültigen Werte für dieses Attribut; jede Option ist typisiert, sodass *WSTextAttributeMetadata* beispielsweise nur an *WSTextAttributeOption* gebunden werden kann.

**Anmerkung:** Bei Optionen gibt es eine Einschränkung; es werden nur Textattribute unterstützt.

die einzelnen Optionen definieren Folgendes:

- *prompt*: Eingabeaufforderung für die Option, die für Pulldown-Menüs geeignet ist, z. B. "Männlich" als Attributoption für das Geschlecht. Anders als bei Eingabeaufforderungen für Metadaten enthalten die Anzeigenamen von Optionen normalerweise keine Interpunktion.
- *description*: Lokalisierte Beschreibung der Option, z. B. "Eine Person mit männlicher Überzeugungskraft". Geeignet für QuickInfo-Text.
- *isDefault*: "true", wenn diese bestimmte Option der Standard ist. Bei Auswahltypen vom Typ MULTIPLE\_SELECT können mehrere Optionen als Standard markiert werden.
- *value*: Der typisierte Optionswert. Wie bei dem Standardwert der Metadaten für das Attribut wird auch dieser Wert von der konkreten Unterklasse für Optionen bereitgestellt. So ist der Wert *WSDecimalAttributeOption*'s z. B. ein Dezimaltyp. (Weitere Informationen finden Sie in der Beschreibung der Attributwerte). Dem oben genannten Beispiel mit dem *Geschlecht* folgend könnte der Wert entweder als eine Zeichenfolge vom Typ "m" (*WSTextAttributeOption*) oder als numerischer Code, 123 (*WSDecimalAttributeOption*), deklariert werden.

---

## WSCampaignInfo

Ein einfacher Wrappertyp in Kampagnenattributdaten.

Er enthält die folgenden Felder:

- *reference*: die Kampagnenreferenz.
- *name*: Kampagnenname (*uacName*); nicht garantiert eindeutig.
- *description*: optionale Kampagnenbeschreibung (*uacDescription*).
- *campaignCode*: der eindeutige Kampagnencode (*uacCampaignCode*); er wird entweder vom Client oder von Campaign zugeordnet.

---

## WSComponentOrFolderInfo

Enthält eine Kombination aus eingebundenen Daten einer Kampagne oder eines Ordnerattributs, wie z. B. einen Anzeigenamen und die zugehörige Referenz.

Er enthält die folgenden Felder:

- *reference*: die Komponenten- oder Ordnerreferenz.
- *name*: Komponenten- oder Ordnername (*uacName*); nicht garantiert eindeutig.
- *description*: optionale Komponenten- oder Ordnerbeschreibung (*uacDescription*).
- *componentCode*: eindeutiger Code für die Komponente bzw. null bei einem Ordner.

---

## WSTargetCellInfo

Ein einfacher Wrapper in Attributdaten für Zeilen mit Zielzellen.

Er enthält die folgenden Felder:

- *reference*: die Zellenreferenz.
- *name*: Zellenname (*uacName*); nicht garantiert eindeutig.
- *description*: optionale Zellenbeschreibung (*uacDescription*).

- *cellCode*: der Zellencode (*uacCellCode*); er wird entweder vom Client oder von Campaign zugeordnet. Es kann erzwungen werden, dass Zellencodes eindeutig sind, indem der IBM Campaign-Konfigurationsparameter *DuplicateCellCodesAllowed* auf "false" festgelegt wird.
- *flowchartName*: optionaler Name des Ablaufdiagramms, an das die Zelle gebunden wird.

---

## WSMetricsInfo

Ein einfacher Wrappertyp in Kampagnenanalysedaten, einschließlich der Anzahl der Kontakte. Er enthält die folgenden Felder:

- *totalContacts*: Ein Long, der die Gesamtanzahl der Kontakte angibt.
- *responses*: Eine typisierte Liste mit Instanzen von *WSMetricsResponse*, wobei jede Instanz Kontaktinformationen für eine Antwort definiert:
  - *typeCode*: Eine Zeichenfolge, die den Antworttypcode definiert, wie z. B. *PHC* für einen telefonischen Kontakt (Phone Call Contact).
  - *count*: Ein Long, der die Häufigkeit angibt, in der dieser Kontakt zustande gekommen ist.

---

## WSRunResults

Ein Wrappertyp in den Ergebnissen eines Ablaufdiagramms, eines Prozessfeldes oder einer Zellenausführung, der möglicherweise noch in Bearbeitung ist und den Ausführungsstatus, das Start- und das Enddatum sowie die Start- und die Endzeit der Ablaufdiagrammausführung und Angaben zur Anzahl umfasst.

Er umfasst die folgenden Felder:

- *sourceReference*: optionale Referenz der Quelle des Ausführungsergebnisses. Gemäß dem Kontext, nach dem die Ausführungsergebnisse abgerufen werden, kann sich diese Referenz auf ein Ablaufdiagramm, ein Prozessfeld des Ablaufdiagramms oder eine Zielzelle beziehen. Die verbleibenden Ergebnisdaten der Ausführung beziehen sich auf jeden Fall auf diese Quelle.
- *flowchartName*: der Name des Ablaufdiagramms, das ausgeführt wurde.
- *flowchartId*: die Datenbank-ID des Ablaufdiagramms.
- *runId*: die Datenbank-ID der Ausführung.
- *typeEnum*: ein aufgezählter Typ, der definiert, bei welcher Ausführung die Ergebnisse generiert wurden, z. B. *PRODUCTION\_PROCESS* (siehe *WSRunTypeEnum*).
- *statusEnum*: ein aufgezählter Typ, der den Ausführungsstatus definiert, z. B. *RUNNING* (siehe *WSRunStatusEnum*).
- *statusCode*: optionaler ganzzahliger Statuscode.
- *statusMessage*: optionale Statusnachricht.
- *startDate*: optionale Angabe von Datum und Zeit im Kalender, zu dem bzw. zu der die Ausführung gestartet wurde; die Angabe beträgt 0, wenn die Ausführung nicht gestartet wurde.
- *endDate*: wie *startDate*, allerdings handelt es sich um die Angabe von Datum und Zeit, zu dem bzw. zu der die Ausführung beendet wurde (Erfolg oder Fehler); die Angabe beträgt 0, wenn die Ausführung nicht gestartet wurde oder noch nicht beendet worden ist.
- *count*: optionale Gesamtanzahl der Kontakte, die bei der Ausführung ausgewählt werden; die Angabe beträgt möglicherweise null, wenn die Ausführung nicht abgeschlossen wurde.

---

## WSOfferInfo

Ein einfacher Wrappertyp in Attributdaten von Angeboten oder Angebotslisten.

Er enthält die folgenden Felder:

- *reference*: die Angebots- oder Angebotslistenreferenz.
- *name*: Angebot oder Name der Angebotsliste (*uacName*); nicht garantiert eindeutig.
- *description*: optionale Beschreibung (*uacDescription*).
- *offerCode*: der Angebotscode (*uacOfferCode*) bei einem Angebot oder null bei einer Angebotsliste. (Nicht garantiert eindeutig.)

---

## WSOfferCodeOrName

Ein einfacher Wrappertyp in Angebotscodes oder Daten aus Namen von Angebotslisten.

Er enthält die folgenden Felder:

- *isCode*: boolescher Wert, der angibt, ob das Feld *codeOrName* ein vorausgesetzter Angebotscode ist (true) oder der Name einer Angebotsliste (false).
- *codeOrName*: der eindeutige Angebotscode (*uacOfferCode*) eines Angebots oder der Name der Angebotsliste.

---

## WSOfferValidationInfo

Ein einfacher Wrappertyp in Informationen zur Angebotsvalidierung.

Er enthält die folgenden Felder:

- *errorCode*: beträgt der Code nicht null, definiert er den alphanumerischen Fehlercode der Validierung. Weitere Informationen zu Fehlercodes finden Sie in der *IStandardDefinitions*-Klasse.
- *errorMessage*: optionale lokalisierte Nachricht, die den Fehler beschreibt (sofern einer aufgetreten ist).
- *codeOrName*: der validierte Fehlercode oder der Name der Angebotsliste.
- *reference*: die Angebots- oder Angebotslistenreferenz, sofern gültig.

---

## WSOfferTemplateInfo

Ein einfacher Wrappertyp in Daten von Angebotsvorlagen.

Er enthält die folgenden Felder:

- *reference*: die Referenz der Angebotsvorlage.
- *name*: Name der Angebotsvorlage; garantiert eindeutig.
- *description*: optionale Beschreibung (*uacDescription*).
- *offerTemplateID*: die eindeutige Datenbank-ID der Angebotsvorlage.

---

## WSBulkOfferInfo

Wird zur Erstellung von Massenangeboten verwendet.

Er enthält die folgenden Felder:

- *offerName*: der Name des Angebots, das erstellt wird.



- *attributes*: ein Array mit *WSAttribute*-Typen, in dem die Angebotsattribute angegeben werden.

---

## WSOfferInfoStatus

Ein Rückgabetypp für die API-Methode *bulkCreateOffers()*, die den Status einer Massenangebotserstellung angibt.

Er enthält die folgenden Felder:

- *name*: der Name des Angebots.
- *code*: der Angebotscode. Schlägt die Angebotserstellung fehl, beträgt er null.
- *description*: Angebotsbeschreibung.
- *reference*: *WSReference* des erstellten Angebots. Schlägt die Angebotserstellung fehl, beträgt sie null.
- *status*: eine Instanz von *WSRequestStatus*, die den Status der Angebotserstellung angibt.



---

## Kapitel 4. SOAP-API-Methoden

Die SOAP-API der IBM Campaign-Services verwendet die folgenden Methoden.

---

### SOAP-API-Methoden: Service

Die SOAP-API von IBM Campaign bietet eine Möglichkeit, Identifikationsinformationen zum Service selbst abzurufen.

#### getServiceInfo

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

Gibt Informationen zum Service zurück, z. B. die aktuelle, unterstützte API-Version und die vollständige Version der zugrunde liegenden IBM Campaign-Instanz.

**Anmerkung:** Für diesen Aufruf sind keine Clientinformationen erforderlich. Zudem werden keine Sicherheitsberechtigungen angewendet.

#### Parameter

Keine.

#### Rückgabe

Gibt eine Instanz vom Typ *WSServiceInfo* zurück.

#### Fehler

Keine.

---

### SOAP-API-Methoden: Attribute

Die meisten Daten von Komponenteninstanzen können von der SOAP-API von IBM Campaign in Form von Attributen oder Metadaten von Attributen angezeigt werden.

In einigen Fällen sind die Metadatendefinitionen für Attribute in IBM Campaign global (z. B. benutzerdefinierte Attribute der Kampagne). In anderen Fällen werden sie auf eine bestimmte Komponente eingeschränkt, z. B. auf Benutzervariablen von Ablaufdiagrammen. Sofern nicht anders angegeben, können alle Attribute gelesen werden, wenn der Client hierfür über ausreichend Sicherheitsberechtigungen verfügt.

**Anmerkung:** Nur Komponenten, die aktiv sind und auf die der Client Zugriff hat, werden von dieser API angezeigt. Öffentliche Unterstützung ist auf ein Subset der verfügbaren APIs begrenzt.

#### getAttributesByName

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

Ruft die benannten Attribute ab, die der angegebenen Komponenteninstanz (unter Umständen leer) zugeordnet sind.

## Parameter

*userCredential*: der Clientbenutzer-Berechtigungsnaehweis.

*requestedLocale*: optionale Ländereinstellung zur Verwendung bei dieser Anforderung; ist keine Ländereinstellung angegeben, werden die Benutzervorgaben für die Ländereinstellung der Benutzer von IBM verwendet. Bei Bedarf wird der Algorithmus zur Standardisierung der Ländereinstellung angewendet.

*partitionName*: optionaler Name der zu verwendenden Kampagnenpartition. Ist keine Partition definiert, wird die Standardpartition verwendet.

*reference*: die Referenz für die Komponenteninstanz mit den gewünschten Attributen. *InvalidComponentException* wird ausgelöst, wenn die Referenz ungültig ist oder die Komponente nicht vorhanden ist.

*names*: optionales Array mit den abzurufenden Attributnamen (keine Anzeigenamen); ist kein Array angegeben, werden alle Attribute zurückgegeben. Löst *AttributeNotFoundException* aus, wenn eines der benannten Attribute nicht vorhanden ist.

## Rückgabe

Eine typisierte Zuordnung von null oder mehr Attributen; der Attributname stellt den Zuordnungseingabeschlüssel und die Attributinstanz den Eingabewert dar.

## Fehler

*InvalidComponentException*, *AttributeNotFoundException*

*AuthorizationException*, *DataException*

**Anmerkung:** Alle diese Ausnahmen sind in der *CampaignServicesException* eingebunden.

## updateAttributes

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

Aktualisiert mindestens ein Attribut der Komponenteninstanz mit den angegebenen Attributwerten.

## Aktualisierungslogik

Die Aktualisierungslogik lautet wie folgt.

Bei jedem Attribut, das in der angegebenen Attributzuordnung enthalten ist:

1. Wenn der Attributname mit einem vorhandenen Attribut übereinstimmt, versuchen Sie, das zugehörige Feld *Werte* mit dem angegebenen Wertfeld zu überschreiben.
2. Wenn das Attribut noch nicht vorhanden ist, *allowCreate* auf "true" festgelegt ist und die zugehörigen Metadaten bekannt sind, dann erstellen Sie das Attribut. Dies gilt sowohl für Metadaten für globale Attribute als auch für Instanzattribute (mit Ausnahme von Ablaufdiagrammen).

3. Wenn der Werttyp oder ein anderer Aspekt der Metadatendefinition des Attributs nicht erfüllt ist oder mindestens einer der angegebenen Werte ungültig ist oder außerhalb des gültigen Bereichs liegt, wird *InvalidAttributeException* ausgelöst.
4. Andernfalls wird *AttributeNotFoundException* ausgelöst, wenn das benannte Attribut nicht vorhanden ist.

**Anmerkung:** Im Ausnahmefall wird keine der Aktualisierungen festgeschrieben.

Diese bestimmte Methode unterstützt nicht die Definition neuer benutzerdefinierter Attribute; verwenden Sie hierfür die Methode `createAttributeMetadata()`.

In allen Fällen unterliegt die Aktualisierungsoperation des Attributs den gewöhnlichen Integritätsbedingungen für die Sicherheit und einer Sicherheitsüberprüfung. Es liegt im Verantwortungsbereich des Clients, zu ermitteln, welche Attribute für eine bestimmte Komponenteninstanz erforderlich sind, die richtigen Typen zu ermitteln usw.

### Parameter

*userCredential*: der Clientbenutzer-Berechtigungsnauchweis.

*requestedLocale*: optionale Ländereinstellung zur Verwendung bei dieser Anforderung.

*partitionName*: optionaler Name der zu verwendenden Kampagnenpartition.

*reference*: die Referenz für die Komponenteninstanz mit den zu aktualisierenden Attributen.

*allowCreate*: gibt an, ob ein neues Attribut erstellt werden sollte, wenn es noch nicht in der Komponente vorhanden ist. (Siehe „Aktualisierungslogik“ auf Seite 24.)

*attributes*: ein Array mit zu aktualisierenden Attributen; mit dem Attributnamen wird das zu aktualisierende Attribut gesucht und mit den neuen Werten wird der bestehende Wert des Attributs als einzelnes Objekt des ordnungsgemäßen Typs oder, falls zutreffend, als Array aktualisiert. (Siehe Kapitel 5, „SOAP-API - Allgemeine Ausnahmen“, auf Seite 63.)

### Rückgabe

Keine.

### Fehler

*InvalidComponentException*, *AttributeNotFoundException*, *InvalidAttributeException*

*AuthorizationException*, *DataException*

## getAttributeMetadataByName

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

Ruft die Metadatendefinitionen für benannte Attribute ab, die an eine bestimmte Komponente oder Vorlage gebunden sind oder global definiert sind.

## Parameter

*userCredential*: der Clientbenutzer-Berechtigungsnaehweis.

*requestedLocale*: optionale Ländereinstellung zur Verwendung bei dieser Anforderung.

*partitionName*: optionaler Name der zu verwendenden Kampagnenpartition.

*reference*: optionale Referenz für die Komponente oder die Vorlage, die die Metadaten für die gewünschten Attribute enthält. Wenn nur `ComponentTypeEnum` angegeben ist, werden nur Komponenten von diesem Typ abgerufen. Wenn keine Referenz angegeben ist, werden beim Abrufen alle globalen Metadatendefinitionen, für alle Komponententypen, zurückgegeben. Löst `InvalidComponentException` aus, wenn die angegebene Referenz ungültig ist.

*names*: optionales Array mit den abzurufenden Metadaten für Attribute. Wenn das Array nicht angegeben ist, werden alle Metadaten der Komponente bzw. alle global definierten Metadaten (wenn keine Referenz angegeben ist) zurückgegeben. Löst `AttributeNotFoundException` aus, wenn mindestens ein Bereich der Metadatendefinitionen für die benannten Attribute nicht vorhanden ist.

## Rückgabe

Eine typisierte Zuordnung von null oder mehr Metadatendefinitionen für Attribute; der Attributname stellt den Zuordnungseingabeschlüssel und die Metadaten der Attribute stellen den Eingabewert dar.

## Fehler

`InvalidComponentException`, `AttributeNotFoundException`

`AuthorizationException`, `DataException`

## createAttributeMetadata

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

Erstellen Sie mindestens eine neue Metadatendefinition für Attribute und verbinden Sie diese optional mit einer bestimmten Komponente oder Vorlage.

## Parameter

*userCredential*: der Clientbenutzer-Berechtigungsnaehweis.

*requestedLocale*: optionale Ländereinstellung zur Verwendung bei dieser Anforderung.

*partitionName*: optionaler Name der zu verwendenden Kampagnenpartition.

*reference*: optionale Referenz für die Komponente oder Vorlage, mit der die Metadaten verbunden sein sollten. Ist diese Referenz nicht angegeben, ist die erstellte Metadatendefinition global. Wenn die Referenz angegeben wird, jedoch nicht gültig ist, wird `InvalidComponentException` ausgelöst.

*attributeMetadata*: ein Array mit den zu verbindenden Metadatendefinitionen für Attribute. Wenn mindestens ein Bereich der angegebenen Metadaten bereits mit

der Komponente verbunden ist, der Name also nicht eindeutig ist, wird *AttributeExistsException* ausgelöst. Löst *InvalidAttributeException* aus, wenn in mindestens einem Bereich der angegebenen Metadaten ein Problem auftritt und diese dadurch intern inkonsistent sind.

## Rückgabe

Keine.

## Fehler

*InvalidComponentException*, *AttributeExistsException*, *InvalidAttributeException*

*AuthorizationException*, *DataException*

## updateAttributeMetadata

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

Aktualisiert mindestens eine Metadatendefinition für Attribute der angegebenen Komponente oder Vorlage und erstellt dabei bei Bedarf optional neue Metadaten-Definitionen.

## Aktualisierungslogik

Die Aktualisierungslogik lautet wie folgt.

Bei jeder Metadatendefinition für Attribute im angegebenen Array:

1. Wenn der Attributname nicht mit vorhandenen Metadaten übereinstimmt, die an die Komponente gebunden sind, gehen Sie basierend auf dem Parameterwert *allowCreate* wie folgt vor:
  - a. *True*: Erstellen Sie eine neue Metadatendefinition. Funktional identisch zur Verwendung der Anforderung *createAttributeMetadata()*.
  - b. *False*: Lösen Sie *AttributeNotFoundException* aus.
2. Wenn der Datentyp der Metadaten für Attribute ein anderer ist, lösen Sie *InvalidAttributeException* aus.
3. Versuchen Sie, die vorhandene Metadatendefinition für Attribute mit den Feldwerten der angegebenen Metadaten zu überschreiben. Lösen Sie andernfalls *InvalidAttributeException* aus. Es werden nur die folgenden Aktualisierungen unterstützt (lösen Sie andernfalls *InvalidAttributeException* aus):
  - a. *name*: kann nicht geändert werden (name ist der Schlüssel!).
  - b. *displayName*: Akzeptieren des neuen Werts.
  - c. *description*: Akzeptieren des neuen Werts.
  - d. *isRequired*: Nur Änderung von *true* zu *false* zulässig.
  - e. *isInternal*: Akzeptieren des neuen Werts.
  - f. *isGenerated*: keine Änderung zulässig.
  - g. *attributeTypeEnum*: keine Änderung zulässig.
  - h. *accessTypeEnum*: Akzeptieren des neuen Werts.
  - i. *selectTypeEnum*: Akzeptieren dieser Übergänge, wenn Optionen bereitgestellt werden:
    - 1) NONE zu SINGLE\_SELECT oder MULTIPLE\_SELECT
    - 2) SINGLE\_SELECT zu MULTIPLE\_SELECT

- j. *options*: Optionen können hinzugefügt, jedoch nicht gelöscht werden. Es werden nur die folgenden Optionsänderungen unterstützt (gemäß der Wertübereinstimmung):
    - 1) *displayName*: Akzeptieren des neuen Werts (keine automatische Berechnung).
    - 2) *description*: Akzeptieren des neuen Werts (keine automatische Berechnung).
    - 3) *isDefault*: Akzeptieren des neuen Werts; muss jedoch SelectTypeEnum entsprechen.
    - 4) *value*: Es ist keine Änderung zulässig (value ist der Schlüssel!).
  - k. *defaultValue* (nur bei Ablaufdiagrammen): Akzeptieren des neuen Standardwerts.
  - l. *maxLength* (nur bei Text): Akzeptieren der neuen Länge, sofern sie größer ist.
4. Wenn die Metadatendefinition für Attribute intern nicht konsistent ist, lösen Sie *InvalidAttributeException* aus.
  5. Suchen Sie bei Bedarf alle Komponenteninstanzen, die auf die aktualisierten Metadaten des Attributs verweisen, und aktualisieren Sie sie entsprechend.

**Anmerkung:** Im Ausnahmefall wird keine der Aktualisierungen festgeschrieben.

In allen Fällen unterliegt die Aktualisierungsoperation des Attributs den gewöhnlichen Integritätsbedingungen für die Sicherheit und einer Sicherheitsüberprüfung.

Siehe `createAttributeMetadata()`, `deleteAttributeMetadata()`

## Parameter

*userCredential*: der Clientbenutzer-Berechtigungsnachweis.

*requestedLocale*: optionale Ländereinstellung zur Verwendung bei dieser Anforderung.

*partitionName*: optionaler Name der zu verwendenden Kampagnenpartition.

*reference*: optionale Referenz für die Komponenteninstanz mit den gewünschten Attributen. Ist keine Referenz angegeben, wird die Aktualisierung auf globale Metadatendefinitionen beschränkt. Löst *InvalidComponentException* aus, wenn die angegebene Referenz ungültig ist.

*allowCreate*: Ist "true" festgelegt, werden Metadatendefinitionen erstellt, die derzeit noch nicht vorhanden sind (funktional äquivalent zur Verwendung der Methode `createAttributeMetadata()`).

*attributeMetadata*: ein Array mit Metadatendefinitionen für Attribute, die aktualisiert werden müssen (und hinzugefügt werden müssen, wenn das Flag *allowCreate* auf "true" festgelegt ist). Mit dem Attributnamen wird die zu aktualisierende Metadatendefinition gesucht. Die verbleibenden Daten werden für die Aktualisierung der vorhandenen Definition verwendet. (Siehe „Aktualisierungslogik“ auf Seite 27.)

## Rückgabe

Keine.



## Fehler

InvalidComponentException, InvalidAttributeException

AuthorizationException, DataException

## deleteAttributeMetadata

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

Löscht mindestens eine Metadatendefinition für benannte Attribute aus der angegebenen Komponente, Vorlage (nur bei Metadaten für benutzerdefinierte Attribute) oder den Metadatendefinitionen für globale Attribute.

Im Rahmen dieser Aufgabe werden mithilfe der Methode alle Komponenten gesucht, die auf die gelöschten Metadaten verweisen und sie entsprechend aktualisieren.

**Anmerkung:** Im Ausnahmefall wird jedoch keiner der Löschvorgänge festgeschrieben.

## Parameter

*userCredential*: der Clientbenutzer-Berechtigungsnachweis.

*requestedLocale*: optionale Ländereinstellung zur Verwendung bei dieser Anforderung.

*partitionName*: optionaler Name der zu verwendenden Kampagnenpartition.

*reference*: optionale Referenz auf die Komponente oder Vorlage, die die zu löschenden Attribute enthält. Ist keine Referenz angegeben, wird der Löschvorgang auf globale Metadatendefinitionen eingeschränkt. Löst *InvalidComponentException* aus, wenn die angegebene Referenz ungültig ist.

**Anmerkung:** Wenn das optionale Array mit Namen der Metadaten für Attribute nicht angegeben ist, versucht diese Methode, alle der Komponente zugeordneten Metadaten für benutzerdefinierte Attribute bzw., wenn keine Referenz angegeben wurde, alle globalen Definitionen zu löschen.

*names*: optionales Array mit Namen der zu löschenden Metadaten für Attribute. Löst *AttributeNotFoundException* aus, wenn mindestens ein Bereich der Metadaten für die benannten Attribute nicht vorhanden ist. Löst *InvalidAttributeException* aus, wenn ein Attribut nicht entfernt werden konnte.

## Rückgabe

Keine.

## Fehler

InvalidComponentException, AttributeNotFoundException, InvalidAttributeException

AuthorizationException, DataException

---

## SOAP-API-Methoden: Kampagnen und Ablaufdiagramme

Die SOAP-API von IBM Campaign unterstützt in Kampagnen und Ablaufdiagrammen die folgenden Operationen (vorbehaltlich der Sicherheitsrichtlinien).

- Erstellen einer Kampagne
- Erkennung (Auflisten von Kampagnen nach verschiedenen Kriterien)
- Erstellen, Lesen und Aktualisieren von Attributen (über Attribut-APIs)
- Stoppen einer Ablaufdiagrammausführung

Kampagnen verfügen über einige ihnen zugeordnete Standardattribute, die von der API angezeigt werden. Diese Liste kann bei Bedarf vom Client durch Hinzufügen benutzerdefinierter Attribute erweitert werden (siehe die Attribut-APIs).

Die Standardkampagnenattribute sind im Folgenden aufgelistet:

- *uacName*: Kampagnenname (nicht garantiert eindeutig).
- *uacDescription*: optionale Zeichenfolge, die die Kampagne beschreibt.
- *uacCampaignCode*: ein Zeichenfolgecode, der die Kampagne eindeutig identifiziert. Er wird üblicherweise von Campaign generiert, kann aber auch vom Client bereitgestellt werden.
- *uacCreateDate*: ein Kalender, in dem das Datum und die Uhrzeit angegeben werden, zu dem bzw. zu der die Kampagne vom Server erstellt wurde.
- *uacUpdateDate*: ein Kalender, in dem das Datum und die Uhrzeit angegeben werden, zu dem bzw. zu der die Kampagne zuletzt vom Server aktualisiert wurde.
- *uacInitiative*: optionale Zeichenfolge, die die Kampagneninitiative definiert.
- *uacObjectives*: optionale Zeichenfolge, die die Ziele der Kampagne identifiziert.
- *uacStartDate*: ein optionaler Kalender, in dem das Datum und die Uhrzeit angegeben werden, zu dem bzw. zu der die Kampagne vom Server gestartet wurde oder gestartet werden soll.
- *uacEndDate*: wie *uacStartDate*, definiert jedoch das Datum und die Uhrzeit, zu dem bzw. zu der die Kampagne abgeschlossen wurde bzw. abgeschlossen werden soll. Muss hinter dem *uacStartDate* liegen.
- *uacLastRunDate*: ein optionaler Kalender, in dem das Datum und die Uhrzeit angegeben werden, zu dem bzw. zu der ein mit der Kampagne verbundenes Ablaufdiagramm zuletzt ausgeführt wurde (sonst null).
- *uacExternalLinkOwner*: eine optionale Zeichenfolge, die den Namen des Besitzers eines externen Links definiert (siehe Attribut *uacExternalLinkReference*). Nur zur Nutzung durch IBM ; muss eine der folgenden sein:
  - "Plan" (jetzt als IBM Marketing Operations bekannt)
  - "Collaborate" (jetzt als IBM Distributed Marketing bekannt)
- *uacExternalLinkId*: eine optionale numerische Datenbank-ID, die von einer anderen IBM Anwendung einem mit dieser Kampagne verbundenen Objekt zugeordnet wird. Nur zur Nutzung durch IBM : siehe auch das Attribut *uacExternalLinkOwner*.

### generateCampaignCode

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

Generiert einen neuen Kampagnencode.

Dieser Code ist garantiert eindeutig. Er unterscheidet sich von dem Wert, der durch einen vorherigen oder späteren Aufruf mit dieser Methode oder der Methode `createCampaign()` zurückgegeben wird, oder dem Wert, der für eine Kampagne generiert und über die grafische Benutzerschnittstelle von IBM Campaign erstellt wurde.

**Anmerkung:** Die Verwendung dieser Methode ist optional, da die API `createCampaign()` einen Kampagnencode für den Client generiert, wenn keiner bereitgestellt wird.

Siehe `createCampaign()`.

### Parameter

*userCredential*: der Clientbenutzer-Berechtigungsnachweis.

*requestedLocale*: optionale Ländereinstellung zur Verwendung bei dieser Anforderung.

*partitionName*: optionaler Name der zu verwendenden Kampagnenpartition. Wenn sich in der Campaign-Installation nur eine Partition befindet, ist dieses Argument unter Umständen null.

### Rückgabe

Der generierte Kampagnencode.

### Fehler

`AuthorizationException`, `DataException`

## deleteCampaigns

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

Löscht angegebene Kampagnen aus dem System.

### Parameter

*userCredential*: der Clientbenutzer-Berechtigungsnachweis.

*partitionName*: optionaler Name der zu verwendenden Kampagnenpartition.

*requestedLocale*: optionale Ländereinstellung zur Verwendung bei dieser Anforderung.

*wsReference*: Referenzen auf zu löschende Kampagnen.

### Rückgabe

Gibt ein Objekt vom Typ `WSDelateCampaignsResponse` zurück.

### Fehler

Löst `CampaignServicesException` aus, wenn die Kampagne nicht vorhanden ist, die Referenz ungültig ist oder keine Referenzen angegeben wurden.

## createCampaign

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

Erstellen Sie eine neue Kampagne für den Client, die Partition und securityPolicy-Name und wenden Sie dabei die angegebenen Attribute an. Alle von dieser API erstellten Kampagnen befinden sich unter diesem Stammordner.

### Parameter

*userCredential*: der Clientbenutzer-Berechtigungsnachweis.

*requestedLocale*: optionale Ländereinstellung zur Verwendung bei dieser Anforderung.

*partitionName*: optionaler Name der zu verwendenden Kampagnenpartition.

*securityPolicyName*: optionaler Name der Sicherheitsrichtlinie für die Kampagne, der zum Erstellen der Kampagne verwendet wird. Bei allen nachfolgenden Vorgängen in dieser Kampagne wird diese Richtlinie verwendet. Ist keine Sicherheitsrichtlinie definiert, wird die globale Richtlinie verwendet.

*name*: der Name, der der neuen Kampagneninstanz (dem zugehörigen Attribut "uacName") zugeordnet wird.

*attributes*: ein optionales Array mit Initialisierungsattributen; die Standardwerte der Kampagne werden durch die angegebenen Attribute überschrieben; andere bleiben unberührt. Beispiel: Wenn das Attribut *uacCampaignCode* angegeben wird, wird es anstelle des automatisch generierten Attributs verwendet. Es ist Aufgabe des Clients, die für die Kampagne erforderlichen Attribute, ihren Typ usw. zu ermitteln.

Löst *AttributeNotFoundException* aus, wenn mindestens eines der benannten Attribute nicht vorhanden ist, oder *InvalidAttributeException*, wenn ein Attributwert ungültig ist (z. B. ein falscher Datentyp).

### Rückgabe

Eine Einzelinstanz einer CampaignInfo für die erstellte Kampagne.

### Fehler

InvalidAttributeException, AttributeNotFoundException

AuthorizationException, DataException

## listCampaignsByPage

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

Zählt eine "Seite" mit Kampagnen auf, die mit den Werten des optionalen Attributs übereinstimmen, und beginnt dabei mit dem angegebenen Seitenoffset. Ordner werden ignoriert.

Sobald sie einmal abgerufen wurde, kann jede zurückgegebene *CampaignInfo* unverändert verwendet werden, um z. B. eine Auswertungsliste anzuzeigen. Alternativ können die Attributmethoden auch verwendet werden, um die Attribute der Kampagne abzurufen oder zu aktualisieren.

Über diese API wird kein Status gewartet. Daher können Aufrufe in einer beliebigen Reihenfolge durchgeführt werden.

## Parameter

*userCredential*: der Clientbenutzer-Berechtigungsnachweis.

*requestedLocale*: optionale Ländereinstellung zur Verwendung bei dieser Anforderung.

*partitionName*: optionaler Name der zu verwendenden Kampagnenpartition.

*attributes*: optionales Array mit Attributen, die übereinstimmen müssen; die Übereinstimmung wird anhand des Namens, des Datentyps und der Werte des Attributs ermittelt; wenn das Attribut Arrays unterstützt, müssen alle angegebenen Werte übereinstimmen. Der implizierte Übereinstimmungsoperator lautet AND. Folglich werden nur Kampagnen zurückgegeben, die mit allen angegebenen Attributwerten übereinstimmen.

Löst *AttributeNotFoundException* aus, wenn kein Attributname vorhanden ist, oder *InvalidAttributeException*, wenn mindestens eines der angegebenen Attribute ungültig ist.

*pageOffset*: das Startoffset aller möglichen Kampagnen für den Beginn der Aufzählung (mit dem Wert null). Beispiel: Wenn bei der Aufzählung 1000 Übereinstimmungen mit Kampagnen gefunden werden und dieser Wert auf 10 festgelegt ist, würde auf der Seite mit der 11. Komponente begonnen werden. Wenn sich das angegebene Offset außerhalb des gültigen Bereichs befindet, wird *RangeException* ausgelöst.

*pageSize*: die maximale Anzahl der abgeglichenen Kampagnen, die für die Seite zurückgegeben werden sollen (500 kann nicht überschritten werden).

## Rückgabe

Eine typisierte Liste mit null oder mehr Datenwrapperinstanzen von *CampaignInfo*, eine für jede abgeglichene Kampagne auf der Seite.

## Fehler

*AttributeNotFoundException*, *InvalidAttributeException*, *RangeException*

*InvalidExecutionContextException*, *AuthorizationException*

## stopFlowchart

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

Diese API stoppt ein derzeit ausgeführtes Ablaufdiagramm. Bei einer einzelnen Listenerkonfiguration kann eine Ablaufdiagrammausführung eindeutig durch die PID identifiziert werden, die der Ablaufdiagrammausführung zugeordnet ist. Die

PID gibt die Prozess-ID des Prozesses `unica_acsvr` an. Wenn mehrere IBM Campaign-Listener konfiguriert wurden, müssen Sie sowohl die Ausführungs-ID, die der Ablaufdiagrammausführung zugeordnet ist, als auch die PID einbeziehen.

## Parameter

*pid*: Einer Ablaufdiagrammausführung zugeordnete Prozess-ID des Prozesses `unica_acsvr`.

*runid*: Einer Ablaufdiagrammausführung zugeordnete Ausführungs-ID. Erforderlicher Parameter bei einer in Gruppen zusammengefassten Listenerkonfiguration. Optionaler Parameter, wenn ein einzelner Listener konfiguriert wurde.

## Rückgabe

Keine

## Fehler

Keine

---

## SOAP-API-Methoden: Zielzellen

Zielzellen sind eine Abstraktion einiger bekannter Subsets von Kampagnenergebnissen, die von IBM Campaign als Arbeitsblatt für Zielzellen (Target Cell Spreadheet, TCS) verwaltet werden. Zielzellen können in einer Kampagne global sein oder einem bestimmten Kampagnenablaufdiagramm zugeordnet sein.

Die SOAP-API von IBM Campaign unterstützt in Zielzellen die folgenden Operationen:

- Erstellen von mindestens einer neuen globalen Zielzelle
- Massenaktualisierung von mindestens einer bestehenden Zielzelle
- Ermittlung (Auflisten von Zielzellen)
- Erstellen, Lesen und Aktualisieren von Attributen (über Attribut-APIs)
- Löschen einer bestehenden Zielzelle
- Aufrufen von Ausführungsergebnissen, die mindestens einer Zelle zugeordnet sind

Zielzellen verfügen über einige Standardattribute, die ihnen zugeordnet sind und von der API angezeigt werden. Diese Liste kann bei Bedarf vom Client erweitert werden, indem Metadatendefinitionen für benutzerdefinierte Attribute hinzugefügt werden (siehe die Attribut-APIs). Alle Metadaten für Attribute können als eine Spalte im TCS betrachtet werden; das Layout des Arbeitsblatts bestimmt der Client.

Standardzielzellenattribute sind:

- *uacName*: Zellenname.
- *uacDescription*: optionale Zeichenfolge, die das Ablaufdiagramm beschreibt.
- *uacCellCode*: eine Codezeichenfolge, die die Zelle eindeutig definiert. Sie wird üblicherweise von Campaign automatisch generiert, kann jedoch auch vom Client bereitgestellt werden.
- *uacCreateDate*: eine Kalenderinstanz, die das Datum & und die Uhrzeit angibt, zu dem bzw. zu der die Zelle vom Server erstellt wurde.

- *uacUpdateDate*: eine Kalenderinstanz, die definiert, wann die Zelle zuletzt vom Server aktualisiert wurde.
- *uacIsControl*: ein boolescher Wert, der angibt, ob es sich um eine Kontrollzelle (true) oder nicht (false) handelt. In anderen Zellen auf diese Zelle wird unter Umständen als Kontrollzelle verwiesen (siehe *uacControlCell*).
- *uacControlCell*: optionale Referenz der Kontrollzelle (bei einer Kontrollzelle nicht zulässig). Siehe das Attribut *uacIsControl*.
- *uacIsApproved*: ein boolescher Wert, der angibt, ob die Zelle genehmigt wird (true) oder nicht (false).
- *uacIsReadOnly*: ein boolescher Wert, der angibt, ob die Zelle schreibgeschützt ist (true) oder nicht (false).
- *uacDisplayOrder*: eine Ganzzahl, die die Reihenfolge dieser Zelle (Zeile) relativ zu anderen in diesem Arbeitsblatt für Zielzellen angibt.
- *uacIsTopDown*: ein boolescher Wert, der angibt, ob es sich bei der Zelle um eine Top-down-Zelle handelt.
- *uacAssignedOffers*: ein optionales Array mit mindestens einer Referenz der Angebote oder Angebotslisten, die dieser Zelle zugeordnet sind (bei einer Kontrollzelle nicht zulässig).
- *uacFlowchartName*: optionaler Name des Ablaufdiagramms, mit dem diese Zelle verbunden ist (Schreibschutz muss über die grafische Benutzerschnittstelle von IBM Campaign festgelegt werden; bei einer Kontrollzelle nicht zulässig).
- *uacFlowchartId*: optionale Datenbank-ID des Ablaufdiagramms, mit dem diese Zelle verbunden ist (schreibgeschützt, wie oben angegeben).

## createTargetCell

```

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

```

Erstellt eine neue kampagnenspezifische Zeile für Zielzellen und wende dabei die angegebenen Attribute pro Zelle und die Benutzerinformationen an.

Die angegebenen Attribute können Standard oder benutzerdefiniert sein; wenn sie benutzerdefiniert sind, müssen jedoch die Metadatendefinitionen der entsprechenden globalen Attribute vorhanden sein.

Sobald die Zielzelle erstellt wurde, können Attributwerte über die APIs der Attribute geändert werden.

Siehe `listTargetCells()`, `bulkCreateTargetCells()`.

Siehe `createAttributeMetadata()`, `listAttributeMetadata()`, `getAttributesByName()`

### Parameter

*userCredential*: der Clientbenutzer-Berechtigungsnachweis.

*requestedLocale*: optionale Ländereinstellung zur Verwendung bei dieser Anforderung.

*partitionName*: optionaler Name der zu verwendenden Kampagnenpartition.

*campaignReference*: die Referenz der Kampagne, die das zu aktualisierende Arbeitsblatt für Zielzellen enthält. Kumuliert eine *InvalidComponentException*, wenn die Kampagne nicht vorhanden ist oder die Referenz ungültig ist.

*attributes*: optionales Array mit TCS-Attributen für die neue Zelle. Die Standardwerte des entsprechenden Zellenattributs werden von den einzelnen angegebenen Attributelementen überschrieben; andere bleiben unberührt. Es ist Aufgabe des Kunden, die für die Zelle erforderlichen Attribute, ihren Typ usw. zu ermitteln. Eine *InvalidAttributeException* wird kumuliert, wenn bei einem bestimmten Attribut ein Problem auftritt.

Wenn Ausnahmen kumuliert werden, löst diese Methode eine *CompositeException* aus und alle Erstellungen werden rückgängig gemacht. Die Liste mit den Ursachen für die Ausnahme enthält zu jedem Attribut, das den Fehler verursacht hat, eine Ausnahme und einen numerischen Index anstelle der *Referenz*, des Attributnamens und üblicherweise des auslösenden Werts. Die Liste mit den Ursachen wird wie die Eingabeliste *attributeList* sortiert.

## Rückgabe

Ein Datenwrapper von *TargetCellInfo* für die erstellte TCS-Zelle.

## Fehler

*InvalidComponentException*, *CompositeException*

*AuthorizationException*, *DataException*

## bulkCreateTargetCells

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

Erstellt viele neue kampagnenspezifische Zeilen für Zielzellen gleichzeitig und wendet dabei die angegebenen Attribute pro Zelle und die Benutzerinformationen an.

Die angegebenen Attribute können Standard oder benutzerdefiniert sein; wenn sie benutzerdefiniert sind, müssen jedoch die Metadatendefinitionen der entsprechenden globalen Attribute vorhanden sein.

Wenn die Zielzelle erstellt wurde, können Attributwerte über die APIs der Attribute geändert werden.

Siehe `listTargetCells()`.

Siehe `createAttributeMetadata()`, `listAttributeMetata()`, `getAttributesByName()`

## Parameter

*userCredential*: der Clientbenutzer-Berechtigungs nachweis.

*requestedLocale*: optionale Ländereinstellung zur Verwendung bei dieser Anforderung.

*partitionName*: optionaler Name der zu verwendenden Kampagnenpartition.



*campaignReference*: die Referenz der Kampagne, die das zu aktualisierende Arbeitsblatt für Zielzellen enthält. Kumuliert eine *InvalidComponentException*, wenn die Kampagne nicht vorhanden ist oder die Referenz ungültig ist.

*attributeList*: optionale Liste mit Attributarrays pro Zelle, jeweils einem für jede zu erstellende Zeile für Zielzellen. Alle für ein bestimmtes Listenelement bereitgestellten Attribute überschreiben die Standardwerte des entsprechenden Zellenattributs; andere bleiben unberührt. Es ist Aufgabe des Kunden, die für die Zelle erforderlichen Attribute, ihren Typ usw. zu ermitteln. Eine *InvalidAttributeException* wird kumuliert, wenn bei einem bestimmten Attribut ein Problem auftritt.

Wenn Ausnahmen kumuliert werden, löst diese Methode eine *CompositeException* aus und alle Erstellungen werden rückgängig gemacht. Die Liste mit den Ursachen für die Ausnahme enthält zu jedem Attribut, das den Fehler verursacht hat, eine Ausnahme und einen numerischen Index anstelle der *Referenz*, des Attributnamens usw. Die Liste mit den Ursachen wie die Eingabeliste *attributeList* sortiert.

## Rückgabe

Eine Liste der Datenwrapper von *TargetCellInfo*, einem für jede erstellte Instanz, die gemäß der Elementreihenfolge des Eingabeparameters *attributesList* sortiert sind.

## Fehler

*InvalidComponentException*, *CompositeException*

*AuthorizationException*, *DataException*

## listTargetCells

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

Listet Informationen zu allen derzeit vorhandenen Zielzellen auf, die mit den angegebenen Attributen übereinstimmen, und zwar entweder für die angegebene Kampagne oder global, wenn keine Kampagne angegeben ist.

Siehe *getAttributeMetadata()*, *getAttributesByName()*.

## Parameter

*userCredential*: der Clientbenutzer-Berechtigungsname.

*requestedLocale*: optionale Ländereinstellung zur Verwendung bei dieser Anforderung.

*partitionName*: optionaler Name der zu verwendenden Kampagnenpartition.

*campaignReference*: Referenz der übergeordneten Kampagne. Löst *InvalidComponentException* aus, wenn die Kampagne nicht vorhanden ist oder die Referenz ungültig ist.

*attributes*: optionales Array mit abzugleichenden Attributen. Der implizierte Übereinstimmungsoperator lautet AND. Folglich werden nur Zellen zurückgegeben, die mit allen angegebenen Attributwerten übereinstimmen.

Löst *InvalidAttributeException* aus, wenn mindestens eines der angegebenen Attribute ungültig ist.

## Rückgabe

Gibt eine Liste mit null oder mehr Instanzen von *TargetCellInfo* für die abgeglichenen Zellen zurück.

## Fehler

*InvalidComponentException*, *InvalidAttributeException*

*AuthorizationException*, *DataException*

## bulkUpdateTargetCells

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

Aktualisierung der Attribute von mindestens einer Zielzelle.

Die Aktualisierungslogik lautet wie folgt.

Bei jedem Element in der bereitgestellten *attributesMap* stellt die Referenz auf die zu aktualisierende Zielzelle den Eingabeschlüssel dar und der Eingabewert ist ein Array von Aktualisierungsattributen für diese Zelle. Wenn die Zielzelle nicht vorhanden ist, wird eine *InvalidComponentException* kumuliert.

Wenn eine Zielzelle gefunden wurde, gehen Sie bei jedem angegebenen Attribut wie folgt vor:

1. Wenn der Attributname mit einem vorhandenen Attribut übereinstimmt, versuchen Sie, das zugehörige Wertfeld mit dem angegebenen Wertfeld zu überschreiben.
2. Wenn der Werttyp oder ein anderer Aspekt der Metadatendefinition des Attributs nicht erfüllt ist oder mindestens einer der angegebenen Werte ungültig ist oder außerhalb des gültigen Bereichs liegt, wird *InvalidAttributeException* kumuliert.
3. Andernfalls wird *AttributeNotFoundException* kumuliert, wenn das benannte Attribut nicht vorhanden ist.

Wenn Ausnahmen kumuliert werden, löst diese Methode eine *CompositeException* aus und alle Aktualisierungen werden rückgängig gemacht. Die Liste mit den Ursachen für die Ausnahmen enthält die oben aufgelisteten Ausnahmen. Bei jedem Attribut, das den Fehler verursacht hat, werden die Referenz und der Attributname aufgezeichnet.

In allen Fällen unterliegt die Aktualisierungsoperation des Attributs den gewöhnlichen Integritätsbedingungen für die Sicherheit und einer Sicherheitsüberprüfung. Es liegt im Verantwortungsbereich des Clients, zu ermitteln, welche Attribute für eine bestimmte Komponenteninstanz erforderlich sind, die richtigen Typen zu ermitteln usw.

## Parameter

*userCredential*: der Clientbenutzer-Berechtigungsnachweis.

*requestedLocale*: optionale Ländereinstellung zur Verwendung bei dieser Anforderung.

*partitionName*: optionaler Name der zu verwendenden Kampagnenpartition.

*attributesMap*: eine Übersicht über die zu aktualisierenden Zielzellen; der Eingabeschlüssel ist die Referenz auf die zu aktualisierende Zelle und der Eingabewert ist ein Array der zu aktualisierenden Attribute. Mit dem Attributnamen wird das zu aktualisierende Attribut gesucht und mit den neuen Attributwerten wird der Wert des vorhandenen Attributs als einfaches Objekt des ordnungsgemäßen Typs oder, falls zutreffend, als Array aktualisiert. Siehe die obenstehenden Ausnahmen.

## Rückgabe

Keine.

## Fehler

ComponentException

AuthorizationException, DataException

## getRunResultsByCell

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

Rufen Sie die Ausführungsergebnisse von mindestens einer Zielzelle ab, möglicherweise bei einem Ablaufdiagramm, das nie gestartet wurde oder noch in Bearbeitung ist.

## Parameter

*userCredential*: der Clientbenutzer-Berechtigungsname.

*requestedLocale*: optionale Ländereinstellung zur Verwendung bei dieser Anforderung.

*partitionName*: optionaler Name der zu verwendenden Kampagnenpartition.

*cellReferences*: ein Array mit Referenzen der Zielzellen, deren Ausführungsergebnisse erwünscht sind. Löst *InvalidComponentException* aus, wenn mindestens eine Zellenreferenz ungültig ist oder auf eine nicht vorhandene Zelle verwiesen wird.

## Rückgabe

Gibt eine typisierte Liste mit Ausführungsergebnissen für die benannten Zellen an, die entsprechend des Arrays mit den Eingabereferenzen sortiert sind.

Jeder Ausführungsstatus lautet `RUNNING`, wenn das Prozessfeld des zugrunde liegenden Ablaufdiagramms noch ausgeführt wird, `FAILED`, wenn die Ausführung aus einem bestimmten Grund fehlgeschlagen ist, oder `NOT_STARTED`, wenn die Ausführung des Prozessfeldes nicht gestartet wurde. Auch Details zum Status werden angegeben.

## Fehler

InvalidComponentException

AuthorizationException, DataException

## bulkDeleteTargetCells

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

Löscht mindestens eine vorhandene Zielzelle und alle davon abhängigen Komponenten (d. h. Ablaufdiagrammverknüpfungen, Attribute).

### Parameter

*userCredential*: der Clientbenutzer-Berechtigungs-nachweis.

*requestedLocale*: optionale Ländereinstellung zur Verwendung bei dieser Anforderung.

*partitionName*: optionaler Name der zu verwendenden Kampagnenpartition.

*cellReferences*: ein Array oder mindestens eine Referenz auf zu löschende Zellen. *InvalidComponentException* wird kumuliert, wenn bei einer der angegebenen Referenzen ein Problem vorliegt oder eine Zelle nicht vorhanden ist.

Wenn Ausnahmen kumuliert werden, löst diese Methode eine *CompositeException* aus und macht alle Löschvorgänge rückgängig. Die Liste mit den Ursachen für die Ausnahmen enthält die oben aufgelisteten Ausnahmen. Für jede Zelle, die den Fehler verursacht hat, wird die Referenz aufgezeichnet.

### Rückgabe

Keine.

### Fehler

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
```

Aktualisiert Attribute angegebener Vorlagen

### Parameter

*userCredential*: der Clientbenutzer-Berechtigungs-nachweis.

*partitionName*: optionaler Name der zu verwendenden Kampagnenpartition.

*requestedLocale*: optionale Ländereinstellung zur Verwendung bei dieser Anforderung.

*wsCampaignReference*: Referenz der übergeordneten Kampagne.

*allowCreate*: wird jetzt nicht verwendet.

*clearExisting*: Wenn das Flag auf "true" festgelegt ist, werden alle vorherigen Werte in der Vorlage gelöscht, die bei der Anforderung nicht gesendet werden.

*wsStaticAttributes*: Liste der statischen Attribute in der Vorlage.

*wsHiddenAttributes*: Liste der ausgeblendeten Attribute in der Vorlage.

*wsParametricAttributes*: Liste der parametrischen Attribute in der Vorlage.

## Rückgabe

Keine.

## Fehler

Löst CampaignServicesException aus, wenn die Angebotsvorlage nicht vorhanden ist, die Referenz ungültig ist oder keine Referenzen angegeben wurden.

## listBottomUpTargetCells

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

Listet Informationen zu allen Bottom-up-Zielzellen auf, die derzeit für die angegebene Kampagne vorhanden sind.

## Parameter

*userCredential*: der Clientbenutzer-Berechtigungsnauchweis.

*partitionName*: optionaler Name der zu verwendenden Kampagnenpartition.

*requestedLocale*: optionale Ländereinstellung zur Verwendung bei dieser Anforderung.

*wsCampaignReference*: Referenz der übergeordneten Kampagne.

## Rückgabe

Gibt bei den abgeglichenen Zellen eine Liste mit null oder mehr Instanzen vom Typ WSTargetCellDetails zurück

## Fehler

Löst CampaignServicesException aus, wenn die Kampagne nicht vorhanden oder die Referenz ungültig ist.

---

## SOAP-API-Methoden: Analysen

Die SOAP-API von IBM Campaign unterstützt das Abrufen einfacher Metriken von IBM Campaign.

## getCampaignMetrics

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

Ruft die Metrik der angegebenen Kampagne ab.

### Parameter

*userCredential*: der Clientbenutzer-Berechtigungs-nachweis.

*requestedLocale*: optionale Ländereinstellung zur Verwendung bei dieser Anforderung.

*partitionName*: optionaler Name der zu verwendenden Kampagnenpartition.

*campaignReference*: die Referenz der übergeordneten Kampagne. Löst *InvalidComponentException* aus, wenn bei der Kampagnenreferenz ein Problem auftritt oder die Kampagne nicht vorhanden ist.

### Rückgabe

Gibt eine Instanz vom Typ *MetricsInfo* für die Kampagne zurück.

### Fehler

*InvalidComponentException*

*AuthorizationException*, *DataException*

---

## SOAP-API-Methoden: Angebote, Angebotslisten, Angebotsvorlagen

Die SOAP-API von IBM Campaign unterstützt die folgenden mit Angeboten verbundenen Operationen.

- Ermittlung: Auflistung nach Ordner (Angebote, Angebotslisten und Unterordner), Attribut (Angebote und Angebotsvorlagen) oder Suchwert (Angebote)
- Validierung
- Informationsabruf (Abrufen von Attributen eines bestimmten Angebots oder einer bestimmten Angebotsvorlage)
- Erstellen, Bearbeiten, Zurückziehen und Löschen von Angeboten

Angeboten sind einige Standardattribute zugeordnet. Diese Liste kann vom Client erweitert werden, indem er Metadatendefinitionen für benutzerdefinierte Attribute hinzufügt (siehe die Attribut-APIs).

Standardattribute von Angeboten sind:

- *uacName*: Angebotsname.
- *uacDescription*: optionale Zeichenfolge, die das Angebot beschreibt.
- *uacOfferCode*: eine Codezeichenfolge, mit der das Angebot eindeutig identifiziert wird. Sie wird üblicherweise von IBM Campaign generiert, kann jedoch unter Umständen auch vom Client bereitgestellt werden.
- *uacCreateDate*: eine Kalenderinstanz, die das Datum und die Uhrzeit angibt, zu dem bzw. zu der das Angebot vom Server erstellt wurde.
- *uacUpdateDate*: eine Kalenderinstanz, die den Zeitpunkt angibt, zu dem das Angebot zuletzt vom Server aktualisiert wurde.

Angebotsvorlagen verfügen auch über Standardattribute und benutzerdefinierte Attribute. Standardattribute von Angebotsvorlagen sind:

- *uacName*: Name der Angebotsvorlage.
- *uacDescription*: optionale Zeichenfolge, die die Angebotsvorlage beschreibt.
- *uacCreateDate*: eine Kalenderinstanz, die das Datum und die Uhrzeit angibt, zu dem bzw. zu der die Angebotsvorlage vom Server erstellt wurde.
- *uacUpdateDate*: eine Kalenderinstanz, die angibt, wann die Angebotsvorlage zuletzt vom Server aktualisiert wurde.

## listOffersAndFolders

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

Listet alle Angebote, Angebotslisten und Ordner unter dem optionalen übergeordneten Ordner auf.

Nachdem sie abgerufen wurde, kann jede zurückgegebene Instanz von *WSComponentOrFolderInfo* unverändert verwendet werden. Beispiel: Um die nächsthöhere Ebene der Ordnerhierarchie anzuzeigen, können die Attribut-APIs zum Abrufen oder Aktualisieren der enthaltenen Angebote verwendet werden.

### Parameter

*userCredential*: der Clientbenutzer-Berechtigungsname.

*requestedLocale*: optionale Ländereinstellung zur Verwendung bei dieser Anforderung.

*partitionName*: optionaler Name der zu verwendenden Kampagnenpartition.

*parentReference*: optionale Referenz des übergeordneten Ordners, die aufgelistet werden soll. Nur die direkt untergeordneten Angebote, Angebotslisten und Ordner dieses übergeordneten Ordners werden aufgezählt. Daher müssen aufeinanderfolgende Aufrufe in dieser API durch die gesamte Ordnerhierarchie navigieren (die Hierarchie ist jedoch sehr flach). Wenn kein übergeordneter Ordner angegeben ist, werden alle Komponenten und Ordner im Stammverzeichnis zurückgegeben.

Löst *InvalidFolderException* aus, wenn ein Problem mit der Referenz des angegebenen übergeordneten Ordners auftritt.

Eine typisierte *Liste* mit null oder mehr Datenwrapperinstanzen von *WSComponentOrFolderInfo*, eine für jede abgeglichene Komponente oder jeden abgeglichenen Ordner.

### Fehler

*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;
```

Zählt eine "Seite" mit Angeboten auf, die bei den Feldern für den Namen, die Beschreibung, createBy oder den Angebotscode die vorgegebenen Suchkriterien enthalten, beginnend mit dem angegebenen Seitenoffset. Die Suche basiert auf der optionalen Ordnerangabe. (Wenn die folderID 0 angegeben ist, wird standardmäßig der Stammangebotsordner verwendet). Übereinstimmungen werden basierend auf einer Übereinstimmung vom Typ "enthält" bei der Suchzeichenfolge zurückgegeben.

Sobald sie einmal abgerufen wurde, kann jede zurückgegebene *WSOfferInfo* unverändert verwendet werden, z. B. um eine Auswertungsliste anzuzeigen. Alternativ können die Attributmethoden auch verwendet werden, um die Attribute des Angebots abzurufen oder zu aktualisieren.

Über diese API wird kein Status erwartet. Daher können Aufrufe in einer beliebigen Reihenfolge durchgeführt werden.

### Parameter

*userCredential*: der Clientbenutzer-Berechtigungsname.

*requestedLocale*: optionale Ländereinstellung zur Verwendung bei dieser Anforderung.

*partitionName*: optionaler Name der zu verwendenden Kampagnenpartition.

*folderID*: die ID des Angebotsordners, der durchsucht werden soll; wenn die folderID 0 angegeben ist, wird der Stammordner durchsucht.

*searchCriteria*: der Suchausdruck.

*includeRetired*: der boolesche Wert, der angibt, ob in Suchergebnissen zurückgewiesene Angebote enthalten sind. Gültige Werte sind TRUE und FALSE, wobei TRUE angibt, dass zurückgewiesene Angebote enthalten sind, und FALSE angibt, dass keine zurückgewiesenen Angebote enthalten sind.

*pageOffset*: das Startoffset aller möglichen Komponenten für den Beginn der Aufzählung (mit dem Wert null). Beispiel: Wenn bei der Aufzählung 1000 Übereinstimmungen mit Angeboten gefunden werden und dieser Wert auf 10 festgelegt ist, würde auf der Seite mit der 11. Komponente begonnen werden. Wenn sich das angegebene Offset außerhalb des gültigen Bereichs befindet, wird eine RangeException ausgelöst.

*pageSize*: die maximale Anzahl der abgeglichenen Komponenten, die für die Seite zurückgegeben werden sollen (500 kann nicht überschritten werden).

### Rückgabe

Gibt eine typisierte Liste mit null oder mehr Datenwrapperinstanzen von *Offer* zurück, eine für jedes zurückgegebene Angebot auf der Seite.



## Fehler

RangeException

## listOffersByPage

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

Zählt eine "Seite" mit Angeboten auf, die mit den Werten des optionalen Attributs übereinstimmen, und beginnt dabei mit dem angegebenen Seitenoffset. Ordner werden ignoriert. Übereinstimmungen werden basierend auf einer "ähnlichen" Übereinstimmung bei Zeichenfolgen (wobei eine Übereinstimmung als ausreichend betrachtet wird, wenn eine Zeichenfolge den abgefragten Wert enthält) und einer genauen Übereinstimmung bei Datumsangaben und Zahlen zurückgegeben.

Nachdem sie abgerufen wurde, kann jede zurückgegebene *OfferInfo* unverändert verwendet werden, um z. B. eine Auswertungsliste anzuzeigen. Alternativ können die Attributmethoden verwendet werden, um die Attribute des Angebots abzurufen oder zu aktualisieren.

Über diese API wird kein Status erwartet. Daher können Aufrufe in einer beliebigen Reihenfolge durchgeführt werden.

### Parameter

*userCredential*: der Clientbenutzer-Berechtigungsname.

*requestedLocale*: optionale Ländereinstellung zur Verwendung bei dieser Anforderung.

*partitionName*: optionaler Name der zu verwendenden Kampagnenpartition.

*attributes*: optionales Array mit Attributen, die übereinstimmen müssen; die Übereinstimmung wird anhand des Namens, des Datentyps und der Werte des Attributs ermittelt; wenn das Attribut Arrays unterstützt, müssen alle angegebenen Werte übereinstimmen. Der implizierte Übereinstimmungsoperator lautet OR. Folglich werden Komponenten zurückgegeben, die mit einem beliebigen angegebenen Attributwert übereinstimmen.

Löst *AttributeNotFoundException* aus, wenn ein Attributname nicht vorhanden ist, oder *InvalidAttributeException*, wenn mindestens eines der angegebenen Attribute ungültig ist.

*pageOffset*: das Startoffset aller möglichen Komponenten für den Beginn der Aufzählung (mit dem Wert null). Beispiel: Wenn bei der Aufzählung 1000 Übereinstimmungen mit Angeboten gefunden werden und dieser Wert auf 10 festgelegt ist, würde auf der Seite mit der 11. Komponente begonnen werden. Wenn sich das angegebene Offset außerhalb des gültigen Bereichs befindet, wird *RangeException* ausgelöst.

*pageSize*: die maximale Anzahl der abgeglichenen Komponenten, die für die Seite zurückgegeben werden sollen (500 kann nicht überschritten werden).

## Rückgabe

Eine typisierte Liste mit null oder mehr Datenwrapperinstanzen von *OfferInfo*, eine für jede abgeglichene Komponente auf der Seite.

## Fehler

*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
```

Erstellt eine neue dynamische Angebotsliste.

## Parameter

*userCredential*: der Clientbenutzer-Berechtigungsnachweis.

*partitionName*: optionaler Name der zu verwendenden Kampagnenpartition.

*requestedLocale*: optionale Ländereinstellung zur Verwendung bei dieser Anforderung.

*name*: Name des zu erstellenden Ordners.

*description*: Beschreibung des neuen Ordners.

*securityPolicyName*: Name der zu verwendenden Sicherheitsrichtlinie.

*parentFolder*: ID des Ordners, in dem die Angebotsliste erstellt werden muss.

*offerListInfo*: Objekt vom Typ *WSSmartListInfo*.

*createdBy*: (optional) Objekt vom Typ *WSApplicationTypeEnum* gibt an, mit welcher Anwendung der Ordner erstellt wurde. Mögliche Werte: *Campaign/Plan/ Collaborate/eMessage*. Ist kein Wert angegeben, wird die Kampagne verwendet.

*creatorObjectId*: (optional) wird im Plan (IBM Marketing Operations) verwendet, um einen Ordner in IBM Campaign mit einem Ordner in Marketing Operations zu verbinden.

## Rückgabe

Gibt ein Objekt vom Typ *WSCreateSmartOfferListResponse* zurück.

## Fehler

Löst *CampaignServicesException* aus, wenn die ID von *parentFolder* ungültig ist oder *offerListInfo* nicht angegeben ist.

Löst *CampaignServicesException* aus, wenn der Listenname doppelt vorhanden ist.

## 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
```

Erstellt eine neue statische Angebotsliste.

### Parameter

*userCredential*: der Clientbenutzer-Berechtigungsnachweis.

*partitionName*: optionaler Name der zu verwendenden Kampagnenpartition.

*requestedLocale*: optionale Ländereinstellung zur Verwendung bei dieser Anforderung.

*name*: Name des zu erstellenden Ordners.

*description*: Beschreibung des neuen Ordners.

*securityPolicyName*: Name der zu verwendenden Sicherheitsrichtlinie.

*parentFolder*: ID des Ordners, in dem die Angebotsliste erstellt werden muss.

*listMembers*: verweist auf Angebote, die in der Angebotsliste enthalten sein sollen.

*createdBy*: (optional) Objekt vom Typ `WSApplicationTypeEnum` gibt an, mit welcher Anwendung der Ordner erstellt wurde. Mögliche Werte: Campaign/Plan/Collaborate/eMessage. Ist kein Wert angegeben, wird Campaign verwendet.

*creatorObjectId*: (optional) wird im Plan (IBM Marketing Operations) verwendet, um einen Ordner in IBM Campaign mit einem Ordner in Marketing Operations zu verbinden.

### Rückgabe

Gibt ein Objekt vom Typ `WSCreateStaticOfferListResponse` zurück

### Fehler

Löst `CampaignServicesException` aus, wenn die ID von `parentFolder` ungültig ist oder `listMembers` ungültig sind.

Löst `CampaignServicesException` aus, wenn der Listenname doppelt vorhanden ist.

## getOffers

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

Listet Details zu Angeboten auf, wie sie in der Anforderung angegeben sind.

## Parameter

*userCredential*: der Clientbenutzer-Berechtigungsnaehweis.

*partitionName*: optionaler Name der zu verwendenden Kampagnenpartition.

*requestedLocale*: optionale Ländereinstellung zur Verwendung bei dieser Anforderung.

*wsCampaignReference*: Referenz der übergeordneten Kampagne.

## Rückgabe

Gibt ein Objekt vom Typ `WSGetOffersResponse` zurück.

## Fehler

Löst `CampaignServicesException` aus, wenn das Angebot nicht vorhanden ist, die Referenz ungültig ist oder keine Referenzen angegeben sind.

## validateOffers

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

Überprüft die angegebenen Angebotscodes oder Namen der Angebotslisten und gibt jeweils Überprüfungsinformationen zurück. "Überprüfung" besteht aus der Prüfung, ob in der Datenbank nur ein übereinstimmendes Angebot oder eine übereinstimmende Angebotsliste vorhanden ist.

Das Objekt `OfferValidationInfo` enthält anstelle von Angebotsinformationen eine Fehlermeldung, wenn null Angebote oder Angebotslisten gefunden werden, die mit dem angegebenen Code oder Namen übereinstimmen. Ein Fehler wird auch dann anstelle einer Übereinstimmung zurückgegeben, wenn der angegebene Code oder Name mit mehreren Angeboten oder Angebotslisten übereinstimmt. Die Liste wird in derselben angegebenen Reihenfolge zurückgegeben. Die Angebotscodes und Namen der Angebotsliste werden basierend auf einer genauen Übereinstimmung mit Angeboten überprüft.

## Parameter

*userCredential*: der Clientbenutzer-Berechtigungsnaehweis.

*requestedLocale*: optionale Ländereinstellung zur Verwendung bei dieser Anforderung.

*partitionName*: optionaler Name der zu verwendenden Kampagnenpartition.

*codeOrNames*: ein Array mit allen zu überprüfenden Angebotscodes oder Namen von Angebotslisten.

**Anmerkung:** Durch diese Methode werden keine Ausnahmen ausgelöst; anstelle einer Überprüfung werden Informationen zu allen angegebenen Codes oder Namen zurückgegeben.

## Rückgabe

Eine typisierte Liste mit null oder mehr Datenwrapperinstanzen von *OfferValidationInfo*.

## Fehler

Keine.

## 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
```

Aktualisiert Details zur dynamischen und statischen offerlist.

## Parameter

*userCredential*: der Clientbenutzer-Berechtigungsnaehweis.

*partitionName*: optionaler Name der zu verwendenden Kampagnenpartition.

*requestedLocale*: optionale Ländereinstellung zur Verwendung bei dieser Anforderung.

*listReference*: Referenz auf Angebotsliste.

*isSmartList*: Flag, das angibt, ob die Liste dynamisch oder statisch ist.

*name*: Name des zu erstellenden Ordners.

*description*: Beschreibung des neuen Ordners.

*listMembers*: verweist auf Angebote, die in der Angebotsliste enthalten sein sollen.

*offerListInfo*: Objekt vom Typ WSSmartListInfo.

*creatorObjectId*: (optional) wird im Plan (IBM Marketing Operations) verwendet, um einen Ordner in IBM Campaign mit einem Ordner in Marketing Operations zu verbinden.

*clearExisting*: Flag, das angibt, ob vorhandene Informationen gelöscht werden müssen. Ist "true" festgelegt, werden vorhandene Listeneinträge vor dem Hinzufügen neuer Einträge gelöscht. Ist "false" festgelegt, werden neue Mitglieder an vorhandene Einträge angehängt.

## Rückgabe

Gibt ein Objekt vom Typ WSEditOfferListResponse zurück.

## Fehler

Löst CampaignServicesException aus, wenn die ID von parentFolder ungültig ist, offerListInfo nicht angegeben ist oder listMembers ungültig sind.

Löst CampaignServicesException aus, wenn der Listenname doppelt vorhanden ist.

## createOffer

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

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

Erstellen Sie ein neues Angebot für den Client und wenden Sie dabei die angegebenen Attribute an.

### Parameter

*authorizationLoginName*: Benutzername des Benutzers, der das Angebot erstellt. Benutzern muss die Berechtigung zum Hinzufügen von Angeboten erteilt werden, damit sie diese Methode verwenden können.

*partitionName*: optionaler Name der zu verwendenden Kampagnenpartition.

*requestedLocale*: optionale Ländereinstellung zur Verwendung bei dieser Anforderung.

*securityPolicyName*: optionaler Name der Sicherheitsrichtlinie für die Kampagne, die zum Erstellen des Angebots verwendet wird. Bei allen nachfolgenden Vorgängen in diesem Angebot wird diese Richtlinie verwendet. Ist keine Sicherheitsrichtlinie definiert, wird die *globale* Richtlinie verwendet.

*name*: der Name, der der neuen Angebotsinstanz (dem zugehörigen Attribut *uacName*) zugeordnet wird.

*folderID*: die ID des Angebotsordners, in dem das Angebot erstellt wird. Diese ID wird auf Richtigkeit überprüft. Wenn sie ungültig ist, wird eine Ausnahme ausgelöst.

*templateName*: erforderlicher (eindeutiger) Name einer vorhandenen Angebotsvorlage, der für das neue Angebot verwendet werden sollte.

*wsAttributes*: ein Array mit Initialisierungsattributen; die Standardwerte des Angebots werden durch die angegebenen Attribute überschrieben; andere bleiben unberührt. Beispiel: Wenn das Attribut *uacOfferCode* angegeben wird, wird es anstelle des automatisch generierten Attributs verwendet. Es ist Aufgabe des Clients, die für das Angebot erforderlichen Attribute, den Typ usw. zu ermitteln.

Löst *CampaignServicesException* aus, wenn eine der folgenden Bedingungen auftritt:

- Der Parameter "folderID" ist ungültig (nicht vorhanden oder nicht vom Typ "Angebot").
- Der Benutzer ist nicht berechtigt, diesen Vorgang durchzuführen.
- In "wsAttributes" werden ungültige Attribute angegeben.
- Es treten andere Laufzeitausnahmebedingungen auf.

## Rückgabe

Eine Einzelinstanz von *OfferInfo* für das erstellte Angebot.

## Fehler

CampaignServicesException

## retireOffers

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

Zieht mindestens ein bestehendes Angebot zurück.

## Parameter

*userCredential*: der Clientbenutzer-Berechtigungsnauchweis.

*requestedLocale*: optionale Ländereinstellung zur Verwendung bei dieser Anforderung.

*partitionName*: optionaler Name der zu verwendenden Kampagnenpartition.

*references*: ein Array mit Referenzen der Angebote, die zurückgezogen werden sollen. *InvalidComponentException* wird ausgelöst, wenn bei einer bestimmten Referenz ein Problem auftritt oder wenn ein Angebot nicht vorhanden ist.

## Rückgabe

Keine.

## Fehler

InvalidComponentException

AuthorizationException, DataException

## deleteOffers

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

Löscht mindestens ein vorhandenes Angebot.

## Parameter

*userCredential*: der Clientbenutzer-Berechtigungsnauchweis.

*requestedLocale*: optionale Ländereinstellung zur Verwendung bei dieser Anforderung.

*partitionName*: optionaler Name der zu verwendenden Kampagnenpartition.

*reference*: ein Array mit Referenzen auf die zu löschenden Angebote. Wenn bei einer angegebenen Referenz ein Problem auftritt oder ein Angebot nicht vorhanden ist, wird *InvalidComponentException* ausgelöst.

## Rückgabe

Keine.

## Fehler

InvalidComponentException

AuthorizationException, DataException

## deleteOffersAndLists

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

Löscht die angegebenen Angebote und Listen.

### Parameter

*userCredential*: der Clientbenutzer-Berechtigungsnaehweis.

*partitionName*: optionaler Name der zu verwendenden Kampagnenpartition.

*requestedLocale*: optionale Ländereinstellung zur Verwendung bei dieser Anforderung.

*offers*: Array mit Referenzen auf ein Angebot oder eine Angebotsliste.

### Rückgabe

Gibt ein Objekt vom Typ WSGetOfferListMembersResponse zurück.

## Fehler

Löst CampaignServicesException aus, wenn die Angebots-ID oder die Angebotslisten-ID ungültig ist.

## listOfferTemplates

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

Listet alle Angebotsvorlagen auf, für die der Benutzer Berechtigungen zum Anzeigen hat.

Nachdem sie abgerufen wurde, kann jede zurückgegebene Instanz von *WSOfferTemplateInfo* unverändert verwendet werden. Alternativ kann mindestens eine Attribut-API verwendet werden, um eine beliebige aufgelistete Vorlage abzurufen oder zu aktualisieren.

### Parameter

*userCredential*: der Clientbenutzer-Berechtigungsnaehweis.

*requestedLocale*: optionale Ländereinstellung zur Verwendung bei dieser Anforderung.

*partitionName*: optionaler Name der zu verwendenden Kampagnenpartition.



## Rückgabe

Eine typisierte Liste mit null oder mehr Datenwrapperinstanzen von *WSOfferTemplateInfo*, eine für jede zurückgegebene Vorlage.

## Fehler

*InvalidExecutionContextException*, *AuthorizationException*

*DataException*

## createTemplate

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

Erstellt eine neue Angebotsvorlage.

## Parameter

*userCredential*: der Clientbenutzer-Berechtigungsnauchweis.

*partitionName*: optionaler Name der zu verwendenden Kampagnenpartition.

*requestedLocale*: optionale Ländereinstellung zur Verwendung bei dieser Anforderung.

*name*: Name der neu erstellten Angebotsvorlage.

*securityPolicyName*: Name der zu verwendenden Sicherheitsrichtlinie.

*wsStaticAttributes*: Liste der statischen Attribute in der Vorlage.

*wsHiddenAttributes*: Liste der ausgeblendeten Attribute in der Vorlage.

*wsParametricAttributes*: Liste der parametrischen Attribute in der Vorlage.

## Rückgabe

Gibt ein Objekt vom Typ *WSCreateTemplateResponse* zurück.

## Fehler

Löst *CampaignServicesException* aus, wenn die Angebotsvorlage nicht vorhanden ist, die Referenz ungültig ist oder keine Referenzen angegeben wurden.

## getOfferTemplate

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

Listet Details zu Angebotsvorlagen entsprechend der Angabe in Referenzen auf.

## Parameter

*userCredential*: der Clientbenutzer-Berechtigungsnaehweis.

*partitionName*: optionaler Name der zu verwendenden Kampagnenpartition.

*requestedLocale*: optionale Ländereinstellung zur Verwendung bei dieser Anforderung.

*wsCampaignReference*: Referenz der übergeordneten Kampagne.

## Rückgabe

Gibt ein Objekt vom Typ `WSGetOfferTemplateResponse` zurück.

## Fehler

Löst `CampaignServicesException` aus, wenn die Angebotsvorlage nicht vorhanden ist, die Referenz ungültig ist oder keine Referenzen angegeben wurden.

## retireOfferTemplates

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

Zieht mindestens eine angegebene Angebotsvorlage zurück.

## Parameter

*userCredential*: der Clientbenutzer-Berechtigungsnaehweis.

*partitionName*: optionaler Name der zu verwendenden Kampagnenpartition.

*requestedLocale*: optionale Ländereinstellung zur Verwendung bei dieser Anforderung.

*wsCampaignReference*: Referenz der übergeordneten Kampagne.

## Rückgabe

Gibt ein Objekt vom Typ `WSRetireOfferTemplatesResponse` zurück.

## Fehler

Löst `CampaignServicesException` aus, wenn die Angebotsvorlage nicht vorhanden ist, die Referenz ungültig ist oder keine Referenzen angegeben wurden.

## getOffersAndListsByPage

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

oder Angebotslisten nach Seite aus.

## Parameter

*userCredential*: der Clientbenutzer-Berechtigungsnachweis.

*partitionName*: optionaler Name der zu verwendenden Kampagnenpartition.

*requestedLocale*: optionale Ländereinstellung zur Verwendung bei dieser Anforderung.

*type*: Typ, der angibt, ob Angebote oder Angebotslisten angefordert wurden.

*pageSize*: die maximale Anzahl der angepassten Komponenten, die für die Seite zurückgegeben werden sollen.

*pageOffset*: das Startoffset aller möglichen Komponenten für den Beginn der Aufzählung (mit dem Wert null). Beispiel: Wenn bei der Aufzählung 1000 Übereinstimmungen mit Angeboten gefunden werden und dieser Wert auf 10 festgelegt ist, würde auf der Seite mit der 11. Komponente begonnen werden. Wenn sich das angegebene Offset außerhalb des gültigen Bereichs befindet, wird eine RangeException ausgelöst.

## Rückgabe

Gibt ein Objekt vom Typ `WSGetOffersAndListsByPageResponse` zurück.

## Fehler

Keine.

## bulkCreateOffers

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

Erstellt Massenangebote mit den Attributen für die einzelnen Angebote, die im Parameter *Angebote* angegeben sind. Alle Angebote werden unter der angegebenen *folderID* mit dem angegebenen *templateName* erstellt.

## Parameter

*authorizationLoginName*: der Clientbenutzer-Berechtigungsnachweis.

*partitionName*: optionaler Name der zu verwendenden Kampagnenpartition.

*requestedLocale*: optionale Ländereinstellung zur Verwendung bei dieser Anforderung.

*securityPolicyName*: optionaler Name der Sicherheitsrichtlinie für die Kampagne, die zum Erstellen des Angebots verwendet wird. Ist keine Sicherheitsrichtlinie definiert, wird die globale Richtlinie verwendet.

*templateName*: Name der im System vorhandenen Angebotsvorlage. Alle Angebote werden mit dieser Vorlage erstellt.

*folderID*: die ID des Angebotsordners, in dem die Angebote erstellt werden. Diese ID wird validiert. Wenn sie ungültig ist, wird eine Ausnahme ausgelöst.

*offers*: ein Array mit Objekten vom Typ *WSBulkOfferInfo*, das den Angebotsnamen und Angebotsattribute definiert. Weitere Einzelheiten finden Sie beim Datentyp *WSBulkOfferInfo*.

### **Rückgabe**

Ein Array mit Instanzen vom Typ *WSOfferInfoStatus* bei jedem Angebot. Enthält den Status und Angebotsinformationen. Der Status zeigt an, ob die Angebotserstellung erfolgreich oder nicht erfolgreich war.

### **Fehler**

*CampaignServicesException*

## **getOfferListDetails**

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

Listet Details zur angegebenen Angebotsliste auf.

### **Parameter**

*userCredential*: der Clientbenutzer-Berechtigungsnachweis.

*partitionName*: optionaler Name der zu verwendenden Kampagnenpartition.

*requestedLocale*: optionale Ländereinstellung zur Verwendung bei dieser Anforderung.

*listReference*: Referenz auf Angebotsliste.

### **Rückgabe**

Gibt ein Objekt vom Typ *WSGetOfferListDetailsResponse* zurück.

### **Fehler**

Löst *CampaignServicesException* aus, wenn Listenreferenzen ungültig sind.

## **getOfferListMembers**

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

Listet Angebotsinformationen in der angegebenen Angebotsliste auf.

### **Parameter**

*userCredential*: der Clientbenutzer-Berechtigungsnachweis.

*partitionName*: optionaler Name der zu verwendenden Kampagnenpartition.

*requestedLocale*: optionale Ländereinstellung zur Verwendung bei dieser Anforderung.

*listReference*: Referenz auf Angebotsliste.

## Rückgabe

Gibt ein Objekt vom Typ `WSDeleteOffersAndListsResponse` zurück.

## Fehler

Löst `CampaignServicesException` aus, wenn die ID von offerlist ungültig ist.

## getOffersByQuery

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

Mit bereitgestellten Angeboten übereinstimmende Listenangebote.

## Parameter

*userCredential*: der Clientbenutzer-Berechtigungsnachweis.

*partitionName*: optionaler Name der zu verwendenden Kampagnenpartition.

*requestedLocale*: optionale Ländereinstellung zur Verwendung bei dieser Anforderung.

*query*: Abfrage für die Suche nach Angeboten. Das Format der Abfrage ist mit dem in der dynamischen offerlist verwendeten Format identisch.

*maxSize*: Maximale Anzahl der aufzulistenden Datensätze.

*includeSubFolder*: Flag, das angibt, ob ein Unterordner bei der Suche mit einbezogen werden muss.

*scopeFolders*: Liste der Ordnerreferenzen zum Durchsuchen bei Angeboten.

## Rückgabe

Gibt ein Objekt vom Typ `WSGetOffersByQueryResponse` zurück.

## Fehler

Löst `CampaignServicesException` aus, wenn Ordnerreferenzen ungültig sind.

## retireOfferLists

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

Zieht mindestens eine angegebene Angebotsliste zurück.

## Parameter

*userCredential*: der Clientbenutzer-Berechtigungsnaehweis.

*partitionName*: optionaler Name der zu verwendenden Kampagnenpartition.

*requestedLocale*: optionale Ländereinstellung zur Verwendung bei dieser Anforderung.

*wsReferences*: Referenz auf Angebotslisten.

## Rückgabe

Keine.

## Fehler

Löst CampaignServicesException aus, wenn Listenreferenzen ungültig sind.

## 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
```

Erstellt einen neuen Ordner vom Typ campaign/offer/sessions/segments.

## Parameter

*userCredential*: der Clientbenutzer-Berechtigungsnaehweis.

*partitionName*: optionaler Name der zu verwendenden Kampagnenpartition.

*requestedLocale*: optionale Ländereinstellung zur Verwendung bei dieser Anforderung.

*name*: Name des zu erstellenden Ordners.

*description*: Beschreibung des neuen Ordners.

*securityPolicyName*: Name der zu verwendenden Sicherheitsrichtlinie.

*parentFolderId*: (optional) ID des übergeordneten Ordners. Ist keine ID angegeben, wird der Ordner im Stammverzeichnis erstellt.

*folderType*: Typ von folder-Offer/session/campaign/segment.

*createdBy*: (optional) Objekt vom Typ WSApplicationTypeEnum gibt an, mit welcher Anwendung der Ordner erstellt wurde. Die möglichen Werte lauten "Campaign", "Plan", "Collaborate" und "eMessage". Ist kein Wert angegeben, wird Campaign verwendet.

**Anmerkung:** Plan = IBM Marketing Operations. Collaborate = IBM Distributed Marketing.

*creatorObjectId*: (optional) wird im Plan (IBM Marketing Operations) verwendet, um einen Ordner in IBM Campaign mit einem Ordner in Marketing Operations zu verbinden.

## Rückgabe

Gibt ein Objekt vom Typ `WSCreateFolderResponse` zurück.

## Fehler

Löst `CampaignServicesException` aus, wenn der Ordner typ ungültig ist oder doppelt vorliegt.

## editFolder

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

Aktualisiert den angegebenen Ordner.

## Parameter

*userCredential*: der Clientbenutzer-Berechtigungs nachweis.

*partitionName*: optionaler Name der zu verwendenden Kampagnenpartition.

*requestedLocale*: optionale Ländereinstellung zur Verwendung bei dieser Anforderung.

*id*: ID des zu aktualisierenden Ordners.

*name*: Name des zu erstellenden Ordners.

*description*: Beschreibung des neuen Ordners.

*folderType*: Typ von folder-Offer/session/campaign/segment.

*creatorObjectId*: (optional) wird im Plan (IBM Marketing Operations) verwendet, um einen Ordner in IBM Campaign mit einem Ordner in Marketing Operations zu verbinden.

*clearExisting*: wird jetzt nicht verwendet.

## Rückgabe

Gibt ein Objekt vom Typ `WSEditFolderResponse` zurück.

## Fehler

Löst `CampaignServicesException` aus, wenn der Ordner typ ungültig ist oder doppelt vorliegt oder wenn die Ordner-ID ungültig ist.

## getSubFoldersList

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

Listet alle Unterordner des angegebenen Ordners auf.

### Parameter

*userCredential*: der Clientbenutzer-Berechtigungsnachweis.

*partitionName*: optionaler Name der zu verwendenden Kampagnenpartition.

*requestedLocale*: optionale Ländereinstellung zur Verwendung bei dieser Anforderung.

*parentFolder*: Referenz auf den Ordner, für den alle Unterordner angefordert werden.

*folderType*: Ordnerart.

### Rückgabe

Gibt ein Objekt vom Typ `WSGetSubFolderListResponse` zurück.

### Fehler

Löst `CampaignServicesException` aus, wenn eine Ordnerreferenz ungültig ist.

## moveFolders

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

Verschiebt einen angegebenen Ordner in einen anderen übergeordneten Ordner.

### Parameter

*userCredential*: der Clientbenutzer-Berechtigungsnachweis.

*partitionName*: optionaler Name der zu verwendenden Kampagnenpartition.

*requestedLocale*: optionale Ländereinstellung zur Verwendung bei dieser Anforderung.

*folderId*: zu löschende Ordner-ID.

*parentFolder*: ID des übergeordneten Ordners.

*destinationFolder*: ID des Zielordners, in den der angegebene Ordner verschoben wird.

*folderType*: Typ von folder-Offer/session/campaign/segment.



## Rückgabe

Gibt ein Objekt vom Typ `WSMoveFolderResponse` zurück.

## Fehler

Löst `CampaignServicesException` aus, wenn der Ordnerartyp oder die ID des übergeordneten Ordners ungültig ist oder wenn die ID des Ordners ungültig ist.

## deleteFolders

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

Löscht angegebene Ordner und alle Elemente in diesen Ordnern aus dem System.

## Parameter

*userCredential*: der Clientbenutzer-Berechtigungsnauchweis.

*partitionName*: optionaler Name der zu verwendenden Kampagnenpartition.

*requestedLocale*: optionale Ländereinstellung zur Verwendung bei dieser Anforderung.

*folderId*: zu löschende Ordner-ID.

*parentFolder*: ID des übergeordneten Ordners.

*deleteChilds*: Flag, das angibt, ob alle Abhängigkeiten des Ordners gelöscht werden müssen. Bei "false" wird keine Abhängigkeit gelöscht.

*folderType*: Typ von folder-Offer/session/campaign/segment.

## Rückgabe

Gibt ein Objekt vom Typ `WSDeleteFolderResponse` zurück.

## Fehler

Löst `CampaignServicesException` aus, wenn der Ordnerartyp oder die Ordner-ID ungültig ist.



---

## Kapitel 5. SOAP-API - Allgemeine Ausnahmen

Die SOAP-API von IBM Campaign löst unter Umständen die folgenden allgemeinen Ausnahmen aus. Alle lokalisierten Nachrichten in der Ausnahme weisen, sofern verfügbar, die angeforderte Ländereinstellung für IBM Campaign auf. Es gelten die gewöhnlichen Fallback-Richtlinien für die Java-Ländereinstellung.

### **RemoteException**

Dieses Element gilt nur für die SOAP-Schnittstelle.

Alle SOAP-Aufrufe in der API können die RemoteException auslösen, wenn ein Fehler auf Systemebene auftritt, wie z. B. ein Problem auf der Prozessebene der SOAP-Rahmenanweisung (Axis). Eine Einschränkung, die im Web-Service WSDL definiert ist, wurde aus einem bestimmten Grund nicht eingehalten.

Geprüfte und ungeprüfte API-Standardausnahmen, wie z. B. DataException, werden als Fehlerstatus zurückgegeben, nicht als eine RemoteException.

Weitere Angaben enthält der Abschnitt zur SOAP-Schnittstelle.

### **AuthenticationException**

Der Benutzer konnte für die angegebene Campaign-Partition nicht authentifiziert werden. Prüfen Sie die Benutzerrolle, die in IBM Marketing Platform definiert ist.

### **AuthorizationException**

Der Benutzer ist nicht berechtigt, den angeforderten Vorgang durchzuführen. Diese Ausnahme kann durch eine beliebige API-Methode ausgelöst werden, daher ist sie nicht deklariert (inaktiviert). Prüfen Sie die Berechtigungen, die der Benutzerrolle in IBM Marketing Platform zugeordnet sind.

### **DataException**

In der zugrunde liegenden Datenbankschicht in Campaign ist eine schwerwiegende Ausnahme aufgetreten (inaktiviert).

Prüfen Sie das IBM Campaign-Ablaufdiagramm und die -Listenerprotokolle auf Einzelheiten.

### **LockException**

Eine temporäre Ausnahme, die ausgelöst wird, wenn der Client versucht, eine Komponente, z. B. ein Ablaufdiagramm, zu aktualisieren, während sie von einem anderen Benutzer bearbeitet wird. Von dieser Ausnahme kann man sich in der Regel erholen, indem gewartet wird und der Vorgang anschließend wiederholt wird. Für die Wiederholungslogik ist der Client verantwortlich.

### **InvalidComponentException**

Es wurde versucht, auf eine ungültige oder unbekannt Komponente (Kampagne, Ablaufdiagramm, Zielzelle) zu verweisen. Die Methode `getComponentReference()`

der Ausnahme gibt die auslösende Komponentenreferenz zurück.

### **InvalidAttributeException**

Eine Ausnahme, die ausgelöst wird, wenn der Client ein ungültiges Attribut bereitstellt oder referenziert, so als würde der falsche Datentyp oder ein Array mit Werten verwendet werden, in dem keiner davon zulässig sind. Die Methode `getAttributeName()` der Ausnahme gibt den Namen des Problemattributs zurück, `getAttributeValue()` gibt den Wert zurück und `getComponentReference()` identifiziert die Komponente (bzw. den Massenindex).

### **AttributeExistsException**

Wird ausgelöst, wenn der Client versucht, die Metadaten eines doppelten Attributs für eine Komponente zu definieren. Die Methode `getAttributeName()` der Ausnahme gibt den Namen des doppelten Attributs zurück; `getComponentReference()` identifiziert die Komponente (bzw. den Massenindex).

### **AttributeNotFoundException**

Wird immer dann ausgelöst, wenn der Client versucht, auf ein unbekanntes Attribut (Kampagne, Ablaufdiagramm, Zielzelle usw.) zu verweisen. Die Methode `getAttributeName()` der Ausnahme gibt den Namen des Attributs ohne Entsprechung zurück; `getComponentReference()` identifiziert die Komponente (bzw. den Massenindex).

### **CompositeException**

Eine `CompositeException` wird von einigen APIs verwendet, um dem Anrufer mehrere Fehler zurückzumelden. Üblicherweise sind mehrere Ursachen damit verbunden; alle Ursachen werden in einer Liste in der Reihenfolge erfasst, in der sie aufgetreten sind. Die Methode `getCauseList()` der Ausnahme gibt diese Liste zurück, die dann weiter überprüft werden kann, um Einzelheiten zu jedem Fehler zu erhalten.

**Anmerkung:** Bevor sie eine `CompositeException` ausgibt, schließt die API ihre Arbeit entweder erfolgreich ab oder macht ihre Arbeit rückgängig. Siehe hierzu z. B. die Massen-APIs für das Arbeitsblatt für Zielzellen im Abschnitt „SOAP-API-Methoden: Zielzellen“ auf Seite 34.

---

## Bevor Sie sich an den technischen Support von IBM wenden

Sollte sich ein Problem nicht mithilfe der Dokumentation beheben lassen, können sich die für den Support zuständigen Kontaktpersonen Ihres Unternehmens telefonisch an den technischen Support von IBM wenden. Verwenden Sie diese Anleitungen, damit das Problem effizient und erfolgreich gelöst wird.

Wenn Sie wissen möchten, wer die für den Support zuständige Kontaktperson Ihres Unternehmens ist, wenden Sie sich an Ihren IBM Administrator.

**Anmerkung:** Der technische Support schreibt bzw. erstellt keine API-Scripts. Wenden Sie sich zur Unterstützung bei der Implementierung unserer API-Angebote an IBM Professional Services.

### Zusammenzustellende Informationen

Halten Sie folgende Informationen bereit, wenn Sie sich an den technischen Support von IBM wenden:

- Kurze Beschreibung der Art Ihres Problems
- Detaillierte Fehlermeldungen, die beim Auftreten des Problems angezeigt werden.
- Schritte zum Reproduzieren des Problems
- Entsprechende Protokolldateien, Sitzungsdateien, Konfigurationsdateien und Daten
- Informationen zu Ihrer -Produkt- und -Systemumgebung, die Sie entsprechend der Beschreibung unter "Systeminformationen" abrufen können.

### Systeminformationen

Bei Ihrem Anruf beim technischen Support von IBM werden Sie um verschiedene Informationen gebeten.

Sofern das Problem Sie nicht an der Anmeldung hindert, finden Sie einen Großteil der benötigten Daten auf der Info-Seite. Dort erhalten Sie Informationen zur installierten IBM Anwendung.

Sie können über **Hilfe > Info** (Help > About) auf die Info-Seite zugreifen. Wenn Sie nicht auf die Info-Seite zugreifen können, prüfen Sie, ob im Installationsverzeichnis Ihrer Anwendung die Datei `version.txt` vorhanden ist.

### Kontaktinformationen für den technischen Support von IBM

Wenn Sie sich an den technischen Support von IBM wenden möchten, finden Sie weitere Informationen auf der Website des technischen Supports für IBM Produkte ([http://www.ibm.com/support/entry/portal/open\\_service\\_request](http://www.ibm.com/support/entry/portal/open_service_request)).

**Anmerkung:** Um eine Supportanforderung einzugeben, müssen Sie sich mit einem IBM Account anmelden. Dieser Account muss mit Ihrer IBM Kundennummer verknüpft sein. Weitere Informationen zum Zuordnen Ihres Accounts zu Ihrer IBM Kundennummer erhalten Sie unter **Unterstützungsressourcen > Gültige Softwareunterstützung** im Support Portal.



---

## Bemerkungen

Die vorliegenden Informationen wurden für Produkte und Services entwickelt, die auf dem deutschen Markt angeboten werden.

Möglicherweise bietet IBM die in dieser Dokumentation beschriebenen Produkte, Services oder Funktionen in anderen Ländern nicht an. Informationen über die gegenwärtig im jeweiligen Land verfügbaren Produkte und Services sind beim zuständigen IBM Ansprechpartner erhältlich. Hinweise auf IBM Lizenzprogramme oder andere IBM Produkte bedeuten nicht, dass nur Programme, Produkte oder Services von IBM verwendet werden können. Anstelle der IBM Produkte, Programme oder Services können auch andere, ihnen äquivalente Produkte, Programme oder Services verwendet werden, solange diese keine gewerblichen oder anderen Schutzrechte von IBM verletzen. Die Verantwortung für den Betrieb von Produkten, Programmen und Services anderer Anbieter liegt beim Kunden.

Für die in diesem Handbuch beschriebenen Erzeugnisse und Verfahren kann es IBM Patente oder Patentanmeldungen geben. Mit der Auslieferung dieses Handbuchs ist keine Lizenzierung dieser Patente verbunden. Lizenzanforderungen sind schriftlich an folgende Adresse zu richten (Anfragen an diese Adresse müssen auf Englisch formuliert werden):

IBM Director of Licensing  
IBM Europe, Middle East + Africa  
Tour Descartes  
2, avenue Gambetta  
92066 Paris La Defense  
France

Trotz sorgfältiger Bearbeitung können technische Ungenauigkeiten oder Druckfehler in dieser Veröffentlichung nicht ausgeschlossen werden. Die hier enthaltenen Informationen werden in regelmäßigen Zeitabständen aktualisiert und als Neuausgabe veröffentlicht. IBM kann ohne weitere Mitteilung jederzeit Verbesserungen und/oder Änderungen an den in dieser Veröffentlichung beschriebenen Produkten und/oder Programmen vornehmen.

Verweise in diesen Informationen auf Websites anderer Anbieter werden lediglich als Service für den Kunden bereitgestellt und stellen keinerlei Billigung des Inhalts dieser Websites dar. Das über diese Websites verfügbare Material ist nicht Bestandteil des Materials für dieses IBM Produkt. Die Verwendung dieser Websites geschieht auf eigene Verantwortung.

Werden an IBM Informationen eingesandt, können diese beliebig verwendet werden, ohne dass eine Verpflichtung gegenüber dem Einsender entsteht.

Lizenznehmer des Programms, die Informationen zu diesem Produkt wünschen mit der Zielsetzung: (i) den Austausch von Informationen zwischen unabhängig voneinander erstellten Programmen und anderen Programmen (einschließlich des vorliegenden Programms) sowie (ii) die gemeinsame Nutzung der ausgetauschten Informationen zu ermöglichen, wenden sich an folgende Adresse:

IBM Corporation  
B1WA LKG1

550 King Street  
Littleton, MA 01460-1250  
USA

Die Bereitstellung dieser Informationen kann unter Umständen von bestimmten Bedingungen - in einigen Fällen auch von der Zahlung einer Gebühr - abhängig sein.

Die Lieferung des in diesem Dokument beschriebenen Lizenzprogramms sowie des zugehörigen Lizenzmaterials erfolgt auf der Basis der IBM Rahmenvereinbarung bzw. der Allgemeinen Geschäftsbedingungen von IBM, der IBM Internationalen Nutzungsbedingungen für Programmpakete oder einer äquivalenten Vereinbarung.

Alle in diesem Dokument enthaltenen Leistungsdaten stammen aus einer kontrollierten Umgebung. Die Ergebnisse, die in anderen Betriebsumgebungen erzielt werden, können daher erheblich von den hier erzielten Ergebnissen abweichen. Einige Daten stammen möglicherweise von Systemen, deren Entwicklung noch nicht abgeschlossen ist. Eine Gewährleistung, dass diese Daten auch in allgemein verfügbaren Systemen erzielt werden, kann nicht gegeben werden. Darüber hinaus wurden einige Daten unter Umständen durch Extrapolation berechnet. Die tatsächlichen Ergebnisse können davon abweichen. Benutzer dieses Dokuments sollten die entsprechenden Daten in ihrer spezifischen Umgebung prüfen.

Alle Informationen zu Produkten anderer Anbieter stammen von den Anbietern der aufgeführten Produkte, deren veröffentlichten Ankündigungen oder anderen allgemein verfügbaren Quellen. IBM hat diese Produkte nicht getestet und kann daher keine Aussagen zu Leistung, Kompatibilität oder anderen Merkmalen machen. Fragen zu den Leistungsmerkmalen von Produkten anderer Anbieter sind an den jeweiligen Anbieter zu richten.

Aussagen über Pläne und Absichten von IBM unterliegen Änderungen oder können zurückgenommen werden und repräsentieren nur die Ziele von IBM.

Alle von IBM angegebenen Preise sind empfohlene Richtpreise und können jederzeit ohne weitere Mitteilung geändert werden. Händlerpreise können unter Umständen von den hier genannten Preisen abweichen.

Diese Veröffentlichung enthält Beispiele für Daten und Berichte des alltäglichen Geschäftsablaufs. Sie sollen nur die Funktionen des Lizenzprogramms illustrieren und können Namen von Personen, Firmen, Marken oder Produkten enthalten. Alle diese Namen sind frei erfunden; Ähnlichkeiten mit tatsächlichen Namen und Adressen sind rein zufällig.

#### COPYRIGHTLIZENZ:

Diese Veröffentlichung enthält Beispielanwendungsprogramme, die in Quellsprache geschrieben sind und Programmier Techniken in verschiedenen Betriebsumgebungen veranschaulichen. Sie dürfen diese Beispielprogramme kostenlos kopieren, ändern und verteilen, wenn dies zu dem Zweck geschieht, Anwendungsprogramme zu entwickeln, zu verwenden, zu vermarkten oder zu verteilen, die mit der Anwendungsprogrammierschnittstelle für die Betriebsumgebung konform sind, für die diese Beispielprogramme geschrieben werden. Diese Beispiele wurden nicht unter allen denkbaren Bedingungen getestet. Daher kann IBM die Zuverlässigkeit, Wartungsfreundlichkeit oder Funktion dieser Programme weder zusagen noch gewährleisten. Die Beispielprogramme werden ohne Wartung (auf "as-is"-Basis)



und ohne jegliche Gewährleistung zur Verfügung gestellt. IBM übernimmt keine Haftung für Schäden, die durch die Verwendung der Beispielpprogramme entstehen.

---

## Marken

IBM, das IBM Logo und ibm.com sind Marken oder eingetragene Marken der IBM Corporation in den USA und/oder anderen Ländern. Weitere Produkt- und Servicennamen können Marken von IBM oder anderen Unternehmen sein. Eine aktuelle Liste der IBM Marken finden Sie auf der Webseite "Copyright and trademark information" unter [www.ibm.com/legal/copytrade.shtml](http://www.ibm.com/legal/copytrade.shtml).

---

## Hinweise zu Datenschutzrichtlinien und Nutzungsbedingungen

IBM Softwareprodukte, einschließlich Software as a Service-Lösungen ("Softwareangebote"), können Cookies oder andere Technologien verwenden, um Informationen zur Produktnutzung zu erfassen, die Endbenutzererfahrung zu verbessern und Interaktionen mit dem Endbenutzer anzupassen oder zu anderen Zwecken. Ein Cookie ist ein Datenelement, das von einer Website an Ihren Browser gesendet wird und dann als Tag auf Ihrem Computer gespeichert werden kann, mit dem Ihr Computer identifiziert wird. In vielen Fällen werden von diesen Cookies keine personenbezogenen Daten erfasst. Wenn ein Softwareangebot, das von Ihnen verwendet wird, die Erfassung personenbezogener Daten anhand von Cookies und ähnlichen Technologien ermöglicht, werden Sie im Folgenden über die hierbei geltenden Besonderheiten informiert.

Abhängig von den bereitgestellten Konfigurationen kann dieses Softwareangebot Sitzungscookies und permanente Cookies verwenden, mit denen der Benutzername des Benutzers und andere personenbezogene Daten zum Zwecke des Sitzungsmanagements, zur Verbesserung der Benutzerfreundlichkeit und zu anderen funktionsbezogenen Zwecken sowie zur Nutzungsüberwachung erfasst werden. Diese Cookies können deaktiviert werden. Durch die Deaktivierung kann jedoch auch die von ihnen bereitgestellte Funktionalität nicht mehr genutzt werden.

Die Erfassung personenbezogener Daten mithilfe von Cookies und ähnlichen Technologien wird durch verschiedene rechtliche Bestimmungen geregelt. Wenn die für dieses Softwareangebot implementierten Konfigurationen Ihnen als Kunde die Möglichkeit bieten, personenbezogene Daten von Endbenutzern über Cookies und andere Technologien zu erfassen, dann sollten Sie ggf. juristische Beratung zu den geltenden Gesetzen für eine solche Datenerfassung in Anspruch nehmen. Dies gilt auch in Bezug auf die Anforderungen, die vom Gesetzgeber in Bezug auf Hinweise und die Einholung von Einwilligungen vorgeschrieben werden.

IBM setzt voraus, dass Kunden folgende Bedingungen erfüllen: (1) Sie stellen einen klar erkennbaren und auffälligen Link zu den Nutzungsbedingungen der Kundenwebsite (z. B. Datenschutzerklärung) bereit. Dieser Link muss wiederum einen Link zu der Vorgehensweise von IBM und des Kunden bei der Datenerhebung und Datennutzung umfassen. (2) Sie weisen darauf hin, dass Cookies und Clear GIFs/Web-Bacons von IBM im Auftrag des Kunden auf dem Computer des Besuchers platziert werden. Dieser Hinweis muss eine Erläuterung hinsichtlich des Zwecks dieser Technologie umfassen. (3) Sie müssen in dem gesetzlich vorgeschriebenen Umfang die Einwilligung von Websitebesuchern einholen, bevor Cookies und Clear GIFs/Web-Bacons vom Kunden oder von IBM im Auftrag des Kunden auf den Geräten der Websitebesucher platziert werden.

Weitere Informationen zur Verwendung verschiedener Technologien einschließlich der Verwendung von Cookies zu diesen Zwecken finden Sie im IBM Online Privacy Statement unter der Webadresse <http://www.ibm.com/privacy/details/us/en> im Abschnitt mit dem Titel "Cookies, Web Beacons and Other Technologies".



