

IBM Campaign  
Version 9 Release 0  
15. Januar 2013

*Spezifikation für die Angebots-API*



**Hinweis**

Vor Verwendung dieser Informationen und des darin beschriebenen Produkts sollten die Informationen unter „Bemerkungen“ auf Seite 53 gelesen werden.

Diese Edition bezieht sich auf Version 9, Release 0, Modifikation 0 von IBM Campaign und alle nachfolgenden Releases und Modifikationen, bis dieser Hinweis in einer Neuedition geändert wird.

© Copyright IBM Corporation 1998, 2013.

# Inhaltsverzeichnis

<b>Kapitel 1. Einführung</b>	<b>1</b>
Allgemeine Zusammenfassung der Änderungen	1
Bestehende Implementierungen mit der Client-API	2
Neue Client-API-JAR-Datei	2
Neue abhängige JAR-Dateien	2
Änderungen beim Client-API-Konstruktor	2
Änderungen bei den parametrisierten Konstrukto- ren der unterstützenden Klassen	2
Bestehende Implementierungen unter direkter Ver- wendung der WSDL	3
WSDL-Speicherposition und Service-URL	3
Generieren von Stubs und Klassen	3
Verwenden von generierten Stubs und unterstüt- zenden Klassen	3
Referenzen	5
Anforderungen	5
Entwurfsüberblick	5
Notes	6
<b>Kapitel 2. Datentypen</b>	<b>9</b>
WSReference	9
WSVersion	9
WSServiceInfo	9
WSAttributeTypeEnum	10
WSAttributeStatusEnum	10
WSAccessTypeEnum	10
WSSelectTypeEnum	10
WSRunStatusEnum	10
WSRunTypeEnum	11
WSAttribute	11
WSAttributeMetadata	12
WSCampaignInfo	14
WSComponentOrFolderInfo	14
WSTargetCellInfo	14
WSMetricsInfo	14
WSRunResults	15
WSOfferInfo	15
WSOfferCodeOrName	16
WSOfferValidationInfo	16
WSOfferTemplateInfo	16
WSBulkOfferInfo	16
WSOfferInfoStatus	17
<b>Kapitel 3. Allgemeine Ausnahmerebedin- gungen</b>	<b>19</b>
RemoteException	19
AuthenticationException	19
AuthorizationException	19
DataException	19
LockException	20
InvalidComponentException	20
InvalidAttributeException	20
AttributeNotFoundException	20
AttributeExistsException	20

CompositeException	20
--------------------	----

## **Kapitel 4. CampaignServices-API-Me- thoden**

Servicemethoden	21
getServiceInfo	21
Attribute	21
getAttributesByName	21
updateAttributes	22
getAttributeMetadataByName	23
createAttributeMetadata	24
updateAttributeMetadata	25
deleteAttributeMetadata	27
Kampagnen	27
generateCampaignCode	28
createCampaign	29
listCampaignsByPage	30
Zielzellenmethoden	31
createTargetCell	32
bulkCreateTargetCells	33
listTargetCells	34
bulkUpdateTargetCells	35
getRunResultsByCell	36
bulkDeleteTargetCells	36
Analyse	37
getCampaignMetrics	37
Methoden für Angebot, Angebotsliste und Ange- botsvorlage	37
listOffersAndFolders	38
searchOffersBasic	39
listOffersByPage	40
validateOffers	41
createOffer	42
retireOffers	43
deleteOffers	43
listOfferTemplates	44
bulkCreateOffers	44

## **Kapitel 5. Verwenden der API**

Verwenden der Client-API-JAR-Datei	47
Code von OfferAPI.java	47
Verwenden der WSDL	49
Leistungsaspekte	49
Paketierung	50

## **Kontakt zum technischen Support von IBM**

## **Bemerkungen**

Marken	55
Hinweise zu Datenschutzrichtlinien und Nutzungs- bedingungen	55



---

## Kapitel 1. Einführung

**Anmerkung:** In diesem Dokument wird der Angebotsteil von Version 3.0 der IBM® Campaign-Services-API behandelt, auch als CampaignServices bekannt. Es werden nur die Angebotsservices unterstützt, die in diesem Handbuch beschrieben werden.

**Anmerkung:** Wenn Sie ein Upgrade auf IBM Campaign Version 8.2 oder höher durchführen und zurzeit die IBM Campaign-Services-API implementiert ist, sind aufgrund der Änderungen an der API, die wiederum durch das Upgrade von Axis Version 1.3 auf Axis2 1.4.1 erforderlich waren, auch Änderungen an Ihrem Anwendungscode erforderlich. Einzelheiten hierzu finden Sie unter „Neue abhängige JAR-Dateien“ auf Seite 2.

In diesem Abschnitt werden die folgenden Hauptthemen behandelt:

- „Allgemeine Zusammenfassung der Änderungen“
- „Bestehende Implementierungen mit der Client-API“ auf Seite 2
- „Bestehende Implementierungen unter direkter Verwendung der WSDL“ auf Seite 3
- „Referenzen“ auf Seite 5
- „Anforderungen“ auf Seite 5
- „Entwurfsüberblick“ auf Seite 5

---

### Allgemeine Zusammenfassung der Änderungen

- Migration der SOAP-Engine von der Axis-Version 1.3 auf Axis2 1.4.1.
- Die WSDL wurde umstrukturiert, um Probleme bei der Handhabung erforderlicher/optionaler Parameter bearbeiten zu können.
- Die JAR-Datei der Client-API wurde aufgrund der WSDL-Änderungen geändert, sodass auch die generierten Stubs und Klassen geändert wurden. Die Methodenparameter der Client-API haben sich nicht geändert, aber die Konstruktoren der unterstützenden Wertobjekte wurden aufgrund der Axis2 WSDL2Java-Konvertierung geändert.
- Die Web-Service-URL wurde geändert und verweist nun auf:

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

Die entsprechende WSDL kann abgerufen werden unter:

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

Aufgrund dieser Änderungen müssen Sie Ihren Anwendungscode ändern, wenn Sie aktuell die API verwenden. Abhängig davon, ob Sie die Client-API oder die WSDL verwenden, erhalten Sie in folgenden Abschnitten weitere Informationen:

- „Bestehende Implementierungen mit der Client-API“ auf Seite 2
- „Bestehende Implementierungen unter direkter Verwendung der WSDL“ auf Seite 3

---

## Bestehende Implementierungen mit der Client-API

Wenn Sie die JAR-Datei der Client-API zur Interaktion mit der Campaign-Webanwendung verwenden, erhalten Sie in den folgenden Abschnitten Informationen zu den für Sie relevanten Änderungen.

### Neue Client-API-JAR-Datei

Da Änderungen an der internen Implementierung der Client-API-Klassen vorgenommen wurden, muss Ihre Java™-Anwendung die neue JAR-Datei verwenden, die sich an folgender Position befindet:

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

Ein Java-Beispiel zur Erstellung eines neuen Angebots enthält der Abschnitt „Code von OfferAPI.java“ auf Seite 47. Dasselbe Beispiel befindet sich in Ihrer Campaign-Installation unter:

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

### Neue abhängige JAR-Dateien

Aufgrund des Upgrades auf Axis2 Version 1.4.1 muss auch Ihre Java-Anwendung für die Verwendung der in Axis2 1.4.1 verteilten JAR-Dateien aktualisiert werden, da CampaignServicesClient30.jar von diesen JAR-Dateien abhängt. Alle abhängigen JAR-Dateien müssen im Java-Klassenpfad Ihrer Anwendung enthalten sein. Sie befinden sich in der Datei Campaign.war unter:

```
<CAMPAIGN_HOME>/Campaign.war
```

Extrahieren Sie die JAR-Dateien aus Campaign.war und fügen Sie sie dem Java-Klassenpfad hinzu.

### Änderungen beim Client-API-Konstruktor

Ändern Sie beim Erstellen des Client-API-Objekts die Web-Service-URL und die Ausnahmebedingungssignatur, wie im Beispiel unten gezeigt.

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

### Änderungen bei den parametrisierten Konstruktoren der unterstützenden Klassen

Bei der AXIS2-Engine haben die generierten Klassen und Stubs keine parametrisierten Konstruktoren mehr. Stattdessen haben diese Klassen nur den argumentlosen Standardkonstruktor mit Settern und Gettern für die Elemente.

Daher wurde:

```
WSReference wsRef = new WSReference(WSComponentTypeEnum typeEnum, Long id);
```

geändert in:

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

---

## Bestehende Implementierungen unter direkter Verwendung der WSDL

Die WSDL des Campaign-Web-Service dient zur Erstellung von clientseitigen Stubs und unterstützenden Klassen mithilfe eines beliebigen Konvertierungstools eines Drittherstellers. Wenn Sie die WSDL zur Interaktion mit der Campaign-Webanwendung verwenden, erhalten Sie in den folgenden Abschnitten Informationen zu den für Sie relevanten Änderungen. In den hier bereitgestellten Beispielen wird das WSDL2Java-Tool Axis2 1.4.1 von Apache verwendet.

### WSDL-Speicherposition und Service-URL

Der Campaign-Web-Service für IBM Campaign ist implementiert unter:

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

Die entsprechende WSDL kann hier abgerufen werden:

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

### Generieren von Stubs und Klassen

Zur Generierung der Stubs und unterstützenden Java-Klassen mit der WSDL kann das WSDL2Java-Tool Axis2 1.4.1 von Apache verwendet werden. Eine Beispiel-Ant-Task ist unten aufgeführt.

Das Tool kann auch über die Befehlszeile mit ähnlichen Argumenten verwendet werden. Die Argumentwerte können an Ihre Umgebung angepasst werden.

**Anmerkung:** Im folgenden WSDL2Java-Konvertierungsbeispiel wird die Standard-ADB-Bindung verwendet.

```
<java classname="org.apache.axis2.wsdl.WSDL2Java" fork="true">
  <classpath refid="axis2.class.path"/> <!--Klassenpfad mit
Axis2-Bibliotheken -->
  <arg value="-uri"/>
  <arg file="CampaignServices30.wsdl"/> <!--Tatsächliche Speicherposition der
WSDL -->
  <arg value="-s"/> <!-- Generierung des Synchronisationscodes -->
  <arg value="-Euwc"/> <!-- Zur Erstellung der Wrapper-
Java-Typen für auf null festlegbare = "true"-Elemente. -->
  <arg value="-uw"/> <!-- Key-Wrapping für Parameter aufheben -->
  <arg value="-u"/> <!-- Klassen entpacken -->
  <arg value="-ns2p"/> <!-- Namensbereich zur Zuordnungspaketierung. Kunden
können eigene Paketnamen haben. -->
  <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 unterstützenden Klassen

Der Stub kann folgendermaßen verwendet werden:

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

serviceStub._getServiceClient().getOptions().setTimeOutInMilliseconds
(webServiceTimeout); //Zeitlimit in Millisekunden.
```

Das Angebot kann folgendermaßen erstellt werden:

```
try{
//Passen Sie Host und Port an Ihre Umgebung an.
String serviceURL = "http://host:port/Campaign/services/CampaignServices30Service";
CampaignServices30ServiceStub serviceStub = new CampaignServices30ServiceStub(serviceURL);
long webServiceTimeout = 2*60*1000; // 2 minutes
serviceStub._getServiceClient().getOptions().setTimeOutInMilliseconds(webServiceTimeout); //Zeitlimit in Millisekunden.

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

WSTextAttribute[] wsAttributes = {nameAttribute};
// Konvertieren zu WSAttributeArrays
WSAttributeArrays obj = new WSAttributeArrays();
obj.setTextAttributes(wsAttributes);
//Passen Sie die Werte der folgenden Variablen an Ihre Umgebung an.
String authorizationLoginName = "asm_admin"; //Benutzername zur Anmeldung
String partitionName = "partition1"; //Use your security policy of Campaign
String securityPolicyName = "Global Policy"; //Verwenden Sie Ihre Sicherheitsrichtlinie aus Campaign

String offerName = "1st Offer"; //Name des zu erstellenden Angebots.
String templateName = "Offer Template"; //Name der bestehenden Angebotsvorlage.
long folderID = 100; //Tatsächliche ID des Ordners, in dem das Angebot erstellt wird.
//Bei folderID <=0 wird das Angebot auf Stammebene erstellt.

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);
//Campaign-Web-Service-Aufruf
WSCreateOfferResponse wsResponse = serviceStub.createOffer(createOfferObject);
// Prozessstatus
WSRequestStatus status = wsResponse.getStatus();
// fertig
WSOfferInfo offerInfo = wsResponse.getOfferInfo(); System.out.println("status = "+status.getStatusType());
System.out.println("offerInfo = "+offerInfo.getName());
} catch (Exception exception) {
//Hier die Ausnahmebedingung behandeln.
exception.printStackTrace();
}
```

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

Bei der Axis2-Engine haben die generierten Klassen und Stubs keine parametrisierten Konstruktoren mehr. Einzelheiten hierzu finden Sie unter „Änderungen bei den parametrisierten Konstruktoren der unterstützenden Klassen“ auf Seite 2.

---

## Referenzen

Für diese Spezifikation wurden folgende Referenzen 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>)

---

## Anforderungen

In diesem Abschnitt werden die Anforderungen zusammengefasst, die das Design behandelt. Die CampaignServices-API muss:

- differenzierten Zugriff für das Erstellen, Erkennen, Lesen und Aktualisieren von IBM Campaign-Komponenten bereitstellen und dabei gleichzeitig Clients von zugrunde liegenden Implementierungsdetails isolieren
- mit bestehenden, grafisch orientierten IBM Campaign-Benutzern koexistieren und sie möglichst wenig beeinflussen
- Datenintegrität garantieren
- die Sicherheitsarchitektur von IBM Campaign unterstützen
- standardisiertes SOAP unterstützen, einschließlich sicherer Authentifizierung

---

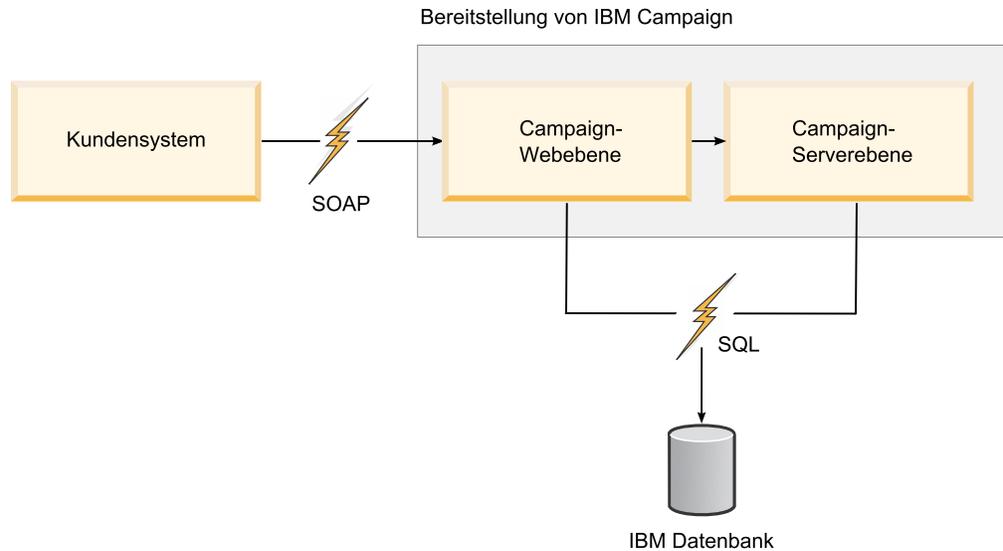
## Entwurfsüberblick

Die CampaignServices-API ist eine Art Front, die eine Clientansicht einer aktiven IBM Campaign-Anwendungsinstanz bereitstellt. Es ist zwar nur ein Teil der IBM Campaign-Funktionalität verfügbar, aber damit können die wichtigsten Aspekte von Campaign gesteuert werden. Die API ist so ausgelegt, dass sie gleichzeitig mit IBM Campaign-Webbenutzer- und anderen -API-Threads verwendet wird.

Die API unterstützt generell die folgenden Operationstypen für Kampagnen, Angebote und Zielzellenkomponenten:

- Erstellen von Komponenten
- Erkennen von Komponenten
- Löschen von Komponenten
- Erstellen, Prüfen und Bearbeiten von Komponentenattributen und Attributmetadaten
- Abrufen von Ablaufdiagrammausführungsergebnissen

Das folgende Diagramm zeigt eine Beispielimplementierung von CampaignServices 3.0.



## Notes

In diesem Abschnitt werden bestimmte Punkte des Designs behandelt.

### Versionen und Abwärtskompatibilität

Generell werden zukünftige Versionen der CampaignServices-API mit früheren Versionen mit allen Nebenversionen und Wartungsreleases mit gemeinsamer Hauptversionsnummer kompatibel sein. IBM behält sich jedoch das Recht vor, keine Kompatibilität mit früheren Versionen für Hauptreleases (x.0) zu gewährleisten, wenn die vorliegende Geschäftssituation bzw. die technischen Umstände dies rechtfertigen.

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

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

Die Nebenversionsnummer dieser API wird inkrementell erhöht, wenn eine der folgenden Änderungen vorgenommen wird:

- Neue Methode hinzugefügt
- Neuer Datentyp hinzugefügt und dessen Verwendung auf neue Methoden beschränkt
- Neuer Typ zu einem Aufzählungstyp hinzugefügt
- Neue Version einer Benutzeroberfläche definiert

IBM wird insbesondere weiterhin die veröffentlichte WSDL, den SOAP-Client und die Version von Apache Axis unterstützen, die zur Implementierung des SOAP-Angebots verwendet werden, und zwar mindestens bis zum nächsten Hauptrelease von IBM. In der Praxis wird das durch die gleichzeitige Unterstützung mehrerer versionsspezifischer Web-Services erreicht. (IBM unterstützt intern bereits mehrere Versionen dieses Service.)

## Benutzerauthentifizierung

Die Authentifizierung dient zur Ermittlung der Benutzeridentität.

**Anmerkung:** In diesem Release liegt die Benutzerauthentifizierung in der Zuständigkeit der Clientanwendung.

## Benutzerautorisierung

Mithilfe der Benutzerautorisierung werden die Berechtigungen eines authentifizierten Benutzers in Bezug auf die über die API zugänglichen Komponenten und Operationen festgelegt.

Es ist möglich, dass ein Benutzer erfolgreich authentifiziert wird, aber nicht über ausreichende Berechtigungen zum Ausführen bestimmter Operationen wie das Bearbeiten von Übersichtsdaten zu Kampagnen verfügt. In dem Fall löst die API-Methode `AuthorizationException` aus.

## Ländereinstellung

API-Anforderungen bieten einen optionalen **requestedLocale**-Parameter zur Definition der Ländereinstellung, die für die jeweilige Anforderung verwendet werden soll. Wenn dieser Parameter nicht angegeben wird, verwendet die API standardmäßig die bevorzugte Ländereinstellung des IBM -Benutzers. Der übliche Java-Algorithmus zum bestmöglichen Abgleich wird verwendet, um Nachrichten und andere lokalisierte Texte in der angeforderten Ländereinstellung zurückzugeben.

Dieser Parameter ist vom Typ `java.util.Locale`-Klasse.

**Anmerkung:** Einige benutzerdefinierte Texte, z. B. Kampagnenbeschreibungen, erscheinen in der Ländereinstellung des Benutzers, der den Text angegeben hat. IBM Campaign versucht nicht, solche Daten zu lokalisieren. Es werden nur Informations-, Warnungs- und Fehlermeldungen von der API lokalisiert.

## Statusverwaltung

Die CampaignServices-API ist statusunabhängig, d. h., es werden keine clientbezogenen Informationen von der API für nachfolgende Aufrufe übernommen.

Beachten Sie jedoch, dass bestimmte API-Aufrufe den Status von zugrunde liegenden Komponenteninstanzen, die von Campaign verwaltet werden, ändern können. Diese Statusänderungen werden gegebenenfalls in der Datenbank gespeichert.



---

## Kapitel 2. Datentypen

In diesem Abschnitt werden die von der CampaignServices-API verwendeten allgemein zugänglichen Datentypen definiert.

---

### WSReference

Ein einfacher Wrapper für eine Datenbank-ID:

- **componentTypeEnum**: ein Auflistungstyp, der den Komponententyp angibt, für den die ID bestimmt ist. Einer der folgenden Werte:
  - FOLDER
  - CAMPAIGN
  - FLOWCHART
  - TCS\_CELL
  - OFFER
  - OFFER\_LIST
  - OFFER\_TEMPLATE
- *id*: ein *Langwert* zur Definition einer eindeutigen, numerischen, datenbankspezifischen ID für diese Referenz.

---

### WSVersion

Ein Wrappertyp, der die unterschiedlichen Komponenten einer Version erfasst, einschließlich folgender Angaben:

- *major*: eine Ganzzahl, die die Hauptversionsnummer definiert, z. B. "8" bei der Vollversion 8.1.2.3.
- *minor*: eine Ganzzahl, die die Nebenversionsnummer definiert, z. B. "1" bei der Vollversion 8.1.2.3.
- *maintenance*: optionale Ganzzahl, die, falls zutreffend, die Wartungsnummer der Version definiert, z. B. "2" bei der Vollversion 8.1.2.3. Wird bei API-Versionen nie bereitgestellt.
- *patch*: optionale Ganzzahl, die, falls zutreffend, die Patchnummer des Release definiert, z. B. "3" bei der Vollversion 8.1.2.3. Wird bei API-Versionen nie bereitgestellt.

---

### WSServiceInfo

Ein einfacher Wrapper für Serviceinformationen. Enthält die folgenden Felder:

- *apiVersion*: eine *WSVersion*-Instanz, die die aktuellste Version der vom Service unterstützten API definiert. (Beachten Sie, dass *apiVersion* nur Informationen zu Haupt- und Nebenversionen enthält.)
- *campaignVersion*: eine *WSVersion*-Instanz, die die Vollversion der zugrunde liegenden IBM Campaign-Instanz definiert.
- *name*: interner Name des Service, z. B. "CampaignServices30Service".

---

## WSAttributeTypeEnum

Ein Aufzählungstyp, der alle möglichen Attributtypen definiert, d. h.:

- STANDARD: durch Campaign definiertes Standard- oder Basisattribut.
- CUSTOM: ein durch eine andere IBM -Anwendung, den Kunden oder einen Dritten definiertes Attribut.
- INPUT\_PARAMETER: ein Eingabeparameter wie z. B. ein zur Ausführung eines IBM Campaign-Ablaufdiagramms verwendetes Attribut.
- OUTPUT\_PARAMETER: ein Ausgabeparameter wie z. B ein Attribut, dessen Wert als Resultat der Ausführung eines IBM Campaign-Ablaufdiagramms ausgeführt wird.

---

## WSAttributeStatusEnum

Eine Auflistung aller möglichen Attributstatuscodes, d. h.:

- ACTIVE: das Attribut ist aktiv und kann beliebig verwendet werden.
- RETIRED: das Attribut wurde gesperrt und sollte nicht verwendet werden.

---

## WSAccessTypeEnum

Ein Aufzählungstyp, der alle möglichen Zugriffstypen für Attributwerte definiert, d. h.:

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

Der Zugriff auf Attribute ist in Bezug auf die Sicherheitsberechtigungen kumulativ, d. h., wenn die Sicherheitsrichtlinie für den Clientbenutzer den Lesezugriff auf ein bestimmtes Attribut verweigert, kann diese Sicherheitseinstellung nicht durch den Attributzugriff außer Kraft gesetzt werden. Die API würde das Attribut gar nicht an den Client zurückgeben.

---

## WSSelectTypeEnum

Definiert alle möglichen Auswahltypen eines bestimmten Attributwerts, d. h.:

- NONE: kein Auswahl (*hasOptions* = false).
- SINGLE\_SELECT: es kann jeweils nur eine Attributoption aus der Liste aller Optionen ausgewählt werden (gilt nur, wenn für das Attribut *hasOptions* festgelegt ist).
- MULTIPLE\_SELECT: ähnlich wie SINGLE\_SELECT, aber es kann mehr als eine Option ausgewählt werden.

---

## WSRunStatusEnum

Ein Auflistungstyp aller möglichen Ausführungsfortschritt von Ablaufdiagrammen, Verzweigungen oder Zellen, d. h.:

- NOT\_STARTED: die Ausführung ist geplant, wurde aber noch nicht gestartet.
- RUNNING: Ausführung in Bearbeitung.
- CANCELLED: die Ausführung wurde abgebrochen, entweder von einem Campaign-Benutzer oder über diese API.
- SUCCEEDED: die Ausführung wurde erfolgreich abgeschlossen.

- FAILED: die Ausführung ist fehlgeschlagen; Fehlerdetails werden separat aufgelistet (siehe „WSRunResults“ auf Seite 15).

---

## WSRunTypeEnum

Ein Aufzählungstyp aller möglichen Ausführungstypen, d. h.:

- 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 Anhängen beliebiger Daten an Komponenteninstanzen, die über die API zugänglich sind. Das können Standarddaten wie der *Name* einer Kampagne sein, Eingabeparameter zur Ausführung von Ablaufdiagrammen wie *Geschlecht* oder beliebige benutzerdefinierte Daten, die durch eine andere IBM -Anwendung oder einen anderen IBM -Kunden festgelegt werden.

**Anmerkung:** In dieser API werden Attribute zum Modellieren der meisten Komponentendaten verwendet, nicht nur benutzerdefinierte Campaign-Attribute.

Komponenten sind generell viele Attribute zugeordnet, die über die CampaignServices-API in Form einer speziellen typisierten Zuordnung, der *AttributeMap* zugänglich sind. Attributdaten werden in der gesamten API als stark typisierte, konkrete Klassen dargestellt, z. B. als *WSDecimalAttribute* bei Attributen, die eine Dezimalzahl (numerische Daten mit doppelter Genauigkeit) enthalten.

Jedes Attribut beinhaltet:

- Den *Namen*: der eindeutige Name des Attributs. Dieser Name dient als Schlüssel zum Zugriff auf das Attribut und dessen Metadaten innerhalb der Komponenteninstanz des Attributs. Das Format des Namens ist nicht definiert; in einigen Fällen wird er durch den Service, den Client oder einen IBM Campaign-Benutzer zugeordnet.

Generell ist dieser Name nicht der Anzeigename, der dem Campaign- oder Clientbenutzer angezeigt wird. Er kann durch die API standardisiert werden, z. B. *uacDescription*, oder durch IBM Campaign bei der Veröffentlichung von Ablaufdiagrammen oder durch die IBM -Anwendung oder den IBM -Kunden beim Definieren der benutzerdefinierten Attribute zugeordnet werden. In jedem Fall ist der Name aber definitiv eindeutig.

- *Metadaten*: (optionale) Informationen zu den Daten des Attributs wie Wertdatentyp, Anzeigename, Beschreibung, Eingabeaufforderungen, Standardwerte, Auswahltypen, Länge (Text), Genauigkeit (Dezimalzahlen), Optionen (Einfach- oder Mehrfachauswahl) etc. Siehe „WSAttributeMetadata“ auf Seite 12.

- **Werte:** ein Array von null oder mehr stark typisierten Wertobjekten. Das Wertfeld wird von der konkreten Attributklasse bereitgestellt; der Typ jedes Werts muss identisch sein und mit der Typendefinition im Metadatenfeld des Attributs übereinstimmen. Es unterstützen jedoch nicht alle Attribute Mehrfachwerte.

Folgende konkrete Attributtypen werden unterstützt:

- **WSBooleanAttribute:** ein Attribut mit booleschem Wert, d. h. *true* oder *false*.
- **WSIntegerAttribute:** ein Ganzzahlwert (*java.lang.Long*).
- **WSDecimalAttribute:** eine Dezimalzahl mit doppelter Genauigkeit (*java.lang.Double*).
- **WSCurrencyAttribute:** ein zusammengesetzter Währungswert, optional mit ISO 4217-Währungscode für den Währungswert, wie "USD" für US-Dollar, und als *Double* erfassten Währungswerten. Wenn der Währungscode nicht bereitgestellt wird, wird der von IBM Campaign verwendete Standardwert vorausgesetzt.  
Unter <http://www.xe.com/symbols.php> ist eine Liste mit Ländern, Währungssymbolen und Codes erhältlich. Beachten Sie dabei, dass sich die Ländereinstellung für einen Währungswert von der bevorzugten Ländereinstellung des Benutzers unterscheiden kann.
- **WSCalendarAttribute:** dessen Werte Kalenderdaten oder Uhrzeiten gemäß der jeweiligen Zeitzone und Ländereinstellung sind.
- **WSTextAttribute:** eine Zeichenfolge mit Unicode-Text (möglicherweise Null oder leer).

**Anmerkung:** Generell unterscheidet sich die Liste möglicher Attribute bei jeder Komponente, es kann aber Überschneidungen geben.

---

## WSAttributeMetadata

WSAttributeMetadata definiert Informationen zu Daten eines bestimmten typisierten Attributs, z. B. Wertdatentyp, lokalisierter Text (Anzeigenname, Beschreibung, Eingabeaufforderungen etc.), Standardwert, zulässiger Wertebereich, Auswahltypen, Optionen (Mehrfach- oder Einzelauswahl) etc. Genau wie Attribute sind auch Attributmetadaten stark typisiert. So muss z. B. an das WSDecimalAttribute *myNumber* das Element *WSDecimalAttributeMetadata* gebunden sein und alle Werte, einschließlich der Attributwerte, des Metadatenstandardwerts und möglicher Optionswerte, werden als *Double* typisiert.

Beschreibungen, Beschriftungen und andere Attributmetadaten sind generell lokalisiert; benutzerdefinierter Text steht aber möglicherweise nur in der Sprache zur Verfügung, in der der Benutzer ihn eingegeben hat. Jeder API-Aufruf beinhaltet eine angeforderte Ländereinstellung, mit der der Client-Code bestimmen kann, in welcher Ländereinstellung der jeweilige Benutzer die lokalisierten Nachrichten angezeigt bekommen möchte. Zur Erfüllung der Anforderung werden die üblichen Fallback-Richtlinien für die Java-Ländereinstellung verwendet.

WSAttributeMetadata enthält folgende Felder:

- *name:* der Attributname, Standardwert oder benutzerdefiniert; auch der Name, der vom Attribut verwendet wird, das an diese Metadaten gebunden ist. Standardattribute werden vom System definiert und haben Standardnamen aus einem reservierten Namensbereich (d. h. sie verwenden ein Präfix "uac"), benutzerdefinierte Namen unterliegen beliebigen anderen Namenskonventionen.

**Anmerkung:** Der Attributname muss eindeutig sein, wird nie lokalisiert und unterliegt einer Längenbeschränkung (abhängig vom Zeicheninhalt und der Da-

tenbank). Bei dem Namen muss die Groß-/Kleinschreibung nicht beachtet werden und er kann aus einer beliebigen Kombination aus Unicode-Buchstaben oder -Ziffern bestehen, inklusive Unterstrich '\_', darf aber nicht mit einer Ziffer beginnen.

- *description*: optionale Beschreibung des Attributs. Geeignet als QuickInfo oder für eine andere Darstellung auf der Benutzeroberfläche.
- Prädikate: unterschiedliche Prädikate zur Beschreibung des Attributs:
  - *isRequired*: true, wenn das Attribut obligatorisch ist.
  - *isInternal*: true, wenn das Attribut durch das System definiert und nur zur internen Verwendung vorgesehen ist (sollte nicht dem Benutzer angezeigt werden).
  - *isGenerated*: true, wenn der Wert des Attributs bei der Erstellung der Komponente automatisch durch IBM Campaign generiert wird, z. B. der Zielzellen-code. Normalerweise gilt bei generierten Werten für *accessTypeEnum* der Wert READ\_ONLY.
  - *hasOptions*: true, wenn das Attribut Optionen hat. Setzt voraus, dass Optionen für diese Metadaten definiert sind und dass für *selectTypeEnum* entweder SINGLE\_SELECT oder MULTIPLE\_SELECT festgelegt ist.
- *typeEnum*: ein Element vom Typ *WSAttributeTypeEnum*, das den Typ des Attributs definiert, z. B. STANDARD oder CUSTOM.
- *statusEnum*: ein Element vom Typ *WSAttributeStatusEnum*, das den Status des Attributs definiert, z. B. ACTIVE.
- *accessTypeEnum*: ein Element vom Typ *WSAccessTypeEnum*, das den Zugriffstyp für den Attributwert definiert, z. B. READ\_ONLY.
- *selectTypeEnum*: ein Element vom Typ *WSAccessTypeEnum*, das den Auswahltyp für das Attribut definiert, z. B. SINGLE. Bei Kampagnen- und Zellenattributen, oder wenn keine Optionen bereitgestellt werden, muss NONE festgelegt sein.
- *componentTypeEnum*: ein Element vom Typ *WSComponentTypeEnum* für alle möglichen Campaign-Komponenten, die über die API zugänglich sind, z. B. CAMPAIGN oder FOLDER.
- *defaultValue* (nur bei Ablaufdiagrammen): optionaler Typenstandardwert des Attributs. Dieser Wert wird durch die konkrete Attributmetadatenklasse bereitgestellt, z. B. ist der Standardwert von *WSTextAttributeMetadata* vom Typ Zeichenfolge. (Siehe hierzu die Beschreibung der Attributwerte.) Für Komponenten, die keine Ablaufdiagramme sind, ist der Standardwert nicht definiert.
- *options*: optionale Liste der Optionen für dieses Attribut. Zusammen definieren die Optionen eines Attributs die genaue Gruppe von gültigen Werten für dieses Attribut; jede Option ist stark typisiert, sodass z. B. an *WSTextAttributeMetadata* nur *WSTextAttributeOption* gebunden sein darf.

**Anmerkung:** Für Optionen gilt eine Beschränkung; es werden nur Textattribute unterstützt.

Jede Option definiert Folgendes:

- *prompt*: Eingabeaufforderung für die Option, geeignet für Pulldown-Menüs, z. B. "Männlich" als Geschlechtsattributoption. Beachten Sie, dass anders als bei der Metadateneingabeaufforderung die Optionsanzeigenamen keine Interpunktion enthalten.
- *description*: lokalisierte Beschreibung der Option, z. B. "Eine Person männlichen Geschlechts". Geeignet für QuickInfo-Text.
- *isDefault*: true, wenn diese Option der Standardwert ist. Beim Auswahltyp MULTIPLE\_SELECT kann mehr als eine Option als Standardwert markiert werden.

- *value*: der typisierte Optionswert. Wie auch beim defaultValue der Attributmetadaten wird dieser Wert durch die konkrete Optionsunterklasse bereitgestellt, wie z. B. der Wert von *WSDecimalAttributeOption* vom Typ Dezimal ist. (Siehe hierzu die Beschreibung der Attributwerte.) Um beim Beispiel *Geschlecht* zu bleiben, könnte der Wert entweder als Zeichenfolge "m" bereitgestellt werden (*WSTextAttributeOption*) oder als numerischer Code 123 (*WSDecimalAttributeOption*).

---

## WSCampaignInfo

Ein einfacher Wrapper für Kampagnenattributdaten.

Enthält die folgenden Felder:

- *reference*: die Referenz der Kampagne.
- *name*: der Kampagnenname (*uacName*); nicht garantiert eindeutig.
- *description*: optionale Beschreibung der Kampagne (*uacDescription*).
- *campaignCode*: der eindeutig Code der Kampagne (*uacCampaignCode*); zugeordnet durch den Client oder Campaign.

---

## WSComponentOrFolderInfo

Enthält eine Kombination aus eingeschlossenen Kampagnen- oder Ordnerattributdaten wie Anzeigenname, Referenz etc.

Enthält die folgenden Felder:

- *reference*: die Referenz der Komponente oder des Ordners.
- *name*: der Name der Komponente oder des Ordners (*uacName*); nicht garantiert eindeutig.
- *description*: optionale Beschreibung der Komponente oder des Ordners (*uacDescription*).
- *componentCode*: eindeutiger Code der Komponente oder Null bei einem Ordner.

---

## WSTargetCellInfo

Ein einfacher Wrapper für Attributdaten einer Zielzellenreihe:

Enthält die folgenden Felder:

- *reference*: die Zellenreferenz.
- *name*: Zellenname (*uacName*); nicht garantiert eindeutig.
- *description*: optionale Beschreibung der Zelle (*uacDescription*).
- *cellCode*: der Zellencode (*uacCellCode*); zugeordnet durch den Client oder Campaign. Beachten Sie, dass für jeden Code erzwungen werden kann, dass er eindeutig ist, indem der Konfigurationsparameter *DuplicateCellCodesAllowed* in IBM Campaign auf false festgelegt wird.
- *flowchartName*: optionaler Name des Ablaufdiagramms, an das die Zelle gebunden ist.

---

## WSMetricsInfo

Ein einfacher Wrapper für Kampagnenanalysedaten, einschließlich Anzahl Kontakte. Enthält die folgenden Felder:

- *totalContacts*: ein Langwert zur Angabe der Gesamtzahl Kontakte.

- *responses*: eine typisierte Liste der *WSMetricsResponse*-Instanzen, wobei jede Instanz die Kontaktinformationen für eine Antwort definiert:
  - *typeCode*: eine Zeichenfolge, die den Antworttypcode definiert, z. B. *PHC* für einen Telefonanrufkontakt.
  - *count*: ein Langwert, der angibt, wie oft der Kontakt zustande kam.

---

## WSRunResults

Ein Wrappertyp für die Ergebnisse eines Ablaufdiagramms, eines Prozessfeldes oder einer Zellenausführung, einschließlich Ausführungsfortschritt, Start- und Endzeit/-datum der Ausführung des Ablaufdiagramms und Zählern.

Folgende Felder sind enthalten:

- *sourceReference*: optionale Referenz der Quelle der Ausführungsergebnisse. Abhängig vom Kontext des Abrufens von Ausführungsergebnissen kann dies auf ein Ablaufdiagramm, ein Ablaufdiagrammprozessfeld oder eine Zielzelle verweisen. In jedem Fall verweisen die verbleibenden Ausführungsergebnisdaten auf diese Quelle.
- *flowchartName*: der Name des ausgeführten Ablaufdiagramms.
- *flowchartId*: die Datenbank-ID des Ablaufdiagramms.
- *runId*: die Datenbank-ID der Ausführung.
- *typeEnum*: ein Auflistungstyp, der definiert, welche Art Ausführung die Ergebnisse generiert hat, z. B. *PRODUCTION\_PROCESS* (siehe *WSRunTypeEnum*).
- *statusEnum*: ein Auflistungstyp, der den Ausführungsfortschritt definiert, z. B. *RUNNING* (siehe *WSRunStatusEnum*).
- *statusCode*: optionaler ganzzahliger Statuscode.
- *statusMessage*: optionale Statusnachricht.
- *startDate*: optionale Angabe von Datum/Uhrzeit des Starts der Ausführung; ist Null, wenn die Ausführung nicht gestartet wurde.
- *endDate*: siehe *startDate*, aber Datum/Uhrzeit des Endes der Ausführung (erfolgreich oder fehlgeschlagen); ist Null, wenn die Ausführung nicht gestartet oder noch nicht beendet wurde.
- *count*: optionale Gesamtzahl der von der Ausführung ausgewählten Kontakte; kann Null sein, wenn die Ausführung nicht abgeschlossen wurde.

---

## WSOfferInfo

Ein einfacher Wrapper für Attributdaten von Angeboten oder Angebotslisten.

Enthält die folgenden Felder:

- *reference*: die Referenz des Angebots oder der Angebotsliste.
- *name*: der Name des Angebots oder 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 Wrapper für Angebotscodes oder Angebotslistennamen.

Enthält die folgenden Felder:

- *isCode*: boolescher Wert, der angibt, ob das Feld *codeOrName* ein angenommener Angebotscode (true) oder der Name einer Angebotsliste ist (false).
- *codeOrName*: der eindeutige Angebotscode (*uacOfferCode*) bei einem Angebot oder der Name der Angebotsliste.

---

## WSOfferValidationInfo

Ein einfacher Wrapper für Angebotsüberprüfungsinformationen.

Enthält die folgenden Felder:

- *errorCode*: definiert, wenn nicht Null, den alphanumerischen Überprüfungsfehlercode. Siehe Klasse *IStandardDefinitions* zu Fehlercodes.
- *errorMessage*: optionale lokalisierte Nachricht, die den Fehler beschreibt (wenn einer auftrat).
- *codeOrName*: der überprüfte Angebotscode oder der Angebotslistenname.
- *reference*: die Referenz des Angebots oder der Angebotsliste, wenn gültig.

---

## WSOfferTemplateInfo

Ein einfacher Wrapper für Angebotsvorlagendaten.

Enthält die folgenden Felder:

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

---

## WSBulkOfferInfo

Wird zur Massenerstellung von Angeboten verwendet.

Enthält die folgenden Felder:

- *offerName*: der Name des Angebots, das erstellt wird.
- *attributes*: ein Array von *WSAttribute*-Typen, das die Angebotsattribute angibt.

---

## WSOfferInfoStatus

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

Enthält die folgenden Felder:

- *name*: der Name des Angebots.
- *code*: der Angebotscode. Ist Null, wenn die Angebotserstellung fehlschlägt.
- *description*: Angebotsbeschreibung.
- *reference*: die WSReference des erstellten Angebots. Ist Null, wenn die Angebotserstellung fehlschlägt.
- *status*: eine Instanz von *WSRequestStatus*, die den Status der Angebotserstellung angibt.



---

## Kapitel 3. Allgemeine Ausnahmebedingungen

In diesem Abschnitt werden einige allgemeine Ausnahmebedingungen erläutert, die von der CampaignServices-API ausgelöst werden.

Alle lokalisierten Nachrichten der Ausnahmebedingung werden in der angeforderten Ländereinstellung ausgegeben, sofern sie für IBM Campaign verfügbar ist. Es finden die üblichen Java-Fallback-Richtlinien für die Ländereinstellung Anwendung.

---

### RemoteException

**Anmerkung:** Dieses Element gilt nur für die SOAP-Benutzeroberfläche.

Alle SOAP-Aufrufe an die API können die Ausnahmebedingung *RemoteException* auslösen, wenn ein Fehler auf Systemebene auftritt, etwa ein Problem in der SOAP-Umschlagsprozessebene (Axis), oder wenn eine in der Web-Service-WSDL definierte Bedingung aus irgendeinem Grund nicht eingehalten wurde.

Allgemeine geprüfte und ungeprüfte API-Ausnahmebedingungen wie *DataException* werden als Fehlerstatus, nicht als Ausnahmebedingung vom Typ *RemoteException* zurückgegeben.

Weitere Informationen erhalten Sie im Abschnitt zur SOAP-Benutzeroberfläche.

---

### AuthenticationException

Der Benutzer konnte für die angegebene Campaign-Partition nicht authentifiziert werden. Überprüfen Sie die in Marketing Platform definierte Benutzerrolle.

---

### AuthorizationException

Der Benutzer ist nicht berechtigt, die angeforderte Operation auszuführen. Diese Ausnahmebedingung kann von jeder API-Methode ausgelöst werden, sie ist also nicht deklariert (nicht geprüft). Prüfen Sie die der Benutzerrolle zugeordneten Berechtigungen in IBM Marketing Platform.

---

### DataException

Eine schwerwiegende Ausnahmebedingung ist in der zugrunde liegenden Datenbankebene in Campaign (ungeprüft) aufgetreten.

Prüfen Sie die Einzelangaben in den Ablaufdiagramm- und Listenerprotokollen zu Campaign.

---

## LockException

Eine temporäre Ausnahmebedingung, die ausgelöst wird, wenn der Client versucht, eine Komponente zu aktualisieren, etwa ein Ablaufdiagramm, während sie von einem anderen Benutzer bearbeitet wird. Generell kann diese Ausnahmebedingung aufgelöst werden, indem eine Zeit lang gewartet und dann die Operation wiederholt wird. Die Wiederholungslogik liegt jedoch in der Zuständigkeit des Clients.

---

## InvalidComponentException

Es wurde ein Versuch unternommen, auf ein ungültiges oder unbekanntes Attribut (Kampagne, Ablaufdiagramm, Zielzelle etc.) zu verweisen. Die Ausnahmebedingungsmethode `getComponentReference()` gibt die Referenz der unzulässigen Komponente zurück.

---

## InvalidAttributeException

Diese Ausnahmebedingung wird ausgelöst, wenn der Client ein ungültiges Attribut bereitstellt oder darauf verweist, z. B. wenn er den falschen Datentyp verwendet oder ein Wertearray, obwohl keines zulässig ist. Die Ausnahmebedingungsmethode `getAttributeName()` gibt den Namen des problematischen Attributs zurück, `getAttributeValue()` gibt den Wert zurück und `getComponentReference()` bestimmt die Komponente (oder den Massenindex).

---

## AttributeNotFoundException

Wird ausgelöst, wenn der Client versucht, auf ein unbekanntes Attribut (Kampagne, Ablaufdiagramm, Zielzelle etc.) zu verweisen. Die Methode `getAttributeName()` der Ausnahmebedingung gibt den Namen des Attributs ohne Entsprechung zurück, `getComponentReference()` ermittelt die Komponente (oder den Massenindex).

---

## AttributeExistsException

Wird ausgelöst, wenn der Client versucht, für eine Komponente doppelte Attributmetadaten zu definieren. Die Methode `getAttributeName()` der Ausnahmebedingung gibt den Namen des doppelten Attributs zurück, `getComponentReference()` ermittelt die Komponente (oder den Massenindex).

---

## CompositeException

Eine *CompositeException* wird von einigen APIs verwendet, um dem aufrufenden Programm mehrfache Fehler zurückzumelden. In der Regel ist mehr als eine Ursache an die Ausnahmebedingung gebunden; alle Ursachen werden als Liste erfasst, und zwar in der Reihenfolge, in der sie aufgetreten sind. Die Ausnahmebedingungsmethode `getCauseList()` gibt diese Liste zurück, die dann untersucht werden kann, um Einzelangaben zu jedem Fehler zu erhalten.

**Anmerkung:** Generell wird die API ihre Arbeit entweder erfolgreich abschließen oder ein Rollback ausführen, bevor sie *CompositeException* auslöst. Siehe hierzu z. B. die im Abschnitt „Zielzellenmethoden“ auf Seite 31 beschriebenen Massen-TCS-APIs.

---

## Kapitel 4. CampaignServices-API-Methoden

In diesem Abschnitt werden die Hauptmethoden definiert, die von der CampaignServices 3.0-API exportiert werden.

---

### Servicemethoden

Mithilfe der API ist es möglich, Identifikationsinformationen zum Service selbst zu bestimmen.

#### getServiceInfo

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

Gibt Informationen zum Service zurück, z. B. die aktuellste unterstützte API-Version, die Vollversion der zugrunde liegenden IBM Campaign-Instanz etc.

**Anmerkung:** Für diesen Aufruf sind keine Clientinformationen notwendig und es finden keine Sicherheitsberechtigungen Anwendung.

#### Parameter

Keine.

#### Rückgabe

Gibt eine Instanz vom Typ *WSServiceInfo* zurück.

#### Fehler

Keine.

---

### Attribute

Die meisten Komponenteninstanzdaten sind über die API als *Attribute* oder *Attributmetadaten* zugänglich. In einigen Fällen gelten die Attributmetadaten definitionen global für IBM Campaign (z. B. benutzerdefinierte Kampagnenattribute), andere sind auf bestimmte Komponenten begrenzt (z. B. Benutzervariablen für Ablaufdiagramme).

Soweit nicht anders festgelegt, können alle Attribute gelesen werden, wenn der Client über ausreichende Sicherheitsberechtigungen verfügt.

**Anmerkung:** Über diese API sind nur aktive Komponenten zugänglich, auf die der Client Zugriff hat. Öffentliche Unterstützung ist auf Angebotsattribute, Angebotsvorlagen und Metadaten beschränkt.

#### getAttributesByName

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

Ruft die genannten Attribute ab, die der angegebenen Komponenteninstanz zugeordnet sind (kann leer sein).

## Parameter

*userCredential*: der Clientbenutzerberechtigungsnauchweis.

*requestedLocale*: optionale Ländereinstellung für diese Anforderung; wird diese Option nicht angegeben, dann wird die Ländereinstellung des IBM -Benutzers verwendet. Falls erforderlich, werden normale Standardalgorithmen für die Ländereinstellung angewendet.

*partitionName*: optionaler Name der zu verwendenden Kampagnenpartition. Wenn diese Option nicht definiert wird, wird die Standardpartition verwendet.

*reference*: die Referenz der Komponenteninstanz, die die gewünschten Attribute enthält. *InvalidComponentException* wird ausgelöst, wenn die Referenz ungültig oder die Komponente nicht vorhanden ist.

*names*: optionales Array von Namen der abzurufenden Attribute (nicht Anzeigenamen); wird diese Option nicht angegeben, werden alle Attribute zurückgegeben. Löst *AttributeNotFoundException* aus, wenn eines der genannten Attribute nicht vorhanden ist.

## Rückgabe

Eine typisierte Zuordnung von null oder mehr Attributen; der Attributname ist der Zuordnungseintragungsschlüssel und die Attributinstanz ist der Eintragungswert.

## Fehler

*InvalidComponentException*, *AttributeNotFoundException*

*AuthorizationException*, *DataException*

**Anmerkung:** All diese Ausnahmebedingungen sind in *CampaignServicesException* eingeschlossen.

## updateAttributes

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

Aktualisiert eines oder mehrere Attribute der Komponenteninstanz mit den bereitgestellten Attributwerten.

## Aktualisierungslogik

Die Aktualisierungslogik ist wie folgt.

Für jedes in der bereitgestellten Attributzuordnung enthaltene Attribut gilt:

1. Wenn der Attributname mit einem bestehenden Attribut übereinstimmt, wird versucht, sein Feld *Werte* mit dem bereitgestellten Wertfeld zu überschreiben.
2. Wenn das Attribut noch nicht vorhanden ist, für *allowCreate* "true" gilt und seine Metadaten bekannt sind, wird das Attribut erstellt. Das gilt für globale Attributmetadaten ebenso wie für Instanzattribute (außer Ablaufdiagramme).

3. Wenn der Wertetyp oder ein anderer Aspekt der Attributmetadatendefinition nicht erfüllt wird oder einer oder mehrere der bereitgestellten Werte ungültig, außerhalb des gültigen Bereichs o. Ä. sind, wird *InvalidAttributeException* ausgelöst.
4. Andernfalls wird *AttributeNotFoundException* ausgelöst, wenn das genannte Attribut nicht vorhanden ist.

**Anmerkung:** Bei Auslösung einer Ausnahmebedingung wird keine Aktualisierung festgeschrieben.

Diese spezielle Methode bietet keine Unterstützung für das Definieren neuer benutzerdefinierter Attribute; verwenden Sie dazu die Methode `createAttributeMetadata()`.

In jedem Fall gelten für Attributaktualisierungsoperationen die üblichen Integritätsbedingungen für die Sicherheit und Gültigkeitsprüfungen. Es liegt in der Zuständigkeit des Clients zu bestimmen, welche Attribute für eine bestimmte Komponenteinstanz erforderlich sind, welche Typen korrekt sind etc.

### Parameter

*userCredential*: der Clientbenutzerberechtigungs-nachweis.

*requestedLocale*: optionale Ländereinstellung für diese Anforderung.

*partitionName*: optionaler Name der zu verwendenden Kampagnenpartition.

*reference*: die Referenz der Komponenteninstanz, die die zu aktualisierenden Attribute enthält.

*allowCreate*: gibt an, ob ein neues Attribut erstellt werden soll, wenn es für die Komponente noch nicht vorhanden ist. (Siehe „Aktualisierungslogik“ auf Seite 22.)

*attributes*: ein Array von zu aktualisierenden Attributen; zum Lokalisieren des zu aktualisierenden Attributs wird der Attributname verwendet und die neuen Werte werden zum Aktualisieren der bestehenden Attributwerte verwendet, als einzelnes Objekt des ordnungsgemäßen Typs oder als Array, falls zutreffend. (Siehe Kapitel 3, „Allgemeine Ausnahmebedingungen“, auf Seite 19.)

### Rückgabe

Keine Rückgabe.

### 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 genannten Attributmetadatendefinitionen ab, die an eine bestimmte Komponente oder Vorlage gebunden oder global definiert sind.

## Parameter

*userCredential*: der Clientbenutzerberechtigungs-nachweis.

*requestedLocale*: optionale Ländereinstellung für diese Anforderung.

*partitionName*: optionaler Name der zu verwendenden Kampagnenpartition.

*reference*: optionale Referenz der Komponente oder Vorlage, die die gewünschten Attributmetadaten enthält. Wenn nur `ComponentTypeEnum` angegeben wird, werden nur Komponenten dieses Typs abgerufen; wenn gar keine Referenz angegeben wurde, werden alle globalen Metadatendefinitionen für alle Komponententypen abgerufen. Löst `InvalidComponentException` aus, wenn die angegebene Referenz ungültig ist.

*names*: optionales Array von Namen der abzurufenden Attributmetadaten; wird diese Option nicht angegeben, werden alle Metadaten - oder global definierte, wenn keine Referenz angegeben wurde - für die Komponente zurückgegeben. Löst `AttributeNotFoundException` aus, wenn genannte Attributmetadatendefinitionen nicht vorhanden sind.

## Rückgabe

Eine typisierte Liste von null oder mehr Attributmetadatendefinitionen; der Attributname ist der Eintragungsschlüssel und die Attributmetadaten sind der Eintragungswert.

## Fehler

`InvalidComponentException`, `AttributeNotFoundException`

`AuthorizationException`, `DataException`

## createAttributeMetadata

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

Erstellt eine oder mehrere Attributmetadatendefinitionen und bindet sie optional an eine bestimmte Komponente oder Vorlage.

## Parameter

*userCredential*: der Clientbenutzerberechtigungs-nachweis.

*requestedLocale*: optionale Ländereinstellung für diese Anforderung.

*partitionName*: optionaler Name der zu verwendenden Kampagnenpartition.

*reference*: optionale Referenz der Komponente oder Vorlage, an die die Metadaten gebunden werden sollen. Wenn diese Option nicht angegeben wird, gilt die erstellte Metadatendefinition global. Wenn die Referenz angegeben wird, aber nicht gültig ist, wird `InvalidComponentException` ausgelöst.

*attributeMetadata*: ein Array von zu bindenden Attributmetadatendefinitionen. Wenn angegebene Metadaten bereits an die Komponente gebunden sind, der Name

also nicht eindeutig ist, wird *AttributeExistsException* ausgelöst. Löst *InvalidAttributeException* aus, wenn ein Problem mit angegebenen Metadaten vorliegt, sie also intern inkonsistent sind.

## Rückgabe

Keine Rückgabe.

## Fehler

*InvalidComponentException*, *AttributeExistsException*, *InvalidAttributeException*

*AuthorizationException*, *DataException*

## updateAttributeMetadata

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

Aktualisiert eine oder mehrere Attributmetadatendefinitionen der angegebenen Komponente oder Vorlage, optional werden neue Metadatendefinitionen erstellt, falls erforderlich.

## Aktualisierungslogik

Die Aktualisierungslogik ist wie folgt.

Für jede im bereitgestellten Array enthaltene Attributmetadatendefinition gilt:

1. Wenn der Attributname mit keinen an die Komponente gebundenen bestehenden Metadaten übereinstimmt, wird auf Basis des Parameterwerts *allowCreate* Folgendes unternommen:
  - a. *True*: neue Metadatendefinition erstellen. Funktional mit der Verwendung der Anforderung *createAttributeMetadata()* identisch.
  - b. *False*: *AttributeNotFoundException* auslösen.
2. Wenn der Attributmetadattyp unterschiedlich ist, *InvalidAttributeException* auslösen.
3. Versuch, die bestehende Attributmetadatendefinition mit den Feldwerten der bereitgestellten Metadaten zu überschreiben, andernfalls *InvalidAttributeException* auslösen. Es werden nur folgende Aktualisierungen unterstützt (andernfalls *InvalidAttributeException* auslösen):
  - a. *name*: kann nicht geändert werden (Name ist der Schlüssel!).
  - b. *displayName*: neuen Wert akzeptieren.
  - c. *description*: neuen Wert akzeptieren.
  - d. *isRequired*: nur Wechsel von *true* auf *false* ermöglichen.
  - e. *isInternal*: neuen Wert akzeptieren.
  - f. *isGenerated*: keine Änderung zulässig.
  - g. *attributeTypeEnum*: keine Änderung zulässig.
  - h. *accessTypeEnum*: neuen Wert akzeptieren.
  - i. *selectTypeEnum*: diese Übergänge akzeptieren, wenn Optionen bereitgestellt werden:
    - 1) NONE zu SINGLE\_SELECT oder MULTIPLE\_SELECT
    - 2) SINGLE\_SELECT zu MULTIPLE\_SELECT

- j. *options*: Optionen können hinzugefügt, aber nicht gelöscht werden. Es werden nur folgende Optionsänderungen unterstützt (gemäß Wertübereinstimmung):
    - 1) *displayName*: neuen Wert akzeptieren (kein Ripple).
    - 2) *description*: neuen Wert akzeptieren (kein Ripple).
    - 3) *isDefault*: neuen Wert akzeptieren; muss aber mit *SelectTypeEnum* übereinstimmen.
    - 4) *value*: keine Änderung zulässig (Wert ist der Schlüssel!).
  - k. *defaultValue* (nur Ablaufdiagramme): neuen Standardwert akzeptieren.
  - l. *maxLength* (nur Text): neue Länge akzeptieren, wenn größer.
4. Wenn die Attributmetadatendefinition in sich nicht konsistent ist, *InvalidAttributeException* auslösen.
  5. Falls erforderlich, alle Komponenteninstanzen suchen, die auf die aktualisierten Attributmetadaten verweisen, und entsprechend aktualisieren.

**Anmerkung:** Bei Auslösung einer Ausnahmebedingung wird keine Aktualisierung festgeschrieben.

In jedem Fall gelten für Attributaktualisierungsoperationen die üblichen Integritätsbedingungen für die Sicherheit und Gültigkeitsprüfungen.

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

## Parameter

*userCredential*: der Clientbenutzerberechtigungsnauchweis.

*requestedLocale*: optionale Ländereinstellung für diese Anforderung.

*partitionName*: optionaler Name der zu verwendenden Kampagnenpartition.

*reference*: optionale Referenz der Komponenteninstanz, die die gewünschten Attribute enthält. Wenn diese Option nicht angegeben wird, wird die Aktualisierung auf globale Metadatendefinitionen eingeschränkt. Löst *InvalidComponentException* aus, wenn die angegebene Referenz ungültig ist.

*allowCreate*: ist dieser Wert true, werden derzeit nicht vorhandene Metadatendefinitionen erstellt (funktional äquivalent zur Verwendung der Methode `createAttributeMetadata()`).

*attributeMetadata*: ein Array von zu aktualisierenden Attributmetadatendefinitionen (die hinzugefügt werden, wenn das Flag *allowCreate* auf true festgelegt ist); der Attributname wird verwendet, um die zu aktualisierende Metadatendefinition zu lokalisieren, und die übrigen Daten werden zur Aktualisierung der bestehenden Definition verwendet. (Siehe „Aktualisierungslogik“ auf Seite 25.)

## Rückgabe

Keine Rückgabe.

## Fehler

*InvalidComponentException*, *InvalidAttributeException*

*AuthorizationException*, *DataException*

## deleteAttributeMetadata

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

Löscht eine oder mehrere genannte Attributmetadatendefinitionen aus der angegebenen Komponente oder Vorlage (nur bei benutzerdefinierten Attributmetadaten) oder den globalen Attributmetadatendefinitionen.

Im Rahmen dieser Task sucht die Methode alle Komponenten, die auf die gelöschten Metadaten verweisen, und aktualisiert sie entsprechend.

**Anmerkung:** Bei Auslösung einer Ausnahmebedingung wird jedoch kein Löschvorgang festgeschrieben.

### Parameter

*userCredential*: der Clientbenutzerberechtigungs-nachweis.

*requestedLocale*: optionale Ländereinstellung für diese Anforderung.

*partitionName*: optionaler Name der zu verwendenden Kampagnenpartition.

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

**Anmerkung:** Wenn das optionale Array der Attributmetadaten-namen nicht angegeben wird, versucht diese Methode, alle benutzerdefinierten Attributmetadaten zu löschen, die der Komponente zugeordnet sind, oder alle globalen Definitionen, wenn die Referenz nicht angegeben wurde.

*names*: optionales Array der Namen der zu löschenden Attributmetadaten. Löst *AttributeNotFoundException* aus, wenn genannte Attributmetadaten nicht vorhanden sind. Löst *InvalidAttributeException* aus, wenn ein Attribut nicht entfernt werden konnte.

### Rückgabe

Keine Rückgabe.

### Fehler

*InvalidComponentException*, *AttributeNotFoundException*, *InvalidAttributeException*

*AuthorizationException*, *DataException*

---

## Kampagnen

Die API unterstützt folgende Operationen für Kampagnen (abhängig von den Sicherheitsberechtigungen):

- Erstellen einer neuen Kampagne
- Erkennen (Auflisten von Kampagnen gemäß unterschiedlicher Kriterien)
- Erstellen, Lesen und Aktualisieren von Attributen (über Attribut-APIs)

Kampagnen sind einige Standardattribute zugeordnet, die über die API zugänglich sind. Diese Liste kann vom Client beliebig erweitert werden, indem benutzerdefinierte Attribute hinzugefügt werden (siehe Attribut-APIs).

Zu den Standardkampagnenattributen gehören:

- *uacName*: Kampagnenname (nicht garantiert eindeutig).
- *uacDescription*: optionale Zeichenfolge zur Beschreibung der Kampagne.
- *uacCampaignCode*: eine Codezeichenfolge zur eindeutigen Bestimmung der Kampagne. Sie wird gewöhnlich automatisch durch Campaign erstellt, kann aber auch vom Client bereitgestellt werden.
- *uacCreateDate*: ein Kalender, der Datum und Uhrzeit der Erstellung der Kampagne durch den Server angibt.
- *uacUpdateDate*: ein Kalender, der Datum und Uhrzeit der letzten Aktualisierung der Kampagne durch den Server angibt.
- *uacInitiative*: optionale Zeichenfolge zur Definition der Kampagneninitiative.
- *uacObjectives*: optionale Zeichenfolge zur Angabe der Kampagnenziele.
- *uacStartDate*: ein optionaler Kalender zur Angabe von Datum und Uhrzeit des Starts der Kampagne durch den Server oder des geplanten Starts.
- *uacEndDate*: wie *uacStartDate*, aber definiert Datum und Uhrzeit des Abschlusses der Kampagne oder des geplanten Abschlusses. Muss nach dem *uacStartDate* liegen.
- *uacLastRunDate*: ein optionaler Kalender zur Angabe von Datum und Uhrzeit, an dem/zu der ein an die Kampagne gebundenes Ablaufdiagramm zuletzt ausgeführt wurde (sonst Null).
- *uacExternalLinkOwner*: eine optionale Zeichenfolge, die den Namen des Eigners eines externen Links definiert (siehe das Attribut *uacExternalLinkReference*). Nur bei IBM ; muss eines der folgenden Elemente sein:
  - "Plan" (jetzt bekannt als IBM Marketing Operations)
  - "Collaborate" (jetzt bekannt als IBM Distributed Marketing)
- *uacExternalLinkId*: eine optionale numerische Datenbank-ID, die durch eine andere IBM -Anwendung einem mit dieser Kampagne verknüpften Objekt zugeordnet wurde. Nur bei IBM : siehe dazu auch das Attribut *uacExternalLinkOwner*.

## generateCampaignCode

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

Generiert einen neuen Kampagnencode.

Dieser Code ist garantiert eindeutig und unterscheidet sich von dem Wert, der durch einen vergangenen oder zukünftigen Aufruf dieser Methode oder der Methode `createCampaign()` zurückgegeben wurde/wird oder der für eine Kampagne generiert wurde, die über die grafische IBM Campaign-Benutzeroberfläche 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 Clientbenutzerberechtigungs-nachweis.

*requestedLocale*: optionale Ländereinstellung für diese Anforderung.

*partitionName*: optionaler Name der zu verwendenden Kampagnenpartition. Wenn es nur eine Partition in der Campaign-Installation gibt, kann dieses Argument null sein.

## Rückgabe

Der generierte Kampagnencode.

## Fehler

AuthorizationException, DataException

## createCampaign

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

Erstellt für Client, Partition und securityPolicyName eine neue Kampagne und wendet die angegebenen Attribute an. Alle von dieser API erstellten Kampagnen befinden sich im Stammordner.

## Parameter

*userCredential*: der Clientbenutzerberechtigungs-nachweis.

*requestedLocale*: optionale Ländereinstellung für diese Anforderung.

*partitionName*: optionaler Name der zu verwendenden Kampagnenpartition.

*securityPolicyName*: optionaler Name der für die Erstellung der Kampagne zu verwendenden Richtlinie für Zugriffsberechtigungen der Kampagne. Alle nachfolgenden Operationen für diese Kampagne verwenden dann diese Richtlinie. Wenn diese Option nicht definiert wird, wird die globale Richtlinie verwendet.

*name*: der Name, der der neuen Kampagneninstanz zugeordnet werden soll (ihr Attribut "uacName").

*attributes*: ein Array von Initialisierungsattributen; wenn ein Attribut angegeben wird, werden die Standardwerte der Kampagne überschrieben; andere bleiben unverändert. Wenn z. B. ein Attribut *uacCampaignCode* angegeben wird, wird es anstelle eines automatisch erstellten Attributs verwendet. Es obliegt dem Client festzulegen, welche Attribute für die Kampagne erforderlich sind, ihre Typen festzulegen etc.

Löst *AttributeNotFoundException* aus, wenn eines oder mehrere der angegebenen Attribute nicht vorhanden sind, oder *InvalidAttributeException*, wenn ein Attributwert ungültig ist (z. B. bei einem falschen Datentyp).

## Rückgabe

Eine Einzelinstanz von 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;
```

Listet eine "Seite" mit Kampagnen auf, die mit den optionalen Attributwerten übereinstimmen, angefangen mit dem angegebenen Seitenoffset. Ordner werden ignoriert.

Nach dem Abrufen können alle zurückgegebenen Elemente vom Typ *CampaignInfo* unverändert verwendet werden, z. B. zum Anzeigen einer Übersichtsliste, oder die Attributmethoden können zum Abrufen oder Aktualisieren der Kampagnenattribute verwendet werden.

Der Status wird von dieser API nicht verwaltet, daher können Aufrufe in beliebiger Reihenfolge erfolgen.

## Parameter

*userCredential*: der Clientbenutzerberechtigungsnauchweis.

*requestedLocale*: optionale Ländereinstellung für diese Anforderung.

*partitionName*: optionaler Name der zu verwendenden Kampagnenpartition.

*attributes*: optionales Array von Attributen, mit denen eine Übereinstimmung vorliegen muss; Name, Datentyp und Werte des Attributs werden verwendet, um die Übereinstimmung zu ermitteln; wenn das Attribut Arrays unterstützt, müssen alle angegebenen Werte übereinstimmen. Der implizierte Übereinstimmungsoperator ist AND, d. h., es werden nur Kampagnen zurückgegeben, die mit allen angegebenen Attributwerten übereinstimmen.

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

*pageOffset*: der Anfangsoffset aller möglichen Kampagnen für den Beginn der Auflistung (Wert null). Wenn die Auflistung z. B. mit 1000 Kampagnen übereinstimmt und dieser Wert auf 10 festgelegt ist, würde die Seite mit der elften Komponente beginnen. *RangeException* wird ausgelöst, wenn sich der angegebene Offset außerhalb des gültigen Bereichs befindet.

*pageSize*: die maximale Anzahl passender Kampagnen, die für die Seite zurückgegeben werden soll (maximal 500).

## Rückgabe

Eine typisierte Liste von null oder mehr Datenwrappereinstanzen vom Typ *CampaignInfo*, eine für jede passende Kampagne auf der Seite.

## Fehler

AttributeNotFoundException, InvalidAttributeException, RangeException

InvalidExecutionContextException, AuthorizationException

---

## Zielzellenmethoden

Zielzellen sind eine Abstraktion von bekannten Untergruppen von Kampagnenergebnissen, die von IBM Campaign in Form einer Zielzelltabelle (Target Cell Spreadsheet, TCS) verwaltet werden. Zielzellen können in einer Kampagne global vorkommen oder einem bestimmten Kampagnenablaufdiagramm zugeordnet sein.

Die API unterstützt folgende Operationen für Zielzellen:

- Erstellen einer oder mehrerer neuer, globaler Zielzellen
- Massenaktualisierung einer oder mehrerer bestehender Zielzellen
- Erkennen (Auflisten von Zielzellen)
- Erstellen, Lesen und Aktualisieren von Attributen (über Attribut-APIs)
- Löschen einer bestehenden Zielzelle
- Abrufen von einer oder mehreren Zellen zugeordneten Ausführungsergebnissen

Zielzellen sind einige Standardattribute zugeordnet, die über die API zugänglich sind. Diese Liste kann vom Client beliebig erweitert werden, indem benutzerdefinierte Attributmetadatendefinitionen hinzugefügt werden (siehe Attribut-APIs). Die einzelnen Attributmetadaten können als Spalte in dem Arbeitsblatt für Zielzellen verstanden werden, wobei das Layout der Tabelle dem Client freisteht.

Zu den Standardzielzellenattributen zählen:

- *uacName*: Zellenname.
- *uacDescription*: optionale Zeichenfolge zur Beschreibung des Ablaufdiagramms.
- *uacCellCode*: eine Codezeichenfolge zur eindeutigen Bestimmung der Zelle. Sie wird gewöhnlich automatisch durch Campaign erstellt, kann aber auch vom Client bereitgestellt werden.
- *uacCreateDate*: eine Kalenderinstanz, die Datum und Uhrzeit der Erstellung der Zelle durch den Server angibt.
- *uacUpdateDate*: eine Kalenderinstanz, die angibt, wann die Zelle zuletzt durch den Server aktualisiert wurde.
- *uacIsControl*: ein boolescher Wert, der angibt, ob es sich um eine Kontrollzelle (true) handelt oder nicht (false). Andere Zellen können auf diese Zelle als Kontrollzelle verweisen (siehe *uacControlCell*).
- *uacControlCell*: optionale Referenz der Kontrollzelle (bei Kontrollzellen nicht zulässig). Siehe Attribut *uacIsControl*.
- *uacIsApproved*: ein boolescher Wert, der angibt, ob es sich um eine genehmigte Zelle (true) handelt oder nicht (false).
- *uacIsReadOnly*: ein boolescher Wert, der angibt, ob es sich um eine schreibgeschützte Zelle (true) handelt oder nicht (false).
- *uacDisplayOrder*: eine Ganzzahl, die die Reihenfolge dieser Zelle (Zeile) in Relation zu anderen Zellen in der TCS angibt.
- *uacIsTopDown*: ein boolescher Wert, der angibt, ob es sich um eine Top-down-Zelle handelt.

- *uacAssignedOffers*: ein optionales Array einer oder mehrerer Referenzen zu Angeboten oder Angebotslisten, die dieser Zelle zugeordnet sind (bei Kontrollzellen nicht zulässig).
- *uacFlowchartName*: optionaler Name des Ablaufdiagramms, mit dem diese Zelle verknüpft ist (Schreibschutz muss über die grafische IBM Campaign-Benutzeroberfläche festgelegt werden; bei Kontrollzellen nicht zulässig).
- *uacFlowchartId*: optionale Datenbank-ID des Ablaufdiagramms, mit dem diese Zelle verknüpft ist (zum Schreibschutz siehe oben).

## createTargetCell

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

Erstellt eine neue kampagnenspezifische Zielzellenzeile und wendet die pro Zelle angegebenen Attribute und Benutzerinformationen an.

Die angegebenen Attribute können Standardattribute oder benutzerdefinierte Attribute sein. Wenn es sich um benutzerdefinierte Attribute handelt, müssen die entsprechenden globalen Attributmetadatendefinitionen allerdings bereits vorhanden sein.

Sobald die Zielzelle erstellt wurde, können Attributwerte mithilfe der Attribut-APIs geändert werden.

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

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

### Parameter

*userCredential*: der Clientbenutzerberechtigungsnauchweis.

*requestedLocale*: optionale Ländereinstellung für diese Anforderung.

*partitionName*: optionaler Name der zu verwendenden Kampagnenpartition.

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

*attributes*: optionales Array von TCS-Attributen für die neue Zelle. Jedes angegebene Attributelement überschreibt die entsprechenden Standardwerte der Zellenattribute; andere bleiben unverändert. Es obliegt dem Client festzulegen, welche Attribute für die Zelle erforderlich sind, ihre Typen festzulegen etc. Kumuliert *InvalidAttributeException*, wenn ein Problem mit einem angegebenen Attribut vorliegt.

Wenn Ausnahmebedingungen kumuliert wurden, löst diese Methode die Ausnahmebedingung *CompositeException* aus und alle Erstellungsvorgänge werden rückgängig gemacht. Die Ausnahmebedingungsursachenliste enthält eine Ausnahmebedingung für jedes Attribut, das den Fehler verursacht hat, und darüber hinaus

einen numerischen Index anstelle von *reference*, den Attributnamen und gewöhnlich den unzulässigen Wert. Die Ursachenliste ist gemäß der eingegebenen *attributeList* sortiert.

## Rückgabe

Ein Datenwrapper vom Typ *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 mehrere neue kampagnenspezifische Zielzellen gleichzeitig, wobei die pro Zelle angegebenen Attribute und Benutzerinformationen angewendet werden.

Die angegebenen Attribute können Standardattribute oder benutzerdefinierte Attribute sein. Wenn es sich um benutzerdefinierte Attribute handelt, müssen die entsprechenden globalen Attributmetadaten definitionen allerdings bereits vorhanden sein.

Sobald die Zielzelle erstellt wurde, können Attributwerte mithilfe der Attribut-APIs geändert werden.

Siehe `listTargetCells()`.

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

## Parameter

*userCredential*: der Clientbenutzerberechtigungs-nachweis.

*requestedLocale*: optionale Ländereinstellung für diese Anforderung.

*partitionName*: optionaler Name der zu verwendenden Kampagnenpartition.

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

*attributeList*: optionale Liste von Attributarrays pro Zelle, eine für jede zu erstellende Zielzellenreihe. Jedes für ein bestimmtes Listenelement angegebene Attribut überschreibt die entsprechenden Standardwerte der Zellenattribute; andere bleiben unverändert. Es obliegt dem Client festzulegen, welche Attribute für die Zelle erforderlich sind, ihre Typen festzulegen etc. Kumuliert *InvalidAttributeException*, wenn ein Problem mit einem angegebenen Attribut vorliegt.

Wenn Ausnahmebedingungen kumuliert wurden, löst diese Methode die Ausnahmebedingung *CompositeException* aus und alle Erstellungsvorgänge werden rückgängig gemacht. Die Ausnahmebedingungsursachenliste enthält eine Ausnahmebe-

dingung für jedes Attribut, das den Fehler verursacht hat, und darüber hinaus einen numerischen Index anstelle von *reference*, den Attributnamen etc. Die Ursachenliste ist gemäß der eingegebenen *attributeList* sortiert.

### Rückgabe

Eine Liste von Datenwrappern vom Typ *TargetCellInfo*, einer für jede erstellte Instanz, sortiert gemäß der Elementreihenfolge der eingegebenen *attributesList*-Parameter.

### Fehler

*InvalidComponentException*, *CompositeException*

*AuthorizationException*, *DataException*

## listTargetCells

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

Listet Informationen zu allen Zielzellen auf, die gegenwärtig vorhanden sind und mit den angegebenen Attributen übereinstimmen, entweder für die angegebene Kampagne, oder global, wenn keine Kampagne angegeben wurde.

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

### Parameter

*userCredential*: der Clientbenutzerberechtigungs-nachweis.

*requestedLocale*: optionale Ländereinstellung für diese Anforderung.

*partitionName*: optionaler Name der zu verwendenden Kampagnenpartition.

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

*attributes*: optionales Array von Attributen, mit denen eine Übereinstimmung vorliegen muss. Der implizierte Übereinstimmungsoperator ist AND, d. h., es werden nur Zellen zurückgegeben, die mit allen angegebenen Attributwerten übereinstimmen.

Löst *InvalidAttributeException* aus, wenn angegebene Attribute ungültig sind.

### Rückgabe

Gibt eine Liste von null oder mehr Instanzen vom Typ *TargetCellInfo* für die passenden Zellen zurück.

### Fehler

*InvalidComponentException*, *InvalidAttributeException*

*AuthorizationException*, *DataException*

## bulkUpdateTargetCells

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

Aktualisiert die Attribute einer oder mehrerer Zielzellen.

Die Aktualisierungslogik ist wie folgt.

Für jedes Element in der bereitgestellten *attributesMap* ist der Eintragungsschlüssel die Referenz der zu aktualisierenden Zielzelle und der Eintragswert ist ein Array von Aktualisierungsattributen für diese Zelle. Wenn die Zielzelle nicht vorhanden ist, wird eine Ausnahmebedingung vom Typ *InvalidComponentException* kumuliert.

Sobald eine Zielzelle für jedes angegebene Attribut lokalisiert wurde, wird Folgendes ausgeführt:

1. Wenn der Attributname mit einem bestehenden Attribut übereinstimmt, wird versucht, sein Wertfeld mit dem bereitgestellten Wertfeld zu überschreiben.
2. Wenn der Wertetyp oder ein anderer Aspekt der Attributmetadatendefinition nicht erfüllt wird oder einer oder mehrere der bereitgestellten Werte ungültig, außerhalb des gültigen Bereichs o. Ä. sind, wird *InvalidAttributeException* kumuliert.
3. Andernfalls wird *AttributeNotFoundException* kumuliert, wenn das genannte Attribut nicht vorhanden ist.

Wenn Ausnahmebedingungen kumuliert wurden, löst diese Methode die Ausnahmebedingung *CompositeException* aus und alle Aktualisierungen werden rückgängig gemacht. In der Ursachenliste der Ausnahmebedingung werden die oben aufgeführten Ausnahmebedingungen aufgelistet. Zu jedem Attribut, das den Fehler verursacht hat, werden Referenz und Attributname erfasst.

In jedem Fall gelten für Attributaktualisierungsoperationen die üblichen Integritätsbedingungen für die Sicherheit und Gültigkeitsprüfungen. Es liegt in der Zuständigkeit des Clients zu bestimmen, welche Attribute für eine bestimmte Komponenteinstanz erforderlich sind, welche Typen korrekt sind etc.

### Parameter

*userCredential*: der Clientbenutzerberechtigungs-nachweis.

*requestedLocale*: optionale Ländereinstellung für diese Anforderung.

*partitionName*: optionaler Name der zu verwendenden Kampagnenpartition.

*attributesMap*: eine Zuordnung von zu aktualisierenden Zielzellen; der Eintragungsschlüssel ist die Referenz der zu aktualisierenden Zelle und der Eintragswert ist ein Array von Aktualisierungsattributen. Zum Lokalisieren des zu aktualisierenden Attributs wird der Attributname verwendet und die neuen Attributwerte werden zum Aktualisieren der bestehenden Attributwerte verwendet, als einzelnes Objekt des ordnungsgemäßen Typs oder als Array, falls zutreffend. Siehe die Ausnahmebedingungen oben.

### Rückgabe

Keine Rückgabe.

## Fehler

ComponentException

AuthorizationException, DataException

## getRunResultsByCell

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

Ruft die Ausführungsergebnisse einer oder mehrerer Zielzellen ab, möglicherweise für ein nie gestartetes Ablaufdiagramm, oder befindet sich noch in Bearbeitung.

### Parameter

*userCredential*: der Clientbenutzerberechtigungsnauchweis.

*requestedLocale*: optionale Ländereinstellung für diese Anforderung.

*partitionName*: optionaler Name der zu verwendenden Kampagnenpartition.

*cellReferences*: ein Array von Referenzen der Zielzellen, deren Ausführungsergebnisse gewünscht werden. Löst *InvalidComponentException* aus, wenn eine genannte Zellenreferenz ungültig ist oder auf eine nicht vorhandene Zelle verweist.

### Rückgabe

Gibt eine typisierte Liste von Ausführungsergebnissen für die genannte Zelle zurück, sortiert gemäß dem eingegebenen Referenzarray.

Der Ausführungsfortschritt ist *RUNNING*, wenn das zugrunde liegende Ablaufdiagrammprozessfeld noch aktiv ist, *FAILED*, wenn die Ausführung aus irgendeinem Grund fehlgeschlagen ist, oder *NOT\_STARTED*, wenn die Ausführung des Prozessfeldes noch nicht gestartet wurde. Statusdetails werden auch bereitgestellt.

## Fehler

InvalidComponentException

AuthorizationException, DataException

## bulkDeleteTargetCells

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

Löscht eine oder mehrere bestehende Zielzellen und alle abhängigen Komponenten (also Ablaufdiagramme, Verknüpfungen, Attribute etc.).

### Parameter

*userCredential*: der Clientbenutzerberechtigungsnauchweis.

*requestedLocale*: optionale Ländereinstellung für diese Anforderung.

*partitionName*: optionaler Name der zu verwendenden Kampagnenpartition.

*cellReferences*: ein Array von einer oder mehreren Referenzen von zu löschenden Zellen. *InvalidComponentException* wird kumuliert, wenn ein Problem mit einer angegebenen Referenz vorliegt oder eine Zelle nicht vorhanden ist.

Wenn Ausnahmebedingungen kumuliert wurden, löst diese Methode die Ausnahmebedingung *CompositeException* aus und alle Löschvorgänge werden rückgängig gemacht. In der Ursachenliste der Ausnahmebedingung werden die oben aufgeführten Ausnahmebedingungen aufgelistet. Zu jeder Zelle, die den Fehler verursacht hat, wird die Referenz erfasst.

### Rückgabe

Keine Rückgabe.

### Fehler

*CompositeException*

*AuthorizationException*, *DataException*

---

## Analyse

Die API unterstützt das Abrufen einfacher Metriken von IBM Campaign.

### getCampaignMetrics

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

Ruft die Metriken für die angegebene Kampagne ab.

### Parameter

*userCredential*: der Clientbenutzerberechtigungsnauchweis.

*requestedLocale*: optionale Ländereinstellung für diese Anforderung.

*partitionName*: optionaler Name der zu verwendenden Kampagnenpartition.

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

### Rückgabe

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

### Fehler

*InvalidComponentException*

*AuthorizationException*, *DataException*

---

## Methoden für Angebot, Angebotsliste und Angebotsvorlage

Die API unterstützt folgende Operationen für Angebote:

- Erkennen (Auflisten nach Ordner (Angebot, Angebotsliste und Unterordner), Attribut (Angebote und Angebotsvorlagen) oder Suchwert (Angebote))
- Überprüfen

- Abrufen von Informationen (Abrufen von Attributen für ein bestimmtes Angebot oder eine bestimmte Angebotsvorlage)
- Erstellen, Bearbeiten, Außerkraftsetzen und Löschen von Angeboten

Angeboten sind eine Reihe von Standardattributen zugeordnet. Diese Liste kann vom Client beliebig erweitert werden, indem benutzerdefinierte Attributmetadaten-Definitionen hinzugefügt werden (siehe Attribut-APIs).

Zu den Standardangebotsattributen gehören:

- *uacName*: Angebotsname.
- *uacDescription*: optionale Zeichenfolge zur Beschreibung des Angebots.
- *uacOfferCode*: eine Codezeichenfolge zur eindeutigen Bestimmung des Angebots. Sie wird gewöhnlich automatisch durch IBM Campaign erstellt, kann aber auch vom Client bereitgestellt werden.
- *uacCreateDate*: eine Kalenderinstanz, die Datum und Uhrzeit der Erstellung des Angebots durch den Server angibt.
- *uacUpdateDate*: eine Kalenderinstanz, die angibt, wann das Angebot zuletzt durch den Server aktualisiert wurde.

Angebotsvorlagen haben auch Standardattribute und benutzerdefinierte Attribute. Zu den Standardangebotsvorlagenattributen gehören:

- *uacName*: Angebotsvorlagenname.
- *uacDescription*: optionale Zeichenfolge zur Beschreibung der Angebotsvorlage.
- *uacCreateDate*: eine Kalenderinstanz, die Datum und Uhrzeit der Erstellung der Angebotsvorlage durch den Server angibt.
- *uacUpdateDate*: eine Kalenderinstanz, die angibt, wann die Angebotsvorlage zuletzt durch den Server aktualisiert wurde.

## listOffersAndFolders

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

Listet alle Angebote, Angebotslisten und Ordner im optionalen übergeordneten Ordner auf.

Nach dem Abrufen können alle *WSComponentOrFolderInfo*-Instanzen unverändert verwendet werden, z. B. zum Anzeigen der nächsten Ebene der Ordnerhierarchie. Die Attribut-APIs können zum Abrufen oder Aktualisieren der enthaltenen Angebote verwendet werden.

### Parameter

*userCredential*: der Clientbenutzerberechtigungs-nachweis.

*requestedLocale*: optionale Ländereinstellung für diese Anforderung.

*partitionName*: optionaler Name der zu verwendenden Kampagnenpartition.

*parentReference*: optionale Referenz des aufzulistenden Stammordners. Nur die direkt untergeordneten Angebote, Angebotslisten und Ordner dieses Stammordners werden aufgelistet, daher sind aufeinanderfolgende Aufrufe dieser API erforderlich, um die gesamte Ordnerhierarchie zu navigieren (sie ist aber meist sehr flach).

Wenn kein übergeordnetes Element angegeben wird, werden alle Komponenten und Ordner des Stammverzeichnisses zurückgegeben.

Löst *InvalidFolderException* aus, wenn ein Problem mit der Referenz des angegebenen Stammordners besteht.

Eine typisierte *Liste* von null oder mehr Datenwrapperinstanzen vom Typ *WSComponentOrFolderInfo*, eine für jede passende Komponente oder jeden passenden 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;
```

Listet eine "Seite" mit Angeboten auf, die das genannte Suchkriterium in den Feldern Name, Beschreibung, "Erstellt durch" oder Angebotscode enthalten, angefangen mit dem angegebenen Seitenoffset. Die Suche basiert auf der optionalen Ordnerangabe. (Wenn für Ordner-ID der Wert 0 angegeben wird, wird standardmäßig der Stammangebotsordner verwendet.) Übereinstimmungen werden zurückgegeben, wenn der Suchbegriff enthalten ist.

Nach dem Abrufen können alle zurückgegebenen Wrapper vom Typ *WSOfferInfo* unverändert verwendet werden, z. B. zum Anzeigen einer Übersichtsliste, oder die Attributmethoden können zum Abrufen oder Aktualisieren der Angebotsattribute verwendet werden.

Der Status wird von dieser API nicht verwaltet, daher können Aufrufe in beliebiger Reihenfolge erfolgen.

## Parameter

*userCredential*: der Clientbenutzerberechtigungsnauchweis.

*requestedLocale*: optionale Ländereinstellung für diese Anforderung.

*partitionName*: optionaler Name der zu verwendenden Kampagnenpartition.

*folderID*: die ID des zu durchsuchenden Angebotsordners; wenn eine *folderID* von 0 angegeben wird, wird der Stammordner durchsucht.

*searchCriteria*: der Suchausdruck

*includeRetired*: der boolesche Wert, der angibt, ob gesperrte Angebote in die Suchergebnisse einbezogen werden. Gültige Werte sind TRUE und FALSE, wobei TRUE angibt, dass gesperrte Angebote einbezogen werden, und FALSE, dass sie nicht einbezogen werden.

*pageOffset*: der Anfangsoffset aller möglichen Komponenten für den Beginn der Auflistung (Wert null). Wenn die Auflistung z. B. mit 1000 Angeboten übereinstimmt und dieser Wert auf 10 festgelegt ist, würde die Seite mit der elften Komponente beginnen. *RangeException* wird ausgelöst, wenn sich der angegebene Offset außerhalb des gültigen Bereichs befindet.

*pageSize*: die maximale Anzahl passender Komponenten, die für die Seite zurückgegeben werden soll (maximal 500).

## Rückgabe

Gibt eine typisierte Liste von null oder mehr Datenwrapperinstanzen vom Typ *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;
```

Listet eine "Seite" mit Angeboten auf, die mit den optionalen Attributwerten übereinstimmen, angefangen mit dem angegebenen Seitenoffset. Ordner werden ignoriert. Übereinstimmungen werden auf Basis eines "like"-Abgleichs zurückgegeben (wobei die Übereinstimmung als ausreichend erachtet wird, wenn eine Zeichenfolge den abgefragten Begriff enthält) und exakte Übereinstimmungen für Daten und Zahlen.

Nach dem Abrufen können alle zurückgegebenen Elemente vom Typ *OfferInfo* unverändert verwendet werden, z. B. zum Anzeigen einer Übersichtsliste, oder die Attributmethoden können zum Abrufen oder Aktualisieren der Angebotsattribute verwendet werden.

Der Status wird von dieser API nicht verwaltet, daher können Aufrufe in beliebiger Reihenfolge erfolgen.

## Parameter

*userCredential*: der Clientbenutzerberechtigungs-nachweis.

*requestedLocale*: optionale Ländereinstellung für diese Anforderung.

*partitionName*: optionaler Name der zu verwendenden Kampagnenpartition.

*attributes*: optionales Array von Attributen, mit denen eine Übereinstimmung vorliegen muss; Name, Datentyp und Werte des Attributs werden verwendet, um die Übereinstimmung zu ermitteln; wenn das Attribut Arrays unterstützt, müssen alle angegebenen Werte übereinstimmen. Der implizierte Übereinstimmungsoperator ist OR, d. h., es werden alle Komponenten zurückgegeben, die mit einem beliebigen angegebenen Attributwert übereinstimmen.

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

*pageOffset*: der Anfangsoffset aller möglichen Komponenten für den Beginn der Auflistung (Wert null). Wenn die Auflistung z. B. mit 1000 Angeboten übereinstimmt und dieser Wert auf 10 festgelegt ist, würde die Seite mit der elften Komponente beginnen. *RangeException* wird ausgelöst, wenn sich der angegebene Offset außerhalb des gültigen Bereichs befindet.

*pageSize*: die maximale Anzahl passender Komponenten, die für die Seite zurückgegeben werden soll (maximal 500).

## Rückgabe

Eine typisierte Liste von null oder mehr Datenwrapperinstanzen vom Typ *OfferInfo*, eine für jede passende Komponente auf der Seite.

## Fehler

*AttributeNotFoundException*, *InvalidAttributeException*, *RangeException*

*InvalidExecutionContextException*, *AuthorizationException*

## validateOffers

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

Überprüft die bereitgestellten Angebotscodes oder Angebotslistenamen und gibt zu jedem einzelnen Element die Überprüfungsergebnisse zurück. Bei der "Überprüfung" wird geprüft, ob tatsächlich nur ein einziges übereinstimmendes Angebot bzw. eine einzige übereinstimmende Angebotsliste in der Datenbank vorhanden ist.

Das Objekt *OfferValidationInfo* enthält eine Fehlermeldung anstelle der Angebotsinformationen, wenn keine Angebote oder Angebotslisten gefunden werden, die mit dem angegebenen Code oder Namen übereinstimmen. Ein Fehler wird auch anstelle einer Übereinstimmung zurückgegeben, wenn der angegebene Code oder Name mit mehreren Angeboten oder Angebotslisten übereinstimmt. Die Liste wird in der angegebenen Reihenfolge zurückgegeben. Angebotscodes und Angebotslistenamen werden auf Basis einer exakten Übereinstimmung mit den Angeboten überprüft.

## Parameter

*userCredential*: der Clientbenutzerberechtigungsbeleg.

*requestedLocale*: optionale Ländereinstellung für diese Anforderung.

*partitionName*: optionaler Name der zu verwendenden Kampagnenpartition.

*codeOrNames*: ein Array aller zu überprüfender Angebotscodes oder Angebotslistenamen.

**Anmerkung:** Von dieser Methode werden keine Ausnahmeregelungen ausgelöst; stattdessen werden Überprüfungsergebnisse zu allen bereitgestellten Codes oder Namen zurückgegeben.

## Rückgabe

Eine typisierte Liste von null oder mehr Datenwrapperinstanzen vom Typ *OfferValidationInfo*.

## Fehler

Keine.

## 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;
```

Erstellt für den Client eine neue Kampagne und wendet die angegebenen Attribute an.

### Parameter

*authorizationLoginName*: der Name des Benutzers, der das Angebot erstellt. Benutzer müssen über die Berechtigung "Angebote hinzufügen" verfügen, um diese Methode verwenden zu können.

*partitionName*: optionaler Name der zu verwendenden Kampagnenpartition.

*requestedLocale*: optionale Ländereinstellung für diese Anforderung.

*securityPolicyName*: optionaler Name der für die Erstellung des Angebots zu verwendenden Richtlinie für Zugriffsberechtigungen der Kampagne. Alle nachfolgenden Operationen für dieses Angebot verwenden dann diese Richtlinie. Wenn diese Option nicht definiert wird, wird die *globale* Richtlinie verwendet.

*name*: der Name, der der neuen Angebotsinstanz zugeordnet werden soll (ihr Attribut *uacName*).

*folderID*: die ID des Angebotsordners, in dem das Angebot erstellt wird. Diese ID wird auf Fehlerfreiheit geprüft und es wird eine Ausnahmereingung ausgelöst, wenn die ID ungültig ist.

*templateName*: erforderlicher (eindeutiger) Name einer bestehenden Angebotsvorlage, die für das neue Angebot verwendet werden soll.

*wsAttributes*: ein Array von Initialisierungsattributen; wenn ein Attribut angegeben wird, werden die Standardwerte des Angebots überschrieben; andere bleiben unverändert. Wenn z. B. ein Attribut *uacOfferCode* angegeben wird, wird es anstelle eines automatisch erstellten Attributs verwendet. Es obliegt dem Client festzulegen, welche Attribute für das Angebot erforderlich sind, seine Typen festzulegen etc.

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

- Der Parameter *folderID* ist ungültig (nicht vorhanden oder nicht vom Typ Angebot).
- Der Benutzer ist nicht berechtigt, diese Operation auszuführen.
- In *wsAttributes* sind ungültige Attribute angegeben.

- Andere Laufzeitausnahmebedingungen treten ein.

### **Rückgabe**

Eine Einzelinstanz von *OfferInfo* für das erstellte Angebot.

### **Fehler**

CampaignServicesException

## **retireOffers**

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

Setzt ein oder mehrere Angebote außer Kraft.

### **Parameter**

*userCredential*: der Clientbenutzerberechtigungs-nachweis.

*requestedLocale*: optionale Ländereinstellung für diese Anforderung.

*partitionName*: optionaler Name der zu verwendenden Kampagnenpartition.

*references*: ein Array von Referenzen der zu sperrenden Angebote. *InvalidComponentException* wird ausgelöst, wenn ein Problem mit einer angegebenen Referenz vorliegt oder ein Angebot nicht vorhanden ist.

### **Rückgabe**

Keine Rückgabe.

### **Fehler**

InvalidComponentException

AuthorizationException, DataException

## **deleteOffers**

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

Löscht ein oder mehrere bestehende Angebote.

### **Parameter**

*userCredential*: der Clientbenutzerberechtigungs-nachweis.

*requestedLocale*: optionale Ländereinstellung für diese Anforderung.

*partitionName*: optionaler Name der zu verwendenden Kampagnenpartition.

*reference*: ein Array von Referenzen der zu löschenden Angebote. *InvalidComponentException* wird ausgelöst, wenn ein Problem mit einer angegebenen Referenz vorliegt oder ein Angebot nicht vorhanden ist.

### **Rückgabe**

Keine Rückgabe.

## Fehler

InvalidComponentException

AuthorizationException, DataException

## listOfferTemplates

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

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

Nach dem Abrufen können alle *WSOfferTemplateInfo*-Instanzen unverändert verwendet werden oder es können mithilfe einer oder mehrerer Attribut-APIs die aufgelisteten Vorlagen abgerufen oder aktualisiert werden.

### Parameter

*userCredential*: der Clientbenutzerberechtigungs-nachweis.

*requestedLocale*: optionale Ländereinstellung für diese Anforderung.

*partitionName*: optionaler Name der zu verwendenden Kampagnenpartition.

### Rückgabe

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

## Fehler

InvalidExecutionContextException, AuthorizationException

DataException

## bulkCreateOffers

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

Erstellt Angebote als Massenausfertigung, wobei die Attribute zu jedem Angebot im Parameter *offers* angegeben werden. Alle Angebote werden unter der angegebenen *folderID* mit dem angegebenen *templateName* erstellt.

### Parameter

*authorizationLoginName*: der Clientbenutzerberechtigungs-nachweis.

*partitionName*: optionaler Name der zu verwendenden Kampagnenpartition.

*requestedLocale*: optionale Ländereinstellung für diese Anforderung.

*securityPolicyName*: optionaler Name der für die Erstellung des Angebots zu verwendenden Richtlinie für Zugriffsberechtigungen der Kampagne. Wenn diese Option nicht definiert wird, 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 das Angebot erstellt wird. Diese ID wird überprüft und es wird eine Ausnahmebedingung ausgelöst, wenn die ID ungültig ist.

*offers*: ein Array von *WSBulkOfferInfo*-Objekten, das den Angebotsnamen und Attribute definiert. Weitere Details hierzu finden Sie im Abschnitt zum Datentyp *WSBulkOfferInfo*.

### **Rückgabe**

Ein Array von *WSOfferInfoStatus*-Instanzen für jedes Angebot. Enthält den Status und Informationen zum Angebot. Der Status gibt an, ob die Erstellung des Angebots erfolgreich war oder nicht.

### **Fehler**

*CampaignServicesException*



---

## Kapitel 5. Verwenden der API

In diesem Abschnitt wird die Verwendung der Campaign-Web-Services-API beschrieben. Außerdem wird in einem Beispiel gezeigt, wie mit dem Campaign-API-Service ein Angebot in Campaign erstellt werden kann.

Zur Verwendung der Campaign-Services-API gibt es zwei Methoden:

- „Verwenden der Client-API-JAR-Datei“
- „Verwenden der WSDL“ auf Seite 49

---

### Verwenden der Client-API-JAR-Datei

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

Diese JAR-Datei befindet sich unter:

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

Im folgenden Beispiel wird beschrieben, wie in Campaign ein neues Angebot im Stammangebotsordner erstellt wird. Dasselbe Beispiel befindet sich unter:

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

**Anmerkung:** In diesem Beispiel werden Testwerte für die Parameter verwendet; Ihre tatsächlichen Werte können sich davon unterscheiden.

Beachten Sie außerdem, dass die URL der Campaign-Web-Services `http://host:port/Campaign/services/CampaignServices30Service` ist, wobei sich Host und Port auf den Hostnamen und die Portnummer der Maschine beziehen, auf dem die Campaign-Webanwendung implementiert ist.

Wenn Sie ein bereitgestelltes Beispiel verwenden, stellen Sie sicher, dass Sie es entsprechend an Ihre Clientumgebung anpassen.

### Code von OfferAPI.java

```
import java.net.URL;  
import java.util.Locale;  
import com.unica.publicapi.campaign.campaignservices.CampaignServicesException;  
import com.unica.publicapi.campaign.campaignservices.attribute.metadata.  
    IAttribute Metadata;  
import com.unica.publicapi.campaign.campaignservices.soap.v30.  
    CampaignServices30SoapClient;  
import com.unica.publicapi.campaign.campaignservices.soap.v30.WSAttribute;  
import com.unica.publicapi.campaign.campaignservices.soap.v30.WSOfferInfo;  
import com.unica.publicapi.campaign.campaignservices.utils.WSAttributeUtils;  
  
/**  
 * Dies ist die Beispiel-Java-Client-Klasse, die die Syntax der Campaign-SOAP-  
 * Services-API angibt.  
 * In diesem Beispiel wird die CampaignServices30SoapClient-Fassade zur Interaktion  
 * mit dem Campaign-Web-Service verwendet. */
```

```

* Hier wird die Erstellung des Angebots gezeigt. Weitere Informationen finden Sie
im API-Handbuch.
*
* @author AGijare
*
*/
public class OfferAPI {

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

    private static void setup(){
        try {
            String protocol = "http"; //http oder https
            String host = "localhost"; //Hostname der implementierten Campaign-Version.
            Hier den ordnungsgemäßen Hostnamen verwenden.
            int port = 7001; //Portnummer der implementierten Campaign-Version
            long timeOut = 2*60*1000; // 2 Minuten
            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) {
        //Passen Sie die Werte der folgenden Variablen an Ihre
        Umgebung an.
        String userName = "user_name"; //Benutzername für die Anmeldung
        String partitionName = "partition1"; //Ordnungsgemäßen Partitionsnamen für
        Campaign verwenden
        Locale loc = Locale.US;
        String securityPolicy = "Global"; //Verwenden Sie Ihre Richtlinie für
        Zugriffsberechtigungen für Campaign

        String offerName = "Offer1";
        String offerTemplate = "Offer Template"; // Vorlage, die zur Erstellung des
        Angebots verwendet wird.
        long folderID = 1002; //Tatsächliche ID des Ordners, in dem das Angebot
        erstellt wird. Bei folderID <=0 wird das Angebot auf Stammebene erstellt.
        //Attribute des Angebots
        WSAttribute[] wsAttributes = {
            WSAttributeUtils.getWSTextAttribute(IAttributeMeta
            data.AC_OFFER_DESCRIPTION_ATTRIBUTE_NAME, null, new String[]{"description "
            + System.currentTimeMillis()})
        };

        setup();

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

```

**Anmerkung:** Um das oben angegebene Java-Beispiel zu kompilieren und auszuführen, müssen alle abhängigen JAR-Dateien im Java-Klassenpfad enthalten sein.

Die Datei CampaignServicesClient30.jar hängt von den JAR-Dateien der Apache Axis2-SOAP-Engine und weiteren allgemeinen Apache-JAR-Dateien ab, die in der Datei Campaign.war unter <CAMPAIGN\_HOME>/Campaign.war zu finden sind. Extrahieren Sie die JAR-Dateien aus Campaign.war und fügen Sie sie dem Java-Klassenpfad hinzu.

---

## Verwenden der WSDL

Campaign-Services können auch mithilfe der Campaign-Web-Services-WSDL-Datei aufgerufen werden:

CampaignServices30.wsdl

die sich hier befindet:

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

oder in der Campaign-Verteilung unter:

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

Die Client-Java-Anwendung benötigt zur Verwendung der von der WSDL erstellten Klassen und Stubs ein beliebiges WSDL-zu-Java-Konvertierungstool eines Drittherstellers. IBM empfiehlt die Verwendung von Apache Axis.

Die Javadocs, die aus den Klassen und Stubs erstellt wurden, die mithilfe der WSDL unter Verwendung von Apache Axis2 generiert wurden, befinden sich unter:

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

**Anmerkung:** Alle abhängigen JAR-Dateien müssen im Java-Klassenpfad einbezogen sein. Die Datei CampaignServicesClient30.jar ist abhängig von Apache Axis2-SOAP-Engine-JAR-Dateien und anderen allgemeinen Apache-JAR-Dateien, die sich in der Datei Campaign.war unter <CAMPAIGN\_HOME>/Campaign.war befinden. Extrahieren Sie die JAR-Dateien aus Campaign.war und fügen Sie sie dem Java-Klassenpfad hinzu.

---

## Leistungsaspekte

Das aktuelle Leistungsprofil der CampaignServices-API-Implementierung ähnelt dem, das über die grafische Benutzeroberfläche bereitgestellt wird. Einige APIs sind speziell für die Leistung ausgelegt. Besonders die `listCampaignsByPage()`-API ermöglicht eine relativ effiziente Paginierung.

Der SOAP-Benutzeroberfläche ist es zu eigen, dass sie Latenzzeiten und Aufwand verursacht, da alle Daten in XML konvertiert werden, was in einigen Fällen recht ausführlich sein kann. So kann z. B. ein einfacher Loopback-SOAP-Aufruf in einem Standardnetz 100 ms dauern (Java 1.4.x war sogar noch langsamer). Die API wurde für typische Portal- und andere Clientanwendungen von Unternehmen optimiert, z. B. see `listOffersByPage()`, daher sollte die SOAP-Leistung angemessen sein.

Der Client muss aber darauf achten, dass keine zu hohe Belastung der für die Webbenutzeranforderungen verantwortlichen CampaignServices-Services entsteht. Generell wird davon ausgegangen, dass der Kapazitätsbedarf eines API-Benutzers nicht den eines typischen IBM Campaign-Webbenutzers überschreitet.

---

## Paketierung

Diese Spezifikation wird zusammen mit dem CampaignServices-Software-Development-Kit (devkits) bereitgestellt, das zusammen mit IBM Campaign installiert wird.

Das vom Installationsprogramm erstellte devkits-Verzeichnis enthält Beispiele, Erstellungs- und Textscripts, Javadoc für öffentliche Klassen und Benutzeroberflächen, Releaseinformationen etc.

---

## Kontakt zum technischen Support von IBM

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. Damit wir Ihnen möglichst schnell helfen können, beachten Sie dabei bitte die Informationen in diesem Abschnitt.

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

### 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 zu der installierten IBM -Anwendung.

Sie können über **Hilfe > Info** (Help > About) auf die Info-Seite zugreifen. Wenn Sie nicht auf die Info-Seite zugreifen können, finden Sie die Versionsnummer der IBM -Anwendung in der Datei `version.txt` im Installationsverzeichnis jeder Anwendung. Wenn die Datei `version.txt` nicht verfügbar ist, verwenden Sie den Befehl `NetTracker admin -buildinfo`, um die Versionsinformationen abzurufen.

### 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.unica.com/about/product-technical-support.htm>).



---

## 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 zu den gegenwärtig im jeweiligen Land verfügbaren Produkten und Services erhalten Sie beim zuständigen IBM Ansprechpartner. 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 in diesem Handbuch beschriebene 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  
170 Tracer Lane  
Waltham, MA 02451  
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 Musterprogramme 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 Musterprogramme 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 Beispielprogramme entstehen.

---

## Marken

IBM, das IBM Logo und [ibm.com](http://ibm.com) sind Marken oder eingetragene Marken der IBM Corporation. Weitere Produkt- und Servicenamen 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 Software-Produkte, einschließlich Software as a Service-Lösungen, ("Softwareangebote") verwenden möglicherweise Cookies oder andere Technologien, um Informationen zur Produktverwendung zu erfassen, mit deren Hilfe die Benutzerfreundlichkeit verbessert werden kann, die Möglichkeiten der Interaktion mit dem Benutzer angepasst oder andere Zwecke verfolgt werden können. 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 implementierten 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-Beacons 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-Beacons 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".



