

Unica Campaign API-Handbuch



Inhalt

Kapitel 1. Unica Campaign – Rest-APIs.....	1
Kapitel 2. Übersicht über die SOAP-API von Unica Campaign.....	2
Übersicht über den SOAP-API-Entwurf.....	2
SOAP-API-Änderungen nach Version.....	4
Bestehende Implementierungen, die die JAR-Datei der Client-API verwenden.....	7
Bestehende Implementierungen, die WSDL direkt verwenden.....	8
SOAP-API-Referenzen.....	12
Kapitel 3. Verwenden der SOAP-API von Unica Campaign.....	13
Verwenden der JAR-Datei der Client-API zum Aufrufen der Unica Campaign- Services.....	13
OfferAPI.java.....	13
Verwenden der WSDL zum Aufrufen der Unica Campaign-Services.....	16
Leistungsüberlegungen.....	17
Kapitel 4. SOAP-API-Datentypen.....	19
WSReference.....	19
WSVersion.....	19
WSServiceInfo.....	20
WSAttributeTypeEnum.....	20
WSAttributeStatusEnum.....	20
WSAccessTypeEnum.....	21
WSSelectTypeEnum.....	21
WSRunStatusEnum.....	21
WSRunTypeEnum.....	22

WSAttribute.....	22
WSAttributeMetadata.....	24
WSCampaignInfo.....	27
WSComponentOrFolderInfo.....	27
WSTargetCellInfo.....	28
WSMetricsInfo.....	28
WSRunResults.....	29
WSOfferInfo.....	30
WSOfferCodeOrName.....	30
WSOfferValidationInfo.....	30
WSOfferTemplateInfo.....	31
WSBulkOfferInfo.....	31
WSOfferInfoStatus.....	31
Kapitel 5. SOAP-API-Methoden.....	33
SOAP API Methoden: Service.....	33
getServiceInfo.....	33
SOAP API Methoden: Attribute.....	33
getAttributesByName.....	34
updateAttributes.....	35
getAttributeMetadataByName.....	37
createAttributeMetadata.....	38
updateAttributeMetadata.....	39
deleteAttributeMetadata.....	42
SOAP API Methoden: Kampagnen und Ablaufdiagramme.....	43
generateCampaignCode.....	45

deleteCampaigns.....	46
createCampaign.....	46
listCampaignsByPage.....	49
stopFlowchart.....	50
SOAP API Methoden: Zielzellen.....	51
createTargetCell.....	53
bulkCreateTargetCells.....	54
listTargetCells.....	56
bulkUpdateTargetCells.....	57
getRunResultsByCell.....	59
bulkDeleteTargetCells.....	60
updateTemplateAttributes.....	61
listBottomUpTargetCells.....	62
SOAP API Methoden: Analyse.....	62
getCampaignMetrics.....	63
SOAP API Methoden: Angebote, Angebotslisten, Angebotsvorlagen.....	63
listOffersAndFolders.....	65
searchOffersBasic.....	66
listOffersByPage.....	67
createSmartOfferList.....	69
createStaticOfferList.....	70
getOffers.....	71
validateOffers.....	72
editOfferList.....	73
createOffer.....	74

retireOffers.....	76
deleteOffers.....	77
deleteOffersAndLists.....	78
listOfferTemplates.....	78
createTemplate.....	79
getOfferTemplate.....	80
retireOfferTemplates.....	81
getOffersAndListsByPage.....	81
bulkCreateOffers.....	82
getOfferListDetails.....	84
getOfferListMembers.....	84
getOffersByQuery.....	85
retireOfferLists.....	86
createFolder.....	86
editFolder.....	88
getSubFoldersList.....	89
moveFolders.....	89
deleteFolders.....	90
Kapitel 6. Allgemeine SOAP-API-Ausnahmen.....	92
Kapitel 7. Index.....	

Kapitel 1. Unica Campaign – Rest-APIs

Unica Campaign bietet Rest-APIs für verschiedene Operationen im Zusammenhang mit Campaign, Angeboten, Ablaufdiagrammen usw. Weitere Einzelheiten zu den Campaign Rest APIs finden Sie in der Swagger-Dokumentation. Um auf die Swagger-Dokumentation zuzugreifen, führen Sie die folgenden Schritte aus.

- Melden Sie sich bei der Unica-Anwendung an.
- Navigieren Sie zu **Einstellungen > Campaign-Einstellungen**.
- Klicken Sie auf **API-Dokumentation**.

Campaign besitzt mehrere APIs. Weitere Einzelheiten finden Sie unter folgendem Link.

https://s3.amazonaws.com/help.hcltechsw.com/unica/Campaign/9.1.2/Campaign/REST_API/RESTAPI_parent.html

Verwenden Sie Platform-Benutzer für die mit Campaign Rest APIs zu verwendende Authentifizierung. Die Benutzerauthentifizierung über Verzeichnisdienste von Drittanbietern wird bei den APIs von Campaign oder Optimize nicht unterstützt.

Kapitel 2. Übersicht über die SOAP-API von Unica Campaign

In der Spezifikation der SOAP-API von Unica Campaign wird Version 3.0 der Anwendungsprogrammierschnittstelle (Application Programming Interface, kurz API), auch als Unica Campaign-Services bekannt, definiert. Diese Spezifikation wird als Teil des Unica Campaign Services Software Developer's Toolkit (Devkits) geliefert, das mit Unica Campaign installiert ist.

Das vom Installationsprogramm bereitgestellte Verzeichnis `<CAMPAIGN_HOME>/devkits/CampaignServicesAPI` enthält Beispiele, Build- und Textskripte, Javadoc für öffentliche Klassen und Schnittstellen sowie VersionsAnmerkunge.

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

- Bieten Sie detaillierten Zugriff auf Unica Campaign Komponenten zum Erstellen, Erkennen, Lesen und Aktualisieren, während Sie Clients von den zugrunde liegenden Implementierungsdetails isolieren.
- Arbeiten Sie mit minimalen Effekten neben der vorhandenen Unica Campaign Benutzeroberfläche.
- Garantieren der Datenintegrität.
- Erfüllen Sie die erforderlichen Sicherheitsdienste von Unica Campaign.
- Unterstützen des standardisierten SOAP (Simple Object Access Protocol), einschließlich sicherer Authentifizierung.

Übersicht über den SOAP-API-Entwurf

Die SOAP-API von Unica Campaign Services ist eine Fassade, die eine Clientansicht einer laufenden Unica Campaign Anwendungsinstanz bietet. Es wird nur eine Teilmenge der Funktionen von Unica Campaign verfügbar gemacht, aber es reicht aus, um wichtige Aspekte der Unica Campaign Funktionalität zu steuern.

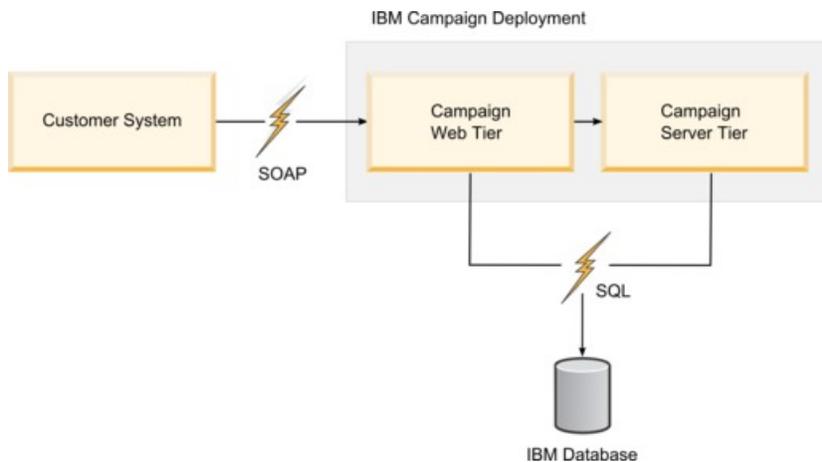
Funktionen und Diagramm

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

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

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

Im folgenden Diagramm wird eine Beispielimplementierung von CampaignServices 3.0 dargestellt.



Benutzerauthentifizierung

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

Benutzerberechtigung

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

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

Ländereinstellung

API-Anforderungen stellen einen optionalen Parameter **requestLocale** bereit, der das Gebietsschema definiert, das zum Ausführen dieser bestimmten Anforderung verwendet werden soll. Ist keine Ländereinstellung definiert, wird in der API standardmäßig die bevorzugte Ländereinstellung des Benutzers verwendet. Der übliche Java™ Best-Effort-Matching-Algorithmus wird verwendet, um Nachrichten und anderen lokalisierten Text im angeforderten Gebietsschema zurückzugeben.

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

 **Anmerkung:** Einige benutzerdefinierte Texte, z. B. Kampagnenbeschreibungen, befinden sich im Gebietsschema des Benutzers, der den Text angegeben hat. Unica Campaign versucht nicht, solche Daten zu lokalisieren. Die API lokalisiert lediglich die Informationen, Warnungsanmerkung und Fehlernachrichten.

Statusverwaltung

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

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

SOAP-API-Änderungen nach Version

In diesem Thema werden Änderungen an der Unica Campaign SOAP API für Kunden identifiziert, die derzeit die API verwenden. Wenn Sie ein Upgrade von einer früheren Version durchgeführt haben, prüfen Sie die folgenden Informationen, um zu ermitteln, ob Sie Änderungen an Ihrem Anwendungscode vornehmen müssen.

Versionen und Abwärtskompatibilität

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

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

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

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

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

unterstützt weiterhin die veröffentlichte WSDL, den SOAP-Client und die Apache Axis-Version, die zur Implementierung des SOAP-Angebots bis zum nächsten Hauptrelease mindestens verwendet wird. In der Praxis wird dies erreicht, indem mehrere versionenspezifische Webdienste gleichzeitig unterstützt werden. (unterstützt bereits mehrere Versionen dieses Dienstes intern.)

Änderungen in v9.1.2

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

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

Änderungen in v9.1.1

stopFlowchartenthält einen neuen Eingabeparameter, runId, um eine Clustered-Listener-Umgebung zu unterstützen.

Änderungen in v9.1

In Version 9.1 wurden keine Änderungen vorgenommen.

Änderungen in v9.0

In Version 9.0 wurden keine Änderungen vorgenommen.

Änderungen in v8.6

In v8.6 implementierte API-Änderungen:

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

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

und die entsprechende WSDL wird abgerufen unter:

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

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

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

- [Bestehende Implementierungen, die die JAR-Datei der Client-API verwenden \(auf Seite 7\)](#)
- [Bestehende Implementierungen, die WSDL direkt verwenden \(auf Seite 8\)](#)

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

Diese Informationen beziehen sich auf ein Upgrade auf Unica Campaign Version 8.6 oder höher und verwenden derzeit die Client-API `.jar`-Datei, um mit der Unica Campaign Webanwendung zu interagieren.

JAR-Datei der Client-API

Ihre Java Anwendung muss die `.jar` Datei verwenden, die sich unter folgender Adresse befindet:

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

Ein Java Beispiel, das die Erstellung neuer Angebote zeigt, finden Sie unter [OfferAPI.java \(auf Seite 13\)](#). Das gleiche Beispiel finden Sie in Ihrer Unica Campaign-Installation unter:

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

Abhängige JAR-Dateien

Als Ergebnis des Upgrades auf AXIS2 Version 1.5.2 muss Ihre Java Anwendung auch auf die Verwendung der AXIS2 1.5.2-Verteilungsdateien `.jar` aktualisieren, da `CampaignServicesClient30.jar` von diesen `.jar` Dateien abhängig ist. Alle abhängigen `.jar` Dateien müssen im Java Klassenpfad Ihrer Anwendung enthalten sein und befinden sich in der `Campaign.war` Datei unter `<CAMPAIGN_HOME>/Campaign.war`.

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

Konstruktor der Client-API

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

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

Parametrisierte Konstruktoren der unterstützenden Klassen

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

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

Bestehende Implementierungen, die WSDL direkt verwenden

Diese Informationen beziehen sich auf ein Upgrade auf Unica Campaign Version 8.6 oder höher und verwenden derzeit die WSDL, um mit der Unica Campaign Webanwendung zu interagieren. Mithilfe der WSDL des Unica Campaign-Web-Services werden durch Verwendung eines Converter-Tools eines anderen Anbieters clientseitige Stubs generiert und Klassen unterstützt. In hier angegebenen Beispielen wird das WSDL2Java-Tool von Apache AXIS2 1.5.2 verwendet.

WSDL-Position und Service-URL

Der Unica Campaign Webdienst für Unica Campaign wird bereitgestellt unter:

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

Die entsprechende WSDL kann an der folgenden Position abgerufen werden:

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

Generieren von Stubs und Klassen

Das WSDL2Java-Tool aus Apache AXIS2 1.5.2 kann verwendet werden, um die Stubs und unterstützenden Java Klassen aus der WSDL zu generieren. Im Folgenden wird eine Ant-Task als Beispiel dargestellt.

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

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

```
<java classname="org.apache.axis2.wsdl.WSDL2Java" fork="true">
  <classpath refid="axis2.class.path"/> <!--Class path having
  AXIS2 libraries -->
  <arg value="-uri"/>
  <arg file="CampaignServices30.wsdl"/> <!--Actual location of
  WSDL -->
  <arg value="-s"/> <!-- Generate sync style code -->
  <arg value="-Euwc"/> <!-- Takes care of generating Wrapper
  java types for nillable = true elements. -->
  <arg value="-uw"/> <!-- Unwrap params -->
  <arg value="-u"/> <!-- Unpack classes -->
  <arg value="-ns2p"/> <!-- Namespace to package mapping. Customer
  can have their own package names. -->
  <arg value="http://webservices.unica.com/campaign/CampaignServices/
  3.0=com.unica.publicapi.campaign.campaignservices.soap.v30"/>
  <arg value="-o"/> <!-- Output directory -->
  <arg file="{autogen.java.dir}"/>
```

```
</java>
```

Verwenden von generierten Stubs und Unterstützen von Klassen

Der Stub kann wie folgt verwendet werden:

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

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

Das Angebot kann wie folgt erstellt werden:

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

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

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

    WSTextAttribute[] wsAttributes = {nameAttribute};
    // convert to WSAttributeArrays
    WSAttributeArrays obj = new WSAttributeArrays();
    obj.setTextAttributes(wsAttributes);
```

```

//Please change the values of following variables to match your
environment.

String authorizationLoginName = "asm_admin"; //login user name
String partitionName = "partition1"; //Use your security policy of
Campaign
String securityPolicyName = "Global Policy"; //Use your security policy
of Campaign

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

//For folderID <=0, offer will be created at root level.

CreateOffer createOfferObject = new CreateOffer();
createOfferObject.setAuthorizationLoginName(authorizationLoginName);
createOfferObject.setPartitionName(partitionName);
createOfferObject.setRequestedLocale(Locale.US.toString());
createOfferObject.setSecurityPolicyName(securityPolicyName);
createOfferObject.setName(offerName);
createOfferObject.setFolderID(folderID);
createOfferObject.setTemplateName(templateName);
createOfferObject.setAttributes(obj);
// make campaign Webservice call
WSCreateOfferResponse wsResponse =
serviceStub.createOffer(createOfferObject);

// process status
WSRequestStatus status = wsResponse.getStatus();

// done
WSOfferInfo offerInfo = wsResponse.getOfferInfo();
System.out.println("status = "+status.getStatusType());
System.out.println("offerInfo = "+offerInfo.getName());

```

```
} catch (Exception exception) {  
    //Handle the Exception here.  
    exception.printStackTrace();  
}
```

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

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

SOAP-API-Referenzen

Die folgenden Referenzen wurden verwendet, um die Campaign SOAP API-Spezifikation vorzubereiten.

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

Kapitel 3. Verwenden der SOAP-API von Unica Campaign

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

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

Unica Campaign Bietet eine Client-API, die SOAP-Webdienste für die Interaktion mit der Unica Campaign Webanwendung verwendet. Dieser Wrapper ist in einer `.jar` Datei gebündelt, mit der die Clientanwendung die Unica Campaign API aufrufen kann.

Die `.jar` Datei finden Sie unter:

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

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

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

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

Außerdem lautet die URL für Unica Campaign Webdienste `http://host:port/Campaign/services/CampaignServices30Service`, wobei sich Host und Port auf den Hostnamen und die Portnummer des Computers beziehen, auf dem die Unica Campaign Webanwendung bereitgestellt wird.

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

OfferAPI.java

Um dieses Java Beispiel zu kompilieren und auszuführen, müssen Sie alle abhängigen `.jar` Dateien in den Java Klassenpfad aufnehmen. Die `CampaignServicesClient30.jar` Datei hängt von den Apache AXIS2 SOAP Engine `.jar` Dateien und anderen gängigen Apache `.jar` Dateien ab, die in `<CAMPAIGN_HOME>/Campaign.war` enthalten sind. Extrahieren Sie die `.jar` Dateien aus `Campaign.war` und fügen Sie sie in den Java-Klassenpfad ein.

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

/**
 * This is the sample java client class that shows the usage of Unica
 * Campaign SOAP
 * services API.
 * This sample uses CampaignServices30SoapClient facade to interact
 * with Unica Campaign
 * web service.
 * Here the creation of Offer is shown. Please refer to the API guide for
 * more details.
 *
 * @author AGijare
 *

```

```

*/
public class OfferAPI {

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

    private static void setup(){
        try {
            String protocol = "http"; //http or https
            String host = "localhost"; //Host name of deployed Campaign.
            Use proper host name here.
            int port = 7001; //port number of deployed Campaign
            long timeOut = 2*60*1000; // 2 minutes
            String servicesURI = "/Campaign/services/
CampaignServices30Service";

            CLIENT = new CampaignServices30SoapClient(
                new URL(protocol, host, port, servicesURI),
                timeOut);
        } catch (Exception exception) {
            exception.printStackTrace();
            System.exit(-1);
        }
    }

    public static void main(String[] args) {
        //Please change the values of following variables to match your
environment.

        String userName = "user_name"; //login user name
        String partitionName = "partition1"; //Use proper partition name
of

```

```

Campaign
    Locale loc = Locale.US;

    String securityPolicy = "Global"; //Use your security policy of
Campaign

    String offerName = "Offer1";

    String offerTemplate = "Offer Template"; // Template from which
Offer will be created.

    long folderID = 1002; //Actual ID of the folder where this offer
will be created. For folderID <=0, offer will be created at root level.

    //Attributes of Offer
    WSAttribute[] wsAttributes = {

        WSAttributeUtils.getWSTextAttribute(IAttributeMeta
data.AC_OFFER_DESCRIPTION_ATTRIBUTE_NAME, null, new String[]{"description
"
+ System.currentTimeMillis()})
    };

    setup();

    try {

        WSOfferInfo wsOfferInfo = CLIENT.createOffer(userName,
partitionName, loc, securityPolicy,

        offerName, folderID, offerTemplate, wsAttributes);

        System.out.println("Created offer: " + wsOfferInfo.getName());
    } catch (CampaignServicesException e) {

        e.printStackTrace();

    }

}
}

```

Verwenden der WSDL zum Aufrufen der Unica Campaign-Services

Unica Campaign services kann mithilfe der WSDL-Datei Unica Campaign web services aufgerufen werden: `CampaignServices30.wsdl`.

Die `CampaignServices30.wsdl` Datei finden Sie unter:

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

oder in der Unica Campaign-Verteilung unter:

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

Die Client Java -Anwendung muss die Klassen und Stubs verwenden, die aus der WSDL mithilfe eines WSDL-Java-Konverter-Tools eines Drittanbieters generiert werden. empfiehlt die Verwendung von Apache AXIS.

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

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

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

Leistungsüberlegungen

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

In der SOAP-Schnittstelle werden von Natur aus Latenzzeit und Mehraufwand eingeführt, da alle Daten in das XML-Format konvertiert werden, das in einigen Fällen ziemlich ausführlich ist. Beispielsweise kann ein einfacher Loopback-SOAP-Aufruf in einem typischen Netzwerk 100 ms dauern (Java 1.4.x war noch langsamer). Die API ist für typische Anwendungsfälle von Portal- und anderen Clientanwendungsgeschäften wie `see listOffersByPage()` optimiert, sodass die SOAP-Leistung angemessen sein sollte.

Der Client muss jedoch aufpassen, dass er die Gemeinkosten bei der normalen CampaignServices-Wartung bei Anforderungen von Webbenutzern nicht zu hoch ansetzt. Im Allgemeinen wird erwartet, dass die Verarbeitungsanforderungen eines API-Benutzers die eines typischen Unica Campaign Webbenutzers nicht überschreiten.

Kapitel 4. SOAP-API-Datentypen

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

WSReference

Ein einfacher Wrapper in einer Datenbank-ID:

- **ComponentTypeEnum**: Ein aufgezählter Typ, der den Komponententyp angibt, für den die ID bestimmt ist. Eine der folgenden:
 - Ordner
 - CAMPAIGN
 - FLOWCHART
 - TCS_CELL
 - OFFER
 - OFFER_LIST
 - OFFER_TEMPLATE
- *id*: Ein *Long*, der eine eindeutige numerische datenbankspezifische Kennung für die Referenz definiert.

WSVersion

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

- *major*: eine Ganzzahl, mit der die Hauptversionsnummer definiert wird, wie z. B. "8" von der Vollversion 8.1.2.3.
- *minor*: eine Ganzzahl, mit der die Hauptversionsnummer definiert wird, wie z. B. "1" von der Vollversion 8.1.2.3.

- *maintenance*: optionale Ganzzahl, mit der, falls zutreffend, die Wartungsnummer der Version definiert wird, wie z. B. "2" von der Vollversion 8.1.2.3. Wird niemals mit einer API-Version bereitgestellt.
- *patch*: optionale Ganzzahl, die gegebenenfalls die Patch-Versionsnummer definiert, z. B. '3' der Vollversion 8.1.2.3. Wird niemals mit einer API-Version bereitgestellt.

WSServiceInfo

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

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

WSAttributeTypeEnum

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

- STANDARD: Standard- oder Basisattribut, das durch Unica Campaign definiert ist.
- BENUTZERDEFINIERT: Ein Attribut, das von einer anderen Anwendung, dem Kunden oder einem anderen Anbieter definiert wurde.
- INPUT_PARAMETER: Ein Eingabeparameter, wie z. B. ein Attribut, mit dem ein Unica Campaign Ablaufdiagramm ausgeführt wird.
- OUTPUT_PARAMETER: Ein Ausgabeparameter, wie z. B. ein Attribut, dessen Wert als Ergebnis eines Ablaufdiagramms entsteht, das in Unica Campaign ausgeführt wird.

WSAttributeStatusEnum

Eine Aufzählung aller möglichen Attributstatuscodes. Dazu zählen:

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

WSAccessTypeEnum

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

- **READ_ONLY**: Der Attributwert kann gelesen und angezeigt, aber nicht geändert werden.
- **READ_WRITE**: Der Attributwert kann gelesen, angezeigt und geändert werden.

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

WSSelectTypeEnum

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

- **KEINER**: keine Auswahl (*hasOptions* ist auf "false" festgelegt).
- **SINGLE_SELECT**: Es kann nur jeweils eine Attributoption aus der Liste der möglichen Optionen ausgewählt werden (nur gültig, wenn das Attribut über *hasOptions* verfügt).
- **MULTIPLE_SELECT**: ähnelt **SINGLE_SELECT**. Ausnahme ist, dass eine oder mehr Optionen gleichzeitig ausgewählt werden können.

WSRunStatusEnum

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

- NOT_STARTED: die Ausführung wurde geplant, jedoch noch nicht gestartet.
- AKTIV: run in Bearbeitung.
- ABGEBROCHEN: die Ausführung wurde abgebrochen, entweder von einem Unica Campaign-Benutzer oder dieser API.
- GELUNGEN: Erfolgreich abgeschlossen.
- FAILED: die Ausführung ist fehlgeschlagen; Fehlerdetails werden separat gemeldet. (Siehe [WSRunResults \(auf Seite 29\)](#).)

WSRunTypeEnum

Ein aufgezählter Typ aller möglichen Ausführungstypen. Dazu zählen:

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

WSAttribute

Attribute bieten einen einfachen, erweiterbaren Mechanismus zum Anhängen beliebiger Daten an Komponenteninstanzen, auf die über die API zugegriffen werden kann, entweder

Standarddaten wie ein Campaign name, Eingabeparameter für Flussdiagramme wie Geschlecht oder beliebige benutzerdefinierte Daten, die von einer anderen Anwendung oder einem anderen Kunden angegeben werden.

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

Komponenten sind im Allgemeinen viele Attribute zugeordnet, die von der CampaignServices-API als speziell typisierte Map angezeigt werden, die als AttributeMap bezeichnet wird. Attributdaten werden in der gesamten API als typisierte konkrete Klasse dargestellt, z. B. WSDecimalAttribute, für Attribute, die Dezimaldaten (numerische Daten mit doppelter Genauigkeit) enthalten.

Jedes Attribut umfasst folgende Komponenten:

- *Name:* der einzigartiger Name des Attributs. Dieser Name dient als Schlüssel für den Zugriff auf das Attribut und die zugehörigen Attributdaten innerhalb der auftretenden Komponenteninstanz. Das Format des Namens ist undefiniert. In einigen Fällen wird es vom Dienst, vom Client oder von einem Unica Campaign Benutzer zugewiesen.

Im Allgemeinen handelt es sich bei diesem Namen nicht um den Anzeigenamen, der einem Unica Campaign-Benutzer oder einem Clientbenutzer angezeigt werden würde. Es kann von der API standardisiert werden, z. B. uacDescription, es kann von Unica Campaign beim Veröffentlichen von Flussdiagrammen zugewiesen werden oder es kann von der Anwendung oder dem Kunden beim Definieren benutzerdefinierter Attribute zugewiesen werden. In allen Fällen ist der Name allerdings garantiert eindeutig.

- *Metadata:* (optional) Informationen zu den Attributdaten, wie z. B. Wertdatentyp, Anzeigenname, Beschreibung, Eingabeaufforderungen, Standardwert, Auswahltyp, Länge (Text), Präzision (Dezimalstellen), Optionen (ob Einfach- oder Mehrfachauswahl). Siehe [WSAttributeMetadata \(auf Seite 24\)](#).
- *Values:* ein Array mit null oder mehr typisierten Wertobjekten. Das Wertfeld wird von der konkreten Attributklasse bereitgestellt; der Typ der einzelnen Werte muss mit der Typdefinition im Feld mit den Metadaten des Attributs identisch sein. Allerdings unterstützen nicht alle Attribute mehrere Werte.

Die folgenden konkreten Attributtypen werden unterstützt:

- **WSBooleanAttribute:** Ein Attribut, dessen Wert ein Boolescher Wert ist, das heißt, *true* oder *false*.
- **WSIntegerAttribute:** ein ganzzahliger Wert (*java.lang.Long*).
- **WSDecimalAttribute:** ein Dezimalzahlwert mit doppelter Genauigkeit (*java.lang.Double*).
- **WSCurrencyAttribute:** ein zusammengesetzter Währungswert, der den optionalen Währungscode ISO 4217 des Währungswerts umfasst, z. B. "USD" für den US-Dollar, und die als *Double* erfassten Währungswerte. Wenn der Währungscode nicht angegeben wird, wird der von Unica Campaign verwendete Standardwert angenommen.

Unter <http://www.xe.com/symbols.php> finden Sie eine Liste der Länder, Währungssymbole und Codes. Die Ländereinstellung, die bei einem Währungswert verwendet wird, kann sich von der bevorzugten Ländereinstellung eines Benutzers unterscheiden.

- **WSCalendarAttribute:** Werte sind, in einigen Zeitzonen und Ländereinstellungen, Kalenderdaten oder Datumsangaben.
- **WSTextAttribute:** eine Zeichenfolge mit Unicodetext (wahrscheinlich null oder leer).

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

WSAttributeMetadata

WSAttributeMetadata definiert Informationen zu den Daten eines bestimmten typisierten Attributs, wie z. B. zum Wertdatentyp, zu lokalisiertem Text (Anzeigenname, Beschreibung, Eingabeaufforderungen), zum zugehörigen Standardwert, zum Bereich der gültigen Werte, zum Auswahltyp und zu Optionen (ob Einfach- oder Mehrfachauswahl). Metadaten für Attribute sind wie Attribute typisiert. Beispielsweise muss an eine WSDecimalAttribute myNumber eine WS DecimalAttributeMetadata gebunden sein, und alle Werte,

einschließlich der Attributwerte, des Metadaten-Standardwerts und möglicher Optionswerte, müssen Double eingegeben werden.

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

WSAttributeMetadata umfasst die folgenden Felder:

- *name*: Der Name des Attributs, Standard oder benutzerdefiniert; auch der Name, der von dem Attribut verwendet wird, das an diese Metadaten bindet. Standardattribute werden vom System definiert und haben Standardnamen in einem reservierten Namensraum (dh sie verwenden ein "uac"-Präfix). Benutzerdefinierte Namen können jede andere Namenskonvention verwenden.

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

- *Beschreibung*: Optionale Beschreibung des Attributwerts. Geeignet für eine QuickInfo oder eine andere Darstellung in der Benutzerschnittstelle.
- *Prädikate*: verschiedene Prädikate, die das Attribut beschreiben:
 - *isRequired*: trifft zu, wenn das Attribut obligatorisch ist.
 - *isInternal*: trifft zu, wenn das Attribut vom System definiert wurde und nur für interne Zwecke verwendet wird (sollte einem Benutzer nicht dargestellt werden).
 - *isGenerated*: true, wenn der Wert oder die Werte des Attributs beim Erstellen der Komponente automatisch von Unica Campaign generiert werden, z. B. ein Zielzellencode. Normalerweise lautet *accessTypeEnum* für generierte Werte READ_ONLY.

- *hasOptions*: trifft zu, wenn das Attribut Optionen hat. Impliziert, dass für diese Metadaten Optionen definiert wurden und *selectTypeEnum* entweder SINGLE_SELECT oder MULTIPLE_SELECT ist.
- *typeEnum*: eine *WSAttributeTypeEnum*, die den Attributtyp definiert, z. B. STANDARD oder CUSTOM (benutzerdefiniert).
- *statusEnum*: Ein *WSAttributeStatusEnum*, das den Status des Attributs definiert, z. B. ACTIVE.
- *accessTypeEnum*: Ein *WSAccessTypeEnum*, das die Art des Zugriffs auf den Attributwert definiert, z. B. READ_ONLY.
- *selectTypeEnum*: Ein *WSAccessTypeEnum*, das den Auswahltyp definiert, der für das Attribut verwendet wird, z. B. SINGLE. Muss für Unica Campaign- und Zellenattribute NONE sein oder wenn keine Optionen bereitgestellt werden.
- *ComponentTypeEnum*: eine *WSComponentTypeEnum* aller möglichen Unica Campaign-Komponenten, die von der API verfügbar gemacht werden, z. B. CAMPAIGN, FOLDER.
- *defaultValue*(nur Ablaufdiagramme): Optionaler Typ Standardwert für das Attribut. Dieser Wert wird von der Metadatenklasse des konkreten Attributs bereitgestellt, z. B. der Standardwert eines *WSTextAttributeMetadata* vom Typ String. (Weitere Informationen finden Sie in der Beschreibung der Attributwerte). Bei Komponenten, bei denen es sich nicht um Ablaufdiagramme handelt, ist der Standardwert undefiniert.
- *options*: optionale Liste mit Optionen für dieses Attribut. Zusammengenommen definieren die Optionen eines Attributs den genauen Satz zulässiger Werte für dieses Attribut. Jede Option ist typisiert, sodass an eine *WSTextAttributeMetadata* beispielsweise nur eine *WSTextAttributeOption* gebunden sein kann.

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

die einzelnen Optionen definieren Folgendes:

- *prompt*: Eingabeaufforderung für die Option, die für Pulldown-Menüs geeignet ist, z. B. "Männlich" als Attributoption für das Geschlecht. Anders als bei Eingabeaufforderungen

für Metadaten enthalten die Anzeigenamen von Optionen normalerweise keine Interpunktion.

- *Beschreibung* Lokalisierte Beschreibung der Option, z. B. "Eine Person mit männlicher Überzeugungskraft". Geeignet für QuickInfo-Text.
- *isDefault*: "true", wenn diese bestimmte Option der Standard ist. Bei Auswahltypen vom Typ MULTIPLE_SELECT können mehrere Optionen als Standard markiert werden.
- *value*: Der typisierte Optionswert. Wie bei den Attributmetadaten defaultValue wird dieser Wert von der konkreten Optionsunterklasse bereitgestellt, z. B. der Wert einer *WSDecimalAttributeOption* vom Typ Decimal. (Weitere Informationen finden Sie in der Beschreibung der Attributwerte). Wenn Sie das obige Beispiel für das *Geschlecht* fortsetzen, kann der Wert entweder als Zeichenfolge "m" (*WSTextAttributeOption*) oder als numerischer Code 123 (*WSDecimalAttributeOption*) deklariert werden.

WSCampaignInfo

Ein einfacher Wrappertyp in Kampagnenattributdaten.

Er enthält die folgenden Felder:

- *referenz*: die Kampagnenreferenz.
- *name*: Kampagnenname (*uacName*); nicht einzigartig garantiert.
- *Beschreibung*: optionale Kampagnenbeschreibung (*uacDescription*).
- *campaignCode*: den eindeutigen Kampagnencode (*uacCampaignCode*); entweder vom Kunden zugewiesen oder Unica Campaign.

WSComponentOrFolderInfo

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

Er enthält die folgenden Felder:

- *referenz*: die Komponenten- oder Ordnerreferenz.

- *name*: Komponenten- oder Ordnername (*uacName*); nicht einzigartig garantiert.
- *Beschreibung*: optionale Kampagnenbeschreibung (*uacDescription*).
- *componentCode*: eindeutiger Code für die Komponente bzw. null bei einem Ordner.

WSTargetCellInfo

Ein einfacher Wrapper in Attributdaten für Zeilen mit Zielzellen.

Er enthält die folgenden Felder:

- *referenz*: die Zellenreferenz.
- *name*: Zellenname (*uacName*); nicht einzigartig garantiert.
- *Beschreibung*: optionale Zellenbeschreibung (*uacDescription*).
- *cellCode*: der Zellencode (*uacCellCode*); entweder vom Kunden oder von der Kampagne zugewiesen. Zellencodes können als eindeutig erzwungen werden, indem der Konfigurationsparameter Unica Campaign *DuplicateCellCodesAllowed* auf false gesetzt wird.
- *flowchartName*: optionaler Name des Ablaufdiagramms, an das die Zelle gebunden wird.

WSMetricsInfo

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

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

WSRunResults

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

Er umfasst die folgenden Felder:

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

- *count*: optionale Gesamtanzahl der Kontakte, die bei der Ausführung ausgewählt werden; die Angabe beträgt möglicherweise null, wenn die Ausführung nicht abgeschlossen wurde.

WSOfferInfo

Ein einfacher Wrappertyp in Attributdaten von Angeboten oder Angebotslisten.

Er enthält die folgenden Felder:

- *referenz*: die Angebots- oder Angebotslistenreferenz.
- *name*: Angebot oder Angebotslistenname (*uacName*); nicht garantiert einzigartig.
- *Beschreibung*: optionale beschreibung (*uacDescription*).
- *offerCode*: der Angebotscode (*uacOfferCode*), wenn es sich um ein Angebot handelt, oder null, wenn es sich um eine Angebotsliste handelt. (Nicht garantiert eindeutig.)

WSOfferCodeOrName

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

Er enthält die folgenden Felder:

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

WSOfferValidationInfo

Ein einfacher Wrappertyp in Informationen zur Angebotsvalidierung.

Er enthält die folgenden Felder:

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

WSOfferTemplateInfo

Ein einfacher Wrappertyp in Daten von Angebotsvorlagen.

Er enthält die folgenden Felder:

- *referenz*: die Referenz der Angebots vorlag.
- *name*: Name der Angebotsvorlage; garantiert eindeutig.
- *Beschreibung*: optionale beschreibung (*uacDescription*).
- *offerTemplateID*: die eindeutige Datenbank-ID der Angebotsvorlage.

WSBulkOfferInfo

Wird zur Erstellung von Massenangeboten verwendet.

Er enthält die folgenden Felder:

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

WSOfferInfoStatus

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

Er enthält die folgenden Felder:

- *name*: Der Name des Angebots.
- *code*: der angebotscode. Schlägt die Angebotserstellung fehl, beträgt sie null.
- *Beschreibung*: Angebotsbeschreibung.
- *referenz*: WSReference des erstellten Angebots. Schlägt die Angebotserstellung fehl, beträgt sie null.
- *status*: eine Instanz von *WSRequestStatus*, die den Status der Angebotserstellung angibt.

Kapitel 5. SOAP-API-Methoden

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

SOAP API Methoden: Service

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

getServiceInfo

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

Gibt Informationen über den Dienst zurück, z. B. die aktuellste unterstützte API-Version, die Vollversion der zugrunde liegenden Unica Campaign Instanz.

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

Parameter

Keine.

Rückgabe

Gibt eine WSServiceInfo Instanz zurück.

Fehler

Keine.

SOAP API Methoden: Attribute

Die meisten Komponenteninstanzdaten können von der Unica Campaign SOAP-API als Attribute oder Attributmetadaten verfügbar gemacht werden.

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

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

getAttributesByName

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

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

Parameter

userCredential: die Client-Benutzeranmeldeinformationen.

requestedLocale: optionales Gebietsschema für diese Anforderung; Wenn nicht angegeben, werden die Ländereinstellungen des Benutzers verwendet. Bei Bedarf wird der Algorithmus zur Standardisierung der Ländereinstellung angewendet.

PartitionName: Optionaler Name der zu verwendenden Kampagnenpartition. Ist keine Partition definiert, wird die Standardpartition verwendet.

referenz: die Referenz für die Komponenteninstanz, die die gewünschten Attribute enthält. InvalidComponentException wird ausgelöst, wenn die Referenz ungültig ist oder die Komponente nicht vorhanden ist.

namen: optionales Array mit Namen von Attributen, die abgerufen werden sollen (keine Anzeigenamen); Wenn 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 stellt den Zuordnungseingabeschlüssel und die Attributinstanz den Eingabewert dar.

Fehler

InvalidComponentException, AttributeNotFoundException

AuthorizationException, DataException

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

updateAttributes

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

Aktualisiert mindestens ein Attribut der Komponenteninstanz mit den angegebenen Attributwerten.

Aktualisierungslogik

Die Aktualisierungslogik lautet wie folgt.

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

1. Wenn der Attributname mit einem vorhandenen Attribut übereinstimmt, versuchen Sie, das *Werte* feld mit dem angegebenen Wertefeld zu überschreiben.
2. Wenn das Attribut noch nicht vorhanden ist, ist *allowCreate* true und seine Metadaten sind bekannt. Erstellen Sie dann das Attribut. Dies gilt sowohl für Metadaten für globale Attribute als auch für Instanzattribute (mit Ausnahme von Ablaufdiagrammen).
3. Wenn der Werttyp oder ein anderer Aspekt der Metadatendefinition des Attributs nicht erfüllt ist oder einer oder mehrere der angegebenen Werte ungültig sind oder außerhalb des Bereichs liegen, lösen Sie *InvalidAttributeException* aus.
4. Andernfalls werfen Sie *AttributeNotFoundException*, wenn das benannte Attribut nicht vorhanden ist.

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

Diese spezielle Methode unterstützt nicht das Definieren neuer benutzerdefinierter Attribute. Verwenden Sie dazu die `createAttributeMetadata()`-Methode.

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

Parameter

`userCredential`: die Client-Benutzeranmeldeinformationen.

`requestedLocale`: optionales Gebietsschema für diese Anforderung.

`PartitionName`: Optionaler Name der zu verwendenden Kampagnenpartition.

`referenz`: Die Referenz für die 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 \(auf Seite 35\)](#).)

Attributen: ein Array von Attributen, die aktualisiert werden sollen; Der Attributname wird verwendet, um das zu aktualisierende Attribut zu suchen, und die neuen Werte werden verwendet, um den Wert des vorhandenen Attributs als einzelnes Objekt des richtigen Typs oder gegebenenfalls als Array zu aktualisieren. (Auf [Allgemeine SOAP-API-Ausnahmen \(auf Seite 92\)](#) beziehen.)

Rückgabe

Keine.

Fehler

InvalidComponentException, AttributeNotFoundException, InvalidAttributeException

AuthorizationException, DataException

getAttributeMetadataByName

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

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

Parameter

userCredential: die Client-Benutzeranmeldeinformationen.

requestedLocale: optionales Gebietsschema für diese Anforderung.

PartitionName: Optionaler Name der zu verwendenden Kampagnenpartition.

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

namen: optionales Array mit Namen der abzurufenden Attributmetadaten. Wenn das Array nicht angegeben ist, werden alle Metadaten der Komponente bzw. alle global definierten Metadaten (wenn keine Referenz angegeben ist) zurückgegeben. Löst AttributeNotFoundException aus, wenn eine oder mehrere der angegebenen Attributmetadatendefinitionen nicht vorhanden sind.

Rückgabe

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

Fehler

InvalidComponentException, AttributeNotFoundException

AuthorizationException, DataException

createAttributeMetadata

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

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

Parameter

userCredential: die Client-Benutzeranmeldeinformationen.

requestedLocale: optionales Gebietsschema für diese Anforderung.

PartitionName: Optionaler Name der zu verwendenden Kampagnenpartition.

referenz: optionale Referenz für die Komponente oder Vorlage, an die die Metadaten gebunden werden sollen. Ist diese Referenz nicht angegeben, ist die erstellte Metadatendefinition global. Wenn die Referenz angegeben wird, jedoch nicht gültig ist, wird `InvalidComponentException` ausgelöst.

attributeMetadata: Ein Array von Attribut-Metadatendefinitionen, die gebunden werden sollen. Wenn eine oder mehrere der angegebenen Metadaten bereits an die Komponente gebunden sind, dh der Name nicht eindeutig ist, lösen Sie `AttributeExistsException` aus. Löst eine `InvalidAttributeException` aus, wenn ein Problem mit einer oder mehreren der angegebenen Metadaten vorliegt, dh intern inkonsistent ist.

Rückgabe

Keine.

Fehler

`InvalidComponentException`, `AttributeExistsException`, `InvalidAttributeException`

`AuthorizationException`, `DataException`

updateAttributeMetadata

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

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

Aktualisierungslogik

Die Aktualisierungslogik lautet wie folgt.

Bei jeder Metadatendefinition für Attribute im angegebenen Array:

1. Wenn der Attributname nicht mit vorhandenen Metadaten übereinstimmt, die an die Komponente gebunden sind, führen Sie die folgenden Schritte basierend auf dem Parameterwert *allowCreate* aus:
 - a. *True*: Erstellen Sie eine neue Metadatendefinition. Funktionell identisch mit der Verwendung der `createAttributeMetadata()` Anfrage.
 - b. *False*: Löst *AttributeNotFoundException* aus.
2. Wenn der Datentyp für Attributmetadaten unterschiedlich ist, lösen Sie *InvalidAttributeException* aus.
3. Versuchen Sie, die vorhandene Attribut-Metadatendefinition mit den Feldwerten der angegebenen Metadaten zu überschreiben, andernfalls lösen Sie *InvalidAttributeException* aus. Es werden nur die folgenden Aktualisierungen unterstützt (lösen Sie andernfalls *InvalidAttributeException* aus):
 - a. *name*: kann nicht geändert werden (name ist der Schlüssel!)
 - b. *displayName*: Neuen Wert Akzeptieren.
 - c. *Beschreibung*: Neuen Wert Akzeptieren.
 - d. *isRequired*: Nur Änderung von *true* zu *false* zulässig.
 - e. *isInternal*: Neuen Wert Akzeptieren.
 - f. *isGenerated*: Keine Änderung erlaubt.
 - g. *attributeTypeEnum*: Keine Änderung erlaubt.
 - h. *accessTypeEnum*: Neuen Wert Akzeptieren.
 - i. *selectTypeEnum*: Akzeptieren dieser Übergänge, wenn Optionen bereitgestellt werden:
 - i. NONE zu SINGLE_SELECT oder MULTIPLE_SELECT
 - ii. SINGLE_SELECT zu MULTIPLE_SELECT

- j. *options*: Optionen können hinzugefügt, jedoch nicht gelöscht werden. Es werden nur die folgenden Optionsänderungen unterstützt (gemäß der Wertübereinstimmung):
 - i. *displayName*: Neuen Wert Akzeptieren (keine welligkeit)
 - ii. *Beschreibung*: Neuen Wert Akzeptieren (keine welligkeit)
 - iii. *isDefault*: Akzeptieren des neuen Werts; muss jedoch SelectTypeEnum entsprechen.
 - iv. *value*: Es ist keine Änderung zulässig (Wert ist der Schlüssel!)
- k. *defaultValue*(nur Ablaufdiagramme): Neuer Standard Wert Akzeptieren.
- l. *maxLength*(nur Text): Akzeptieren Sie die neue Länge, wenn sie größer ist.

4. Wenn die Attributmetadatendefinition intern nicht konsistent ist, lösen Sie *InvalidAttributeException* aus.
5. Suchen Sie bei Bedarf alle Komponenteninstanzen, die auf die aktualisierten Metadaten des Attributs verweisen, und aktualisieren Sie sie entsprechend.

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

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

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

Parameter

`userCredential`: die Client-Benutzeranmeldeinformationen.

`requestedLocale`: optionales Gebietsschema für diese Anforderung.

`PartitionName`: Optionaler Name der zu verwendenden Kampagnenpartition.

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

`allowCreate`: Wenn true, werden Metadatendefinitionen erstellt, die derzeit nicht vorhanden sind (funktional äquivalent zur Verwendung der `createAttributeMetadata()` Methode).

attributeMetadata: Ein Array von Attribut-Metadatendefinitionen, die aktualisiert werden sollen (und hinzugefügt werden, wenn das Flag `allowCreate` `true` ist). Mit dem Attributnamen wird die zu aktualisierende Metadatendefinition gesucht. Die verbleibenden Daten werden für die Aktualisierung der vorhandenen Definition verwendet. (Auf [Aktualisierungslogik \(auf Seite 40\)](#) beziehen.)

Rückgabe

Keine.

Fehler

`InvalidComponentException`, `InvalidAttributeException`

`AuthorizationException`, `DataException`

deleteAttributeMetadata

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

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

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

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

Parameter

`userCredential:` die Client-Benutzeranmeldeinformationen.

requestedLocale: optionales Gebietsschema für diese Anforderung.

PartitionName: Optionaler Name der zu verwendenden Kampagnenpartition.

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

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

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

Rückgabe

Keine.

Fehler

InvalidComponentException, AttributeNotFoundException, InvalidAttributeException

AuthorizationException, DataException

SOAP API Methoden: Kampagnen und Ablaufdiagramme

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

- Erstellen einer Kampagne
- Erkennung (Auflisten von Kampagnen nach verschiedenen Kriterien)

- Erstellen, Lesen und Aktualisieren von Attributen (über Attribut-APIs)
- Stoppen einer Ablaufdiagrammausführung

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

Die Standardkampagnenattribute sind im Folgenden aufgelistet:

- *uacName*: Kampagnenname (nicht einzigartig garantiert).
- *uacDescription*: optionale Zeichenfolge, die die Kampagne beschreibt.
- *uacCampaignCode*: Ein Zeichenfolgcocode, der die Kampagne eindeutig identifiziert. Er wird üblicherweise von Campaign generiert, 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, die die Kampagneninitiative definiert.
- *uacObjectives*: optionale Zeichenfolge, die die Ziele der Kampagne identifiziert.
- *uacStartDate*: Ein optionaler Kalender, der Datum und Uhrzeit angibt, zu denen die Kampagne vom Server gestartet wurde oder deren Start geplant ist.
- *uacEndDate*: Wie bei *uacStartDate*, definiert jedoch das Datum und die Uhrzeit, zu der die Kampagne abgeschlossen wurde oder deren Abschluss geplant ist. Muss nach dem *uacStartDate* sein.
- *uacLastRunDate*: Ein optionaler Kalender, der Datum und Uhrzeit der letzten Ausführung eines an die Kampagne gebundenen Flussdiagramms angibt (ansonsten null).
- *uacExternalLinkOwner*: Eine optionale Zeichenfolge, die den Namen des Eigentümers eines externen Links definiert (siehe Attribut *uacExternalLinkReference*). nur benutzen; muss eine der folgenden sein:
 - "Plan" (bekannt als Unica Plan)
 - "Collaborate" (bekannt als Unica Collaborate)

- *uacExternalLinkId*: Eine optionale numerische Datenbankkennung, die von einer anderen Anwendung einem mit dieser Kampagne verknüpften Objekt zugewiesen wird. nur verwenden: Siehe 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 vorherigen oder zukünftigen Aufruf dieser Methode oder der `createCampaign()`-Methode zurückgegeben wird, oder dem Wert, der für eine Kampagne generiert wird, die von der erstellt wird Unica Campaign GUI.

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

Siehe `createCampaign()`.

Parameter

`userCredential`: die Client-Benutzeranmeldeinformationen.

`requestedLocale`: optionales Gebietsschema für diese Anforderung.

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

Rückgabe

Der generierte Kampagnencode.

Fehler

AuthorizationException, DataException

deleteCampaigns

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

Löscht angegebene Kampagnen aus dem System.

Parameter

userCredential: die Client-Benutzeranmeldeinformationen.

PartitionName: Optionaler Name der zu verwendenden Kampagnenpartition.

requestedLocale: optionales Gebietsschema für diese Anforderung.

wsReference: Referenzen auf zu löschende Kampagnen.

Rückgabe

Gibt ein Objekt vom Typ WSDeleteCampaignsResponse zurück.

Fehler

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

createCampaign

```
CampaignInfo createCampaign(String userCredential,  
String partitionName,  
Locale requestedLocale,  
String securityPolicyName,  
WSReference wsReference,  
String name, Attribute[] attributes)
```

```
throws InvalidFolderException, AttributeNotFoundException,
    InvalidAttributeException;
```

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

Verwenden Sie den Parameter `WSReference`, um den Kampagnenordner anzugeben, um Kampagnen unter einem bestimmten Ordner zu erstellen.

Beispiel

```
private static void createCampaign
    (String userName, String partitionName, Locale loc, String
    securityPolicy,
        String campaignName, long campaignfolderID)
{
    WSAttribute[] wsAttributes = { WSAttributeUtils.getWSTextAttribute
        (IAttributeMetadata.AC_CAMPAIGN_DESCRIPTION_ATTRIBUTE_NAME,
    null, new String[]
        { "description " + System.currentTimeMillis() }) };

    try
    {
        WSReference wsReference = WSAttributeUtils.getWSReference
            (WSComponentTypeEnum.FOLDER, campaignfolderID);
        WSCampaignInfo wsCampaignInfo = CLIENT.createCampaign
            (userName, partitionName, loc, securityPolicy, wsReference,
    campaignName, wsAttributes);

        System.out.println("Created Campaign with Name: " +
    wsCampaignInfo.getName()
            + " CampaignCode: " + wsCampaignInfo.getCampaignCode());
    }
}
```

```
catch (CampaignServicesException e)
{
    e.printStackTrace();
}
}
```

Parameter

userCredential: die Client-Benutzeranmeldeinformationen.

requestedLocale: optionales Gebietsschema für diese Anforderung.

PartitionName: Optionaler Name der zu verwendenden Kampagnenpartition.

securityPolicyName: Optionaler Name der Campaign Sicherheitsrichtlinie, die zum Erstellen der Kampagne verwendet werden soll. Bei allen nachfolgenden Vorgängen in dieser Kampagne wird diese Richtlinie verwendet. Ist keine Sicherheitsrichtlinie definiert, wird die globale Richtlinie verwendet.

wsReference: Optionaler Name des Kampagnenordners, in dem die Kampagnen erstellt werden sollen.

name: Der Name, der neuen Kampagneninstanz zugewiesen werden soll (das Attribut "uacName").

Attributen: ein optionales Array von Initialisierungsattributen; Alle angegebenen Attribute überschreiben die Standardwerte der Kampagne. andere bleiben unberührt. Wenn beispielsweise ein uacCampaignCode-Attribut angegeben wird, wird es anstelle eines automatisch generierten Attributs verwendet. Es ist Aufgabe des Clients, die für die Kampagne erforderlichen Attribute, ihren Typ usw. zu ermitteln.

Löst `AttributeNotFoundException` aus, wenn eines oder mehrere der genannten Attribute nicht vorhanden sind, oder `InvalidAttributeException`, wenn ein Attributwert ungültig ist (z. B. falscher Datentyp).

Rückgabe

Eine Einzelinstanz einer `CampaignInfo` für die erstellte Kampagne.

Fehler

InvalidAttributeException, AttributeNotFoundException

AuthorizationException, DataException

listCampaignsByPage

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

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

Nach dem Abrufen kann jede zurückgegebene CampaignInfo unverändert verwendet werden, um beispielsweise eine Zusammenfassungsverzeichnisliste anzuzeigen, oder die Attributmethoden können zum Abrufen oder Aktualisieren der Attribute der Unica-Kampagne verwendet werden.

Diese API verwaltet keinen Status, sodass Sie sie in beliebiger Reihenfolge aufrufen können.

Parameter

userCredential: die Client-Benutzeranmeldeinformationen.

requestedLocale: optionales Gebietsschema für diese Anforderung.

PartitionName: Optionaler Name der zu verwendenden Kampagnenpartition.

Attributen: optionales Array von passenden Attributen; Der Name, der Datentyp und die Werte des Attributs werden verwendet, um die Übereinstimmung zu bestimmen. Wenn das Attribut Arrays unterstützt, müssen alle angegebenen Werte übereinstimmen. Der

implizierte Übereinstimmungsoperator lautet AND. Folglich werden nur Kampagnen zurückgegeben, die mit allen angegebenen Attributwerten übereinstimmen.

Löst `AttributeNotFoundException` aus, wenn kein Attributname vorhanden ist, oder `InvalidAttributeException`, wenn eines oder mehrere der angegebenen Attribute ungültig sind.

`pageOffset`: der Startoffset aller möglichen Kampagnen zum Beginn der Aufzählung (nullwertig). Beispiel: Wenn bei der Aufzählung 1000 Übereinstimmungen mit Kampagnen gefunden werden und dieser Wert auf 10 festgelegt ist, würde auf der Seite mit der 11. Komponente begonnen werden. Eine `RangeException` wird ausgelöst, wenn der angegebene Offset außerhalb des Bereichs liegt.

`pageSize`: Die maximale Anzahl übereinstimmender Kampagnen, die für die Seite zurückgegeben werden sollen (darf 500 nicht überschreiten).

Rückgabe

Eine typisierte Liste mit null oder mehr `CampaignInfo` Daten-Wrapper Instanzen, eine für jede übereinstimmende Kampagne auf der Seite.

Fehler

`AttributeNotFoundException`, `InvalidAttributeException`, `RangeException`

`InvalidExecutionContextException`, `AuthorizationException`

stopFlowchart

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

Diese API stoppt ein derzeit ausgeführtes Ablaufdiagramm. Bei einer einzelnen Listenerkonfiguration kann eine Ablaufdiagrammausführung eindeutig durch die PID identifiziert werden, die der Ablaufdiagrammausführung zugeordnet ist. Die PID gibt die Prozess-ID des `unica_acsvr` Prozesses an. Wenn mehrere Unica Campaign Listener konfiguriert sind, müssen Sie sowohl die Lauf-ID, die dem Flussdiagrammlauf zugeordnet ist, als auch die PID angeben.

Parameter

pid: Prozess-ID des `unica_acsvr` Prozesses, der einem Flussdiagrammlauf zugeordnet ist.

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

Rückgabe

Keine

Fehler

Keine

SOAP API Methoden: Zielzellen

Zielzellen sind eine Abstraktion für einige bekannte Teilmengen von Kampagnenergebnissen, die von Unica Campaign als Target Cell Spreadsheet (TCS) verwaltet werden. Zielzellen können in einer Kampagne global sein oder einem bestimmten Kampagnenablaufdiagramm zugeordnet sein.

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

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

Zielzellen verfügen über einige Standardattribute, die ihnen zugeordnet sind und von der API angezeigt werden. Diese Liste kann bei Bedarf vom Client erweitert werden, indem Metadatendefinitionen für benutzerdefinierte Attribute hinzugefügt werden (siehe die

Attribut-APIs). Alle Metadaten für Attribute können als eine Spalte im TCS betrachtet werden; das Layout des Arbeitsblatt bestimmt der Client.

Standardzielzellenattribute sind:

- *uacName*: Zellenname.
- *uacDescription*: optionale Zeichenfolge, die das Ablaufdiagramm beschreibt.
- *uacCellCode*: eine Codezeichenfolge, die die Zelle eindeutig definiert. Wird normalerweise von Unica Campaign automatisch generiert, kann jedoch vom Client bereitgestellt werden.
- *uacCreateDate*: Eine Kalenderinstanz, die das Datum und die Uhrzeit angibt, zu der die Zelle vom Server erstellt wurde.
- *uacUpdateDate*: Eine Kalenderinstanz, die definiert, wann die Zelle das letzte Mal vom Server aktualisiert wurde.
- *uacIsControl*: ein boolescher Wert, der angibt, ob es sich um eine Kontrollzelle (true) oder nicht (false) handelt. Andere Zellen können diese Zelle als Kontrollzelle bezeichnen (siehe *uacControlCell*).
- *uacControlCell*: optionale Referenz der Kontrollzelle (bei einer Kontrollzelle nicht zulässig). Siehe das Attribut *uacIsControl*.
- *uacIsApproved*: Ein Boolescher Wert, der angibt, ob die Zelle genehmigt ist (true) oder nicht (false).
- *uacIsReadOnly*: Ein Boolescher Wert, der angibt, ob die Zelle schreibgeschützt ist (true) oder nicht (false).
- *uacDisplayOrder*: Eine Ganzzahl, die die Reihenfolge dieser Zelle (Zeile) relativ zu anderen Zellen in der Zielzellen-Tabelle angibt.
- *uacIsTopDown*: ein boolescher Wert, der angibt, ob es sich bei der Zelle um eine Top-down-Zelle handelt.
- *uacAssignedOffers*: Ein optionales Array mit einer oder mehreren Referenzen von Angeboten oder Angebotslisten, die dieser Zelle zugewiesen sind (nicht zulässig, wenn eine Kontrollzelle vorhanden ist).
- *uacFlowchartName*: Optionaler Name des Flussdiagramms, mit dem diese Zelle verknüpft ist (schreibgeschützt - muss über die Unica Campaign GUI festgelegt werden; nicht zulässig, wenn eine Kontrollzelle vorhanden ist).

- *uacFlowchartId*: optionale Datenbank-ID des Ablaufdiagramms, mit dem diese Zelle verbunden ist (schreibgeschützt, wie oben angegeben).

createTargetCell

```

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

```

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

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

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

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

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

Parameter

`userCredential`: die Client-Benutzeranmeldeinformationen.

`requestedLocale`: optionales Gebietsschema für diese Anforderung.

`PartitionName`: Optionaler Name der zu verwendenden Kampagnenpartition.

`campaignReference`: Die Referenz der Kampagne, die die zu aktualisierende Zielzellen-Tabelle enthält. Kumuliert eine `InvalidComponentException`, wenn die Kampagne nicht vorhanden ist oder die Referenz ungültig ist.

Attributen: optionales Array von TCS-Attributen für die neue Zelle. Die Standardwerte des entsprechenden Zellenattributs werden von den einzelnen angegebenen Attributelementen überschrieben; andere bleiben unberührt. Es ist Sache des Kunden, die für die Zelle erforderlichen Attribute, ihre Typen usw. zu bestimmen. Akkumuliert eine `InvalidAttributeException`, wenn ein Problem mit einem angegebenen Attribut vorliegt.

Wenn Ausnahmen akkumuliert werden, löst diese Methode eine `CompositeException` aus und alle Erstellungen werden rückgängig gemacht. Die Liste der Ursachen der Ausnahme enthält eine Ausnahme für jedes Attribut, das den Fehler verursacht hat, und einen numerischen Index anstelle der Referenz, den Namen des Attributs und normalerweise den betreffenden Wert. Die Ursachenliste ist wie bei der Eingabe attributliste geordnet.

Rückgabe

Ein `TargetCellInfo` Daten Wrapper für die erstellte TCS-Zelle.

Fehler

`InvalidComponentException`, `CompositeException`

`AuthorizationException`, `DataException`

bulkCreateTargetCells

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

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

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

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

Siehe `listTargetCells()`.

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

Parameter

`userCredential`: die Client-Benutzeranmeldeinformationen.

`requestedLocale`: optionales Gebietsschema für diese Anforderung.

`PartitionName`: Optionaler Name der zu verwendenden Kampagnenpartition.

`campaignReference`: Die Referenz der Kampagne, die die zu aktualisierende Zielzellen-Tabelle enthält. Kumuliert eine `InvalidComponentException`, wenn die Kampagne nicht vorhanden ist oder die Referenz ungültig ist.

`attributeList`: Optionale Liste von Attribut-Arrays pro Zelle, eines für jede zu erstellende Zielzellenzeile. Alle für ein bestimmtes Listenelement bereitgestellten Attribute überschreiben die Standardwerte des entsprechenden Zellenattributs; andere bleiben unberührt. Es ist Sache des Kunden, die für die Zelle erforderlichen Attribute, ihre Typen usw. zu bestimmen. Akkumuliert eine `InvalidAttributeException`, wenn ein Problem mit einem angegebenen Attribut vorliegt.

Wenn Ausnahmen akkumuliert werden, löst diese Methode eine `CompositeException` aus und alle Erstellungen werden rückgängig gemacht. Die Liste der Ursachen der Ausnahme enthält eine Ausnahme für jedes Attribut, das den Fehler verursacht hat, sowie einen numerischen Index anstelle der Referenz und den Namen des Attributs usw. Die Ursachenliste ist wie bei der Eingabe `attributliste` geordnet.

Rückgabe

Eine Liste von TargetCellInfo-Daten-Wrappern, einer für jede erstellte Instanz, geordnet nach der Elementreihenfolge des EingabeattributsList-Parameters.

Fehler

InvalidComponentException, CompositeException

AuthorizationException, DataException

listTargetCells

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

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

Siehe `getAttributeMetadata()`, `getAttributesByName()`.

Parameter

`userCredential`: die Client-Benutzeranmeldeinformationen.

`requestedLocale`: optionales Gebietsschema für diese Anforderung.

`PartitionName`: Optionaler Name der zu verwendenden Kampagnenpartition.

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

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

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

Rückgabe

Gibt eine Liste mit null oder mehr `TargetCellInfo` Instanzen für die übereinstimmenden Zellen zurück.

Fehler

`InvalidComponentException`, `InvalidAttributeException`

`AuthorizationException`, `DataException`

bulkUpdateTargetCells

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

Aktualisierung der Attribute von mindestens einer Zielzelle.

Die Aktualisierungslogik lautet wie folgt.

Für jedes Element in der angegebenen `AttributeMap` ist der Eingabeschlüssel die Referenz der zu aktualisierenden Zielzelle, und der Eingabewert ist ein Array von Aktualisierungsattributen für diese Zelle. Wenn die Zielzelle nicht vorhanden ist, akkumulieren Sie eine `InvalidComponentException`.

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

1. Wenn der Attributname mit einem vorhandenen Attribut übereinstimmt, versuchen Sie, das Wertefeld mit dem angegebenen Wertefeld zu überschreiben.
2. Wenn der Wertetyp oder ein anderer Aspekt der Metadatendefinition des Attributs nicht erfüllt ist oder einer oder mehrere der angegebenen Werte ungültig sind oder außerhalb des Bereichs liegen, akkumulieren Sie eine *InvalidAttributeException*.
3. Andernfalls akkumulieren Sie *AttributeNotFoundException*, wenn das benannte Attribut nicht vorhanden ist.

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

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

Parameter

`userCredential`: die Client-Benutzeranmeldeinformationen.

`requestedLocale`: optionales Gebietsschema für diese Anforderung.

`PartitionName`: Optionaler Name der zu verwendenden Kampagnenpartition.

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

Rückgabe

Keine.

Fehler

ComponentException

AuthorizationException, DataException

getRunResultsByCell

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

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

Parameter

userCredential: die Client-Benutzeranmeldeinformationen.

requestedLocale: optionales Gebietsschema für diese Anforderung.

PartitionName: Optionaler Name der zu verwendenden Kampagnenpartition.

cellReferences: ein Array mit Referenzen der Zielzellen, deren Ausführungsergebnisse erwünscht sind. Löst eine InvalidComponentException aus, wenn eine oder mehrere der Zellreferenzen ungültig sind oder auf eine nicht vorhandene Zelle verweisen.

Rückgabe

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

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

Fehler

InvalidComponentException

AuthorizationException, DataException

bulkDeleteTargetCells

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

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

Parameter

userCredential: die Client-Benutzeranmeldeinformationen.

requestedLocale: optionales Gebietsschema für diese Anforderung.

PartitionName: Optionaler Name der zu verwendenden Kampagnenpartition.

cellReferences: ein Array oder eine oder mehrere Referenzen von Zellen, die gelöscht werden sollen. InvalidComponentException wird kumuliert, wenn bei einer der angegebenen Referenzen ein Problem vorliegt oder eine Zelle nicht vorhanden ist.

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

Rückgabe

Keine.

Fehler

CompositeException

AuthorizationException, DataException

updateTemplateAttributes

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

Aktualisiert Attribute angegebener Vorlagen

Parameter

userCredential: die Client-Benutzeranmeldeinformationen.

PartitionName: Optionaler Name der zu verwendenden Kampagnenpartition.

requestedLocale: optionales Gebietsschema für diese Anforderung.

wsCampaignReference: Referenz der übergeordneten Kampagne.

allowCreate: wird jetzt nicht verwendet.

clearExisting: Wenn das Flag auf true gesetzt ist, werden alle früheren Werte in der Vorlage gelöscht, die nicht in der Anforderung gesendet wurden.

wsStaticAttributes: Liste der statischen Attribute in der Vorlage.

wsHiddenAttributes: Liste der versteckten Attribute in der Vorlage.

wsParametricAttributes: Liste der parametrischen Attribute in der Vorlage.

Rückgabe

Keine.

Fehler

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

listBottomUpTargetCells

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

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

Parameter

userCredential: die Client-Benutzeranmeldeinformationen.

PartitionName: Optionaler Name der zu verwendenden Kampagnenpartition.

requestedLocale: optionales Gebietsschema für diese Anforderung.

wsCampaignReference: Referenz der übergeordneten Kampagne.

Rückgabe

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

Fehler

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

SOAP API Methoden: Analyse

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

getCampaignMetrics

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

Ruft die Metrik der angegebenen Kampagne ab.

Parameter

`userCredential`: die Client-Benutzeranmeldeinformationen.

`requestedLocale`: optionales Gebietsschema für diese Anforderung.

`PartitionName`: Optionaler Name der zu verwendenden Kampagnenpartition.

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

Rückgabe

Gibt eine `MetricsInfo` -Instanz für die Kampagne zurück.

Fehler

`InvalidComponentException`

`AuthorizationException`, `DataException`

SOAP API Methoden: Angebote, Angebotslisten, Angebotsvorlagen

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

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

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

Standardattribute von Angeboten sind:

- *uacName*: Angebotsname
- *uacDescription*: optionale Zeichenfolge, die das Angebot beschreibt.
- *uacOfferCode*: Eine Codezeichenfolge, die das Angebot eindeutig identifiziert. Wird normalerweise von Unica Campaign generiert, kann jedoch vom Client bereitgestellt werden.
- *uacCreateDate*: Eine Kalenderinstanz, die das Datum und die Uhrzeit angibt, zu der das Angebot vom Server erstellt wurde.
- *uacUpdateDate*: Eine Kalenderinstanz, die angibt, wann das Angebot zuletzt vom Server aktualisiert wurde.

Angebotsvorlagen verfügen auch über Standardattribute und benutzerdefinierte Attribute.

Standardattribute von Angebotsvorlagen sind:

- *uacName*: Name der Angebotsvorlage.
- *uacDescription*: optionale Zeichenfolge, die die Angebotsvorlage beschreibt.

- *uacCreateDate*: Eine Kalenderinstanz, die das Datum und die Uhrzeit angibt, zu der die Angebotsvorlage vom Server erstellt wurde.
- *uacUpdateDate*: Eine Kalenderinstanz, die angibt, wann die Angebotsvorlage zuletzt vom Server aktualisiert wurde.

listOffersAndFolders

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

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

Nach dem Abrufen kann jede zurückgegebene *WSComponentOrFolderInfo* Instanz unverändert verwendet werden, um beispielsweise die nächste Ebene der Ordnerhierarchie anzuzeigen. Die Attribut-APIs können zum Abrufen oder Aktualisieren enthaltener Angebote verwendet werden.

Parameter

userCredential: die Client-Benutzeranmeldeinformationen.

requestedLocale: optionales Gebietsschema für diese Anforderung.

PartitionName: Optionaler Name der zu verwendenden Kampagnenpartition.

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

Löst eine `InvalidFolderException` aus, wenn ein Problem mit der angegebenen übergeordneten Ordnerreferenz vorliegt.

Eine typisierte Liste mit null oder mehr `WSComponentOrFolder Info-Daten-Wrapper`-Instanzen, eine für jede übereinstimmende Komponente oder jeden Ordner.

Fehler

`InvalidFolderException`

`InvalidExecutionContextException`, `AuthorizationException`

searchOffersBasic

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

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

Nach dem Abrufen kann jede zurückgegebene `WSOfferInfo` unverändert verwendet werden, um beispielsweise eine Zusammenfassungsverzeichnisliste anzuzeigen, oder die Attributmethoden können zum Abrufen oder Aktualisieren der Attribute des Angebots verwendet werden.

Diese API verwaltet keinen Status, sodass Sie sie in beliebiger Reihenfolge aufrufen können.

Parameter

`userCredential`: die Client-Benutzeranmeldeinformationen.

requestedLocale: optionales Gebietsschema für diese Anforderung.

PartitionName: Optionaler Name der zu verwendenden Kampagnenpartition.

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

searchCriteria: der Suchbegriff

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

pageOffset: der Startversatz aller möglichen Komponenten, um die Aufzählung zu beginnen (Nullwert). Beispiel: Wenn bei der Aufzählung 1000 Übereinstimmungen mit Angeboten gefunden werden und dieser Wert auf 10 festgelegt ist, würde auf der Seite mit der 11. Komponente begonnen werden. Eine RangeException wird ausgelöst, wenn der angegebene Offset außerhalb des Bereichs liegt.

pageSize: Die maximale Anzahl übereinstimmender Komponenten, die für die Seite zurückgegeben werden sollen (darf 500 nicht überschreiten).

Rückgabe

Gibt eine typisierte Liste mit null oder mehr Angebot daten-Wrapper Instanzen zurück, eine für jedes zurückgegebene Angebot auf der Seite.

Fehler

RangeException

listOffersByPage

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

```
throws AttributeNotFoundException, InvalidAttributeException,
    RangeException;
```

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

Nach dem Abrufen kann jede zurückgegebene OfferInfo unverändert verwendet werden, um beispielsweise eine Zusammenfassungsverzeichnisliste anzuzeigen, oder die Attributmethoden können zum Abrufen oder Aktualisieren der Attribute des Angebots verwendet werden.

Diese API verwaltet keinen Status, sodass Sie sie in beliebiger Reihenfolge aufrufen können.

Parameter

userCredential: die Client-Benutzeranmeldeinformationen.

requestedLocale: optionales Gebietsschema für diese Anforderung.

PartitionName: Optionaler Name der zu verwendenden Kampagnenpartition.

Attributen: optionales Array von passenden Attributen; Der Name, der Datentyp und die Werte des Attributs werden verwendet, um die Übereinstimmung zu bestimmen. Wenn das Attribut Arrays unterstützt, müssen alle angegebenen Werte übereinstimmen. Der implizierte Übereinstimmungsoperator lautet OR. Folglich werden Komponenten zurückgegeben, die mit einem beliebigen angegebenen Attributwert übereinstimmen.

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

pageOffset: der Startversatz aller möglichen Komponenten, um die Aufzählung zu beginnen (nullwertig). Beispiel: Wenn bei der Aufzählung 1000 Übereinstimmungen mit Angeboten gefunden werden und dieser Wert auf 10 festgelegt ist, würde auf der Seite mit der 11.

Komponente begonnen werden. Eine RangeException wird ausgelöst, wenn der angegebene Offset außerhalb des Bereichs liegt.

pageSize: Die maximale Anzahl übereinstimmender Komponenten, die für die Seite zurückgegeben werden sollen (darf 500 nicht überschreiten).

Rückgabe

Eine typisierte Liste mit null oder mehr OfferInfo Daten-Wrapper Instanzen, eine für jede übereinstimmende Komponente auf der Seite.

Fehler

AttributeNotFoundException, InvalidAttributeException, RangeException

InvalidExecutionContextException, AuthorizationException

createSmartOfferList

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

Erstellt eine neue dynamische Angebotsliste.

Parameter

userCredential: die Client-Benutzeranmeldeinformationen.

PartitionName: Optionaler Name der zu verwendenden Kampagnenpartition.

requestedLocale: optionales Gebietsschema für diese Anforderung.

name: Name des zu erstellenden Ordners.

Beschreibung: Beschreibung des neuen Ordners.

securityPolicyName: Name der zu verwendenden Sicherheitsrichtlinie.

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

offerListInfo: Objekt vom Typ WSSmartListInfo.

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

creatorObjectId: (optional) Wird von Plan (HCL-Plan) verwendet, um einen Ordner in Campaign mit einem Ordner in Plan zu verknüpfen.

Rückgabe

Gibt ein Objekt vom Typ WSCreateSmartOfferListResponse zurück.

Fehler

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

Löst CampaignServicesException aus, wenn der Listenname doppelt vorhanden ist.

createStaticOfferList

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

Erstellt eine neue statische Angebotsliste.

Parameter

userCredential: die Client-Benutzeranmeldeinformationen.

PartitionName: Optionaler Name der zu verwendenden Kampagnenpartition.

requestedLocale: optionales Gebietsschema für diese Anforderung.

name: Name des zu erstellenden Ordners.

Beschreibung: Beschreibung des neuen Ordners.

securityPolicyName: Name der zu verwendenden Sicherheitsrichtlinie.

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

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

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

creatorObjectId: (optional) Wird von Plan (HCL-Plan) verwendet, um einen Ordner in Campaign mit einem Ordner in Plan zu verknüpfen.

Rückgabe

Gibt ein Objekt vom Typ `WSCreateStaticOfferListResponse` zurück

Fehler

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

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

getOffers

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

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

Parameter

userCredential: die Client-Benutzeranmeldeinformationen.

PartitionName: Optionaler Name der zu verwendenden Kampagnenpartition.

requestedLocale: optionales Gebietsschema für diese Anforderung.

wsCampaignReference: Referenz der übergeordneten Kampagne.

Rückgabe

Gibt ein Objekt vom Typ WSGetOffersResponse zurück.

Fehler

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

validateOffers

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

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

Das Objekt OfferValidationInfo enthält anstelle von Angebotsinformationen eine Fehlernachricht, wenn null Angebote oder Angebotslisten gefunden werden, die mit dem angegebenen Code oder Namen übereinstimmen. Ein Fehler wird auch dann anstelle einer Übereinstimmung zurückgegeben, wenn der angegebene Code oder Name mit mehreren Angeboten oder Angebotslisten übereinstimmt. Die Liste wird in derselben angegebenen

Reihenfolge zurückgegeben. Die Angebotscodes und Namen der Angebotsliste werden basierend auf einer genauen Übereinstimmung mit Angeboten überprüft.

Parameter

`userCredential`: die Client-Benutzeranmeldeinformationen.

`requestedLocale`: optionales Gebietsschema für diese Anforderung.

`PartitionName`: Optionaler Name der zu verwendenden Kampagnenpartition.

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

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

Rückgabe

Eine typisierte Liste mit null oder mehr `OfferValidationInfo` Daten-Wrapper Instanzen.

Fehler

Keine.

editOfferList

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

Aktualisiert Details zur dynamischen und statischen offerlist.

Parameter

`userCredential`: die Client-Benutzeranmeldeinformationen.

`PartitionName`: Optionaler Name der zu verwendenden Kampagnenpartition.

`requestedLocale`: optionales Gebietsschema für diese Anforderung.

`listReference`: Referenz auf Angebotsliste.

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

`name`: Name des zu erstellenden Ordners.

`Beschreibung`: Beschreibung des neuen Ordners.

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

`offerListInfo`: Objekt vom Typ `WSSmartListInfo`.

`creatorObjectId`: (optional) Wird von Plan (HCL-Plan) verwendet, um einen Ordner in Campaign mit einem Ordner in Plan zu verknüpfen.

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

Rückgabe

Gibt ein Objekt vom Typ `WSEditOfferListResponse` zurück.

Fehler

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

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

createOffer

```
OfferInfo createOffer(String userCredential, String partitionName,  
                      Locale requestedLocale,  
                      String securityPolicyName,
```

```

        String name, String templateName,
        Attribute[] attributes)
    throws InvalidFolderException, AttributeNotFoundException,
        InvalidAttributeException;

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

```

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

Parameter

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

PartitionName: Optionaler Name der zu verwendenden Kampagnenpartition.

requestedLocale: optionales Gebietsschema für diese Anforderung.

securityPolicyName: Optionaler Name der Kampagnen-Sicherheitsrichtlinie, die zum Erstellen des Angebots verwendet werden soll. Bei allen nachfolgenden Vorgängen in diesem Angebot wird diese Richtlinie verwendet. Ist keine Sicherheitsrichtlinie definiert, wird die globale Richtlinie verwendet.

name: Der Name, der der neuen Angebotsinstanz zugewiesen werden soll (das Attribut `uacName`).

folderID: Die ID des Angebotsordners, in dem die Angebote erstellt werden. Diese ID wird auf Richtigkeit überprüft. Wenn sie ungültig ist, wird eine Ausnahme ausgelöst.

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

`wsAttributes`: ein Array von Initialisierungsattributen; Alle angegebenen Attribute überschreiben die Standardwerte des Angebots. andere bleiben unberührt. Wenn beispielsweise ein `uacOfferCode` Attribut angegeben wird, wird es anstelle eines automatisch generierten Attributs verwendet. Es ist Aufgabe des Clients, die für das Angebot erforderlichen Attribute, den Typ usw. zu ermitteln.

Löst eine `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, diesen Vorgang durchzuführen.
- In "wsAttributes" werden ungültige Attribute angegeben.
- Es treten andere Laufzeitausnahmebedingungen auf.

Rückgabe

Eine Einzelinstanz von `OfferInfo` für das erstellte Angebot.

Fehler

`CampaignServicesException`

retireOffers

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

Zieht mindestens ein bestehendes Angebot zurück.

Parameter

`userCredential`: die Client-Benutzeranmeldeinformationen.

`requestedLocale`: optionales Gebietsschema für diese Anforderung.

`PartitionName`: Optionaler Name der zu verwendenden Kampagnenpartition.

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

Rückgabe

Keine.

Fehler

InvalidComponentException

AuthorizationException, DataException

deleteOffers

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

Löscht mindestens ein vorhandenes Angebot.

Parameter

userCredential: die Client-Benutzeranmeldeinformationen.

requestedLocale: optionales Gebietsschema für diese Anforderung.

PartitionName: Optionaler Name der zu verwendenden Kampagnenpartition.

referenz: eine Reihe von Referenzen der zu löschenden Angebote.

InvalidComponentException wird ausgelöst, wenn ein Problem mit einer angegebenen Referenz vorliegt oder kein Angebot vorhanden ist.

Rückgabe

Keine.

Fehler

InvalidComponentException

AuthorizationException, DataException

deleteOffersAndLists

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

Löscht die angegebenen Angebote und Listen.

Parameter

userCredential: die Client-Benutzeranmeldeinformationen.

PartitionName: Optionaler Name der zu verwendenden Kampagnenpartition.

requestedLocale: optionales Gebietsschema für diese Anforderung.

Angebote: Array von Referenzen entweder Angebot oder Angebotsliste.

Rückgabe

Gibt ein Objekt vom Typ WSGetOfferListMembersResponse zurück.

Fehler

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

listOfferTemplates

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

```
throws CampaignServicesException;
```

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

Nach dem Abrufen kann jede zurückgegebene WSOfferTemplateInfo Instanz unverändert verwendet werden, oder eine oder mehrere der Attribut-APIs können zum Abrufen oder Aktualisieren einer aufgelisteten Vorlage verwendet werden.

Parameter

userCredential: die Client-Benutzeranmeldeinformationen.

requestedLocale: optionales Gebietsschema für diese Anforderung.

PartitionName: Optionaler Name der zu verwendenden Kampagnenpartition.

Rückgabe

Eine typisierte Liste mit null oder mehr WSOfferTemplateInfo Daten-Wrapper Instanzen, eine für jede zurückgegebene Vorlage.

Fehler

InvalidExecutionContextException, AuthorizationException

DataException

createTemplate

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

Erstellt eine neue Angebotsvorlage.

Parameter

userCredential: die Client-Benutzeranmeldeinformationen.

PartitionName: Optionaler Name der zu verwendenden Kampagnenpartition.

requestedLocale: optionales Gebietsschema für diese Anforderung.

name: Name der neu erstellten Angebotsvorlage.

securityPolicyName: Name der zu verwendenden Sicherheitsrichtlinie.

wsStaticAttributes: Liste der statischen Attribute in der Vorlage.

wsHiddenAttributes: list of hidden attributes in template.

wsParametricAttributes: Liste der parametrischen Attribute in der Vorlage.

Rückgabe

Gibt ein Objekt vom Typ `WSCreateTemplateResponse` zurück.

Fehler

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

getOfferTemplate

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

Listet Details zu Angebotsvorlagen entsprechend der Angabe in Referenzen auf.

Parameter

userCredential: die Client-Benutzeranmeldeinformationen.

PartitionName: Optionaler Name der zu verwendenden Kampagnenpartition.

requestedLocale: optionales Gebietsschema für diese Anforderung.

wsCampaignReference: Referenz der übergeordneten Kampagne.

Rückgabe

Gibt ein Objekt vom Typ `WSGetOfferTemplateResponse` zurück.

Fehler

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

retireOfferTemplates

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

Zieht mindestens eine angegebene Angebotsvorlage zurück.

Parameter

`userCredential`: die Client-Benutzeranmeldeinformationen.

`PartitionName`: Optionaler Name der zu verwendenden Kampagnenpartition.

`requestedLocale`: optionales Gebietsschema für diese Anforderung.

`wsCampaignReference`: Referenz der übergeordneten Kampagne.

Rückgabe

Gibt ein Objekt vom Typ `WSRetireOfferTemplatesResponse` zurück.

Fehler

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

getOffersAndListsByPage

```
public WSGetOffersAndListsByPageResponse getOffersAndListsByPage
```

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

oder Angebotslisten nach Seite aus.

Parameter

`userCredential`: die Client-Benutzeranmeldeinformationen.

`PartitionName`: Optionaler Name der zu verwendenden Kampagnenpartition.

`requestedLocale`: optionales Gebietsschema für diese Anforderung.

`typ`: Typ, der angibt, ob Angebote oder Angebotslisten angefordert wurden.

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

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

Rückgabe

Gibt ein Objekt vom Typ `WSGetOffersAndListsByPageResponse` zurück.

Fehler

Keine.

bulkCreateOffers

```
WSOfferInfoStatus[] bulkCreateOffers(String authorizationLoginName,
    String partitionName, Locale requestedLocale,
```

```
String securityPolicyName, String templateName, long folderID,
WSBulkOfferInfo[] offers)
throws CampaignServicesException;
```

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

Parameter

authorizationLoginName: die Client-Benutzeranmeldeinformationen.

PartitionName: Optionaler Name der zu verwendenden Kampagnenpartition.

requestedLocale: optionales Gebietsschema für diese Anforderung.

securityPolicyName: Optionaler Name der Kampagnen-Sicherheitsrichtlinie, die zum Erstellen des Angebots verwendet werden soll. Ist keine Sicherheitsrichtlinie definiert, wird die globale Richtlinie verwendet.

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

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

Angebote: Ein Array von WSBulkOfferInfo, das den Angebotsnamen und die Attribute definiert. Weitere Einzelheiten finden Sie beim Datentyp WSBulkOfferInfo.

Rückgabe

Ein Array von WSOfferInfoStatus Instanzen für jedes Angebot. Enthält den Status und Angebotsinformationen. Der Status zeigt an, ob die Angebotserstellung erfolgreich oder nicht erfolgreich war.

Fehler

CampaignServicesException

getOfferListDetails

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

Listet Details zur angegebenen Angebotsliste auf.

Parameter

userCredential: die Client-Benutzeranmeldeinformationen.

PartitionName: Optionaler Name der zu verwendenden Kampagnenpartition.

requestedLocale: optionales Gebietsschema für diese Anforderung.

listReference: Referenz auf Angebotsliste.

Rückgabe

Gibt ein Objekt vom Typ WSGetOfferListDetailsResponse zurück.

Fehler

Löst CampaignServicesException aus, wenn Listenreferenzen ungültig sind.

getOfferListMembers

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

Listet Angebotsinformationen in der angegebenen Angebotsliste auf.

Parameter

userCredential: die Client-Benutzeranmeldeinformationen.

PartitionName: Optionaler Name der zu verwendenden Kampagnenpartition.

requestedLocale: optionales Gebietsschema für diese Anforderung.

listReference: Referenz auf Angebotsliste.

Rückgabe

Gibt ein Objekt vom Typ `WSDeleteOffersAndListsResponse` zurück.

Fehler

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

getOffersByQuery

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

Mit bereitgestellten Angeboten übereinstimmende Listenangebote.

Parameter

userCredential: die Client-Benutzeranmeldeinformationen.

PartitionName: Optionaler Name der zu verwendenden Kampagnenpartition.

requestedLocale: optionales Gebietsschema für diese Anforderung.

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

maxSize: Maximale Anzahl der aufzulistenden Datensätze.

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

scopeFolders: Liste der Ordnerreferenzen zum Durchsuchen bei Angeboten.

Rückgabe

Gibt ein Objekt vom Typ `WSGetOffersByQueryResponse` zurück.

Fehler

Löst `CampaignServicesException` aus, wenn Ordnerreferenzen ungültig sind.

retireOfferLists

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

Zieht mindestens eine angegebene Angebotsliste zurück.

Parameter

`userCredential`: die Client-Benutzeranmeldeinformationen.

`PartitionName`: Optionaler Name der zu verwendenden Kampagnenpartition.

`requestedLocale`: optionales Gebietsschema für diese Anforderung.

`wsReferences`: Referenz auf Angebotslisten.

Rückgabe

Keine.

Fehler

Löst `CampaignServicesException` aus, wenn Listenreferenzen ungültig sind.

createFolder

```
public WSCreateFolderResponse createFolder(String userCredential,  
String partitionName, Locale requestedLocale, String name,  
String description, String securityPolicyName,
```

```

long parentFolderId,WSFolderTypeEnum folderType,
WSApplicationTypeEnum createdBy,long creatorObjectId)
throws CampaignServicesException

```

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

Parameter

userCredential: die Client-Benutzeranmeldeinformationen.

PartitionName: Optionaler Name der zu verwendenden Kampagnenpartition.

requestedLocale: optionales Gebietsschema für diese Anforderung.

name: Name des zu erstellenden Ordners.

Beschreibung: Beschreibung des neuen Ordners.

securityPolicyName: Name der zu verwendenden Sicherheitsrichtlinie.

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

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

createdBy: (optional) Objekt vom Typ WSApplicationTypeEnum gibt an, mit welcher Anwendung der Ordner erstellt wurde. Die möglichen Werte sind Unica Campaign, Unica Plan, Collaborate und Unica Deliver. Wenn nicht angegeben, wird Unica Campaign verwendet.

 **Anmerkung:** Plan=Unica Plan. Collaborate = Verteiltes Marketing.

creatorObjectId: (optional) Wird von Plan (Unica Plan) verwendet, um einen Ordner in Unica Campaign mit einem Ordner in Unica Plan zu verknüpfen.

Rückgabe

Gibt ein Objekt vom Typ WSCreateFolderResponse zurück.

Fehler

Löst CampaignServicesException aus, wenn der Ordnerartyp ungültig ist oder doppelt vorliegt.

editFolder

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

Aktualisiert den angegebenen Ordner.

Parameter

userCredential: die Client-Benutzeranmeldeinformationen.

PartitionName: Optionaler Name der zu verwendenden Kampagnenpartition.

requestedLocale: optionales Gebietsschema für diese Anforderung.

id: ID des zu aktualisierenden Ordners.

name: Name des zu erstellenden Ordners.

Beschreibung: Beschreibung des neuen Ordners.

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

creatorObjectId: (optional) Wird von Plan (HCL-Plan) verwendet, um einen Ordner in Campaign mit einem Ordner in Plan zu verknüpfen.

clearExisting: wird jetzt nicht verwendet.

Rückgabe

Gibt ein Objekt vom Typ WSEditFolderResponse zurück.

Fehler

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

getSubFoldersList

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

Listet alle Unterordner des angegebenen Ordners auf.

Parameter

userCredential: die Client-Benutzeranmeldeinformationen.

PartitionName: Optionaler Name der zu verwendenden Kampagnenpartition.

requestedLocale: optionales Gebietsschema für diese Anforderung.

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

folderType: Ordnerart

Rückgabe

Gibt ein Objekt vom Typ WSGetSubFolderListResponse zurück.

Fehler

Löst CampaignServicesException aus, wenn eine Ordnerreferenz ungültig ist.

moveFolders

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

```
throws CampaignServicesException
```

Verschiebt einen angegebenen Ordner in einen anderen übergeordneten Ordner.

Parameter

userCredential: die Client-Benutzeranmeldeinformationen.

PartitionName: Optionaler Name der zu verwendenden Kampagnenpartition.

requestedLocale: optionales Gebietsschema für diese Anforderung.

folderId: zu löschende Ordner-ID.

parentFolder: ID des übergeordneten Ordners.

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

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

Rückgabe

Gibt ein Objekt vom Typ WSMoveFolderResponse zurück.

Fehler

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

deleteFolders

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

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

Parameter

userCredential: die Client-Benutzeranmeldeinformationen.

PartitionName: Optionaler Name der zu verwendenden Kampagnenpartition.

requestedLocale: optionales Gebietsschema für diese Anforderung.

folderId: zu löschende Ordner-ID.

parentFolder: ID des übergeordneten Ordners.

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

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

Rückgabe

Gibt ein Objekt vom Typ WSDeleteFolderResponse zurück.

Fehler

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

Kapitel 6. Allgemeine SOAP-API-Ausnahmen

Die SOAP-API von Unica Campaign löst unter Umständen die folgenden allgemeinen Ausnahmen aus. Alle lokalisierten Ausnahmemeldungen befinden sich im angeforderten Gebietsschema, sofern sie für Unica Campaign verfügbar sind. Es gelten die üblichen Java Fallback-Richtlinien für das Gebietsschema.

RemoteException

Dieses Element gilt nur für die SOAP-Schnittstelle.

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

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

Weitere Angaben enthält der Abschnitt zur SOAP-Schnittstelle.

AuthenticationException

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

AuthorizationException

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

DataException

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

Überprüfen Sie das Unica Campaign Ablaufdiagramm und die Listener-Protokolle auf Details.

LockException

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

InvalidComponentException

Es wurde versucht, auf eine ungültige oder unbekannte Komponente (Kampagne, Ablaufdiagramm, Zielzelle) zu verweisen. Die `getComponentReference()` Methode der Ausnahme gibt die Referenz der betreffenden Komponente zurück.

InvalidAttributeException

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

AttributeExistsException

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

AttributeNotFoundException

Wird ausgelöst, wenn der Client versucht, auf ein unbekanntes Attribut (Kampagne, Flussdiagramm, Zielzelle usw.) zu verweisen. Die `getAttributeName()` Methode

der Ausnahme gibt den Namen des nicht übereinstimmenden Attributs zurück.

`GetComponentReference()` identifiziert die Komponente (oder den Massenindex).

CompositeException

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

 **Anmerkung:** Bevor sie eine `CompositeException` ausgibt, schließt die API ihre Arbeit entweder erfolgreich ab oder macht ihre Arbeit rückgängig. Siehe hierzu z. B. die Massen-APIs für das Arbeitsblatt für Zielzellen im Abschnitt [SOAP API Methoden: Zielzellen \(auf Seite 51\)](#).