

**IBM Campaign**  
버전 9 릴리스 0  
2013년 1월 15일

**오픈 API 스펙**

**IBM**

참고

이 정보와 이 정보가 지원하는 제품을 사용하기 전에, 57 페이지의 『주의사항』의 정보를 읽으십시오.

이 개정판은 새 개정판에 별도로 명시하지 않는 한, IBM Campaign 버전 9, 릴리스 0, 수정 0 및 모든 후속 릴리스와 수정에 적용됩니다.

© Copyright IBM Corporation 1998, 2013.

# 목차

제 1 장 소개 . . . . .	1
높은 레벨 요약 . . . . .	1
클라이언트 API를 사용하여 기존 구현 . . . . .	2
새 클라이언트 API .jar 파일 . . . . .	2
새 종속 .jar 파일 . . . . .	2
클라이언트 API 생성자의 변경사항 . . . . .	2
지원 클래스의 매개변수화된 생성자에 대한 변경사항 . . . . .	3
WSDL을 직접 사용하여 기존 구현 . . . . .	3
WSDL 위치 및 서비스 URL . . . . .	3
스텝 및 클래스 생성 . . . . .	3
생성된 스텝 사용 및 클래스 지원 . . . . .	4
참조 . . . . .	5
요구사항 . . . . .	5
디자인 개요 . . . . .	5
참고 . . . . .	6
제 2 장 데이터 유형 . . . . .	9
WSReference . . . . .	9
WSVersion . . . . .	9
WSServiceInfo . . . . .	9
WSAttributeTypeEnum . . . . .	10
WSAttributeStatusEnum . . . . .	10
WSAccessTypeEnum . . . . .	10
WSSelectTypeEnum . . . . .	10
WSRunStatusEnum . . . . .	11
WSRunTypeEnum . . . . .	11
WSAttribute . . . . .	11
WSAttributeMetadata . . . . .	13
WSCampaignInfo . . . . .	14
WSComponentOrFolderInfo . . . . .	15
WSTargetCellInfo . . . . .	15
WSMetricsInfo . . . . .	15
WSRunResults . . . . .	16
WSOfferInfo . . . . .	16
WSOfferCodeOrName . . . . .	17
WSOfferValidationInfo . . . . .	17
WSOfferTemplateInfo . . . . .	17
WSBulkOfferInfo . . . . .	17
WSOfferInfoStatus . . . . .	18
제 3 장 공통 예외 . . . . .	19
RemoteException . . . . .	19

AuthenticationException . . . . .	19
AuthorizationException . . . . .	19
DataException . . . . .	19
LockException . . . . .	20
InvalidComponentException . . . . .	20
InvalidAttributeException . . . . .	20
AttributeNotFoundException . . . . .	20
AttributeExistsException . . . . .	20
CompositeException . . . . .	20
제 4 장 CampaignServices API 메소드 . . . . .	23
서비스 메소드 . . . . .	23
getServiceInfo . . . . .	23
속성 . . . . .	23
getAttributesByName . . . . .	24
updateAttributes . . . . .	24
getAttributeMetadataByName . . . . .	26
createAttributeMetadata . . . . .	27
updateAttributeMetadata . . . . .	27
deleteAttributeMetadata . . . . .	29
캠페인 . . . . .	30
generateCampaignCode . . . . .	31
createCampaign . . . . .	32
listCampaignsByPage . . . . .	33
대상 셀 메소드 . . . . .	34
createTargetCell . . . . .	35
bulkCreateTargetCells . . . . .	36
listTargetCells . . . . .	37
bulkUpdateTargetCells . . . . .	38
getRunResultsByCell . . . . .	39
bulkDeleteTargetCells . . . . .	40
분석 . . . . .	41
getCampaignMetrics . . . . .	41
오퍼, 오퍼 목록 및 오퍼 템플릿 메소드 . . . . .	41
listOffersAndFolders . . . . .	42
searchOffersBasic . . . . .	43
listOffersByPage . . . . .	44
validateOffers . . . . .	45
createOffer . . . . .	46
retireOffers . . . . .	47
deleteOffers . . . . .	48
listOfferTemplates . . . . .	48

bulkCreateOffers . . . . .	49	패키징 . . . . .	54
<b>제 5 장 API 사용.</b> . . . . .	<b>51</b>	<b>IBM 기술 지원 담당자에게 문의</b> . . . . .	<b>55</b>
클라이언트 API jar 사용 . . . . .	51	주의사항 . . . . .	57
코드 OfferAPI.java . . . . .	51	상표. . . . .	59
WSDL 사용 . . . . .	53		
성과 고려사항 . . . . .	54		

---

## 제 1 장 소개

**참고:** 이 문서는 CampaignServices라고도 불리는 IBM® Campaign Services Application Programming Interface 버전 3.0의 오피 부분을 정의합니다. 오피 서비스는 이 안내서에서 설명된 것만 지원됩니다.

**참고:** IBM Campaign 버전 8.2 이상으로 업그레이드할 예정이고 현재 구현된 IBM Campaign Services API가 있는 경우 AXIS 버전 1.3에서 AXIS2 1.4.1로 업그레이드하는 데 필요한 API 변경은 애플리케이션 코드 변경이 필요합니다. 자세한 내용은 2 페이지의 『새 종속 .jar 파일』을 참조하십시오.

이 절에는 다음과 같은 핵심 주제가 포함됩니다.

- 『높은 레벨 요약』
- 2 페이지의 『클라이언트 API를 사용하여 기존 구현』
- 3 페이지의 『WSDL을 직접 사용하여 기존 구현』
- 5 페이지의 『참조』
- 5 페이지의 『요구사항』
- 5 페이지의 『디자인 개요』

---

### 높은 레벨 요약

- AXIS 버전 1.3에서 AXIS2 1.4.1로 SOAP 엔진 마이그레이션.
- WSDL은 필수/선택적 매개변수를 처리하는데 발생하는 문제를 해결하기 위해 재구성되었습니다.
- 클라이언트 API .jar 파일이 WSDL 수정 결과로 변경되었고 또한 생성된 스텝 및 클래스도 변경되었습니다. 클라이언트 API 메소드 매개변수는 변경되지 않았지만 AXIS2 WSDS2Java 변환기 사용 [8b]지원 값 오브젝트의 생성자가 수정되었습니다.
- 웹 서비스 URL은 다음을 지정하도록 변경되었습니다.

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

해당 WSDL은 다음 위치에서 검색할 수 있습니다.

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

이러한 변경의 결과로, 현재 API를 사용하고 있는 경우 애플리케이션 코드를 수정해야 합니다. 클라이언트 API 또는 WSDL을 사용하는 지에 따라 세부사항은 다음 섹션을 참조하십시오.

- 『클라이언트 API를 사용하여 기존 구현』
- 3 페이지의 『WSDL을 직접 사용하여 기존 구현』

---

## 클라이언트 API를 사용하여 기존 구현

Campaign 웹 애플리케이션과 상호작용하도록 클라이언트 API .jar 파일을 사용하는 경우 다음 섹션에서 반드시 숙지해야 하는 변경사항에 대해 설명합니다.

### 새 클라이언트 API .jar 파일

클라이언트 API 클래스의 내부 구현을 변경하면 Java™ 애플리케이션에서 다음에 위치한 새 .jar 파일을 사용해야 합니다.

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

새 오피어 작성을 표시하는 Java 예제에 대해서는 51 페이지의 『코드 OfferAPI.java』를 참조하십시오. 동일한 예제를 Campaign의 다음 설치 위치에서 찾을 수 있습니다.

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

### 새 종속 .jar 파일

AXIS2 버전 1.4.1로 업그레이드하면, Java 애플리케이션도 AXIS2 1.4.1 배포 .jar 파일을 사용하여 업그레이드해야 합니다. CampaignServicesClient30.jar은 이러한 .jar 파일에 종속됩니다. 모든 종속 .jar 파일은 사용자 애플리케이션의 Java 클래스 경로에 포함되어야 하며 다음 위치에 있는 Campaign.war 파일에서 찾을 수 있습니다.

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

.jar 파일을 Campaign.war에서 추출하여 Java 클래스 경로에 포함시키십시오.

### 클라이언트 API 생성자의 변경사항

클라이언트 API 오브젝트를 생성하는 동안 아래 예제에서 보여지는 대로 웹 서비스 URL 및 예외 서명이 변경됩니다.

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

## 지원 클래스의 매개변수화된 생성자에 대한 변경사항

AXIS2 엔진을 사용하면 생성된 클래스 및 스텝은 더 이상 매개변수화된 생성자를 갖지 않습니다. 대신 이러한 클래스는 멤버에 대해 Setter 및 Getter가 있는 인수 없는 기본 생성자만 갖게 됩니다.

다음 항목은,

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

아래와 같이 변경되었습니다.

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

---

## WSDL을 직접 사용하여 기존 구현

Campaign 웹 서비스의 WSDL은 다른 써드파티 변환기 도구를 사용하여 지원 클래스와 클라이언트측 스텝을 생성하는 데 사용됩니다. Campaign 웹 애플리케이션과 상호 작용하는데 WSDL을 사용하는 경우 다음 섹션에서 반드시 숙지해야 하는 변경 사항에 대해 설명합니다. 여기에서 제공된 예제는 Apache AXIS2 1.4.1에서 WSDL2Java 도구를 사용합니다.

### WSDL 위치 및 서비스 URL

IBM Campaign에 대한 Campaign 웹 서비스는 다음 위치에 배포되었습니다.

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

해당 WSDL은 다음 위치에서 검색할 수 있습니다.

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

### 스텝 및 클래스 생성

Apache AXIS2 1.4.1의 WSDL2Java 도구는 스텝을 생성하는데 사용되며 WSDL에서 Java 클래스를 지원하는데 사용됩니다. 샘플 Ant 작업은 다음과 같습니다.

도구는 유사한 인수 세트를 사용하여 명령창에서 사용할 수 있습니다. 인수 값은 사용자 환경에 맞게 수정할 수 있습니다.

참고: 기본 ADB 바인딩이 다음 WSDL2Java 변환기 예제에 대해 사용됩니다.

```
<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>

```

## 생성된 스텝 사용 및 클래스 지원

스텝은 다음과 같이 사용될 수 있습니다.

```

CampaignServices30ServiceStub serviceStub = new
CampaignServices30ServiceStub(serviceURL);

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

```

오피는 다음과 같이 작성될 수 있습니다.

```

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();
}

```



이 예제에서 createOffer()는 이제 CreateOffer 유형의 하나의 매개변수만 수락합니다.

AXIS2 엔진을 사용하면 생성된 클래스 및 스텝은 더 이상 매개변수화된 생성자를 갖지 않습니다. 자세한 내용은 3 페이지의 『지원 클래스의 매개변수화된 생성자에 대한 변경사항』을 참조하십시오.

---

## 참조

다음 참조는 이 스펙을 준비하는데 사용됩니다.

- "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, June 24, 2003 (<http://www.w3.org/TR/soap/>)
- "JAX-RPC 1.1", Sun Microsystems, October 14, 2003 (<http://java.sun.com/webservices/jaxrpc/index.jsp>)
- Apache 웹 서비스 작업 그룹(<http://ws.apache.org/axis2>)

---

## 요구사항

이 섹션에서는 디자인에서 언급한 요구사항을 요약합니다. CampaignServices API는 다음이 가능해야 합니다.

- 하위 구현 세부사항에서 클라이언트를 분리하고 IBM Campaign 구성요소에 대한 세밀한 작성, 발견, 읽기 및 업데이트 액세스 제공
- 기존 IBM Campaign GUI 기반 사용자와 함께 상주하며 효과 최소화
- 데이터 무결성 보증
- IBM Campaign의 보안 아키텍처 지원
- 보안 인증을 포함하여 산업 표준 SOAP 지원

---

## 디자인 개요

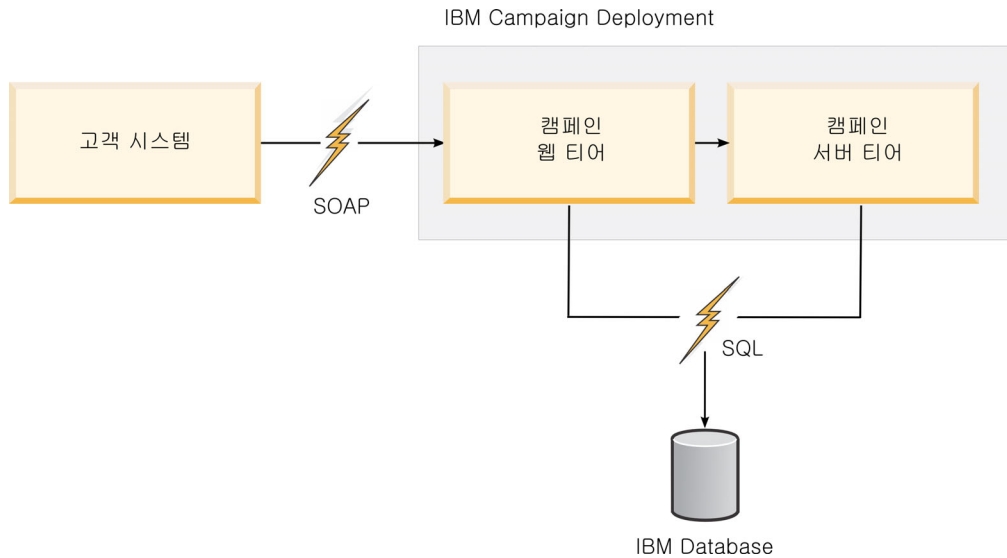
CampaignServices API는 실행 중인 IBM Campaign 애플리케이션 인스턴스의 클라이언트 보기를 제공하는 외관(facade)입니다. IBM Campaign 기능의 서브세트만 노출되지만 Campaign 기능의 핵심 요소를 이끌어내는데 충분합니다. API는 현재의 IBM Campaign 웹 사용자와 다른 API 스투드에서 사용하도록 디자인되었습니다.

일반적으로 API는 캠페인, 오피 및 대상 셀 구성요소에 대한 다음 유형의 작업을 지원 합니다.

- 구성요소 작성
- 구성요소 발견

- 구성요소 삭제
- 구성요소 속성 및 속성 메타데이터 작성, 검사 및 수정
- 플로우차트 실행 결과 페치

다음 다이어그램에서 CampaignServices 3.0의 샘플 배치를 표시합니다.



## 참고

이 섹션에서는 디자인에 대한 특정 포인트에 대해 설명합니다.

### 버전 및 역방향 호환성

일반적으로 CampaignServices API의 이후 버전은 동일한 주 버전 번호를 공유하는 모든 부속 및 유지보수 릴리스가 있는 이전 버전과 호환됩니다. 그러나 IBM 는 비즈니스 또는 기술 사례가 보증하는 경우 주요 릴리스("x.0")에 대한 이전 버전과의 호환성을 파기할 수 있는 권리를 갖습니다.

이 API의 주 버전 번호는 다음 변경사항 중 하나라도 적용될 경우 증가됩니다.

- 데이터 해석이 변경됨
- 비즈니스 로직(즉 서비스 메소드 기능)이 변경됨
- 메소드 매개변수 및/또는 리턴 유형이 변경됨

API의 부 버전 번호는 다음 변경사항 중 하나라도 적용될 경우 증가됩니다.

- 새 메소드가 추가됨
- 새 데이터 유형이 추가되고 사용법이 새 메소드에 제한됨
- 새 유형이 열거 유형에 추가됨
- 인터페이스의 새 버전이 정의됨

특히 IBM 는 적어도 다음 주요 IBM 릴리스가 발표되기 전까지 공개된 WSDL, SOAP 클라이언트를 지원하고 SOAP 오퍼링을 구현하는 데 사용되는 Apache Axis 버전을 지원합니다. 실제로 이 지원은 몇 가지 버전 특정 웹 서비스를 동시에 지원함으로써 가능합니다. IBM 는 이 서비스의 몇 가지 버전을 내부적으로 이미 지원하고 있습니다.

## 사용자 인증

인증은 사용자 ID를 설정합니다.

참고: 이 릴리스의 경우, 사용자 인증은 클라이언트 애플리케이션의 책임입니다.

## 사용자 권한 부여

권한 부여된 사용자에게 권한을 부여하는 것은 구성요소 및 API에 의해 노출된 조작과 관련됩니다.

사용자는 성공적으로 인증될 수 있지만 캠페인 요약 정보 편집과 같은 일부 작업을 수행하는데 권한이 충분하지 않을 수 있습니다. 이런 경우 API 메소드에서 `AuthorizationException`을 예외 처리합니다.

## 로케일

API 요청은 특정 요청을 수행하는데 사용할 로케일을 정의하는 `requestedLocale` 선택적 매개변수에 대해 제공합니다. 정의되지 않은 경우 API는 기본적으로 IBM 사용자 기본 설정 로케일을 사용합니다. 일반 Java best-effort 일치 알고리즘은 메시지 및 다른 로케일로 번역된 텍스트를 요청된 로케일로 리턴하는데 사용됩니다.

이 매개변수는 `java.util.Locale` 클래스 유형입니다.

참고: 캠페인 설명과 같은 일부 사용자 특정 텍스트는 텍스트를 지정한 사용자의 로케일이 사용됩니다. IBM Campaign에서는 그런 데이터의 현지화는 시도하지 않습니다. 정보, 경고 및 오류 메시지만 API에서 현지화합니다.

## State management

CampaignServices API는 상태가 없으며 이는 호출에 대해 API에서 저장한 클라이언트별 정보가 없다는 것을 의미합니다.

특정 API 호출은 Campaign에서 관리하는 하위 구성요소 인스턴스의 상태를 변경할 수 있으며 이러한 상태 변경은 데이터베이스에 대해 지속될 수 있습니다.



---

## 제 2 장 데이터 유형

이 섹션에서는 CampaignServices API에서 사용된 public 데이터 유형을 정의합니다.

---

### WSReference

데이터베이스 ID에 대한 단순 래퍼.

- **componentTypeEnum**: 구성요소 유형(열거형). 다음 중 하나입니다.
  - FOLDER
  - CAMPAIGN
  - FLOWCHART
  - TCS\_CELL
  - OFFER
  - OFFER\_LIST
  - OFFER\_TEMPLATE
- *id*: Long 유형, 참조에 대한 고유한 숫자의 데이터베이스 특정 ID를 정의함.

---

### WSVersion

다음을 포함하여 버전의 다양한 구성요소를 캡처하는 래퍼 유형.

- *major*: 전체 버전 8.1.2.3의 '8'과 같이 주 버전 번호를 정의하는 정수.
- *minor*: 전체 버전 8.1.2.3의 '1'과 같이 부 버전 번호를 정의하는 정수.
- *maintenance*: 전체 버전 8.1.2.3의 '2'와 같이 버전의 유지보수 번호를 정의하는 선택적 정수(적용 가능한 경우). API 버전을 제공하지는 않음.
- *patch*: 전체 버전 8.1.2.3의 '3'과 같이 패치 릴리스 번호를 정의하는 선택적 정수(적용 가능한 경우). API 버전을 제공하지는 않음.

---

### WSServiceInfo

서비스 정보에 대한 단순 래퍼 유형. 다음 필드가 포함됩니다.

- *apiVersion*: *WSVersion* 인스턴스, 서비스에서 지원하는 API의 최신 버전을 정의함. (*apiVersion*은 주 버전 및 부 버전 정보만을 포함합니다.)
- *campaignVersion*: *WSVersion* 인스턴스, 하위 IBM Campaign 인스턴스의 전체 버전을 정의함.
- *name*: "CampaignServices30Service"와 같은 서비스의 내부 이름.

---

## WSAttributeTypeEnum

가능한 모든 속성 유형(열거형)은 다음 중 하나입니다.

- STANDARD: Campaign에서 정의한 표준 또는 기본 속성.
- CUSTOM: 다른 IBM 애플리케이션, 사용자 또는 일부 다른 써드파티에서 정의한 속성.
- INPUT\_PARAMETER: IBM Campaign 플로우차트를 실행하는데 사용된 속성과 같은 입력 매개변수.
- OUTPUT\_PARAMETER: IBM Campaign에서 실행한 플로우차트 결과처럼 완료 되는 속성의 값과 같은 출력 매개변수.

---

## WSAttributeStatusEnum

가능한 모든 속성 상태 코드의 나열. 다음 중 하나입니다.

- ACTIVE: 속성이 활성이며 사용할 수 있음.
- RETIRED: 속성이 서비스에서 제거되었으며 사용할 수 없음.

---

## WSAccessTypeEnum

가능한 모든 속성 값 액세스 유형(열거형)은 다음 중 하나입니다.

- READ\_ONLY: 속성 값을 읽고 표시할 수 있지만 수정할 수 없습니다.
- READ\_WRITE: 속성 값을 읽고 표시하고 수정할 수 있습니다.

속성 액세스는 보안 사용 권한에 부가적입니다. 예를 들어 클라이언트 사용자에게 대한 보안 정책이 특정 속성에 대한 읽기 액세스를 거부하면 속성 액세스는 보안 설정을 재 정의할 수 없습니다. 실제로 API는 클라이언트에 속성을 리턴하지 않습니다.

---

## WSSelectTypeEnum

특정 속성 값에 대해 가능한 모든 선택 유형을 정의합니다. 다음 중 하나입니다.

- NONE: 선택사항 없음(*hasOptions*는 false입니다).
- SINGLE\_SELECT: 가능한 옵션 목록에서 한 번에 하나의 속성 옵션만 선택할 수 있음(*hasOptions* 속성인 경우만 유효함).
- MULTIPLE\_SELECT: SINGLE\_SELECT와 유사함. 한 번에 하나 이상의 옵션을 선택할 수 있음.

---

## WSRunStatusEnum

가능한 모든 플로우차트, 브랜치 또는 셀 실행 상태에 대한 유형(열거형)은 다음 중 하나입니다.

- NOT\_STARTED: 실행이 스케줄링되어 있지만 아직 시작하지 않음.
- RUNNING: 실행 중.
- CANCELLED: Campaign 사용자 또는 이 API를 통해 실행이 취소됨.
- SUCCEEDED: 실행이 성공적으로 완료됨.
- FAILED: 실행에 실패함. 오류 세부사항은 따로 보고됩니다. (16 페이지의 『WSRunResults』를 참조하십시오.)

---

## WSRunTypeEnum

가능한 모든 실행 유형(열거형). 다음 중 하나입니다.

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

---

## WSAttribute

속성은 API를 통해 액세스 가능한 구성요소 인스턴스로 임시 데이터를 연결하기 위한 단순하고 확장 가능한 메커니즘을 제공합니다. 이러한 데이터에는 캠페인 이름 같은 표준 데이터, 성별 같은 플로우차트 실행 입력 매개변수, 다른 IBM 애플리케이션 또는 IBM 고객이 지정한 임시 사용자 정의 데이터가 포함됩니다.

**참고:** 이 API에서 속성은 Campaign 사용자 정의 속성뿐만 아니라 대부분의 구성요소 데이터를 모델링하는 데 사용됩니다.

일반적으로 구성요소에는 연결되어 있는 많은 속성이 있으며 이러한 속성은 *AttributeMap*이라는 특수한 유형의 맵으로 CampaignServices API에서 제공됩니다. 속성 데이터는 API 전체에 걸쳐 엄격한 유형의 구체적 클래스로 표현됩니다. 예를 들어 *WSDecimalAttribute*는 10진(백 정밀도 숫자) 데이터를 포함하는 속성입니다.

각 속성에는 다음이 포함됩니다.

- **이름:** 속성의 고유 이름. 이 이름은 속성이 존재하는 구성요소 인스턴스 내에서 속성 및 해당 메타데이터를 액세스하기 위한 키로 제공됩니다. 이름의 형식은 정의되지 않지만 몇몇 경우에 서비스, 클라이언트 또는 IBM Campaign 사용자가 지정합니다.

일반적으로 이 이름은 Campaign 또는 클라이언트 사용자에게 표시되는 표시 이름이 아닙니다. 이름은 *uacDescription*과 같이 API에서 표준화될 수 있으며, 플로우 차트를 공개할 때 IBM Campaign에서 지정할 수 있거나 사용자 정의 속성을 정의할 때 IBM 애플리케이션 또는 고객이 지정할 수 있습니다. 모든 경우에서 이름은 고유성이 보장됩니다.

- **메타데이터:** (선택사항) 값 데이터 유형, 표시 이름, 설명, 프롬프트, 기본값, 선택 유형, 길이(텍스트), 정밀도(10진), 옵션(단일 또는 복수 선택의 경우) 등과 같은 속성의 데이터에 관한 정보. 13 페이지의 『WSAttributeMetadata』를 참조하십시오.
- **값:** 0 또는 더 엄격한 유형의 값 오브젝트의 배열. 값 필드는 구체적 속성 클래스에서 제공되며 각 값의 유형은 동일하고 속성의 메타데이터 필드에 있는 유형 정의에 부합되어야 합니다. 모든 속성이 여러 값을 지원하는 것은 아닙니다.

다음 구체적 속성 유형이 지원됩니다.

- **WSBooleanAttribute:** 값이 부울(*true* 또는 *false*)인 속성.
- **WSIntegerAttribute:** 정수 값(*java.lang.Long*).
- **WSDecimalAttribute:** 배 정밀도 10진 값(*java.lang.Double*).
- **WSCurrencyAttribute:** 미국 달러의 경우 "*USD*"와 같이 통화 값의 선택적 ISO 4217 통화 코드를 포함하는 복합 통화 값 그리고 *Double*로 캡처된 통화 값. 통화 코드가 제공되지 않는 경우 IBM Campaign에서 사용되는 기본값이 사용됩니다.

국가 목록, 통화 기호 및 코드는 <http://www.xe.com/symbols.php> 사이트를 참조하십시오. 통화 값에 사용되는 로케일은 사용자가 선호하는 로케일과 다를 수 있습니다.

- **WSCalendarAttribute:** 값이 일부 시간대 및 로케일의 날짜 또는 시간인 속성.
- **WSTextAttribute:** 유니코드 텍스트의 문자열(널 또는 공백 가능).

**참고:** 일반적으로 가능한 속성의 목록은 각각의 구성요소 유형에 대해 다르지만 목록은 겹쳐질 수 있습니다.



---

## WSAttributeMetadata

WSAttributeMetadata는 값 데이터 유형, 현지화된 텍스트(표시 이름, 설명, 프롭र्ट 등), 해당 기본값, 허용 가능한 값 범위, 선택 유형, 옵션(단일 또는 복수 선택의 경우) 등과 같은 특정 유형 속성의 데이터에 관한 정보를 정의합니다. 속성과 마찬가지로 속성 메타데이터는 엄격한 유형입니다. 예를 들어 `WSDecimalAttribute myNumber`는 바인드된 `WSDecimalAttributeMetadata`가 있어야 하며 속성 값, 메타데이터 기본값 및 가능한 옵션 값을 포함하여 모든 값은 `Double` 유형입니다.

설명, 레이블 및 기타 속성 메타데이터 텍스트는 일반적으로 현지화되지만 사용자 정의 텍스트는 사용자가 입력한 언어로만 사용할 수 있습니다. 각 API 호출에는 특정 사용자가 현지화된 메시지를 표시하려는 로케일을 정의하기 위해 클라이언트 코드가 사용할 수 있는 요청된 로케일이 포함되어 있습니다. 일반적인 Java 로케일 폴백 정책은 이러한 요청을 실행하는 데 사용됩니다.

WSAttributeMetadata에는 다음 필드가 포함됩니다.

- **이름:** 속성의 이름(표준 또는 사용자 정의). 이 메타데이터에 바인드하는 속성에서 사용되는 이름이기도 합니다. 표준 속성은 시스템에서 정의되며 예약된 네임스페이스의 표준 이름을 가지고 있습니다(즉 "uac" 접두부 사용). 사용자 정의 이름은 다른 이름 지정 규칙을 사용할 수 있습니다.

**참고:** 속성 이름은 고유해야 하고 현지화되지 않아야 하며 길이 제한(문자 콘텐츠 및 데이터베이스에 따라 결정)이 있습니다. 이름은 대소문자를 구분하지 않으며 유니코드 문자, 숫자 또는 밑줄 문자(\_)의 조합으로 구성될 수 있지만 숫자로 시작할 수는 없습니다.

- **설명:** 속성의 선택적 설명. 도구 팁 또는 기타 사용자 인터페이스 프리젠테이션에 적합합니다.
- **연산 술어:** 속성을 설명하는 다양한 연산 술어.
  - *isRequired:* 속성이 필수인 경우 참.
  - *isInternal:* 속성이 시스템에서 정의되고 내부 사용 전용(사용자에게 표시되지 않음)인 경우 참.
  - *isGenerated:* 대상 셀 코드와 같은 속성의 값이 구성요소가 작성될 때 IBM Campaign에서 자동으로 생성되는 경우 참. 일반적으로 *accessTypeEnum*은 생성된 값에 대해 `READ_ONLY`가 됩니다.
  - *hasOptions:* 속성에 옵션이 있는 경우 참. 이 메타데이터에 대해 옵션이 정의되고 *selectTypeEnum*이 `SINGLE_SELECT` 또는 `MULTIPLE_SELECT`임을 의미합니다.
- **typeEnum:** `STANDARD` 또는 `CUSTOM`과 같은 속성의 유형을 정의하는 *WSAttributeTypeEnum*.
- **statusEnum:** `ACTIVE`와 같은 속성의 상태를 정의하는 *WSAttributeStatusEnum*.

- *accessTypeEnum*: READ\_ONLY와 같은 속성 값에 대한 액세스 유형을 정의하는 *WSAccessTypeEnum*.
- *selectTypeEnum*: SINGLE과 같은 속성에 사용되는 선택사항의 유형을 정의하는 *WSAccessTypeEnum*. 옵션이 제공되지 않는 경우 또는 캠페인 및 셀 속성에 대해 NONE이어야 합니다.
- *componentTypeEnum*: CAMPAIGN, FOLDER 등과 같은 API에서 제공하는 모든 사용 가능한 Campaign 구성요소의 *WSComponentTypeEnum*.
- *defaultValue*(플로우차트 전용): 속성에 대한 선택적 유형 기본값. 이 값은 구체적 속성 메타데이터 클래스에서 제공됩니다. 예를 들어 *WSTextAttributeMetadata*의 기본값은 String 유형입니다. 자세한 내용은 속성 값의 설명 부분을 참조하십시오. 플로우차트가 아닌 구성요소에 대해서는 기본값이 정의되지 않습니다.
- *options*: 이 속성에 대한 옵션의 선택적 목록. 함께 사용하는 경우 속성의 옵션은 해당 속성에 대해 허용 가능한 값의 정확한 세트를 정의합니다. 각 옵션은 엄격한 유형입니다. 예를 들어 *WSTextAttributeMetadata*는 바인드된 *WSTextAttributeOption*만 가질 수 있습니다.

**참고:** 옵션에서는 텍스트 속성만 지원되는 제한사항이 있습니다.

각 옵션은 다음을 정의합니다.

- *prompt*: 성별 속성 옵션으로 사용되는 "Male"과 같은 풀다운 메뉴에 적합한 옵션의 *prompt*. 메타데이터 *prompt*와는 다르게 옵션 표시 이름은 일반적으로 마침표를 포함하지 않습니다.
- *description*: "A person of the male persuasion"과 같은 옵션의 현지화된 설명. 도구 팁 텍스트에 적합합니다.
- *isDefault*: 이 특정 옵션이 기본값인 경우 참. MULTIPLE\_SELECT 선택 유형에서는 둘 이상의 옵션이 기본값으로 표시될 수 있습니다.
- *value*: 입력된 옵션 값. 속성 메타데이터 *defaultValue*와 마찬가지로 이 값은 구체적 옵션 서브클래스에서 제공됩니다. 예를 들어 *WSDecimalAttributeOption* 값은 Decimal 유형입니다. 자세한 내용은 속성 값의 설명 부분을 참조하십시오. 위에 있는 성별 예제에서 값은 문자열 "m"(*WSTextAttributeOption*) 또는 숫자 코드 123(*WSDecimalAttributeOption*)으로 선언될 수 있습니다.

---

## WSCampaignInfo

캠페인 속성 데이터에 대한 단순 래퍼.

다음 필드가 포함됩니다.

- *reference*: 캠페인 참조.
- *name*: 캠페인 이름(*uacName*); 고유하지 않을 수 있음.

- *description*: 선택적 캠페인 설명(*uacDescription*).
- *campaignCode*: 고유한 캠페인 코드(*uacCampaignCode*); 클라이언트 또는 Campaign에서 지정함.

---

## WSComponentOrFolderInfo

표시 이름, 해당 참조와 같은 폴더 속성 데이터 또는 줄 바꿈된 캠페인의 조합을 포함합니다.

다음 필드가 포함됩니다.

- *reference*: 구성요소 또는 폴더의 참조.
- *name*: 구성요소 또는 폴더 이름(*uacName*); 고유하지 않을 수 있음.
- *description*: 선택적 구성요소 또는 폴더 설명(*uacDescription*).
- *componentCode*: 구성요소에 대한 고유한 코드 또는 폴더인 경우는 널(null).

---

## WSTargetCellInfo

대상 셀 행 속성 데이터에 대한 단순 래퍼.

다음 필드가 포함됩니다.

- *reference*: 셀 참조.
- *name*: 셀 이름(*uacName*); 고유하지 않을 수 있음.
- *description*: 선택적 셀 설명(*uacDescription*).
- *cellCode*: 셀 코드(*uacCellCode*); 클라이언트 또는 캠페인에서 지정함. 셀 코드는 IBM Campaign의 *DuplicateCellCodesAllowed* 구성 매개변수를 false로 설정하여 고유하도록 강제 실행될 수 있습니다.
- *flowchartName*: 셀이 바인드될 플로우차트의 선택적 이름.

---

## WSMetricsInfo

컨택 번호를 포함하여 캠페인 분석 데이터에 대한 단순 래퍼 유형. 다음 필드가 포함됩니다.

- *totalContacts*: long 유형, 컨택 총계를 제공함.
- *responses*: *WSMetricsResponse* 인스턴스의 형식화된 목록, 각 인스턴스에서 하나의 응답에 대한 컨택 정보를 정의함.
  - *typeCode*: 전화 호출 컨택에 대해 *PHC*와 같은 응답 유형 코드를 정의하는 문자열.
  - *count*: long 유형, 이 컨택이 발생한 횟수를 제공함.

---

## WSRunResults

실행 상태, 플로우차트 실행 시작 및 종료 날짜/시간 및 계수를 포함하여 플로우차트, 프로세스 상자 또는 셀 실행, 여전히 실행 중일 수 있는, 결과에 대한 랩퍼 유형.

다음 필드가 포함됩니다.

- *sourceReference*: 실행 결과 소스의 선택적 참조. 실행 결과가 페치되는 컨텍스트에 따라 플로우차트, 플로우차트 프로세스 상자 또는 대상 셀을 참조할 수 있습니다. 이런 경우 남은 실행 결과 데이터는 이 소스를 참조합니다.
- *flowchartName*: 실행되는 플로우차트의 이름.
- *flowchartId*: 플로우차트에 대한 데이터베이스 ID.
- *runId*: 실행 데이터베이스 ID.
- *typeEnum*: PRODUCTION\_PROCESS와 같은 실행에서 생성하는 결과 종류를 정의하는 유형(열거형). (*WSRunTypeEnum* 참조)
- *statusEnum*: RUNNING과 같은 실행 상태를 정의하는 유형(열거형). (*WSRunStatusEnum* 참조)
- *statusCode*: 선택적 정수 상태 코드.
- *statusMessage*: 선택적 상태 메시지.
- *startDate*: 실행이 시작된 선택적 일정 날짜 시간. 실행이 시작되지 않은 경우 값은 널(null)이 됩니다.
- *endDate*: *startDate*와 유사하지만 실행이 종료된 날짜 시간(성공 또는 실패). 실행이 시작되었지만 아직 종료되지 않은 경우 값은 널(null)이 됩니다.
- *count*: 실행에서 선택한 컨텍의 선택적 총 개수. 실행이 완료되지 않은 경우 0 또는 널(null)일 수 있습니다.

---

## WSOfferInfo

오퍼 또는 오퍼 목록 속성 데이터에 대한 단순 랩퍼.

다음 필드가 포함됩니다.

- *reference*: 오퍼 또는 오퍼 목록의 참조.
- *name*: 오퍼 또는 오퍼 목록 이름(*uacName*); 고유하지 않을 수 있음.
- *description*: 선택적 설명(*uacDescription*).
- *offerCode*: 오퍼 또는 오퍼 목록이 널(null)인 경우의 오퍼 코드(*uacOfferCode*). (고유하지 않을 수 있음)

---

## WSOfferCodeOrName

오퍼 코드 또는 오퍼 목록 이름 데이터에 대한 단순 래퍼 유형.

다음 필드가 포함됩니다.

- *isCode*: *codeOrName* 필드가 가정된 오퍼 코드(true) 또는 오퍼 목록의 이름(false) 인지 표시하는 부울.
- *codeOrName*: 오퍼 또는 오퍼 목록의 이름인 경우라면 고유 오퍼 코드(*uacOfferCode*).

---

## WSOfferValidationInfo

오퍼 유효성 검증 정보에 대한 단순 래퍼 유형.

다음 필드가 포함됩니다.

- *errorCode*: 널(null)이 아닌 경우 영숫자 유효성 검증 오류 코드를 정의합니다. 오류 코드에 대해서는 *IStandardDefinitions* 클래스를 참조하십시오.
- *errorMessage*: 선택적으로 현지화된 메시지에서 오류를 설명합니다(발생한 경우).
- *codeOrName*: 유효성이 검증된 오퍼 코드 또는 오퍼 목록 이름.
- *reference*: 오퍼 또는 오퍼 목록 참조(유효한 경우).

---

## WSOfferTemplateInfo

오퍼 템플릿 데이터에 대한 단순 래퍼 유형.

다음 필드가 포함됩니다.

- *reference*: 오퍼 템플릿의 참조.
- *name*: 오퍼 템플릿 이름, 고유함.
- *description*: 선택적 설명(*uacDescription*).
- *offerTemplateID*: 고유 오퍼 템플릿 데이터베이스 ID.

---

## WSBulkOfferInfo

별크에서 오퍼를 작성하는데 사용됩니다.

다음 필드가 포함됩니다.

- *offerName*: 작성 중인 오퍼의 이름.
- *attributes*: 오퍼 속성을 표시하는 *WSAttribute* 유형의 배열.

---

## WSOfferInfoStatus

*bulkCreateOffers()* API 메소드에 대한 리턴 유형으로 벌크 오퍼 작성의 상태를 나타냅니다.

다음 필드가 포함됩니다.

- *name*: 오퍼의 이름.
- *code*: 오퍼 코드. 오퍼 작성에 실패하면 널(null)이 됩니다.
- *description*: 오퍼 설명.
- *reference*: 작성된 오퍼의 WReference. 오퍼 작성에 실패하면 널(null)이 됩니다.
- *status*: WSRequestStatus의 인스턴스로 오퍼 작성의 상태를 나타냅니다.

---

## 제 3 장 공통 예외

이 섹션에서는 CampaignServices API에서 예외 처리하는 공통 예외에 대해 설명합니다.

모든 예외는 IBM Campaign에 사용 가능한 경우 요청된 로케일로 메시지가 번역됩니다. 일반적인 Java 로케일 폴백 정책을 적용합니다.

---

### RemoteException

참고: 이 항목은 SOAP 인터페이스에만 적용됩니다.

API에 대한 모든 SOAP 호출은 SOAP 엔벨로프 처리 계층(Axis)에 문제점이 있거나 웹 서비스 WSDL에서 정의한 제한조건이 일부 이유로 위반된 경우와 같은, 시스템 레벨 오류가 발생하면 *RemoteException* 예외를 처리합니다.

*DataException*과 같은 보통의 선택하거나 선택하지 않은 API 예외는 *RemoteException* 오류가 아닌 오류 상태로 리턴됩니다.

세부사항에 대해서는 SOAP 인터페이스 섹션을 참조하십시오.

---

### AuthenticationException

사용자는 지정된 Campaign 파티션에 대해 인증을 받을 수 없습니다. Marketing Platform에서 설정된 사용자 역할을 확인하십시오.

---

### AuthorizationException

사용자는 요청된 조작을 수행할 권한이 부여되지 않았습니다. 이 예외는 API 메소드로 처리되므로 선언되지 않습니다(선택되지 않음). IBM Marketing Platform의 사용자 역할에 지정된 권한을 확인하십시오.

---

### DataException

치명적 예외가 Campaign의 하위 데이터베이스 계층에서 발생했습니다(선택되지 않음).

세부사항에 대해서는 Campaign 플로우차트 및 리스너 로그를 확인하십시오.

---

## LockException

다른 사용자가 편집하고 있는 동안 클라이언트에서 플로우차트와 같은 구성요소를 업데이트하려고 시도하면 임시 예외 처리됩니다. 일반적으로 이 예외는 어느 정도 대기한 후에 조작을 다시 시도하면 복구될 수 있습니다. 재시도 로직은 클라이언트의 책임입니다.

---

## InvalidComponentException

유효하지 않거나 알 수 없는 구성요소를 참조하려고 시도했습니다(캠페인, 플로우차트, 대상 셀 등). 예외의 `getComponentReference()` 메소드에서 올바르지 않은 구성요소의 참조를 리턴합니다.

---

## InvalidAttributeException

클라이언트가 유효하지 않은 데이터 유형을 사용하거나 허용되지 않는 값의 배열을 사용하는 경우와 같이, 유효하지 않은 속성을 제공하거나 참조한 경우 예외가 처리됩니다. 예외의 `getAttributeName()` 메소드에서 문제점이 있는 속성의 이름을 리턴합니다. `getAttributeValue()`에서 값을 리턴하고 `getComponentReference()`에서 구성요소(또는 벌크 색인)를 식별합니다.

---

## AttributeNotFoundException

클라이언트에서 알 수 없는 속성(캠페인, 플로우차트, 대상 셀 등)을 참조하려고 시도할 때마다 예외 처리됩니다. 예외의 `getAttributeName()` 메소드에서 일치하지 않은 속성의 이름을 리턴합니다. `getComponentReference()`는 구성요소(또는 벌크 색인)를 식별합니다.

---

## AttributeExistsException

클라이언트에서 구성요소에 대해 복제된 속성 메타데이터를 정의하려고 하면 예외 처리가 됩니다. 예외의 `getAttributeName()` 메소드에서 복제된 속성의 이름을 리턴합니다. `getComponentReference()`는 구성요소(또는 벌크 색인)를 식별합니다.

---

## CompositeException

*CompositeException*은 일부 API에서 여러 오류를 호출자에게 다시 보고하는데 사용됩니다. 일반적으로 바인드된 하나 이상의 원인이 있습니다. 모든 원인은 발생한 순서대로 목록으로 캡처됩니다. 예외의 `getCauseList()` 메소드는 이 목록을 리턴하며 목록은 각 오류의 세부사항을 가져오기 위해 이후에 검사하는데 사용될 수 있습니다.



참고: 일반적으로 API는 성공정으로 완료하거나 *CompositeException* 예외를 처리하기 전에 해당 작업을 롤백합니다. 예를 들어 34 페이지의 『대상 셀 메소드』에서 설명된 벌크 대상 셀 스프레드시트 API를 참조하십시오.



---

## 제 4 장 CampaignServices API 메소드

이 절에서는 CampaignServices 3.0 API에서 내보낸 프린시플 메소드를 정의합니다.

---

### 서비스 메소드

API는 서비스 자체에 대한 정보 식별을 결정하는 방법을 제공합니다.

#### getServiceInfo

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

지원하는 최신 API 버전, 하위 IBM Campaign 인스턴스의 전체 버전 등의 서비스에 대한 정보를 리턴합니다.

참고: 이 호출에는 클라이언트 정보는 필요하지 않으며 보안 사용 권한이 적용되지 않습니다.

#### 매개변수

없음

#### 리턴

*WSServiceInfo* 인스턴스로 리턴합니다.

#### 오류

없음

---

### 속성

대부분의 구성요소 인스턴스 데이터는 *attributes* 또는 *metadata* 속성과 같은 API에 의해 노출됩니다. 일부 경우 속성 메타데이터 정의가 IBM Campaign에 대해 전체적으로 적용되지만(캠페인 사용자 정의 속성과 같은), 나머지 다른 경우는 특정 구성요소에 제한됩니다(플로우차트 사용자 변수와 같은).

따로 표시되지 않는 한, 클라이언트가 실행할 수 있는 충분한 보안 사용 권한을 갖고 있다면 모든 속성을 읽을 수 있습니다.

참고: 활성 상태이며 클라이언트가 액세스할 수 있는 구성요소만 API에 의해 노출됩니다. Public 지원은 오픈 속성, 오픈 템플릿 및 메타데이터로 제한됩니다.

## getAttributesByName

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

지정된 구성요소 인스턴스(비어 있을 수 있음)와 연관된 이름 지정된 속성을 페치합니다.

### 매개변수

*userCredential*: 클라이언트 사용자 신임 정보.

*requestedLocale*: 이 요청에 사용할 선택적 로케일. 제공되지 않으면 IBM 사용자 로케일 환경 설정이 사용됩니다. 필요한 경우 일반 로케일 기본값 지정 알고리즘이 적용됩니다.

*partitionName*: 사용할 캠페인 파티션의 선택적 이름. 정의되지 않은 경우 기본 파티션이 사용됩니다.

*reference*: 원하는 속성을 포함하는 구성요소 인스턴스의 *Reference*. 참조가 유효하지 않거나 구성요소가 존재하지 않는 경우 *InvalidComponentException* 예외가 처리됩니다.

*names*: 페치할 속성 이름의 배열(표시 이름이 아님). 제공되지 않은 경우 모든 속성이 리턴됩니다. 이름 지정된 속성 중 하나가 존재하지 않는 경우 *AttributeNotFoundException* 예외를 처리합니다.

### 리턴

0개 이상의 속성의 형식화된 맵. 속성 이름은 맵 항목 키이며 속성 인스턴스는 항목 값입니다.

### 오류

*InvalidComponentException*, *AttributeNotFoundException*

*AuthorizationException*, *DataException*

참고: 이러한 모든 예외는 *CampaignServicesException* 내부에서 줄 바꿈됩니다.

## updateAttributes

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

구성요소 인스턴스의 하나 이상의 속성을 제공된 속성 값으로 업데이트합니다.

## 업데이트 로직

업데이트 로직은 다음과 같습니다.

제공된 배열에 포함된 각 속성의 경우:

1. 속성 이름이 기존 속성과 일치하는 경우 제공된 값 필드로 기존 값 필드 겹쳐쓰기를 시도하십시오.
2. 속성이 아직 없는 경우 *allowCreate*이 true이고 해당 메타데이터를 알 수 있으면 속성을 작성합니다. 이것은 인스턴스 속성뿐만 아니라 글로벌 속성 메타데이터에도 적용됩니다(플로우차트 제외).
3. 값 유형 또는 속성 메타데이터 정의의 일부 다른 요소가 부합되지 않는 경우 또는 하나 이상의 제공된 값이 유효하지 않거나 범위를 벗어나는 등의 경우 *InvalidAttributeException* 예외를 처리합니다.
4. 그렇지 않고 이름 붙여진 속성이 없는 경우 *AttributeNotFoundException* 예외가 처리됩니다.

**참고:** 예외가 발생하는 경우 아무런 업데이트도 커밋되지 않습니다.

이 특정 메소드는 새 사용자 정의 속성 정의를 지원하지 않습니다.

*createAttributeMetadata()* 메소드를 사용하십시오.

모든 경우에 있어서 속성 업데이트 작업은 일반 보안 제한조건 및 유효성 검증의 적용을 받습니다. 어떤 속성이 특정 구성요소 인스턴스에 필요한지, 올바른 유형은 무엇인지 등을 결정하는 것은 클라이언트의 책임입니다.

## 매개변수

*userCredential*: 클라이언트 사용자 신임 정보.

*requestedLocale*: 이 요청에 사용할 선택적 로케일.

*partitionName*: 사용할 캠페인 파티션의 선택적 이름.

*reference*: 업데이트할 속성을 포함하는 구성요소 인스턴스에 대한 참조.

*allowCreate*: 구성요소에 대한 새 속성이 아직 존재하지 않는 경우 작성해야 하는지 여부를 표시합니다. (『업데이트 로직』를 참조하십시오.)

*attributes*: 업데이트될 속성의 배열; 속성 이름은 업데이트할 속성을 찾는데 사용되며 새 값은 기존 속성의 값을 적합한 유형 또는 배열의 단순 오브젝트로 업데이트하는데 사용됩니다(적용 가능한 경우). (19 페이지의 제 3 장 『공통 예외』를 참조하십시오.)

## 리턴

없음

## 오류

InvalidComponentException, AttributeNotFoundException, InvalidAttributeException

AuthorizationException, DataException

## getAttributeMetadataByName

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

특정 구성요소, 템플릿에 바인드되거나 또는 글로벌 정의된, 이름 지정된 속성 메타데이터 정의를 폐치합니다.

## 매개변수

*userCredential*: 클라이언트 사용자 신임 정보.

*requestedLocale*: 이 요청에 사용할 선택적 로케일.

*partitionName*: 사용할 캠페인 파티션의 선택적 이름.

*reference*: 원하는 속성 메타데이터를 포함하는 구성요소 또는 템플릿에 대한 선택적 참조. ComponentTypeEnum만 제공되면 폐치는 구성요소의 유형에 제한됩니다. 만약 참조가 전혀 제공되지 않는 경우에는 모든 구성요소 유형에 대해 폐치에서 모든 글로벌 메타데이터 정의를 리턴합니다. 제공된 참조가 유효하지 않으면 *InvalidComponentException* 예외를 처리합니다.

*names*: 폐치할 속성 메타데이터 이름의 선택적 배열. 제공되지 않으면 구성요소에 대한 모든 메타데이터가 또는 참조가 없는 경우, 전체적으로 정의된 메타데이터가 리턴됩니다. 하나 이상의 지정된 속성 메타데이터 정의가 없는 경우 *AttributeNotFoundException* 예외를 처리합니다.

## 리턴

0개 이상의 속성 메타데이터 정의의 형식화된 맵. 속성 이름은 맵 항목 키이며 속성 메타데이터는 항목 값입니다.

## 오류

InvalidComponentException, AttributeNotFoundException

AuthorizationException, DataException

## createAttributeMetadata

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

하나 이상의 새 속성 메타데이터 정의를 작성하고 특정 구성요소 또는 템플릿에 선택적으로 바인드하십시오.

### 매개변수

*userCredential*: 클라이언트 사용자 신임 정보.

*requestedLocale*: 이 요청에 사용할 선택적 로케일.

*partitionName*: 사용할 캠페인 파티션의 선택적 이름.

*reference*: 메타데이터를 바인드해야 하는 구성요소 또는 템플릿에 대한 선택적 참조. 제공되지 않는 경우 작성된 메타데이터 정의가 전체 적용됩니다. 참조가 제공되었지만 유효하지 않다면 *InvalidComponentException* 예외를 처리합니다.

*attributeMetadata*: 바인드할 속성 메타데이터 정의 배열. 하나 이상의 지정된 메타데이터가 이미 구성요소에 바인드된 경우 즉, 이름이 고유하지 않은 경우 *AttributeExistsException* 예외를 처리합니다. 하나 이상의 지정된 메타데이터에 문제점이 있거나 즉 내부적으로 일치하지 않은 경우 *InvalidAttributeException* 예외를 처리합니다.

### 리턴

없음

### 오류

*InvalidComponentException*, *AttributeExistsException*, *InvalidAttributeException*

*AuthorizationException*, *DataException*

## updateAttributeMetadata

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

지정된 구성요소 또는 템플릿에 대한 하나 이상의 속성 메타데이터 정의를 업데이트합니다. 선택적으로 새 메타데이터 정의를 작성합니다(필요한 경우).

## 업데이트 로직

업데이트 로직은 다음과 같습니다.

제공된 배열에 포함된 각 속성 메타데이터 정의의 경우:

1. 속성 이름이 구성요소에 바인드된 기존 메타데이터와 일치하지 않는 경우 *allowCreate* 매개변수 값에 따라 다음을 수행하십시오.
  - a. *True*: 새 메타데이터 정의를 작성합니다. 기능적으로 *createAttributeMetadata()* 요청을 사용하는 것과 동일합니다.
  - b. *False*: *AttributeNotFoundException* 예외 처리를 합니다.
2. 속성 메타데이터 데이터 유형이 다른 경우 *InvalidAttributeException* 예외 처리를 하십시오.
3. 제공된 메타데이터의 필드 값으로 기존 속성 메타데이터 정의 겹쳐쓰기를 시도하고 이외의 경우 *InvalidAttributeException* 예외 처리를 하십시오. 다음 업데이트만 지원됩니다(이외의 경우 *InvalidAttributeException* 예외 처리).
  - a. *name*: 변경이 허용되지 않습니다(*name*은 키입니다!).
  - b. *displayName*: 새 값을 수락합니다.
  - c. *description*: 새 값을 수락합니다.
  - d. *isRequired*: *true*에서 *false*로의 변경만 허용됩니다.
  - e. *isInternal*: 새 값을 수락합니다.
  - f. *isGenerated*: 변경이 허용되지 않습니다.
  - g. *attributeTypeEnum*: 변경이 허용되지 않습니다.
  - h. *accessTypeEnum*: 새 값을 수락합니다.
  - i. *selectTypeEnum*: 옵션이 제공되는 경우 다음 전이를 수락합니다.
    - 1) NONE에서 SINGLE\_SELECT 또는 MULTIPLE\_SELECT로
    - 2) SINGLE\_SELECT에서 MULTIPLE\_SELECT로
  - j. *options*: 옵션은 추가할 수 있지만 삭제할 수는 없습니다. 다음 옵션 변경만 지원됩니다(값 일치에 따라).
    - 1) *displayName*: 새 값을 수락합니다(과급 효과 없음).
    - 2) *description*: 새 값을 수락합니다(과급 효과 없음).
    - 3) *isDefault*: 새 값을 수락하지만 *SelectTypeEnum*과 일치해야 합니다.
    - 4) *value*: 변경이 허용되지 않습니다(*value*는 키입니다!).
  - k. *defaultValue*(플로우차트 전용): 새 기본값을 수락합니다.
  - l. *maxLength*(텍스트 전용): 더 큰 경우 새 길이 값이 허용됩니다.
4. 속성 메타데이터 정의가 내부적으로 지속적이지 않은 경우 *InvalidAttributeException* 예외 처리를 하십시오.



5. 필요한 경우 업데이트된 속성 메타데이터를 참조하는 모든 구성요소 인스턴스를 찾아서 적절하게 업데이트하십시오.

**참고:** 예외가 발생하는 경우 아무런 업데이트도 커밋되지 않습니다.

모든 경우에 있어서 속성 업데이트 작업은 일반 보안 제한조건 및 유효성 검증의 적용을 받습니다.

`createAttributeMetadata()`, `deleteAttributeMetadata()`를 참조하십시오.

## 매개변수

*userCredential*: 클라이언트 사용자 신임 정보.

*requestedLocale*: 이 요청에 사용할 선택적 로케일.

*partitionName*: 사용할 캠페인 파티션의 선택적 이름.

*reference*: 원하는 속성을 포함하는 구성요소 인스턴스의 선택적 참조. 제공되지 않으면 업데이트는 글로벌 메타데이터 정의로 제한됩니다. 제공된 참조가 유효하지 않으면 *InvalidComponentException* 예외를 처리합니다.

*allowCreate*: true인 경우 현재 존재하지 않는 메타데이터 정의가 작성됩니다(기능적으로 동일한 `createAttributeMetadata()` 메소드를 사용하여).

*attributeMetadata*: 업데이트할 속성 메타데이터 정의 배열(*allowCreate* 플래그가 true인 경우 추가됨). 속성 이름은 업데이트할 메타데이터 정의를 찾는데 사용되며 나머지 데이터는 기존 정의를 업데이트하는데 사용됩니다. (28 페이지의 『업데이트 로직』을 참조하십시오.)

## 리턴

없음

## 오류

*InvalidComponentException*, *InvalidAttributeException*

*AuthorizationException*, *DataException*

## deleteAttributeMetadata

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

지정된 구성요소, 템플릿(사용자 정의 속성 메타데이터 전용) 또는 글로벌 속성 메타데이터 정의에서 하나 이상의 이름 지정된 속성 메타데이터 정의를 삭제합니다.

이 작업의 일부로, 삭제된 메타데이터를 참조하고 적절하게 업데이트하는 모든 구성요소를 메소드에서 찾습니다.

**참고:** 그러나 예외 이벤트에서 삭제된 정의가 없는 경우도 커밋합니다.

## 매개변수

*userCredential*: 클라이언트 사용자 신임 정보.

*requestedLocale*: 이 요청에 사용할 선택적 로케일.

*partitionName*: 사용할 캠페인 파티션의 선택적 이름.

*reference*: 삭제될 속성을 포함하는 구성요소 또는 템플릿의 선택적 참조. 제공되지 않으면 삭제는 글로벌 메타데이터 정의로 제한됩니다. 제공된 참조가 유효하지 않으면 *InvalidComponentException* 예외를 처리합니다.

**참고:** 속성 메타데이터 이름의 선택적 배열이 제공되지 않으면 이 메소드는 구성요소와 연관된 모든 사용자 정의 속성 메타데이터를, 또는 참조가 제공되지 않는 경우에는 모든 글로벌 정의를 삭제하려고 시도합니다.

*names*: 삭제할 속성 메타데이터 이름의 선택적 배열. 하나 이상의 이름 지정된 속성 메타데이터가 없는 경우 *AttributeNotFoundException* 예외를 처리합니다. 속성을 제거할 수 없으면 *InvalidAttributeException* 예외를 처리합니다.

## 리턴

없음

## 오류

*InvalidComponentException*, *AttributeNotFoundException*, *InvalidAttributeException*

*AuthorizationException*, *DataException*

---

## 캠페인

API는 캠페인에서 다음 작업을 지원합니다(보안 권한의 적용을 받음).

- 새 캠페인 작성
- 검색(다양한 기준으로 캠페인 나열)
- 속성 작성, 읽기 및 업데이트(속성 API 사용)

캠페인에는 API에서 제공되고 연관되어 있는 많은 표준 속성이 있습니다. 이 목록은 클라이언트가 사용자 정의 속성을 추가하여 원하는 대로 확장할 수 있습니다(속성 API 참조).

표준 캠페인 속성은 다음과 같습니다.

- *uacName*: 캠페인 이름(고유성은 보장되지 않음).
- *uacDescription*: 캠페인을 설명하는 선택적 문자열.
- *uacCampaignCode*: 캠페인을 고유하게 식별하는 문자열 코드. 일반적으로 Campaign에서 자동으로 생성하지만 클라이언트가 제공할 수도 있습니다.
- *uacCreateDate*: 캠페인이 서버에서 작성된 날짜 및 시간을 정의하는 일정.
- *uacUpdateDate*: 캠페인이 서버에서 마지막으로 업데이트된 날짜 및 시간을 표시하는 일정.
- *uacInitiative*: 캠페인 이니셔티브를 정의하는 선택적 문자열.
- *uacObjectives*: 캠페인의 목표를 식별하는 선택적 문자열.
- *uacStartDate*: 캠페인이 서버에서 시작되거나 시작할 예정인 날짜 및 시간을 제공하는 선택적 스케줄.
- *uacEndDate*: *uacStartDate*와 마찬가지로이지만 캠페인이 완료되거나 완료 스케줄링된 날짜 및 시간을 정의합니다. *uacStartDate* 이후여야 합니다.
- *uacLastRunDate*: 캠페인에 바인드된 모든 플로우차트가 마지막으로 실행된 날짜 및 시간을 제공하는 선택적 일정(실행되지 않은 경우 널).
- *uacExternalLinkOwner*: 외부 링크의 소유자 이름을 정의하는 선택적 문자열 (*uacExternalLinkReference* 속성 참조). IBM 사용 전용이고 다음 중 하나이어야 합니다.
  - “Plan”(이제 IBM Marketing Operations로 알려짐)
  - “Collaborate”(이제 IBM Distributed Marketing으로 알려짐)
- *uacExternalLinkId*: 다른 IBM 애플리케이션이 이 캠페인에 링크된 오브젝트로 지정한 선택적 숫자 데이터베이스 ID. IBM 사용 전용이고 *uacExternalLinkOwner* 속성도 참조하십시오.

## generateCampaignCode

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

### 새 캠페인 코드 생성

이 코드는 고유하며 이 메소드 또는 `createCampaign()` 메소드에 대한 이전 또는 이후 호출에서 리턴된 값이나 IBM Campaign GUI를 통해 작성된 캠페인의 생성 값과 다릅니다.

**참고:** 이 메소드의 사용은 선택적입니다. 메소드가 제공되지 않으면 `createCampaign()` API에서 클라이언트에 대한 캠페인 코드를 생성합니다.

`createCampaign()`을 참조하십시오.

## 매개변수

*userCredential*: 클라이언트 사용자 신임 정보.

*requestedLocale*: 이 요청에 사용할 선택적 로케일.

*partitionName*: 사용할 캠페인 파티션의 선택적 이름. Campaign 설치에 하나의 파티션만 있는 경우 이 매개변수는 널(null)입니다.

## 리턴

생성된 캠페인 코드.

## 오류

AuthorizationException, DataException

## createCampaign

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

지정된 속성을 적용하여 클라이언트, 파티션 및 securityPolicyName에 대한 새 캠페인을 작성합니다. 이 API에서 작성한 모든 캠페인은 루트 폴더 아래에 있습니다.

## 매개변수

*userCredential*: 클라이언트 사용자 신임 정보.

*requestedLocale*: 이 요청에 사용할 선택적 로케일.

*partitionName*: 사용할 캠페인 파티션의 선택적 이름.

*securityPolicyName*: 캠페인을 작성하는데 사용하는 캠페인 보안 정책의 선택적 이름. 이 캠페인에 대한 모든 후속 작업은 이 정책을 사용합니다. 정의되지 않은 경우 글로벌 정책이 사용됩니다.

*name*: 새 캠페인 인스턴스를 지정하는 이름(해당 “uacName” 속성).

*attributes*: 초기화 속성의 선택적 배열. 제공된 속성은 캠페인의 기본값을 겹쳐줍니다. 다른 값은 변경되지 않습니다. 예를 들어 *uacCampaignCode* 속성이 제공되는 경우 자동 생성되는 속성을 대신하여 사용됩니다. 캠페인에 필요한 속성, 속성 유형 등을 결정하는 것은 클라이언트의 몫입니다.

하나 이상의 이름 지정된 속성이 없는 경우 *AttributeNotFoundException* 예외를 처리하거나 속성 값이 유효하지 않은 경우(유효하지 않은 데이터 유형) *InvalidAttributeException* 예외가 처리됩니다.

## 리턴

작성된 캠페인에 대한 *CampaignInfo*의 단일 인스턴스.

## 오류

*InvalidAttributeException*, *AttributeNotFoundException*

*AuthorizationException*, *DataException*

## listCampaignsByPage

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

선택적 속성 값과 일치하며 지정된 페이지 오프셋으로 시작하는 캠페인의 "페이지"를 열거합니다. 폴더는 무시됩니다.

일단 검색하면, 리턴된 각 *CampaignInfo*는 요약 목록을 표시하는 경우와 같이 있는 그대로 사용될 수 있습니다. 또는 속성 메소드는 캠페인의 속성을 업데이트하거나 폐치하는데 사용될 수 있습니다.

이 API로 유지보수되는 상태는 없으므로 어느 순서로든 호출을 작성할 수 있습니다.

## 매개변수

*userCredential*: 클라이언트 사용자 신임 정보.

*requestedLocale*: 이 요청에 사용할 선택적 로케일.

*partitionName*: 사용할 캠페인 파티션의 선택적 이름.

*attributes*: 일치하는 속성의 선택적 배열. 속성의 이름, 데이터 유형, 값은 일치를 결정하는데 사용됩니다. 속성이 배열을 지원하는 경우 지정된 모든 값이 일치해야 합니다. 내재된 일치 연산자는 AND이며 제공된 모든 속성 값이 일치하는 캠페인만 리턴됩니다.

속성 이름이 존재하지 않는 경우 *AttributeNotFoundException* 예외를 처리하거나 또는 하나 이상의 제공된 속성이 유효하지 않은 경우 *InvalidAttributeException* 예외가 처리됩니다.

*pageOffset*: 열거(값이 0인)를 시작하는 가능한 모든 캠페인의 시작 오프셋. 예를 들어 열거에서 1000개 캠페인과 일치하고 이 값이 10으로 설정된 경우 페이지는 11번째 구성요소에서 시작합니다. 제공된 오프셋이 범위를 벗어난 경우 *RangeException* 예외가 처리됩니다.

*pageSize*: 페이지에 대해 리턴하는 최대 일치 캠페인의 수(500을 초과할 수 없음).

## 리턴

0개 이상의 *CampaignInfo* 데이터 랩퍼 인스턴스를 페이지에서 일치된 각 캠페인에 대해 하나씩 리턴합니다.

## 오류

*AttributeNotFoundException*, *InvalidAttributeException*, *RangeException*

*InvalidExecutionContextException*, *AuthorizationException*

---

## 대상 셀 메소드

대상 셀은 IBM Campaign에서 대상 셀 스프레드시트(TCS)로 관리되는 캠페인 결과의 알려진 몇몇 서브세트에 대한 추상입니다. 대상 셀은 캠페인에 대해 포괄적이거나 특정 캠페인 플로우차트와 연관될 수 있습니다.

API는 대상 셀에서 다음 작업을 지원합니다.

- 하나 이상의 새 글로벌 대상 셀 작성
- 하나 이상의 기존 대상 셀 대량 업데이트
- 검색(대상 셀 나열)
- 속성 작성, 읽기 및 업데이트(속성 API 사용)
- 기존 대상 셀 삭제
- 하나 이상의 셀과 연관된 실행 결과 폐치

대상 셀에는 API에서 제공되고 연관되어 있는 많은 표준 속성이 있습니다. 이 목록은 클라이언트가 사용자 정의 속성 메타데이터 정의를 추가하여 원하는 대로 확장할 수 있습니다(속성 API 참조). 각각의 속성 메타데이터는 TCS의 열로 생각될 수 있으며 스프레드시트의 레이아웃은 클라이언트가 조정할 수 있습니다.

표준 대상 셀 속성은 다음과 같습니다.

- *uacName*: 셀 이름.
- *uacDescription*: 플로우차트를 설명하는 선택적 문자열.
- *uacCellCode*: 셀을 고유하게 식별하는 코드 문자열. 일반적으로 Campaign에서 자동으로 생성하지만 클라이언트가 제공할 수도 있습니다.
- *uacCreateDate*: 셀이 서버에서 작성된 날짜 및 시간을 제공하는 일정 인스턴스.

- *uacUpdateDate*: 셀이 서버에서 업데이트된 마지막 시간을 정의하는 일정 인스턴스.
- *uacIsControl*: 이 셀이 제어 셀인지(true) 아닌지(false)를 표시하는 부울. 다른 셀은 이 셀을 제어 셀로 참조할 수 있습니다(*uacControlCell* 참조).
- *uacControlCell*: 제어 셀의 선택적 참조(제어 셀의 경우 허용되지 않음). *uacIsControl* 속성을 참조하십시오.
- *uacIsApproved*: 셀이 허용되었는지(true) 아닌지(false)를 표시하는 부울.
- *uacIsReadOnly*: 셀이 읽기 전용인지(true) 아닌지(false)를 표시하는 부울.
- *uacDisplayOrder*: 대상 셀 스프레드시트의 다른 셀과 비교하여 이 셀(열)의 순서를 부여하는 정수.
- *uacIsTopDown*: 셀이 하향식인지를 표시하는 부울.
- *uacAssignedOffers*: 이 셀에 지정된 하나 이상의 오퍼 또는 오퍼 목록의 참조에 대한 선택적 배열(제어 셀의 경우 허용되지 않음).
- *uacFlowchartName*: 이 셀이 링크된 플로우차트의 선택적 이름(IBM Campaign GUI를 통해 읽기 전용으로 설정되어야 하고 제어 셀의 경우 허용되지 않음).
- *uacFlowchartId*: 이 셀이 링크된 플로우차트의 선택적 데이터베이스 ID(위와 같이 읽기 전용임).

## createTargetCell

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

지정된 셀당 속성 및 사용자 정보를 적용하여 새 캠페인 특정 셀 행을 작성합니다.

지정된 속성은 표준이거나 또는 사용자 정의일 수 있습니다. 그러나 사용자 정의라면 해당 글로벌 속성 메타데이터 정의가 이미 있어야 합니다.

일단 대상 셀이 작성되면, 속성 값은 속성 API를 사용하여 변경할 수 있습니다.

`listTargetCells()`, `bulkCreateTargetCells()`를 참조하십시오.

`createAttributeMetadata()`, `listAttributeMetadata()`, `getAttributesByName()`를 참조하십시오.

### 매개변수

*userCredential*: 클라이언트 사용자 신임 정보.

*requestedLocale*: 이 요청에 사용할 선택적 로케일.

*partitionName*: 사용할 캠페인 파티션의 선택적 이름.

*campaignReference*: 업데이트할 대상 셀 스프레드시트를 포함하는 캠페인의 참조. 캠페인이 존재하지 않거나 참조가 유효하지 않은 경우 *InvalidComponentException*이 누적됩니다.

*attributes*: 새 셀에 대한 TCS 속성의 선택적 배열. 제공된 각 속성 요소는 해당 셀 속성의 기본값을 겹쳐줍니다. 다른 값은 변경되지 않습니다. 셀, 셀 유형 등에서 필요한 속성을 결정하는 것은 클라이언트의 몫입니다. 지정된 속성에 문제점이 있는 경우 *InvalidAttributeException*이 누적됩니다.

예외가 누적되는 경우 이 메소드는 *CompositeException* 예외 처리를 하고 모든 작업이 완료되지 않습니다. 예외의 원인 목록에는 오류를 발생시킨 각 속성에 대한 예외가 포함되고 *reference* 대신 숫자 색인 및 속성 이름이, 일반적으로 올바른 값이 포함됩니다. 원인 목록은 *attributeList*의 입력 순서대로 정렬됩니다.

## 리턴

TCS 셀에 대한 *TargetCellInfo* 데이터 래퍼.

## 오류

*InvalidComponentException*, *CompositeException*

*AuthorizationException*, *DataException*

## bulkCreateTargetCells

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

지정된 셀당 속성 및 사용자 정보를 사용하여 새 캠페인 특정 대상 셀 행을 한 번에 여러 개 작성합니다.

지정된 속성은 표준이거나 또는 사용자 정의일 수 있습니다. 그러나 사용자 정의라면 해당 글로벌 속성 메타데이터 정의가 이미 있어야 합니다.

일단 대상 셀이 작성되면, 속성 값은 속성 API를 사용하여 변경할 수 있습니다.

*listTargetCells()*를 참조하십시오.

*createAttributeMetadata()*, *listAttributeMetadata()*, *getAttributesByName()*를 참조하십시오.



## 매개변수

*userCredential*: 클라이언트 사용자 신임 정보.

*requestedLocale*: 이 요청에 사용할 선택적 로케일.

*partitionName*: 사용할 캠페인 파티션의 선택적 이름.

*campaignReference*: 업데이트할 대상 셀 스프레드시트를 포함하는 캠페인의 참조. 캠페인이 존재하지 않거나 참조가 유효하지 않은 경우 *InvalidComponentException*이 누적됩니다.

*attributeList*: 셀당 속성 배열의 선택적 목록, 작성할 각 대상 셀 행에 대해 하나씩. 특정 목록 요소에 대해 제공된 모든 속성은 해당 셀 속성의 기본값을 겹쳐줍니다. 다른 값은 변경되지 않습니다. 셀, 셀 유형 등에서 필요한 속성을 결정하는 것은 클라이언트의 몫입니다. 지정된 속성에 문제점이 있는 경우 *InvalidAttributeException*이 누적됩니다.

예외가 누적되는 경우 이 메소드는 *CompositeException* 예외 처리를 하고 모든 작업이 완료되지 않습니다. 예외의 원인 목록에는 오류를 발생시킨 각 속성에 대한 예외가 포함되고 *reference* 대신 숫자 색인 및 속성 이름 등이 포함됩니다. 원인 목록은 *attributeList*의 입력 순서대로 정렬됩니다.

## 리턴

입력 *attributesList* 매개변수의 요소 순서 대로 작성된 각 인스턴스에 대해 하나인 *TargetCellInfo* 데이터 랩퍼의 목록.

## 오류

*InvalidComponentException*, *CompositeException*

*AuthorizationException*, *DataException*

## listTargetCells

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

지정된 속성과 일치하는 현재 존재하는 모든 대상 셀에 대한 정보를, 지정된 캠페인에 대해 또는 캠페인이 지정되지 않은 경우 전체적으로 열거합니다.

*getAttributeMetadata()*, *getAttributesByName()*를 참조하십시오.

## 매개변수

*userCredential*: 클라이언트 사용자 신임 정보.

*requestedLocale*: 이 요청에 사용할 선택적 로케일.

*partitionName*: 사용할 캠페인 파티션의 선택적 이름.

*campaignReference*: 상위 캠페인의 참조. 캠페인이 존재하지 않거나 참조가 유효하지 않은 경우 *InvalidComponentException* 예외가 처리됩니다.

*attributes*: 일치할 속성의 선택적 배열. 내재된 일치 연산자는 AND이며 제공된 모든 속성 값이 일치하는 셀만 리턴됩니다.

하나 이상의 지정된 속성이 유효하지 않으면 *InvalidAttributeException* 예외가 처리됩니다.

## 리턴

일치된 셀에 대해 0개 이상의 *TargetCellInfo* 인스턴스의 목록을 리턴합니다.

## 오류

*InvalidComponentException*, *InvalidAttributeException*

*AuthorizationException*, *DataException*

## bulkUpdateTargetCells

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

하나 이상의 대상 셀의 속성을 업데이트합니다.

업데이트 로직은 다음과 같습니다.

제공된 *attributesMap*의 각 요소의 경우, 항목 키는 업데이트할 대상 셀의 참조이며 항목 값은 해당 셀에 대한 업데이트 속성의 배열입니다. 대상 셀이 없는 경우에는 *InvalidComponentException*을 수집하십시오.

대상 셀이 찾아지면 지정된 각 속성에 대해 다음을 수행하십시오.

1. 속성 이름이 기존 속성과 일치하는 경우 제공된 값 필드로 기존 값 필드 겹쳐쓰기를 시도하십시오.
2. 값 유형 또는 속성 메타데이터 정의의 일부 다른 요소가 부합되지 않는 경우 또는 하나 이상의 제공된 값이 유효하지 않거나 범위를 벗어나는 등의 경우 *InvalidAttributeException*을 수집하십시오.

3. 그렇지 않고 이름 붙여진 속성이 없는 경우 *AttributeNotFoundException*을 수집하십시오.

예외가 수집되는 경우 이 메소드는 *CompositeException* 예외 처리를 하고 모든 업데이트는 완료되지 않습니다. 예외의 원인 목록에는 위에 나열된 예외가 포함됩니다. 오류를 발생시킨 각 속성에 대해, 참조 및 속성 이름 두 가지 모두 기록됩니다.

모든 경우에 있어서 속성 업데이트 작업은 일반 보안 제한조건 및 유효성 검증의 적용을 받습니다. 어떤 속성이 특정 구성요소 인스턴스에 필요한지, 올바른 유형은 무엇인지 등을 결정하는 것은 클라이언트의 책임입니다.

### 매개변수

*userCredential*: 클라이언트 사용자 신임 정보.

*requestedLocale*: 이 요청에 사용할 선택적 로케일.

*partitionName*: 사용할 캠페인 파티션의 선택적 이름.

*attributesMap*: 업데이트할 대상 셀의 맵. 항목 키는 업데이트할 셀의 참조이며 항목 값은 업데이트 속성의 배열입니다. 속성 이름은 업데이트할 속성을 찾는데 사용되며 새 속성 값은 기존 속성의 값을 적합한 유형 또는 배열의 단순 오브젝트로 업데이트하는데 사용됩니다(적용 가능한 경우). 위 예외를 참조하십시오.

### 리턴

없음

### 오류

ComponentException

AuthorizationException, DataException

## getRunResultsByCell

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

한 번도 시작하지 않았거나 또는 여전히 진행 중인 플로우차트에 대해 가능한, 하나 이상의 대상 셀의 실행 결과를 가져옵니다.

### 매개변수

*userCredential*: 클라이언트 사용자 신임 정보.

*requestedLocale*: 이 요청에 사용할 선택적 로케일.

*partitionName*: 사용할 캠페인 파티션의 선택적 이름.

*cellReferences*: 실행 결과가 요구되는 대상 셀의 참조 배열. 하나 이상의 셀 참조가 유효하지 않거나 존재하지 않는 셀을 참조하는 경우 *InvalidComponentException* 예외를 처리합니다.

## 리턴

입력 참조 배열에 따라 정렬된, 이름 지정된 셀에 대한 실행 결과의 형식화된 목록을 리턴합니다.

각 실행 상태는, 하위 플로우차트 프로세스 상자가 여전히 실행 중인 경우 **RUNNING**, 어떤 이유 때문에 실패한 경우 **FAILED**, 프로세스 상자 실행이 시작되지 않은 경우라면 **NOT\_STARTED**로 표시됩니다. 상태 세부사항도 제공됩니다.

## 오류

*InvalidComponentException*

*AuthorizationException*, *DataException*

## bulkDeleteTargetCells

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

하나 이상의 기존 대상 셀 및 모든 종속 구성요소(플로우차트 링크, 속성 등)를 삭제합니다.

## 매개변수

*userCredential*: 클라이언트 사용자 신임 정보.

*requestedLocale*: 이 요청에 사용할 선택적 로케일.

*partitionName*: 사용할 캠페인 파티션의 선택적 이름.

*cellReferences*: 삭제될 셀의 하나 이상의 참조 배열. *InvalidComponentException*은 지정된 참조 중 하나에 문제점이 있거나 셀이 존재하지 않는 경우 누적됩니다.

예외가 누적되면 이 메소드에서 *CompositeException* 예외를 처리하며 모든 삭제 작업이 완료되지 않습니다. 예외의 원인 목록에는 위에 나열된 예외가 포함됩니다. 오류를 발생시킨 각 속성에 대해 참조가 기록됩니다.

## 리턴

없음

## 오류

CompositeException

AuthorizationException, DataException

---

## 분석

API는 IBM Campaign에서 단순 메트릭의 검색을 지원합니다.

### getCampaignMetrics

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

지정된 캠페인에 대한 메트릭을 페치합니다.

#### 매개변수

*userCredential*: 클라이언트 사용자 신임 정보.

*requestedLocale*: 이 요청에 사용할 선택적 로케일.

*partitionName*: 사용할 캠페인 파티션의 선택적 이름.

*campaignReference*: 상위 캠페인의 참조. 캠페인 참조에 문제점이 있거나 캠페인이 존재하지 않는 경우 *InvalidComponentException* 예외가 처리됩니다.

#### 리턴

캠페인에 대한 *MetricsInfo* 인스턴스로 리턴합니다.

## 오류

InvalidComponentException

AuthorizationException, DataException

---

## 오퍼, 오퍼 목록 및 오퍼 템플릿 메소드

API는 오퍼에서 다음 작업을 지원합니다.

- 검색(폴더(오퍼, 오퍼 목록, 하위 폴더), 속성(오퍼 및 오퍼 템플릿) 또는 검색 값(오퍼))
- 유효성 검증
- 정보 검색(특정 오퍼 또는 오퍼 템플릿의 속성 검색)
- 오퍼 작성, 편집, 폐기 및 삭제

오퍼에는 연관되어 있는 많은 표준 속성이 있습니다. 이 목록은 클라이언트가 사용자 정의 속성 메타데이터 정의를 추가하여 원하는 대로 확장할 수 있습니다(속성 API 참조).

표준 오퍼 속성은 다음과 같습니다.

- *uacName*: 오퍼 이름.
- *uacDescription*: 오퍼를 설명하는 선택적 문자열.
- *uacOfferCode*: 오퍼를 고유하게 식별하는 코드 문자열. 일반적으로 IBM Campaign 에서 자동으로 생성하지만 클라이언트가 제공할 수도 있습니다.
- *uacCreateDate*: 오퍼가 서버에서 작성된 날짜 및 시간을 제공하는 일정 인스턴스.
- *uacUpdateDate*: 오퍼가 서버에서 업데이트된 마지막 시간을 정의하는 일정 인스턴스.

오퍼 템플릿도 표준 및 사용자 정의 속성을 가지고 있습니다. 표준 오퍼 템플릿 속성은 다음과 같습니다.

- *uacName*: 오퍼 템플릿 이름.
- *uacDescription*: 오퍼 템플릿을 설명하는 선택적 문자열.
- *uacCreateDate*: 서버에서 오퍼 템플릿을 작성한 날짜 및 시간을 제공하는 일정 인스턴스.
- *uacUpdateDate*: 서버에서 오퍼 템플릿을 업데이트한 마지막 시간을 정의하는 일정 인스턴스.

## listOffersAndFolders

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

선택적 상위 폴더 아래에 있는 오퍼, 오퍼 목록, 폴더를 열거합니다.

일단 검색하면, 리턴된 각 *WSComponentOrFolderInfo* 인스턴스는 있는 그대로 사용될 수 있습니다. 예를 들어 폴더 계층 구조의 다음 레벨을 표시하려면 속성 API를 포함한 오퍼를 업데이트하거나 폐치하는데 사용할 수 있습니다.

### 매개변수

*userCredential*: 클라이언트 사용자 신임 정보.

*requestedLocale*: 이 요청에 사용할 선택적 로케일.

*partitionName*: 사용할 캠페인 파티션의 선택적 이름.

*parentReference*: 나열할 상위 폴더의 선택적 참조. 바로 하위 오퍼, 오퍼 목록, 이 상위 폴더의 폴더만 누적되기 때문에 전체 폴더 계층 구조를 탐색하려면 이 API에 대한 연속적인 호출이 필요합니다. 그러나 일반적으로 구조는 아주 간단합니다. 상위가 제공되지 않은 경우에는 루트 아래의 모든 구성요소 및 폴더가 리턴됩니다.

지정된 상위 폴더 참조에 문제점이 있는 경우 *InvalidFolderException* 예외를 처리합니다.

0개 이상의 *WSComponentOrFolderInfo* 데이터 래퍼 인스턴스의 형식화된 목록, 일치된 각 구성요소 또는 폴더에 하나씩.

## 오류

*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;
```

지정된 페이지 오프셋으로 시작하고 이름, 설명, 작성자 또는 오퍼 코드 필드의 주어진 검색 기준을 포함하는 오퍼의 "페이지"를 열거합니다. 검색은 선택적 폴더 입력을 기준으로 합니다. (folderID에 0이 제공되면 루트 오퍼 폴더가 기본적으로 사용됩니다.) 일치하는 검색 문자열의 경우 "contains" 일치를 기준으로 리턴됩니다.

일단 검색하면 리턴된 각 *WSOfferInfo*는 요약 목록을 표시하는 경우와 같이 있는 그대로 사용될 수 있습니다. 또는 속성 메소드는 오퍼 속성을 업데이트하거나 폐치하는데 사용될 수 있습니다.

이 API에서 유지보수하는 상태는 없으므로 어느 순서로든 호출을 작성할 수 있습니다.

## 매개변수

*userCredential*: 클라이언트 사용자 신임 정보.

*requestedLocale*: 이 요청에 사용할 선택적 로케일.

*partitionName*: 사용할 캠페인 파티션의 선택적 이름.

*folderID*: 검색할 오퍼 폴더의 ID. folderID에 0을 지정하면 루트 폴더가 검색됩니다.

*searchCriteria*: 검색 구문.

*includeRetired*: 검색 결과에 폐기된 오퍼를 포함할지 여부를 지정하는 부울 값. 유효한 값은 TRUE 및 FALSE. TRUE는 폐기된 오퍼를 포함하며 FALSE는 폐기된 오퍼는 포함하지 않음을 표시합니다.

*pageOffset*: 열거(값이 0인)를 시작하는 가능한 모든 구성요소의 시작 오프셋. 예를 들어 열거에서 1000개 오퍼와 일치하고 이 값이 10으로 설정된 경우 페이지는 11번째 구성요소에서 시작합니다. 제공된 오프셋이 범위를 벗어난 경우 `RangeException` 예외가 처리됩니다.

*pageSize*: 페이지에 대해 리턴하는 최대 일치 구성요소의 수(500을 초과할 수 없음).

## 리턴

0개 이상의 *Offer* 데이터 랩퍼 인스턴스의 형식화된 목록을 페이지에서 일치된 각 구성요소에 대해 하나씩 리턴합니다.

## 오류

`RangeException`

## listOffersByPage

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

선택적 속성 값과 일치하며 지정된 페이지 오프셋으로 시작하는 오퍼의 "페이지"를 열거합니다. 폴더는 무시됩니다. 문자열에 대해서는 "like" 일치를 기준으로 (일치는 문자열이 쿼리를 사용한 값을 포함하는 경우 충분하다고 간주되는) 그리고 날짜 및 숫자에 대해서는 정확한 일치를 기준으로 리턴됩니다.

일단 검색하면, 리턴된 각 *OfferInfo*는 요약 목록을 표시하는 경우와 같이 있는 그대로 사용될 수 있습니다. 또는 속성 메소드는 오퍼의 속성을 업데이트하거나 폐치하는데 사용될 수 있습니다.

이 API에서 유지보수하는 상태는 없으므로 어느 순서로든 호출을 작성할 수 있습니다.

## 매개변수

*userCredential*: 클라이언트 사용자 신임 정보.

*requestedLocale*: 이 요청에 사용할 선택적 로케일.

*partitionName*: 사용할 캠페인 파티션의 선택적 이름.



*attributes*: 일치하는 속성의 선택적 배열. 속성의 이름, 데이터 유형 및 값은 일치를 결정하는데 사용됩니다. 속성이 배열을 지원하는 경우 지정된 모든 값이 일치해야 합니다. 내재된 일치 연산자는 OR이며 제공된 모든 속성 값이 일치하는 구성요소만 리턴됩니다.

속성 이름이 존재하지 않는 경우 *AttributeNotFoundException* 예외를 처리하거나 또는 하나 이상의 제공된 속성이 유효하지 않은 경우 *InvalidAttributeException* 예외가 처리됩니다.

*pageOffset*: 열거(값이 0인)를 시작하는 가능한 모든 구성요소의 시작 오프셋. 예를 들어 열거에서 1000개 오프와 일치하고 이 값이 10으로 설정된 경우 페이지는 11번째 구성요소에서 시작합니다. 제공된 오프셋이 범위를 벗어난 경우 *RangeException* 예외가 처리됩니다.

*pageSize*: 페이지에 대해 리턴하는 최대 일치 구성요소의 수(500을 초과할 수 없음).

## 리턴

0개 이상의 *OfferInfo* 데이터 랩퍼 인스턴스의 형식화된 목록을 페이지에서 일치된 각 구성요소에 대해 하나씩 리턴합니다.

## 오류

*AttributeNotFoundException*, *InvalidAttributeException*, *RangeException*

*InvalidExecutionContextException*, *AuthorizationException*

## validateOffers

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

제공된 오퍼 코드 또는 오퍼 목록 이름의 유효성을 검증하고 각각의 유효성 검증 정보를 리턴합니다. “유효성 검증”은 데이터베이스에 하나와 하나의 일치하는 오퍼가 있는지 오퍼 목록이 있는지 확인하도록 구성됩니다.

*OfferValidationInfo* 오브젝트는 제공된 코드 또는 이름과 일치하는 오퍼 또는 오퍼 목록이 없는 경우 Offer 정보 대신에 오류 메시지를 포함합니다. 또한 제공된 코드 또는 이름이 여러 오퍼 또는 오퍼 목록과 일치하는 경우에도 일치 대신 오류가 리턴됩니다. 제공된 순서대로 리턴됩니다. 오퍼 코드 및 오퍼 목록 이름은 오퍼의 정확한 일치를 기준으로 유효성을 검증합니다.

## 매개변수

*userCredential*: 클라이언트 사용자 신임 정보.

*requestedLocale*: 이 요청에 사용할 선택적 로케일.

*partitionName*: 사용할 캠페인 파티션의 선택적 이름.

*codeOrNames*: 유효성을 검증할 모든 오퍼 코드 또는 오퍼 목록 이름의 배열.

**참고:** 이 메소드에서는 예외를 처리하지 않습니다. 대신 유효성 검증 정보가 모든 코드 또는 제공된 이름에 대해 리턴됩니다.

## 리턴

0개 이상의 *OfferValidationInfo* 데이터 래퍼 인스턴스의 형식화된 목록.

## 오류

없음

## 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;
```

지정된 속성을 적용하여 클라이언트에 대한 새 오퍼를 작성합니다.

## 매개변수

*authorizationLoginName*: 오퍼를 작성하는 사용자의 이름. 이 메소드를 사용하려면 오퍼 추가 권한이 사용자에게 부여되어야 합니다.

*partitionName*: 사용할 캠페인 파티션의 선택적 이름.

*requestedLocale*: 이 요청에 사용할 선택적 로케일.

*securityPolicyName*: 오퍼를 작성하는 데 사용할 캠페인 보안 정책의 선택적 이름. 이 오퍼에 대한 모든 후속 작업은 이 정책을 사용합니다. 정의되지 않은 경우에는 글로벌 정책이 사용됩니다.

*name*: 새 오퍼 인스턴스를 지정하는 이름(해당 *uacName* 속성).

*folderID*: 오퍼가 작성될 오퍼 폴더의 ID. 이 ID는 정확성에 대한 유효성이 검증되며 ID가 유효하지 않을 경우 예외 처리를 합니다.

*templateName*: 새 오퍼에 사용되어야 하는 기존 오퍼 템플릿의 필수(고유) 이름.

*wsAttributes*: 초기화 속성의 배열. 제공된 모든 속성은 오퍼의 기본값을 겹쳐쓰고 나머지는 변경되지 않습니다. 예를 들어 *uacOfferCode* 속성이 제공되는 경우 자동으로 생성되는 속성을 대신하여 사용됩니다. 오퍼에 필요한 속성, 속성 유형 등을 결정하는 것은 클라이언트의 몫입니다.

다음 조건 중 하나가 발생하는 경우 *CampaignServicesException* 예외 처리를 합니다.

- *folderID* 매개변수가 유효하지 않습니다(존재하지 않거나 오퍼 유형이 아님).
- 사용자에게 이 작업을 수행할 권한이 부여되지 않았습니다.
- 유효하지 않은 속성이 *wsAttributes*에서 제공됩니다.
- 기타 런타임 예외가 발생합니다.

## 리턴

작성된 오퍼에 대한 *OfferInfo*의 단일 인스턴스.

## 오류

*CampaignServicesException*

## retireOffers

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

하나 이상의 기존 오퍼를 폐기합니다.

## 매개변수

*userCredential*: 클라이언트 사용자 신임 정보.

*requestedLocale*: 이 요청에 사용할 선택적 로케일.

*partitionName*: 사용할 캠페인 파티션의 선택적 이름.

*references*: 폐기될 오퍼의 참조 배열. *InvalidComponentException*은 특정 참조에 문제점이 있거나 오퍼가 존재하지 않는 경우 예외 처리됩니다.

## 리턴

없음

## 오류

InvalidComponentException

AuthorizationException, DataException

## deleteOffers

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

하나 이상의 기존 오퍼를 삭제합니다.

### 매개변수

*userCredential*: 클라이언트 사용자 신임 정보.

*requestedLocale*: 이 요청에 사용할 선택적 로케일.

*partitionName*: 사용할 캠페인 파티션의 선택적 이름.

*reference*: 삭제될 오퍼의 참조 배열. *InvalidComponentException*은 지정된 참조에 문제점이 있거나 오퍼가 존재하지 않는 경우 예외 처리됩니다.

### 리턴

없음

## 오류

InvalidComponentException

AuthorizationException, DataException

## listOfferTemplates

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

사용자가 볼 수 있는 권한이 있는 모든 오퍼 템플릿을 열거합니다.

일단 검색하면 리턴된 각 *WSOfferTemplateInfo* 인스턴스는 있는 그대로 사용될 수 있고 또는 하나 이상의 속성 API를 나열된 템플릿을 업데이트하거나 폐치하는데 사용할 수 있습니다.

### 매개변수

*userCredential*: 클라이언트 사용자 신임 정보.

*requestedLocale*: 이 요청에 사용할 선택적 로케일.

*partitionName*: 사용할 캠페인 파티션의 선택적 이름.

## 리턴

0개 이상의 *WSOfferTemplateInfo* 데이터 래퍼 인스턴스의 형식화된 목록을 리턴된 각 템플릿에 하나씩 리턴합니다.

## 오류

*InvalidExecutionContextException*, *AuthorizationException*

*DataException*

## bulkCreateOffers

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

*offers* 매개변수에서 지정된 각 오퍼에 대한 속성으로 벌크에서 오퍼를 작성합니다. 모든 오퍼는 지정된 *templateName*을 사용하여 지정된 *folderID* 아래에 작성됩니다.

## 매개변수

*authorizationLoginName*: 클라이언트 사용자 신임 정보.

*partitionName*: 사용할 캠페인 파티션의 선택적 이름.

*requestedLocale*: 이 요청에 사용할 선택적 로케일.

*securityPolicyName*: 오퍼를 작성하는 데 사용할 캠페인 보안 정책의 선택적 이름. 정의되지 않은 경우 글로벌 정책이 사용됩니다.

*templateName*: 시스템의 기존 오퍼 템플릿 이름. 모든 오퍼는 이 템플릿을 사용하여 작성됩니다.

*folderID*: 오퍼가 작성될 오퍼 폴더의 ID. 이 ID는 유효성을 검증하며 ID가 유효하지 않을 경우 예외 처리를 합니다.

*offers*: 오퍼 이름 및 속성을 정의하는 *WSBulkOfferInfo* 오브젝트의 배열. 자세한 내용은 *WSBulkOfferInfo* 데이터 유형을 참조하십시오.

## 리턴

각 오퍼에 대한 *WSOfferInfoStatus* 인스턴스의 배열. 상태 및 오퍼 정보를 포함합니다. 상태는 오퍼 작성이 성공했는지 여부를 표시합니다.

## 오류

CampaignServicesException

---

## 제 5 장 API 사용

이 섹션에서는 Campaign 웹 서비스 API를 사용하는 방법에 대해 설명합니다. 또한 Campaign API 서비스를 사용하여 Campaign에서 오퍼를 작성하는 예제를 표시하고 있습니다.

Campaign 서비스 API를 사용하는 다음의 두 가지 접근 방식이 있습니다.

- 『클라이언트 API .jar 사용』
- 53 페이지의 『WSDL 사용』

---

### 클라이언트 API .jar 사용

IBM Campaign은 Campaign 웹 애플리케이션과 상호작용하도록 SOAP 웹 서비스를 사용하는 클라이언트 API를 제공합니다. 이 랩퍼는 .jar 파일에 번들되며 클라이언트 애플리케이션은 Campaign API를 호출하는데 사용할 수 있습니다.

이 .jar 파일은 다음 위치에서 찾을 수 있습니다.

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

다음 예제에서 Campaign의 루트 오퍼 폴더 레벨의 새 오퍼 작성을 보여줍니다. 동일한 샘플은 다음 위치에서 찾을 수 있습니다.

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

**참고:** 예제는 매개변수에 대해 일부 더미 값을 사용합니다. 사용자 실제 값은 다를 수 있습니다.

Campaign 웹 서비스에 대한 URL은 `http://host:port/Campaign/services/CampaignServices30Service`이며 여기서 호스트 및 포트는 Campaign 웹 애플리케이션이 배치된 시스템의 호스트 이름 및 포트 번호를 말합니다.

제공된 샘플을 사용하는 경우 사용자 클라이언트 환경에 맞게 수정했는지 확인하십시오.

#### 코드 OfferAPI.java

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

```

CampaignServices30SoapClient;
import com.unica.publicapi.campaign.campaignservices.soap.v30.WSAttribute;
import com.unica.publicapi.campaign.campaignservices.soap.v30.WSOfferInfo;
import com.unica.publicapi.campaign.campaignservices.utils.WSAttributeUtils;

/**
 * 이 클래스는 Campaign의 SOAP 서비스 API 사용법을
 * 보여주는 샘플 Java 클라이언트 클래스입니다.
 * 이 샘플은 CampaignServices30SoapClient 퍼사드(facade)를 사용하여 Campaign
 * 웹 서비스와 상호작용합니다.
 * 여기에 Offer 작성법이 표시되어 있습니다. 자세한 내용은 API 안내서를
 * 참조하십시오.
 *
 * @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();
    }
}
}

```

**참고:** 위에 있는 Java 샘플을 컴파일하고 실행하려면 Java 클래스 경로에 있는 모든 종속 .jar 파일을 포함시켜야 합니다. CampaignServicesClient30.jar 파일은 Apache AXIS2 SOAP 엔진 .jar 파일 및 기타 공용 Apache .jar 파일에 종속되며 <CAMPAIGN\_HOME>/Campaign.war에 위치한 Campaign.war 파일에서 찾을 수 있습니다. .jar 파일을 Campaign.war에서 추출하여 Java 클래스 경로에 포함시키십시오.

---

## WSDL 사용

Campaign 서비스는 다음 Campaign 웹 서비스 WSDL 파일을 사용하여 호출될 수 있습니다.

CampaignServices30.wsdl

이 파일은 다음 위치에서 찾을 수 있습니다.

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

또는 Campaign 배포에서는 다음 위치에 있습니다.

<CAMPAIGN\_HOME>/devkits/CampaignServicesAPI/lib/

클라이언트 Java 애플리케이션은 모든 썬드파티 WSDL-to-Java 변환기 도구를 사용하여 WSDL에서 생성된 클래스 및 스텝을 사용할 필요가 있습니다. IBM 에서는 Apache AXIS 사용을 권장합니다.

Apache AXIS2를 사용하여 WSDL에서 생성된 클래스 및 스텝을 통해 작성된 javadocs는 다음 위치에서 찾을 수 있습니다.

<CAMPAIGN\_HOME>/devkits/CampaignServicesAPI/javadocs/index.html

**참고:** 모든 종속 .jar 파일은 Java 클래스 경로에 포함되어야 합니다.

CampaignServicesClient30.jar 파일은 Apache AXIS2 SOAP 엔진 .jar 파일 및 기타 공용 Apache .jar 파일에 종속되며 <CAMPAIGN\_HOME>/Campaign.war에 위치한 Campaign.war 파일에서 찾을 수 있습니다. .jar 파일을 Campaign.war에서 추출하여 Java 클래스 경로에 포함시키십시오.

---

## 성과 고려사항

현재의 CampaignServices API 구현 성과 프로파일은 GUI를 통해 경험한 대로 애플리케이션의 프로파일과 유사합니다. 일부 API는 성과에 대해 명시적으로 디자인되었습니다. 특히 `listCampaignsByPage()` API는 상대적으로 효과적인 페이지 매김을 허용합니다.

속성에 의해 SOAP 인터페이스는 모든 데이터가 XML 양식으로 변환되기 때문에(일부 경우는 정확히 반대로) 지연 및 오버헤드를 도입합니다. 예를 들어, 단순 루프백 SOAP 호출은 일반적 네트워크에서 100ms이 소요됩니다(Java 1.4.x는 더 느림). API는 일반 포털에 대해 최적화되었으며 다른 클라이언트 애플리케이션 비즈니스에서는 `see listOffersByPage()`와 같은 경우를 사용하므로 SOAP 성과가 적합해야 합니다.

그러나 클라이언트는 웹 사용자 요청을 위해 일반 CampaignServices를 지원하는데 너무 큰 부담을 갖길 원하지 않습니다. 일반적으로 API 사용자 처리는 일반 IBM Campaign 웹 사용자의 처리를 초과하지 않을 것으로 예상됩니다.

---

## 패키징

이 스펙은 IBM Campaign과 함께 설치된 CampaignServices Software Developer's Toolkit(devkits)의 일부로 제공됩니다.

설치 프로그램에서 설치한 devkits 디렉토리에는 예제, 빌드 및 텍스트 스크립트, public 클래스 및 인터페이스에 대한 javadoc, 릴리스 노트 등이 포함되어 있습니다.

---

## IBM 기술 지원 담당자에게 문의

문서를 참조해도 문제점을 해결할 수 없는 경우, 회사의 전담 지원 담당자가 IBM 기술 지원 담당자와의 통화를 기록할 수 있습니다. 이 절의 정보를 사용하여 문제점을 효율적으로 해결하십시오.

회사의 전담 지원 담당자가 아닌 경우에는 IBM 관리자에게 문의하여 정보를 얻을 수 있습니다.

### 수집해야 하는 정보

IBM 기술 지원 담당자에게 문의하기 전에 다음 정보를 수집해야 합니다.

- 문제점의 특성에 대한 간단한 설명
- 해당 문제점이 발생할 때 표시되는 자세한 오류 메시지
- 문제점을 재현할 수 있는 자세한 단계
- 관련 로그 파일, 세션 파일, 구성 파일 및 데이터 파일
- "시스템 정보"에서 설명한 방법에 따라 얻을 수 있는 제품 및 시스템 환경에 대한 정보

### 시스템 정보

IBM 기술 지원 담당자와 통화할 때 환경 정보를 요청하는 경우가 있습니다.

문제점 때문에 로그인에 불가능한 경우 외에는, 설치된 IBM 애플리케이션에 대한 정보를 제공하는 제품 정보 페이지에서 이러한 정보 대부분을 얻을 수 있습니다.

도움말 > 제품 정보를 선택하여 제품 정보 페이지에 액세스할 수 있습니다. 제품 정보 페이지에 액세스할 수 없는 경우에는 각 애플리케이션의 설치 디렉토리 아래에 있는 version.txt 파일을 사용하여 모든 IBM 애플리케이션의 버전 번호를 알 수 있습니다. version.txt 파일을 사용할 수 없는 경우에는 NetTracker admin -buildinfo 명령을 사용하여 버전 정보를 얻을 수 있습니다.

### IBM 기술 지원 담당자에게 문의

IBM 기술 지원 담당자에게 문의하는 방법은 IBM 제품 기술 지원 웹 사이트 (<http://www.unica.com/about/product-technical-support.htm>)를 참조하십시오.



---

## 주의사항

이 정보는 미국에서 제공되는 제품 및 서비스용으로 작성된 것입니다.

IBM은 다른 국가에서 이 책에 기술된 제품, 서비스 또는 기능을 제공하지 않을 수도 있습니다. 현재 사용할 수 있는 제품 및 서비스에 대한 정보는 한국 IBM 담당자에게 문의하십시오. 이 책에서 IBM 제품, 프로그램 또는 서비스를 언급했다고 해서 해당 IBM 제품, 프로그램 또는 서비스만을 사용할 수 있다는 것을 의미하지는 않습니다. IBM의 지적 재산을 침해하지 않는 한, 기능상으로 동등한 제품, 프로그램 또는 서비스를 대신 사용할 수도 있습니다. 그러나 비IBM 제품, 프로그램 또는 서비스의 운영에 대한 평가 및 검증은 사용자의 책임입니다.

IBM은 이 책에서 다루고 있는 특정 내용에 대해 특허를 보유하고 있거나 현재 특허 출원 중일 수 있습니다. 이 책을 제공한다고 해서 특허에 대한 라이선스까지 부여하는 것은 아닙니다. 라이선스에 대한 의문사항은 다음으로 문의하십시오.

135-700

서울특별시 강남구 도곡동 467-12, 군인공제회관빌딩

한국 아이.비.엠 주식회사

고객만족센터

전화번호: 080-023-8080

2바이트(DBCS) 정보에 관한 라이선스 문의는 한국 IBM 고객만족센터에 문의하거나 다음 주소로 서면 문의하시기 바랍니다.

Intellectual Property Licensing

Legal and Intellectual Property Law

IBM Japan Ltd.

1623-14, Shimotsuruma, Yamato-shi

Kanagawa 242-8502 Japan

다음 단락은 현지법과 상충하는 영국이나 기타 국가에서는 적용되지 않습니다. IBM은 타인의 권리 비침해, 상품성 및 특정 목적에의 적합성에 대한 묵시적 보증을 포함하여 (단, 이에 한하지 않음) 명시적 또는 묵시적인 일체의 보증 없이 이 책을 "현상태대로" 제공합니다. 일부 국가에서는 특정 거래에서 명시적 또는 묵시적 보증의 면책사항을 허용하지 않으므로, 이 사항이 적용되지 않을 수도 있습니다.

이 정보에는 기술적으로 부정확한 내용이나 인쇄상의 오류가 있을 수 있습니다. 이 정보는 주기적으로 변경되며, 변경된 사항은 최신판에 통합됩니다. IBM은 이 책에서 설명한 제품 및/또는 프로그램을 사전 통지 없이 언제든지 개선 및/또는 변경할 수 있습니다.

이 정보에서 언급되는 비IBM의 웹 사이트는 단지 편의상 제공된 것으로, 어떤 방식으로든 이들 웹 사이트를 옹호하고자 하는 것은 아닙니다. 해당 웹 사이트의 자료는 본 IBM 제품 자료의 일부가 아니므로 해당 웹 사이트 사용으로 인한 위험은 사용자 본인이 감수해야 합니다.

IBM은 귀하의 권리를 침해하지 않는 범위 내에서 적절하다고 생각하는 방식으로 귀하가 제공한 정보를 사용하거나 배포할 수 있습니다.

(i) 독립적으로 작성된 프로그램과 기타 프로그램(본 프로그램 포함)간의 정보 교환 및  
(ii) 교환된 정보의 상호 이용을 목적으로 본 프로그램에 관한 정보를 얻고자 하는 라이선스 사용자는 다음 주소로 문의하십시오.

135-700

서울특별시 강남구 도곡동 467-12, 군인공제회관빌딩

한국 아이.비.엠 주식회사

고객만족센터

이러한 정보는 해당 조건(예를 들면, 사용료 지불 등)하에서 사용될 수 있습니다.

이 정보에 기술된 라이선스가 부여된 프로그램 및 프로그램에 대해 사용 가능한 모든 라이선스가 부여된 자료는 IBM이 IBM 기본 계약, IBM 프로그램 라이선스 계약(IPLA) 또는 이와 동등한 계약에 따라 제공한 것입니다.

본 문서에 포함된 모든 성과 데이터는 제어된 환경에서 산출된 것입니다. 따라서 다른 운영 환경에서 얻어진 결과는 상당히 다를 수 있습니다. 일부 성능은 개발 단계 시스템에서 측정되었을 수 있으므로 이러한 측정치가 일반적으로 사용되고 있는 시스템에서도 동일하게 나타날 것이라고는 보증할 수 없습니다. 또한 일부 성능은 추정을 통해 추측되었을 수도 있으므로 실제 결과는 다를 수 있습니다. 이 책의 사용자는 해당 데이터를 본인의 특정 환경에서 검증해야 합니다.

비IBM 제품에 관한 정보는 해당 제품의 공급업체, 공개 자료 또는 기타 범용 소스로부터 얻은 것입니다. IBM에서는 이러한 비IBM 제품을 반드시 테스트하지 않았으므로, 이들 제품과 관련된 성과의 정확성, 호환성 또는 기타 주장에 대해서는 확인할 수 없습니다. 비IBM 제품의 성능에 대한 의문사항은 해당 제품의 공급업체에 문의하십시오.

IBM이 제시하는 방향 또는 의도에 관한 모든 언급은 특별한 통지 없이 변경될 수 있습니다.

여기에 나오는 모든 IBM의 가격은 IBM이 제시하는 현 소매가이며 통지 없이 변경될 수 있습니다. 실제 판매가는 다를 수 있습니다.

이 정보에는 일상적인 비즈니스 활동에서 사용되는 자료 및 보고서에 대한 예제가 들어 있습니다. 이들 예제에는 개념을 가능한 완벽하게 설명하기 위하여 개인, 회사, 상표 및 제품의 이름이 사용될 수 있습니다. 이들 이름은 모두 가공의 것이며 실제 기업의 이름 및 주소와 유사하더라도 이는 전적으로 우연입니다.

저작권 라이선스:

이 정보에는 여러 운영 플랫폼에서의 프로그래밍 기법을 보여주는 원어로 된 샘플 애플리케이션이 들어 있습니다. 귀하는 이러한 샘플 프로그램의 작성 기준이 된 운영 플랫폼의 API(Application Programming Interface)에 부합하는 애플리케이션을 개발, 사용, 판매 또는 배포할 목적으로 추가 비용 없이 이들 샘플 프로그램을 어떠한 형태로든 복사, 수정 및 배포할 수 있습니다. 이러한 샘플 프로그램은 모든 조건하에서 완전히 테스트된 것은 아닙니다. 따라서 IBM은 이러한 프로그램의 신뢰성, 서비스 가능성 또는 기능을 보증하거나 진술하지 않습니다. 본 샘플 프로그램은 일체의 보증 없이 "현상태 대로" 제공됩니다. IBM은 귀하의 샘플 프로그램 사용과 관련되는 손해에 대해 책임을 지지 않습니다.

이 정보를 소프트카피로 확인하는 경우에는 사진과 컬러 삽화가 제대로 나타나지 않을 수도 있습니다.

---

## 상표

IBM, IBM 로고 및 ibm.com은 전세계 여러 국가에 등록된 International Business Machines Corp.의 상표 또는 등록상표입니다. 기타 제품 또는 서비스 이름은 IBM 또는 타사의 상표입니다. 현재 IBM 상표 목록은 웹의 『저작권 및 상표 정보』 ([www.ibm.com/legal/copytrade.shtml](http://www.ibm.com/legal/copytrade.shtml))에 있습니다.

