|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  |  | | |  |
|  | | |
|  | | |
|  | | | |
|  | **SDP Solution**  **API Reference (Subscribe, SOAP)** | |  |
|  | |
| **Issue** | **01** |
| **Date** | **2016-03-16** |
| HUAWEI TECHNOLOGIES CO., LTD. | |

|  |
| --- |
| **Copyright © Huawei Technologies Co., Ltd. 2016. All rights reserved.**  No part of this document may be reproduced or transmitted in any form or by any means without prior written consent of Huawei Technologies Co., Ltd.  **Trademarks and Permissions**  and other Huawei trademarks are trademarks of Huawei Technologies Co., Ltd.  All other trademarks and trade names mentioned in this document are the property of their respective holders.  **Notice**  The purchased products, services and features are stipulated by the contract made between Huawei and the customer. All or part of the products, services and features described in this document may not be within the purchase scope or the usage scope. Unless otherwise specified in the contract, all statements, information, and recommendations in this document are provided "AS IS" without warranties, guarantees or representations of any kind, either express or implied.  The information in this document is subject to change without notice. Every effort has been made in the preparation of this document to ensure accuracy of the contents, but all statements, information, and recommendations in this document do not constitute a warranty of any kind, express or implied. |

|  |  |
| --- | --- |
| Huawei Technologies Co., Ltd. | |
| Address: | Huawei Industrial Base  Bantian, Longgang  Shenzhen 518129  People's Republic of China |
| Website: | <http://www.huawei.com> |
| Email: | [support@huawei.com](mailto:support@huawei.com) |
|  |  |

Contents

[1 Overview 1](#_Toc443999962)

[1.1 API Functions 1](#_Toc443999963)

[1.2 Level of Requirement for Parameters 2](#_Toc443999964)

[1.3 Request Format 3](#_Toc443999965)

[1.4 Response Format 4](#_Toc443999966)

[1.5 Namespace 5](#_Toc443999967)

[1.6 SOAPAction 5](#_Toc443999968)

[2 API for Subscribing To a Service 6](#_Toc443999969)

[2.1 Process 6](#_Toc443999970)

[2.2 subscribeProduct 7](#_Toc443999971)

[2.2.1 Function 7](#_Toc443999972)

[2.2.2 Request URI 8](#_Toc443999973)

[2.2.3 Request 8](#_Toc443999974)

[2.2.4 Response 16](#_Toc443999975)

[2.2.5 Error Codes 17](#_Toc443999976)

[3 API for Unsubscribing From a Service 20](#_Toc443999977)

[3.1 Process 20](#_Toc443999978)

[3.2 unSubscribeProduct 21](#_Toc443999979)

[3.2.1 Function 21](#_Toc443999980)

[3.2.2 Request URI 22](#_Toc443999981)

[3.2.3 Request 22](#_Toc443999982)

[3.2.4 Response 30](#_Toc443999983)

[3.2.5 Error Codes 31](#_Toc443999984)

Figures

[**Figure 2-1** Process of subscribing to an SP service 7](#_Toc443999985)

[**Figure 3-1** Process of unsubscribing from an SP service 21](#_Toc443999986)

Tables

[**Table 1-1** Functions of Subscribe capability APIs 1](#_Toc443999987)

[**Table 1-2** Level of requirement for parameters 2](#_Toc443999988)

[**Table 1-3** Request format 3](#_Toc443999989)

[**Table 1-4** Success response format 4](#_Toc443999990)

[**Table 1-5** Error response format 5](#_Toc443999991)

[**Table 1-6** Namespaces of Subscribe capability APIs 5](#_Toc443999992)

[**Table 2-1** Parameters in a subscribeProductRequest message header 9](#_Toc443999993)

[**Table 2-2** Parameters in a subscribeProductRequest message body 12](#_Toc443999994)

[**Table 2-3** UserID parameters 12](#_Toc443999995)

[**Table 2-4** SubInfo parameters 12](#_Toc443999996)

[**Table 2-5** NamedParameter parameters 15](#_Toc443999997)

[**Table 2-6** Parameters in a subscribeProductResponse message body 16](#_Toc443999998)

[**Table 2-7** subscribeProduct error codes 17](#_Toc443999999)

[**Table 3-1** Parameters in a unSubscribeProductRequest message header 23](#_Toc444000000)

[**Table 3-2** Parameters in a UnSubscribeProductRequest message body 26](#_Toc444000001)

[**Table 3-3** UserID parameters 26](#_Toc444000002)

[**Table 3-4** SubInfo parameters 26](#_Toc444000003)

[**Table 3-5** NamedParameter parameters 29](#_Toc444000004)

[**Table 3-6** Parameters in a unSubscribeProductResponse message body 30](#_Toc444000005)

[**Table 3-7** unSubscribeProduct error codes 31](#_Toc444000006)

# Overview

About This Chapter

[1.1 API Functions](#EN-US_TOPIC_0008794457)

[1.2 Level of Requirement for Parameters](#EN-US_TOPIC_0008794130)

[1.3 Request Format](#EN-US_TOPIC_0008794133)

[1.4 Response Format](#EN-US_TOPIC_0008794470)

[1.5 Namespace](#EN-US_TOPIC_0008794469)

[1.6 SOAPAction](#EN-US_TOPIC_0008794472)

## API Functions

The Subscribe capability interfaces provided by the SDP are used to subscribe to and unsubscribed from services by an SP portal or the carrier customer service system.

[Table 1-1](#table35176150) describes functions of Subscribe capability APIs provided by the SDP.

Functions of Subscribe capability APIs

| Function | Description | API |
| --- | --- | --- |
| Subscribing to a service | The SP portal or the carrier customer service system (functioning as the client) invokes the subscribeProduct API to send service subscription messages to the SDP (functioning as the server).  The SP portal or the carrier customer service system must code based on the API field requirements so that send correct requests to the SDP. The SDP sends a response within 60 seconds by default. | [subscribeProduct](#EN-US_TOPIC_0008794459) |
| Unsubscribing from a service | The SP portal or the carrier customer service system (functioning as the client) invokes the unSubscribeProduct API to send service unsubscription messages to the SDP (functioning as the server).  The SP portal or the carrier customer service system must code based on the API field requirements so that send correct requests to the SDP. The SDP sends a response within 60 seconds by default. | [unSubscribeProduct](#EN-US_TOPIC_0008794456) |

## Level of Requirement for Parameters

The App must develop APIs based on the level of requirement for each parameter.

Level of requirement for parameters

| Type | Description |
| --- | --- |
| Mandatory | A parameter is always mandatory in a request.    Parameters with the **Mandatory** requirement are used for access authentication or service processing. If a parameter with the **Mandatory** requirement is left empty in a request, access authentication or service processing fails and the request fails. |
| Conditional | A parameter is mandatory or optional in specified conditions.    Parameters with the **Conditional** requirement are used for access authentication or service processing in specified conditions. If the specified conditions is met but a parameter with the **Conditional** requirement is left empty in a request, access authentication or service processing fails and the request fails. |
| Optional | A parameter is always optional.    Parameters with the **Optional** requirement are not used for service processing. |

## Request Format

The SDP provides the SOAP request in the following format:

<soapenv:Envelope>   
 <soapenv:Header>   
 <parameter>…</parameter>   
 …   
 </soapenv:Header>   
 <soapenv:Body>   
 <parameter>…</parameter>   
 …   
 </soapenv:Body>   
 </soapenv:Envelope>

Request format

| Element | Description |
| --- | --- |
| <soapenv:Envelope> | Root element in a request, which specifies the namespace. |
| <soapenv:Header> | Request header.  Parameters in this element are defined by the SDP and are mainly information to be processed by the SDP services, including access authentication parameters. |
| <soapenv:Body> | Request body.  Parameters in this element comply with the SOAP protocol. |



When Apps invoke SDP APIs, requests cannot contain the following XML characters: & > < ' "

If the preceding characters are really required, you must add escape characters before such characters. Otherwise, SDP APIs will fail to be invoked.

## Response Format

Success Response Format

The SDP provides the SOAP API success responses in the following format:

<soapenv:Envelope>   
 <soapenv:Body>   
 <parameter>…</parameter>   
 …   
 </soapenv:Body>   
 </soapenv:Envelope>

Success response format

| Element | Description |
| --- | --- |
| <soapenv:Envelope> | Root element in a success response, which specifies the namespace. |
| <soapenv:Body> | Success response body.  Parameters in this element comply with the SOAP protocol. |

Error Response Format

The SDP provides the SOAP API error responses in the following format:

<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">   
<soapenv:Body><ns1:subscribeProductResponse xmlns:ns1="http://www.csapi.org/schema/parlayx/subscribe/manage/v1\_0/local">   
<ns1:subscribeProductRsp>   
<result>5034402</result>   
<resultDescription>Sp password is not accepted!</resultDescription>   
</ns1:subscribeProductRsp>   
</ns1:subscribeProductResponse>   
</soapenv:Body></soapenv:Envelope>

Error response format

| Element | Description |
| --- | --- |
| <soapenv:Envelope> | Root element in an error response, which specifies the namespace. |
| <soapenv:Body> | Error response body, contains the <soapenv:Fault> and <detail> elements. This element specifies the error code and error details. |
| <result> | Error code. |
| <resultDescription> | Error details. |

## Namespace

The SP must follow the specified namespaces of data types when developing Subscribe capability APIs.

* [Table 1-6](#table65479256) describes the namespaces of Subscribe capability APIs.
* The namespace of data types used by the Subscribe capability APIs are **http://www.csapi.org/schema/parlayx/subscribe/v1\_0**.

Namespaces of Subscribe capability APIs

| Namespace | API |
| --- | --- |
| http://www.csapi.org/wsdl/parlayx/subscribe/manage/v1\_0 | * [subscribeProduct](#EN-US_TOPIC_0008794459) * [unSubscribeProduct](#EN-US_TOPIC_0008794456) |

## SOAPAction

Leave the **SOAPAction** parameter empty.

The following is an example of the **SOAPAction** parameter setting in an HTTP header:

SOAPAction: ""

# API for Subscribing To a Service

About This Chapter

[2.1 Process](#EN-US_TOPIC_0008794455)

[2.2 subscribeProduct](#EN-US_TOPIC_0008794459)

## Process

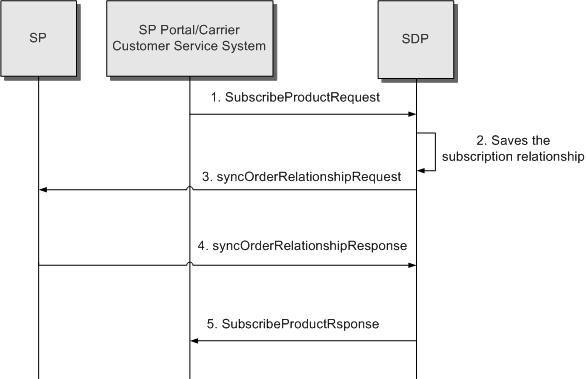
The function for a user to subscribe to a service on an SP portal or in a carrier customer service system applies to the following scenarios:

* A user logs in to an SP portal, the web or the WAP, to subscribes to a service released in the SDP. The SP portal sends a message indicating the subscription to the SDP.
* A user logs in to the carrier customer service system to subscribe to a service released on the SDP. The carrier customer service system sends a message indicating the subscription to the SDP.

When a user subscribes to a service on the SP portal or in the carrier customer service system, the SP portal or the carrier customer service system maintains the mapping between services on the SP portal or the carrier customer service system and those in the SDP and obtains product IDs of the services from the SDP.

[Figure 2-1](#fig18553230) shows the process of interface message exchange when a user subscribes to an SP service on an SP portal or in the carrier customer service system.

Process of subscribing to an SP service



The process is as follows:

1. A user subscribes to a service on an SP portal or in the carrier customer service system. The SP portal or the carrier customer service system sends a subscription request to the SDP.
2. The SDP receives the request and generates a subscription relationship.
3. The SDP invokes the SyncOrderRelation interface of the DataSync (SOAP) to synchronize the subscription relationship to the SP.
4. The SP saves the subscription relationship and sends a response to the SDP.
5. The SDP sends the response to the SP portal or the carrier customer service system.

## subscribeProduct

### Function

The SP portal or the carrier customer service system (functioning as the client) invokes the subscribeProduct API to send service subscription messages to the SDP (functioning as the server).

The SP portal or the carrier customer service system must code based on the API field requirements so that send correct requests to the SDP. The SDP sends a response within 60 seconds by default.

### Request URI

The request URI is the destination URI of subscribeProduct messages sent by an SP portal or the carrier customer service system to the SDP to subscribe to a service. The URI is provided by the SDP in the following format:

**http://*IP*:*Port*/SubscribeManageService/services/SubscribeManage**

In the format, *IP* and *Port* indicate the service IP address and SOAP port number of the API provided by the SDP. Contact carriers to obtain the IP address and port number.

### Request

The SP portal or the carrier customer service system functions as the client and sends a **subscribeProductRequest** message to the SDP to subscribe to a service.

Example

<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/" xmlns:loc="http://www.csapi.org/schema/parlayx/subscribe/manage/v1\_0/local">   
 <soapenv:Header>   
 <tns:RequestSOAPHeader xmlns:tns="http://www.huawei.com.cn/schema/common/v2\_1">   
 <tns:spId>35000001</tns:spId>   
 <tns:spPassword>c5216e519a071d601bedd150f3fcd026</tns:spPassword>   
 <tns:timeStamp>20080101010101</tns:timeStamp>   
 </tns:RequestSOAPHeader>   
 </soapenv:Header>   
 <soapenv:Body>   
 <loc:subscribeProductRequest>   
 <loc:subscribeProductReq>   
 <userID>   
 <ID>8612312312123</ID>   
 <type>0</type>   
 </userID>   
 <subInfo>   
 <productID>12345678</productID>   
 <operCode>zh</operCode>   
 <isAutoExtend>0</isAutoExtend>   
 <channelID>2</channelID>   
 <extensionInfo>   
 <namedParameters>   
 <key>keyword</key>   
 <value>sub</value>   
 </namedParameters>   
 </extensionInfo>   
 </subInfo>   
 </loc:subscribeProductReq>   
 </loc:subscribeProductRequest>   
 </soapenv:Body>   
 </soapenv:Envelope>

Message Header Parameters

[Table 2-1](#table30083052) describes parameters in a **subscribeProductRequest** message header.

Parameters in a subscribeProductRequest message header

| Parameter | Type | Length | Level of Requirement | Description |
| --- | --- | --- | --- | --- |
| spId | xsd: string | 21 | Mandatory | Partner ID.  Partner ID automatically allocated by the SDP to the SP after successful registration. For an SP, use the registered account to log in to the SDP management portal and query the account information, or obtain it from the notification email after successful registration.  [Example] 35000001 |
| spPassword | xsd: string | 100 | Conditional | Authentication key for the SDP to authenticate partners.  The SDP supports authentication by SP ID + Password, SP ID + IP address + Password, or SP ID + IP address. Partners select an authentication mode during registration. If a partner selects authentication by SP ID + Password or SP ID + IP address + Password, this parameter is mandatory in requests sent by this partner.  The value is a character string encrypted. The encryption formula is as follows:   * SHA-256: spPassword = Base64(SHA-256(spId + Password + timeStamp)) * MD5: spPassword = MD5(spId + Password + timeStamp)   In the formula:   * **spId** and **timeStamp**: authentication ID and timestamp. * **Password**: access password allocated by the SDP to a partner.   An SP can obtain the password from the email notification received after successful registration.A Developer can log in to the Developer Portal, choose Member Center > Account > Registration Information > Invoke Password, and set the password.An Enterprise must contact the carrier.  NOTE  The default value is SHA-256. To retain features of earlier versions, the SP uses the MD5 algorithm in the connection to the SDP, which might cause security risks.  [Example] C5216E519A071D601BEDD150F3FCD026  6 |
| timeStamp | xsd: string | 14 | Conditional | Time stamp (UTC time).  The value is used in encryption of **spPassword**.  This parameter is mandatory when the **spPassword** parameter is required.  [Format] *yyyyMMddHHmmss*  [Example] 20080101010101 |

Message Body Parameters

[Table 2-2](#table21941795) describes parameters in a **subscribeProductRequest** message body.

Parameters in a subscribeProductRequest message body

| Parameter | Type | Length | Level of Requirement | Description |
| --- | --- | --- | --- | --- |
| userID | UserID | - | Mandatory | User information, including user ID and user type.  For details, see [Table 2-3](#table11517585). |
| subInfo | SubInfo | - | Mandatory | Service information.  For the **SubInfo** information, see [Table 2-4](#table29491149). |

UserID parameters

| Parameter | Type | Length | Level of Requirement | Description |
| --- | --- | --- | --- | --- |
| ID | String | 30 | Mandatory | User ID.  [Example] 8613012312123 |
| type | int | 2 | Mandatory | User type.   * 0: MSISDN * 10: user fake ID * Other values: reserved.   [Example] 0 |

SubInfo parameters

| Parameter | Type | Length | Level of Requirement | Description |
| --- | --- | --- | --- | --- |
| productID | String | 21 | Mandatory | Product ID.  The ID is automatically allocated by the SDP when an SP or the carrier releases a subscribable product. To obtain the product ID, log in to the SDP management portal and view the value of **Product ID** in the pricing information on the service details page.  [Example] 1000000201 |
| operCode | String | 40 | Optional | Operation code.  This field can be set to a user's language or area information. The SDP saves this field and synchronizes it to the SP using the interface for synchronizing subscription relationships.  [Example] zh |
| isAutoExtend | INT2 | 2 | Optional | Indicates whether to automatically renew the subscription.   * 0: No * 1: Yes   NOTE  The SDP does not contain this field. Whether to automatically renew a subscription relationship is determined when an SP releases a service on the SDP management portal.  [Example] 0 |
| channelID | INT4 | 4 | Mandatory | Access mode, namely, original channel for a user to initiate a service request.   * 1: WEB * 2: SMS * 3: USSD * 4: IVR * 5: CC (CRM) * 6: PDA * 7: XHTML (WAP 2) (This value is used in the SDP baseline version.) * 21: ODP * 99: Other * 100: SP website * 101: UE(User Equipment) * 105: PCClient * 116: WAP AOC * 117: USSD VIP * 120: channel of unsubscription caused by user deregistration * 124: channel of unsubscription caused by the blacklist * 125: SP website (added channel type in the MTN project, which is used for subscription on the SP website.)   [Example] 1 |
| extensionInfo | NamedParameter | - | Optional | Extended parameter.  This parameter is contained but meaningless in the request. Therefore, the SP can ignore this parameter.  For the **NamedParameter** information, see [Table 2-5](#table18184283). |

NamedParameter parameters

| Parameter | Type | Length | Level of Requirement | Description |
| --- | --- | --- | --- | --- |
| key | String | - | Mandatory | ID of the extended field.  The value is specified by the SDP.  [Example] keyword |
| value | String | 512 | Mandatory | Subscribable command words.  [Example] sub |

### Response

The SDP functions as the server, processes subscribeProductRequest messages received from the SP, and sends subscribeProductResponse messages to the SP.

This topic provides a success response example and describes parameters in the response. If a request fails, the SDP sends an error response that contains an error code.

Example

<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">   
 <soapenv:Body>   
 <ns1:subscribeProductResponse xmlns:ns1="http://www.csapi.org/schema/parlayx/subscribe/manage/v1\_0/local">   
 <ns1:subscribeProductRsp>   
 <result>0</result>   
 <resultDescription>successful</resultDescription>   
 </ns1:subscribeProductRsp>   
 </ns1:subscribeProductResponse>   
 </soapenv:Body>   
 </soapenv:Envelope>

Message Body Parameters

[Table 2-6](#table23176056) describes parameters in a subscribeProductResponse message body.

Parameters in a subscribeProductResponse message body

| Parameter | Type | Level of Requirement | Description |
| --- | --- | --- | --- |
| result | String | Mandatory | * 0 or 00000000: The request succeeds. * Other values: The request is abnormal. For the error codes, see [2.2.5 Error Codes](#EN-US_TOPIC_0008794131). |
| resultDescription | String | Optional | Subscription result. |

### Error Codes

[Table 2-7](#table46642793) describes subscribeProduct error codes that the SDP may return upon an exception. For details about the error codes, see the *SDP Solution Error Code Reference*.

subscribeProduct error codes

| Error Code | Description |
| --- | --- |
| 10001211 | The field in the request is incorrect. |
| 10001318 | An internal error occurs in the SDP: message fails to be sent between internal components. |
| 10002035 | An internal error occurs in the SDP: internal component process times out. |
| 10000000-10009999 | An internal error occurs in the SDP. |
| 22007006 | The version number is incorrect. A version number must contain 1 to 5 bytes. |
| 22007007 | The message ID is incorrect. A message ID must contain 1 to 40 bytes. |
| 22007008 | The user does not exist. |
| 22007013 | The fee deduction parameter is incorrect. |
| 22007014 | The user ID is invalid. |
| 22007034 | The user ID is invalid. |
| 22007038 | The user has no permission to subscribe to this product because the product is a gift product. |
| 22007044 | Notification of subscription relationships fails. |
| 22007053 | Configuration of this user is not found in the SDP system. |
| 22007071 | Subnet information cannot be found. |
| 22007201 | The product has been subscribed to. |
| 22007203 | The product does not exist. |
| 22007206 | The product is not in the validity period. The product may have expired or have not taken effect. |
| 22007208 | On-demand products cannot be subscribed to. |
| 22007210 | The user status is abnormal, that is, it is not defined in the system. |
| 22007211 | The user has been deregistered. |
| 22007220 | The notification fails to be sent to the SP. |
| 22007225 | The user's bonus points is insufficient for the subscription. |
| 22007227 | The product does not support bonus point payment. |
| 22007230 | The product cannot be subscribed to by a third party. |
| 22007233 | Subscription is being confirmed. |
| 22007238 | The subscription relationship already exists. The product cannot be subscribed to again. |
| 22007268 | The user's credibility is insufficient and the user cannot subscribe to the product. |
| 22007301 | The user is suspended. |
| 22007303 | The user is paused. |
| 22007304 | The user's subscription capability is suspended and the user cannot subscribe to the product. |
| 22007306 | The user is blacklisted and cannot subscribe to the product. |
| 22007322 | The promotional product fails to be subscribed to in the non-promotion period. |
| 22007330 | The account balance is insufficient. |
| 22007331 | The charged number is incorrect. |
| 22007332 | The external system charging times out. |
| 22007333 | Other errors occur during charging in the SDP. |
| 22007334 | The user information response from an external system times out. |
| 22007363 | The user is graylisted and cannot subscribe to the product. |
| 22007505 | The channel is invalid. The value range of **ChannelID** in the request is invalid. |
| 22007629 | The user is locked. |
| 22007863 | Subscription request has been processed successfully and waiting for user confirm |
| 22007999 | Failure due to other causes. |
| 22999998 | The service returned by the external charging system is unavailable. |
| 22999999 | External system error. |

# API for Unsubscribing From a Service

About This Chapter

[3.1 Process](#EN-US_TOPIC_0008794126)

[3.2 unSubscribeProduct](#EN-US_TOPIC_0008794456)

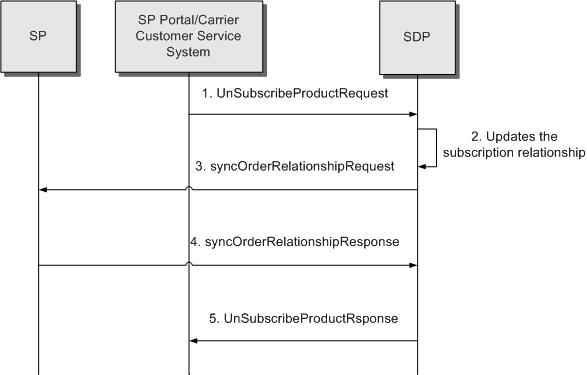
## Process

The function for a user to unsubscribe from a service on an SP portal or in a carrier customer service system applies to the following scenarios:

* A user logs in to an SP portal, the web or the WAP, to unsubscribes from a service released in the SDP. The SP portal sends a message indicating the unsubscription to the SDP.
* A user logs in to the carrier customer service system to unsubscribe from a service released on the SDP. The carrier customer service system sends a message indicating the unsubscription to the SDP.

[Figure 3-1](#fig22350181) shows the process of interface message exchange when a user unsubscribes from an SP service on an SP portal or in the carrier customer service system

Process of unsubscribing from an SP service



The process is as follows:

1. A user unsubscribes from a service on an SP portal or in the carrier customer service system. The SP portal or the carrier customer service system sends an unsubscription request to the SDP.
2. The SDP receives the request and updates the subscription relationship.
3. The SDP invokes the SyncOrderRelation interface of the DataSync (SOAP) to request the SP to update the subscription relationship.
4. The SP updates the subscription relationship and sends a response to the SDP.
5. The SDP sends the response to the SP portal or the carrier customer service system.

## unSubscribeProduct

### Function

The SP portal or the carrier customer service system (functioning as the client) invokes the unSubscribeProduct API to send service unsubscription messages to the SDP (functioning as the server).

The SP portal or the carrier customer service system must code based on the API field requirements so that send correct requests to the SDP. The SDP sends a response within 60 seconds by default.

### Request URI

The request URI is the destination URI of unSubscribeProduct messages sent by an SP portal or the carrier customer service system to the SDP to unsubscribe from a service. The URI is provided by the SDP in the following format:

**http://*IP*:*Port*/SubscribeManageService/services/SubscribeManage**

In the format, *IP* and *Port* indicate the service IP address and SOAP port number of the API provided by the SDP. Contact carriers to obtain the IP address and port number.

### Request

The SP portal or the carrier customer service system functions as the client and sends a **unSubscribeProductRequest** message to the SDP to unsubscribe from service.

Example

<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/" xmlns:loc="http://www.csapi.org/schema/parlayx/subscribe/manage/v1\_0/local">   
 <soapenv:Header>   
 <tns:RequestSOAPHeader xmlns:tns="http://www.huawei.com.cn/schema/common/v2\_1">   
 <tns:spId>35000001</tns:spId>   
 <tns:spPassword>c5216e519a071d601bedd150f3fcd026</tns:spPassword>   
 <tns:timeStamp>20080101010101</tns:timeStamp>   
 </tns:RequestSOAPHeader>   
 </soapenv:Header>   
 <soapenv:Body>   
 <loc:unSubscribeProductRequest>   
 <loc:unSubscribeProductReq>   
 <userID>   
 <ID>8612312312123</ID>   
 <type>0</type>   
 </userID>   
 <subInfo>   
 <productID>1000000201</productID>   
 <operCode>zh</operCode>   
 <isAutoExtend>0</isAutoExtend>   
 <channelID>1</channelID>   
 <extensionInfo>   
 <namedParameters>   
 <key>keyword</key>   
 <value>unsub</value>   
 </namedParameters>   
 </extensionInfo>   
 </subInfo>   
 </loc:unSubscribeProductReq>   
 </loc:unSubscribeProductRequest>   
 </soapenv:Body>   
 </soapenv:Envelope>

Message Header Parameters

[Table 3-1](#table13166761) describes parameters in a **unSubscribeProductRequest** message header.

Parameters in a unSubscribeProductRequest message header

| Parameter | Type | Length | Level of Requirement | Description |
| --- | --- | --- | --- | --- |
| spId | xsd: string | 21 | Mandatory | Partner ID.  Partner ID automatically allocated by the SDP to the SP after successful registration. For an SP, use the registered account to log in to the SDP management portal and query the account information, or obtain it from the notification email after successful registration.  [Example] 35000001 |
| spPassword | xsd: string | 100 | Conditional | Authentication key for the SDP to authenticate partners.  The SDP supports authentication by SP ID + Password, SP ID + IP address + Password, or SP ID + IP address. Partners select an authentication mode during registration. If a partner selects authentication by SP ID + Password or SP ID + IP address + Password, this parameter is mandatory in requests sent by this partner.  The value is a character string encrypted. The encryption formula is as follows:   * SHA-256: spPassword = Base64(SHA-256(spId + Password + timeStamp)) * MD5: spPassword = MD5(spId + Password + timeStamp)   In the formula:   * **spId** and **timeStamp**: authentication ID and timestamp. * **Password**: access password allocated by the SDP to a partner.   An SP can obtain the password from the email notification received after successful registration.A Developer can log in to the Developer Portal, choose Member Center > Account > Registration Information > Invoke Password, and set the password.An Enterprise must contact the carrier.  NOTE  The default value is SHA-256. To retain features of earlier versions, the SP uses the MD5 algorithm in the connection to the SDP, which might cause security risks.  [Example] C5216E519A071D601BEDD150F3FCD026  6 |
| timeStamp | xsd: string | 14 | Conditional | Time stamp (UTC time).  The value is used in encryption of **spPassword**.  This parameter is mandatory when the **spPassword** parameter is required.  [Format] *yyyyMMddHHmmss*  [Example] 20080101010101 |

Message Body Parameters

[Table 3-2](#table16832661) describes parameters in a **unSubscribeProductRequest** message body.

Parameters in a UnSubscribeProductRequest message body

| Parameter | Type | Level of Requirement | Description |
| --- | --- | --- | --- |
| userID | UserID | Mandatory | User information, including user ID and user type.  For details, see [Table 3-3](#table34013637). |
| subInfo | SubInfo | Mandatory | Service information.  For the **SubInfo** information, see [Table 3-4](#table21216187). |

UserID parameters

| Parameter | Type | Length | Level of Requirement | Description |
| --- | --- | --- | --- | --- |
| ID | String | 30 | Mandatory | User ID.  [Example] 8613012312123 |
| type | int | 2 | Mandatory | User type.   * 0: MSISDN * 10: user fake ID * Other values: reserved.   [Example] 0 |

SubInfo parameters

| Parameter | Type | Length | Level of Requirement | Description |
| --- | --- | --- | --- | --- |
| productID | String | 21 | Mandatory | Product ID.  The ID is automatically allocated by the SDP when an SP or the carrier releases a subscribable product. To obtain the product ID, log in to the SDP management portal and view the value of **Product ID** in the pricing information on the service details page.  [Example] 1000000201 |
| operCode | String | 40 | Optional | Operation code.  This field can be set to a user's language or area information. The SDP saves this field and synchronizes it to the SP using the interface for synchronizing subscription relationships.  This field is optional. The SDP does not need to obtain the value of this field from the message.  [Example] zh |
| isAutoExtend | INT2 | 2 | Optional | Indicates whether to automatically renew the subscription.   * 0: No * 1: Yes   NOTE  The SDP does not contain this field. Whether to automatically renew a subscription relationship is determined when an SP releases a service on the SDP management portal.  [Example] 0 |
| channelID | INT4 | 4 | Mandatory | Access mode, namely, original channel for a user to initiate a service request.   * 1: WEB * 2: SMS * 3: USSD * 4: IVR * 5: CC (CRM) * 6: PDA * 7: XHTML (WAP 2) (This value is used in the SDP baseline version.) * 21: ODP * 99: Other * 100: SP website * 101: UE(User Equipment) * 105: PCClient * 116: WAP AOC * 117: USSD VIP * 120: channel of unsubscription caused by user deregistration * 124: channel of unsubscription caused by the blacklist * 125: SP website (added channel type in the MTN project, which is used for subscription on the SP website.)   [Example] 1 |
| extensionInfo | NamedParameter | - | Optional | Extended parameter.  This parameter is contained but meaningless in the request. Therefore, the SP can ignore this parameter.  For the **NamedParameter** information, see [Table 3-5](#table24908498). |

NamedParameter parameters

| Parameter | Type | Length | Level of Requirement | Description |
| --- | --- | --- | --- | --- |
| key | String | - | Mandatory | ID of the extended field.  The value is specified by the SDP.  [Example] keyword |
| value | String | 512 | Mandatory | Unsubscribable command words.  [Example] unsub |

### Response

The SDP functions as the server, processes unSubscribeProductRequest messages received from the SP, and sends unSubscribeProductResponse messages to the SP.

This topic provides a success response example and describes parameters in the response. If a request fails, the SDP sends an error response that contains an error code.

Example

<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">   
 <soapenv:Body>   
 <ns1:unSubscribeProductResponse xmlns:ns1="http://www.csapi.org/schema/parlayx/subscribe/manage/v1\_0/local">   
 <ns1:unSubscribeProductRsp>   
 <result>0</result>   
 <resultDescription>successful</resultDescription>   
 </ns1:unSubscribeProductRsp>   
 </ns1:unSubscribeProductResponse>   
 </soapenv:Body>   
 </soapenv:Envelope>

Message Body Parameters

[Table 3-6](#table48410523) describes parameters in a unSubscribeProductResponse message body.

Parameters in a unSubscribeProductResponse message body

| Parameter | Type | Level of Requirement | Description |
| --- | --- | --- | --- |
| result | String | Mandatory | * 0 or 00000000: The request succeeds. * Other values: The request is abnormal. For the error codes, see [3.2.5 Error Codes](#EN-US_TOPIC_0008794462). |
| resultDescription | String | Optional | Subscription result. |

### Error Codes

[Table 3-7](#table3324459) describes unSubscribeProduct error codes that the SDP may return upon an exception. For details about the error codes, see the *SDP Solution Error Code Reference*.

unSubscribeProduct error codes

| Error Code | Description |
| --- | --- |
| 10001211 | The field in the request is incorrect. |
| 10001318 | An internal error occurs in the SDP: message fails to be sent between internal components. |
| 10002035 | An internal error occurs in the SDP: internal component process times out. |
| 10000000-10009999 | An internal error occurs in the SDP. |
| 22007006 | The version number is incorrect. A version number must contain 1 to 5 bytes. |
| 22007007 | The message ID is incorrect. A message ID must contain 1 to 40 bytes. |
| 22007008 | The user does not exist. |
| 22007013 | The fee deduction parameter is incorrect. |
| 22007014 | The user ID is invalid. |
| 22007034 | The user ID is invalid. |
| 22007038 | The user has no permission to subscribe to this product because the product is a gift product. |
| 22007044 | Notification of subscription relationships fails. |
| 22007053 | Configuration of this user is not found in the SDP system. |
| 22007071 | Subnet information cannot be found. |
| 22007201 | The product has been subscribed to. |
| 22007203 | The product does not exist. |
| 22007206 | The product is not in the validity period. The product may have expired or have not taken effect. |
| 22007208 | On-demand products cannot be subscribed to. |
| 22007210 | The user status is abnormal, that is, it is not defined in the system. |
| 22007211 | The user has been deregistered. |
| 22007220 | The notification fails to be sent to the SP. |
| 22007225 | The user's bonus points is insufficient for the subscription. |
| 22007227 | The product does not support bonus point payment. |
| 22007230 | The product cannot be subscribed to by a third party. |
| 22007233 | Subscription is being confirmed. |
| 22007238 | The subscription relationship already exists. The product cannot be subscribed to again. |
| 22007268 | The user's credibility is insufficient and the user cannot subscribe to the product. |
| 22007301 | The user is suspended. |
| 22007303 | The user is paused. |
| 22007304 | The user's subscription capability is suspended and the user cannot subscribe to the product. |
| 22007306 | The user is blacklisted and cannot subscribe to the product. |
| 22007322 | The promotional product fails to be subscribed to in the non-promotion period. |
| 22007330 | The account balance is insufficient. |
| 22007331 | The charged number is incorrect. |
| 22007332 | The external system charging times out. |
| 22007333 | Other errors occur during charging in the SDP. |
| 22007334 | The user information response from an external system times out. |
| 22007363 | The user is graylisted and cannot subscribe to the product. |
| 22007505 | The channel is invalid. The value range of **ChannelID** in the request is invalid. |
| 22007629 | The user is locked. |
| 22007999 | Failure due to other causes. |
| 22999998 | The service returned by the external charging system is unavailable. |
| 22999999 | External system error. |