Web Service API

Table of Contents

The Roku Web Service API allows subscribers who signed up on Roku to use your service on other platforms. For instance, this API can be used to determine whether a subscription on Roku is still valid so the subscriber can be granted access to a platform other than Roku. Among other features, this REST API provides functionality for transaction events such as verifying purchases, canceling subscriptions, issuing credits or refunds, and updating billing cycles.

The Web Service API supports Roku billing services. After setting up your channel's in-channel products as described in Roku Billing and In-Channel Purchasing, you can use the Web Service API at runtime to verify entitlement and manage customer transactions.

Setting Up Roku Web Services

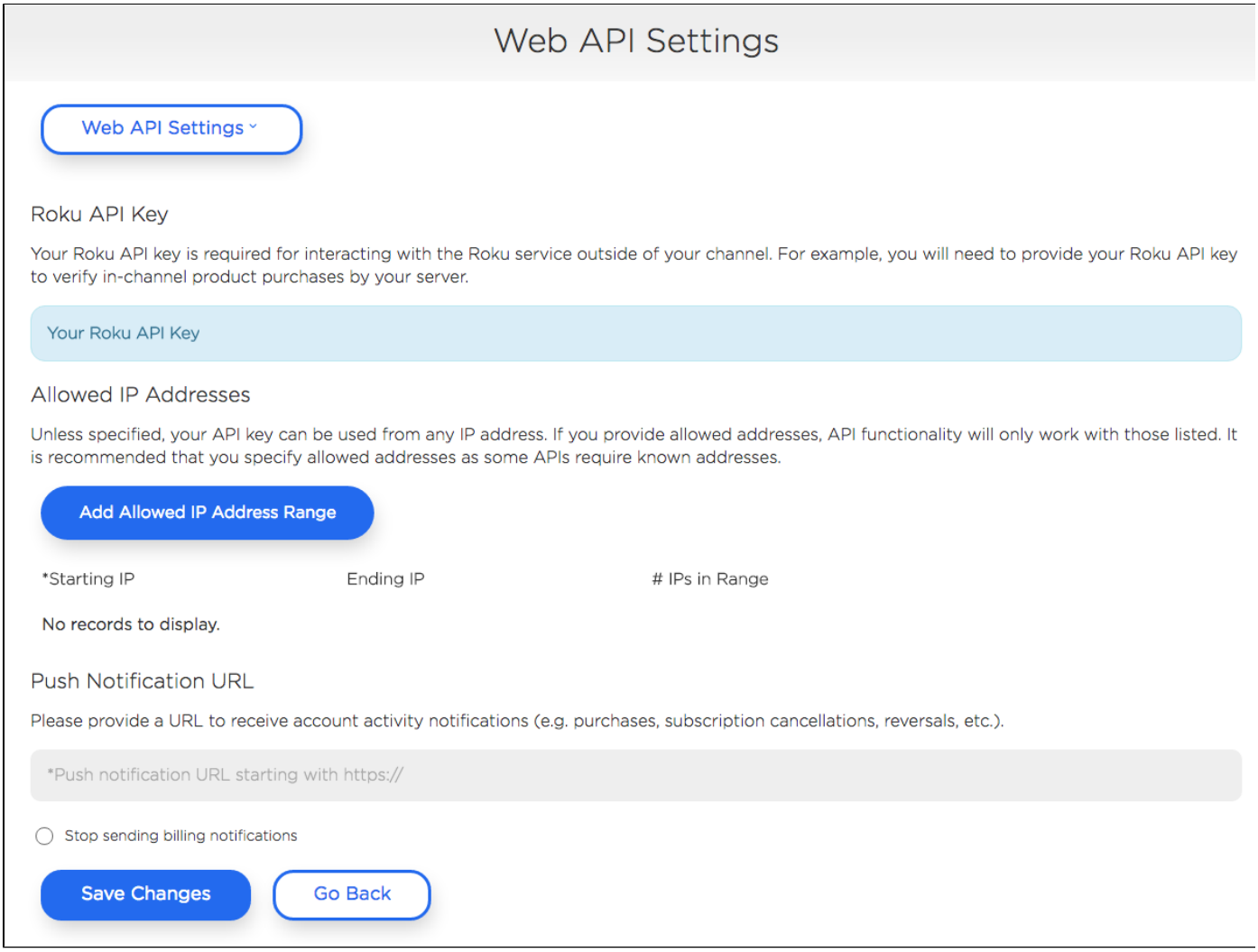
Before using the Web Service API, use the Developer Dashboard to find your partner API key. On the Developer Dashboard Web API Settings screen, Roku assigns your partner API key as the "Roku API Key." You cannot change your partner API key. Keep it safe but accessible because you must use it to perform transactions.

On the Web API Settings screen you can set a range of IP addresses that transaction service requests can come from. Doing so prevents requests from other IP addresses that don't use your partner API key. A request is valid if Roku receives it from the range of IP addresses specified, or if no range is specified, a request is valid from any IP address using your partner API key.

Also on the Web Settings API screen, you specify the URL endpoint at which you will receive push notifications for transaction events such as purchases, cancellations, refunds, and credits.

Note: If the endpoint to receive push notifications fails for a specific message for three consecutive days, Roku stops sending that notification.

Note: Once an endpoint has failed for 100 messages within 10 days, the endpoint is blacklisted. Roku stops sending all notifications and checks the box `Stop sending billing notifications` on the Web API Settings page. When the endpoint is valid again and can receive push notifications, you can uncheck `Stop sending billing notifications`.



> *See Developer Dashboard for an overview of available features.*

API Calls

The Web Service API endpoint is https://apipub.roku.com.

You can use either XML or JSON format for your requests.

- To use JSON, include the following header in the request: `accept: application/json`
- To use XML, include the following header in the request: `accept: text/xml`

Validate Transaction

This process validates a transaction that was returned from a channel or product purchase in the Channel Store. It also helps in optimizing the process of dunning.

Dunning is the process of methodically communicating with the users their subscription expiration status and re-attempting a charge to their MOPs to ensure collection of payment.

`isEntitled` is a parameter that is implemented to determine a user's subscription eligibility and supplement validating a Channel Store transaction.

To use the parameter, the entitlement service that the channel has should run a nightly sync with the Roku Channel Store channel in which it iterates all expired subscriptions and calls `validate-transaction`, that passes the stored transaction ID for the subscription. The channel then needs to read `isEntitled` in their entitlement decision making (See Response example below).

It is highly recommended for channels with in-channel products to use `isEntitled` as it ensures that the users don't find themselves in a situation where they are paying for a subscription to a service that is not authenticating them. Moreover, it can possibly increase customer retention for the channel.

When `isEntitled` is used by the channel, both the `expirationDate` and `isEntitled` should be compared to determine the accurate state of a user's subscription status. In general, use the `isEntitled` field to determin whether to entitle a user. The `cancelled` and `expirationDate` fi elds are used to determine when to check again.

> If you are checking at a particular frequency (such as every day or every few days) and are comfortable with continuing to provide entitlement to the user until your next check, you may skip looking at the `cancelled` and `expirationDate` fields. Just use the `isEnti tled` flag and entitle the user until the next check.

Following are the actions to be taken based on the current subscription state of the user:

Subscription State	isEntitled	cancelled	expirationDate	Expected Action
Active subscription	True	False	Future date	Entitle the user. Check again a day after the `expirationDate`
Actively cancelled subscription with expiration date in the future	True	True	Future date	Entitle the user, check again a day after the `expirationDate` before removing entitlement on your side. You should see `isEntitled` set to False, in which case you can remove entitlement. `isEntitled` may return True if the user changed their mind and resubscribed before the expiration date, in which case they should be treated as having an active subscription.
Actively cancelled subscription on expiry date. (Before subscription expiration timestamp)	True	True	Current date	Entitle user. Check again the next day.
Actively cancelled subscription on expiry date. (After subscription expiration timestamp)	False	True	Current date	Remove entitlement. Cancel subscription.
Actively cancelled subscription after expiry date	False	True	Past date	Remove entitlement. Cancel subscription.
Subscription in dunning	True	False	Current or Past date	Entitle user. Check again next day.
Subscription after unsuccessful dunning. (Passively canceled subscription)	False	True	Past date	Remove entitlement. Cancel subscription.

Note: Your `partnerAPIKey` can be found on the Web API Settings screen in the field Your Roku API Key.

Heading:

```
https://apipub.roku.com/listen/transaction-service.svc/validate-transaction/{partnerAPIKey}/{transactionid}
```

Response example:

```
<result>
  <transactionId>{transactionid}</transactionId>
  <purchaseDate>2012-07-22T14:59:50</purchaseDate>
  <channelName>123Video</channelName>
  <productName>123Video Monthly Subscription</productName>
  <productId>NETMONTH</productId>
  <amount>9.99</amount>
  <currency>USD</currency>
  <isEntitled>true</isEntitled>
  <quantity>1</quantity>
  <rokuCustomerId>abcdefghijklmnop</rokuCustomerId>
  <expirationDate>2012-08-22T14:59:50</expirationDate>
  <originalPurchaseDate>2010-08-22T14:59:50</originalPurchaseDate>
  <status>Success</status>
  <errorMessage>error_message</errorMessage>
</result>
```

Validate Refund

Validates a transaction that was returned from a channel or product purchase in the Channel Store.

Note: Your `partnerAPIKey` can be found on the Web API Settings screen in the field Your Roku API Key.

Heading:

```
https://apipub.roku.com/listen/transaction-service.svc/validate-refund/{partnerAPIKey}/{refundId}
```

Response example:

```
<result>
  <transactionId>{transactionid}</transactionId>
  <purchaseDate>2012-07-22T14:59:50</purchaseDate>
  <channelName>123Video</channelName>
  <productName>123Video Monthly Subscription</productName>
  <productId>NETMONTH</productId>
  <amount>9.99</amount>
  <currency>USD</currency>
  <quantity>1</quantity>
  <expirationDate>2012-08-22T14:59:50</expirationDate>
  <originalPurchaseDate>2010-08-22T14:59:50</originalPurchaseDate>
  <status>Success</status>
  <errorMessage>error_message</errorMessage>
</result>
```

Cancel Subscription

Cancels the subscription that corresponds to the `transactionId`. The channel associated with the `transactionId` must be owned by the developer associated with partnerAPIKey.

Note: Your `partnerAPIKey` can be found on the Web API Settings screen in the field Your Roku API Key.

Heading:

```
https://apipub.roku.com/listen/transaction-service.svc/cancel-subscription
```

Request example:

```
POST https://apipub.roku.com/listen/transaction-service.svc/cancel-subscription HTTP/1.1
content-type: text/xml
accept: text/xml
<cancel>
  <partnerAPIKey>{apikey}</partnerAPIKey>
  <transactionId>{transactionId}</transactionId>
  <cancellationDate>2012-08-22T14:59:50</cancellationDate>
  <partnerReferenceId>{partnerReferenceId}</partnerReferenceId>
</cancel>
```

Response example:

```
<result>
  <status>Success</status>
  <errorMessage>error_message</errorMessage>
</result>
```

Refund Subscription

Refunds the subscription that corresponds to the `transactionId`. The channel associated with `transactionId` must be owned by the developer associated with partnerAPIKey. A `refundId` is returned to validate if needed at a later date.

Note: Your `partnerAPIKey` can be found on the Web API Settings screen in the field Your Roku API Key.

Heading:

```
https://apipub.roku.com/listen/transaction-service.svc/refund-subscription
```

Request example:

```
POST https://apipub.roku.com/listen/transaction-service.svc/refund-subscription HTTP/1.1
content-type: text/xml
accept: text/xml
<cancel>
  <partnerAPIKey>{apikey}</partnerAPIKey>
  <transactionId>{transactionId}</transactionId>
  <amount>2.99</amount>
  <partnerReferenceId>{partnerReferenceId}</partnerReferenceId>
  <comments>Customer was not impressed</comments>
</cancel>
```

Response example:

```
<result>
  <status>Success</status>
  <errorMessage>{error message goes here}</errorMessage>
  <refundId>{refundId}</refundId>
</result>
```

Update Billing Cycle

Updates the billing cycle of a subscription with the provided `transactionId`. The subscription must be owned by the developer that owns the included `partnerAPIKey`. The `newBillCycleDate` must be included in an ISO-8601 format.

Note: Your `partnerAPIKey` can be found on the Web API Settings screen in the field Your Roku API Key.

Heading:

```
https://apipub.roku.com/listen/transaction-service.svc/update-bill-cycle
```

Request example:

```
POST https://apipub.roku.com/listen/transaction-service.svc/update-bill-cycle HTTP/1.1
content-type: application/json
accept: application/json
{
  "partnerAPIKey":"02E763BE3642C0459CF79F5E011CB1CE5642",
  "newBillCycleDate":" 2014-03-28",
  "transactionId":"1C63E500EF094DB4A83CA2CF00B7EB4E"
}
```

Response example:

```
{
  "errorCode":null,
  "errorDetails":null,
  "errorMessage":"",
  "status":0
}
```

Issue Service Credit

Used to issue a service credit to a particular `rokuCustomerID`. The channel/product must be owned by the developer that owns the included `partnerAPIKey`.

Note: Your `partnerAPIKey` can be found on the Web API Settings screen in the field Your Roku API Key.

If the service credit is for a channel, there must be an included `channelID`. For an in-app product, there must be both a `channelID` and `productID`. The response will include a `partnerReferenceId` that can be used later to find the service credit in the Roku system.

Heading:

```
https://apipub.roku.com/listen/transaction-service.svc/issue-service-credit
```

Request example:

```
POST https://apipub.roku.com/listen/transaction-service.svc/issue-service-credit HTTP/1.1
accept: application/json
content-type: application/json
{
  "partnerAPIKey":"02E763BE3642C0459CF79F5E011CB1CE5642",
  "amount":5.00,
  "channelId":"10081",
  "comments":"This is sample json",
  "partnerReferenceId":" ",
  "productId":" ",
  "rokuCustomerId":"AC4D2FD61F624451A61AQ2CF00A766A1"
}
```

Response example:

```
{
  "errorCode":null,
  "errorDetails":null,
  "errorMessage":"",
  "status":0,
  "ReferenceId":"1409"
}
```

Push Notifications

Push notifications are sent from Roku to your listener web server. The push notification contains a response key. You then send back a notification response containing the response key.

You can receive four types of push notifications:

- Sale (purchase)
- Cancellation
- Refund
- Credit

A sample response for each type of push notification is given below.

Security Example

To ensure security, Roku sends a `responseKey` as shown in the push example below. You must return this `responseKey` in the response content. Neither of these have a cryptographic signature.

Before downloading the content, Roku checks that the size of your `responseKey` matches the size of the `responseKey` Roku sent to you. This prevents hackers from responding with overwhelmingly large chunks of data attempting to crash the Roku Web Service.

Additionally, you are required to send an `ApiKey` header with the value containing your partner API key that was issued on the Developer Dashboard.

Note: Your `partnerAPIKey` can be found on the Web API Settings screen in the field Your Roku API Key.

The `ApiKey` header must have a content length of 32. When sending the notification, the subscriber cannot redirect in any way. If a redirect attempt is made, the request will fail. The request times out after 10 seconds.

Push Example

```
{
"customerId": "abcdd8c616cc4802989da2d800cfabcd",
"transactionType": "Credit",
"transactionId": "1243",
"channelId": "484",
"channelName": "Ssaldsdld",
"productCode": "abcdabcdd68d4ff3abcd",
"productName": "Monthly Sub",
"price": 5.00,
"total": 5.00,
"currency": "usd",
"partnerReferenceId": "prefid",
"comments": "Service Credit Test",
"eventDate": "2017-11-20T20:34:21.1781901Z",
"responseKey":"abcdabcd6b1649f681a408f1beebabcd"
}
https://pushNotificationEndpoint
```

Response Example

```
HTTP/1.1 200 OK
ApiKey: {partnerAPIKey}
Content-Length: 32
abcdabcd6b1649f681a408f1beebabcd
```

Example of Each Push Notification Response

Sale (purchase) Response Example

```
{
  "customerId":"ac4d2fd61f624451a61aa2cf00a766a1",
  "transactionType":"Sale",
  "transactionId":"aa3f3a2479ea4e0c88d9a2d500f33e74",
  "channelId":"26558",
  "productCode":"testProd123",
  "price":0.99,
  "tax":0.00,
  "total":0.99,
  "currency":"usd",
  "comments":"New order processed.",
  "eventDate":"2014-02-17T22:45:37.496125Z",
  "responseKey":"659a9e3f6b1649f681a408f1beeb2766"
}
```

Cancellation Response Example

```
{
  "customerId": "6a4d984e7aee47d18975a2d800cb707b",
  "transactionType": "Cancellation",
  "transactionId": "260a7b6d8e4a4e7b8b0ba2d800cb7142",
  "channelId": "445",
  "productCode": "fb435917cefc4f66b36c",
  "productName":"Monthly Sub",
  "expirationDate": "2014-02-20T20:20:42.6473452Z",
  "originalTransactionId": "a82e4abdab0247fb9a2ca2d800cb712d",
  "originalPurchaseDate": "2014-02-20T20:20:42",
  "partnerReferenceId": "CANCELTESTNOTIFICATION",
  "comments":"Cancellation for Monthly Sub",
  "eventDate": "2014-02-20T20:20:42.6903495Z",
  "responseKey":"c1a73677590948e68215586dfa9455b5"
}
```

Refund Response Example

```
{
  "customerId":"ac4d2fd61f624451a61aa2cf00a766a1",
  "transactionType":"Refund",
  "transactionId":"970625d44a544b78964ba2d6011231bd",
  "channelId":"26558",
  "productCode":"testProd123",
  "price":-0.99,
  "tax":0.00,
  "total":-0.99,
  "currency":"usd",
  "originalTransactionId":"dccc31583aa1441e8c76a2d600b28716",
  "originalPurchaseDate":"2014-02-18T18:49:59",
  "partnerReferenceId":"test",
  "comments":"test",
  "eventDate":"2014-02-19T00:38:20.231375Z",
  "responseKey":"c1a73677590948e68215586dfa9455b5"
}
```


Credit Response Example


```
{
  "customerId": "bbd2d8c616cc4802989da2d800cf2b81",
  "transactionType": "Credit",
  "transactionId": "1243",
  "channelId": "484",
  "channelName": "StwzAbNxMbX779Ia",
  "productCode": "f478e3a9d68d4ff390bb",
  "productName": "Monthly Sub",
  "price": 5.00,
  "total": 5.00,
  "currency": "usd",
  "partnerReferenceId": "prefid",
  "comments": "Service Credit Test",
  "eventDate": "2014-02-20T20:34:21.1781901Z",
  "responseKey":"659a9e3f6b1649f681a408f1beeb2766"
}
```