

**Subscription Management API**

Robert J. Muller

June 2, 2015

643 Bair Island Road Suite 403  
Redwood City, CA 94063

The API calls are URIs that identify resources or actions. You can use the GET, POST, PUT, and DELETE HTTP operations to query, create, update, or delete the resources, respectively. The API calls have comments (#) that describe the resource or action. The substitution value <id> is a unique integer identifier for an object, used throughout the documentation. You submit new and updated data values in the body of the POST or PUT operations.

# Partners

A *partner* is a system that comprises a unit for subscription, authentication, and authorization. The user accesses the partner system through a well-defined set of uniform resource identifiers (URIs) identified by a set of regular-expression patterns. Partners have individual subscription terms for display on subscription pages. Terms have period, price, and group discount percentage. Partners have regular expression patterns that identify the complete set of URIs for the partner, or alternatively identify the partner from a URI. Partners have subscriptions by parties. Parties have IP counts and limit values. Partners also own access rules (combinations of URI patterns and access types.

* **/partners** # all partners
* **/partners?partnerId=<id>** # a specific partner
* **/partners/patterns** # all partner URI patterns
* **/partners/patterns?partnerId=<id>** # all patterns for a partner
* **/partners/patterns?patternId=<id>** # a specific partner URI pattern
* **/partners/terms** # all partner subscription terms
* **/partners/terms?partnerId=<id>** # all partner subscription terms for a partner
* **/partners/terms?termId=<id>** # a specific partner subscription term

# Parties

A *party* is an entity that is some kind of system participant. A party has an id and a type and a set of IP ranges, continuous ranges of IPv4 or IPv6 addresses, that constitute the set of IP addresses that the party will subscribe when they subscribe to a partner. The IP range has a unique id, a starting IP address, and an ending IP address.

* **/parties** # all parties
* **/parties?partyId=<id>** # a specific party
* **/parties?partyType=<type>** # all parties of the type
* **/parties/ipranges** # all ip ranges
* **/parties/ipranges?partyId=<id>** # all the IP ranges for a party
* **/parties/ipranges?ipRangeId=<id>** # a specific IP range

# Subscriptions

A *subscription* is an agreement by a party to take and pay for access to a partner system. A subscription has a set of transactions (initial subscription, renewal, refund). Each subscription has a set of transactions that shows the transaction history of the subscription (initial subscription, renewals, and refunds).

The subscription has a party ID, a partner ID, a start date, and end Date, and a subscription ID. The combination of party ID and partner ID uniquely identifies the subscription; the subscription ID provides an alternate, single-number identifier for the subscription.

* **/subscriptions** # all subscriptions
* **/subscriptions?partnerId=<id>** # all subscriptions for a partner
* **/subscriptions?subscriptionId=<id>** # a specific subscription
* **/subscriptions/active** # all active subscriptions
* **/subscriptions/active?partyId=<id>** # all active subscriptions for a party
* **/subscriptions/active?partnerId=<id>** # all active subscriptions for a partner
* **/subscriptions/active?ip=<ip>?partnerId=<id>** # all active subscriptions for a partner
* **/subscriptions/active?partnerId=<id>&partyId=<id>** # all active subscriptions for a partner and party with an ip address in a subscribed ip range
* **/subscriptions/transactions** # all subscription transactions
* **/subscriptions/transactions/<id>** # a specific subscription transaction

# Meters

A *meter* is a count of page views. The meter subsystem comprises a set of partner-ip-address counts and a set of limits to enforce on those counts. The user accesses the system, which increments the counts and enforces the limits.

The count has a partner ID, an IP address, and an integer count. The limit has a partner ID, a name, and an integer limit value representing the count of page views at which to limit access. <ip> is an IPv4 or IPv6 internet protocol address when it occurs in the URI.

* **/meters** # all meter counts
* **/meters?partnerId=<id>** # all meter counts for a partner
* **/meters/ip** # all ip meter counts
* **/meters/ip?partnerId=<id>** # all ip meter counts for a partner
* **/meters/ip?partnerId=<id>&ip=<ip>** # the meter count for an ip accessing a partner
* **/meters/ip/<ip>/increment?partnerId=<id>** # increment the meter count for an ip accessing a partner
* **/meters/ip/limit&partnerId=<id>>&ip=<ip>** # check the status of an ip accessing a partner, returns OK, WARNING, BLOCK
* **/meters/limits/warningLimit?partnerId=<id>** # warning limit for a partner
* **/meters/limits/meteringLimit?partnerId=<id>** # metering limit for a partner

# Payment

Payment is the process of paying for a subscription. Payment interfaces to stripe.com to allow the user to pay for a subscription. The API for payment persists no data in the database. The URI supports only the POST operation with appropriate data in the body from the subscription user interface.

* **/payments** # POST a payment

# Authentication

*Authentication* is the process of verifying the identity of a user. The Subscription Management System uses a username-and-password authentication scheme as well as a key token scheme for authenticating callers of the API.

The username uniquely identifies the party, and supplying a password that matches the stored password verifies that identity. The user-password also corresponds to an optional email address and organization name.

* **/authentications** # all the username-password combinations for parties
* **/authentications?username=<username>** # retrieve, update, delete specific username
* **/authentications/login** # determine if logged in from cookie, otherwise authenticate using password

The API key authenticates a caller of the API and authorizes the calling of any API method. Phoenix creates the API keys and distributes them to calling programs as required.

* **/apikeys** # all the API key tokens

# Authorization

*Authorization* is the process of granting or refusing access to a partner. The Subscription Management System specifies partners as sets of URIs identified by regular expression patterns.

A pattern is a regular expression pattern identified by an id. An access type is a content type identified by an id with a name. An access rule relates a partner to a pattern and an access type and consists of the three ids from those objects plus a single-number, unique identifier for the rule. The access API call determines whether to authorize access to a URI for a party by finding the appropriate access rule that covers the URI; if no such rule exists, the default is to authorize the request (free content).

* **/authorizations/patterns** # all URI patterns
* **/authorizations/patterns/<id>** # a specific URI pattern
* **/authorizations/accessTypes** # all access types
* **/authorizations/accessTypes/<id>** # a specific access type
* **/authorizations/accessRules** # all access rules
* **/authorizations/accessRules/<id>** # a specific access rule
* **/authorizations/access?ip=<ip>&url=<uri>&partyId=<id>&partnerId=<id>** # whether the party or ip address has access for a given URI for a given partner; returns OK, Warning, NeedSubscription
* **/authorizations/subscription? ip=<ip>&url=<uri>&partyId=<id>&partnerId=<id>** # whether the party or ip address has access for a given URI for a given partner; returns OK, Warning, NeedSubscription

# Logging

*Logging* is the process of persisting a description of some kind of activity starting at a specific date and time and ending at a specific date and time. The Subscription Management System currently logs page views and sessions. A page view is an access of a partner with a URI. A session is a set of possibly empty related page views. Note that a session is not partner-specific.

* **/session-logs/sessions** # all sessions
* **/session-logs/sessions?startDatetime=<date>&endDatetime=<date>  
  &partyId=<id>&ip=<ip>** # selected set of sessions based on some combination of parameters, all of which are optional
* **/session-logs/sessions/create** # create a new session
* **/session-logs/page-views** # all page views
* **/session-logs/page-views?startDatetime=<date>&endDatetime=<date>  
  &partyId=<id>&ip=<ip>** # all page views between two dates, inclusive, and/or associated with a session for a specified party, and/or associated with a session with a specified ip address
* **/session-logs/page-views/create** # create a new page view in a session, set end date of session; body contains session id, page-view date-time, ip address