

Commerce Partner API Release Notes

Documentation Version	Release Notes
December 21, 2023	<ul style="list-style-type: none"> Query parameters to bypass restrictions added to Preview / Create Transfer <ul style="list-style-type: none"> ignore-order-returns : send true to bypass returnable purchase restriction expire-open-pas : send true to bypass open PA restriction (will expire the open PAs and their associated licenses) See VIP to VIPMP Seamless Move (Transfers) for more information
December 5, 2023	<ul style="list-style-type: none"> Added new Market Segment column in monthly recon file
November 14, 2023	<ul style="list-style-type: none"> External Reference ID can now be updated for Reseller, Customer, and Order Added PREVIEW_RENEWAL order type (capability to preview what renewal order will be based on customer's renewal preferences) Implemented rules for Sign Introductory Offers (maximum purchase quantity of 1 and subscription is non-renewable) <ul style="list-style-type: none"> Errors will be returned (existing error codes) if an attempt is made to purchase more than 1 quantity or enable autorenewal Transfers are now allowed for VIP customers in the renewal window. See Seamless Move Eligibility Rules for full details VIP customers who committed to 3YC prior to March 27th, 2023 are not eligible to transfer to VIPMP unless their 3YC end date is before their next anniversary date Added reseller-id query parameter to Get Order History API Account ID and Anniversary Date have been added to customer-facing emails Partners may now choose to receive two invoices and reconciliation files per month – one for new orders and one for returns <ul style="list-style-type: none"> Default partner setup is one invoice/recon file
	<ul style="list-style-type: none">

In the table of contents and throughout this document, sections that have been updated for this release are highlighted in yellow.

Introduction	4
Identity Management & Authentication	5
Health Check	6
1. Ping.....	6
2. Authenticated Ping	6
Reseller Account Management	7
1. Create Reseller Account	7
2. Get Reseller Account.....	10
3. Update Reseller Account	11
Customer Account Management.....	12
Market Segment	12
Market Segment API Changes	12
Market Segment Workflows	13
Three Year Commit	14
Introduction to Three Year Commit	14
3YC Workflows	14
3YC API Changes	15
3YC Flow Diagrams	18
1. Create Customer Account.....	19
2. Get Customer Account	22
3. Update Customer Account	24
Order Management.....	26
Volume Discounting.....	26
1. Create Order.....	27
Create Order Scenarios	29
2. Get Order Details	34
3. Update Order ExternalReferenceId (New API).....	35
4. Get Order History for Customer.....	36
Subscription Management.....	38
Auto-Renewal Scenarios	38
1. Get Subscription Details	39
1. Get All Subscriptions for a Customer	40
2. Update Subscription Auto-Renewal	41
VIP to VIPMP Seamless Move (Transfers):.....	42
Seamless Move Eligibility Rules	42
1. Preview Offers	43
3. Transfer Subscriptions.....	44
4. Get Transfer Details	46

Appendix 47

- 1. Headers47
- 2. Idempotency – Correlation ID Header47
- 3. HTTP Status Codes48
- 4. Status Codes & Error Handling49**
- 5. Supported Countries and Locales58
- 6. Resources and Fields62
- 7. Validations/Regular Expressions.....67
- 8. Support68

Introduction

This API allows partner cloud marketplaces to create and manage both Reseller and Customer accounts, as well as place orders of select Adobe products for those Customers. Before placing an Order, both a Reseller and Customer account must exist. Each Customer must be tied to a Reseller, and each Reseller must be tied to a Distributor.

The Distributor → Reseller → Customer → Order relationships are unidirectionally one-to-many. This means one Distributor can have many Resellers, but each reseller may only belong to one Distributor. Similarly, a Reseller may have many Customers, but a Customer can only be tied to one Reseller. If a reseller wishes to work with multiple distributors, a new Reseller account must be created for each Distributor. This is the same for customers with multiple resellers: a new Customer account must be created for each Reseller.

- Marketplace presents Adobe Terms & Conditions to Reseller
- If a Reseller accepts T&C, marketplace creates an account for reseller with **Create Reseller Account**
- Sometime prior to an order being placed, Reseller creates an account for a Customer and the marketplace calls **Create Customer Account**
 - Customer resource is returned with link to **Get Customer Account Details**
 - The end customer's address and contact information can be updated through a call to **Update Customer Account**
- Once a Customer account exists, Reseller places an order in marketplace. Marketplace then calls **Create Order** for all eligible Adobe products in the order
 - Order resource is returned with links to **Get Order Details** for the order itself and **Get Subscription Details** for any subscriptions associated with the order
 - Orders can be cancelled within 14 days of placing the order. When a Reseller cancels an order in the marketplace, the marketplace should call **Cancel Order** to send a cancellation to Adobe
- Subscriptions that are created by an order will be set to auto-renew by default. This option, along with the quantity of licenses to renew, can be updated using an **Update Subscription Auto-Renewal** call

Customers can also be transferred from the VIP buying program to VIPMP. This is done through the transfer APIs (Preview Offers, Transfer Subscriptions, and Get Transfer Details). The **Preview Offers** call is an optional API to preview a customer's eligibility for transfer, as well as their current subscriptions and renewal date(s).

Identity Management & Authentication

The Commerce Partner API uses Adobe's standard JWT (Service Account) Authentication.

All API calls for a distributor originate from that distributor's IMS Organization. After a direct partner signs the contract and provides necessary onboarding information, Adobe creates an IMS Organization for that partner with admins that the partner specified. Admins can manage users (other admins and developers) using the [Adobe Admin Console](#). Once the organization is created, production credentials can be generated self-service through the [Adobe Developer Console](#). System Admins and Developers of the organization will have access to the Developer Console.

All API calls (with the exception of /ping) requires a valid IMS access token. To get a valid access token, you must use the credentials (from the developer console) to generate a JWT and exchange the JWT for an access token using the IMS exchange API. For more information on creating a project in the Developer Console and retrieving an access token, please visit these links:

- <https://www.adobe.io/authentication/auth-methods.html#!AdobeDocs/adobeio-auth/master/AuthenticationOverview/ServiceAccountIntegration.md>
- <https://www.adobe.io/authentication/auth-methods.html#!AdobeDocs/adobeio-auth/master/JWT/JWT.md>

For access to the sandbox environment, credentials will be created by Adobe and shared. The authentication flow is the same in the sandbox environment (generating JWT, exchanging JWT for access token) – the difference is in the Partner API URL and the IMS URL.

	Sandbox	Production
Partner API Base URL	https://partnersandbox-stage.adobe.io	https://partners.adobe.io
IMS Base URL	https://ims-na1-stg1.adobelogin.com	https://ims-na1.adobelogin.com
Metascope	ent_partners_sdk	ent_partners_sdk

Health Check

1. Ping

GET /ping

REQUEST HEADERS:

X-Api-Key: <your-api-key>

REQUEST BODY: None

RESPONSE BODY:

pong

2. Authenticated Ping

GET /partnerservice/ping

REQUEST HEADERS:

X-Api-Key: <your-api-key>

Authorization: Bearer <token>

REQUEST BODY: None

RESPONSE BODY:

pong

Assumptions:

- The authenticated ping endpoint is used to verify that the service is available and that your API Key and Authorization Token are valid

Reseller Account Management

1. Create Reseller Account

POST /v3/resellers

REQUEST HEADERS: [See Headers Section](#)

REQUEST BODY: Reseller resource without read-only fields

```
{
  "distributorId" : "345434543",
  "externalReferenceId" : "888",
  "companyProfile" : {
    "companyName" : "Fairmont",
    "preferredLanguage" : "en-US",
    "marketSegments" : ["COM", "EDU"],
    "address" : {
      "country" : "US",
      "region" : "CA",
      "city" : "San Jose",
      "addressLine1" : "200 Fairmont Ave",
      "addressLine2" : "Apt 123",
      "postalCode" : "95110-1234",
      "phoneNumber" : "800-123-4567"
    },
    "contacts" : [{
      "firstName" : "Mickey",
      "lastName" : "Mouse",
      "email" : "mickey@mouse.com",
      "phoneNumber" : "408-123-4567"
    }]
  }
}
```

RESPONSE BODY: Reseller resource

```
{
  "distributorId": "345434543",
  "externalReferenceId": "888",
  "resellerId": "5556667778",
  "companyProfile": {
    "companyName": "Fairmont",
    "preferredLanguage": "en-US",
    "marketSegments": ["COM", "EDU"],
    "address": {
      "country": "US",
      "region": "CA",
      "city": "San Jose",
      "addressLine1": "200 Fairmont Ave",
      "addressLine2": "Apt 123",
      "postalCode": "95110-1234",
      "phoneNumber": "800-123-4567"
    },
    "contacts": [{
      "firstName": "Mickey",
      "lastName": "Mouse",
      "email": "mickey@mouse.com",
      "phoneNumber": "408-123-4567"
    }]
  },
  "creationDate": "2019-05-02T22:49:52Z",
  "status": "1000",
  "links": {
    "self": {
      "uri": "/v3/resellers/5556667778",
      "method": "GET",
      "headers": []
    }
  }
}
```

HTTP STATUS CODES:

201	Reseller account successfully created
400	Bad request
401	Invalid Authorization token
403	Invalid API Key

Assumptions:

- Partner will present reseller with T&Cs and make this call after reseller agrees
- Reseller Account ID is created by this service and returned synchronously
- Market segment is only commercial for M1
- externalReferenceId should be used to pass in the marketplace's identifier for the reseller
 - Optional and does not need to be unique
- Distributor ID -> Reseller ID -> Customer ID relationship must be unique

- Synchronous call will return an account with status "1002" (Pending) while final validation happens asynchronously. The account may then become Active or Inactive. Pending resellers can create customers, but not update the reseller account.
- Contact names for existing email addresses cannot be changed through this API. If you send a different name for an existing email, that name will be returned in the synchronous response, but it will not persist in our system. The existing first and last name of the contact will be used and returned in any GET calls
- Resellers can be created with optional market segments
 - If no market segments are sent, the reseller will be created with COM
- Resellers may sell into multiple market segments but must always have at least one (default COM)

2. Get Reseller Account

GET /v3/resellers/<reseller-id>

REQUEST HEADERS: [See Headers Section](#)

REQUEST BODY: None

RESPONSE BODY: Same as Create Reseller (Reseller resource)

```
{
  "distributorId": "345434543",
  "externalReferenceId": "888",
  "resellerId": "5556667778",
  "companyProfile": {
    "companyName": "Fairmont",
    "preferredLanguage": "en-US",
    "marketSegments": ["COM", "EDU"],
    "address": {
      "country": "US",
      "region": "CA",
      "city": "San Jose",
      "addressLine1": "200 Fairmont Ave",
      "addressLine2": "Apt 123",
      "postalCode": "95110-1234",
      "phoneNumber": "800-123-4567"
    },
    "contacts": [{
      "firstName": "Mickey",
      "lastName": "Mouse",
      "email": "mickey@mouse.com",
      "phoneNumber": "408-123-4567"
    }]
  },
  "creationDate": "2019-05-02T22:49:52Z",
  "status": "1000",
  "links": {
    "self": {
      "uri": "/v3/resellers/5556667778",
      "method": "GET",
      "headers": []
    }
  }
}
```

HTTP STATUS CODES:

200	Reseller account details successfully returned
400	Bad request
401	Invalid Authorization token
403	Invalid API Key
404	Invalid reseller ID

3. Update Reseller Account

PATCH /v3/resellers/<reseller-id>

REQUEST HEADERS: [See Headers Section](#)

REQUEST BODY: CompanyProfile for reseller with optional externalReferenceId

```
{
  "externalReferenceId" : "999",
  "companyProfile" : {
    "companyName" : "Fairmont",
    "preferredLanguage" : "en-US",
    "marketSegments" : ["COM", "EDU"],
    "address" : {
      "country" : "US",
      "region" : "CA",
      "city" : "San Jose",
      "addressLine1" : "200 Fairmont Ave",
      "addressLine2" : "Apt 123",
      "postalCode" : "95110-1234",
      "phoneNumber" : "800-123-4567"
    },
    "contacts" : [
      {
        "firstName" : "Mickey",
        "lastName" : "Mouse",
        "email" : "mickey@mouse.com",
        "phoneNumber" : "408-123-4567"
      }
    ]
  }
}
```

RESPONSE BODY: Reseller resource

```
{ Reseller resource }
```

HTTP STATUS CODES:

200	Reseller account successfully updated
400	Bad request
401	Invalid Authorization token
403	Invalid API Key
404	Invalid Reseller ID

Assumptions:

- Update replaces existing data
- Non-editable fields (in red) need to be passed in but need to match the original values. Otherwise, an error is returned
- externalReferenceId can now be updated

Customer Account Management

Market Segment

Market Segment API Changes

Reseller Resource

- **Market Segments:** New field *companyProfile.marketSegments*
 - List of market segments the reseller can sell into
 - Can be set during Create Reseller and Update Reseller
 - Default is [COM]

Customer Resource

- **Market Segment:** New field *companyProfile.marketSegment*
 - Can be set during Create Customer and Update Customer
 - A customer may only have a single market segment
 - Default is COM
- **Market Sub-Segments:** New field *companyProfile.marketSubSegments*
 - Read Only: Market subsegments are set by the customer in the Adobe Admin Console
 - Currently subsegments are only applicable for the EDU market segment

Market Segment (reseller/customer)

value	description
COM	Commercial (default)
EDU	Education
GOV	Government

Market Sub-Segment (customer only)

value	description
K_12	K-12 School
HIGHER_ED	Higher Education
NON_PROFIT	Nonprofit

Market Segment Workflows

1. Setting Market Segments (Reseller)

- Reseller market segments can be set as part of a create or update request
- A reseller's market segments may be changed at any point, but removing a market segment may cause future orders/renewals to fail
- All existing resellers with no market segment will be initialized with only COM market segment

[\[POST Create Reseller API\]](#)

[\[PATCH Update Reseller API\]](#)

2. Setting Market Segment (Customer)

- Customer market segment can be set as part of a create or update request
- A customer's market segments can only be changed if the customer has no active subscriptions
- All existing customers with no market segment will be initialized to COM

[\[POST Create Customer API\]](#)

[\[PATCH Update Customer API\]](#)

3. Setting Market Sub-Segment (Customer only)

- A customer's market sub-segment can only be set by the customer in the Adobe Admin Console
- SANDBOX: For testing, the sub-segment can be set in the sandbox partner portal

[\[GET Customer API\]](#)

4. Transferring Customers to your marketplace (Customer only)

- Customers with any market segment can be transferred as long as the new reseller is enabled to sell into the customer's market segment

[\[POST Create Transfer API\]](#)

[\[GET Customer API\]](#)

5. Purchasing Offers for a Market Segment

- Customers may only purchase offers in their market segment
- Resellers may only purchase offers in a market segment they are enabled for
- Requests to purchase (2129 – Reason Code: INELIGIBLE_MARKET_SEGMENT)

[\[POST Create/Preview Order API\]](#)

New/Updated Error Codes

Error Code	Error Message	Applicable APIs	Change Details
2135 (new error code)	Invalid market segment used for customer	Create Customer, Update Customer	New Error Code
2129 (existing error code)	Customer is not eligible to purchase Offer ID	Preview Order, Create Order	New Additional Details Reason Code: INELIGIBLE_MARKET_SEGMENT

Three Year Commit

Introduction to Three Year Commit

3-Year Commit (3YC) is a loyalty program that allows a customer to get a **greater discount level** as well as a **price lock** for 3 terms (the current term and 2 additional terms). The customer commits to a minimum quantity that they must purchase *and maintain* throughout the 3-year commitment period.

It is important to use the Preview Order API to determine the correct discount level. Preview Order uses the customer's 3YC commitment as well as any accepted commitment requests in order to return the Offer ID with the best discount.

3YC Workflows

There are four ways that a customer can become enrolled in 3YC. In all scenarios, the customer is required to accept the commitment terms (the minimum quantities and commitment end date) in the Adobe Admin Console before receiving any discounts.

- 1. New Customer (partner-initiated):** During customer creation, partners may set the requested minimum quantities, which would start the 3YC customer acceptance workflow after the account becomes active.
[\[POST Create Customer API\]](#)
[\[GET Get Customer API\]](#)
- 2. Existing Customer (partner-initiated):** Existing customers (with active 3YC or not) can be updated with new requested minimum quantities, which would start the 3YC customer acceptance workflow once the account is updated from the request.
[\[PATCH Update Customer API\]](#)
[\[GET Get Customer API\]](#)
- 3. Existing Customer (customer-initiated):** Existing customers (not already in 3YC) that meet the 3YC requirements may be presented an offer in Adobe admin Console to enter 3YC *with the minimum quantities set to the customer's current quantities*.
[\[GET Get Customer API\]](#)
- 4. Existing VIP Customer (partner-initiated):** Existing VIP customers with active 3YC can be transferred to VIPMP and retain their 3YC commitment terms, which can be retrieved using the GET Customer API.
[\[POST Transfer Subscriptions API\]](#)
[\[GET Get Customer API\]](#)

3YC API Changes

Customer Resource

- New *benefits* array of *benefit* objects
- New potential 3YC discount levels
 - **LICENSE:** 12, 13, 14
 - **CONSUMABLES:** TA, TB, TC, TD, TE, TF, TG
- Discount levels will be updated whenever the commitment status changes (new commitment or expiration)
- During Order Preview, if the customer has an ACCEPTED *commitmentRequest* for a quantity that would be reached in that order, the preview response will use the discount level for that *commitmentRequest*
 - Placing the order for that quantity would make the *commitmentRequest* COMMITTED as well as creating the *commitment* object
- Customer can be enrolled in 3YC for either LICENSE or CONSUMABLES or both.
 - If only enrolled for one, there is no minimum quantity for the other type and the customer gets no discount for that type
 - Can enroll in one *offerType* first and then add the other later. The committed offerType as well as the new offerType must be included in *commitmentRequest*

Object	Sample Request	Notes
commitmentRequest	<pre> REQUEST { "companyProfile":{ "companyName":"Fairmont", "preferredLanguage":"en-US", "address":{ "country":"US", "region":"CA", "city":"San Jose", "addressLine1":"200 Fairmont Ave", "addressLine2":"Apt 123", "postalCode":"95110-1234", "phoneNumber":"800-123-4567" }, "contacts":[] }, "benefits":[{ "type":"THREE_YEAR_COMMIT", "commitmentRequest":{ "minimumQuantities":[{ "offerType":"LICENSE", "quantity":10 }, { "offerType":"CONSUMABLES", "quantity":1000 }] } }] } </pre>	<ul style="list-style-type: none"> Used to request 3YC for a customer without a commitment Used to request 3YC quantity increase for a customer with an existing commitment Will overwrite existing <i>commitmentRequest</i> Can not be requested alongside <i>recommitmentRequest</i> or if customer has an existing <i>recommitmentRequest</i> Does not need to include quantities for all offerTypes endDate will be 2 years after the current cotermDate <ul style="list-style-type: none"> If customer does not have a cotermDate (they have not placed an order) at the time of acceptance, the endDate will be 3 years from the acceptance date and the cotermDate will be set at that point

recommitmentRequest**REQUEST**

```

{
  "companyProfile":{
    "companyName":"Fairmont",
    "preferredLanguage":"en-US",
    "address":{
      "country":"US",
      "region":"CA",
      "city":"San Jose",
      "addressLine1":"200 Fairmont Ave",
      "addressLine2":"Apt 123",
      "postalCode":"95110-1234",
      "phoneNumber":"800-123-4567"
    },
    "contacts":[

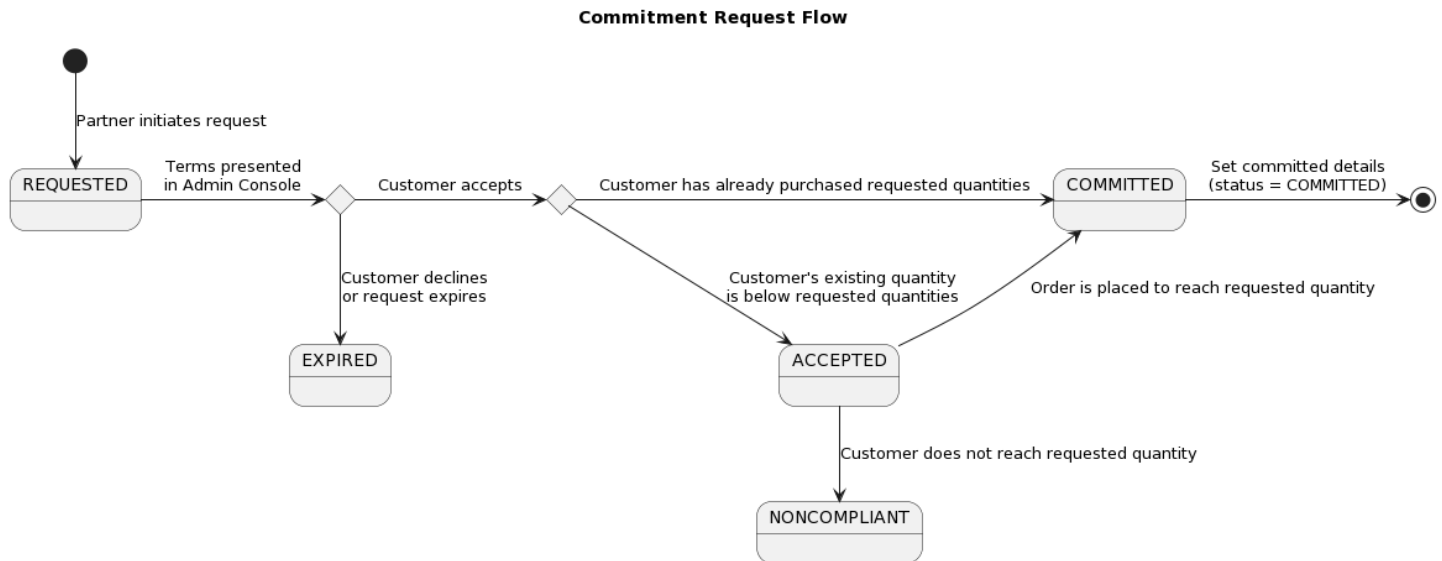
    ]
  },
  "benefits":[
    {
      "type":"THREE_YEAR_COMMIT",
      "recommitmentRequest":{
        "minimumQuantities":[
          {
            "offerType":"LICENSE",
            "quantity":10
          },
          {
            "offerType":"CONSUMABLES",
            "quantity":1000
          }
        ]
      }
    }
  ]
}

```

- Used to request 3YC recommitment
- Must have existing commitment and less than 30 days before the commitment end date
- Will overwrite existing recommitmentRequest
- Can not be requested alongside *commitmentRequest* or if customer has an existing *commitmentRequest*

3YC Flow Diagrams

The diagram below shows the lifecycle of the commitmentRequest object. The same lifecycle applies for the recommitmentRequest, with the difference being that recommitmentRequest can't move to COMMITTED status until the current commitment ends.



```

"benefits":[
  {
    "type":"THREE_YEAR_COMMIT",
    "commitment":{...},
    "commitmentRequest":{
      "startDate":"2022-07-17",
      "endDate":"2025-04-28",
      "status":"REQUESTED",
      "minimumQuantities":[
        {
          "offerType":"LICENSE",
          "quantity":10
        },
        {
          "offerType":"CONSUMABLES",
          "quantity":1000
        }
      ]
    },
    "recommitmentRequest":{...}
  }
]
  
```

1. Create Customer Account

POST /v3/customers

REQUEST HEADERS: [See Headers Section](#)

REQUEST BODY: Customer resource without read-only fields

```
{
  "resellerId" : "5556667778",
  "externalReferenceId" : "342",
  "companyProfile" : {
    "companyName" : "Fairmont",
    "preferredLanguage" : "en-US",
    "marketSegment" : "EDU",
    "address" : {
      "country" : "US",
      "region" : "CA",
      "city" : "San Jose",
      "addressLine1" : "200 Fairmont Ave",
      "addressLine2" : "Apt 123",
      "postalCode" : "95110-1234",
      "phoneNumber" : "800-123-4567"
    },
    "contacts" : [{
      "firstName" : "Donald",
      "lastName" : "Duck",
      "email" : "donald@duck.com",
      "phoneNumber" : "408-123-4567"
    }]
  }
}
```

RESPONSE BODY: Customer resource

```
{
  "externalReferenceId" : "342",
  "customerId" : "9876543210",
  "resellerId" : "5556667778",
  "companyProfile" : {
    "companyName" : "Fairmont",
    "preferredLanguage" : "en-US",
    "marketSegment" : "EDU",
    "address" : {
      "country" : "US",
      "region" : "CA",
      "city" : "San Jose",
      "addressLine1" : "200 Fairmont Ave",
      "addressLine2" : "Apt 123",
      "postalCode" : "95110-1234",
      "phoneNumber" : "800-123-4567"
    },
    "contacts" : [
      {
        "firstName" : "Donald",
        "lastName" : "Duck",
        "email" : "donald@duck.com",
        "phoneNumber" : "408-123-4567"
      }
    ]
  },
  "discounts" : [{
    "offerType" : "LICENSE",
    "level" : "02",
  }],
  "cotermDate" : "",
  "creationDate" : "2019-05-02T22:49:52Z",
  "status" : "1002",
  "links" : {
    "self" : {
      "uri" : "/v3/customers/9876543210",
      "method" : "GET",
      "headers" : []
    }
  }
}
```

HTTP STATUS CODES:

201	Customer account created
400	Bad request
401	Invalid Authorization token
403	Invalid API Key
404	Invalid Reseller ID

Assumptions:

- Customer Account id is created by this service and returned synchronously
- Customer payment instrument is managed on the partner marketplace
- cotermdate is normally calculated when a customer's first order is placed
 - Most subscriptions will end or renew on the cotermdate (with the exception of Stock Credit Pack)
- cotermdate can be sent in the request to create an Extended Term customer
- externalReferenceId should be used to pass in the marketplace's Customer ID
 - Optional and does not need to be unique
- Contacts will be admins for the customer's account and receive an admin welcome email
 - Contact names for existing email addresses cannot be changed. If you send a different name for an existing email, that name will be returned in the synchronous response but it will not persist in our system. The existing first and last name of the contact will be used and returned in any GET calls
- If the Create Customer call succeeds, it will always return a Customer with status **1002** (Pending) while final validation happens asynchronously. The account may then become active or inactive. Pending customers can place orders, but they may not update their account using the Update Customer API until the account becomes active or inactive. All orders placed for a pending customer will fail if the customer becomes inactive
- discounts contains a list of objects with the customer's discount levels for each offerType
- Customer can be created with a specified market segment
 - Reseller must be enabled for that market segment
 - If no market segment is sent in the request, the customer will be commercial (COM)

2. Get Customer Account

GET /v3/customers/<customer-id>

REQUEST HEADERS: [See Headers Section](#)

REQUEST BODY: None

RESPONSE BODY: Customer resource (same as Create Customer)

```
{
  "externalReferenceId" : "342"
  "customerId" : "9876543210",
  "resellerId" : "5556667778",
  "companyProfile" : {
    "companyName" : "Fairmont",
    "preferredLanguage" : "en-US",
    "marketSegment" : "EDU",
    "marketSubSegments" : ["K_12"],
    "address" : {
      "country" : "US",
      "region" : "CA",
      "city" : "San Jose",
      "addressLine1" : "200 Fairmont Ave",
      "addressLine2" : "Apt 123",
      "postalCode" : "95110-1234",
      "phoneNumber" : "800-123-4567"
    },
    "contacts" : [{
      "firstName" : "Mickey",
      "lastName" : "Mouse",
      "email" : "mickey@mouse.com",
      "phoneNumber" : "408-123-4567"
    }
  ],
  "discounts" : [{
    "offerType" : "LICENSE",
    "level" : "02",
  }],
  "cotermDate" : "2020-05-02",
  "creationDate" : "2019-05-02T22:49:52Z",
  "status" : "1000",
  "links" : {
    "self" : {
      "uri" : "/v3/customers/9876543210",
      "method" : "GET",
      "headers" : []
    }
  }
}
```

HTTP STATUS CODES:

200	Customer account details successfully returned
400	Bad request
401	Invalid Authorization token
403	Invalid API Key
404	Invalid customer ID

3. Update Customer Account

PATCH /v3/customers/<customer-id>

REQUEST HEADERS: [See Headers Section](#)

REQUEST BODY: CompanyProfile object with optional externalReferenceId

```
{
  "externalReferenceId" : "99999",
  "companyProfile" : {
    "companyName" : "Fairmont",
    "preferredLanguage" : "en-US",
    "marketSegment" : "COM",
    "address" : {
      "country" : "US",
      "region" : "CA",
      "city" : "San Jose",
      "addressLine1" : "200 Fairmont Ave",
      "addressLine2" : "Apt 123",
      "postalCode" : "95110-1234",
      "phoneNumber" : "800-123-4567"
    },
    "contacts" : [
      {
        "firstName" : "Mickey",
        "lastName" : "Mouse",
        "email" : "mickey@mouse.com",
        "phoneNumber" : "408-123-4567"
      }
    ]
  }
}
```

RESPONSE BODY: Same as Get Customer Account Details (Customer resource)

```
{ Customer resource }
```

TECHNICAL NOTE: Any contacts specified in this call will receive the admin welcome email. This can be used to resend the email if an end-user did not receive it.

HTTP STATUS CODES:

200	Account successfully updated
400	Bad request
401	Invalid Authorization token
403	Invalid API Key
404	Invalid Customer ID

Assumptions:

- Update replaces existing data

- Non-editable fields (in red) need to be passed in but need to match the original values. Otherwise, an error is returned
- New contacts will be admins for customer's account.
- Any contacts that are removed will still remain admins. To remove admins, a customer admin must use the Adobe Admin Console.
- The name of an existing contact cannot be changed through this API. Customer admin must use the Admin Console to update users' names.
- Customer market segment can be changed through the Update Customer API **only if the customer has no active subscriptions**
 - Reseller must be enabled for the new market segment
- Customer externalReferenceId may now be changed through the Update Customer API

Benefits with either *commitmentRequest* **OR** *recommitmentRequest* may be sent in the request to:

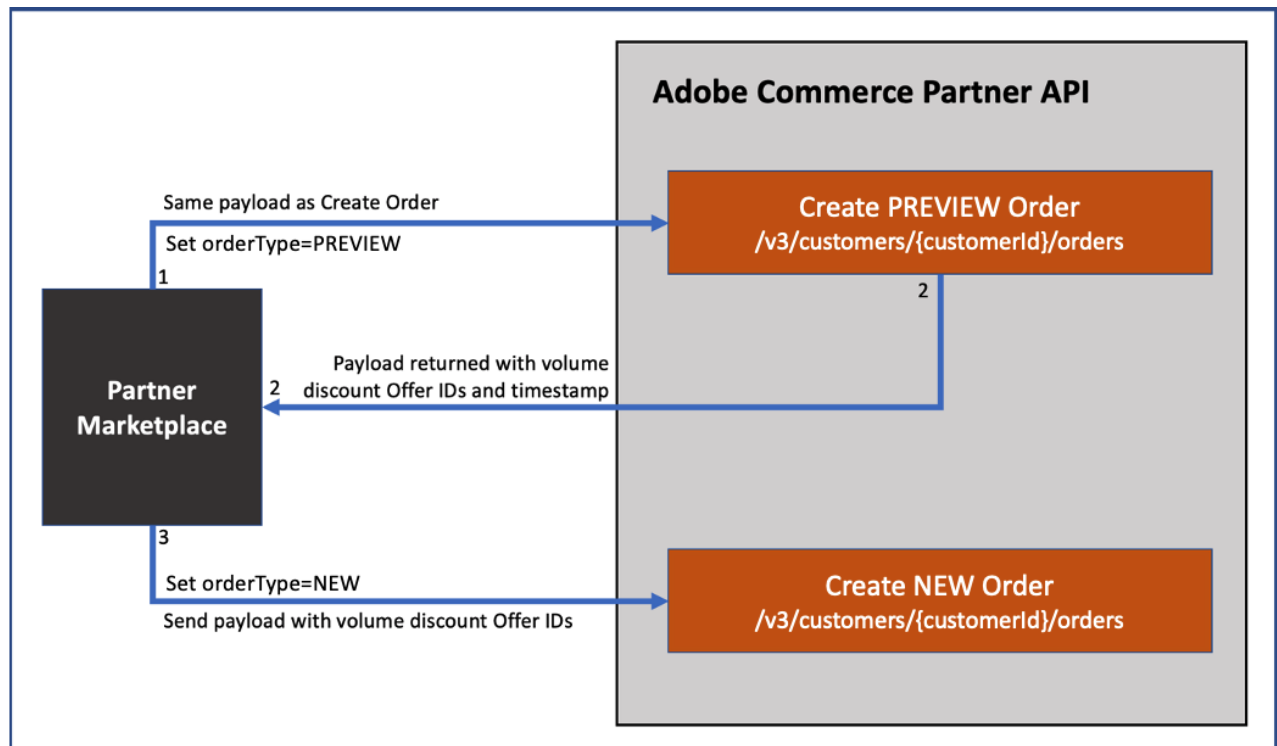
- Request new 3YC for existing customer
- Request 3YC quantity increase for customer with existing commitment
- Request 3YC *recommitment* for customer with existing commitment
 - Only allowed if the customer has a COMMITTED commitment

Order Management

Volume Discounting

To ensure the best volume discount Offer IDs are used for each order, it is recommended to create an order with type PREVIEW prior to placing an order. Creating a PREVIEW order is an optional call made before placing an order that will determine the best volume discount Offer IDs for the order that the customer is eligible for. The payload is the same as creating a NEW order, and the response returns that payload with the best available offer IDs, which can then be used to place an order through the Create Order API (if the orderType is changed from PREVIEW to NEW)

- Creating a PREVIEW order does not actually place an order. An order with type NEW will need to be created in order to place an order
- A customer's volume discount level will be evaluated after each order. It is based on the customer's current level and the cumulative quantity of volume-discount-eligible products in that order
- Customers transferring their subscriptions from VIP will retain their existing discount level
- A customer's discount level will be reevaluated at renewal time and can go up or down depending on the total renewal quantity across all their subscriptions



1. Create Order

POST /v3/customers/<customer-id>/orders

REQUEST HEADERS: [See Headers Section](#)

REQUEST BODY: Order resource without read-only fields

```
{
  "externalReferenceId" : "759", (optional)
  "currencyCode" : "USD",
  "orderType" : "NEW | RETURN | PREVIEW | PREVIEW_RENEWAL",
  "referenceOrderId" : "", (for returns only)
  "lineItems" : [
    {
      "extLineItemNumber" : 4,
      "offerId" : "80004567EA01A12",
      "quantity" : 1
    }
  ]
}
```

Assumptions:

- orderType is required in the Create Order request
 - must be NEW, RETURN, PREVIEW, or PREVIEW_RENEWAL
- referenceOrderId is required for RETURN orders and should not be included for other order types

RESPONSE BODY: Order resource

```
{
  "externalReferenceId" : "759",
  "orderId" : "0123456789",
  "customerId" : "9876543210",
  "orderType" : "NEW",
  "referenceOrderId" : "",
  "currencyCode" : "USD",
  "creationDate" : "2019-05-02T22:49:54Z",
  "status" : "1002",
  "lineItems" : [
    {
      "extLineItemNumber" : 4,
      "offerId" : "80004567EA01A12",
      "quantity" : 1,
      "subscriptionId" : "",
      "status" : "1002"
    }
  ],
  "links" : {
    "self" : {
      "uri" : "/v3/customers/9876543210/orders/0123456789",
      "method" : "GET",
      "headers" : []
    }
  }
}
```

Create Order Scenarios

Action	Sample Request/Response	Notes
NEW	<p>REQUEST</p> <pre>{ "orderType" : "NEW", "externalReferenceId" : "759", "currencyCode" : "USD", "lineItems" : [{ "extLineItemNumber" : 4, "offerId" : "80004567EA01A12", "quantity" : 1 }] }</pre> <hr/> <p>RESPONSE</p> <pre>{ "referenceOrderId" : "", "orderType" : "NEW", "externalReferenceId" : "759", "customerId" : "9876543210", "orderId" : "5120008001", "currencyCode" : "USD", "creationDate" : "2019-05-02T22:49:54Z", "status" : "1002", "lineItems" : [{ "extLineItemNumber" : 4, "offerId" : "80004567EA01A12", "quantity" : 1, "status" : "1002", "subscriptionId" : "", }], "links" : {...} }</pre>	<ul style="list-style-type: none"> • The subscriptionId may be blank while the order is processing. Once the order is complete (status 1000), then all subscriptionIds will be populated and will not change • Create order will be used to add seats for the current term • Order ID is created by this service and returned synchronously • Order may fail asynchronously, and status will be updated to reflect the failure

RETURN (Cancellation)	<div> REQUEST <pre> { "referenceOrderId" : "0123456789", "orderType" : "RETURN", "externalReferenceId" : "759", "currencyCode" : "USD", "lineItems" : [{ "extLineItemNumber" : 4, "offerId" : "80004567EA01A12", "quantity" : 1 }] } </pre> </div> <hr/> <div> RESPONSE <pre> { "referenceOrderId" : "0123456789", "orderType" : "RETURN", "externalReferenceId" : "759", "orderId" : "911000833", "customerId" : "9876543210", "currencyCode" : "USD", "creationDate" : "2019-05-02T22:49:54Z", "status" : "1002", "lineItems" : [{ "extLineItemNumber" : 4, "offerId" : "80004567EA01A12", "quantity" : 1, "subscriptionId" : "", "status" : "1002" }], "links" : {...} } </pre> </div>
---------------------------------	---

PREVIEW**REQUEST**

```
{
  "orderType" : "PREVIEW",
  "externalReferenceId" : "759",
  "currencyCode" : "USD",
  "lineItems" : [
    {
      "extLineItemNumber" : 4,
      "offerId" : "80004567EA01A12",
      "quantity" : 1
    }
  ]
}
```

RESPONSE

```
{
  "referenceOrderId" : "",
  "orderType" : "PREVIEW",
  "externalReferenceId" : "759",
  "orderId" : "",
  "customerId" : "9876543210",
  "currencyCode" : "USD",
  "creationDate" : "2019-05-02T22:49:54Z",
  "status" : ""
  "lineItems" : [
    {
      "extLineItemNumber" : 4,
      "offerId" : "80004567EA01A12",
      "quantity" : 1,
      "subscriptionId" : "",
      "status" : ""
    }
  ]
}
```

- No orderId, subscriptionId, status, links
- Response can be used to place a new order request if the orderType is changed to **NEW**
- Best available offer ID for this customer and this order is returned
 - Input Offer ID can be any level representing the same product
 - If the Offer IDs in the request are a better discount than customer is allowed, then the correct lower-level Offer ID will be returned.
 - For a **NEW** order, the request would be rejected if the customer is not eligible for an Offer ID
- If NEW order would be rejected, then PREVIEW will be rejected with the same error

PREVIEW_RENEWAL	<p>REQUEST</p> <pre>{ "orderType" : "PREVIEW_RENEWAL" }</pre> <hr/> <p>RESPONSE</p> <pre>{ "referenceOrderId" : "", "orderType" : "PREVIEW_RENEWAL", "externalReferenceId" : "759", "orderId" : "", "customerId" : "9876543210", "currencyCode" : "USD", "creationDate" : "2019-05-02T22:49:54Z", "status" : "" "lineItems" : [{ "extLineItemNumber" : 4, "offerId" : "80004567EA01A12", "quantity" : 1, "subscriptionId" : " e0b170437c4e96ac5428364f674dffNA", "status" : "" }] }</pre>	<ul style="list-style-type: none"> • No orderId, status, links. • The response indicates what would be in the RENEWAL order based on the auto-renewal preferences (autoRenewal.enabled and autoRenewal.renewalQuantity) on the customer's subscriptions • If the customer does not have any subscriptions with autoRenewal enabled, then an error will be returned • Best available offer IDs for the renewal order are returned
------------------------	---	--

HTTP STATUS CODES:

202	Order accepted
400	Bad request
401	Invalid Authorization token
403	Invalid API Key
404	Invalid Customer ID

Assumptions:

- Renewals will be managed at the subscription level. No new order is expected from the marketplace. Adobe will create a renewal order on a customer's cotermDate if the customer has subscriptions set to autorenew. The renewal Order ID can be retrieved by parsing the response from the Get Order History API (orderType **RENEWAL**)
- Offer IDs are defined and communicated in published price lists
- Each Offer ID is only valid for certain currencies. The currency code send in the request must match the Offer IDs as well as the currency that the distributor is configured for
- externalReferenceId is used to pass in an ID for financial reconciliation
 - The externalReferenceId does not need to be unique and is an optional field

Restrictions:

- Maximum quantity per line item:
 - 10,000 for Team products
 - 200,000 for Enterprise products
- Maximum line items per order:
 - 499
- Maximum length of externalReferenceId:
 - 35 characters
- Maximum extLineItemNumber:
 - 999999

2. Get Order Details

GET /v3/customers/<customer-id>/orders/<order-id>

REQUEST HEADERS: [See Headers Section](#)

REQUEST BODY: None

RESPONSE BODY: Same as Create Order (Order resource)

```
{
  "externalReferenceId" : "759",
  "orderId" : "0123456789",
  "customerId" : "9876543210",
  "orderType" : "NEW",
  "referenceOrderId" : "",
  "referencedOrderId" : "",
  "currencyCode" : "USD",
  "creationDate" : "2019-05-02T22:49:54Z",
  "status" : "1000",
  "lineItems" : [
    {
      "extLineItemNumber" : 4,
      "offerId" : "80004567EA01A12",
      "quantity" : 1,
      "subscriptionId" : "86756309",
      "status" : "1000"
    }
  ],
  "links" : {
    "self" : {
      "uri" : "/v3/customers/9876543210/orders/0123456789",
      "method" : "GET",
      "headers" : []
    }
  }
}
```

HTTP STATUS CODES:

200	Order details successfully retrieved
400	Bad request
401	Invalid Authorization token
403	Invalid API Key
404	Invalid Customer or Order ID

3. Update Order ExternalReferenceId

PATCH /v3/customers/<customer-id>/orders/<order-id>

REQUEST HEADERS: [See Headers Section](#)

REQUEST BODY: Updated externalReferenceId

```
{
  "externalReferenceId" : "34567"
}
```

RESPONSE BODY: Order resource with updated externalReferenceId

```
{
  "externalReferenceId" : "34567",
  "orderId" : "0123456789",
  "customerId" : "9876543210",
  "orderType" : "NEW",
  "referenceOrderId" : "",
  "referencedOrderId" : "",
  "currencyCode" : "USD",
  "creationDate" : "2019-05-02T22:49:54Z",
  "status" : "1000",
  "lineItems" : [
    {
      "extLineItemNumber" : 4,
      "offerId" : "80004567EA01A12",
      "quantity" : 1,
      "subscriptionId" : "86756309",
      "status" : "1000"
    }
  ]
}
```

HTTP STATUS CODES:

200	Order successfully updated
400	Bad request
401	Invalid Authorization token
403	Invalid API Key
404	Invalid Customer or Order ID

Assumptions:

- Only externalReferenceId may be updated through this API
- The updated externalReferenceId will be reflected in the monthly reconciliation file only if the update is made prior to the reconciliation file being generated

4. Get Order History for Customer

GET /v3/customers/<customer-id>/orders

REQUEST HEADERS: [See Headers Section](#)

QUERY PARAMETERS:

param	values	default	description
order-type	NEW TRANSFER RENEWAL	All	
reseller-id	Valid reseller IDs	All	Regardless of reseller id, only orders for the partner making the request will be returned
status	1000 1002 1004 1026	All	
reference-order-id	Valid order IDs	All	
offer-id	Valid offer IDs	All	
start-date	2019-05-02 2019-05-02T22:49:54Z	Current term start date	Date without timestamp or with timestamp in Zulu time
end-date	2019-05-02 2019-05-02T22:49:54Z	now	Date without timestamp or with timestamp in Zulu time
limit	1...100	25	Page size (max number of orders to return)
offset	0...N	0	Where to start the page

Assumptions:

- All parameters included in request (besides limit and offset) will be queried together using AND
- Any parameters that are not included will default to the value in the default column
- For **start-date** and **end-date**, a date without timestamp will be evaluated to midnight UTC
 - Dates with timestamp will only be accepted in ISO-8601 format with "Zulu" (UTC) time zone. This is the same format that all datetimes are in CPAPI responses
- The **order-type**, **status**, **offer-id** parameters can be included multiple times to specify multiple values. Including multiple values for a parameter will OR the values in the query
 - Example: To query for orders that are either status 1000 or 10002:
/v3/customers/<customer-id>?status=1000&status=1002
- Results will be sorted in descending order by creationDate
- Maximum value for offset is the total number of results. Values higher than the total count will return an error
- If a limit greater than the maximum (100) is sent, it will default to the maximum value (100)
- Links to **next** and **prev** will be included in the response if there is a valid next or previous page in the result set

RESPONSE BODY: Collection of Order resources

```
{
  "totalCount" : 9,
  "count" : 3,
  "offset" : 3,
  "limit" : 3,
  "items" : [
    { Order resource },
    { Order resource }
  ],
  "links" : {
    "self" : {
      "uri" : "/v3/customers/<customer-id>/orders?offset=3&limit=3",
      "method" : "GET",
      "headers" : []
    },
    "next" : {
      "uri" : "/v3/customers/<customer-id>/orders?offset=6&limit=3",
      "method" : "GET",
      "headers" : []
    },
    "prev" : {
      "uri" : "/v3/customers/<customer-id>/orders?offset=0&limit=3",
      "method" : "GET",
      "headers" : []
    }
  }
}
```

HTTP STATUS CODES:

200	Successfully returned order history
400	Bad request
401	Invalid Authorization token
403	Invalid API Key
404	Invalid Customer ID

Subscription Management

Auto-Renewal Scenarios

```
"autoRenewal" : {  
    "enabled" : true,  
    "renewalQuantity" : 7  
}
```

Auto-renewal preferences are managed for each subscription independently through the subscription's `autoRenewal` object. These preferences can be updated through the *Update Subscription Auto-Renewal* endpoint. The **enabled** field determines whether or not a subscription will be autorenewed, and the **renewalQuantity** manages the quantity that will be renewed.

The autorenewal preferences are not evaluated until renewal time. You are free to enable/disable autorenewal and change the quantity throughout the length of the subscription. On the customer's `cotermDate`, all of the customer's subscriptions with autorenewal enabled will renew for the specified quantity.

At renewal time, there are two possible scenarios for each subscription:

- Enabled is set to **false** – subscription will become inactive (status 1004)
- Enabled is set to **true** – subscription will be marked for renewal (status will remain 1000) for the amount specified in `renewalQuantity`. During this brief period while waiting for the renewal order to be fulfilled, the `autoRenewal` preferences may not be updated

In order to ensure all subscriptions are renewed (or not) in accordance with the autorenewal preferences, please **make sure the autorenewal preferences are set by midnight UTC the morning of the customer's cotermDate**. The customer's subscriptions will be renewed at some point soon after midnight UTC, but the exact time may change or vary by region.

If the `renewalQuantity` is higher than the current ordered quantity, then additional licenses will be provisioned after renewal. If the `renewalQuantity` less than the current ordered quantity, then some licenses will be revoked upon renewal.

Adobe will create a renewal order on a customer's `cotermDate` if the customer has subscriptions set to autorenew. The renewal order details and Order ID can be retrieved by parsing the response from the Get Order History API. This ID can then be used in subsequent Get Order calls.

1. Get Subscription Details

GET /v3/customers/<customer-id>/subscriptions/<subscription-id>

HEADERS: [See Headers Section](#)

REQUEST BODY: None

RESPONSE BODY: Subscription resource

```
{
  "subscriptionId" : "8675309",
  "currentQuantity" : 10,
  "usedQuantity" : 2,
  "offerId" : "65304470CA01012",
  "autoRenewal" : {
    "enabled" : true,
    "renewalQuantity" : 5
  },
  "renewalDate" : "2020-05-20",
  "creationDate" : "2019-05-20T22:49:55Z",
  "status" : "1000",
  "links" : {
    "self" : {
      "uri" : "/v3/customers/9876543210/subscriptions/8675309",
      "method" : "GET",
```

HTTP STATUS CODES:

200	Subscription details successfully retrieved
400	Bad request
401	Invalid Authorization token
403	Invalid API Key
404	Invalid Customer or Subscription ID

Assumptions:

- Each subscription corresponds to a product (or group of products)
- Subscriptions are created for line items upon order creation
 - Link to subscription will be added to LineItem
- New orders of a product with an existing subscription will not create a new subscription
 - Quantity will be added to the existing subscription
- currentQuantity is the current license quantity. May differ from ordered quantity while orders are processing
- usedQuantity is the license/consumable usage count.
- offer id is the base discount level offer id.
- renewalDate will be the customer's coterminDate for most subscriptions
 - Stock Credit Pack subscriptions will have a renewalDate as 1 year from order date

2. Get All Subscriptions for a Customer

GET /v3/customers/<customer-id>/subscriptions

HEADERS: [See Headers Section](#)

REQUEST BODY: None

RESPONSE BODY: Collection of Subscription resources

```
{
  "totalCount" : 2,
  "items" : [
    { Subscription resource },
    { Subscription resource }
  ]
}
```

HTTP STATUS CODES:

200	Successfully returned all subscriptions
400	Bad request
401	Invalid Authorization token
403	Invalid API Key
404	Invalid Customer ID

Assertions:

- Active subscriptions for VIP customers will be included
- No parameters/filters are implemented at this time

3. Update Subscription Auto-Renewal

PATCH /v3/customers/<customer-id>/subscriptions/<subscription-id>

HEADERS: [See Headers Section](#)

REQUEST BODY: updated autoRenewal dictionary

```
{
  "autoRenewal" : {
    "enabled" : true,
    "renewalQuantity" : 7
  }
}
```

RESPONSE BODY: Subscription resource

```
{ Subscription resource }
```

HTTP STATUS CODES:

200	AutoRenewal options updated
400	Bad request
401	Invalid Authorization token
403	Invalid API Key
404	Invalid Customer or Subscription ID

Assumptions:

- Enabled flag can only be set to true or false
- Setting enabled to true without the renewalQuantity field will mark all active licenses in the subscription for autorenewal
- Setting enabled to false with a renewalQuantity will disable autorenewal, the renewalQuantity will be ignored
- The autoRenewal preferences can only be updated for an active subscription
- The autoRenewal object is only evaluated at renewal time (the customer's cotermDate)
- If the renewalQuantity is higher than currentQuantity at renewal time, then the additional licenses will be included in renewal. If it is lower, then licenses will be removed at renewal.
- If the renewalQuantity has been explicitly set, the renewalQuantity will not be increased or decreased by additional orders or cancellations (until subscription becomes inactive)
- Three states of autorenewal:
 - Disabled
 - Enabled with renewal quantity
 - Enabled without explicit renewal quantity (all purchased licenses will be renewed. The renewalQuantity will still be returned with the number of purchased licenses)

Restrictions:

- Maximum renewalQuantity:
 - 10,000 for Team products
 - 200,000 for Enterprise products (not supported yet)

VIP to VIPMP Seamless Move (Transfers):

Existing customers can be transferred from VIP to VIPMP. Customer consent is implied by sending the customer's VIP Membership ID in the transfer request. The Preview Offers call is used to check a customer's eligibility for transfer and to preview the offers that would be transferred for their existing VIP subscriptions.

Seamless Move Eligibility Rules

Products

- Customer must have at least one active VIP product
- For a Customer, each existing VIP product must have an equivalent VIPMP product

Three-Year Commit (3YC)

- Customers with Three-Year Commit are only eligible for transfer if they have accepted 3YC after March 27th, 2023 or their 3YC end date is before their next anniversary date

Membership

- Customer must have a valid existing VIP Membership ID (indirect or direct)
- Customer admin must have accepted VIP Terms & Conditions

Renewal Window

- Customers within renewal window are now allowed to transfer to VIPMP
 - If Customer is in the renewal window prior to their anniversary date, they are only allowed to transfer if they have fully renewed

Region

- Customer must be in the list of [Supported Countries](#)

Reseller

- Reseller must have accepted the VIPMP Terms & Conditions
- Reseller must be approved to sell in the country of the customer
- Reseller must be enabled to sell into the customer's market segment

Compliance

- Customer must not have any unpaid VIP Purchase Orders
- Customer must not have any outstanding VIP Purchase Authorizations
 - This can be bypassed by passing an optional query parameter that will expire all outstanding Purchase Authorizations

Returns

- Customer must not have any orders that can be returned (as per VIP return policy)
 - This can be bypassed by passing an optional query parameter that will disable rollback to VIP

Note: Adobe reserves the right to disallow seamless move based solely on Adobe's discretion.

1. Preview Offers

GET /v3/memberships/<membership-id>/offers

HEADERS: [See Headers Section](#)

QUERY PARAMETERS:

param	values	default	description
ignore-order-return	true false	false	If true, customers with returnable purchases will be eligible for transfer.
expire-open-pas	true false	false	If true, customers with open Purchase Authorizations will be eligible for transfer.

REQUEST BODY: None

RESPONSE BODY:

```
{
  "totalCount" : 1,
  "items" : [
    {
      "offerId" : "12345678CA01A12",    // Offer ID for marketplace to use
      "quantity" : 10
      "renewalDate" : "2020-06-08"
    }
  ]
}
```

HTTP STATUS CODES:

200	Preview returned
400	Bad request
401	Invalid Authorization token
403	Invalid API Key
404	Invalid Membership ID

[Full set of transfer eligibility error codes](#)

Assumptions:

- Each item corresponds to one subscription
- The subscription is active until the renewalDate
 - All subscriptions will have auto-renewal enabled by default upon transfer (so they will renew on the renewalDate)
- If the customer is in the renewal window after their anniversary date and *has not renewed some products*, the non-renewed products will be included with renewalDate in the past. These items will be created as inactive subscriptions – the same behavior as if the customer had products in VIPMP that did not renew

2. Transfer Subscriptions**POST** /v3/memberships/<membership-id>/transfers**HEADERS:** [See Headers Section](#)**QUERY PARAMETERS:**

param	values	default	description
ignore-order-return	true false	false	If true, customers with returnable purchases can be transferred. Note: setting to true will disable rollback to VIP and the “returnable” purchase can no longer be returned.
expire-open-pas	true false	false	If true, customers with open Purchase Authorizations can be transferred. Any open Pas will be expired during the async portion of the transfer.

REQUEST BODY:

```
{
  "resellerId" : "999888777"
}
```

RESPONSE BODY:

```
{
  "transferId" : "5555luaigdfads555",
  "customerId" : "",
  "membershipId" : "12345678",
  "resellerId" : "999888777",
  "creationDate" : "2019-12-10T22:49:55Z",
  "status" : "1002",
  "lineItems" : [
    {
      "lineItemNumber" : 1,
      "offerId" : "12345678CA01A12",
      "quantity" : 10,
      "subscriptionId" : ""
    }
  ],
  "links" : {
    "self" : {
      "uri" : "/v3/memberships/12345678/transfers/5555luaigdfads555",
      "method" : "GET",
      "headers" : []
    }
  }
}
```

HTTP STATUS CODES:

202	Transfer request received/initiated
400	Bad request
401	Invalid Authorization token
403	Invalid API Key
404	Invalid Membership ID / Invalid Reseller ID

[Full set of transfer eligibility error codes](#)

3. Get Transfer Details

GET /v3/memberships/<membership-id>/transfers/<transfer-id>

HEADERS: [See Headers Section](#)

REQUEST BODY: None

RESPONSE BODY: Transfer resource

```
{
  "transferId" : "5555luaigdfads555",
  "customerId" : "10008675",
  "membershipId" : "12345678",
  "resellerId" : "999888777",
  "creationDate" : "2019-12-10T22:49:55Z",
  "status" : "1000",
  "lineItems" : [
    {
      "lineItemNumber" : 1,
      "offerId" : "12345678CA01A12",
      "quantity" : 10,
      "subscriptionId" : "8675309"
    }
  ],
  "links" : {
    "self" : {
      "uri" : "/v3/memberships/12345678/transfers/5555luaigdfads555",
      "method" : "GET",
      "headers" : []
    }
  }
}
```

HTTP STATUS CODES:

202	Transfer request received/initiated
400	Bad request
401	Invalid Authorization token
403	Invalid API Key
404	Invalid Membership or Transfer ID

Appendix

1. Headers

X-Request-Id	A unique identifier for the call. The value should be reset for every single request. If this is not provided, then a request ID will be automatically generated. Using a duplicate request ID may return an error.
X-Correlation-Id	Required. A unique identifier for the call. This is to ensure idempotency. In the case of a timeout, the retry call could include the same value. Upon receiving some response, the value should be reset for the next call.
Accept	Required. Specifies the response type. Must be "application/json" for proper usage.
Content-Type	Required. Specifies the request type. Must be "application/json" for proper usage.
Authorization	Required. Authorization token in the form "Bearer <token>"
X-API-Key	Required. The API Key for your integration

2. Idempotency – Correlation ID Header

The **X-Correlation-Id** is expected for all requests to the API. Where **X-Request-Id** should be unique per request, **X-Correlation-Id** should be unique per *intent*. Intent in this case means the data in the request and the attempted action.

- This means that upon receiving some response from Adobe (success or failure), the intent has been satisfied and the Correlation ID should not be used again.
- However, if no response is received from Adobe (in the case of a timeout), the same Correlation ID could be used again for the retry call (if the retry has the same intent).
- If a response was generated by Adobe for a given correlation ID, any subsequent requests to the same API with the same Correlation ID will receive that same response
- If a call fails with a 400 Bad Request error and you try again after fixing the data, the new call must have a new Correlation ID (as that would be a different intent).
- If the Correlation ID header is not included, a 400 error with status code **4119 – Correlation ID is Invalid or Missing** will be returned. See the Status Codes section below for more info.

Duplicate Request IDs:

- If the Correlation ID is duplicate, Request ID is not evaluated – the original response for that intent will be returned
- If the Correlation ID is different and Request ID is a duplicate, then a 400 error with status code **4120 – Duplicate request id, same requestId has already been processed** will be returned

3. HTTP Status Codes

2xx – Success

4xx – Error in client's request

5xx – Error in Adobe servers

200	OK	Everything worked as expected
201	Created	Resource has been successfully created
202	Accepted	Request has been accepted and will be acted on later
400	Bad Request	The request was unable to be completed due to some client error
401	Unauthorized	The authorization token is invalid
403	Forbidden	The API key is invalid
404	Not Found	The requested resource does not exist
429	Too Many Requests	Too many requests have been made in a brief period. Currently there is no limiting being done, so this status code should not appear.
500	Internal Server Error	There was an error on Adobe's end. Please try request again

4. Status Codes & Error Handling

Resource Status Codes

Status Code	Description	Applicable Resources
1000	Resource Status: Green Account - Active Order - Complete/Filled Subscription - Active	Reseller Account, Customer Account, Order, Subscription Transfer
1002	Resource Status: Yellow Account - Pending Order - Open Subscription - Pending	Reseller Account, Customer Account, Order, Subscription Transfer
1004	Resource Status: Red Account - Inactive Order - Failed Subscription - Inactive If there is another status code that describes the status better, that code will be used instead.	Reseller Account, Customer Account, Order, Subscription Transfer
1008	Order Status: Cancelled	Order
1010	Account Status: Inactive Reason: Invalid Address	Reseller Account, Customer Account
1012	Account Status: Inactive Reason: Account is blocked	Reseller Account, Customer Account
1014	Account Status: Inactive Reason: Customer already exists with the same Company Name and primary admin	Customer Account
1020	Order Status: Failed Reason: Distributor is inactive	Order

Status Code	Description	Applicable Resources
1022	Order Status: Failed Reason: Reseller is inactive	Order
1024	Order Status: Failed Reason: Customer is inactive	Order
1026	Order Status: Failed Reason: Customer ID is invalid	Order

Error Handling

Any Commerce Partner API status response with a **4xx** or **5xx** HTTP status code includes an error message consisting of a code field containing the HTTP response code, and a message field describing the error. Below is an example error response:

RESPONSE BODY:

```
{
  "code" : "1117",
  "message" : "Some Fields are Invalid"
  "additionalDetails" : [
    "companyProfile.companyName",
    "companyProfile.contacts[0].firstName"
  ]
}
```

Some error responses will include additionalDetails, an array of strings with more information on the specific error. These strings *may include* the names of fields with errors, line-item numbers, more specific reasoning, or other information.

Error Code	Message	Applicable API calls	HTTP Status Code
1114	Invalid Distributor	Create Reseller Account	400
1115	Invalid Reseller	Create Customer Account, Get Reseller Account, Update Reseller Account	404

Error Code	Message	Applicable API calls	HTTP Status Code
1116	Invalid Customer	Get Customer Account, Update Customer Account, Get Order History, Get All Subscriptions	404
1117	Some fields are invalid For debugging purposes, the names of the fields will be included in the invalidFields array in the response body.	Create Reseller, Update Reseller, Create Customer, Update Customer, Create Order, Update Subscription Auto-Renewal	400
1118	Invalid Address Note: If address and other fields are invalid, then 1117 will be used	Create Reseller, Update Reseller, Create Customer, Update Customer	400
1119	Some fields are not editable	Update Customer, Update Reseller, Cancel Order, Update Subscription Auto-Renewal	400
1121	Request contains additional unexpected fields	All	400
1122	Request is missing required fields	All (except for GETs)	400
1123	Duplicate externalReferenceId	Deprecated	400
1124	Internal Server Error. Please try again later	All	500
1125	Account cannot be updated while status is pending or blocked	Update Reseller, Update Customer	400

Error Code	Message	Applicable API calls	HTTP Status Code
1126	Customer cannot be created because reseller account is inactive	Create Customer	400
1127	Account already exists	Create Customer	400
1132	Invalid value for search parameter	Get Order History (V3)	400
1133	Offset out of range	Get Order History (V3)	400
2115	Invalid Customer or Order ID	Get Order Details, Cancel Order	404
2116	Invalid status	Cancel Order	400
2117	Cancellation window has closed	Cancel Order	400
2118	Order has already been cancelled	Cancel Order	400
2119	Too many line items	Preview Order, Create Order	400
2120	Line item quantity out of range	Preview Order, Create Order	400
2121	Duplicate line item numbers	Preview Order, Create Order	400
2122	Line item Offer ID is invalid	Preview Order, Create Order	400

Error Code	Message	Applicable API calls	HTTP Status Code
2123	extLineItemNumber out of range	Preview Order, Create Order	400
2124	Distributor not allowed to sell the program	Preview Order, Create Order	400
2125	Distributor not allowed to sell in the currency for the region	Preview Order, Create Order	400
2126	externalReferenceId exceeds maximum character limit	Create Order, Create Reseller , Create Customer	400
2128	Currency is not valid for Offer ID	Preview Order, Create Order	400
2129	Customer is not eligible to purchase Offer ID at line item: #	Create Order	400
2130	Line item Offer ID does not match original order	Create Return Order (V3)	400
2131	extLineItemNumber does not match original order	Create Return Order (V3)	400
2132	Line item quantity does not match original order	Create Return Order (V3)	400
2133	Line item has already been returned	Create Return Order (V3)	400
2134	Line item cannot be returned as the order already expired	Create Return Order (V3)	400
2135	Invalid market segment used for customer	Create Reseller Update Reseller Create Customer Update Customer	400

Error Code	Message	Applicable API calls	HTTP Status Code
2136	Please review the renewal settings. Auto-renewal needs to be turned on for at least 1 quantity	Create PREVIEW_RENEWAL Order	400
3115	Invalid Customer or Subscription ID	Get Subscription Details, Update Subscription Auto-Renewal	404
3116	renewalQuantity out of range	Update Subscription Auto-Renewal	400
3117	currentQuantity out of range		400
3118	orderQuantity out of range		400
3119	Inactive Subscription is not Editable	Update Subscription Auto-Renewal	400
3120	Update could not be performed because it would create an invalid renewal order	Update Subscription Auto-Renewal	400
4115	API key is invalid or missing	All [Adobe I/O]	403
4116	Authorization token is invalid	All [Adobe I/O]	401
4117	Authorization token is missing	All [Adobe I/O]	403
4118	Too many requests	All [Adobe I/O]	429
4119	Correlation ID is Invalid or Missing	All	400
4120	Duplicate request id, same requestId has already been processed	All	400
5115	Invalid Membership ID	Preview Offers, Transfer Subscriptions	400

Error Code	Message	Applicable API calls	HTTP Status Code
5116	Invalid Membership or Transfer ID	Get Transfer Details	400
5117	Customer is not eligible for transfer (see table below for REASON_CODE list)	Preview Offers, Transfer Subscriptions	400
5118	Customer has already been transferred	Preview Offers, Transfer Subscriptions	400
5119	Customer cannot be transferred because reseller account is inactive	Transfer Subscriptions	400
5120	Customer cannot be transferred because there are no admin contacts	Preview Offers, Transfer Subscriptions	400
5121	Transfer currently in progress for this customer	Preview Offers, Transfer Subscriptions	400

2129 Ineligible Purchase REASON_CODE List

These REASON_CODE values will be included in the additionalDetails array for 2129 errors.

REASON_CODE	Description	Action Required
INELIGIBLE_SWITCH	The order includes products that are incompatible with each other or with a product the customer already has.	Try removing the incompatible line item and placing the order again
INELIGIBLE_ADD_ON	Customer is attempting to purchase an add-on product without the base product.	Try placing the order again with the base product included
INELIGIBLE_CONSUMED	RETURN order only: customer is attempting to return a consumable product that has already been consumed.	Order cannot be returned if the transactions are consumed

REASON_CODE	Description	Action Required
INELIGIBLE_MARKET_SEGMENT	The order includes an item that belongs to a market segment that either the reseller or the customer do not belong to.	Ensure the correct offer is used for the customer and reseller's market segment.

5117 Ineligible Transfer REASON_CODE List

These REASON_CODE values will be included in the additionalDetails array for 5117 errors.

REASON_CODE	Description	Action Required
INVALID_COUNTRY	Customer is not in a valid country for this distributor	Try a distributor in a different region or wait until the distributor supports customer's country.
BAD_MARKET_SEGMENT	Customer's market segment is either not supported for this distributor or for VIPMP	Try a different distributor that supports customer's market segment or wait for a VIPMP change to allow the market segment.
INACTIVE	VIP Account is inactive	Customer/reseller must reactivate VIP account.
EXTENDED_TERM_3YC	Customer's contract term length is not supported by VIPMP	Wait for terms to be supported by VIPMP or wait until current term ends.
OPEN_PURCHASE_AUTH	Customer has open purchase authorization(s)	Open PAs must be cancelled/returned (if possible) or converted to order and paid for. Can also be bypassed with <i>expire-open-pas</i> query parameter
CONTRACT_NOT_ACCEPTED	Customer's VIP contract has not been accepted	Customer must set up an administrator to accept VIP contract.
IN_WINDOW_PARTIAL_RENEWAL	Customer is in the renewal window prior to their anniversary date and they have partially renewed.	Customer must either renew all active licenses or wait until their anniversary date.

REASON_CODE	Description	Action Required
RETURNABLE_PURCHASE	Customer has order(s) still in the return window	Wait until return window has closed (14 days after last order). Can also be bypassed with <i>ignore-order-return</i> query parameter.
NO_TARGET_SKU	Customer has product(s) that are not valid VIPMP product(s)	Wait for products to be supported in VIPMP.
UNBILLED_ORDER	Customer has unbilled order items	Customer must pay for all orders. If they have not received an invoice, then the customer needs to contact Adobe.
UNPAID_INVOICE	Customer has open/unpaid invoice(s)	Customer must pay for open invoice(s).

5. Supported Countries and Locales

Supported/Unsupported Countries

All [ISO 3166-1 2-character country codes](#) are supported, except for the following table. For the most up to date list of supported countries and their postal code regexes, please visit the [Partner Sandbox Portal](#).

ISO Code	Country Name	Unsupported Reason/Remediation
AQ	Antarctica	No population
AX	Åland Islands	Use country FI with region 01
BL	Saint Barthélemy	Use country FR with region BL
BQ	BES Isalnds	Use country NL with region BQ1, BQ2, or BQ3
BV	Bouvet Island	No population
CU	Cuba	Embargoed
CW	Curacao	Use country NL with region CW
EH	Western Sahara	
GG	Guernsey	Use country GB
GS	South Georgia and The South Sandwich Islands	
HM	Heard Island and McDonald Islands	No population
IM	Isle of Man	Use country GB
IO	British Indian Ocean Territory	Use country GB
IR	Iran	Embargoed

ISO Code	Country Name	Unsupported Reason/Remediation
JE	Jersey	Use country GB
KM	Comoros	
KP	North Korea	Embargoed
MF	Saint Martin (French part)	Use country FR
PN	Pitcairn	
PW	Republic of Palau	
SD	Sudan	Embargoed
SJ	Svalbard and Jan Mayen Islands	
SS	South Sudan	
SY	Syria	Embargoed
TF	French Southern Territories	Use country FR

Supported Locales (Preferred Language)

The following table has a list of all supported locales for the **preferredLanguage** field of Reseller and Customer Account creation.

Locale
bg-BG
cs-CZ
da-DK

Locale
de-DE
en-AE
en-GB
en-IL
en-US
en-XM
es-ES
es-LA
es-MX
es-NA
et-EE
fi-FI
fr-CA
fr-FR
fr-MA
fr-XM
hr-HR
hu-HU
it-IT
ja-JP

Locale
ko-KR
lt-LT
lv-LV
nb-NO
nl-NL
pl-PL
pt-BR
ro-RO
ru-RU
sk-SK
sl-SI
sr-RS
sr-SP
sv-SE
tr-TR
uk-UA
zh-CN
zh-HANS
zh-HANT
zh-TW

6. Resources and Fields

Please see the [Validations/Regular Expressions](#) section for any regular expressions for specific fields.

Reseller (top-level resource)

Property	Type	Description	Range/Limits
externalReferenceId	string Optional	Marketplace's ID for reseller. ID does not need to be unique.	Max: 35 characters
distributorId	string	Adobe-generated unique ID for distributor tied to this reseller	Max: 40 characters
resellerId (read only)	string	Unique ID for reseller created upon account creation	Max: 40 characters
companyProfile	CompanyProfile resource	Information about the reseller	
creationDate (read only)	String (datetime)	Date and time of account creation in UTC	
status (read only)	string	Status code for the reseller account	4 characters
links (read only)	Links resource	Deep links to get customer account details	

Customer (top-level resource)

Property	Type	Description	Range/Limits
externalReferenceId	string Optional	Marketplace's ID for customer. ID does not need to be unique.	Max: 35 characters
customerId (read only)	string	Unique ID for customer created upon account creation	Max: 40 characters
resellerId	string	ID of reseller tied to customer	Max: 40 characters
companyProfile	CompanyProfile resource	Information about the customer	
cotermDate (read only)	String (date)	Date that renewal order is to be placed. Should be one year after first order is provisioned (if a 1-yr term) and gets updated upon each renewal order.	10 characters
creationDate (read only)	String (datetime)	Date and time of account creation in UTC	
status (read only)	string	Status code of customer account	4 characters
links (read only)	Links resource	Deep links to get customer account details	

CompanyProfile

Property	Type	Description	Range/Limits
companyName	string	Name of company	Min: 4 characters Max: 80 characters
preferredLanguage	string	Preferred language for company (IETF hyphenated language tag)	Max: 40 characters
address	Address resource	Address of company	
contacts	Array of Contact resources	Points of contact for admin console	Min: One contact No limit

Address

Property	Type	Description	Range/Limits
country	string	Country of address (ISO 3166 standard)	2 or 3 characters
region	string	State/region of the address (ISO 3166-2 standard – only send code after hyphen). Ex. US-CA → CA	Max: 255 characters
city	string	City	Max: 40 characters
addressLine1	string	Line 1 of the address	Max: 60 characters
addressLine2	string	Optional. Line 2 of the address	Max: 60 characters
postalCode	string	Zip code/postal code of address. Must match city/region/country.	Max: 40 characters
phoneNumber	string	Optional. Phone number associated with this address	Max: 40 characters

Contact

Property	Type	Description	Range/Limits
email	string	Email address of the contact	Max: 240 characters
firstName	string	First name of the contact	Max: 35 characters
lastName	string	Last name of the contact	Max: 35 characters
phoneNumber	string	Optional. Phone number of the contact	Max: 40 characters

Order (top-level resource)

Property	Type	Description	Range/Limits
externalReferenceId	string Optional	Reconciliation ID for this order (ex. PO Number, Order ID, etc.). ID does not need to be unique.	Max: 35 characters
orderId (read only)	string	Unique identifier for order that is created upon order creation	Max: 40 characters
customerId (read only)	string	ID of the customer that placed this order	Max: 40 characters
referenceOrderId	string required for RETURN orders	The ID of the order being returned in a RETURN order.	Max: 40 characters
orderType	String (enum)	Type of the order. Valid values: NEW, PREVIEW, RETURN, TRANSFER, RENEWAL	
lineItems	Array of LineItem resources	Itemized list of the offers and their quantity for this order	Min: 1 item Max: 499 items
currencyCode	string	Currency used for placing the order (ISO 4217 format).	3 characters
creationDate (read only)	String (datetime)	Date and time the order was created in UTC	
status (read only, except for cancelling)	string	Status code of the order	4 characters
links (read only)	Links resource	Deep links to get order details	

LineItem

Property	Type	Description	Range/Limits
extLineItemNumber	integer	Marketplace's unique index for item in order	Min: 0 Max: 999999 Must be unique
offerId	string	The ID of the offer being purchased	
quantity	integer	Quantity of this offer purchased in this order	Min: 1 Max: 10,000 for Team / 599,000 for Enterprise 10,000 for
subscriptionId (read only)	string	ID of the Subscription resource associated with this line item.	Max: 40 characters
status (read only)	string	Status code of the line item	4 characters

Subscription (top-level resource)

Property	Type	Description	Range/Limits
subscriptionId (read only)	string	Unique identifier of the subscription resource	Max: 40 characters
currentQuantity (read only)	integer	Total quantity of the subscription that a customer has	Max: 10,000 for Team / 599,000 for Enterprise
usedQuantity	integer	Total used quantity for the license or the consumable	
offerId	string	The base discount level offer id	
autoRenewal	AutoRenewal object	Resource to manage auto-renewal flag and quantity	
creationDate (read only)	String (datetime)	Date and time of subscription creation in UTC	
status (read only)	string	Current status code of the subscription	4 characters
links (read only)	Links resource	Deep links to get subscription details	

AutoRenewal

Property	Type	Description	Range/Limits
enabled	boolean	Boolean flag denoting whether or not autoRenewal is enabled	true or false
renewalQuantity	integer	Optional. Total quantity to be renewed at the end of the contract period (can be higher than the current subscription quantity)	Min: 1 Max: 10,000 for Team / 599,000 for Enterprise

Links

Property	Type	Description	Range/Limits
self	Link object	Link to get data about this resource	
next	Link object	Link to get data about the next resource	
prev	Link object	Link to get data about the previous resource	

Link

Property	Type	Description	Range/Limits
uri	string	The URI to access this link	
method	string	The method used for the URI	
headers	Array of Key:Value pairs	The headers for the link. Note: Any headers in this are in addition to the required headers specified in the Headers section above.	

7. Validations/Regular Expressions

Field Name	Resource/Object	Regular Expression (Java String)	Notes
companyName	CompanyProfile	<code>^[\p{L}\p{N}_ ,.&&'\'() \\\\"/> </code>	Full Unicode letter and number support – no normalization is done.
firstName	Contact	<code>^[\p{L}\p{N}-_ ,.&&'\'"]{1,35}\$</code>	Full Unicode letter and number support – no normalization is done.
lastName	Contact	<code>^[\p{L}\p{N}-_ ,.&&'\'"]{1,35}\$</code>	Full Unicode letter and number support – no normalization is done.
email	Contact	<code>(?:[a-zA-Z0-9!#\$%&'*/+=?^_`{ }~-]+(?:\.(?:[a-zA-Z0-9!#\$%&'*/+=?^_`{ }~-]+)* \"(?:[\x01-\x08\x0b\x0c\x0e-\x1f\x21\x23-\x5b\x5d-\x7f] \\[\x01-\x09\x0b\x0c\x0e-\x7f])*\")@(?:(?:[a-zA-Z0-9](?:[a-zA-Z0-9-]*[a-zA-Z0-9])?\. (?:(?:25[0-5] 2[0-4][0-9] [01]?[0-9][0-9]?)\.){3}(?:25[0-5] 2[0-4][0-9] [01]?[0-9][0-9]?) [a-zA-Z0-9-]*[a-zA-Z0-9]:(?:[\x01-\x08\x0b\x0c\x0e-\x1f\x21-\x5a\x53-\x7f] \\[\x01-\x09\x0b\x0c\x0e-\x7f])+)\\))</code>	Source: emailregex.com
postalCode	Address	Varies based on country. Please see the Supported Countries and Locales section	

8. Support

Contact Support

- For development and integration support, please contact partnerapi-integration-support@adobe.com
- For production API support, please contact partnerapi-support@adobe.com

Reference Links

- Additional details about IMS [access tokens](#)
- Additional details about the [JWT authentication workflow](#)
- Explore more tutorials provided by Adobe I/O [here](#)
- JWT libraries and SDKs can be found [here](#)