PRISM API
GETTING STARTED

TABLE OF CONTENTS

DOCUMENT CONTROL

VERSION	AUTHOR	DESCRIPTION	DATE
0.1	Cameron McFie	Initial Draft	1 June 2016
1.0	Cameron McFie	Final	2 June 2016
1.1	Cameron McFie	Update to support V2 API Draft	24 January 2017
1.1.1	Cameron McFie	Update to Creating new order in Microsoft CSP	3 April 2017
2.0	Cameron McFie	Update to show version 2 API calls for Microsoft CSP	26 April 2017
2.1	Cameron McFie	Update to Microsoft Tenant Creation call to match updates in Production	29 May 2017
2.2	Cameron McFie	Update to include Azure Usage APIs	8 November 2017
2.3	Cameron McFie	Updated to use PRISM name instead of Prime	19 March 2018
2.3.1	Patrick Johannessen	Update product name and URLs	6 April 2018
2.3.2	Vlad Mouratch	Update styling and links	21 September 2018
2.4	Vlad Mouratch	Added MCA consent section	26 September 2018
2.4.1	Hana Yoo	Updated styling	12 November 2018
2.4.2	Vlad Mouratch	Added Annual Microsoft CSP order provisioning	23 November 2018
2.4.3	Al Borns	MCuA update	05 December 2019
2.4.4	Vanessa Loayza	Format updates	05 December 2019

BACKGROUND

rhipe is proud to have built a new self-service portal called "PRISM". PRISM has been developed from ground up with the partner in mind. The idea of this portal is provide our partners a single pane of glass to manage every program they have with rhipe.

The portal has been built with an API first strategy so that every feature that is added to the portal is added to our APIs first and then the GUI is built after.

This document is to explain each of the steps involved in getting started with the PRISM APIs.

PRISM GLOSSARY OF TERMS

Before delving into detail on the PRISM APIs you will need to understand some terms used in the PRISM platform to understand the design of the PRISM API's.

Customer

PRISM has been built to store a tiered hierarchy of customers. Each customer has one parent and can have any number of children. The hierarchy is used for both securing access and for billing.

A Customer is an entity that could potentially sign up for a service. Example Customers are rhipe, a partner of rhipe's, or a customer of one of rhipe's partners.

Program

rhipe has many programs that our partners can use to sign up. Examples of these are:

- Microsoft SPLA
- Microsoft CSP
- Trend Micro
- VMWare
- Veeam
- Citrix
- RedHat
- rhipe Cloud Support
- And many others…

We have both Core programs and Add-on programs. Add-on programs are smaller programs that are directly linked with a parent program. An example of this is rhipe Cloud Support is an Add-on program for Microsoft CSP. All billing for rhipe Cloud Support is tied to existing CSP subscriptions.

Product

A product is a purchasable item within a program. Each product has a stock keeping unit (SKU), product name, unit type (e.g. per user), and other attributes discuss in further detail below.

Contract Agreement

A contract agreement is an agreement between rhipe and one of their partners allowing partners to resell a program to their customers.

Tenant

A tenant is a record of an end-customer purchasing a program.

Subscription

A subscription is a record of a product being purchased by an end-customer

REQUIREMENTS FOR ACCESS TO THE API

Before accessing the PRISM APIs the following requirements need to be met:

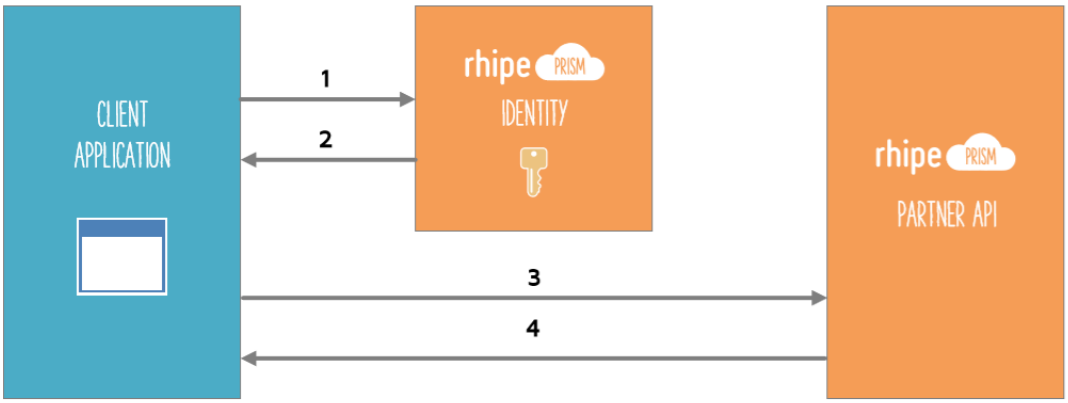
- You are a signed partner with rhipe
- You have signed up to the program you wish to use the APIs for
- You have a development team that are skilled in using the OAuth2 Client Credential flow and accessing REST APIs

AUTHENTICATION

The PRISM APIs have been secured using OAuth 2.0 Client Credentials Grant Flow.

The OAuth 2.0 Client Credentials Grant Flow permits a web service (a *confidential client*) to use its own credentials to authenticate when calling another web service, instead of impersonating a user. In this scenario, the client is typically a middle-tier web service, a daemon service, or web site.

Client Credentials Grant Flow Diagram



1. The client application authenticates to the PRISM identity token issuance endpoint and requests an access token
2. The PRISM identity token issuance endpoint issues the access token
3. The access token is used to access the secured PRISM Partner API
4. Data from the PRISM Partner API is returned to the client application

Requesting a Client Id and Client Secret from rhipe

To request access to the PRISM API's simply go into PRISM Portal and click into the "Customer Management" section. Against your Customer (should be the first on the list) click View. On the "Customer Details" page you will see a "Manage Client Credentials" button where you can request a client id and secret.

Requesting an Access Token

To request an access token, use an HTTP POST to the tenant-specific Azure AD endpoint.

```
https://identity.prismportal.online/core/connect/token
```

Service-to-Service Access Token Request

A service-to-service access token request contains the following parameters. All parameters are required.

PARAMETER	DESCRIPTION
grant_type	[Required] Specifies the requested response type. In a Client Credentials Grant flow, the value must be **client_credentials**.
client_id	[Required] Specifies the client id of the calling application. The client id is provided to you by rhipe
client_secret	[Required] Enter a key registered for the calling application. The client secret is provided to you by rhipe
scope	[Required] Specifies the scope of the credential. In all cases the value must be **rhipeapi**.

Example

The following HTTP POST requests an access token for the PRISM API. The client_id identifies the client application that requests the access token.

```
POST /core/connect/token HTTP/1.1
Host: identity.prismportal.online
Content-Type: application/x-www-form-urlencoded

grant_type=client_credentials&client_id=625bc9f6-3bf6-4b6d-94ba-
e97cf07a22de&client_secret=qkDwDJlDfig2lpeuUZYKH1Wb8q1V0ju6slLxQQqhJ+s=&scope=rhip
eapi
```

Service-to-Service Access Token Response

A success response contains a JSON OAuth 2.0 response with the following parameters.

PARAMETER	DESCRIPTION
access_token	The requested access token. The calling client application can use this token to authenticate to the PRISM API
token_type	Indicates the token type value. The only type that PRISM Identity supports is **Bearer**
expires_in	How long the access token is valid (in seconds).

Example

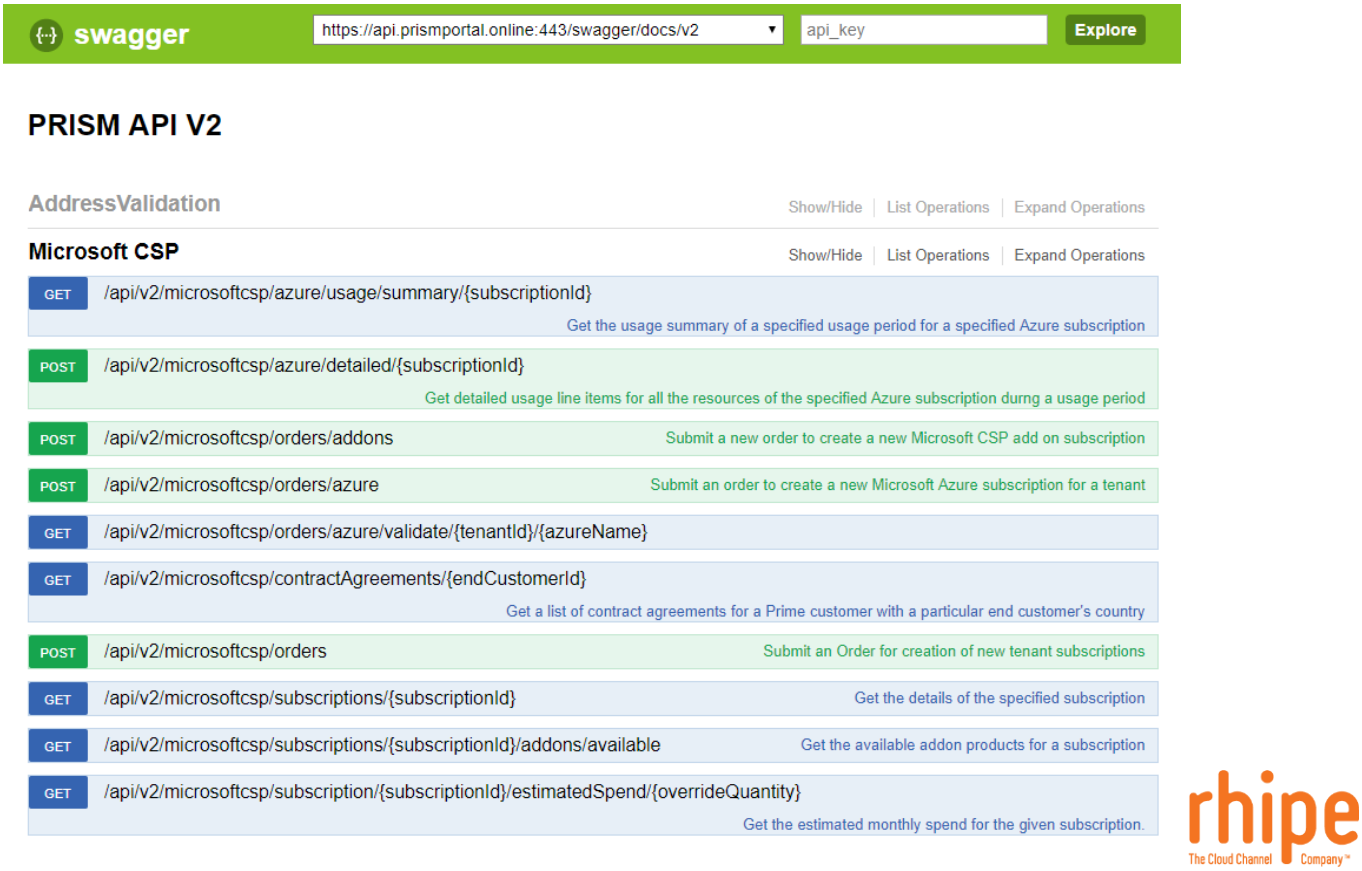
The following example shows a success response to a request for an access token to a web service.

```
{
    "access_token": "02791d1ff4e1e6afd93376f417d605fe",
    "expires_in": 3600,
    "token_type": "Bearer",
}
```

DOCUMENTATION

Swagger documentation of the APIs are available at https://api.prismportal.online/swagger/ui/index.

To see the Swagger documentation for V2 of the API you can change the URL in the Swagger interface to have V2 at the end.



ACCESSING THE PRISM API

The PRISM API is available at the following URL:

```
https://api.prismportal.online/api
```

Authorization Header

In every HTTP request you make to the API you need to include the following header:

```
Authorization: Bearer <access token received from PRISM Identity>
```

Please note: The word "Bearer" is case sensitive.

API Versioning

In every call you need to specify the version of the API you wish to call as part of the API URL. An example of this is as follows

```
https://api.prismportal.online/api/v2
```

PartnerCustomerId Header

If you are a multi-tier reseller you may need to impersonate as another customer to perform actions on their behalf. To do this simply add the following HTTP request header:

```
Prime-PartnerCustomerId: <GUID of the customer you wish to impersonate as>
```

PREPARING YOUR ENVIRONMENT

Before using the PRISM API to manage programs we suggest using the PRISM API to populate your database with the following content:

1. A list of Programs
2. A list of Contract Agreements
3. A list of Products and their Pricing
4. A list of Existing Customers
5. A list of Customers that have ordered from a Program

Retrieving a List of Programs

To retrieve a list of the Programs you have access to perform a HTTP GET request as follows

GET /api/v1/me/programs

Response Class


```
[
  {
    "programId": "string",
    "programName": "string",
    "programShortName": "string",
    "vendorId": "string",
    "vendorName": "string",
    "customerHasAgreement": true,
    "rank": 0
  }
]
```

Retrieving a List of Contract Agreements

To retrieve a list of the Contract Agreements you have perform a HTTP GET request as follows.

GET /api/v2/contractagreements

Response Class


```
[
  {
    "customerId": "string",
    "contractAgreementId": "string",
    "agreementStartDate": "2017-03-14T03:55:42.289Z",
    "agreementEndDate": "2017-03-14T03:55:42.289Z",
    "agreementNumber": "string",
    "countryCode": "string",
    "countryFullName": "string",
    "sourceCurrencyCode": "string",
    "targetCurrencyCode": "string",
    "signedDate": "2017-03-14T03:55:42.289Z",
    "programId": "string",
    "status": "string",
    "programName": "string",
    "email": "string",
    "notificationEmailAddress": "string",
    "customerName": "string",
    "vendorProgramType": "string",
    "salesTerritory": "string",
    "microsoftMarketCategory": {
      "isCorporate": true,
      "isGovernment": true,
      "isAcademic": true
    }
  }
]
```

Retrieving a List of Products (Including Price) for a Tenant

To retrieve a list of the Products you have access to within a program and your prices for each product perform a HTTP POST request as follows.

Note: The {agreementId} is the contract agreement number associated with the program you are interested in.

GET /api/v2/microsoftcsp/tenants/{tenantId}/products/available

Response Class


```
{
  "programId": "string",
  "programName": "string",
  "countryFullName": "string",
  "currencyCode": "string",
  "productGroups": [
    {
      "groupName": "string",
      "products": [
        {
          "productId": "string",
          "productName": "string",
          "productGroup": "string",
          "productSku": "string",
          "price": 0,
          "minQty": 0,
          "maxQty": 0,
          "qty": 0,
          "unitPrice": 0,
          "productUnit": "string",
          "retailPrice": 0,
          "commitmentValue": 0,
          "tieredProductSku": "string",
          "productType": "string",
          "productDescription": "string",
          "productShortDescription": "string",
          "productRestrictions": "string"
        }
      ],
      "isExpanded": true,
      "order": 0
    }
  ],
  "showRetailPrice": true
}
```

Retrieving a List of Existing Customers

To retrieve a list of existing Customers directly associated with your Customer context (including your own Customer object) format a GET request as follows

GET /api/v1/me/customers

Response Class


```
[
  {
    "customerId": "string",
    "customerName": "string",
    "customerNotificationEmail": "string",
    "parentCustomerId": "string",
    "registrationNumber": "string",
    "signedWithRhipe": "2016-06-02T05:10:09.727Z",
    "webUrl": "string",
    "mainPhone": "string",
    "fax": "string",
    "street1": "string",
    "street2": "string",
    "street3": "string",
    "city": "string",
    "state": "string",
    "postcode": "string",
    "country": "string",
    "customerProgramAgreements": [
      {
        "id": "string",
        "programId": "string",
        "customerId": "string",
        "agreementStartDate": "2016-06-02T05:10:09.728Z",
        "agreementEndDate": "2016-06-02T05:10:09.728Z",
        "contractAgreementId": "string",
        "billingPeriod": "string",
        "programReferenceId": "string",
        "programReferenceLabel": "string",
        "programName": "string"
      }
    ]
  }
]
```

Retrieve a List of Microsoft CSP Tenants

GET /api/v2/microsoftcsp/tenants

Request parameter (optional if you want to get tenants for a specific customer)

```
    "endCustomerId": "string",
```

Response Class

```
{
  "items": [
    {
      "tenantId": "string",
      "tenantDomain": "string",
      "tenantStatus": "string",
      "customerId": "string",
      "customerName": "string",
      "contractAgreementId": "string",
      "programId": "string",
      "isProvisioning": true,
      "hasOrderPending": true
    }
  ]
}
```

MICROSOFT CSP MANAGEMENT

Order for Provisioning Resources in Microsoft CSP

In PRISM Portal this is the order of calls we make to provision resources in PRISM:

Creating New PRISM Customer

Check to see if the Address details are valid
POST /api/v2/customers/validate

Create the new PRISM Customer
POST /api/v2/customers

Creating New Microsoft CSP Tenant

Check to see if the .onmicrosoft domain name is valid
GET /api/v1/microsoftcsp/validate/domain/{domainprefix}

Create a new Microsoft CSP Tenant
POST /api/v2/microsoftcsp/tenants

Add a new CSP Subscription to a Microsoft CSP Tenant

Get a list of products that are able to be ordered for a particular subscription
GET /api/v2/microsoftcsp/tenants/{tenantId}/products/available

Order new subscription
POST /api/v2/microsoftcsp/orders

Add a new CSP Addon (e.g. Advanced Threat Protection) to a CSP Subscription

Get a list of addons available for a particular subscription
GET /api/v2/microsoftcsp/subscriptions/{subscriptionId}/addons/available

Add Add-on
POST /api/v2/microsoftcsp/orders/addons

Make Changes to a Subscription

Update subscription Qty
POST /api/v2/microsoftcsp/subscriptions/{SubscriptionId}/quantity/{newQuantity}

Suspend a subscription
POST /api/v2/microsoftcsp/subscriptions/{SubscriptionId}/suspend

Resume a subscription
POST /api/v2/microsoftcsp/subscriptions/{SubscriptionId}/resume

Add Support to a Tenant

Get a list of support plans available
GET /api/v2/microsoftcsp/tenants/{tenantId}/supportplans/available

Add a support plan to a tenant
POST /api/v2/microsoftcsp/tenants/{tenantId}/supportplans/

Creating a New PRISM Customer

The first step to adding an order for a new Microsoft CSP tenant is to create a customer in PRISM.

The items highlighted in yellow are required fields.

POST /api/v2/customers

Request Class

```
{
  "customerName": "string",
  "webUrl": "string",
  "mainPhone": "string",
  "fax": "string",
  "email": "string",
  "address": {
    "street1": "string",
    "street2": "string",
    "street3": "string",
    "city": "string",
    "state": "string",
    "postcode": "string",
    "country": "string"
  },
  "billingAddress": {
    "street1": "string",
    "street2": "string",
    "street3": "string",
    "city": "string",
    "state": "string",
    "postcode": "string",
    "country": "string"
  }
}
```

Creating a new Microsoft CSP Tenant

For Office 365, CRM Online and Microsoft Azure to work you need to create an Azure AD tenant for the new customer.

All the fields are required.

The endCustomerId is the customerId of the PRISM customer, the defaultDomainPrefix is the prefix for the .onmicrosoft.com domain and the contact details are the contact that Microsoft will use in the event of issues.

POST /api/v2/microsoftcsp/tenants

Request Class

```
{
  "defaultDomainPrefix": "string",
  "password": "string",
  "contractAgreementId": "string",
  "endCustomerId": "string",
  "contactFirstName": "string",
  "contactLastName": "string",
  "contactPhoneNumber": "string",
  "contactEmailAddress": "string",
  "microsoftMarketCategory": "Corporate"
}
```

Creating a New Microsoft CSP Order

Now that you have created a PRISM customer and an associated Microsoft CSP Tenant you can place an order for licenses against that tenant. You can create both monthly and annual CSP orders.

POST /api/v2/microsoftcsp/orders

Request Class

```
{
  "tenantId": "string",
  "products": [
    {
      "productId": "string",
      "qty": 0
    },
  "orderBillingCycle": "Monthly"
  ]
}
```

orderBillingCycle can take values of "Monthly" (default value) and "Annual". This parameter is optional.

An example to create an order for 10 monthly x Office 365 Enterprise E1's and 5 x Office 365 Business Premium for Microsoft CSP with tenant Id of 88B6571F-35AF-4A82-89B4-E0C75F7D7866 would be as follows (please note that orderBillingCycle parameter is omitted, meaning it will be monthly order):

POST /api/v2/orders

Request Class

```
{
  "tenantId": "88B6571F-35AF-4A82-89B4-E0C75F7D7866",
  "products": [
    {
      "productId": "91fd106f-4b2c-4938-95ac-f54f74e9a239",
      "qty": 10
    },
    {
      "productId": "031c9e47-4802-4248-838e-778fb1d2cc05",
      "qty": 5
    }
  }
```

Please note: In the Swagger documentation that it has different HTTP status codes in the response. As an example if accepted it will return 202 ACCEPTED with an order id and a job status link. You can then poll /api/v2/longrunningtasks to get updates on the long running job.

Getting Details of a Subscription

GET /api/v2/subscriptions/{subscriptionId}

Response Class


```
{
  "subscriptionId": "string",
  "programReferenceId": "string",
  "programReferenceLabel": "string",
  "productId": "string",
  "friendlyName": "string",
  "quantity": 0,
  "firstPurchased": "2017-01-23T23:45:21.730Z",
  "commitmentEndDate": "2017-01-23T23:45:21.730Z",
  "status": "string",
  "addonSubscriptions": [
    {
      "parentSubscriptionId": "string",
      "subscriptionId": "string",
      "programReferenceId": "string",
      "programReferenceLabel": "string",
      "productId": "string",
      "friendlyName": "string",
      "quantity": 0,
      "firstPurchased": "2017-01-23T23:45:21.730Z",
      "commitmentEndDate": "2017-01-23T23:45:21.730Z",
      "status": "string",
      "addonSubscriptions": [
        {}
      ],
      "resourceId": "string",
      "unit": "string",
      "displayTemplate": "string"
    }
  ],
  "resourceId": "string",
  "unit": "string",
  "displayTemplate": "string"
}
```

Changing License Quantity on a Subscription

To change the total number of licenses used by a subscription simply send a POST message as follows:

POST /api/v2/subscriptions/{subscriptionId}/quantity/{newQuantity}

Note: Please look at the Swagger documentation for HTTP status codes in the responses. Subscriptions may be locked if another provisioning job is in progress for that subscription.

Suspending a Subscription

To suspend a subscription simply send a POST message as follows:

POST /api/v2/subscriptions/{subscriptionId}/suspend

Note: Please look at the Swagger documentation for HTTP status codes in the responses. Subscriptions may be locked if another provisioning job is in progress for that subscription.

Resuming a Subscription

To resume a subscription from a suspended state simply send a POST message as follows:

POST /api/v2/subscriptions/{subscriptionId}/resume

Note: Please look at the Swagger documentation for HTTP status codes in the responses. Subscriptions may be locked if another provisioning job is in progress for that subscription.

Getting a List of Tenants and their Associated Subscriptions

To get a list of Microsoft CSP tenants and any subscriptions associated with them simply do a GET request as follows:

GET /api/v2/contractagreements/{contractAgreementId}/tenants

Response Class


```
[
  {
    "tenantId": "string",
    "programReferenceId": "string",
    "programReferenceLabel": "string",
    "contractAgreementId": "string",
    "programId": "string",
    "programName": "string",
    "vendorProgramType": "string",
    "customerId": "string",
    "customerName": "string",
    "subscriptions": [
      {
        "subscriptionId": "string",
        "programReferenceId": "string",
        "programReferenceLabel": "string",
        "productId": "string",
        "friendlyName": "string",
        "quantity": 0,
        "firstPurchased": "2017-01-23T23:45:21.709Z",
        "commitmentEndDate": "2017-01-23T23:45:21.709Z",
        "status": "string",
        "addonSubscriptions": [
          {
            "parentSubscriptionId": "string",
            "subscriptionId": "string",
            "programReferenceId": "string",
            "programReferenceLabel": "string",
            "productId": "string",
            "friendlyName": "string",
            "quantity": 0,
            "firstPurchased": "2017-01-23T23:45:21.709Z",
            "commitmentEndDate": "2017-01-23T23:45:21.709Z",
            "status": "string",
            "addonSubscriptions": [
              {}
            ],
            "resourceId": "string",
            "unit": "string",
            "displayTemplate": "string"
          }
        ],
        "resourceId": "string",
        "unit": "string",
        "displayTemplate": "string"
      }
```

```
    ],
    "isvAddons": [
      {
        "subscriptionId": "string",
        "programReferenceId": "string",
        "programReferenceLabel": "string",
        "productId": "string",
        "friendlyName": "string",
        "quantity": 0,
        "firstPurchased": "2017-01-23T23:45:21.709Z",
        "commitmentEndDate": "2017-01-23T23:45:21.709Z",
        "status": "string",
        "addonSubscriptions": [
          {
            "parentSubscriptionId": "string",
            "subscriptionId": "string",
            "programReferenceId": "string",
            "programReferenceLabel": "string",
            "productId": "string",
            "friendlyName": "string",
            "quantity": 0,
            "firstPurchased": "2017-01-23T23:45:21.709Z",
            "commitmentEndDate": "2017-01-23T23:45:21.709Z",
            "status": "string",
            "addonSubscriptions": [
              {}
            ],
            "resourceId": "string",
            "unit": "string",
            "displayTemplate": "string"
          }
        ],
        "resourceId": "string",
        "unit": "string",
        "displayTemplate": "string"
      }
    ],
    "partnerCustomerId": "string",
    "estimatedMonthlySpend": 0,
    "estimatedMonthlySpendCurrencyCode": "string"
  }
]
```

DEFINE A CUSTOMER AS A RESELLER OF A PROGRAM (3 Tiered)

Retrieving a List of Contract Agreements allow partner to define 3-tiered reseller

GET
/api/v1/{partnerCustomerId}/contractagreements/availableforreseller/{resellerCountryCode}/{resellerCustomerId}

Response Class


```
[
    {
      "CustomerId": "Guid",
      "ContractAgreementId": "Guid",
      "AgreementStartDate": "2016-06-15T02:55:12.737Z ",
      "AgreementEndDate": "2016-06-15T02:55:12.737Z ",
      "AgreementNumber": "string",
      "CountryCode": "string",
      "CountryFullName": "string",
      "SourceCurrencyCode": "string",
      "TargetCurrencyCode": "string",
      "SignedDate": "2016-06-15T02:55:12.737Z ",
      "ProgramId": "Guid",
      "Status": "string",
      "ProgramName": "string",
      "Email": "string",
      "NotificationEmailAddress": "string",
      "CustomerName": "string",
      "VendorProgramType": "string",
      "SalesTerritory": "string"
    },
    {
        ...
    }
]
```


Example

Partner A (customerId is 88B6571F-35AF-4A82-89B4-E0C75F7D7866) has Microsoft CSP, Veeam and SPLA programs signed, and they want to make their end customer B (customerId is B45EB400-933A-43EB-B3F0-84DA02D77F8E) as a reseller of any of those 3 programs and want to see if it possible.

To add an end customer as a reseller, we need to check if a program is available in the end customer's country and also if the end customer has already booked the program.

The request URL could be
GET api/v1/*88B6571F-35AF-4A82-89B4-E0C75F7D7866*/contractagreements/availableforreseller/au/*B45EB400-933A-43EB-B3F0-84DA02D77F8E*

If only CSP is allow for resell and it's only available in B's country, then the response only brings back A's contract agreement of Microsoft CSP program.

Define an end customer as a reseller of a program

POST
/api/v1/{partnerCustomerId}/contractagreement/{contractAgreementId}/reseller/{resellerCustomerId}/{resellerAgreementNumber}

No content in responses

Example

Partner A wants to add its end customer B as a reseller of Microsoft CSP program.

Partner A's customer Id is 88B6571F-35AF-4A82-89B4-E0C75F7D7866
End Customer B's customer Id is B45EB400-933A-43EB-B3F0-84DA02D77F8E, and they have a MPN ID as 46282194.

Partner A has an active contract agreement (ContractAgreementId is 6E837502-06FE-43BA-8AB4-2A00F5027472) for Microsoft CSP.

To add B as a reseller of Microsoft CSP program, the request URL should be like following

POST api/v1/*88B6571F-35AF-4A82-89B4-E0C75F7D7866*/contractagreement/*6E837502-06FE-43BA-8AB4-2A00F5027472*/reseller/*B45EB400-933A-43EB-B3F0-84DA02D77F8E*/*46282194*

The returned HTTP status code indicates whether the operation succeeded or not.

Microsoft Cloud & Customer Agreements

On the 1st of October Microsoft has introduced a new universally transferable and adaptive agreement called the Microsoft Customer Agreement (MCuA). This agreement will be the only agreement available after the 31st January 2020. To view the agreement, visit https://aka.ms/customeragreement

While the legacy Microsoft Cloud Agreement is still valid, this agreement will be phased out and the MCuA will need to be signed in order to create new tenants, increase/decrease seats or suspend subscriptions.

The PRISM API will accept both agreements until early January 2020, after this point only the Microsoft Customer Agreement will be accepted.

Certify customer consent to Microsoft Cloud Agreement (MCA)

The type of agreement is set in the **type** attribute.

Agreement Type	
MicrosoftCustomerAgreement	The newest agreement provided by Microsoft.
MicrosoftCloudAgreement	Legacy agreement **Note:** This agreement will no longer be accepted after January 2020. Any existing tenants who wish to transact with Microsoft will need to have accepted the MCuA

Certify customer consent to Microsoft Customer Agreement (MCuA)

POST /api/v2/microsoftcsp/tenants/{tenantId}/agreements

Request class:

```
{
  "type": "MicrosoftCustomerAgreement",
  "dateAgreed": "2018-09-24T03:54:42.404Z",
  "primaryContactEmail": "string",
  "primaryContactFirstName": "string",
  "primaryContactLastName": "string",
  "primaryContactPhoneNumber": "string"
}
```

Update a tenant from Microsoft Cloud Agreement to Microsoft Customer Agreement

To update a customers' acceptance to the new Microsoft Customer Agreement simply pass a new acceptance request as depicted above, ensuring MicrosoftCustomerAgreement is passed in the **type** attribute.

Get customer consent details

GET /api/v2/microsoftcsp/tenants/{tenantId}/agreements

Response class:


```
[
  {
    "id": "string",
    "tenantName": "string",
    "vendorTenantId": "string",
    "type": "MicrosoftCustomerAgreement",
    "dateAgreed": "2019-12-26T03:16:08.451Z",
    "primaryContactEmail": "string",
    "primaryContactFirstName": "string",
    "primaryContactLastName": "string",
    "primaryContactPhoneNumber": "string"
  }
]
```

AZURE USAGE

Getting Azure Usage Summary

GET /api/v2/microsoftcsp/azure/usage/summary/{subscriptionId}?startDate=2017-10-01T00:00:00.000Z&endDate=2017-11-09T00:00:00.000Z

Response Class


```
{
  "subscriptionId": "string",
  "subscriptionName": "string",
  "startDate": "2017-11-08T09:27:49.721Z",
  "endDate": "2017-11-08T09:27:49.721Z",
  "currency": "string",
  "totalCost": 0,
  "lastUpdateDate": "2017-11-08T09:27:49.721Z",
  "resourceUsageSummaries": [
    {
      "resourceId": "string",
      "resourceName": "string",
      "resourceCategory": "string",
      "resourceSubCategory": "string",
      "currency": "string",
      "cost": 0
    }
  ]
}
```


Example

Partner A wants to get the Azure spend to date for a particular Azure subscription for one of their customers.

The subscription Id of the subscription is D95231ED-856E-4645-AE22-B565A2A32628 and the partner wants to see the summarised spend from 1 October 2017 to 9 October 2017.

In this example the request URL should be like following

POST /api/v2/microsoftcsp/azure/usage/summary/d95231ed-856e-4645-ae22-b565a2a32628?startDate=2017-10-15T00:00:00.000Z&endDate=2017-11-09T00:00:00.000Z

Getting Azure Daily Detailed Usage Items

GET /api/v2/microsoftcsp/azure/detailed/{subscriptionId}

Request Class


```
{
  "startDate": "2017-10-01T00:00:00.000Z",
  "endDate": "2017-10-31T00:00:00.000Z"
}
```


Response Class


```
{
  "usageLineItems": [
    {
      "resourceId": "string",
      "usageDate": "2017-11-08T09:27:49.724Z",
      "customerName": "string",
      "advisorMpnId": "string",
      "billingMpnId": "string",
      "primaryDomain": "string",
      "subscriptionId": "string",
      "resourceGroup": "string",
      "instanceData": "string",
      "tags": "string",
      "meterCategory": "string",
      "meterSubCategory": "string",
      "meterName": "string",
      "meterRegion": "string",
      "quantity": 0,
      "unit": "string",
      "sourceCurrency": "string",
      "targetCurrency": "string",
      "rrp": 0,
      "cost": 0
    }
  ],
  "token": {
    "nextPartitionKey": "string",
    "nextRowKey": "string",
    "nextTableName": "string",
    "targetLocation": "Primary"
  }
}
```