



Quest Enterprise Suite Adequacy API Developer Guide

Last Updated: July 17, 2023

Table of Contents

Overview	2
QuickStart	2
Uploading Your Data to Quest	4
Uploading Accesspoints	5
Uploading Providers	6
Checking the Status of Uploads to a Network Version	10
Retrieving the Status of All Filtered Network Uploads	11
Retrieving Providers from a Filtered Network	11
Servicing Provider Report	13
Creating an HSD File Format for Upload to the HPMS Portal	16
Creating a Project Based on a Filtered Network	16
Project Management	20
Updating Project User Sharing	20
Getting a List of Existing Projects	21
Getting and Setting Project Information	22
Getting Project Status	23
Getting Project Ids by Network Ids	23
Network Management	24
Updating Filtered Network Sharing	24
Getting Provider Networks	25
Getting Provider Network Details	27
Getting Provider Network Version Details	28
Getting and Setting Provider Network Version Status	29

Overview

The QES Adequacy API Developer Guide explains how to programmatically determine whether the provider coverage for each specialty by county meets regulatory requirements. Maintaining the adequacy of your provider network reduces regulatory risk and maximizes member satisfaction.

For example, using the QES Adequacy API you can:

- Determine the counties in your network that fail to meet the regulatory guidelines.
- Export the network adequacy data in the health services delivery (HSD) file format required for upload to HPMS.

QuickStart

This QuickStart explains how to:

- Obtain an authorization token so you can access the Adequacy API endpoints.
- Generate the Specialty Summary by County report to determine which counties do and do not have adequate coverage for each specialty.
- Generate the Servicing Provider Report, with an option to output in HSD format for upload to HPMS.

1. Obtain an authorization (access) token:

- a. Request a client id and client secret from Quest Analytics Client Services.
- b. Submit a POST request to the following endpoint, passing the client id and client secret in the request body. For example:

```
POST /token HTTP/1.1
```

```
Host: https://uat-api.questanalytics.com/enterprise/auth
```

```
Content-Type: application/json
```

```
Subscription-Key: {{subscriptionKey}}
```

```
Content-Length: 145
```

```
{
  "client_id": "{{clientId}}",
  "client_secret": "{{clientSecret}}",
  "scope": "default",
  "grant_type": "client_credentials"
}
```

The request returns a large authorization token (approximately 1800 characters).

- ### 2. Retrieve the Adequacy Summary Results to find out which counties will pass and which will fail when you upload your adequacy data to the government. To retrieve the report, submit a GET request to the following endpoint. The request takes the ProjectId in the URL and the authorization token in the request body. For example:

```
GET /enterprise/proj-mgmt/v3/Projects/{{projectId}}/adequacy/results/summaries
HTTP/1.1
```

Host: <https://uat-api.questanalytics.com>

Authorization: Bearer {{accessToken}}

This request returns a data array which contains one element for each specialty in each county. So if there are 100 specialties and 40 counties, the array contains 4000 data elements. Each data element consists of standard fields defined by Quest and custom fields (cargo columns) defined by the client. For example, here is the data element for the cardiology specialty in Anderson county in Tennessee:

```
{
  "State": "Tennessee",
  "County": "44000 - Anderson",
  "CountyStatus": "Active",
  "CountyClass": "M",
  "MembershipName": "Primary Membership",
  "FullMemberCount": 570,
  "SpecialtyGroupCode": "008",
  "SpecialtyGroup": "Cardiology",
  "FullMemberCountWithAccess": 450,
  "MemberAccessPercentage": "79",
  "FullMemberCountWithoutAccess": 120,
  "MemberWithoutAccessPercentage": "21",
  "AccessMetCriteria": "N",
  "AccessCustomization": "N",
  "AccessTelehealthCredit": "",
  "ConAccessCredit": "",
  "AverageCoveringDistance": "---",
  "MinimumServicingProviders": 2,
  "InRegionProviderCount": 1.0,
  "DistinctServicingProviderCount": 1.0,
  "ServicingMetCriteria": "N",
  "OverallMetCriteria": "N",
  "GapAgeDays": "---",
  "MarketMemberAccessPercentage": "---",
  "MarketInRegionProviderCount": "---",
  "MarketDistinctServicingProviderCount": "---",
  "MarketPercentage": "---",
  "ImpactIndicator": ""
}
```

}

A county has adequate coverage for a specialty if OverallMetCriteria = Y. This is true only when both ServicingMetCriteria and AccessMetCriteria are equal to Y:

- ServicingMetCriteria indicates whether the number of servicing providers with the specialty is equal to or greater than the required number for the county.

For example, if the guidelines mandate that a county has 2 cardiologists (MinimumServicingProviders=2) and there is only 1 (InRegionProviderCount=1), then ServicingMetCriteria = N.

- AccessMetCriteria indicates whether enough members have access to the specialty (i.e. the providers are in the right geographic area to cover the mandated percentage of members).

For example, if the guidelines mandate that 85% of members have access to a cardiologist in a county, but MemberAccessPercentage=79, then AccessMetCriteria=N.

Uploading Your Data to Quest

Clients typically employ a weekly cadence of uploading data via the API, for example Monday – Friday at 7 PM EST. The data uploaded to Quest on weekdays is processed at 10 PM, CST. Data uploaded on the weekends is processed immediately. The processing includes geocoding the data and rerunning the network adequacy projects associated with the new data.

To upload your data to Quest:

1. Create a network version by submitting a POST request to the following endpoint. For example:

```
POST /enterprise/data-mgmt/v3/ProviderNetworks/{{providerNetworkId}}/versions
HTTP/1.1
```

Host: <https://uat-api.questanalytics.com>

Authorization: Bearer {{accessToken}}

The providerNetworkId path parameter is typically provided to you by your client-services representative but can also be retrieved using the data-mgmt/v3/ProviderNetworks endpoint.

The POST request returns the id of the new empty version. For example:

```
{
  "providerNetworkVersion": {
    "providerNetworkVersionId": 312312
  }
}
```

2. There are two ways to upload a batch of data:

- If your data is in a flat file format, upload a batch of accesspoints to the new empty network version. An accesspoint is the combination of a unique provider (NPI), location, and specialty.
- If your data is in a hierarchical format, upload a batch of providers to the new empty network version.

These two methods are described in the following two sections.

Uploading Accesspoints

To add provider network accesspoints, submit a POST request like the following:

```
POST /enterprise/data-mgmt/v3/ProviderNetworks/{{providerNetworkId}}/versions/
{{providerNetworkVersionId}}/accesspoints?isPartial=true HTTP/1.1
```

The POST takes the following parameters:

- The same providerNetworkId you specified in step 1.
- The providerNetworkVersionId returned in step 1 (e.g. 312312).
- The isPartial query string parameter. If this is the first of several batches of providers being uploaded, set isPartial to true. For the last batch, set isPartial to false.

Note: An alternate way to signal you are finished uploading data is to use the SetProviderNetworkVersionStatus API.

In the body of the POST, specify one record for each accesspoint being uploaded. For example:

```
{
  "NPI": "1811454077",
  "OrganizationNPI": "1811454077",
  "Name": "Aaron, Benjamin T.",
  "FirstName": "",
  "MiddleName": "",
  "LastName": "",
  "Degree": "",
  "CMSSpecialtyCode": "001",
  "SpecialtyDescription": "",
  "TaxonomyCode": "207K00000X",
  "Address1": "8401 Hwy 111",
  "Address2": "",
  "City": "Byrdstown",
  "State": "TN",
  "Zip": "38549",
  "Phone": "",
  "Fax": "",
  "Email": "",
  "IsPCPOrPCP_SpecialistFlag": "",
  "ContractType": "DS",
```

```

    "AcceptingNewPatients": "Y",
    "MedicalGroupAffiliationOrDC": "Byrdstown Medical Center",
    "MedicareCertifiedBeds": "4",
    "GroupAffiliationUniqueIdentifier": "",
    "UsesCMSMAContractAmendment": "N",
    "TaxId": "12354124123",
    "LastPaidClaimsDate": "",
    "TotalClaimsPaid": "",
    "TotalClaimants": "",
    "CAQHId": "",
    "QualityIndicators_HEDIS_CAHPS_Measures": "",
    "ContractId": "",
    "GeoInfo": "Address-Exact",
    "Latitude": "36573656",
    "Longitude": "85152618",
    "CountySSA": "44680",
    "CountyName": "Pickett",
    "IncludeInCloudUpload": "Y",
    "IncludeInHSDTable": "Y",
    "RecordSource": "Network A",
    "AsOfDate": "10/25/2019",
    "Network": "A",
    "AccessPointType": "Facility"
  }

```

Uploading Providers

To add provider network providers, submit a POST request like the following:

```

POST /enterprise/data-mgmt/v3/ProviderNetworks/{{providerNetworkId}}/versions/
{{providerNetworkVersionId}}/providers?isPartial=true HTTP/1.1

```

The POST takes the following parameters:

- The same providerNetworkId you specified in step 1.
- The providerNetworkVersionId returned in step 1 (e.g. 312312).
- The isPartial query string parameter. If this is the first of several batches of providers being uploaded, set isPartial to true. For the last batch, set isPartial to false.

Note: An alternate way to signal you are finished uploading data is to use the `SetProviderNetworkVersionStatus` API.

In the body of the POST, specify one record for each provider being uploaded. The record is structured hierarchically, specifying all the data for a provider, such as all the locations and specialties for that provider. For example:

```
{
  "externProviderId": 0,
  "npi": 1261051,
  "tin": "128-1040",
  "name": "Lara Vang",
  "prefix": "Dr.",
  "firstName": "Dorothea",
  "middleName": "Mallory",
  "lastName": "Walker",
  "suffix": null,
  "credentials": null,
  "gender": "female",
  "networkIdentifier": "MAPP, DSNP",
  "AccessPointType": "Facility",
  "medicalGroupAffiliations": [
    "qui",
    "velit",
    "dolor",
    "et",
    "Lorem"
  ],
  "locations": [
    {
      "externLocationId": 0,
      "practiceName": "COMVEY",
      "address1": "748 Aviation Road",
      "address2": null,
      "city": "Alfarata",
      "state": "South Carolina",

```

```

    "zip": 12682,
    "plus4": null,
    "phone": "+1 (848) 547-3139",
    "fax": "+1 (846) 545-2057",
    "cargoField2": true,
    "serviceLocationWebsite": "www.questanalytics.com",
    "specialties": [
      {
        "externSpecialtyCode": "consequat",
        "taxonomyCode": "164W00000X",
        "cmsSpecialtyCode": "001"
      }
    ],
    "acceptingNewPatients": true,
    "medicareCertifiedBeds": false
  },
  {
    "externLocationId": 1,
    "practiceName": "QUILTIGEN",
    "address1": "999 Beard Street",
    "address2": null,
    "city": "Clinton",
    "state": "Mississippi",
    "zip": 11894,
    "plus4": null,
    "phone": "+1 (813) 443-2193",
    "fax": "+1 (824) 547-3540",
    "specialties": [
      {
        "cargoField2": true,
        "externSpecialtyCode": "sit",
        "taxonomyCode": "",
        "cmsSpecialtyCode": "001"
      }
    ]
  },

```

```

    "acceptingNewPatients": true,
    "medicareCertifiedBeds": false
  },
  {
    "externLocationId": 2,
    "practiceName": "PLASMOSIS",
    "address1": "734 Madison Place",
    "address2": null,
    "city": "Ballico",
    "state": "South Dakota",
    "zip": 89961,
    "plus4": null,
    "phone": "+1 (886) 573-3748",
    "fax": "+1 (806) 422-2876",
    "specialties": [
      {
        "cargoField2": false,
        "externSpecialtyCode": "est",
        "taxonomyCode": "164W00000X",
        "cmsSpecialtyCode": "001"
      }
    ],
    "acceptingNewPatients": true,
    "medicareCertifiedBeds": true
  },
  {
    "externLocationId": 3,
    "practiceName": "BLUPLANET",
    "address1": "920 Navy Walk",
    "address2": null,
    "city": "Chase",
    "state": "Florida",
    "zip": 10402,
    "plus4": null,
    "phone": "+1 (848) 592-2172",

```

```

    "fax": "+1 (932) 496-3016",
    "specialties": [
      {
        "cargoField2": false,
        "externSpecialtyCode": "tempor",
        "taxonomyCode": "164W00000X",
        "cmsSpecialtyCode": "067"
      }
    ],
    "acceptingNewPatients": true,
    "medicareCertifiedBeds": true
  }
],
"contractType": null,
"employmentStatus": null,
"underContract": "Y",
"medicareCCN": null,
"acceptsMedicareAssignment": null
}

```

Checking the Status of Uploads to a Network Version

To get the upload status by version id, submit a GET request to the following endpoint:

`/v3/ProviderNetworks/{providerNetworkId}/versions/{providerNetworkVersionId}/upload`

The GET takes the following parameters:

- The `providerNetworkId` is the network for which you created a new version.
- The `providerNetworkVersionId` is the id of the network version.

The request returns a record of the following form which indicates the status of the upload:

```

{
  "started": "2021-04-12T20:11:25.1216242",
  "completed": null,
  "status": "UploadComplete",
  "accessPointCount": 2,
  "providerNetworkId": 30453,
  "providerNetworkVersionId": 249767,
  "missingFields": null
}

```

```
}
```

Status: Can be created, uploadcomplete, geocodecomplete, activated, or archived. A status of activated indicates the new version has been published.

accessPointCount: The number of accesspoints uploaded and saved in accesspoint format. Regardless of whether you are uploading accesspoints (flat file format) or providers (hierarchical file format), the data is stored internally as accesspoints.

Retrieving the Status of All Filtered Network Uploads

To retrieve the status of the uploads to every filtered network, submit a request to the following endpoint:

GET /v3/upload

The request returns one record of the following form for each filtered network:

```
{
  "started": "2023-04-11T05:00:43.1755787",
  "completed": "2023-04-11T05:00:54.2317505",
  "status": "Final",
  "accessPointCount": 2,
  "providerNetworkId": 30453,
  "providerNetworkVersionId": 324169,
  "missingFields": null
}
```

Retrieving Providers from a Filtered Network

The POST request you use to upload providers to a filtered network has a corresponding Get request which you can use to retrieve the providers for a specific filtered network, passing in a page size and count to limit the data returned. For example:

GET /enterprise/data-mgmt/v3/ProviderNetworks//{{providerNetworkId}}/versions/{{providerNetworkVersionId}}/providers?pageSize=10&pageNumber=1

This request returns one record of the following form for each provider:

```
{
  "npi": "1811454077",
  "tin": null,
  "name": "Aaron, Benjamin T.",
  "prefix": null,
  "firstName": "",
  "middleName": "",

```

"lastName": "",
"suffix": null,
"credentials": null,
"gender": null,
"address1": "8401 Hwy 111",
"address2": "",
"city": "Byrdstown",
"state": "TN",
"zip": "38549",
"countySSA": "44680",
"latitude": "36573656",
"longitude": "85152618",
"geoInfo": "AddressExact",
"phone": "",
"fax": "",
"taxonomyCode": "207K00000X",
"practiceName": null,
"plus4": null,
"cmsSpecialtyCode": "001",
"contractType": "DS",
"acceptingNewPatients": "True",
"employmentStatus": null,
"underContract": null,
"medicalGroupAffiliation": null,
"medicareCertifiedBeds": "4",
"medicareCCN": null,
"acceptsMedicareAssignment": "False",
"externZip": null,
"externSpecialtyCode": null,
"externRowId": null,
"externProviderId": null,

```

    "externLocationId": null,
    "providerNetworkUploadId": "61390",
    "entityType": "Individual",
    "letterOfIntent": null
  }

```

Note: To retrieve all providers for a filtered network:

*GET enterprise/data-
mgmt/v3/ProviderNetworks/{providerNetworkId}/versions/{providerNetworkV
ersionId}/providers/all*

Servicing Provider Report

The Servicing Provider Report contains one record for each provider in each county. Use cases for this data vary depending on your business logic. Two common use cases are:

- Re-ingest the data back into your system to feed internal provider dashboards.
 - Analyze the “serves percent” field to understand provider coverage.
1. To get Servicing Provider Results, submit a GET request to the following endpoint. The request takes a projectId path parameter and an authorization token in the request body. For example:

GET /enterprise/proj-mgmt/v3/Projects/{projectId}//adequacy/results/servicing-providers HTTP/1.1

Host: https://uat-api.questanalytics.com

Authorization: Bearer {{accessToken}}

This response contains a data array which includes one element of the following form for each provider in each county:

```

{
  "ServicingState": "TN",
  "ServicingCountySSA": "44130",
  "ServicingCountyName": "Clay",
  "ServicingSpecialtyGroupCode": "S03",
  "ServicingSpecialtyGroup": "Primary Care",
  "NPI": "1811454077",
  "Name": "Aaron, Benjamin T.",
  "Phone": "",
  "Fax": ""
}

```

"Address": "8401 Hwy 111",
"Address2": "",
"City": "Byrdstown",
"State": "TN",
"Zip": "38549",
"CountySSA": "44680",
"CountyName": "Pickett",
"Latitude": 36573656,
"Longitude": 85152618,
"GeoInfo": "Address-Exact",
"SpecialtyCode": "001",
"Specialty": "General Practice",
"MedicareCCN": "",
"MedicareCertifiedBeds": "4",
"Serves": 82.3,
"AcceptsMedicareAssignment": "",
"Prefix": "",
"First": "",
"Middle": "",
"Last": "",
"Suffix": "",
"Credentials": "",
"Gender": "",
"Taxonomy": "207K00000X",
"ContractType": "DS",
"AcceptingNewPatients": "Y",
"MedGroupAffiliation": "",
"Quality": "",
"UnderContract": "",
"MedicareClaims": "",
"MedicareClaimants": "",

```

    "IsNextClosestProvider": "",
    "NextClosestAverageDistance": "",
    "NextClosestIsServicing": "",
    "NextClosestMemberCount": "",
    "ReasonForExcludingNcp": "",
    "CommentNcp": "",
    "ExclusionVerifiedSourcesNcp": ""
  }

```

The following table provides explanations for those fields in the response that are not self-explanatory:

MedicareCCN	The Medicare Certification Number is a unique identifier for a Facility. No longer in direct use by CMS, but some plans still use this information.
MedicareCertifiedBeds	Number of beds inside a hospital that have been certified by CMS.
Serves	Percentage of members covered within time/distance at this location.
AcceptsMedicareAssignment	Whether or not the provider/facility has accepted the Medicare reimbursement rate.
ContractType	Medicare HSD field. DC = Direct Contract. DS = Downstream Contract.
MedGroupAffiliation	Medicare HSD field. Name of the affiliated Medical Group, if any.
Quality	Metric used to measure the Quality of a provider (this field typically uses client-identified data).
UnderContract	Medicare HSD field. Does the health plan use the CMS Contract Amendment to contract with the provider? (Y or N)
MedicareClaims	Dollar amount of Medicare Claims. Driven by CMS Claims data (2 years in arrears).
MedicareClaimants	Number of Medicare Claims. Driven by CMS Claims data (2 years in arrears).
IsNextClosestProvider	Must have exceptions package add-on purchased for this to show up. Indicates whether the provider is a next closest provider outside of time/distance for specialty exception purposes.
NextClosestAverageDistance	This field is available only when client has purchased exceptions package add-on. Indicates the average distance to members if the Provider is a next closest provider outside of time and distance.
NextClosestIsServicing	This field is available only when client has purchased exceptions package add-on. Indicates if the Provider is already within time/distance of other members (Y or N).
NextClosestMemberCount	This field is available only when client has purchased exceptions package add-on. Indicates the number of members this provider is the closest to, outside of time and distance.

ReasonForExcludingNcp	This field is available only when client has purchased exceptions package add-on. Indicates a reason if this provider was excluded (for example, Retired or Deceased).
CommentNcp	This field is available only when client has purchased exceptions package add-on. Indicates user entered comments explaining why this provider was excluded.
ExclusionVerifiedSourcesNcp	This field is available only when client has purchased exceptions package add-on. Indicates a list of Verified Sources where this excluded providers information was found.
LetterOfIntent	Medicare HSD Field. Indicates whether or not the health plan is using a Letter of Intent with the provider on their HSD table, in lieu of a signed contract.

2. To analyze the potential impact of the provider's quality on your business, evaluate the relationship between the Serves field and the Quality field. The Serves field indicates the percent of the county served by the provider, as calculated by Quest Analytics based on the latitude and longitude of the provider's office location. The Quality field contains the quality score, as determined by your internal metrics.
 - If a provider's Quality score is high and the provider serves a large percentage of members in the county, negotiations with that provider should focus on financial incentives to retain the provider in the network .
 - If a provider's Quality score is low and the provider serves a large percentage of members in the county, it could be difficult to sell plans to prospective members.
 - If a provider's Quality score is low and the provider serves a small percentage of members in the county, you might consider removing that provider from the network to improve the overall quality rating.

Creating an HSD File Format for Upload to the HPMS Portal

To create a copy of the Servicing Provider Results in the HSD format required by CMS, submit a GET request to the following endpoint:

```
GET /enterprise/proj-mgmt/v3/Projects/{{projectId}}//adequacy/results/servicing-providers?HSD=true HTTP/1.1
```

Host: <https://uat-api.questanalytics.com>

Authorization: Bearer {{accessToken}}

The request takes the following parameters:

- The projectId path parameter.
- The HSD query string parameter.
- The authorization token in the request body.

Creating a Project Based on a Filtered Network

Network data files can be hundreds of thousands or even millions of lines long. When your business logic dictates that you analyze or process a smaller group of providers, the best practice is to create a

filtered network and then create a project based on that network. For example, you might want to create a network that consists only of your HMO product, and then analyze that network using a project you create.

1. Before creating new filtered networks, the best practice is to check all the existing filtered network versions associated with your network to make sure there is no duplication. To get provider network versions, submit a GET request to the following endpoint:

```
enterprise/data-mgmt/v3/ProviderNetworks/{providerNetworkId}/versions
```

The response contains a JSON structure that includes basic information about the network, followed by one record for each filtered network associated with the network. For example:

```
{
  "providerNetwork": {
    "providerNetworkId": 30453,
    "name": "Primary Network",
    "description": " Primary Network",
    "dateCreated": "2020-11-30T17:06:09.33",
    "dateLastUpdated": "2022-09-29T16:09:40.337",
    "dateArchived": "0001-01-01T00:00:00",
    "isFilteredProviderNetwork": false,
    "sourceProviderNetworkId": 0,
    "providerNetworkType": "Standard",
    "status": "Active",
    "versions": [
      {
        "providerNetworkVersionId": 246912,
        "status": "Created",
        "dateCreated": "2020-11-30T17:06:09.377Z",
        "dateLastUpdated": "2020-11-30T17:06:09.377",
        "dateActivated": null,
        "dateArchived": "2021-04-20T15:55:34.12Z"
      },
      {
        "providerNetworkVersionId": 321929,
```

```

        "status": "Activated",
        "dateCreated": "2023-04-04T05:02:39.49Z",
        "dateLastUpdated": "2023-04-04T05:02:46.813",
        "dateActivated": "2023-04-04T05:02:46.813Z",
        "dateArchived": null
    }
]
}
}

```

2. To create a filtered provider network, submit a POST request to the following endpoint:

POST /enterprise/data-mgmt/v3/provider networks/filtered HTTP/1.1

Host: https://uat-api.questanalytics.com

Authorization: Bearer {{accessToken}}

Content-Type: application/json

Content-Length: 268

```

{
  "name": "HMO",
  "description": "HMO",
  "sourceProviderNetworkId": 12345,
  "criteria": [{
    "field": "NetworkIdentifier",
    "value": "HMO",
    "operator": "="
  }],
  "sharedUsers": []
}

```

The body of the POST takes the following parameters:

- The name and description should be pneumatic and indicate the purpose of the filtered network.
 - The sourceProviderNetworkId is the id of the full network from which to extract the filtered network.
 - The criteria JSON object specifies how the full network should be filtered. In this example, the network will be filtered to include only those providers where the value of NetworkIdentifier is equal to HMO. NetworkIdentifier can be any column name in your data.
3. To create a project, submit a POST request to the following endpoint:

```

POST /enterprise/proj-mgmt/v3/Projects HTTP/1.1
Host: https://uat-api.questanalytics.com
Authorization: Bearer {{accessToken}}
Content-Type: application/json
Content-Length: 533
{
  "name": "HMO",
  "description": "HMO",
  "providerNetworkId": 12345,
  "adequacyTemplateId": "585af255-21a4-4402-895b-81f50d4e1af5",
  "countyStatuses":[
    {
      "state":"TN",
      "countySSA":"44000",
      "countyZips": [37716,37717],
      "status":"approved",
    },
    {
      "state":"TN",
      "countySSA":"44010",
      "countyZips": [37718,37719],
      "status":"active",
    }
    {
      "state":"TN",
      "countySSA":"44020",
      "countyZips": [],
      "status":"opportunity",
    }
  ]
}

```

The POST request takes the following parameters:

- The name and description should be pneumonic and indicate the purpose of the filtered network.
- The ProviderNetworkId is the identifier for the filtered network created in the previous step.

- The `adequacyTemplateId` is used by the project to determine the time and/or distance standards that will be used. These are pre-defined templates licensed by your company (time/distance standards cannot be edited). The `AdequacyTemplateId` can be obtained by calling the `Get Templates` API.
- `excludeSpecialtyGroupCodes` is a list of comma-separated specialties to be excluded from the project.
- `countyStatuses` contains one object for each county which specifies the following information.
 - `state`: The two-digit post office code for the state in which the county is located.
 - `countySSA`: The Social Security Administration code for the county.
 - `countyZips`: A comma-separated list of the post office zip codes in the county if you need a partial county analyzed. Otherwise, leave this empty, and the full county will be analyzed.
 - `status`: The status of the plan sales in the county. There are four statuses:
 - `Pending`: The regulator has not yet approved the application to sell your product in the county.
 - `Approved`: The regulator has approved the application to sell your product in the county for the next calendar year.
 - `Active`: Your product is currently being sold in the county.
 - `Opportunity`: Evaluate this county for market potential to determine if an application to sell should be submitted to the regulator.

Project Management

Updating Project User Sharing

To allow end-users to see a project in the QES User Interface Project Dashboard, submit a PUT request to the following endpoint, passing the list of end-user email addresses in the body of the request:

PUT `/enterprise/proj-mgmt/v3/Projects/{projectId}/sharedusers`

Pass the list of end-user email addresses in the body of the request. For example:

```
{
  "sharedUsers": ["chris.smith@yourcompany.com"]
}
```

The request returns the number of email addresses that were successfully added and the email addresses of users that could not be added. For example, if `chris.smith@yourcompany.com` is successfully added, the response looks like this:

```
{
  "successfullySharedCount": 1,
  "failedSharingUsers": []
}
```

And if `chris.smith@yourcompany.com` cannot be added, the response looks like this:

```
{
  "successfullySharedCount": 0,
  "failedSharingUsers": [
    "chris.smith@yourcompany.com"
  ]
}
```

Getting a List of Existing Projects

Before creating new projects, the best practice is to check all the existing projects to make sure there is no duplication. To retrieve a list of all existing projects, submit a GET request to the Get Projects API. For example:

GET /enterprise/proj-mgmt/v3/Projects

The response contains one record for each project which includes detailed information for the project. For example:

```
{
  "projectId": 30435,
  "name": "Project Name",
  "description": "This is a description of the project",
  "dateCreated": "2020-11-10T19:05:50.79",
  "dateLastUpdated": "2023-03-21T17:24:07.793",
  "dateArchived": null,
  "datePurged": null,
  "isShared": false,
  "memberSetId": 25225,
  "providerNetworkId": 30425,
  "prospectProviderNetworkId": null,
  "isRunning": false,
  "countyStatuses": null,
  "providerNetworkName": "Primary Network",
  "prospectNetworkName": null,
  "templateName": "MAC 2021 Distance",
  "templateDescription": "Medicare template collection.",
  "memberSetName": "Medicare Advantage CY2021 Benes",
  "countyClassMapping": "CMS County Class Mapping (2019.4)"
}
```

Getting and Setting Project Information

Before updating project information, the best practice is to check the existing projects information to determine if any updates are needed, such as service area changes.

To retrieve detailed information about a specific project, submit a GET request to the following endpoint:

```
GET /enterprise/proj-mgmt/v3/Projects/{{projectId}}
```

The response returns a JSON object like the following:

```
{
  "project": {
    "projectId": 31499,
    "name": "API Update test 9919bcc0-726b-47b8-892c-b4c00b8919bd",
    "description": "API Update test 9182338a-7ebe-4c19-8ed1-6396a3c88026",
    "dateCreated": "2021-10-25T19:04:41.657",
    "dateLastUpdated": "2023-04-11T03:00:24.157",
    "dateArchived": null,
    "datePurged": null,
    "isShared": true,
    "memberSetId": 30494,
    "providerNetworkId": 30453,
    "prospectProviderNetworkId": null,
    "isRunning": false,
    "countyStatuses": {
      "statuses": [
        {
          "state": "TN",
          "countySSA": "44000",
          "countyZips": [
            "37716",
            "37717"
          ],
          "status": "Approved"
        },
        {
          "state": "TN",
          "countySSA": "44010",
```

```

        "countyZips": [],
        "status": "Approved"
    }
]
},
"providerNetworkName": "QESI Primary Test Network",
"prospectNetworkName": null,
"templateName": "MAC 2022 Distance",
"templateDescription": "Medicare template collection.",
"memberSetName": "Medicare Advantage CY2022 Benes",
"countyClassMapping": "CMS County Class Mapping (2020.4)"
}
}

```

To update a project, submit a PUT request to the following endpoint, passing the updates in the request body:

PUT /enterprise/proj-mgmt/v3/Projects/{{projectId}}

Getting Project Status

A change in project status occurs when a project is updated with new providers and is rerun by QES to generate the new adequacy analysis. To get project status, submit a GET request to the following endpoint:

GET /enterprise/proj-mgmt /v3/Projects/{{projectId}}/status

The response indicates the status of the project and the status of the last run. For example:

```

{
  "projectId": 31499,
  "projectStatuses": {
    "projectStatus": "Active",
    "projectRunStatus": "Succeeded"
  }
}

```

The possible values for projectStatus are: active or archived

The possible values for projectRunStatus are: created, queued, running, completed, or failed

Getting Project Ids by Network Ids

To retrieve a collection of Project Ids associated with a specific list of networks Ids, submit a GET request to the following endpoint:

/enterprise/proj-mgmt/v3/Projects/networks?networkIdList=[{{providerNetworkId}}]

The response contains a list of project ids for each network id submitted in the request. For example:

```
{
  "projectIds": [
    31147,
    31499,
    33441
  ]
}
```

Network Management

Updating Filtered Network Sharing

To add users who must see a filtered network in the QES Project Dashboard, submit a PUT request to the following endpoint:

PUT /enterprise/data-mgmt/v3/providernetworks/filtered/{{filteredNetworkId}}/sharedUsers

Pass the list of end-user email addresses in the body of the request. For example:

```
{
  "sharedUsers": ["chris.smith@yourcompany.com"]
}
```

The request returns the number of email addresses that were successfully added and the email addresses of users that could not be added. For example, if chris.smith@yourcompany.com is successfully added, the response looks like this:

```
{
  "successfullySharedCount": 1,
  "failedSharingUsers": []
}
```

And if chris.smith@yourcompany.com cannot be added, the response looks like this:

```
{
  "successfullySharedCount": 0,
  "failedSharingUsers": [
    "chris.smith@yourcompany.com"
  ]
}
```

Getting Provider Networks

Before creating new filtered networks, the best practice is to check the existing networks to determine if any new ones need to be created.

To get all networks, submit a GET request to the following endpoint:

GET /enterprise/data-mgmt/v3/ProviderNetworks

Note: Set the isFiltered flag to true to retrieve only filtered networks.

The response contains detailed information about each network and its versions, such as the date last updated and the status. For example:

```
{
  "providerNetworks": [
    {
      "providerNetworkId": 31573,
      "name": "Test Network",
      "description": "Test Network",
      "dateCreated": "2021-11-24T18:46:58.94",
      "dateLastUpdated": "2021-11-24T18:46:58.94",
      "dateArchived": "0001-01-01T00:00:00",
      "isFilteredProviderNetwork": false,
      "sourceProviderNetworkId": 0,
      "providerNetworkType": "Standard",
      "status": "Active",
      "versions": [
        {
          "providerNetworkVersionId": 261573,
          "status": "Activated",
          "dateCreated": "2021-11-24T18:46:59.003",
          "dateLastUpdated": "2021-11-24T18:47:12.14",
          "dateActivated": "2021-11-24T18:47:12.14",
          "dateArchived": null
        }
      ]
    },
    {
      "providerNetworkId": 30450,
      "name": "Primary Network",
```

```

    "description": "Primary Network",
    "dateCreated": "2020-11-23T19:19:34.31",
    "dateLastUpdated": "2020-11-23T19:19:34.31",
    "dateArchived": "0001-01-01T00:00:00",
    "isFilteredProviderNetwork": false,
    "sourceProviderNetworkId": 0,
    "providerNetworkType": "Standard",
    "status": "Active",
    "versions": [
      {
        "providerNetworkVersionId": 246889,
        "status": "Created",
        "dateCreated": "2020-11-23T19:19:34.31",
        "dateLastUpdated": "2020-11-23T19:19:34.31",
        "dateActivated": null,
        "dateArchived": null
      },
      {
        "providerNetworkVersionId": 246890,
        "status": "Activated",
        "dateCreated": "2020-11-23T19:27:36.557",
        "dateLastUpdated": "2020-11-23T19:29:31.273",
        "dateActivated": "2020-11-23T19:29:31.273",
        "dateArchived": null
      },
      {
        "providerNetworkVersionId": 291827,
        "status": "Activated",
        "dateCreated": "2022-12-19T20:38:15.3",
        "dateLastUpdated": "2022-12-19T20:38:15.3",
        "dateActivated": "2022-12-19T20:38:15.3",
        "dateArchived": null
      }
    ]
  }
]

```

```
}
```

Getting Provider Network Details

To get information for a specific network, submit a get request to the following endpoint:

```
GET /enterprise/data-mgmt/v3/ProviderNetworks/{{providerNetworkId}}
```

The response contains a large JSON object which lists general network information first, such as name, creation date, and status, following by one record for each version of the network. For example:

```
{
  "providerNetwork": {
    "providerNetworkId": 30453,
    "name": "Primary Network",
    "description": "Primary Network",
    "dateCreated": "2020-11-30T17:06:09.33",
    "dateLastUpdated": "2022-09-29T16:09:40.337",
    "dateArchived": "0001-01-01T00:00:00",
    "isFilteredProviderNetwork": false,
    "sourceProviderNetworkId": 0,
    "providerNetworkType": "Standard",
    "status": "Active",
    "versions": [
      {
        "providerNetworkVersionId": 322569,
        "status": "Activated",
        "dateCreated": "2023-04-06T05:02:42.223Z",
        "dateLastUpdated": "2023-04-06T05:02:52.63",
        "dateActivated": "2023-04-06T05:02:52.63Z",
        "dateArchived": null
      },
      {
        "providerNetworkVersionId": 322888,
        "status": "Activated",
        "dateCreated": "2023-04-07T05:01:39.847Z",
        "dateLastUpdated": "2023-04-07T05:01:48.013",
```

```

    "dateActivated": "2023-04-07T05:01:48.013Z",
    "dateArchived": null
  },
  {
    "providerNetworkVersionId": 322889,
    "status": "Activated",
    "dateCreated": "2023-04-07T05:01:41.63Z",
    "dateLastUpdated": "2023-04-07T05:01:48.35",
    "dateActivated": "2023-04-07T05:01:48.35Z",
    "dateArchived": null
  }
]
}
}

```

Getting Provider Network Version Details

To get a provider network version, submit a GET request to the following endpoint:

```
GET enterprise/data-mgmt/v3/ProviderNetworks/{{providerNetworkId}}/versions/{{providerNetworkVersionId}}
```

The request returns a JSON object that contains details about the network version. For example:

```

{
  "providerNetworkVersion": {
    "providerNetworkId": 30453,
    "dateCompleted": "2021-04-12T20:11:30.103Z",
    "uploads": [
      {
        "providerNetworkUploadId": 61390,
        "groupName": "QESI",
        "sourceName": "Provider Network Worker",
        "dateStarted": "2021-04-12T20:11:28.49",
        "dateFinished": "2021-04-12T20:11:29.9",
        "uploadUrl": null
      },
      {

```

```

        "providerNetworkUploadId": 61391,
        "groupName": "QESI",
        "sourceName": "Provider Network Worker",
        "dateStarted": "2021-04-12T20:11:30.287",
        "dateFinished": "2021-04-12T20:11:30.877",
        "uploadUrl": null
    },
    {
        "providerNetworkUploadId": 61392,
        "groupName": "QESI",
        "sourceName": "Provider Network Worker",
        "dateStarted": "2021-04-12T20:11:30.347",
        "dateFinished": "2021-04-12T20:11:31.207",
        "uploadUrl": null
    }
],
"rowCount": 6,
"providerNetworkVersionId": 249767,
"status": "Activated",
"dateCreated": "2021-04-12T20:11:23.783Z",
"dateLastUpdated": "2021-04-12T20:11:30.373",
"dateActivated": "2021-04-12T20:11:30.373Z",
"dateArchived": "2021-04-20T15:55:34.12Z"
}
}

```

Getting and Setting Provider Network Version Status

To get the status of a network version, submit a GET request to the following endpoint:

/v3/ProviderNetworks/{{providerNetworkId}}/versions/{{providerNetworkVersionId}}/status

The request returns the status and echoes back the providerNetworkVersionId. For example:

```

{
    "status": "Activated",
    "providerNetworkVersionId": 123456
}

```

For partial uploads, after the final partial submission, you must set the status to 'Upload Complete'. To set the status of a network version, submit a PUT request to the following endpoint:

/v3/ProviderNetworks/{{providerNetworkId}}/versions/{{providerNetworkVersionId}}/status/{{Status}}

For example:

/v3/ProviderNetworks/34567/versions/123456/status/UploadComplete