

InCommon®



InCommon Certificate Manager

Discovery API

InCommon
c/o Internet2
1000 Oakbrook Drive, Suite 300
Ann Arbor MI, 48104

Table of Contents

1 Introduction.....	3
2 Discovery Tasks.....	3
2.1 API method for Creating a New Discovery Task.....	4
2.2 API Method for Editing a Task by Task ID.....	8
2.3 API Method for Getting Details of a Task by Task ID.....	12
2.4 API Method for Getting Number of Existing Tasks.....	14
2.5 API Method for Getting List of Existing Tasks.....	15
2.6 API Method for Starting Discovery Scan for a Specific task.....	16
2.7 API Method for Getting Status of a Specific Task.....	17
2.8 API Method for Stopping Discovery Scan for a Specific Task.....	18
2.9 API Method for Removing a Task by Task ID.....	19
3 Scan History.....	20
3.1 API Method for Getting Details of a Scan for a Specific Task.....	20
3.2 API Method for Getting Details of Discovered SSL Certificates from Last Scan.....	22
3.3 API Method for Getting Details of Discovered SSL Certificates from a Specific Scan.....	25
4 Auto-Assignment Rules.....	28
4.1 API Method for Creating a New Rule.....	29
4.2 API Method for Editing a Rule.....	32
4.3 API Method for Getting Number of Existing Rules.....	34
4.4 API Method for Getting the List of Existing Rules.....	35
4.5 API Method for Getting Rule Details.....	36
4.6 API Method for Removing a Rule.....	38

1 Introduction

The Discovery API automates frequently performed operations to accelerate certificate discovery for customers with dynamically changing IP ranges. The Discovery API is of RESTful type.

Discovery API provides access for 3 resources:

- [Discovery Tasks](#)
- [Scan History](#)
- [Auto-Assignment Rules](#)

2 Discovery Tasks

The 'Tasks' resource contains information about planned discovery scans. A task comprises general information (task name, agent, ranges to scan), assignment rules, scan schedule, and has a 'Status' parameter.

You can access the 'Tasks' screen by authenticating yourself using one of the following two methods:

1. Authentication via Username and Password
 - Users should have InCommon CM login credentials and the correct customer login URI

The URI for the username/password authentication is:

- <https://cert-manager.com:443/api/discovery/v1/task>

2. Authentication via Username and a Client Certificate

- Admins should have 'Certificate Auth' enabled. The authentication certificate must be requested and issued via InCommon CM and active at the moment of authentication.

The URI for the username/client certificate authentication is:

- <https://cert-manager.com:443/private/api/discovery/v1/task>

The Web Application Description Language (WADL) file can be accessed via the following URI:

- <https://cert-manager.com:443/api/application.wadl>

There are nine methods available:

- [API method for Creating a New Discovery Task](#)
- [API Method for Editing a Task by Task ID](#)
- [API Method for Getting Details of a Task by Task ID](#)
- [API Method for Getting Number of Existing Tasks](#)
- [API Method for getting List of Existing Tasks](#)
- [API Method for Starting Discovery Scan for a Specific task](#)
- [API Method for Getting Status of a Specific Task](#)
- [API Method for Stopping Discovery Scan for a Specific Task](#)
- [API Method for Removing a Task by Task ID](#)

2.1 API method for Creating a New Discovery Task

Mandatory Fields are marked in red.

HTTP Method	Resource Parameters	Query Parameters	Description
POST		<pre> { "name": "Task Name", "agent": "Agent Name" *, "ranges": [{ "address": "DNS/IP/CIDR", "ports": "port, port, ..." }, { "address": "DNS/IP/CIDR", "ports": "port-port" }, ...] "rules": ["RuleName", "RuleName"], "frequency": "Frequency" **, "timeZone": "TimeZone" ***, "time": { "hours": "hour", "minutes": "minutes" } </pre>	<p>Enables administrators to create a new task. Ranges must be fully supported by the agent.</p>

		}	
		}	

Response

In case of success:

HTTP 200 – OK

```
{  
  "taskId": id  
}
```

In case of ranges partial/no match (for Agent = Auto):

HTTP 400 with error message: *"There are no available agents for all the specified private ranges"*.

In case, a mandatory parameter was not provided:

HTTP 400 with error message: *"[Parameter] cannot be empty"*.

In case, an invalid parameter was provided:

HTTP 400 with error message: *"[Parameter] contains invalid value"*.

Example

HTTP POST /api/discovery/v1/task

```
{  
  "name": "TestTask",  
  "agent": "Agent 1",  
  "ranges": [  
    {  
      "address": "10.100.10.15/32",  
      "ports": "443, 8080"    }  
  ]  
}
```

```
},  
{  
  "address": "cert-manager.com",  
  "ports": "443-680"  
}  
]  
"rules": [  
  "RuleForAWS"  
],  
"frequency": "Monthly",  
"timeZone": "UTC+08:45 - CWST",  
"time": {  
  "hours": "10",  
  "minutes": "23"  
}  
}  
response:  
HTTP 200 – OK  
{  
  "taskId": 51  
}
```

* "agent" parameter can be Agent's name or 'Auto'.

** "frequency" parameter can have the following values:

- 'Manual',
- 'Daily',
- 'Weekly',

- 'Monthly',
- 'Quarterly',
- 'Semi-Annually',
- 'Annually'.

*** "timeZone" parameter can have the following values:

- "UTC-12:00 – BIT",
- "UTC-11:30 - NUT"
- "UTC-11:00 – SST",
- "UTC-10:00 - HAST, HST, TAHT, CKT",
- "UTC-09:30 - MART, MIT",
- "UTC-09:00 - AKST, GAMT, GIT, HADT",
- "UTC-08:00 - PST, CHOT, CIST, AKDT",
- "UTC-07:00 - MST, PDT",
- "UTC-06:00 - CST, EAST, GALT, MDT",
- "UTC-05:00 - CST, ORAT, PET, CHOT",
- "UTC-04:30 – VET",
- "UTC-04:00 - AST, ECT, EDT, BOT, CLT ...",
- "UTC-03:30 - NST, NT",
- "UTC-03:00 - ADT, ROTT, ART, BRT, CLST ...",
- "UTC-02:30 – NDT",
- "UTC-02:00 - FNT, GST, UYST",
- "UTC-01:00 - EGT, AZOST, CVT",
- "UTC+00:00 - GMT, UCT, UTC, WET, EGST",
- "UTC+01:00 - BST, CET, WEDT, WEST, DFT ...",
- "UTC+02:00 - CAT, CEDT, CEST, EET, HAEC ...",
- "UTC+03:00 - EAT, EEDT, EEST, FET, AST ...",
- "UTC+03:30 – IRST",
- "UTC+04:00 - AMT, AST, AZT, GET, GST ...",
- "UTC+04:30 – AFT",
- "UTC+05:00 - AMST, HMT, MAWT, MVT, PKT ...",
- "UTC+05:30 - IST, SLT",
- "UTC+05:45 – NPT",
- "UTC+06:00 - BIOT, BST, BTT, OMST, VOST",
- "UTC+06:30 - CCT, MMT, MST",

- "UTC+07:00 - CXT, DAVT, DDUT, HOVT, ICT...",
- "UTC+08:00 - WST, ACT, AWST, BDT, CT...",
- "UTC+08:45 – CWST",
- "UTC+09:00 - AWDT, JST, KST, TLT, YAKT",
- "UTC+09:30 - ACST, CST",
- "UTC+10:00 - EST, AEST, ChST, ChST, CHUT...",
- "UTC+10:30 - ACDT, CST, LHST",
- "UTC+11:00 - AEDT, KOST, LHST, MIST, NCT...",
- "UTC+11:30 – NFT",
- "UTC+12:00 - FJT, GILT, MAGT, MHT, NZST...",
- "UTC+12:45 – CHAST",
- "UTC+13:00 - NZDT, PHOT, TOT",
- "UTC+13:45 – CHADT",
- "UTC+14:00 - LINT, TKT".

2.2 API Method for Editing a Task by Task ID

Mandatory Fields are marked in red.

HTTP Method	Resource Parameters	Query Parameters	Description
PUT		<pre> { "taskId": id, "name": "Task Name", "agent": "Agent Name" *, "ranges": [{ "address": "DNS/IP/CIDR", "ports": "port, port, ..." }, { "address": "DNS/IP/CIDR", "ports": "port, port, ..." }] } </pre>	Enables Admin to edit a task by id.

		<pre> }], "rules": ["RuleName", ...], "frequency": "Frequency" **, "timeZone": "TimeZone" ***, "time": { "hours": "hour", "minutes": "minutes" } } </pre>	
--	--	--	--

Response

In case of success:

HTTP 200 - OK

In case of ranges partial/no match(for Agent = Auto), HTTP 400 with error message: *"There are no available agents for all the specified private ranges"*.

In case, a mandatory parameter was not provided, HTTP 400 with error message: *"[Parameter] cannot be empty"*.

In case, an invalid parameter was provided, HTTP 400 with error message: *"[Parameter] contains invalid value"*.

In case no task with such id was found, HTTP 400 with error message: *'Not Found [specified value]'*.

Example

HTTP PUT /api/discovery/v1/task

```
{
```

```
"taskId": 51,  
"name": "Test Task 2",  
"agent": "Agent 3",  
"ranges": [  
  {  
    "address": "10.100.10.15/32",  
    "ports": "443, 8080"  
  },  
  {  
    "address": "cert-manager.com",  
    "ports": "443-680"  
  }  
]  
"rules": [  
  "RuleForAWS"  
],  
"frequency": "Daily",  
"timeZone": "UTC+08:45 - CWST",  
"time": {  
  "hours": "10",  
  "minutes": "23"  
}  
}  
response:  
HTTP 200 - OK
```

* "agent" parameter can be Agent's name or 'Auto'.

** "frequency" parameter can have the following values:

- 'Manual',
- 'Daily',
- 'Weekly',
- 'Monthly',
- 'Quarterly',
- 'Semi-Annually',
- 'Annually'.

*** "timeZone" parameter can have the following values:

- "UTC-12:00 – BIT",
- "UTC-11:30 - NUT"
- "UTC-11:00 – SST",
- "UTC-10:00 - HAST, HST, TAHT, CKT",
- "UTC-09:30 - MART, MIT",
- "UTC-09:00 - AKST, GAMT, GIT, HADT",
- "UTC-08:00 - PST, CHOT, CIST, AKDT",
- "UTC-07:00 - MST, PDT",
- "UTC-06:00 - CST, EAST, GALT, MDT",
- "UTC-05:00 - CST, ORAT, PET, CHOT",
- "UTC-04:30 – VET",
- "UTC-04:00 - AST, ECT, EDT, BOT, CLT...",
- "UTC-03:30 - NST, NT",
- "UTC-03:00 - ADT, ROTT, ART, BRT, CLST...",
- "UTC-02:30 – NDT",
- "UTC-02:00 - FNT, GST, UYST",
- "UTC-01:00 - EGT, AZOST, CVT",
- "UTC+00:00 - GMT, UCT, UTC, WET, EGST",
- "UTC+01:00 - BST, CET, WEDT, WEST, DFT...",
- "UTC+02:00 - CAT, CEDT, CEST, EET, HAEC...",
- "UTC+03:00 - EAT, EEDT, EEST, FET, AST...",
- "UTC+03:30 – IRST",
- "UTC+04:00 - AMT, AST, AZT, GET, GST...",
- "UTC+04:30 – AFT",

- "UTC+05:00 - AMST, HMT, MAWT, MVT, PKT...",
- "UTC+05:30 - IST, SLT",
- "UTC+05:45 – NPT",
- "UTC+06:00 - BIOT, BST, BTT, OMST, VOST",
- "UTC+06:30 - CCT, MMT, MST",
- "UTC+07:00 - CXT, DAVT, DDUT, HOVT, ICT...",
- "UTC+08:00 - WST, ACT, AWST, BDT, CT...",
- "UTC+08:45 – CWST",
- "UTC+09:00 - AWDT, JST, KST, TLT, YAKT",
- "UTC+09:30 - ACST, CST",
- "UTC+10:00 - EST, AEST, ChST, ChST, CHUT...",
- "UTC+10:30 - ACDT, CST, LHST",
- "UTC+11:00 - AEDT, KOST, LHST, MIST, NCT...",
- "UTC+11:30 – NFT",
- "UTC+12:00 - FJT, GILT, MAGT, MHT, NZST...",
- "UTC+12:45 – CHAST",
- "UTC+13:00 - NZDT, PHOT, TOT",
- "UTC+13:45 – CHADT",
- "UTC+14:00 - LINT, TKT".

2.3 API Method for Getting Details of a Task by Task ID

HTTP Method	Resource Parameters	Query Parameters	Description
GET	taskid		Enables administrators to get the details of a task by task id.

Response

In case of success:

HTTP 200 - OK

```
{
  "name": "Task Name",
```

```
"agent": "Agent Name",  
  
"ranges": [  
  
  {  
  
    "address": "DNS/IP/CIDR",  
  
    "ports": "port, port, ..."  
  
  },  
  
  {  
  
    "address": "DNS/IP/CIDR",  
  
    "ports": "port, port, ..."  
  
  }  
  
],  
  
"rules": [  
  
  "RuleName",  
  
  ...  
  
],  
  
"frequency": "Frequency",  
  
"timeZone": "TimeZone",  
  
"time": {  
  
  "hours": "hour",  
  
  "minutes": "minutes"  
  
}  
  
}  
  
In case no task with such Id was found, HTTP 400 with error message:  
  
'Not Found [specified value]'.
```

Example

```
HTTP GET /api/discovery/v1/task/{taskId}
```

```
response:
```

```
HTTP 200 - OK
```

```
{
```

```
"name": "Test Task 2",
```

```
"agent": "Agent 3",
```

```
"ranges": [
```

```
{
```

```
"address": "10.100.10.15/32",
```

```
"ports": "443, 8080"
```

```
}
```

```
]
```

```
"rules": [
```

```
"RuleForAWS"
```

```
],
```

```
"frequency": "Daily",
```

```
"timeZone": "UTC+08:45 - CWST",
```

```
"time": {
```

```
"hours": "10",
```

```
"minutes": "23"
```

```
}
```

```
}
```

2.4 API Method for Getting Number of Existing Tasks

HTTP Method	Resource Parameters	Query Parameters	Description
-------------	---------------------	------------------	-------------

GET	count		Enables administrators to get the number of existing tasks.
------------	-------	--	---

Response
<p>In case of success:</p> <p>HTTP 200 - OK</p> <pre>{ "count": Number }</pre> <p>In case of error: HTTP code.</p>

Example
<p>HTTP GET /api/discovery/v1/task/count</p> <p>response:</p> <p>HTTP 200 - OK</p> <pre>{ "count": 1 }</pre>

2.5 API Method for Getting List of Existing Tasks

HTTP Method	Resource Parameters	Query Parameters	Description
GET	? position={position} & size={count_result}		Enables administrators to get the list of existing tasks using pagination. 'Position' and 'size' attributes are

			<p>optional.</p> <p>In case not specified, default size = 15, default position = 0.</p> <p>Negative values are substituted by their modulus. The max size value is 200.</p>
--	--	--	---

Response
<p>In case of success:</p> <p>HTTP 200 - OK</p> <pre>{ "ids": [id, id, id, ...] }</pre> <p>In case no values are found with such position, HTTP 200 with blank response.</p> <p>In case position value has invalid format: <i>HTTP 404 Not Found</i>.</p>

Example
<p>HTTP GET /api/discovery/v1/task?position=0&size=<size></p> <p>response:</p> <p>HTTP 200 - OK</p> <pre>{ "ids": [51, 52, 53] }</pre>

2.6 API Method for Starting Discovery Scan for a Specific task

Mandatory Fields are marked in red.

HTTP Method	Resource Parameters	Query Parameters	Description
-------------	---------------------	------------------	-------------

POST	start	<pre>{ "taskId": id }</pre>	Enables administrators to start a discovery scan for a specific task.
-------------	-------	-------------------------------	---

Response
<p>In case of success:</p> <p>HTTP 200 - OK</p> <p>In case, a mandatory parameter was not provided, HTTP 400 with error message: <i>"[Parameter] cannot be empty"</i>.</p> <p>In case, an invalid parameter was provided, HTTP 400 with error message: <i>"[Parameter] contains invalid value"</i>.</p> <p>In case no task with such Id was found, HTTP 400 with error message: <i>'Not Found [specified value]'</i>.</p>

Example
<p>HTTP POST /api/discovery/v1/task/start</p> <pre>{ "taskId": 51 }</pre> <p>response:</p> <p>HTTP 200 - OK</p>

2.7 API Method for Getting Status of a Specific Task

HTTP Method	Resource Parameters	Query Parameters	Description
GET	status/taskid		Enables administrators to get the status of a specific task.

Response

In case of success:

In case of success:

HTTP 200 - OK

```
{
  "status": "STATUS" *
}
```

In case no task with such Id was found, HTTP 400 with error message: *'Not Found [specified value]'*.

Example

HTTP GET /api/discovery/v1/task/status/{taskId}

response:

HTTP 200 - OK

```
{
  "status": "SUCCESSFUL"
}
```

2.8 API Method for Stopping Discovery Scan for a Specific Task

Mandatory Fields are marked in red.

HTTP Method	Resource Parameters	Query Parameters	Description
POST	stop	<pre>{ "taskId": id }</pre>	Enables administrators to stop Discovery scan for a specific task.

Response

In case of success:

HTTP 200 - OK

In case, a mandatory parameter was not provided, HTTP 400 with error message: "[Parameter] cannot be empty".

In case, an invalid parameter was provided, HTTP 400 with error message: "[Parameter] contains invalid value".

In case no task with such Id was found, HTTP 400 with error message: 'Not Found [specified value]'.

Example

HTTP POST /api/discovery/v1/task/stop

```
{
  "taskId": 51
}
```

response:

HTTP 200 - OK

2.9 API Method for Removing a Task by Task ID

HTTP Method	Resource Parameters	Query Parameters	Description
DELETE	taskId		Enables administrators to delete a specific task by id. Certificates in the network assets interface with status = 'Unmanaged' will be removed. Certificates processed by assignment rules or manually brought under management will remain.

Response

In case of success:

HTTP 200 - OK

In case no task with such Id was found, HTTP 400 with error message: 'Not Found [specified value]'.

In case taskId is not provided, HTTP 405 with error message: *'Method Not Allowed'*.

Example

HTTP DELETE /api/discovery/v1/task/{taskId}

response:

HTTP 200 - OK

3 Scan History

'Scan History' provides results from completed discovery tasks. Each response shows details of SSL certificates discovered by up to five previously completed scans.

1. Authentication via Username and Password

- Users should have InCommon CM login credentials and the correct customer login URI

The URI for the username/password authentication is:

- <https://cert-manager.com:443/api/discovery/v1/history>

2. Authentication via Username and a Client Certificate

- Admins should have 'Certificate Auth' enabled. The authentication certificate must requested and issued via InCommon CM and active at the moment of authentication.

The URI for the username/client certificate authentication is:

- <https://cert-manager.com:443/private/api/discovery/v1/history>

The Web Application Description Language (WADL) file can be accessed via the following URI:

- <https://cert-manager.com:443/api/application.wadl>

There are three methods available:

- [API Method for Getting Details of a Scan for a Specific Task](#)
- [API Method for Getting Details of Discovered SSL Certificates from Last Scan](#)
- [API Method for Getting Details of Discovered SSL Certificates from a Specific Scan](#)

3.1 API Method for Getting Details of a Scan for a Specific Task

HTTP Method	Resource Parameters	Query Parameters	Description
GET	taskId		Enables administrators to get complete scan history details for a specific discovery task, including the

			scan results and number of discovered SSL certificates.
--	--	--	---

Response

In case of success:

response:

HTTP 200 - OK

```
{
  "scanResults": [
    {
      "scanResultId": id,
      "date": "mm/dd/yyyy hour:minutes:seconds timeZone",
      "status": "STATUS",
      "sslCertsFound": number
    },
    {
      "scanResultId": id,
      "date": "mm/dd/yyyy hour:minutes:seconds timeZone",
      "status": "STATUS",
      "sslCertsFound": number
    }
  ]
}
```

In case no task with such Id was found, HTTP 400 with error message: *'Not Found [specified value]'*.

Example

```

HTTP GET /api/discovery/v1/history/{taskId}

response:

HTTP 200 – OK

{
  "scanResults": [
    {
      "scanResultId": 52,
      "date": "05/26/2017 17:20:17 GMT",
      "status": "SUCCESSFUL",
      "sslCertsFound": 50
    },
    {
      "scanResultId": 53,
      "date": "05/27/2017 14:20:17 GMT",
      "status": "FAILED",
      "sslCertsFound": 0
    }
  ]
}

```

3.2 API Method for Getting Details of Discovered SSL Certificates from Last Scan

HTTP Method	Resource Parameters	Query Parameters	Description
GET	lastresult/{taskId}? position={position} & size={count_result}		Enables administrators to get details of discovered SSL certificates from the last run scan. The max number (size) of

			<p>certificates that can be fetched is 200.</p> <p>Default size is 15 and the default starting position is 0. 'Position' and 'size' attributes are optional.</p> <p>For one certificate, only one host name or IP address will be displayed.</p>
--	--	--	--

Response

HTTP 200 - OK

```
{
```

```
"scanResults":
```

```
[
```

```
{
```

```
"ipAddress": "ipAddress",
```

```
"hostname": "hostName",
```

```
"commonName": "commonName",
```

```
"validTo": "mm/dd/yyyy hours:minutes:seconds timeZone",
```

```
"issuer": "Issuer Info",
```

```
"subject": "subject",
```

```
"validFrom": "mm/dd/yyyy hours:minutes:seconds timeZone",
```

```
"subjectAltName": "domain name",
```

```
"keyAlgorithm": "algorithm",
```

```
"keySize": size,
```

```
"signatureAlgorithm": "algorithm",
```

```
"inventory": " ",
```

```
"serialNumber": "serialNumber",  
"md5Fingerprint": "MD5 value",  
"sha1Fingerprint": "SHA1 value",  
"cipher": "TLS_ECH_AES_128_CBC_SHA256",  
"keyUsage": "Key Usage",  
extendedKeyUsage: "Extended Key Usage"  
}  
]  
}
```

If no task was found with the Id then you will see a HTTP 400 error with the message: *'Not Found [specified value]*.

If no values are found for the position then you will see a HTTP 200 error with blank response.

Example

HTTP GET /api/discovery/v1/history/result/{scanResultId}?position=0&size=<1>

response:

HTTP 200 - OK

```
{  
  [  
    {  
      "ipAddress": "10.10.10.01",  
      "hostname": "host",  
      "commonName": "qwerty.com",  
      "validTo": "05/26/2018 23:59:59 GMT",  
      "issuer": "CN=COMODO RSA Organization Validation Secure Server CA,O=COMODO CA
```



```

Limited,L=Salford,ST=Greater Manchester,C=GB",

"subject": "CN=*.ccmqa.com, OU=PremiumSSL Wildcard,OU=Hosted by Comodo Certificate Manager Demo, OU=QA
Odessa, O=QA_ccm_Odessa, STREET=Bazarnaya 63,L=Odessa,ST=Odessa,PostalCode=65000,C=UA",

validFrom": "05/26/2017 00:00:00 GMT",

"subjectAltName": "ccmqa.com",

"keyAlgorithm": "RSA",

"keySize": 2048,

"signatureAlgorithm": "SHA256withRSA",

"inventory": "",

"serialNumber": "569989854",

"md5Fingerprint": "a3629c93ec6bb5db79e3d91279854de4",

"sha1Fingerprint": "590ad2ab0808560926dfc1d2c42d547241000a1e",

"cipher": "TLS_ECDHE_RSA_WITH_AES_128_CBC_SHA256",

"keyUsage": "Digital Signature\nKey Encipherment",

"extendedKeyUsage": "1.3.6.1.5.5.7.3.1\n1.3.6.1.5.5.7.3.2"

}

```

3.3 API Method for Getting Details of Discovered SSL Certificates from a Specific Scan

HTTP Method	Resource Parameters	Query Parameters	Description
GET	/result/{scanResultId}? position={position} & size={count_result}		<p>Enables administrators to get the details of discovered SSL certificates from a specific scan indicated by its ID.</p> <p>The max number (size) of certificates that can be fetched is 200.</p> <p>Default size is 15 and the default starting position is 0.</p>

			<p>'Position' and 'size' attributes are optional.</p> <p>For one certificate, only one host name or IP address will be displayed.</p>
--	--	--	---

Response

In case of success:

response:

HTTP 200 - OK

```
{
  "scanResults":
  [
    {
      "ipAddress": "ipAddress",
      "hostname": "hostName",
      "commonName": "commonName",
      "validTo": "mm/dd/yyyy hours:minutes:seconds timeZone",
      "issuer": "Issuer Info",
      "subject": "subject",
      "validFrom": "mm/dd/yyyy hours:minutes:seconds timeZone",
      "subjectAltName": "domain name",
      "keyAlgorithm": "algorithm",
      "keySize": size,
      "signatureAlgorithm": "algorithm",
      "inventory": "Order Number ",
      "serialNumber": "serialNumber",
```

```
"md5Fingerprint": "MD5 value",
"sha1Fingerprint": "SHA1 value",
"cipher": "TLS_ECH_AES_128_CBC_SHA256",
"keyUsage": "Key Usage",
extendedKeyUsage": "Extended Key Usage"
}
]
}
```

If no task was found with the Id then you will see a HTTP 400 error with the message: *'Not Found [specified value]*.

If no values are found for the position then you will see a HTTP 200 error with blank response.

Example

HTTP GET /api/discovery/v1/history/result/{scanResultId}?position=0&size=<1>

response:

HTTP 200 - OK

```
{
  [
    {
      "ipAddress": "10.10.10.01",
      "hostname": "host",
      "commonName": "qwerty.com",
      "validTo": "05/26/2018 23:59:59 GMT",
      "issuer": "CN=COMODO RSA Organization Validation Secure Server CA,O=COMODO CA
Limited,L=Salford,ST=Greater Manchester,C=GB",
```

```
"subject": "CN=*.ccmqa.com, OU=PremiumSSL Wildcard,OU=Hosted by Comodo Certificate Manager Demo, OU=QA Odessa, O=QA_ccm_Odessa, STREET=Bazarnaya 63,L=Odessa,ST=Odessa,PostalCode=65000,C=UA",
"validFrom": "05/26/2017 00:00:00 GMT",
"subjectAltName": "ccmqa.com",
"keyAlgorithm": "RSA",
"keySize": 2048,
"signatureAlgorithm": "SHA256withRSA",
"inventory": "",
"serialNumber": "569989854",
"md5Fingerprint": "a3629c93ec6bb5db79e3d91279854de4",
"sha1Fingerprint": "590ad2ab0808560926dfc1d2c42d547241000a1e",
"cipher": "TLS_ECDHE_RSA_WITH_AES_128_CBC_SHA256",
"keyUsage": "Digital Signature\nKey Encipherment",
"extendedKeyUsage": "1.3.6.1.5.5.7.3.1\n1.3.6.1.5.5.7.3.2"
}
]
}
```

4 Auto-Assignment Rules

Assignment Rules are associated with discovery tasks to assign 'Unmanaged' certificates (those not issued by CCM) to a particular Organization or Department.

You can access the assignment rules screen by authenticating yourself using one of the following two methods:

1. Authentication via Username and Password

- Users should have InCommon CM login credentials and the correct customer login URI

The URI for the username/password authentication is:

- <https://cert-manager.com:443/api/discovery/v1/assignmentrule>

2. Authentication via Username and a Client Certificate

- Admins should have 'Certificate Auth' enabled. The authentication certificate must be requested and issued via InCommon CM and active at the moment of authentication.

The URI for the username/client certificate authentication is:

- <https://cert-manager.com:443/private/api/discovery/v1/assignmentrule>

The Web Application Description Language (WADL) file can be accessed via the following URI:

- <https://cert-manager.com:443/api/application.wadl>

There are six API methods available:

- [API Method for Creating a New Rule](#)
- [API Method for Editing a Rule](#)
- [API Method for Getting Number of Existing Rules](#)
- [API Method for Getting the List of Existing Rules](#)
- [API Method for Getting Rule Details](#)
- [API Method for Removing a Rule](#)

4.1 API Method for Creating a New Rule

Mandatory Fields are marked in red.

HTTP Method	Resource Parameters	Query Parameters	Description
POST		<pre>{ "name": "Rule Name", "orgId" ***: number, "filters": [{ "filterType": "FILTER_TYPE" *, "matchType": "MATCH_TYPE" **, "value": "value" }, { "filterType": "FILTER_TYPE" *, "matchType": "MATCH_TYPE" **, "value": "value" }] }</pre>	Enables administrators to create a new auto-assignment rule.

		}	
--	--	---	--

Response

In case of success:

HTTP 200 – OK

```
{  
  "ruleId": id  
}
```

If a mandatory parameter is missing you will see a HTTP 400 error with the message: "*[Parameter] cannot be empty*".

If an invalid parameter was provided you will see a HTTP 400 error with the message: "*[Parameter] contains invalid value*".

Example

HTTP POST /api/discovery/v1/assignmentrule

```
{  
  "name": "Rule_1",  
  "orgId": 2,  
  "filters": [  
    {  
      "filterType": "COMMON_NAME",  
      "matchType": "MATCHES",  
      "value": "*"   
    },  
    {  
      "filterType": "ORGANIZATION",
```

```
"matchType": "CONTAINS",  
"value": "Organization API"  
}  
]  
}
```

response:

HTTP 200 – OK

```
{  
"ruleId": 51  
}
```

* "filterType" available values:

- "COMMON_NAME",
- "ORGANIZATION",
- "ORGANIZATION_UNIT",
- "CITY",
- "STATE",
- "COUNTRY",
- "SUBJECT_ALT_NAME",
- "ISSUER",
- "IP",
- "PORT".

** "matchType" available values:

- "MATCHES",
- "STARTS_WITH",
- "ENDS_WITH",
- "CONTAINS",
- "MATCH_REGEX".

*** "orgId" can be either Organization ID or Department ID.

4.2 API Method for Editing a Rule

Mandatory Fields are marked in red.

HTTP Method	Resource Parameters	Query Parameters	Description
PUT		<pre>{ "ruleId": number, "name": "Rule Name", "orgId" ***: number, "filters": [{ "filterType": "FILTER_TYPE" *, "matchType": "MATCH_TYPE" **, "value": "value" }, { "filterType": "FILTER_TYPE" *,, "matchType": "MATCH_TYPE" **, "value": "value" }] }</pre>	<p>Enables admins to edit an auto-assignment rule by specifying a rule id.</p>

Response

In case of success - HTTP 200 - OK

If a mandatory parameter is missing you will see a HTTP 400 error with the message: "[Parameter] cannot be empty".

If an invalid parameter was provided you will see a HTTP 400 error with the message: "[Parameter] contains invalid value".

If no rule was found with the supplied Id you will see a HTTP 400 error with the message: 'Not Found [specified value]'.

Example

HTTP PUT /api/discovery/v1/assignmentrule

```
{
  "ruleId": 52,
  "name": "Rule_1",
  "orgId": 2,
  "filters":
  [
    {
      "filterType": "COMMON_NAME",
      "matchType": "MATCHES",
      "value": "*"
    },
    {
      "filterType": "ORGANIZATION",
      "matchType": "CONTAINS",
      "value": "ddd"
    }
  ]
}
```

response:

HTTP 200 - OK

* "filterType" available values:

- "COMMON_NAME",
- "ORGANIZATION",

- "ORGANIZATION_UNIT",
- "CITY",
- "STATE",
- "COUNTRY",
- "SUBJECT_ALT_NAME",
- "ISSUER",
- "IP",
- "PORT".

** "matchType" available values:

- "MATCHES",
- "STARTS_WITH",
- "ENDS_WITH",
- "CONTAINS",
- "MATCH_REGEX".

*** "orgId" can be either Organization ID or Department ID.

4.3 API Method for Getting Number of Existing Rules

HTTP Method	Resource Parameters	Query Parameters	Description
GET	count		Enables administrators to get the number of existing auto-assignment rules.

Response

In case of success:

HTTP 200 - OK

```
{
  "count": 5
}
```

In case of error: appropriate HTTP code.

Example

HTTP GET /api/discovery/v1/assignmentrule/count

response:

HTTP 200 - OK

```
{
  "count": 4
}
```

4.4 API Method for Getting the List of Existing Rules

HTTP Method	Resource Parameters	Query Parameters	Description
GET	? position=<position>&size=<count_result>		<p>Enables admins to get a list of existing auto-assignment rules using pagination. 'Position' and 'size' attributes are optional.</p> <p>In case not specified, default size = 15, default position = 0.</p> <p>Negative values are substituted by their modulus. The max size value is 200.</p>

Response

In case of success:

HTTP 200 - OK

```
{
  "ids": [id, id, id, ...]
```

```
}

In case no values are found with such position, HTTP 200 with blank response.
```

Example

```
HTTP GET /api/discovery/v1/assignmentrule?position=0&size=<3>

response:

HTTP 200 - OK

{

"ids": [51, 52, 53]

}
```

4.5 API Method for Getting Rule Details

HTTP Method	Resource Parameters	Query Parameters	Description
GET	ruleid		Enables admins to view an auto-assignment rule by specifying its ID.

Response

```
In case of success:

HTTP 200 - OK

{

"name": "Rule Name",

"orgId": number,

"filters": [

{

"filterType": "FILTER_TYPE" *,
```

```
"matchType": "MATCH_TYPE" **,
"value": "value"
},
{
"filterType": "FILTER_TYPE" *,,
"matchType": "MATCH_TYPE" **,
"value": "value"
}
]
}
```

In case no rule with such Id has been found, HTTP 400 with error message:

'Not Found [specified value].

Example

HTTP GET /api/discovery/v1/assignmentrule/{ruleId}

response:

HTTP 200 - OK

```
{
"name": "Rule_1",
"orgId": 2,
"filters": [
{
"filterType": "COMMON_NAME",
"matchType": "MATCHES",
"value": "**"
```

```

},
{
  "filterType": "ORGANIZATION",
  "matchType": "CONTAINS",
  "value": "Organization API"
}
]
}

```

4.6 API Method for Removing a Rule

HTTP Method	Resource Parameters	Query Parameters	Description
DELETE	ruleid		Enables administrators to delete a specific auto-assignment rule by specifying its rule id.

Response
<p>In case of success:</p> <p>HTTP 200 - OK</p> <p>In case no rule with such id was found, HTTP 400 with error message: <i>'Not Found [specified value]'</i>.</p>

Example
<p>HTTP DELETE /api/discovery/v1/assignmentrule/{ruleId}</p> <p>response:</p> <p>HTTP 200 - OK</p>