Documentation for the DigiCert Retail API

Oct 25, 2014

Table of Contents

Introduction

API Basics

All API requests are submitted via REST-based URLs using REST features including HEADER-based authentication and JSON/XML request types.

The requests will be accepted through port 443, which is the default port for SSL. The character set encoding of data in API calls and responses is UTF-8. To ensure a well-formed request, make sure that the `USER-AGENT` and `CONTENT-LENGTH` headers are specified in the request.

Authentication

The REST API requires access to the existing Digicert APIs and utilizes the same keys. However to authenticate with the service, it uses **Basic HTTP Authentication over SSL**.

As basic authentication is an established standard, it is easy to implement. Using cURL:

```
curl -u "account_id:api_key" ...
```

Building the header manually:

1. Combine the account id and api key into a string: `account_id:api_key`
2. Base64 encode the string
3. Add the encoded string to the header: `Authorization: Basic {BASE64_ENCODED_STRING}`

So this is also valid in cURL:

```
curl -H "Authorization: Basic {BASE64_ENCODED_STRING}"
```

Requests

Requests consist of two or three parts: Method, Endpoint, and Body. Every request has a method and endpoint, but some also include a body. The body is formatted based on the `CONTENT-TYPE` header sent.

Method

The API uses four standard HTTP methods and several custom methods. The standard methods supported are `GET`, `HEAD`, `OPTIONS`, and `POST`. The custom methods are `APPROVE`, `REISSUE`, `REJECT`, `RENEW`, and `REVOKE`. The client can use any of these methods directly. This is the preferred method. To use with cURL:

```
curl -u "account_id:api_key" -X {CUSTOM_METHOD}...
```

So, to revoke a request:

```
curl -u "account_id:api_key" -X REVOKE ...
```

Some client libraries may not allow custom HTTP methods. For these libraries, there is an additional way to set the method for a request: use a custom HTTP header.

To use:

1. Set the HTTP method as `POST`.
2. Create a header named `X-HTTP-METHOD-OVERRIDE` and set the value to be one of the custom methods.

To use with cURL:

```
curl -u "account_id:api_key" -X POST -H "X-HTTP-Method-Override: REVOKE"...
```

Endpoint

REST requests use endpoints to retrieve and alter data. Endpoints are generally split into two types:

1. *Collection*: Represents a group of items. Examples include `/order` where `POST` creates a new item and `GET` returns a list of items.
   Item: Represents a single item. These endpoints usually have an id attached at the end of the URL. Examples include `/order/00112233` where `POST` affects the particular order and `GET` returns details about the order.

Specific endpoints will be described later when explaining how to make specific requests.

Body and Content Type

Most requests require passing in either `JSON` or `XML` formatted data. Some requests may require standard content types like `image/jpeg`. If an endpoint supports or requires a format other than `JSON` or `XML`, it will be noted in the documentation. Generally `GET` requests are the only ones that do not require sending formatted data. With some endpoints, `GET` requests will support passing in additional parameters to filter the results.

For requests that require additional data, instead of using the generic `application/json` and `application/xml` content types, the API supports custom content types that also include a version number.

The following content types are currently supported:

1. `application/vnd.digicert.rest-v1+xml`
2. `application/vnd.digicert.rest-v1+json`

To use, create a header:

```
Content-Type: application/vnd.digicert.rest-v1+json
```

A full cURL request may be something like:

```
curl -u "account_id:api_key" -X POST \
-H "Content-Type: application/vnd.digicert.rest-v1+json" \
--data '{"foo":"bar"}' https://api.digicert.com/fez/00112233
```

Responses

Responses consist of two parts: Headers and Body. The body is formatted based on the content type requested in the `ACCEPT` header.

Headers

Every response will include an appropriate HTTP header. This header is based off of the RFC 2616 specification.

Code	Description
200	General success response
201	Created: Useful for creation of requests, orders, etc
204	No Content: For successful requests that don't require a response
301	Moved Permanently: Returned in the unlikely event that a URL has changed. Will also return a `LOCATION` header with new URL. Clients should resubmit this request and submit future requests to this new URL
302	Moved Temporarily: Returned in the unlikely event that a URL has changed temporarily. Will also return a `LOCATION` header with new URL. Clients should resubmit this single request to this new URL
304	Content not modified: Useful when accessing a URL while waiting for a response. Only used if an `IF-NONE-MATCH` header was passed
400	General client error

Code	Description
401	Unauthorized: Returned if the page is accessed without a valid account/API Key combination
403	User doesn't have permission to perform the requested action
404	Returned if the page doesn't exist or the API doesn't have permission to interact with a particular item
405	Method not found. Returned if an endpoint doesn't support the method that was passed. It will also return an `ALLOW` header with supported methods
406	If the client requests an incompatible response (i.e. the client wants a v1 response for an endpoint only available in v2+), or the client doesn't specify a valid acceptable content-type
500	Unexpected behavior that the API couldn't recover from
503	The system is currently unavailable

For most of the `GET` requests, the API will also return an `ETAG` header so that clients can also pass in an `IF-NONE-MATCH` header.

Body and Content Type

For simplicity, the API support the same content types for both responses and requests.

For `GET` requests, the client does not send additional data to the API. For these requests, it is more appropriate to send an `ACCEPT` header than a `CONTENT-TYPE` header. However, if a `CONTENT-TYPE` header is passed and not an `ACCEPT` header (or the header is `Accept: */*`), the API will default to using the value set in the `CONTENT-TYPE`.

For other requests, a client only needs to send an `ACCEPT` header if the request is sent in one format and the response should be sent in a different one (i.e. the request is a `JSON`-encoded string, but the response needs to be in `XML`).

If the API cannot determine which format the client wants, it will return a 406 header.

Successful Response

All success responses will return an HTTP code in the 200 – 399 range. Clients receiving these headers will know that their request was successfully handled and will expect an appropriate response.

As an example, a client submits a `POST` request to the `/enterprise/certificate/ssl` endpoint. It will expect a `request_id` that could used in subsequent Approval Requests. It would expect a response similar to below:

JSON (application/vnd.digicert.rest--)	XML (application/vnd.digicert.rest-v1+xml)

`{`
`  "request_id":7902`
`}`
	`<response>`
	`  <request_id>7902</request_id>`
	`</response>`

Error Response

Error responses will return an HTTP code in 400+ range. Clients receiving these headers know that their request failed for various reasons. The API should always at least return a 500 error if there was a fatal error that wasn't caught. Most error responses return a special content type, based on the initial type being requested: `application/vnd.digicert.error+xml` for XML based requests, and `application/vnd.digicert.error+json` for JSON based. This content type is based on emerging standards.

For instance, a client wants to reject a pending request, but forgot to include the required NOTE. They would make a `REJECT` request to the `/request/7902` endpoint, and would receive a 400

response with the following body:

JSON (application/vnd.digicert.error+json)	XML (application/vnd.digicert.error+xml)

{
  "errors":[
  {
  "code":"required_param:note",
  "description":"Note is required"
  }
  ]
}
	<errors>
	  <error>
	  <code>required_param:note</code>
	  <description>Note is required</description>
	  </error>
	</errors>

Endpoint Overview

HEAD Requests

Any endpoint that supports a `GET` method also supports the `HEAD` method. As defined by RFC 2616, a `HEAD` request is treated the same as a `GET` request, except that it doesn't contain a response body.

OPTIONS Requests

All endpoints support an `OPTIONS` method call. The `OPTIONS` method as defined in RFC 2616 returns the same `ALLOW` header as a 405 error response.

Order Management

Overview

Ordering a Certificate

Ordering a certificate through the API is a multi-step process. First a pending certificate order will be created. The pending order will be validated by DigiCert and then the certificate will be issued. After the certificate has been issued, you can retrieve it using the order id and it will be returned through the API as a PEM encoded x509 certificate (base64).

The order endpoints may send email to the appropriate users. Some of the emails that are sent have action items that must be completed before the certificate can be issued.

Method	Endpoint	Action
POST	/order/ssl	Create a pending order for a new SSL Certificate
POST	/order/code	Create a pending order for a code signing certificate
GET	/order/{order_id}	View order status and other details
GET	/order/{order_id}/certificate	Retrieve the certificate once the order has been processed and certificate issued

Reissuing or Rekeying an SSL Certificate

Reissuing or rekeying a certificate involves multiple steps. First a request to either rekey or reissue the certificate must be submitted. Once submitted, the certificate will be queued to be reissued. Once the certificate has been reissued, you can retrieve it using the order id and it will be returned through the API as a PEM encoded x509 certificate (base64).

Method	Endpoint	Action
REISSUE	/order/{order_id}	Rekeying a certificate is a simplified form of reissuing, where the key is the only change being made. The reissue request supports either a simplified rekeying or a standard reissue.
GET	/order/{order_id}	View order status and other details
GET	/order/{order_id}/certificate	Retrieve the certificate once the order has been processed and certificate issued

Order an SSL Certificate

Request Endpoint

Method	URL
POST	https://api.digicert.com/order/ssl

Request Parameters

Parameters are encoded in one of the content types that is accepted by the endpoint.

Parameter Name	Req/Opt	Allowed Values	Default	Description
csr	Required	[string]		Certificate Signing Request. To create a CSR from your server, visit DigiCert's website at http://www.digicert.com/csr-creation.htm for details.
validity	Required	1, 2, 3		Certificate validity length in years. For EV certificates, only 1 and 2 year lengths are allowed.
common_name	Optional	[string]	[decoded from CSR]	The name to be secured in the certificate. A sample common name: example.com
sans	Optional	[string]	[decoded from CSR]	Subject Alternative Names. A comma delimited list of names to be secured in the certificate. Adding SANs names incurs additional cost.
server_type	Optional	see Server Types	-1	Server platform. Defaults to Other.
signature_hash	Optional	sha1, sha256	sha1	The certificate's signing algorithm hash
org_unit	Optional	[string]	[blank]	This is the OU field to be included in the certificate.
org_name	Required	[string]		Name of organization which owns common name.
org_addr1	Required	[string]		Address line 1 of the organization's address.
org_addr2	Optional	[string]	[blank]	Address line 2 of the organization's address.
org_city	Required	[string]		City of organization's address.
org_state	Required	[string]		State / Province of the organization's address.
org_zip	Required	[string]		Zip or Postal code of the organization's address.
org_country	Required	see Country Codes		A two character, lowercase abbreviation for country.
telephone	Optional**	[numeric string]	[blank]	Main telephone number of the organization.
org_contact_job_title	Optional**	[string]	[blank]	Title of contact working at the organization.
org_contact_firstname	Required	[string]		First name of the organization contact.
org_contact_lastname	Required	[string]		Last name of the organization contact.

Parameter Name	Req/Opt	Allowed Values	Default	Description
org_contact_email	Required	[email address]		Email address of the organization contact.
org_contact_telephone	Required	[numeric string]		Telephone number of the organization contact.
org_contact_telephone_ext	Optional	[numeric string]	[blank]	Extension number of the organization contact.
ev	Optional	[boolean]	false	Whether this is an EV certificate request or not.

** *telephone and org_contact_job_title fields are required for EV certificates.*

Response

Parameter Name	Data Type	Description
order_id	[integer]	The unique identifier of the certificate order related to the request

Sample Request

Endpoint

```
https://api.digicert.com/order/ssl
```

Headers

```
Authorization: Basic MDAxMDA3OnNreWZhbGw=
Content-Length: 496
Content-Type: application/vnd.digicert.rest-v1+json
User-Agent: MyAPIConsumer/0.42
```

Body

```
{
  "server_type": "2",
  "validity": "2",
  "common_name": "www.example.com",
  "org_name": "Tampa Bay Cray LLC",
  "org_addr1": "1234 Sesame Street",
  "org_city": "Tampa",
  "org_state": "FL",
  "org_zip": "33615",
  "org_country": "US",
  "org_contact_firstname": "Joe",
  "org_contact_lastname": "Bloggs",
  "org_contact_email": "joe@example.com",
  "org_contact_telephone": "555-123-1234",
  "csr": "-----BEGIN CERTIFICATE REQUEST-----\n[CSR GOES HERE]\n-----END
CERTIFICATE REQUEST-----"
}
```

Sample Response

Note that with this sample request, it would return `JSON`. Including `XML` for comparison.

Status Code: 201

Headers

```
Content-Length: 25
Content-Type: application/vnd.digicert.rest-v1+json
```

Body

JSON (application/vnd.digicert.rest--)	XML (application/vnd.digicert.rest-v1+xml)

<pre>{
  "order_id":00222307
}</pre>
	<pre><response>
	  <order_id>00222307</order_id>
	</response></pre>

Order a Code Signing Certificate

Request Endpoint

Method	URL
POST	https://api.digicert.com/order/code

Request Parameters

Parameters are encoded in one of the content types that is accepted by the endpoint.

Parameter Name	Req/Opt	Allowed Values	Default	Description
validity	Required	1, 2, 3		Certificate validity length in years.
platform	Required*	see EV CS Platforms, CS Platforms		The platform or hardware token on which your certificate will be installed. Only optional if you choose to have a token shipped to you.
cs_provisioning_method	Optional*	'ship_token', 'client_app'		For EV Code Signing Certificates. Request a hardware token, or install yourself using our tool. See EV CS Platforms.
cs_ship_method	Optional*	'standard', 'expedited'	'standard'	Shipping method for hardware token if ordered
ship_name	Optional**	[string]		Name of person hardware token should be shipped to.
ship_addr1	Optional**	[string]		Shipping address line 1 for hardware token
ship_addr2	Optional	[string]	[blank]	Shipping address line 2 for hardware token
ship_city	Optional**	[string]		Shipping city for hardware token
ship_state	Optional**	[string]		Shipping state for hardware token
ship_zip	Optional**	[string]		Shipping zip code for hardware token
ship_country	Optional**	[string]		Shipping country for hardware token
org_unit	Optional	[string]	[blank]	This is the OU field to be included in the certificate.
org_name	Required	[string]		Name of organization which owns common name.
org_addr1	Required	[string]		Address line 1 of the organization's address.
org_addr2	Optional	[string]	[blank]	Address line 2 of the organization's address.

Parameter Name	Req/Opt	Allowed Values	Default	Description
org_city	Required	[string]		City of organization's address.
org_state	Required	[string]		State / Province of the organization's address.
org_zip	Required	[string]		Zip or Postal code of the organization's address.
org_country	Required	see Country Codes		A two character, lowercase abbreviation for country.
telephone	Required	[numeric string]	[blank]	Main telephone number of the organization.
org_contact_job_title	Required	[string]	[blank]	Title of contact working at the organization.
org_contact_firstname	Required	[string]		First name of the organization contact.
org_contact_lastname	Required	[string]		Last name of the organization contact.
org_contact_email	Required	[valid email address]		Email address of the organization contact.
org_contact_telephone	Required	[numeric string]		Telephone number of the organization contact.
org_contact_telephone_ext	Optional	[numeric string]	[blank]	Extension number of the organization contact.
ev	Optional	'true' or 'false'	'false'	Whether this is an EV certificate request or not.

** Fields are required in some cases, not allowed in others. See descriptions.*
*** Shipping address fields are all required if a hardware token is being shipped, unless none are present, in which case the token will be shipped to the organization address.*

Response

Parameter Name	Data Type	Description
order_id	[integer]	The unique identifier of the certificate order related to the request

Sample Request

Endpoint

```
https://api.digicert.com/order/code
```

Headers

```
Authorization: Basic MDAxMDA3OnNreWZhbGw=
Content-Length: 488
Content-Type: application/vnd.digicert.rest-v1+json
User-Agent: MyAPIConsumer/0.42
```

Body

```
{
  "validity": "1",
  "platform": "2",
  "cs_provisioning_method": "client_app",
```

```
  "org_name": "My Company",
  "org_addr1": "19 Windsor Court",
  "org_addr2": "",
  "org_city": "Tampa",
  "org_state": "FL",
  "org_zip": "33615",
  "org_country": "US",
  "telephone": "555-555-1234",
  "org_contact_firstname": "Gary",
  "org_contact_lastname": "McCoy",
  "org_contact_job_title": "CTO",
  "org_contact_email": "gm@example.com",
  "org_contact_telephone": "555-555-1234",
  "ev": "true"
}
```

Sample Response

Note that with this sample request, it would return `JSON`. Including `XML` for comparison.

Status Code: 201

Headers

```
Content-Length: 25
Content-Type: application/vnd.digicert.rest-v1+json
```

Body

JSON (application/vnd.digicert.rest--)	XML (application/vnd.digicert.rest-v1+xml)
`{ "order_id":00222307 }`	`<response> <order_id>00222307</order_id> </response>`

Order a Wifi Certificate

Request Endpoint

Method	URL
POST	https://api.digicert.com/order/wfa

Request Parameters

Parameters are encoded in one of the content types that is accepted by the endpoint.

Parameter Name	Req/Opt	Allowed Values	Default	Description
csr	Required	[string]		Certificate Signing Request. To create a CSR from your server, visit DigiCert's website at http://www.digicert.com/csr-creation.htm for details.
validity	Required	1, 2, 3		Certificate validity length in years. For EV certificates, only 1 and 2 year lengths are allowed.
common_name	Optional	[string]	[decoded from CSR]	The name to be secured in the certificate. A sample common name: example.com
sans	Optional	[string]	[decoded from CSR]	Subject Alternative Names. A comma delimited list of names to be secured in the certificate. Adding SANs names incurs additional cost.
server_type	Optional	see Server Types	-1	Server platform. Defaults to Other.
org_unit	Optional	[string]	[blank]	This is the OU field to be included in the certificate.
org_name	Required	[string]		Name of organization which owns common name.
org_addr1	Required	[string]		Address line 1 of the organization's address.
org_addr2	Optional	[string]	[blank]	Address line 2 of the organization's address.
org_city	Required	[string]		City of organization's address.
org_state	Required	[string]		State / Province of the organization's address.
org_zip	Required	[string]		Zip or Postal code of the organization's address.
org_country	Required	see Country Codes		A two character, lowercase abbreviation for country.
telephone	Optional**	[numeric string]	[blank]	Main telephone number of the organization.
org_contact_job_title	Optional**	[string]	[blank]	Title of contact working at the organization.
org_contact_firstname	Required	[string]		First name of the organization contact.
org_contact_lastname	Required	[string]		Last name of the organization contact.
org_contact_email	Required	[email address]		Email address of the organization contact.
org_contact_telephone	Required	[numeric string]		Telephone number of the organization contact.

Parameter Name	Req/Opt	Allowed Values	Default	Description
org_contact_telephone_ext	Optional	[numeric string]	[blank]	Extension number of the organization contact.
default_logo_url	Required	[string]		The URL of a publicly-accessible logo PNG, JPG, or GIF image to be included in this certificate. Changing the image after requesting the certificate will result in a broken certificate.
additional_logos	Optional	[collection]		A set of additional logos for different languages.
language	Required	See ISO 639-2 Languages		A valid ISO 639-2 language code.
url	Required	[string]		See description for default_logo_url.
friendly_names	Required	[collection]		A set of friendly names for different languages. At least one friendly name is required.
language	Required	See ISO 639-2 Languages		A valid ISO 639-2 language code.
text	Required	[string]		A language-specific friendly name for this wifi certificate.

Response

Parameter Name	Data Type	Description
order_id	[integer]	The unique identifier of the certificate order related to the request

Sample Request

Endpoint

```
https://api.digicert.com/order/wfa
```

Headers

```
Authorization: Basic MDAxMDA3OnNreWZhbGw=
Content-Length: 975
Content-Type: application/vnd.digicert.rest-v1+json
User-Agent: MyAPIConsumer/0.42
```

Body

```
{
  "server_type": "2",
  "validity": "2",
  "common_name": "www.example.com",
  "org_name": "Tampa Bay Cray LLC",
  "org_addr1": "1234 Sesame Street",
  "org_city": "Tampa",
  "org_state": "FL",
  "org_zip": "33615",
  "org_country": "US",
  "org_contact_firstname": "Joe",
  "org_contact_lastname": "Bloggs",
  "org_contact_email": "joe@example.com",
  "org_contact_telephone": "555-123-1234",
```

```
  "csr": "-----BEGIN CERTIFICATE REQUEST-----\n[CSR GOES HERE]\n-----END
CERTIFICATE REQUEST-----",
  "default_logo_url": "http://www.example.com/wifi-logos/default.png",
  "additional_logos": [
    {
        "language": "spa",
        "url": "http://www.example.com/wifi-logos/spanish.png"
    },
    {
        "language": "deu",
        "url": "http://www.example.com/wifi-logos/german.png"
    }
  ],
  "friendly_names": [
    {
        "language": "spa",
        "text": "Wifi Publica"
    },
    {
        "language": "kor",
        "text": "□□ □□□□"
    }
  ]
}
```

Sample Response

Note that with this sample request, it would return `JSON`. Including `XML` for comparison.

Status Code: 201

Headers

```
Content-Length: 25
Content-Type: application/vnd.digicert.rest-v1+json
```

Body

JSON (application/vnd.digicert.rest--)	XML (application/vnd.digicert.rest-v1+xml)
{ "order_id":00222307 }	<response> <order_id>00222307</order_id> </response>

Rekey or Reissue an SSL Certificate

Rekeying is the process by which the private and public key is changed for a certificate. It is a simplified reissue, where only the CSR Is changed.

Reissuing is the process by which domain names are added or removed from a certificate. Once a request is validated and approved, the certificate will be reissued with the new common_name and sans specified. Reissuing to add names may incur additional cost, so removing unused names can free up paid name slots in the certificate. Unlimited reissues are available during the lifetime of the certificate. New names added to a certificate that do not share the base domain of the common_name may take additional time to validate. If this API call is made before a previous pending reissue has been validated and issued, the previous reissue request is automatically rejected and replaced with the current request.

Request Endpoint

Method	URL
REISSUE	`https://api.digicert.com/order/{order_id}`

Request Parameters

Parameters are encoded in one of the content types that is accepted by the endpoint.

Parameter Name	Req/Opt	Allowed Values	Default	Description
csr	Optional	[string]	[existing CSR]	Certificate Signing Request. To create a CSR from your server, more information is available at http://www.digicert.com/csr-creation.htm.
sans	Optional	[string]	[existing value]	Subject Alternative Names. A comma delimited list of names to be secured in the certificate.
server_type	Optional	see Server Types	[existing value]	Server platform. Defaults to server_type of order.
signature_hash	Optional	sha1, sha256	[existing value]	The certificate's signing algorithm hash
comments	Optional	[string]	[blank]	Notes about this order for customer's internal use.

Response

Parameter Name	Data Type	Description
pending_name_changes	[collection]	Details of name changes (if any) are listed in this collection.
added_names	[set]	Any added names are listed here.
added_name	[string]	A name to be added in the reissue, example: example.edu
dropped_names	[set]	Any removed names are listed here.
dropped_name	[string]	A name to be removed in the reissue, example: example.edu
rejected_reissue	[set]	Details of the rejected reissue (if any) are listed here.
common_name	[string]	The Common Name of the rejected reissue request.
sans	[collection]	The list of SANs in the rejected reissue.

Parameter Name	Data Type	Description
san	[string]	Subject Alternative Name contained in the rejected reissue.

Sample Request

Endpoint

```
https://api.digicert.com/order/222307
```

Headers

```
Authorization: Basic MDAxMDA3OnNreWZhbGw=
Content-Length: 239
Content-Type: application/vnd.digicert.rest-v1+json
User-Agent: MyAPIConsumer/0.42
X-HTTP-Method-Override: REISSUE
```

Body

```
{
  "comments": "",
  "common_name": "www.example.com",
  "sans": "a.example.com,b.example.com,c.example.com,d.example.com,e.example.com",
  "csr": "-----BEGIN CERTIFICATE REQUEST-----\n[CSR GOES HERE]\n-----END
CERTIFICATE REQUEST-----"
}
```

Sample Response

Note that with this sample request, it would return `JSON`. Including `XML` for comparison.

Status Code: 201

Headers

```
Content-Length: 331
Content-Type: application/vnd.digicert.rest-v1+json
```

Body

JSON (application/vnd.digicert.rest-v1+json)	XML (application/vnd.digicert.rest-v1+xml)

<pre>{
  "pending_name_changes":{
  "added_names":[
  "c.example.com",
  "d.example.com",
  "f.example.com"
  ],
  "dropped_names":[
  "x.example.com",
  "y.example.com"
  ]
  },
  "rejected_reissue":{
  "common_name":"www.example.com",
  "sans":[
  "a.example.com",
  "q.example.com"
  ]
  }
}</pre>
	<pre><response>
	  <pending_name_changes>
	  <added_names>
	  <added_name>c.example.com</added_name>
	  <added_name>d.example.com</added_name>
	  <added_name>f.example.com</added_name>
	  </added_names>
	  <dropped_names>
	  <dropped_name>x.example.com</dropped_name>
	  <dropped_name>y.example.com</dropped_name>
	  </dropped_names>
	  </pending_name_changes>
	  <rejected_reissue>
	  <common_name>www.example.com</common_name>
	  <sans>
	  <san>a.example.com</san>
	  <san>q.example.com</san>
	  </sans>
	  </rejected_reissue>
	</response></pre>

Retrieve an Issued Certificate

Once a certificate has been issued, it can be retrieved through the API by making a `GET` request to the certificate endpoint. The API will return the certificate, the intermediate certificate, the root certificate, and the certificates in PKCS7. If the certificate has not yet been issued you will get an error code of *cert_unavailable_processing*.

This endpoint will return an `ETAG` header that can be used on subsequent calls. If nothing has changed, a 304 response will be returned.

Request Endpoint

Method	URL
GET	https://api.digicert.com/order/{order_id}/certificate

Response

Parameter Name	Data Type	Description
order_id	[integer]	Unique identifier for certificate order.
serial	[string]	The serial Number of the Certificate
certs	[set]	A data set containing the certificates
certificate	[string]	The certificate
intermediate	[string]	The intermediate certificate
root	[string]	The root certificate
pkcs7	[string]	PKCS7 of the certificates

Sample Request

Endpoint

```
https://api.digicert.com/order/222307/certificate
```

Headers

```
Accept: application/vnd.digicert.rest-v1+json
Authorization: Basic MDAxMDA3OnNreWZhbGw=
User-Agent: MyAPIConsumer/0.42
```

Body

None

Sample Response

Note that with this sample request, it would return `JSON`. Including `XML` for comparison.

Status Code: 200

Headers

```
Content-Length: 429
Content-Type: application/vnd.digicert.rest-v1+json
Etag: "eac8e208119b9d690be0e2c6981f3c5e"
```

Body

JSON (application/vnd.digicert.rest-v1+json)

```
{
  "order_id":"00276176",
  "serial":"07C2EDE40FEEA2AA03C0615F32D3A26D",
  "certs":{
    "certificate":"-----BEGIN CERTIFICATE-----\n[CERT HERE]\n-----END CERTIFICATE-----\n",
    "intermediate":"-----BEGIN CERTIFICATE----\n[CERT HERE]\n-----END CERTIFICATE-----\n",
    "root":"-----BEGIN CERTIFICATE-----\n[CERT HERE]\n-----END CERTIFICATE-----\n",
    "pkcs7":"-----BEGIN PKCS7-----\n[CERT HERE]\n-----END PKCS7-----\n"
  }
}
```

XML (application/vnd.digicert.rest-v1+xml)

```
<response>
  <order_id>00276176</order_id>
  <serial>07C2EDE40FEEA2AA03C0615F32D3A26D</serial>
  <certs>
    <certificate>
      -----BEGIN CERTIFICATE-----
MIIFMDCCBBigAwIBAgIQB8Lt5A/uoqoDwGFfMtOibTANBgkqhkiG9w0BAQUFADBu
MRMwEQYKCZImiZPyLGQBGRYDY29tMR0wGwYKCZImiZPyLGQBGRYNRGlnaUNlcnQt
[...]+KBK5AAZ7S8N4t3xL4Xgd8Z5QTIwlCj4iYETgC+e6ypRx1EnwkwPTM+v/a3qBAh8
gErk3blCmLLK4ToULBp7R4/fZlyl4htejyDULMSLexKdA+AJ
-----END CERTIFICATE-----
    </certificate>
    <intermediate>
       -----BEGIN CERTIFICATE----
MIIE4zCCA8ugAwIBAgIQByZq8O6u1HhZsthDZLeKTTANBgkqhkiG9w0BAQUFADBx
MRMwEQYKCZImiZPyLGQBGRYDY29tMR0wGwYKCZImiZPyLGQBGRYNRGlnaUNlcnQt
[...]MdK3Empl2nll0vkxKKckM/6g1/zotmUC/L+rCw+75nlYYDN4EhN+2hqN0H7ekiMI
BuUMrn/zOOCE/T51TFmQmVKYhw8mnF6HTeoNqSxiHYsQGDQodFA9847E7puXKWC7
v2w7v6F5aA==
-----END CERTIFICATE-----
    </intermediate>
    <root>
      -----BEGIN CERTIFICATE-----
MIIDzzCCAregAwIBAgIQAblXTWj/ZrPV05Rb6HjP2TANBgkqhkiG9w0BAQUFADBx
MRMwEQYKCZImiZPyLGQBGRYDY29tMR0wGwYKCZImiZPyLGQBGRYNRGlnaUNlcnQt
[...]ijD+28PMnpMH6xOUsJHxpviYGF1Ph2aZvuILgnMGMqAwGp/bdJrHSoyta7CMhNuX
Wg3SKbaUKKz/uA3rMEHFAJ9C4w==
-----END CERTIFICATE-----
    </root>
    <pkcs7>
       -----BEGIN PKCS7-----
MIIOGQYJKoZIhvcNAQcCoIIOCjCCDgYCAQExADALBgkqhkiG9w0BBwGggg3uMIIF
MDCCBBigAwIBAgIQB8Lt5A/uoqoDwGFfMtOibTANBgkqhkiG9w0BAQUFADBuMRMw
[...]ik9Be/i+DFEZQMy7ZfHGEtUx7BuYoNN88eWJfHg5QvbyVd7LwS3LYoow/tvDzJ6T
B+sTlLCR8ab4mBhdT4dmmb7iC4JzBjKgMBqf23Sax0qMrWuwjITbl1oN0im2lCis
/7gN6zBBxQCfQuMxAA==
-----END PKCS7-----
    </pkcs7>
  </certs>
</response>
```

View Order Details

Once the certificate order has been created, periodically view the order details to check the order status.

This endpoint will return an `ETAG` header that can be used on subsequent calls. If nothing has changed, a 304 response will be returned.

Request Endpoint

Method	URL
GET	`https://api.digicert.com/order/{order_id}`

Response

Parameter Name	Data Type	Description
certificate_details	[collection]	Container for certificate details
order_id	[integer]	Unique identifier for certificate order.
status	[string]	Status of order. Usually 'pending issuance' or 'issued'.
product_name	[string]	Product name of order.
validity	[string]	Validity Period of order. Usually 1, 2 or 3 Year(s).
org_unit	[string]	The OU field of the certificate.
common_name	[string]	The Common Name field of the certificate.
sans	[collection]	A collection that contains one or more SANs, (if any).
san	[string]	A Subject Alternative Name (SAN) of the certificate.
order_date	[datetime]	The date the original certificate was ordered. Format: d-M-Y. Example: 19-JAN-2012
valid_from	[datetime]	The start date of the certificate's validity (if issued). Format: d-M-Y. Example: 19-JAN-2012
valid_till	[datetime]	The end date of the certificate's validity (if issued). Format: d---Y. Example: 19-JAN-2013
server_type	[string]	The server software type of the order.
server_type_name	[string]	Name of the server software type of the order.
site_seal_token	[string]	Eight-character site seal token.
pending_reissue	[collection]	If there is a pending reissue, this section will be populated.
common_name	[string]	The Common Name in the current pending reissue.
sans	[collection]	A collection that contains one or more SANs, (if any).
san	[string]	A Subject Alternative Name (SAN) of the pending reissue.
receipt	[collection]	Once the certificate has been issued, a receipt is generated
order_total	[float]	The total paid for the order

Sample Request

Endpoint

```
https://api.digicert.com/order/222307
```

Headers

```
Accept: application/vnd.digicert.rest-v1+json
Authorization: Basic MDAxMDA3OnNreWZhbGw=
```

```
User-Agent: MyAPIConsumer/0.42
```

Body

None

Sample Response

Note that with this sample request, it would return `JSON`. Including `XML` for comparison.

Status Code: 200

Headers

```
Content-Length: 582
Content-Type: application/vnd.digicert.rest-v1+json
Etag: "eac8e208119b9d690be0e2c6981f3c5e"
```

Body

JSON (application/vnd.digicert.rest-v1+json)

```
{
  "certificate_details":{
    "order_id":"00222307",
    "status":"issued",
    "product_name":"Unified Communications",
    "validity":"1 Year(s)",
    "org_unit":"",
    "common_name":"www.example.com",
    "sans":[
       "a.example.com"
    ],
    "order_date":"03-FEB-2012",
    "valid_from":"03-FEB-2012",
    "valid_till":"07-FEB-2013",
    "server_type":"2",
    "server_type_name":"Apache"
  },
  "pending_reissue":{
    "common_name":"www.example.com",
    "sans":[
       "a.example.com",
       "b.example.com"
    ]
  },
  "receipt":{
    "order_total":"175.00"
  }
}
```

XML (application/vnd.digicert.rest-v1+xml)

```
<response>
  <certificate_details>
    <order_id>00222307</order_id>
    <product_name>Unified Communications</product_name>
    <validity>1 Year(s)</validity>
    <org_unit></org_unit>
      <common_name>www.example.com</common_name>
      <sans>
        <san>a.example.com</san>
      </sans>
      <status>issued</status>
      <order_date>03-FEB-2012</order_date>
      <valid_from>03-FEB-2012</valid_from>
      <valid_till>07-FEB-2013</valid_till>
      <server_type>2</server_type>
      <server_type_name>Apache</server_type_name>
    </certificate_details>
    <pending_reissue>
      <common_name>www.example.com</common_name>
      <sans>
        <san>a.example.com</san>
        <san>b.example.com</san>
      </sans>
    </pending_reissue>
    <receipt>
      <order_total>175.00</order_total>
    </receipt>
</response>
```

Sample Code

Reissue Certificate

For the examples below:

Account ID	001007
API Key	n9hq0fp9q63htpmt7xcthztt5n4zx721
Order ID	111222

Using cURL

User creates a JSON string (named *data.json* for simplicity and formatted for readability):

```
{
  "server_type":2,
  "csr":"-----BEGIN CERTIFICATE REQUEST-----\n[CSR GOES HERE]\n-----END CERTIFICATE
REQUEST-----"
}
```

User uses curl from the commandline to send the REISSUE request:

```
curl -u "001007:n9hq0fp9q63htpmt7xcthztt5n4zx721" -X REISSUE \
-H "Content-Type: application/vnd.digicert.rest-v1+json" \
--data @data.json https://api.digicert.com/order/111222
```

Using C#

```
using System;
using System.Text;
using System.Net;
using System.IO;

namespace ConsoleApplication1
{
  class Program
  {
    static void Main(string[] args)
    {
      string post_data = @"{
""server_type"":2,
""csr"":""-----BEGIN CERTIFICATE REQUEST-----\n[CSR GOES HERE]\n-----END
CERTIFICATE REQUEST-----""
      }";

      byte[] postBytes = Encoding.UTF8.GetBytes(post_data);
      string uri = "https://api.digicert.com/order/111222";

      HttpWebRequest request = (HttpWebRequest)WebRequest.Create(uri);
      request.KeepAlive = false;
      request.ProtocolVersion = HttpVersion.Version10;
      request.Method = "REISSUE";
      request.UserAgent = "DigiCertAPIClient/0.1";
      request.ContentType = "application/vnd.digicert.rest-v1+json";
      request.ContentLength = postBytes.Length;
```

```
      request.Headers.Add("Authorization", "Basic " + GetEncodedCredential());

      try
      {
        Stream requestStream = request.GetRequestStream();
        requestStream.Write(postBytes, 0, postBytes.Length);
        requestStream.Close();

        HttpWebResponse response = (HttpWebResponse)request.GetResponse();//throws
WebException
        Console.WriteLine(new StreamReader(response.GetResponseStream()).ReadToEnd
());
      }
      catch (WebException e)
      {
        Console.WriteLine(e.Message);
      }
    }

    static private String GetEncodedCredential()
    {
      return Convert.ToBase64String(ASCIIEncoding.ASCII.GetBytes("001007" + ":" +
"n9hq0fp9q63htpmt7xcthztt5n4zx721"));
    }
  }
}
```

Using PHP

```
$arr = array(
"server_type" => 2,
"csr" => "-----BEGIN CERTIFICATE REQUEST-----\n[CSR GOES HERE]\n-----END
CERTIFICATE REQUEST-----");

$json = json_encode($arr);
$headers = array(
'Content-Type: application/vnd.digicert.rest-v1+json'
);

$url = 'https://api.digicert.com/order/111222';

$ch = curl_init();
curl_setopt($ch, CURLOPT_URL, $url);
curl_setopt($ch, CURLOPT_USERPWD, '001007:n9hq0fp9q63htpmt7xcthztt5n4zx721');
curl_setopt($ch, CURLOPT_RETURNTRANSFER, 1);

curl_setopt($ch, CURLOPT_CUSTOMREQUEST, 'REISSUE');
curl_setopt($ch, CURLOPT_POSTFIELDS, $json);
curl_setopt($ch, CURLOPT_HTTPHEADER, $headers);

$result = curl_exec($ch);
```

Using Ruby

```
require 'net/http'
```

```
require 'uri'

uri = URI.parse 'https://api.digicert.com/order/111222'

request = Net::HTTP::POST.new uri.request_uri
request.basic_auth('001007','n9hq0fp9q63htpmt7xcthztt5n4zx721')

request['Content-Type'] = 'application/vnd.digicert.rest-v1+json'
request['X-HTTP-Method-Override'] = 'REISSUE'

#Note that this JSON body is formatted for readability
request.body = <<JSON
{
"server_type":2,
"csr":"-----BEGIN CERTIFICATE REQUEST-----\n[CSR GOES HERE]\n-----END CERTIFICATE
REQUEST-----"
}
JSON

http = Net::HTTP.new(uri.host, uri.port)
http.use_ssl = true

result = http.request(request)
```

Retrieve Certificate

For the examples below:

Account ID	001007
API Key	n9hq0fp9q63htpmt7xcthztt5n4zx721
Order ID	111222

`GET` requests are very simple. After authenticating with the service, having the client library access the URL is all that is required.

Using cURL

```
curl -u "001007:n9hq0fp9q63htpmt7xcthztt5n4zx721" \
-H "Accept: application/vnd.digicert.rest-v1+json" \
https://api.digicert.com/order/111222/certificate
```

Using C#

```
using System;
using System.Text;
using System.Net;
using System.IO;

namespace ConsoleApplication1
{
  class Program
  {
    static void Main(string[] args)
    {
      string uri = "https://api.digicert.com/order/111222/certificate";

      HttpWebRequest request = (HttpWebRequest)WebRequest.Create(uri);
      request.KeepAlive = false;
      request.ProtocolVersion = HttpVersion.Version10;
      request.Method = "GET";
      request.UserAgent = "DigiCertAPIClient/0.1";
      request.ContentType = "application/vnd.digicert.rest-v1+json";
      request.Headers.Add("Authorization", "Basic " + GetEncodedCredential());

      try
      {
        HttpWebResponse response = (HttpWebResponse)request.GetResponse();//throws
WebException
        Console.WriteLine(new StreamReader(response.GetResponseStream()).ReadToEnd
());
      }
      catch (WebException e)
      {
        Console.WriteLine(e.Message);
      }
    }

    static private String GetEncodedCredential()
```

```
    {
      return Convert.ToBase64String(ASCIIEncoding.ASCII.GetBytes("001007" + ":" +
"n9hq0fp9q63htpmt7xcthztt5n4zx721"));
    }
  }
}
```

Using PHP

```
$url = 'https://api.digicert.com/order/111222/certificate';

$headers = array(
  'Accept: application/vnd.digicert.rest-v1+json',
);

$ch = curl_init();
curl_setopt($ch, CURLOPT_URL, $url);
curl_setopt($ch, CURLOPT_USERPWD, '001007:n9hq0fp9q63htpmt7xcthztt5n4zx721');
curl_setopt($ch, CURLOPT_RETURNTRANSFER, 1);

curl_setopt($ch, CURLOPT_HTTPHEADER, $headers);

$result = curl_exec($ch);
```

Using Ruby

```
require 'net/http'
require 'uri'

uri = URI.parse 'https://api.digicert.com/order/111222/certificate'

request = Net::HTTP::Get.new uri.request_uri
request.basic_auth('001007','n9hq0fp9q63htpmt7xcthztt5n4zx721')

request['Accept'] = 'application/vnd.digicert.rest-v1+json'

http = Net::HTTP.new(uri.host, uri.port)
http.use_ssl = true

result = http.request request
```

Errors and Troubleshooting

As mentioned in the introduction, error responses will return an HTTP code in 400+ range. Clients receiving these headers know that their request failed for various reasons. The API should always at least return a 500 error if there was a fatal error that wasn't caught. Most error responses return a special content type, based on the initial type being requested:
`application/vnd.digicert.error+xml` for XML based requests, and
`application/vnd.digicert.error+json` for JSON based. This content type is based on emerging standards.

In the case of 406 errors, the API doesn't know what format the client requested, so it will send `application/vnd.digicert.error+json.`

Error Return Values

Parameter Name	Data Type	Description
errors	[collection]	A collection of error codes
error	[set]	A data set containing information about a specific error
code	[string]	An error code that identifies the error that occurred
description	[string]	A human readable description of the error

Sample XML Error Response

For instance, a client wants to reject a pending request, but forgot to include the required NOTE. They would make a `REJECT` request to the `/request/7902` endpoint, and would receive a 400 response with the following body:

JSON (application/vnd.digicert.error+json)

```
{
  "errors":[
    {
      "code":"required_param:note",
      "description":"Note is required"
    }
  ]
}
```

XML (application/vnd.digicert.error+xml)

```
<errors>
  <error>
    <code>required_param:note</code>
    <description>Note is required</description>
  </error>
</errors>
```

Appendix

Country Codes

USA	us
Canada	ca
United Kingdom	gb
Afghanistan	af
Åland Islands	ax
Albania	al
Algeria	dz
American Samoa	as
Andorra	ad
Angola	ao
Anguilla	ai
Antarctica	aq
Antigua and Barbuda	ag
Argentina	ar
Armenia	am
Aruba	aw
Australia	au
Austria	at
Azerbaijan	az
Bahamas	bs
Bahrain	bh
Bangladesh	bd
Barbados	bb
Belarus	by
Belgium	be
Belize	bz
Benin	bj
Bermuda	bm
Bhutan	bt
Bolivia	bo
Bonaire, Sint Eustatius and Saba	bq
Bosnia and Herzegovina	ba
Botswana	bw
Bouvet Island	bv
Brazil	br

British Indian Ocean Territory	io
Brunei Darussalam	bn
Bulgaria	bg
Burkina Faso	bf
Burundi	bi
Cambodia	kh
Cameroon	cm
Cape Verde	cv
Cayman Islands	ky
Central African Republic	cf
Chad	td
Chile	cl
China	cn
Christmas Island	cx
Cocos (Keeling) Islands	cc
Colombia	co
Comoros	km
Congo	cg
Congo, the Democratic Republic of the	cd
Cook Islands	ck
Costa Rica	cr
Côte d'Ivoire	ci
Croatia	hr
Curaçao	cw
Cyprus	cy
Czech Republic	cz
Denmark	dk
Djibouti	dj
Dominica	dm
Dominican Republic	do
Ecuador	ec
Egypt	eg
El Salvador	sv
Equatorial Guinea	gq
Eritrea	er

Estonia	ee
Ethiopia	et
Falkland Islands (Malvinas)	fk
Faroe Islands	fo
Fiji	fj
Finland	fi
France	fr
French Guiana	gf
French Polynesia	pf
French Southern Territories	tf
Gabon	ga
Gambia	gm
Georgia	ge
Germany	de
Ghana	gh
Gibraltar	gi
Greece	gr
Greenland	gl
Grenada	gd
Guadeloupe	gp
Guam	gu
Guatemala	gt
Guernsey	gg
Guinea	gn
Guinea-Bissau	gw
Guyana	gy
Haiti	ht
Heard and McDonald Islands	hm
Honduras	hn
Hong Kong	hk
Hungary	hu
Iceland	is
India	in
Indonesia	id
Iraq	iq

Ireland	ie
Isle of Man	im
Israel	il
Italy	it
Jamaica	jm
Japan	jp
Jersey	je
Jordan	jo
Kazakhstan	kz
Kenya	ke
Kiribati	ki
Korea, South	kr
Kuwait	kw
Kyrgyzstan	kg
Laos	la
Latvia	lv
Lebanon	lb
Lesotho	ls
Liberia	lr
Libya	ly
Liechtenstein	li
Lithuania	lt
Luxembourg	lu
Macao	mo
Macedonia, Republic of	mk
Madagascar	mg
Malawi	mw
Malaysia	my
Maldives	mv
Mali	ml
Malta	mt
Marshall Islands	mh
Martinique	mq
Mauritania	mr
Mauritius	mu
Mayotte	yt
Mexico	mx
Micronesia, Federated States of	fm
Moldova	md
Monaco	mc
Mongolia	mn
Montenegro	me
Montserrat	ms
Morocco	ma
Mozambique	mz
Myanmar	mm
Namibia	na
Nauru	nr
Nepal	np
Netherlands	nl
Netherlands Antilles	an
New Caledonia	nc
New Zealand	nz
Nicaragua	ni
Niger	ne
Nigeria	ng
Niue	nu
Norfolk Island	nf
Northern Mariana Islands	mp
Norway	no
Oman	om
Pakistan	pk
Palau	pw
Palestinian Territory	ps
Panama	pa
Papua New Guinea	pg
Paraguay	py
Peru	pe
Philippines	ph
Pitcairn Island	pn
Poland	pl
Portugal	pt
Puerto Rico	pr
Qatar	qa
Réunion	re
Romania	ro
Russia	ru
Rwanda	rw
S. Georgia and S. Sandwich Isls.	gs
Saint Barthélemy	bl
Saint Kitts & Nevis	kn
Saint Martin	mf
Saint Lucia	lc
Saint Vincent and The Grenadines	vc
Samoa	ws
San Marino	sm
Sao Tome and Principe	st
Saudi Arabia	sa
Senegal	sn
Serbia	rs
Seychelles	sc
Sierra Leone	sl
Singapore	sg
Sint Maarten	sx
Slovakia	sk
Slovenia	si
Solomon Islands	sb
Somalia	so
South Africa	za
South Sudan	ss
Spain	es
Sri Lanka	lk
St. Helena	sh
St. Pierre and Miquelon	pm
Suriname	sr
Svalbard and Jan Mayen Islands	sj
Swaziland	sz
Sweden	se
Switzerland	ch
Taiwan	tw
Tajikistan	tj
Tanzania	tz
Thailand	th
Timor-Leste	tl
Togo	tg
Tokelau	tk
Tonga	to
Trinidad and Tobago	tt
Tunisia	tn
Turkey	tr

Turkmenistan	tm
Turks and Caicos Islands	tc
Tuvalu	tv
U.S. Minor Outlying Islands	um
Uganda	ug
Ukraine	ua

United Arab Emirates	ae
Uruguay	uy
Uzbekistan	uz
Vanuatu	vu
Vatican City	va
Venezuela	ve
Vietnam	vn

Virgin Islands, British	vg
Virgin Islands, U.S.	vi
Wallis and Futuna Islands	wf
Western Sahara	eh
Yemen	ye
Zambia	zm
Zimbabwe	zw

Server Types

Apache	2
BEA Weblogic 8 & 9	42
Barracuda	41
Bea Weblogic 7 and older	29
Cisco	30
Citrix (Other)	39
Citrix Access Essentials	46
Citrix Access Gateway 4.x	50
Citrix Access Gateway 5.x and higher	58
F5 Big-IP	31
F5 FirePass	32
IBM HTTP Server	7
Java Web Server (Javasoft / Sun)	10
Juniper	33
Lighttpd	44
Lotus Domino	11
Mac OS X Server	49
Microsoft Exchange Server 2003	47
Microsoft Exchange Server 2007	36
Microsoft Exchange Server 2010	48
Microsoft Exchange Server 2013	68
Microsoft Forefront Unified Access Gateway	66
Microsoft IIS 1.x to 4.x	13
Microsoft IIS 5 or 6	14
Microsoft IIS 7	40
Microsoft IIS 8	67
Microsoft Live Communications Server 2005	37
Microsoft Lync Server 2010	59
Microsoft Lync Server 2013	69
Microsoft OCS R2	60
Microsoft Office Communications Server 2007	38
Microsoft Small Business Server 2008 & 2011	62
Netscape Enterprise Server	15
Netscape iPlanet	9
Novell NetWare	17
Novell iChain	65
OTHER	-1
Oracle	18
Qmail	34
SunOne	35
Tomcat	24
WebStar	26
Zeus Web Server	28
cPanel	43
nginx	45

Code Signing Certificate Platforms

Adobe AIR	52
Apple OS X	53
Microsoft Authenticode	51
Microsoft Kernel-Mode Code	64
Microsoft Office VBA	54
Mozilla	56
Other	57
Sun Java	55

EV Code Signing Certificate Platforms

EV Code Signing Certificates must be installed on to a hardware token device. DigiCert offers two provisioning options:

1. We can ship you a hardware token with your EV Code Signing Certificate already provisioned for you (`cs_provisioning_method = 'ship_token'`)
2. If you have a qualified hardware token (see chart below), you can provision your own certificate using the tool found at http://www.digicert.com/util/DigiCertHardwareCertificateInstaller.zip (`cs_provisioning_method = 'client_app'`)

AEP Keyper	15
ARX PrivateServer	16
Bull Trustway Crypto PCI	17
Other FIPS 140-2 Level 2 Device	-1
SafeNet eToken 5100	6
SafeNet eToken 5105	7
SafeNet eToken 5200	8
SafeNet eToken 5205	9
SafeNet eToken PRO 72K	3
SafeNet eToken PRO Anywhere	2
SafeNet iKey 4000	10
Safenet Luna	12
Thales nShield	13
Utimaco CryptoServer	14

ISO 639-2 Language Codes

Abkhazian	abk
Achinese	ace
Acoli	ach
Adangme	ada
Adyghe	ady
Afar	aar
Afrikaans	afr
Ainu	ain
Akan	aka
Albanian	sqi
Aleut	ale
Amharic	amh
Angika	anp
Arabic	ara
Aragonese	arg
Arapaho	arp
Arawak	arw
Armenian	hye
Assamese	asm
Asturian	ast
Avaric	ava
Awadhi	awa
Aymara	aym
Azerbaijani	aze
Balinese	ban
Baluchi	bal
Bambara	bam
Basa	bas
Bashkir	bak
Basque	eus
Beja	bej
Belarusian	bel
Bemba	bem
Bengali	ben
Bhojpuri	bho
Bikol	bik
Bilin	byn
Bini	bin
Bislama	bis
Bosnian	bos
Braj	bra
Breton	bre
Buginese	bug
Bulgarian	bul
Buriat	bua
Burmese	mya
Caddo	cad
Catalan	cat
Cebuano	ceb
Central Khmer	khm
Chamorro	cha
Chechen	che
Cherokee	chr
Cheyenne	chy
Chinese	zho
Chinook	chn
Chipewyan	chp
Choctaw	cho
Chuukese	chk
Chuvash	chv
Cornish	cor
Corsican	cos
Cree	cre
Creek	mus
Crimean Tatar	crh
Croatian	hrv
Czech	ces
Dakota	dak
Danish	dan
Dargwa	dar
Delaware	del
Dhivehi	div
Dinka	din
Dogri	doi
Dogrib	dgr
Duala	dua
Dutch	nld
Dyula	dyu
Dzongkha	dzo
Eastern Frisian	frs
Efik	efi
Ekajuk	eka
English	eng
Erzya	myv
Estonian	est
Ewe	ewe
Ewondo	ewo
Fang	fan
Fanti	fat
Faroese	fao
Fijian	fij
Filipino	fil
Finnish	fin
Fon	fon
French	fra
Friulian	fur
Fulah	ful
Ga	gaa
Galibi Carib	car
Galician	glg
Ganda	lug
Gayo	gay
Gbaya	gba
Georgian	kat
German	deu
Gilbertese	gil
Gondi	gon
Gorontalo	gor
Grebo	grb
Guarani	grn
Gujarati	guj
Gwich’in	gwi
Haida	hai
Haitian	hat
Hausa	hau
Hawaiian	haw
Hebrew	heb

Herero	her
Hiligaynon	hil
Hindi	hin
Hiri Motu	hmo
Hmong	hmn
Hungarian	hun
Hupa	hup
Iban	iba
Icelandic	isl
Igbo	ibo
Iloko	ilo
Inari Sami	smn
Indonesian	ind
Ingush	inh
Inuktitut	iku
Inupiaq	ipk
Irish	gle
Italian	ita
Japanese	jpn
Javanese	jav
Judeo-Arabic	jrb
Judeo-Persian	jpr
Kabardian	kbd
Kabyle	kab
Kachin	kac
Kalaallisut	kal
Kalmyk	xal
Kamba (Kenya)	kam
Kannada	kan
Kanuri	kau
Kara-Kalpak	kaa
Karachay-Balkar	krc
Karelian	krl
Kashmiri	kas
Kashubian	csb
Kazakh	kaz
Khasi	kha
Kikuyu	kik
Kimbundu	kmb
Kinyarwanda	kin
Kirghiz	kir
Komi	kom
Kongo	kon
Konkani	kok
Korean	kor
Kosraean	kos
Kpelle	kpe
Kuanyama	kua
Kumyk	kum
Kurdish	kur
Kurukh	kru
Kutenai	kut
Ladino	lad
Lahnda	lah
Lamba	lam
Lao	lao
Latvian	lav
Lezghian	lez
Limburgan	lim
Lingala	lin
Lithuanian	lit
Low German	nds
Lower Sorbian	dsb
Lozi	loz
Luba-Katanga	lub
Luba-Lulua	lua
Luiseno	lui
Lule Sami	smj
Lunda	lun
Luo	luo
Lushai	lus
Luxembourgish	ltz
Macedo-Romanian	rup
Macedonian	mkd
Madurese	mad
Magahi	mag
Maithili	mai
Makasar	mak
Malagasy	mlg
Malay	msa
Malayalam	mal
Maltese	mlt
Manchu	mnc
Mandar	mdr
Mandingo	man
Manipuri	mni
Manx	glv
Maori	mri
Mapudungun	arn
Marathi	mar
Mari	chm
Marshallese	mah
Marwari	mwr
Masai	mas
Mende	men
Mi'kmaq	mic
Minangkabau	min
Mirandese	mwl
Greek	ell
Mohawk	moh
Moksha	mdf
Mongo	lol
Mongolian	mon
Mossi	mos
N'Ko	nqo
Nauru	nau
Navajo	nav
Ndonga	ndo
Neapolitan	nap
Nepali	nep
Newari	new
Nias	nia
Niuean	niu
Nogai	nog
North Ndebele	nde
Northern Frisian	frr
Northern Sami	sme
Norwegian	nor
Norwegian Bokmål	nob
Norwegian Nynorsk	nno
Nyamwezi	nym
Nyanja	nya
Nyankole	nyn

Nyoro	nyo
Nzima	nzi
Occitan	oci
Ojibwa	oji
Oriya	ori
Oromo	orm
Osage	osa
Ossetian	oss
Palauan	pau
Pampanga	pam
Pangasinan	pag
Panjabi	pan
Papiamento	pap
Pedi	nso
Persian	fas
Pohnpeian	pon
Polish	pol
Portuguese	por
Pushto	pus
Quechua	que
Rajasthani	raj
Rapanui	rap
Rarotongan	rar
Romanian	ron
Romansh	roh
Romany	rom
Rundi	run
Russian	rus
Samoan	smo
Sandawe	sad
Sango	sag
Santali	sat
Sardinian	srd
Sasak	sas
Scots	sco
Scottish Gaelic	gla
Selkup	sel
Serbian	srp
Serer	srr
Shan	shn
Shona	sna
Sichuan Yi	iii
Sicilian	scn
Sidamo	sid
Siksika	bla
Sindhi	snd
Sinhala	sin
Skolt Sami	sms
Slave (Athapascan)	den
Slovak	slk
Slovenian	slv
Somali	som
Soninke	snk
South Ndebele	nbl
Southern Altai	alt
Southern Sami	sma
Southern Sotho	sot
Spanish	spa
Sranan Tongo	srn
Sukuma	suk
Sundanese	sun
Susu	sus
Swahili	swa
Swati	ssw
Swedish	swe
Swiss German	gsw
Syriac	syr
Tagalog	tgl
Tahitian	tah
Tajik	tgk
Tamashek	tmh
Tamil	tam
Tatar	tat
Telugu	tel
Tereno	ter
Tetum	tet
Thai	tha
Tibetan	bod
Tigre	tig
Tigrinya	tir
Timne	tem
Tiv	tiv
Tlingit	tli
Tok Pisin	tpi
Tokelau	tkl
Tonga (Nyasa)	tog
Tonga (Tonga Islands)	ton
Tsimshian	tsi
Tsonga	tso
Tswana	tsn
Tumbuka	tum
Turkish	tur
Turkmen	tuk
Tuvalu	tvl
Tuvinian	tyv
Twi	twi
Udmurt	udm
Uighur	uig
Ukrainian	ukr
Umbundu	umb
Upper Sorbian	hsb
Urdu	urd
Uzbek	uzb
Vai	vai
Venda	ven
Vietnamese	vie
Votic	vot
Walloon	wln
Waray	war
Washo	was
Welsh	cym
Western Frisian	fry
Wolaytta	wal
Wolof	wol
Xhosa	xho
Yakut	sah
Yao	yao
Yapese	yap
Yiddish	yid
Yoruba	yor
Zapotec	zap
Zaza	zza
Zenaga	zen

Zhuang	zha	Zulu	zul	Default	zxx
		Zuni	zun		