Venafi Zero Touch PKI (ZTPKI) API 1.12.0 OAS 3.0

Venafi provides customers the ability to manage the lifecycle of PKI certificates via the web based ZTPKI REST API service. This document outlines the methods to control the generation, retrieval and revocation of certificates.

ZTPKI groups methods into 3 categories when dealing with the certificate lifecycle:

- Policies
- Certificate Requests
- Certificates

Policies

Describes the required, optional and default value for data elements that make up a certificate request. Includes:

- Subject DN components - CN, OU, O, C, etc.
- SAN elements - DNSName, IPAddress, RFC822Name, UPN, URI, etc.
- Custom Extensions - X509v3 extensions included in the certificate, i.e. Microsoft Application Policies, Template Information, etc.
- Custom Fields - descriptive non-certificate attributes used by the customer for reporting/identification, i.e. Department, Charge code, etc.

ACTION	URI	DESCRIPTION
GET	/api/v2/policies	Retrieves all policies available to user
GET	/api/v2/policies/{policyId}	Retrieves a given policy

Certificate Requests

Describes all requests to generate certificates. After issued, links to a certificateId for operations on the actual certificate.

ACTION	URI	DESCRIPTION
POST	/api/v2/csr	Creates a new request, returning certRequestId
GET	/api/v2/csr/{certRequestId}	Retrieves a given certRequest, including all details
GET	/api/v2/csr/{certRequestId}/status	Retrieves issuance status of certRequest, including certificateId if issued

Certificates

Describes all certificates issued by the user or others the user has access to.

ACTION	URI	DESCRIPTION
POST	/api/v2/certificates	Retrieves all certificates available to user
GET	/api/v2/certificates/{certificateId}	Retrieves a given certificate, including status & PEM format cert
GET	/api/v2/certificates/{certificateId}/der	Retrieves a binary DER format certificate
GET	/api/v2/certificates/{certificateId}/pem	Retrieves a PEM format certificate
PATCH	/api/v2/certificates/{certificateId}	Revokes a given certificate, given a certificateId

ACTION	URI	DESCRIPTION
PATCH	/api/v2/certificates/{serialNumber}	Revokes a given certificate, given a serialNumber and issuerDN
GET	/api/v2/certificates/{certificateId}/status	Retrieves current revocation status

Authentication

All REST methods are require authentication via the "HTTP Holder-Of-Key Authentication Scheme", known as "Hawk Authentication" (see https://github.com/mozilla/hawk/blob/main/API.md). Hawk is an HTTP authentication scheme providing mechanisms for making authenticated HTTP requests with partial cryptographic verification of the request and response, covering the HTTP method, request URI, host, and optionally the request payload.

With Hawk Auth, each user will be issued a two randomly generated strings by Venafi - an ID that will appear in the request, as well as a key that will be used to sign the request. Each REST method requires an `Authorization` header formatted as below:

```
Authorization: Hawk id="<hawk id>", ts="<timestamp in seconds>", nonce="<random string>", mac="
<Base64 HMAC>", hash="<SHA256 Hash>"
```

Also, please note that any REST actions that require a payload (POST/PATCH, etc.) will require the "hash" element for payload validation - see Payload Validation for more information. Responses may contain a `Server-Authorization` header with a hash of the response payload - see Response Payload Validation.

See the Hawk Authentication page on GitHub for links to implementations in NodeJS, .Net, Ruby, Java, etc.

ZTPKI provides REST API methods to manage Hawk credentials for a user. Each user account is allowed up to 5 sets of Hawk credentials to access the REST API. It is recommended that users "roll" their credentials from the initial values set by Venafi and on as as needed basis for security. Optional comments can be added to credentials for descriptive purposes. The following actions are available:

ACTION	URI	DESCRIPTION
POST	/api/v2/hawk	Creates a new request, returning certRequestId
GET	/api/v2/hawk	Retrieves all credentials associated with the current user
GET	/api/v2/hawk/{hawkId}	Retrieves a specific Hawk credential
PUT	/api/v2/hawk/{hawkId}	Rolls a credential by establishing a new secret key
DELETE	/api/v2/hawk/{hawkId}	Disables and removes a credential

Certificate Request Pseudocode

Requesting a certificate will normally flow like this:

```
Generate CSR

Submit Request via POST /api/v2/csr, returning certRequestId

While CertRequest issuanceStatus == APPROVAL_REQUIRED | IN_PROCESS | PENDING
    Get CertRequest Status via GET /api/v2/csr/{certRequestId}/status

If issuanceStatus == ISSUED
    Download PEM certificate via GET /api/v2/certificates/{certificateId}
```

Please contact Venafi for sample code in various languages.

Servers

https://ztpki-staging.venafi.com/api/v2 - Staging server, used for customer integration testing

Authorize

Certificates

POST `/certificates/` Get Certificates

Parameters

No parameters

Request body — application/json

Filter certificates

Example Value | Schema

```
{
  "common_name": "string",
  "serial": "string",
  "not_before": "string",
  "not_after": "string",
  "expired": true,
  "renewed": true,
  "created_since": "string",
  "updated_since": "string",
  "status": "VALID",
  "owner": "string",
  "account": "string",
  "organization": "string",
  "policy": "string",
  "fingerprint": "string",
  "limit": 0,
  "offset": 0,
  "sort_type": "string",
  "sort_direction": "asc",
  "pem": true,
  "span": 0,
  "imported": true,
  "discovered": true,
  "selfSigned": true
}
```

Responses

Code	Description	Links
200	A list of Certificates Media type application/json Controls `Accept` header.	*No links*

Code Description Links

Example Value | Schema

```
{
  "count": 0,
  "items": [
    {
      "id": "string",
      "commonName": "string",
      "serial": "string",
      "notBefore": "string",
      "notAfter": "string",
      "revocationStatus": "VALID",
      "SANs": [
        "string"
      ],
      "policy": {
        "name": "string"
      },
      "imported": true,
      "discovered": true,
      "selfSigned": true
    }
  ]
}
```

422 InvalidRequestError *No links*

Media type

application/json ⌄

Example Value | Schema

```
{
  "message": "string"
}
```

500 WrappedError *No links*

Media type

application/json ⌄

Example Value | Schema

```
{
  "message": "string"
}
```

PATCH `/certificates/{id}`

Revoke a certificate given a certificate id or serial_number and issuerDN

Parameters

Name	Description
id * required `string` *(path)*	Certificate id or serial number id

Request body required application/json ⌄

Revocation reason and optional issuer DN

Example Value | Schema

```
{
  "reason": 0,
  "revocationDate": "string",
  "issuerDN": "string"
}
```

Responses

Code	Description	Links
200	A Certificate revocation request status	*No links*

Media type

application/json ⌄

Controls `Accept` header.

Example Value | Schema

```
{
  "id": "string",
  "revocationStatus": "VALID",
  "revocationReason": 0,
  "revocationDate": "string"
}
```

Code	Description	Links
403	NotAllowedError	*No links*

Media type

application/json ⌄

Example Value | Schema

```
{
  "message": "string"
}
```

Code	Description	Links
404	NotFoundError	*No links*

Media type

application/json ⌄

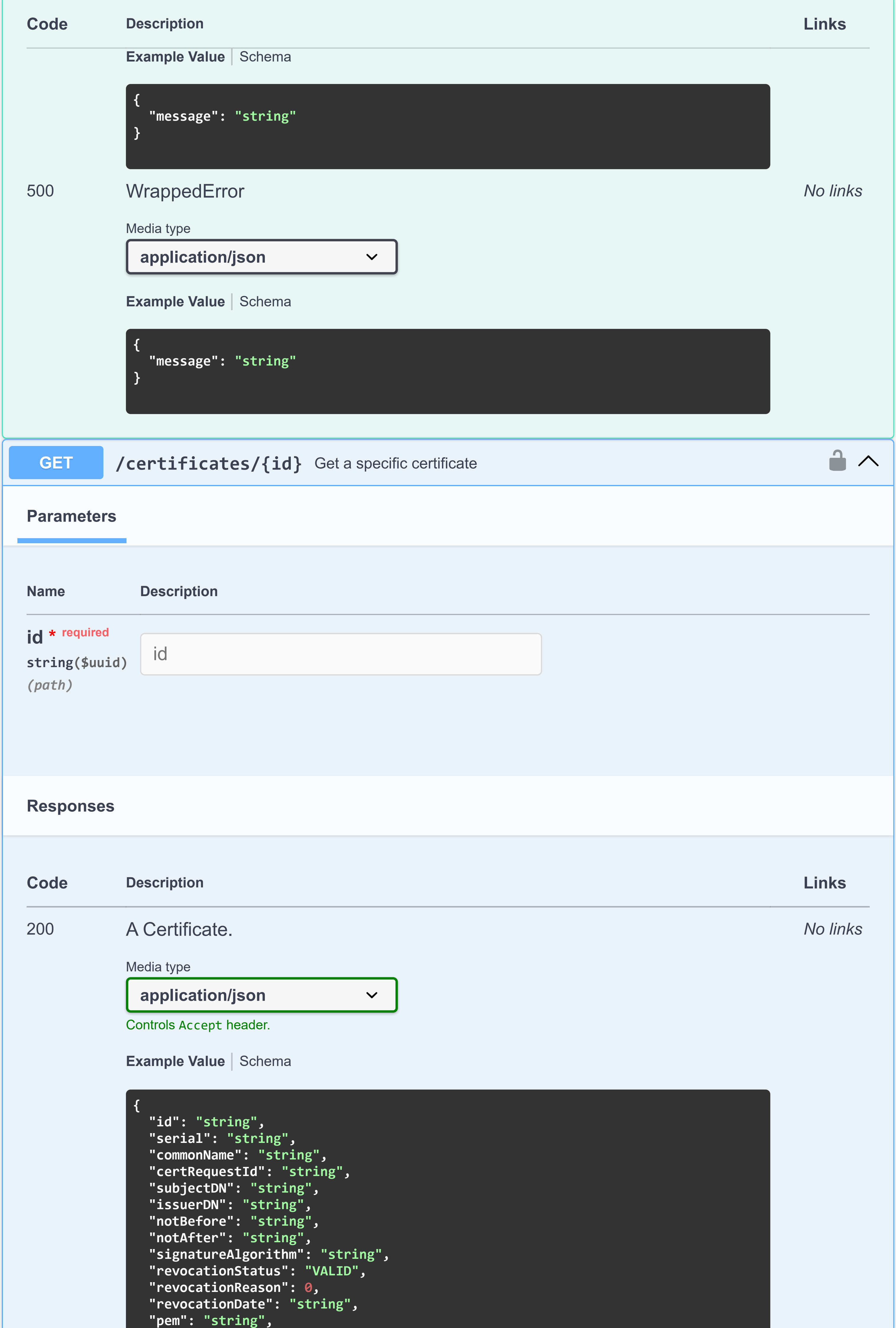
Example Value | Schema

```
{
  "message": "string"
}
```

Code	Description	Links
422	InvalidRequestError	*No links*

Media type

application/json ⌄



Code	Description	Links
	Example Value | Schema	

```
{
  "message": "string"
}
```

Code	Description	Links
500	WrappedError	*No links*

Media type

application/json

Example Value | Schema

```
{
  "message": "string"
}
```

GET `/certificates/{id}` Get a specific certificate

Parameters

Name	Description
id * required `string($uuid)` *(path)*	id

Responses

Code	Description	Links
200	A Certificate.	*No links*

Media type

application/json

Controls `Accept` header.

Example Value | Schema

```
{
  "id": "string",
  "serial": "string",
  "commonName": "string",
  "certRequestId": "string",
  "subjectDN": "string",
  "issuerDN": "string",
  "notBefore": "string",
  "notAfter": "string",
  "signatureAlgorithm": "string",
  "revocationStatus": "VALID",
  "revocationReason": 0,
  "revocationDate": "string",
  "pem": "string",
```

Code | Description | Links

```
  "imported": true,
  "discovered": true,
  "selfSigned": true,
  "createdAt": "string",
  "SANs": [
    "string"
  ],
  "policy": {
    "id": "string",
    "name": "string"
  },
  "user": {
    "id": "string",
    "firstName": "string",
```

404 NotFoundError — *No links*

Media type

application/json

Example Value | Schema

```
{
  "message": "string"
}
```

500 WrappedError — *No links*

Media type

application/json

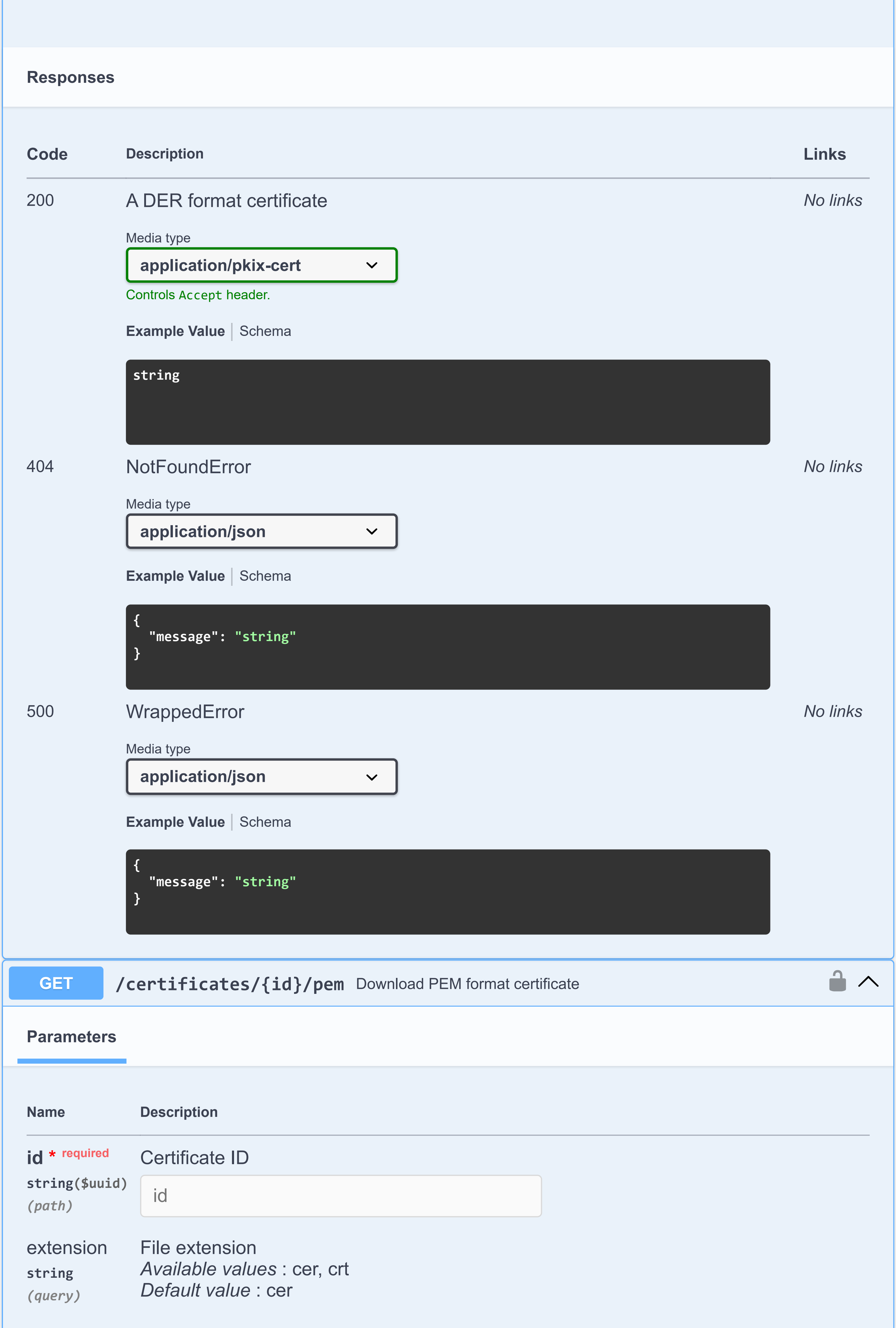
Example Value | Schema

```
{
  "message": "string"
}
```

GET `/certificates/{id}/der` Download DER format certificate

Parameters

Name	Description
id * required `string($uuid)` *(path)*	Certificate ID id
extension `string` *(query)*	File extension *Available values* : cer, crt *Default value* : cer cer
chain `boolean` *(query)*	Include Chain --



Responses

Code	Description	Links
200	A DER format certificate	*No links*
404	NotFoundError	*No links*
500	WrappedError	*No links*

200 A DER format certificate

Media type

application/pkix-cert

Controls `Accept` header.

Example Value | Schema

```
string
```

404 NotFoundError

Media type

application/json

Example Value | Schema

```
{
  "message": "string"
}
```

500 WrappedError

Media type

application/json

Example Value | Schema

```
{
  "message": "string"
}
```

GET `/certificates/{id}/pem` Download PEM format certificate

Parameters

Name	Description
id * required `string($uuid)` *(path)*	Certificate ID id
extension `string` *(query)*	File extension *Available values* : cer, crt *Default value* : cer

Name	Description
	cer
chain boolean *(query)*	Include Chain --

Responses

Code	Description	Links
200	A PEM format certificate	*No links*

Media type

application/x-pem-file

Controls `Accept` header.

Example Value | Schema

```
string
```

Code	Description	Links
404	NotFoundError	*No links*

Media type

application/json

Example Value | Schema

```
{
  "message": "string"
}
```

Code	Description	Links
500	WrappedError	*No links*

Media type

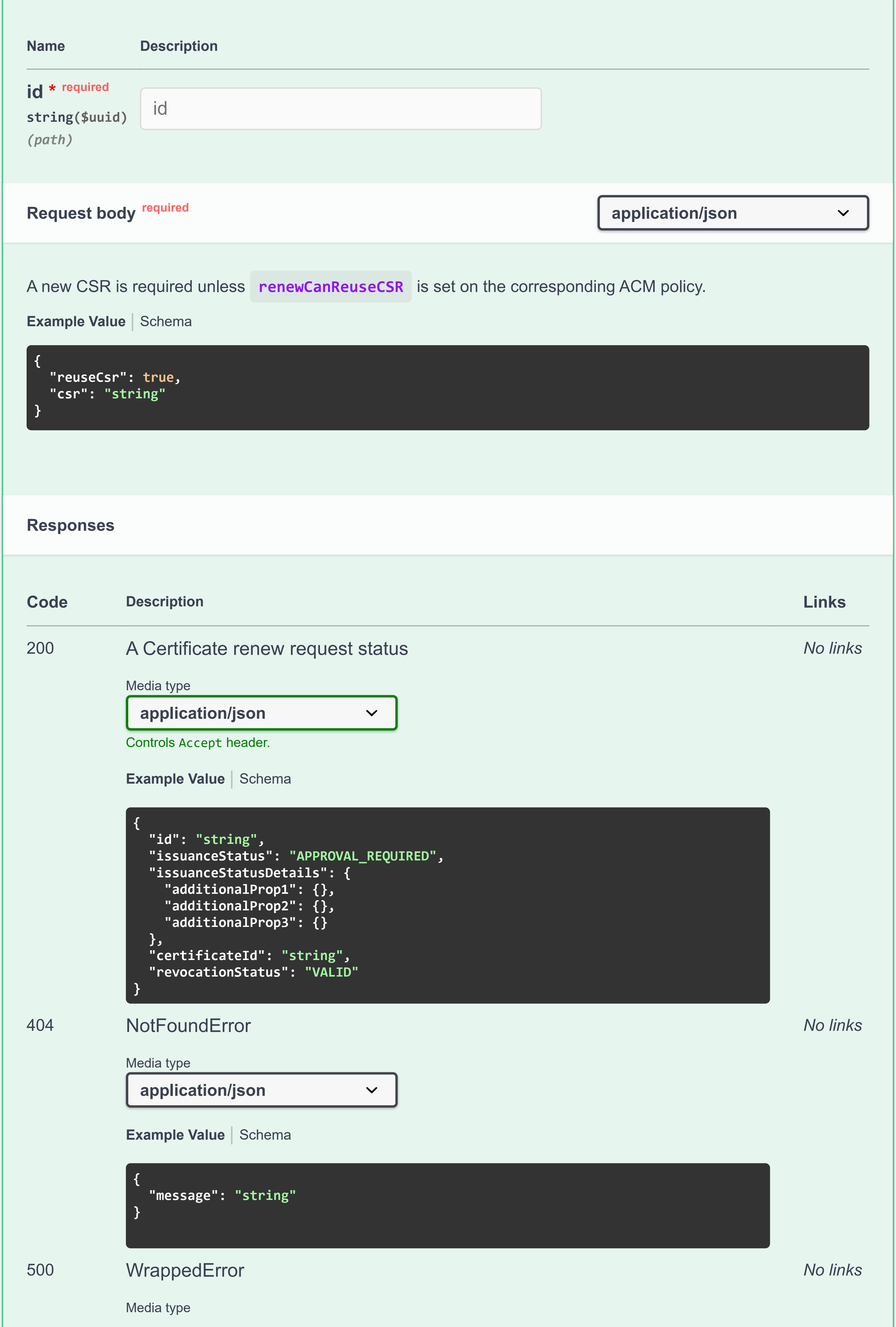
application/json

Example Value | Schema

```
{
  "message": "string"
}
```

POST `/certificates/{id}/renew` Renew a certificate given a certificate id.

Parameters



Name	Description
id * required string($uuid) *(path)*	id

Request body required

application/json

A new CSR is required unless `renewCanReuseCSR` is set on the corresponding ACM policy.

Example Value | Schema

```
{
  "reuseCsr": true,
  "csr": "string"
}
```

Responses

Code	Description	Links
200	A Certificate renew request status	*No links*

Media type

application/json

Controls `Accept` header.

Example Value | Schema

```
{
  "id": "string",
  "issuanceStatus": "APPROVAL_REQUIRED",
  "issuanceStatusDetails": {
    "additionalProp1": {},
    "additionalProp2": {},
    "additionalProp3": {}
  },
  "certificateId": "string",
  "revocationStatus": "VALID"
}
```

Code	Description	Links
404	NotFoundError	*No links*

Media type

application/json

Example Value | Schema

```
{
  "message": "string"
}
```

Code	Description	Links
500	WrappedError	*No links*

Media type

Code	Description	Links

application/json ⌄

Example Value | Schema

```
{
  "message": "string"
}
```

PATCH `/certificates/{id}/status` Set the renewed status of a certificate

Parameters

Name	Description
id * required `string($uuid)` *(path)*	id

Request body required

application/json ⌄

Contains the updated status for the certificate. Can be either "RENEWED" (to signify the certificate has been renewed already); or "VALID" to un-set the status from "RENEWED".

Example Value | Schema

```
{
  "revocationStatus": "VALID"
}
```

Responses

Code	Description	Links
200	A Certificate request status	*No links*

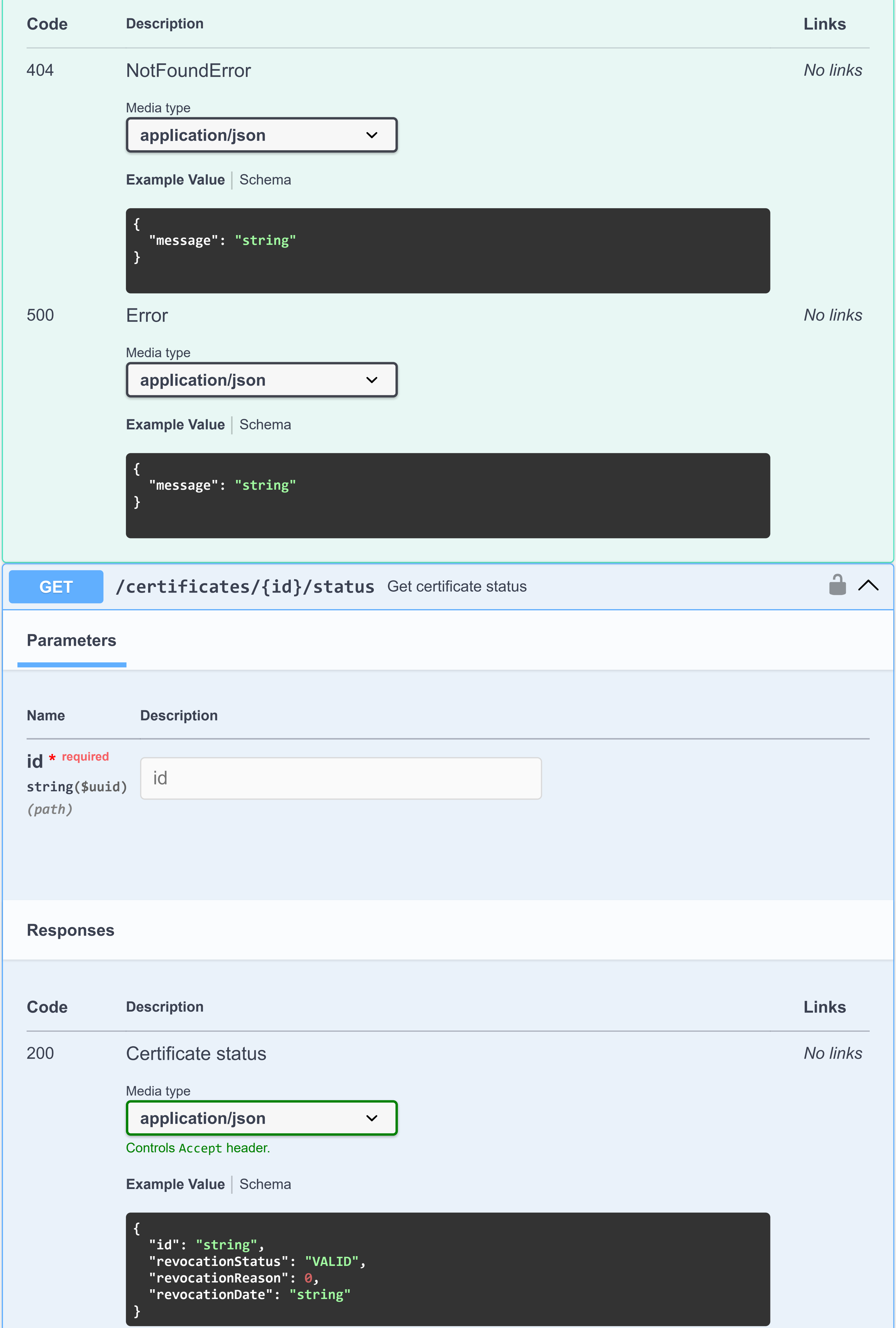
Media type

application/json ⌄

Controls `Accept` header.

Example Value | Schema

```
{
  "id": "string",
  "revocationStatus": "VALID",
  "revocationReason": 0,
  "revocationDate": "string"
}
```



Code	Description	Links
404	NotFoundError Media type application/json **Example Value** | Schema `{ "message": "string" }`	*No links*
500	Error Media type application/json **Example Value** | Schema `{ "message": "string" }`	*No links*

404 NotFoundError example value:

```
{
  "message": "string"
}
```

500 Error example value:

```
{
  "message": "string"
}
```

GET `/certificates/{id}/status` Get certificate status

Parameters

Name	Description
id * required `string($uuid)` *(path)*	id

Responses

Code	Description	Links
200	Certificate status Media type application/json Controls `Accept` header. **Example Value** | Schema	*No links*

```
{
  "id": "string",
  "revocationStatus": "VALID",
  "revocationReason": 0,
  "revocationDate": "string"
}
```

Code	Description	Links
404	NotFoundError	*No links*

Media type

application/json

Example Value | Schema

```
{
  "message": "string"
}
```

Certificate Requests

POST `/csr/` Create a certificate request

Parameters

No parameters

Request body required

application/json

Example Value | Schema

```
{
  "policy": "string",
  "csr": "string",
  "validity": {
    "years": 0,
    "months": 0,
    "days": 0,
    "notAfter": "string"
  },
  "dnComponents": {
    "CN": "string",
    "OU": [
      "string"
    ],
    "O": "string",
    "L": "string",
    "ST": "string",
    "C": "string",
    "DC": [
      "string"
    ],
    "EMAIL": "string",
    "AEID": "string"
  },
  "subjectAltNames": {
    "DNSNAME": [
      "string"
    ],
```

Responses

Code Description Links

200

Media type

application/json

Controls `Accept` header.

Example Value | Schema

```
{
  "id": "string",
  "issuanceStatus": "APPROVAL_REQUIRED",
  "issuanceStatusDetails": {
    "additionalProp1": {},
    "additionalProp2": {},
    "additionalProp3": {}
  },
  "certificateId": "string",
  "revocationStatus": "VALID"
}
```

No links

422 InvalidRequestError

Media type

application/json

Example Value | Schema

```
{
  "message": "string"
}
```

No links

500 Unknown

Media type

application/json

Example Value | Schema

```
{
  "message": "string"
}
```

No links

GET `/csr/{id}` Retrieves a given certRequest, including all details

Parameters

Name	Description
id * required `string($uuid)` *(path)*	id

Responses

<table>
<tr><th>Code</th><th>Description</th><th>Links</th></tr>
<tr><td>200</td><td>

A Certificate Request

Media type

application/json ⌄

Controls `Accept` header.

Example Value | Schema

```
{
  "id": "string",
  "source": "ACM",
  "fingerprint": "string",
  "csr": "string",
  "commonName": "string",
  "details": {
    "additionalProp1": {},
    "additionalProp2": {},
    "additionalProp3": {}
  },
  "issuanceStatus": "APPROVAL_REQUIRED",
  "createdAt": "string",
  "policy": {
    "id": "string",
    "name": "string"
  },
  "user": {
    "id": "string",
    "firstName": "string",
    "lastName": "string",
    "email": "string"
  }
}
```

</td><td>

No links

</td></tr>
<tr><td>404</td><td>

NotFoundError

Media type

application/json ⌄

Example Value | Schema

```
{
  "message": "string"
}
```

</td><td>

No links

</td></tr>
<tr><td>500</td><td>

WrappedError

Media type

application/json ⌄

Example Value | Schema

```
{
  "message": "string"
}
```

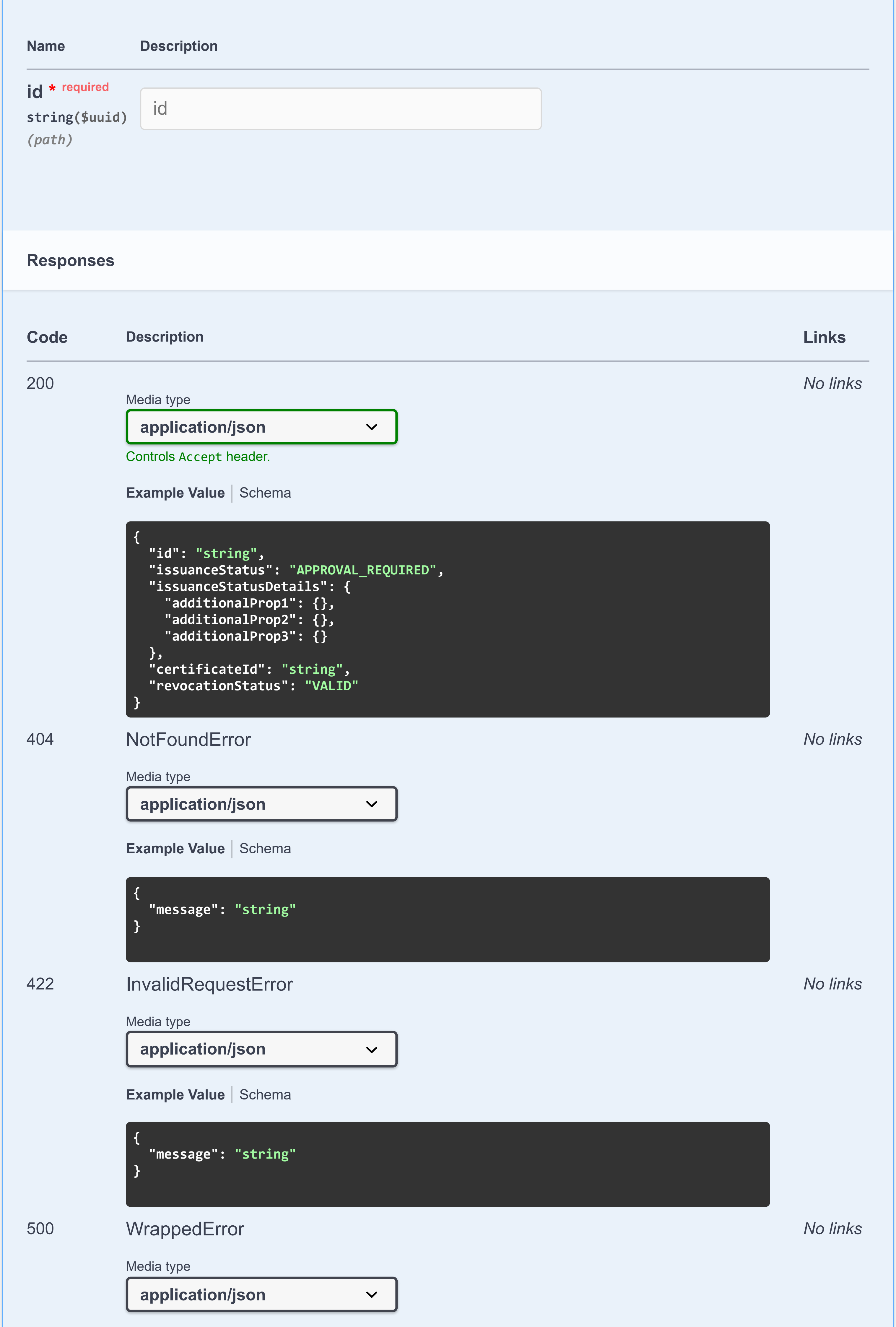
</td><td>

No links

</td></tr>
</table>

GET `/csr/{id}/status` Retrieves issuance status of certRequest, including certificateId if issued

Parameters



Name	Description
id * required string($uuid) *(path)*	id

Responses

Code	Description	Links
200		*No links*

Media type

application/json

Controls `Accept` header.

Example Value | Schema

```
{
  "id": "string",
  "issuanceStatus": "APPROVAL_REQUIRED",
  "issuanceStatusDetails": {
    "additionalProp1": {},
    "additionalProp2": {},
    "additionalProp3": {}
  },
  "certificateId": "string",
  "revocationStatus": "VALID"
}
```

Code	Description	Links
404	NotFoundError	*No links*

Media type

application/json

Example Value | Schema

```
{
  "message": "string"
}
```

Code	Description	Links
422	InvalidRequestError	*No links*

Media type

application/json

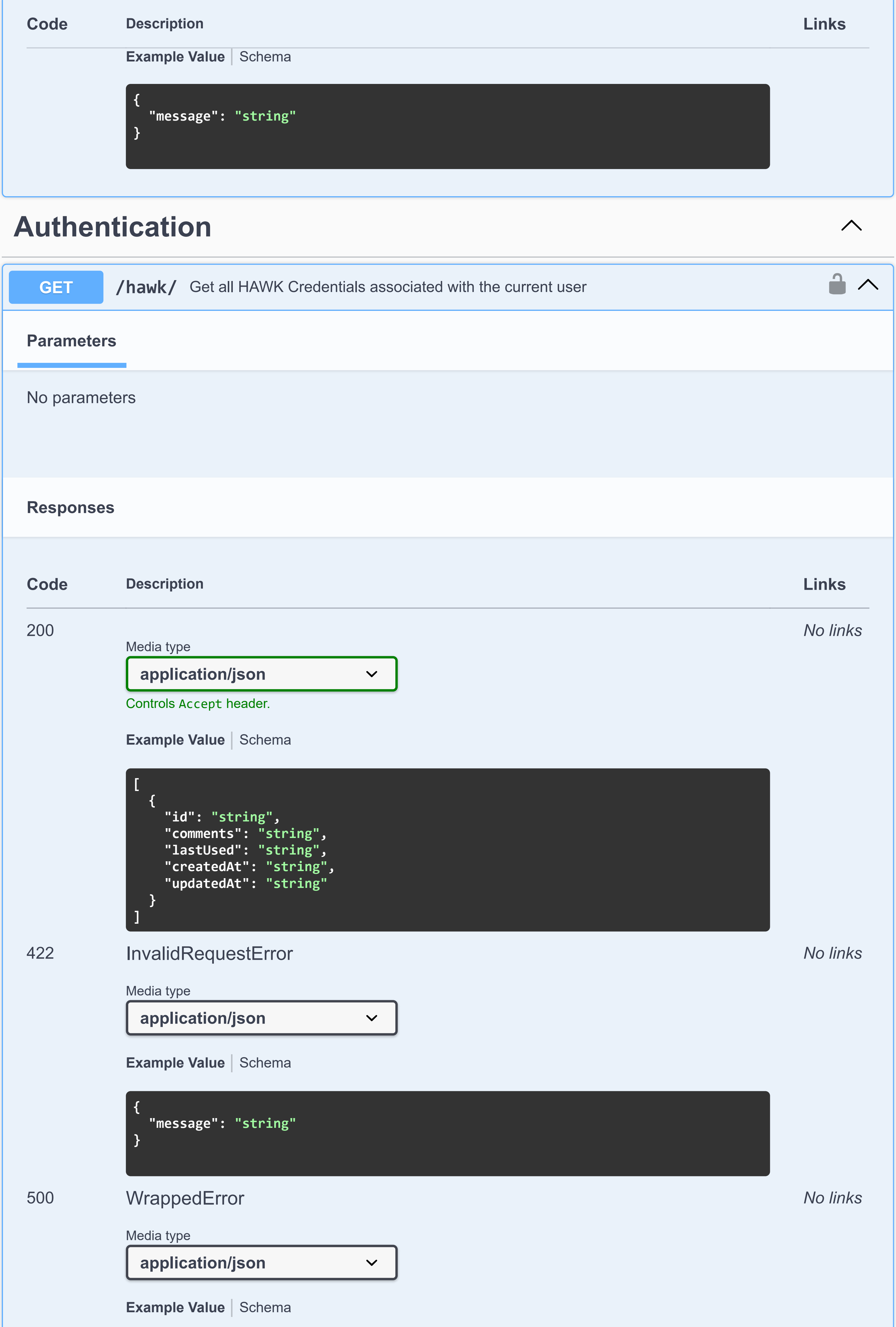
Example Value | Schema

```
{
  "message": "string"
}
```

Code	Description	Links
500	WrappedError	*No links*

Media type

application/json



Code	Description	Links
	Example Value | Schema	

```
{
  "message": "string"
}
```

Authentication

GET `/hawk/` Get all HAWK Credentials associated with the current user

Parameters

No parameters

Responses

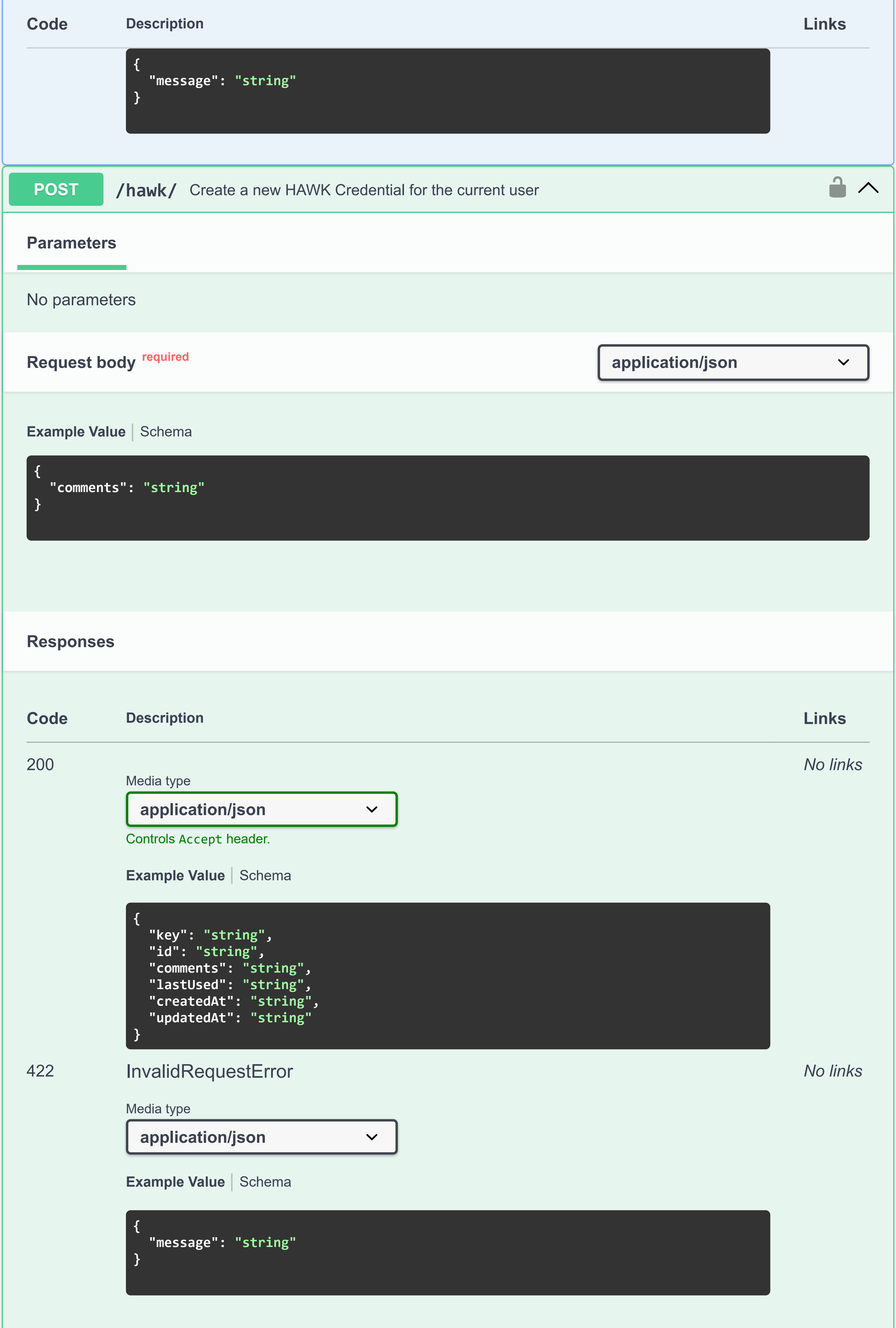
Code	Description	Links
200	Media type application/json Controls `Accept` header. Example Value | Schema	*No links*

```
[
  {
    "id": "string",
    "comments": "string",
    "lastUsed": "string",
    "createdAt": "string",
    "updatedAt": "string"
  }
]
```

Code	Description	Links
422	InvalidRequestError Media type application/json Example Value | Schema	*No links*

```
{
  "message": "string"
}
```

Code	Description	Links
500	WrappedError Media type application/json Example Value | Schema	*No links*



Code	Description	Links

```
{
  "message": "string"
}
```

POST /hawk/ Create a new HAWK Credential for the current user

Parameters

No parameters

Request body required

application/json

Example Value | Schema

```
{
  "comments": "string"
}
```

Responses

Code	Description	Links
200		*No links*

Media type

application/json

Controls `Accept` header.

Example Value | Schema

```
{
  "key": "string",
  "id": "string",
  "comments": "string",
  "lastUsed": "string",
  "createdAt": "string",
  "updatedAt": "string"
}
```

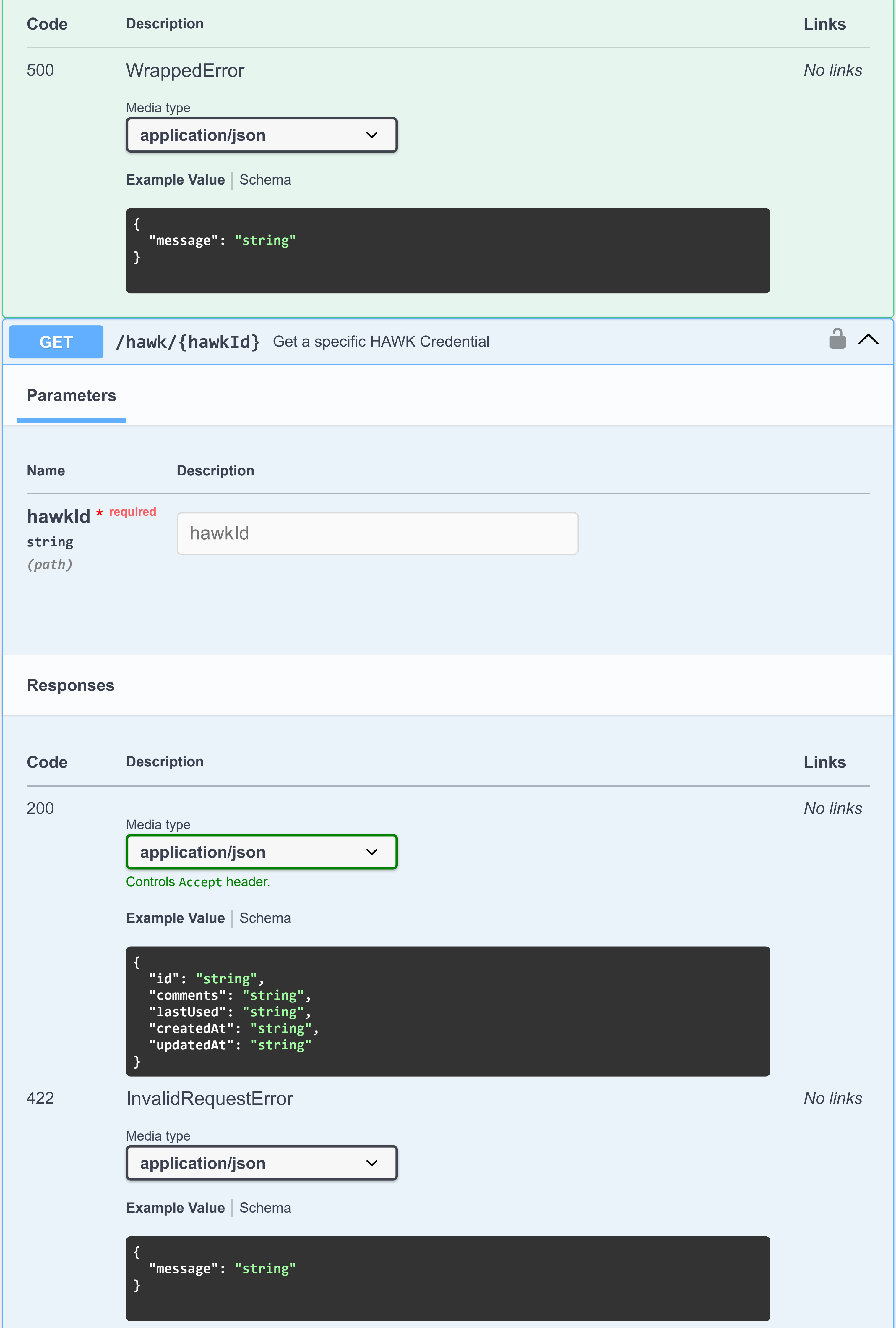
422	InvalidRequestError	*No links*

Media type

application/json

Example Value | Schema

```
{
  "message": "string"
}
```



Code	Description	Links
500	WrappedError Media type application/json Example Value | Schema	*No links*

```
{
  "message": "string"
}
```

GET `/hawk/{hawkId}` Get a specific HAWK Credential

Parameters

Name	Description
hawkId * required `string` *(path)*	hawkId

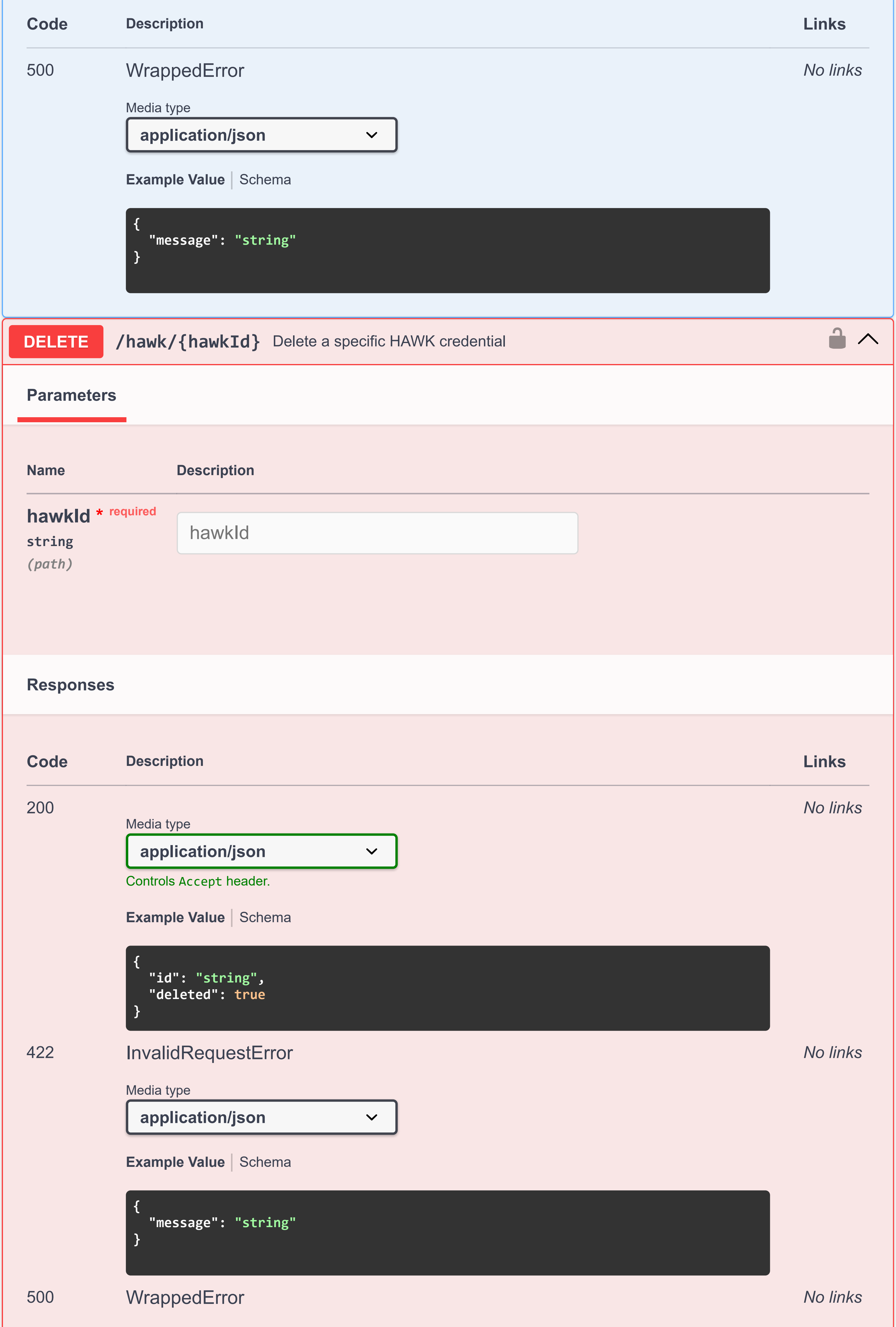
Responses

Code	Description	Links
200	Media type application/json Controls `Accept` header. Example Value | Schema	*No links*

```
{
  "id": "string",
  "comments": "string",
  "lastUsed": "string",
  "createdAt": "string",
  "updatedAt": "string"
}
```

Code	Description	Links
422	InvalidRequestError Media type application/json Example Value | Schema	*No links*

```
{
  "message": "string"
}
```



Code	Description	Links
500	WrappedError Media type application/json Example Value | Schema	*No links*

```
{
  "message": "string"
}
```

DELETE `/hawk/{hawkId}` Delete a specific HAWK credential

Parameters

Name	Description
hawkId * required `string` *(path)*	hawkId

Responses

Code	Description	Links
200	Media type application/json Controls `Accept` header. Example Value | Schema	*No links*

```
{
  "id": "string",
  "deleted": true
}
```

Code	Description	Links
422	InvalidRequestError Media type application/json Example Value | Schema	*No links*

```
{
  "message": "string"
}
```

Code	Description	Links
500	WrappedError	*No links*

Code	Description	Links
	Media type application/json ⌄ **Example Value** | Schema	

```
{
  "message": "string"
}
```

PUT `/hawk/{hawkId}` Create a new secret for a specific HAWK credential.

Parameters

Name	Description
hawkId * required `string` *(path)*	hawkId

Request body — application/json ⌄

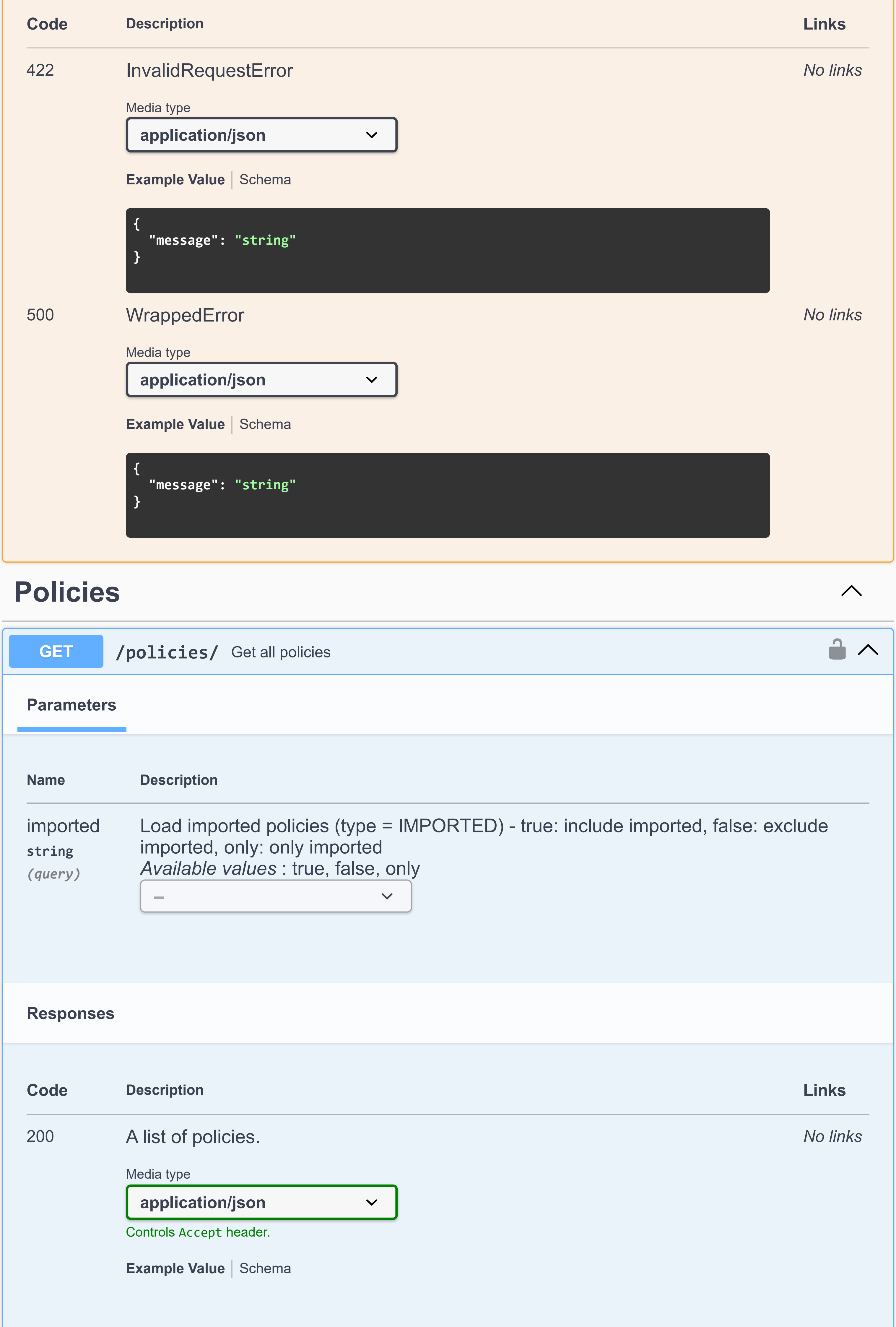
Example Value | Schema

```
{
  "comments": "string"
}
```

Responses

Code	Description	Links
200	Media type application/json ⌄ Controls `Accept` header. **Example Value** | Schema	*No links*

```
{
  "key": "string",
  "id": "string",
  "comments": "string",
  "lastUsed": "string",
  "createdAt": "string",
  "updatedAt": "string"
}
```



Code	Description	Links
422	InvalidRequestError	*No links*

Media type

application/json

Example Value | Schema

```
{
  "message": "string"
}
```

Code	Description	Links
500	WrappedError	*No links*

Media type

application/json

Example Value | Schema

```
{
  "message": "string"
}
```

Policies

GET `/policies/` Get all policies

Parameters

Name	Description
imported `string` *(query)*	Load imported policies (type = IMPORTED) - true: include imported, false: exclude imported, only: only imported *Available values* : true, false, only --

Responses

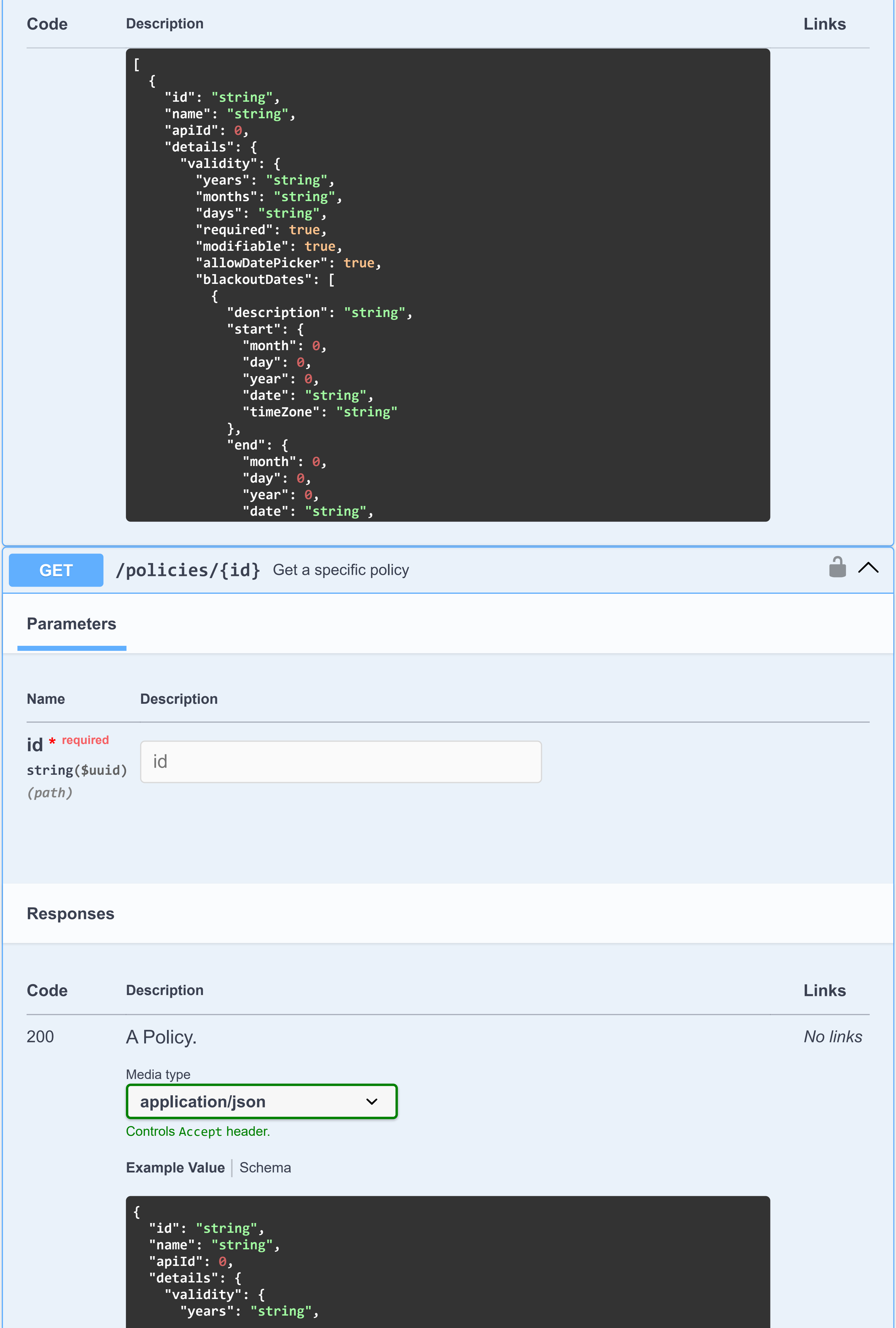
Code	Description	Links
200	A list of policies.	*No links*

Media type

application/json

Controls `Accept` header.

Example Value | Schema



Code	Description	Links

```
[
  {
    "id": "string",
    "name": "string",
    "apiId": 0,
    "details": {
      "validity": {
        "years": "string",
        "months": "string",
        "days": "string",
        "required": true,
        "modifiable": true,
        "allowDatePicker": true,
        "blackoutDates": [
          {
            "description": "string",
            "start": {
              "month": 0,
              "day": 0,
              "year": 0,
              "date": "string",
              "timeZone": "string"
            },
            "end": {
              "month": 0,
              "day": 0,
              "year": 0,
              "date": "string",
```

GET `/policies/{id}` Get a specific policy

Parameters

Name	Description
id * required string($uuid) *(path)*	id

Responses

Code	Description	Links
200	A Policy.	*No links*

Media type

application/json

Controls `Accept` header.

Example Value | Schema

```
{
  "id": "string",
  "name": "string",
  "apiId": 0,
  "details": {
    "validity": {
      "years": "string",
```

Code | Description | Links

```
      "months": "string",
      "days": "string",
      "required": true,
      "modifiable": true,
      "allowDatePicker": true,
      "blackoutDates": [
        {
          "description": "string",
          "start": {
            "month": 0,
            "day": 0,
            "year": 0,
            "date": "string",
            "timeZone": "string"
          },
          "end": {
            "month": 0,
            "day": 0,
            "year": 0,
            "date": "string",
            "timeZone": "string"
```

404 NotFoundError *No links*

Media type

application/json

Example Value | Schema

```
{
  "message": "string"
}
```

500 WrappedError *No links*

Media type

application/json

Example Value | Schema

```
{
  "message": "string"
}
```

Schemas

```
AccountConfigurationDigicert {
    description:           Do not modify this file directly. It is automatically generated from the
                           model definition.

    accountId*               > [...]
    organizationId*          > [...]
}
```

AccountConfigurationIdentrust ⌄ {

```
    description:          Do not modify this file directly. It is automatically generated from the
                          model definition.

    accountId*              > [...]
    accountType*            > [...]
    apiKey*                 > [...]
}
```

AccountConfigurationLogging ⌄ {

```
    description:          Do not modify this file directly. It is automatically generated from the
                          model definition.

    accountId*              > [...]
    bucketName*             > [...]
    useCompression*         > [...]
    accountName             > [...]
    regionName              > [...]
    pathPrefix              > [...]
    awsAccessKeyId          > [...]
}
```

CertRequest ⌄ {

```
    id*                     > [...]
    source*               CertRequestSource > [...]
    fingerprint*            > [...]
    csr*                    > [...]
    commonName*             > [...]
    details*                > {...}
    issuanceStatus*       IssuanceStatus > [...]
    createdAt*              > [...]
    policy*               NamedObject > {...}
    user*                 UserMinimal > {...}
}
```

```
CertRequestBody ⌄ {
    policy*                        > [...]
    csr*                           > [...]
    validity                       CertRequestBodyValidity > {...}
    dnComponents*                  CertRequestBodyDnComponents > {...}
    subjectAltNames                CertRequestBodySubjectAltNames > {...}
    customFields                   > {...}
    customExtensions               > {...}
    comment                        > [...]
    expiryEmails                   > [...]
    clearRemindersCertificateId    > [...]
    CN                             > [...]
}
```

```
CertRequestBodyDnComponents ⌄ {
    CN                       > [...]
    OU                       > [...]
    O                        > [...]
    L                        > [...]
    ST                       > [...]
    C                        > [...]
    DC                       > [...]
    EMAIL                    > [...]
    AEID                     > [...]
}
```

```
CertRequestBodySubjectAltNames ⌄ {
    DNSNAME                  > [...]
    IPADDRESS                > [...]
    RFC822NAME               > [...]
    UPN                      > [...]
    URI                      > [...]
}
```

```
CertRequestBodyValidity ⌄ {
    years                    > [...]
    months                   > [...]
    days                     > [...]
    notAfter                 > [...]
}
```

```
CertRequestSource ⌄ string
Enum:
  › Array [ 8 ]
```

```
CertRequestStatus ⌄ {
    id*                      › [...]
    issuanceStatus*          IssuanceStatus › [...]
    issuanceStatusDetails    › {...}
    certificateId            › [...]
    revocationStatus         CertRequestStatusRevocationStatus › [...]
}
```

```
CertRequestStatusRevocationStatus ⌄ string
Enum:
  › Array [ 5 ]
```

```
Certificate ⌄ {
    id*                      › [...]
    serial*                  › [...]
    commonName*              › [...]
    certRequestId            › [...]
    subjectDN*               › [...]
    issuerDN*                › [...]
    notBefore*               › [...]
    notAfter*                › [...]
    signatureAlgorithm*      › [...]
    revocationStatus*        RevocationStatusEnum › [...]
    revocationReason*        › [...]
    revocationDate           › [...]
    pem*                     › [...]
    imported*                › [...]
    discovered*              › [...]
    selfSigned*              › [...]
    createdAt*               › [...]
    SANs*                    › [...]
    policy*                  NamedObject › {...}
    user*                    UserMinimal › {...}
    account*                 NamedObject › {...}
    organization*            NamedObject › {...}
    expiryNotifications*     › [...]
}
```

```
CertificateStatus ⌄ {
    id*                     › [...]
    revocationStatus*       RevocationStatusEnum › [...]
    revocationReason        RevocationReasons › [...]
    revocationDate          › [...]
}
```

```
GetCertificatesPayload ⌄ {
    common_name             › [...]
    serial                  › [...]
    not_before              › [...]
    not_after               › [...]
    expired                 › [...]
    renewed                 › [...]
    created_since           › [...]
    updated_since           › [...]
    status                  RevocationStatusEnum › [...]
    owner                   › [...]
    account                 › [...]
    organization            › [...]
    policy                  › [...]
    fingerprint             › [...]
    limit                   › [...]
    offset                  › [...]
    sort_type               › [...]
    sort_direction          SortDirectionEnum › [...]
    pem                     › [...]
    span                    › [...]
    imported                › [...]
    discovered              › [...]
    selfSigned              › [...]
}
```

```
GetCertificatesResponse ⌄ {
    count                   › [...]
    items                   › [...]
}
```

```
GetCertificatesResponseItem ⌄ {
    id                      › [...]
    commonName              › [...]
    serial                  › [...]
    notBefore               › [...]
    notAfter                › [...]
    revocationStatus        RevocationStatusEnum › [...]
    SANs                    › [...]
    policy                  NameObject › {...}
    imported*               › [...]
    discovered*             › [...]
    selfSigned*             › [...]
}
```

```
HawkCredential ⌄ {
    description:            Do not modify this file directly. It is automatically generated from the
                            model definition.

    id*                     › [...]
    comments                › [...]
    lastUsed                › [...]
    createdAt*              › [...]
    updatedAt*              › [...]
}
```

```
HawkCredentialComment ⌄ {
    comments                › [...]
}
```

```
HawkCredentialComments ⌄ {
    comments                › [...]
}
```

```
HawkCredentialDeleteResults ⌄ {
    id                      › [...]
    deleted                 › [...]
}
```

```
HawkCredentialWithKey ⌄ {
    key*                    > [...]
    id*                     > [...]
    comments                > [...]
    lastUsed                > [...]
    createdAt*              > [...]
    updatedAt*              > [...]
}
```

```
IssuanceStatus ⌄ string
Enum:
  > Array [ 8 ]
```

```
MockPolicy ⌄ {
    description:            Do not modify this file directly. It is automatically generated from the
                            model definition.

    id*                     > [...]
    certificateType*        > [...]
    organizationId*         > [...]
}
```

```
NameObject ⌄ {
    name                    > [...]
}
```

```
NamedObject ⌄ {
    id*                     > [...]
    name*                   > [...]
}
```

```
Policy ⌄ {
    id*                      > [...]
    name*                    > [...]
    apiId*                   > [...]
    details*                PolicyDetails > {...}
    enabled*                PolicyEnabled > {...}
    organizationId*          > [...]
    certificateAuthorityId*  > [...]
    MockPolicy              MockPolicy > {...}
}
```

```
PolicyDetails ⌄ {
    validity              PolicyDetailsValidity > {...}
    dnComponents*         > [...]
    subjectAltNames       > [...]
    approvalRequired      > [...]
    expiryEmails          PolicyDetailsExpiryEmails > {...}
    customFields          > [...]
    customExtensions      > [...]
    validator             > [...]
    renewCanReuseCSR      > [...]
}
```

```
PolicyDetailsCustomExtensions ⌄ {
    description:          Do not modify this file directly. It is automatically generated from the
                          model definition.

    oid                   > [...]
    label                 > [...]
    required              > [...]
    modifiable            > [...]
    defaultValue          > [...]
}
```

```
PolicyDetailsCustomFields ⌄ {
    description:          Do not modify this file directly. It is automatically generated from the
                          model definition.

    tag                   > [...]
    label                 > [...]
    required              > [...]
    modifiable            > [...]
    defaultValue          > [...]
    values                > [...]
}
```

```
PolicyDetailsDnComponents ⌄ {
    tag                   PolicyDetailsDnComponentsTag > [...]
    label                 > [...]
    required              > [...]
    modifiable            > [...]
    defaultValue          > [...]
    copyAsFirstSAN        > [...]
    regEx                 > [...]
}
```

```
PolicyDetailsDnComponentsTag ⌄ string
Enum:
  › Array [ 9 ]
```

```
PolicyDetailsExpiryEmails ⌄ {
    tag                  PolicyDetailsExpiryEmailsTag › [...]
    label                 › [...]
    required              › [...]
    modifiable            › [...]
    defaultValue          › [...]
}
```

```
PolicyDetailsExpiryEmailsTag ⌄ string
Enum:
  › Array [ 1 ]
```

```
PolicyDetailsSubjectAltNames ⌄ {
    tag                  PolicyDetailsSubjectAltNamesTag › [...]
    label                 › [...]
    required              › [...]
    modifiable            › [...]
    defaultValue          › [...]
    regEx                 › [...]
}
```

```
PolicyDetailsSubjectAltNamesTag ⌄ string
Enum:
  › Array [ 5 ]
```

```
PolicyDetailsValidity ⌄ {
    years                 › [...]
    months                › [...]
    days                  › [...]
    required              › [...]
    modifiable            › [...]
    allowDatePicker       › [...]
    blackoutDates         › [...]
    maxValue             ValidityPeriod › {...}
    defaultValue         ValidityPeriod › {...}
}
```

```
PolicyDetailsValidityBlackoutDate ⌄ {
    description:          Do not modify this file directly. It is automatically generated from the
                          model definition.

    month                  › [...]
    day                    › [...]
    year                   › [...]
    date                   › [...]
    timeZone               › [...]
}
```

```
PolicyDetailsValidityBlackoutDates ⌄ {
    description            › [...]
    start*                PolicyDetailsValidityBlackoutDate › {...}
    end*                  PolicyDetailsValidityBlackoutDate › {...}
    timeZone               › [...]
}
```

```
PolicyEnabled ⌄ {
    description:          Do not modify this file directly. It is automatically generated from the
                          model definition.

    ui                     › [...]
    rest                   › [...]
    acme                   › [...]
    scep                   › [...]
    est                    › [...]
}
```

```
RenewCertificatePayload ⌄ {
    reuseCsr               › [...]
    csr                    › [...]
}
```

```
RenewCertificateStatus ⌄ {
    revocationStatus      SetRenewStatusEnum › [...]
}
```

```
ResultError ⌄ {
    message*               › [...]
}
```

RevocationReasons ⌄ number

Enum:

› Array [5]

RevocationStatusEnum ⌄ string

Enum:

› Array [7]

RevokeCertificateReasonIssuerDN ⌄ {

```
    reason*            RevocationReasons › [...]
    revocationDate     › [...]
    issuerDN           › [...]
}
```

SetRenewStatusEnum ⌄ string

Enum:

› Array [2]

SortDirectionEnum ⌄ string

Enum:

› Array [2]

UserMinimal ⌄ {

```
    description:       Do not modify this file directly. It is automatically generated from the
                       model definition.

    id*                › [...]
    firstName*         › [...]
    lastName*          › [...]
    email              › [...]
}
```

ValidityPeriod ⌄ {

```
    description:       Do not modify this file directly. It is automatically generated from the
                       model definition.

    years              › [...]
    months             › [...]
    days               › [...]
}
```