Access Revocation API v1 Specification

Overview

The Access Revocation API works in conjunction with the Segmented Media Protection feature, "Token Authentication" available for use with Adaptive Media Delivery (AMD). This API lets you generate a "blacklist" of specific Akamai Edge Tokens, using their unique "token identifier" to flag them as revoked. A request that includes a revoked token fails access checks and the request is blocked by Akamai. You can set a time to live for this revocation period to have these tokens automatically "unrevoked," or you can manually remove tokens from a blacklist with the API. The API also lets you review your blacklists and token revocation settings.

Resources

This section provides details on the API's various operations.

API summary

Operation	Method	Endpoint
List all blacklists	GET	/taas/v1/blacklists
Add a blacklist	POST	/taas/v1/blacklists
Delete a blacklist	DELETE	/taas/v1/blacklists/{blacklistId}
List blacklist ARL properties	GET	/taas/v1/blacklists/{blacklistId}/properties
Get blacklist identifier count information	GET	/taas/v1/blacklists/{blacklistId}/meta
List all identifiers	GET	/taas/v1/blacklists/{blacklistId}/identifiers
Revoke tokens	POST	/taas/v1/blacklists/{blacklistId}/identifiers/add
Unrevoke tokens	POST	/taas/v1/blacklists/{blacklistId}/identifiers/remove
Get an identifier	GET	/taas/v1/blacklists/{blacklistId}/identifiers/{tokenId}

List all blacklists

Returns a list of all blacklists.

GET `/taas/v1/blacklists`

Status 200 `application/json`

Object type: [Blacklist](#)

Response body:

```
[
    {
        "id": 1,
        "name": "customer-blacklist1",
        "contractId": "1-ABCDE",
        "createdTime": 1569238487,
        "createdBy": "ccare2"
    },
    {
        "id": 2,
        "name": "customer-blacklist2",
        "contractId": "2-BCDE",
        "createdBy": "ccare2",
        "createdTime": 1569238504
    }
]
```

Steps:

1. Make a GET request to `/taas/v1/blacklists`.
2. Review the list of the [blacklist](#) objects returned. (The response is an array of objects, not a single object as shown in the example here.)

Add a blacklist

Add a new list to house token identifiers that you want revoked from access. Currently, only a single blacklist is supported. There isn't an operation that lets you modify these settings for a blacklist, so be sure you provide the proper details when adding. However, you can [revoke](#) or [unrevoke](#) token identifiers from this blacklist, as necessary.

POST `/taas/v1/blacklists`

Content-Type: `application/json`

Object type: [Blacklist](#)

Request body:

```
{
    "name": "Baseball-ws-2019",
    "contractId": "1-ABCDE"
}
```

Status 202 `application/json`

Object type: [Blacklist](#)

Response body:

```
{
    "id": 1,
    "name": "Baseball-ws-2019",
    "contractId": "1-ABCDE"
}
```

Steps:

1. If necessary, work with your account representative to obtain to Akamai `contractId` assigned to your instance of Access Revocation.
2. Build a new [blacklist](#) object. Include a unique `name` for the blacklist and also include your applicable `contractId`.
3. POST the object to `/taas/v1/blacklists`.
4. The operation responds with a [blacklist](#) object. Store the `id` value from the response. (This is the `blacklistId` used in several other operations.)
5. [Revoke tokens](#) to add token identifiers to the blacklist.

Delete a blacklist

Used to delete a specific blacklist.

DELETE `/taas/v1/blacklists/{blacklistId}`

Sample: `/taas/v1/blacklists/120964235098767`

Parameter	Type	Sample	Description
URL path parameters			
`blacklistId`	Number	`120964235098767`	A unique numeric identifier Akamai assigns to a blacklist.

Status 204

Steps:

1. List all blacklists and store the `id` of the blacklist you want to delete. (This is its `blacklistId`.)
2. Make a DELETE request to `/taas/v1/blacklists/{blacklistId}`.

List blacklist ARL properties

Get a list of Property Manager-specific information for each AMD property that uses this blacklist.

GET `/taas/v1/blacklists/{blacklistId}/properties`

Sample: `/taas/v1/blacklists/120964235098767/properties`

Parameter	Type	Sample	Description
URL path parameters			
`blacklistId`	Number	`120964235098767`	A unique numeric identifier Akamai assigns to a blacklist.

Status 200 `application/json`
Object type: Properties
Response body:

```
[
    {
        "arlFileId": 12345,
        "propertyId": 3456789,
        "propertyName": "customer-foo.com"
    },
    {
        "arlFileId": 56789,
        "propertyId": 12345678,
        "propertyName": "customer-bar.com"
    }
]
```

Steps:

1. List all blacklists and store the `id` for the blacklist you want to review ARL properties. (This is its `blacklistId`.)
2. Make a GET request to `/taas/v1/blacklists/{blacklistId}/properties`.
3. Review the list of the properties objects returned. (The response is an array of objects, not a single object as shown in the example here.)

Get blacklist identifier count information

The current count and maximum number of identifiers allowed in the blacklist.

GET `/taas/v1/blacklists/{blacklistId}/meta`

Sample: `/taas/v1/blacklists/120964235098767/meta`

Parameter	Type	Sample	Description
URL path parameters			
`blacklistId`	Number	`120964235098767`	A unique numeric identifier Akamai assigns to a blacklist.

Status 200 `application/json`

Object type: Meta

Response body:

```
{
    "count": 500,
    "limit": 25000
}
```

Steps:

1. List all blacklists and store the `id` for the blacklist you want to review count total information. (This is its `blacklistId`.)
2. Make a GET request to `/taas/v1/blacklists/{blacklistId}/meta`.
3. Review the list of meta objects that's returned.

List all identifiers

Lists all of the identifiers that belong to a specific blacklist and review their details.

GET /taas/v1/blacklists/{blacklistId}/identifiers
Sample: /taas/v1/blacklists/120964235098767/identifiers

Parameter	Type	Sample	Description
URL path parameters			
blacklistId	Number	120964235098767	A unique numeric identifier Akamai assigns to a blacklist.

Status 200 application/json
Object type: Identifier
Response body:

```
[
    {
        "id": "sqwoieksjbsdf3455",
        "ttl": 86400
    },
    {
        "id": "9082349u534589824",
        "ttl": 21600
    },
    {
        "id": "68sd899ff09sdf8787",
        "ttl": 25200
    }
]
```

Steps:

1. List all blacklists and store the id for the blacklist you want to review identifiers. (This is its blacklistId.)
2. Make a GET request to /taas/v1/blacklists/{blacklistId}/identifiers.
3. Review the list of identifier objects that's returned. (The response is an array of objects, not a single object as shown in the example here.)

Revoke tokens

Add one or more token identifiers to a blacklist to set them as revoked.

POST `/taas/v1/blacklists/{blacklistId}/identifiers/add`
Sample: `/taas/v1/blacklists/120964235098767/identifiers/add`
Content-Type: `application/json`
Object type: [Identifier](#)
Request body:

```
[
    {
        "id": "sdasd345466dg",
        "durationSeconds": 18000
    },
    {
        "id": "utrfhasdf8990",
        "durationSeconds": 3600
    }
]
```

Parameter	Type	Sample	Description
URL path parameters			
`blacklistId`	Number	`120964235098767`	A unique numeric identifier Akamai assigns to a blacklist.

Status 200 `application/json`
Object type: [Meta](#)
Response body:

```
{
    "count": 500,
    "limit": 25000
}
```

Steps:

1. If you haven't already, you need to <u>add a blacklist</u>. Ensure that you've also stored the `id` assigned to the blacklist--it's `blacklistId`. (You can <u>list all blacklists</u> to view your existing blacklists to gather this value.)
2. Obtain the `session_id` from the access token that you've configured to support Token Authentication. This value serves as the token identifier `id`. (See the *<u>AMD - Implementation Guide</u>* for Token Authentication instructions.)
3. Build a new <u>Identifier</u> object. Add objects for each token you want revoked. Include the identifier `id` for each, and optionally include the `durationSeconds` member to set a time to live for the revocation.
4. POST the object to `/taas/v1/blacklists/{blacklistId}/identifiers/add`.
5. The operation responds with the <u>meta</u> object, that shows the current count of token identifiers in the blacklist and the maximum number of identifiers it can house.

Unrevoke tokens

Remove token identifiers from a blacklist to "unrevoke" them.

POST `/taas/v1/blacklists/{blacklistId}/identifiers/remove`

Sample: `/taas/v1/blacklists/120964235098767/identifiers/remove`

Content-Type: `application/json`

Object type: <u>Identifier</u>

Request body:

```
[
    "sqwoieksjbsdf3455",
    "9082349u534589824",
    "68sd899ff09sdf8787"
]
```

Parameter	Type	Sample	Description
URL path parameters			
`blacklistId`	Number	`120964235098767`	A unique numeric identifier Akamai assigns to a blacklist.

Status 200 `application/json`

Object type: <u>Meta</u>

Response body:

```
{
    "count": 500,
    "limit": 25000
}
```

Steps:

1. List all blacklists and store the `id` for the blacklist you want to unrevoke tokens. (This is its `blacklistId`.)
2. Obtain the `session_id` from the access tokens that you want to unrevoke. This value serves as the token identifier `id`. (Token Authentication is described in the *AMD - Implementation Guide*.)
3. Build a new Identifier object. Include the identifier `id` for each token identifier, and separate multiple entries with a comma.
4. POST the object to `/taas/v1/blacklists/{blacklistId}/identifiers/remove`.
5. The operation responds with the meta object, that shows the current count of token identifiers in the blacklist and the maximum number of identifiers it can house.

Get an identifier

Get a specific identifier that belongs to a specific blacklist and review its details.

GET `/taas/v1/blacklists/{blacklistId}/identifiers/{tokenId}`
Sample: `/taas/v1/blacklists/120964235098767/identifiers/sqwoieksjbsdf3455`

Parameter	Type	Sample	Description
URL path parameters			
`blacklistId`	Number	`120964235098767`	A unique numeric identifier Akamai assigns to a blacklist.
`tokenId`	String	`sqwoieksjbsdf3455`	The token identifier portion of the complete Akamai Token. A token identifier can contain a maximum of 36 alphanumeric characters, hyphens, or underscores.

Status 200 `application/json`
Object type: Identifier

Response body:

```
{
    "id": "sqwoieksjbsdf3455",
    "ttl": 86400
}
```

Steps:

1. List all blacklists and store the `id` for the blacklist you want to review. (This is its `blacklistId`.)
2. List all identifiers and store the `id` for the desired token identifier.
3. Make a GET request to `/taas/v1/blacklists/{blacklistId}/identifiers/{tokenId}`.
4. Review the specific identifier that's returned.

Data

The data schema tables below list membership requirements as follows:

✓	Member is required in requests, or always present in responses, even if its value is empty or `null`.
✗	Member is out of scope, and irrelevant to the specified interaction context. If you include the member in that context, it either triggers an error, or is ignored.

Blacklist

A list of all blacklists, and the parameters set for each.

Sample GET response:

```
[
    {
        "id": 1,
        "name": "customer-blacklist1",
        "contractId": "1-ABCDE",
        "createdTime": 1569238487,
        "createdBy": "ccare2"
    },
    {
        "id": 2,
        "name": "customer-blacklist2",
```

```
        "contractId": "2-BCDE",
        "createdBy": "ccare2",
        "createdTime": 1569238504
    }
]
```

Blacklist members

Member	Type	GET	POST	Description
`Blacklist`: A list of all blacklists, and the parameters set for each.				
`contractId`	String	✓	✓	The Akamai-assigned Contract ID for your instance of Access Revocation.
`createdBy`	String	✓	✗	The Control Center login ID used to create the blacklist.
`createdTime`	Integer	✓	✗	The Unix epoch time when the blacklist was created.
`id`	String	✓	✗	A unique numeric identifier assigned to a blacklist.
`name`	String	✓	✓	A friendly name for the blacklist. The name can only contain alphanumeric characters and hyphens.

Properties

A list of Akamai resource locator (ARL) file information for each AMD property that uses a blacklist.

Sample GET response:

```
[
    {
        "arlFileId": 12345,
        "propertyId": 3456789,
        "propertyName": "customer-foo.com"
    },
    {
        "arlFileId": 56789,
        "propertyId": 12345678,
        "propertyName": "customer-bar.com"
```

```
    }
]
```

Properties members

Member	Type	Required	Description
Properties: A list of Akamai resource locator (ARL) file information for each AMD property that uses a blacklist.			
arlFileId	Integer	✓	The unique ARL file identifier assigned to the AMD property that has Token Revocation enabled.
propertyId	Integer	✓	The unique property ID assigned to the AMD property.
propertyName	String	✓	The property name set for the AMD property.

Meta

The current count of token identifiers in a blacklist and the maximum number of identifiers it can house.

Sample GET response:

```
{
    "count": 500,
    "limit": 25000
}
```

Meta members

Member	Type	Required	Description
Meta: The current count of token identifiers in a blacklist and the maximum number of identifiers it can house.			
count	Integer	✓	The total number of token identifiers in the blacklist.
limit	Integer	✓	The maximum number of token identifiers that can be added to the blacklist.

Identifier

A list of all token authentication tokens, and the parameters set for each.

Sample GET response:

```
[
    {
        "id": "sqwoieksjbsdf3455",
        "ttl": 86400
    },
    {
        "id": "9082349u534589824",
        "ttl": 21600
    },
    {
        "id": "68sd899ff09sdf8787",
        "ttl": 25200
    }
]
```

Identifier members

Member	Type	Required	Description
`Identifier`: A list of all token authentication tokens, and the parameters set for each.			
`id`	String	✓	The unique identifier assigned to the token.
`ttl`	Integer	✓	The time to live (TTL) for the blacklist, in seconds. After this time elapses, Akamai removes this token from the blacklist.

Errors

This section provides details on the data object that reflects the API's common response to error cases, and lists the API's range of response status codes for both error and success cases.

Error responses

When the API encounters an error, it responds with an error code and an error message that provides details that can be useful for debugging.

This shows a typical error response when you invoke an operation a non-existent `blacklistId`.

```
{
  "type": "resource-not-found",
  "title": "Resource Not Found",
  "instance": "0a037412-6aac-4c7f-9e45-b29fb874db86",
  "status": 404,
  "detail": "Resource Not Found (details=[No blacklist exists with given ID 123.])"
}
```

HTTP status codes

The API produces this set of HTTP status codes for both success and failure scenarios:

Code	Description
200	The operation was successful.
202	Accepted.
204	No content.
400	Bad Request.
403	Access is forbidden.
404	Not Found.
500	Internal Server Error.