

# Content Control Utility (CCU) V3 API – with Fast Purge



This API allows you to purge any set of cached URLs, Content Provider (CP) codes, and cache tags, but much faster. With Fast Purge enabled, version 3 (v3) purges take about five seconds to propagate globally. Fast Purge is available to all customers. <https://developer.akamai.com/api/purge/ccu/overview.html>

Note: You can invalidate content on Staging and Production networks. The Production network is assumed by default.

## Invalidate URLs

1. Create a request expressed as server paths for a set of objects:

```
{
  "objects": [
    "https://www1.example.com/some/path",
    "http://www2.example.com/some/other/path"
  ]
}
```

2. **POST** the request to `/ccu/v3/invalidate/url/production` or to `/ccu/v3/invalidate/url/staging`.

## Invalidate CP Codes (limited availability)

1. Create a request for a set of objects expressed as CP codes:

```
{
  "objects": [
    12345,
    98765
  ]
}
```

2. **POST** the request to `/ccu/v3/invalidate/cpcode/production` or to `/ccu/v3/invalidate/cpcode/staging`.

## Invalidate Cache Tags (available to select beta customers only)

1. Create a request for a set of objects expressed as cache tags:

```
{
  "objects": [
    "black-friday",
    "electronics",
    "laptops",
    "tablets"
  ]
}
```

2. **POST** the request to `/ccu/v3/invalidate/tag/production` or to `/ccu/v3/invalidate/tag/staging`.

**NOTE:** In addition to Invalidate, the API also supports Delete method that removes rather than invalidates content identified by URLs, CP codes, and cache tags. Delete should be used judiciously as it results in a higher load on your origin compared to Invalidate.

## Update the Origin's Last-Modified

When making a request, edge servers send an **If-Modified-Since** request to the origin. Make sure to serve your replacement origin content with an updated **Last-Modified** timestamp header.

## Check a Purge's Completion Time

When **POST** ing a new purge request, we recommend that you log the response object in case you need to contact Technical Support.

```
{
  "httpStatus": 201,
  "detail": "Request accepted.",
  "estimatedSeconds": 5,
  "purgeId": "043f-4af0-843f-aaf0043faaf0",
  "supportId": "17PY1321286429616716-211907680"
}
```

# Content Control Utility (CCU) V2



This API allows you to purge any set of cached URLs or content controlled under a CP code. However, if you have set up your content to cache on Akamai's edge servers, this overrides that behavior and forces a refresh from your origin. Purges with version 2 (v2) take approximately four to seven minutes to propagate globally.

<https://developer.akamai.com/api/purge/ccu-v2/overview.html>

## Purge a Set of URLs

1. Choose whether you want to purge the Staging or Production network.
2. Collect a set of URL objects to purge:

```
{
  "objects": [
    "http://www.example.com/graphics/picture.gif",
    "http://www.example.com/documents/brochure.pdf"
  ],
  "action": "remove",
  "type": "ar1",
  "domain": "production"
}
```

3. **POST** to `/ccu/v2/queues/default`.

## Purge Everything Under a CP Code

This purges logically grouped sets of URLs.

1. Create a request with your numeric CP codes as the objects:

```
{
  "objects" : [ 6848, 44 ],
  "action": "remove",
  "type": "cpcode"
}
```

2. **POST** to `/ccu/v2/queues/default`

## Purge High-Priority Items

Purging high-priority items works the same as CCU V2 Purge, but requests should be **POST** ed to the `/ccu/v2/queues/emergency` endpoint. Requests in the emergency queue go to the head of the line, before default requests, but are still constrained by the CCU V2 purge processing frequency rate. Only 10 emergency requests are allowed at a time. Consider using CCU V3 Fast Purge for purging items with tight time constraints.

## Check Status of a Purge

This verifies that your purge request was accepted.

1. When **POST** ing a new purge request, store the response object:

```
{
  "purgeId": "57799d8b-10e4-11e4-9088-62ece60caaf0",
  "estimatedSeconds": 420,
  "pingAfterSeconds": 420,
  "progressUri": "/ccu/v2/purges/57799d8b-10e4-11e4-9088-62ece60caaf0",
  "supportId": "17PY1405953363409286-284546144",
  "httpStatus": 201,
  "detail": "Request accepted."
}
```

2. After **pingAfterSeconds** has elapsed, use the **purgeId** to **GET** from `/ccu/v2/purges/{purgeId}`:

**GET** `/ccu/v2/purges/57799d8b-10e4-11e4-9088-62ece60caaf0`

3. Check the **purgeStatus** for Done. Otherwise, repeat if In-Progress after waiting another **pingAfterSeconds**.

```
{
  "completionTime": "2014-07-21T14:42:18Z",
  "httpStatus": 200,
  "originalEstimatedSeconds": 420,
  "originalQueueLength": 0,
  "purgeId": "57799d8b-10e4-11e4-9088-62ece60caaf0",
  "purgeStatus": "Done",
  "submissionTime": "2014-07-21T14:39:30Z",
  "supportId": "17SY1405954814899441-292938848"
}
```

## Check the Purge Queue

Each URL or CP code you purge is an item in the queue, and there's a limit of 10,000. It's recommended to purge by CP code to keep the queue length small and manageable.

1. Make a **GET** request to `/ccu/v2/queues/default` or `/ccu/v2/queues/emergency`.
2. Check the current `queueLength`:

```
{
  "httpStatus" : 200,
  "queueLength" : 17,
  "detail" : "The queue may take a minute to reflect new or removed requests.",
  "supportId" : "17QY1321286863376510-220300384"
}
```