LAAWS Crawler Service API

An API for managing crawler tasks
More information: https://www.lockss.org/
Contact Info: lockss-support@lockss.org
Version: 1.0.0
BasePath:/
BSD-3-Clause
https://www.lockss.org/support/open-source-license/

Access

1. HTTP Basic Authentication

Methods

[Jump to Models]

Table of Contents

Default

- POST /crawls/
- DELETE /crawls/{crawlId}
- DELETE /crawls/
- GET /crawls/{crawlId}
- GET /crawlers/{crawler}
- GET /crawlers/
- GET /crawls/

Status

- GET /status

Default

POST /crawls/

Up

Request a crawl using a descriptor (**addCrawl**)

Use the information found in the descriptor object to initiate a crawl.

Consumes

This API call consumes the following media types via the Content-Type request header:

- `application/json`

Request body

crawRequest crawRequest **(required)**
Body Parameter —

Return type

crawRequest

Example data

Content-Type: application/json

```
{
  "repairList" : [ "repairList", "repairList" ],
  "auId" : "auId",
  "crawlKind" : "newContent",
  "crawler" : "lockss"
}
```

Produces
This API call produces the following media types according to the Accept request header; the media type will be conveyed by the Content-Type response header.

- `application/json`

Responses
202
The crawl request has been queued for operation. crawRequest
400
Bad Request
401
Unauthorized
403
Forbidden
404
Not Found
500
Internal Server Error

DELETE /crawls/{crawlId}

Up

Remove or stop a crawl (**deleteCrawlById**)

Delete a crawl given the crawl identifier, stopping any current processing, if necessary

Path parameters

crawlId (required)
Path Parameter — The identifier of the requested crawl

Return type
crawRequest

Example data
Content-Type: application/json

```
{
  "repairList" : [ "repairList", "repairList" ],
  "auId" : "auId",
  "crawlKind" : "newContent",
  "crawler" : "lockss"
}
```

Produces
This API call produces the following media types according to the Accept request header; the media type will be conveyed by the Content-Type response header.

- `application/json`

Responses
200
The deleted crawl crawRequest
401
Unauthorized
403
Forbidden
404
Not Found
500
Internal Server Error

DELETE /crawls/

Up

Delete all of the currently queued and active crawl requests (**deleteCrawls**)

Halt and delete all of the currently queued and active crawls

Query parameters

id (required)
Query Parameter — The crawl id

Produces
This API call produces the following media types according to the Accept request header; the media type will be conveyed by the Content-Type response header.

- `application/json`

Responses
200
All jobs successfully stopped and deleted
401
Unauthorized
403
Forbidden
500
Internal Server Error

GET /crawls/{crawlId}

Up

Get the crawl info for this job (**getCrawlById**)

Get the job represented by this crawl id

Path parameters

crawlId (required)
Path Parameter — The identifier of the requested crawl

Return type
status

Example data
Content-Type: application/json

```
{
  "msg" : "msg",
  "code" : 0
}
```

Produces
This API call produces the following media types according to the Accept request header; the media type will be conveyed by the Content-Type response header.

- `application/json`

Responses
200
The crawl status of the requested crawl status
401
Unauthorized
404
Not Found
500
Internal Server Error

GET /crawlers/{crawler}

Up

Get the status of the crawler (**getCrawlerStatus**)

Get the status of the crawler

Path parameters

crawler (required)
Path Parameter — The identifier of the requested crawl

Return type
crawlerStatus

Example data
Content-Type: application/json

```
{
  "ausWantCrawl" : 1,
  "isEnabled" : true,
  "failed" : 6,
  "ausCrawling" : [ "ausCrawling", "ausCrawling" ],
  "isStarted" : true,
  "ausEligibleCrawl" : 5,
  "successful" : 0
}
```

Produces
This API call produces the following media types according to the Accept request header; the media type will be conveyed by the Content-Type response header.

- `application/json`

Responses
200
The status of the crawler crawlerStatus
401

Unauthorized
500
Internal Server Error

GET /crawlers/

Up

Get the list of supported crawlers. (**getCrawlers**)

Return the list of supported crawlers.

Return type
inline_response_200

Example data
Content-Type: application/json

```
{
  "crawls" : [ "crawls", "crawls" ]
}
```

Produces
This API call produces the following media types according to the Accept request header; the media type will be conveyed by the Content-Type response header.

- `application/json`

Responses
200
The crawler list. inline_response_200

GET /crawls/

Up

Get a list of active crawls. (**getCrawls**)

Get a list of all currently active crawls or a pageful of the list defined by the continuation token and size

Query parameters

limit (optional)
Query Parameter — The number of jobs per page default: 50

continuationToken (optional)
Query Parameter — The continuation token of the next page of jobs to be returned.

Return type
crawlPageInfo

Example data
Content-Type: application/json

```
{
  "jobs" : [ {
    "auId" : "auId",
    "auName" : "auName",
    "sources" : {
      "urls" : [ "urls", "urls" ],
```

```
      "pageInfo" : {
        "curLink" : "curLink",
        "resultsPerPage" : 1,
        "totalCount" : 6,
        "continuationToken" : "continuationToken",
        "nextLink" : "nextLink"
      }
    },
    "pending" : {
      "urls" : [ "urls", "urls" ],
      "pageInfo" : {
        "curLink" : "curLink",
        "resultsPerPage" : 1,
        "totalCount" : 6,
        "continuationToken" : "continuationToken",
        "nextLink" : "nextLink"
      }
    },
    "bytesFetched" : 2,
    "type" : "type",
    "error" : {
      "pageInfo" : {
        "curLink" : "curLink",
        "resultsPerPage" : 1,
        "totalCount" : 6,
        "continuationToken" : "continuationToken",
        "nextLink" : "nextLink"
      },
      "errors" : [ {
        "severity" : "Warning",
        "url" : "url",
        "errorMsg" : "errorMsg"
      }, {
        "severity" : "Warning",
        "url" : "url",
        "errorMsg" : "errorMsg"
      } ]
    },
    "isActive" : true,
    "refetchDepth" : 9,
    "excluded" : {
      "urls" : [ "urls", "urls" ],
      "pageInfo" : {
        "curLink" : "curLink",
        "resultsPerPage" : 1,
        "totalCount" : 6,
        "continuationToken" : "continuationToken",
        "nextLink" : "nextLink"
      }
    },
    "mimeTypes" : {
      "counts" : 5,
      "mimeType" : "mimeType"
    },
    "isError" : true,
    "startTime" : "yyyy-MM-ddTHH:mm:ss.SSSZ",
    "key" : "key",
    "priority" : 5,
```

```
    "startUrls" : [ "startUrls", "startUrls" ],
    "proxy" : "proxy",
    "depth" : 7,
    "notModified" : {
      "urls" : [ "urls", "urls" ],
      "pageInfo" : {
        "curLink" : "curLink",
        "resultsPerPage" : 1,
        "totalCount" : 6,
        "continuationToken" : "continuationToken",
        "nextLink" : "nextLink"
      }
    },
    "isWaiting" : true,
    "parsed" : {
      "urls" : [ "urls", "urls" ],
      "pageInfo" : {
        "curLink" : "curLink",
        "resultsPerPage" : 1,
        "totalCount" : 6,
        "continuationToken" : "continuationToken",
        "nextLink" : "nextLink"
      }
    },
    "endTime" : "yyyy-MM-ddTHH:mm:ss.SSSZ",
    "errors" : {
      "urls" : [ "urls", "urls" ],
      "pageInfo" : {
        "curLink" : "curLink",
        "resultsPerPage" : 1,
        "totalCount" : 6,
        "continuationToken" : "continuationToken",
        "nextLink" : "nextLink"
      }
    },
    "status" : {
      "msg" : "msg",
      "code" : 0
    },
    "fetched" : {
      "urls" : [ "urls", "urls" ],
      "pageInfo" : {
        "curLink" : "curLink",
        "resultsPerPage" : 1,
        "totalCount" : 6,
        "continuationToken" : "continuationToken",
        "nextLink" : "nextLink"
      }
    }
  }, {
    "auId" : "auId",
    "auName" : "auName",
    "sources" : {
      "urls" : [ "urls", "urls" ],
      "pageInfo" : {
        "curLink" : "curLink",
        "resultsPerPage" : 1,
        "totalCount" : 6,
```

```
      "continuationToken" : "continuationToken",
      "nextLink" : "nextLink"
    }
  },
  "pending" : {
    "urls" : [ "urls", "urls" ],
    "pageInfo" : {
      "curLink" : "curLink",
      "resultsPerPage" : 1,
      "totalCount" : 6,
      "continuationToken" : "continuationToken",
      "nextLink" : "nextLink"
    }
  },
  "bytesFetched" : 2,
  "type" : "type",
  "error" : {
    "pageInfo" : {
      "curLink" : "curLink",
      "resultsPerPage" : 1,
      "totalCount" : 6,
      "continuationToken" : "continuationToken",
      "nextLink" : "nextLink"
    },
    "errors" : [ {
      "severity" : "Warning",
      "url" : "url",
      "errorMsg" : "errorMsg"
    }, {
      "severity" : "Warning",
      "url" : "url",
      "errorMsg" : "errorMsg"
    } ]
  },
  "isActive" : true,
  "refetchDepth" : 9,
  "excluded" : {
    "urls" : [ "urls", "urls" ],
    "pageInfo" : {
      "curLink" : "curLink",
      "resultsPerPage" : 1,
      "totalCount" : 6,
      "continuationToken" : "continuationToken",
      "nextLink" : "nextLink"
    }
  },
  "mimeTypes" : {
    "counts" : 5,
    "mimeType" : "mimeType"
  },
  "isError" : true,
  "startTime" : "yyyy-MM-ddTHH:mm:ss.SSSZ",
  "key" : "key",
  "priority" : 5,
  "startUrls" : [ "startUrls", "startUrls" ],
  "proxy" : "proxy",
  "depth" : 7,
  "notModified" : {
```

```
      "urls" : [ "urls", "urls" ],
      "pageInfo" : {
        "curLink" : "curLink",
        "resultsPerPage" : 1,
        "totalCount" : 6,
        "continuationToken" : "continuationToken",
        "nextLink" : "nextLink"
      }
    },
    "isWaiting" : true,
    "parsed" : {
      "urls" : [ "urls", "urls" ],
      "pageInfo" : {
        "curLink" : "curLink",
        "resultsPerPage" : 1,
        "totalCount" : 6,
        "continuationToken" : "continuationToken",
        "nextLink" : "nextLink"
      }
    },
    "endTime" : "yyyy-MM-ddTHH:mm:ss.SSSZ",
    "errors" : {
      "urls" : [ "urls", "urls" ],
      "pageInfo" : {
        "curLink" : "curLink",
        "resultsPerPage" : 1,
        "totalCount" : 6,
        "continuationToken" : "continuationToken",
        "nextLink" : "nextLink"
      }
    },
    "status" : {
      "msg" : "msg",
      "code" : 0
    },
    "fetched" : {
      "urls" : [ "urls", "urls" ],
      "pageInfo" : {
        "curLink" : "curLink",
        "resultsPerPage" : 1,
        "totalCount" : 6,
        "continuationToken" : "continuationToken",
        "nextLink" : "nextLink"
      }
    }
  } ],
  "pageInfo" : {
    "curLink" : "curLink",
    "resultsPerPage" : 1,
    "totalCount" : 6,
    "continuationToken" : "continuationToken",
    "nextLink" : "nextLink"
  }
}
```


Produces

This API call produces the following media types according to the Accept request header; the media type will be conveyed by the Content-Type response header.

- application/json

Responses
200
The requested crawls crawlPageInfo
400
Bad Request
401
Unauthorized
409
Conflict
500
Internal Server Error

Status

GET /status

Up

Get the status of the service (**getStatus**)

Get the status of the service

Return type
apiStatus

Example data
Content-Type: application/json

```
{
  "ready" : true,
  "version" : "version"
}
```

Produces
This API call produces the following media types according to the Accept request header; the media type will be conveyed by the Content-Type response header.

- application/json

Responses
200
The status of the service apiStatus
401
Unauthorized
500
Internal Server Error

Models

[Jump to Methods]

Table of Contents

1. apiStatus -
2. crawRequest -
3. crawlPageInfo -
4. crawlStatus -

5. crawlerStatus -
6. errorCounter -
7. inline_response_200 -
8. mimeCounter -
9. pageInfo -
10. status -
11. urlCounter -
12. urlError -

apiStatus - Up

The status information of the service

version
String The version of the service

ready
Boolean The indication of whether the service is available

crawlRequest - Up

A descriptor for a LOCKSS crawl.

auId
String The unique au id for this crawled unit.

crawlKind
String The kind of crawl being performed. For now this is either new content or repair.
Enum:
newContent
repair

crawler (optional)
String The crawler to use for this crawl, should be one of the crawlers returned by /crawlers

repairList (optional)
array[String] The repair urls in a repair crawl

crawlPageInfo - Up

A display page of jobs

jobs
array[crawlStatus] The jobs displayed in the page

pageInfo
pageInfo

crawlStatus - Up

key
String The id for the crawl.

auId
String The id for the au.

auName
String The name for the au.

type
String The type of crawl.

startUrls
array[String] An array of start urls

startTime
date The time the crawl began in ISO-8601 format: date

endTime
date The time the crawl ended in ISO-8601 format: date

status
status

error (optional)
errorCounter

isWaiting
Boolean True if the crawl wating to start.

isActive
Boolean True if the crawl is active.

isError
Boolean True if the crawl has errored.

priority
Integer The priority for this crawl. format: int32

fetched (optional)
urlCounter

excluded (optional)
urlCounter

notModified (optional)
urlCounter

parsed (optional)
urlCounter

sources (optional)
urlCounter

pending (optional)
urlCounter

errors (optional)
urlCounter

mimeTypes (optional)
mimeCounter

bytesFetched (optional)
Integer The number of bytes fetched. format: int62

depth (optional)
Integer The depth of the crawl. format: int32

refetchDepth (optional)
Integer The refetch depth of the crawl. format: int32

proxy (optional)
String The proxy used for crawling.

crawlerStatus - Up

successful

Integer The number of sucessful crawls format: int32

failed
Integer The number of failed crawls format: int32

ausCrawling
array[String] An array of aus currently being crawled

ausWantCrawl
Integer The number of aus that want crawls format: int32

ausEligibleCrawl
Integer The number of aus eligible for crawls format: int32

isEnabled
Boolean Is the crawl manager enabled

isStarted
Boolean Is the crawl starter running

errorCounter - Up

A list of urls with errors and their error information.

pageInfo
pageInfo

errors
array[urlError] An list of urls with error description

inline_response_200 - Up

A list of crawlers.

crawls (optional)
array[String] An array of crawler names

mimeCounter - Up

A counter for mimeTypes seen during a crawl.

mimeType
String The mime type to count.

counts
Integer number of urls of mimeType. format: int32

pageInfo - Up

The information related to pagination of content

totalCount
Integer The total number of elements to be paginated format: int32

resultsPerPage
Integer The number of results per page format: int32

continuationToken
String The continuation token

curLink
String The link to the current page

nextLink (optional)
String The link to the next page

status - Up

The existing state of a job

code
Integer The numeric value for the current state format: int32

msg
String A text message defining the current state

urlCounter - Up

pageInfo
pageInfo

urls
array[String] An list of urls

urlError - Up

A url and a description and severity of the error

url
String The url with the error.

errorMsg
String The text message describing the error.

severity
String The severity of the error
Enum:
Warning
Error
Fatal