



[Product Bulk File API Request](#)

[Product Bulk API Request](#)

[Product Media API Requests](#)

[Logging API Requests](#)

Product Bulk File API Request

Description:

For managing thousands of products in a single payload, SureDone provides an API endpoint for uploading CSV files, generating file exports and checking the status of both. Depending on the number and which channel integrations are implemented determines what the valid file headers are for a specific user bulk upload template. Take a look at our [comprehensive bulk header guide](#) to learn more about the bulk file format as this functionality is a basic requirement for enterprise online retailers. Note: system notifications are sent by default for bulk uploads and exports. To specify a different email address for the bulk results notification, pass in the variable "sd_bulk_email" with the notification email address.

CURL Request Example: (bulk file upload)

```
curl --request POST --url https://api.suredone.com/v1/bulk \
--header 'content-type: multipart/form-data' \
--header 'x-auth-integration: partnername' \
--header 'x-auth-user: testapixxx' \
--header 'x-auth-token:
4AFA32F39D6B7EC1F2589D7EF92CAE7D0A3A2B9846DA8E1C372C6CA0BF043669A227F
C5F5D415EB41USL7PKJB5LIJIPKUA VSYTM2YD7W0' \
--form 'bulk_file=@/vagrant/test-upload.csv'
```

CURL Response Example:

```
{
  "result": "success",
  "message": "Upload success",
  "request_file": "20161014-123705-1-bulk-requests.csv",
  "result_file": "20161014-123705-1-bulk-results.csv",
  "type": "items",
  "id": "guid",
  "time": "2016-10-14 12:37:05",
  "actions": 1,
  "email": "hello@suredone.com"
}
```

CURL Request Example: (bulk file result status)

```
curl --request GET --url
https://api.suredone.com/v1/bulk/results/20161014-121734-1-bulk-results.csv \
--header 'content-type: application/x-www-form-urlencoded' \
--header 'x-auth-integration: partnername' \
--header 'x-auth-user: testapixxx' \
--header 'x-auth-token:
4AFA32F39D6B7EC1F2589D7EF92CAE7D0A3A2B9846DA8E1C372C6CA0BF043669A227F
C5F5D415EB41USL7PKJB5LIJIPKUA VSYTM2YD7W0'
```

CURL Response Example:

```
{
  "result": "success",
  "message": "Bulk file completed successfully",
  "url":
"https://s3.amazonaws.com/assets.suredone.net/v1021/system/bulks/20161014-123705-1-bulk-results.csv?AWSAccessKeyId=AKIAJP7QQ5CECS7IZ5KA&Expires=1476449377&Signature=P9I5xt6Ar1bTA%2BsWa8Aklz5fQVQ%3D"
}
```

CURL Request Example: (bulk file export request)

```
curl --request POST --url https://api.suredone.com/v1/bulk/exports \  
--header 'content-type: application/x-www-form-urlencoded' \  
--header 'x-auth-integration: partnername' \  
--header 'x-auth-user: testapixxx' \  
--header 'x-auth-token:  
4AFA32F39D6B7EC1F2589D7EF92CAE7D0A3A2B9846DA8E1C372C6CA0BF043669A227F  
C5F5D415EB41USL7PKJB5LIJIPKUAVSYTM2YD7W0' \  
--data 'q=price:>999&type=items&mode=include&fields=guid,title,price'
```

CURL Response Example:

```
{  
  "result": "success",  
  "message": "Export success",  
  "export_file": "20161014-124811-1104-items-export.csv",  
  "type": "items",  
  "time": "2016-10-14 12:48:11",  
  "actions": 1104,  
  "email": "hello@suredone.com",  
  "params": "price:>999",  
  "mode": "include",  
  "fields": {  
    "0": "guid",  
    "1": "title",  
    "2": "price"  
  }  
}
```

CURL Request Example: (bulk export status)

```
curl --request GET --url  
https://api.suredone.com/v1/bulk/exports/20161014-121734-1-bulk-results.csv \  
--header 'content-type: application/x-www-form-urlencoded' \  
--header 'x-auth-integration: partnername' \  
--header 'x-auth-user: testapixxx' \  
--header 'x-auth-token:  
4AFA32F39D6B7EC1F2589D7EF92CAE7D0A3A2B9846DA8E1C372C6CA0BF043669A227F  
C5F5D415EB41USL7PKJB5LIJIPKUAVSYTM2YD7W0'
```

CURL Response Example:

```
{  
  "result": "success",  
  "message": "Export completed successfully",
```

```
"url":  
"https://s3.amazonaws.com/assets.suredone.net/1021/system/exports/20161014-124811-1  
104-items-export.csv?AWSAccessKeyId=AKIAJP7QQ5CECS7IZ5KA&Expires=1476450103&Si  
gnature=WJmqvD2kx3t29THAUsVaRvKgQo8%3D"  
}
```

Response Fields:

result: API response; return values will be either “success”, “failure”, or “error”
message: api success or error messages
type: editor type, will always be “items” for product requests
id: field used as product identifier
time: timestamp of api request
actions: number of actions requested
email: email address where processing notifications will be sent to
request_file: formatted file name of bulk upload request file
result_file: formatted file name of bulk upload result file
export_file: formatted file name of bulk export file
url: temporary link for a completed download of bulk file or export

Recommendation:

This api request is a necessity for the majority of enterprise online sellers who may have the needs to upload thousands of data updates per day. We recommend implementing this functionality if the demographic includes enterprise or growing online sellers.

Product Bulk API Request

Description:

The product editor api endpoint contains functionality for managing multiple products with a single upload. This functionality powers our production product listing and search interfaces available at <https://app.suredone.com/#!/products> These interfaces are built with AngularJS and may be inspected with web browser tools for discovering a working example of this functionality. This endpoint returns a response immediately and also follows many of the same paradigms described in our bulk upload guides <https://www.suredone.com/guides/introduction-to-bulk/> and <https://s3.amazonaws.com/assets.suredone.com/guides/media/bulk/SureDone-Bulk-Headers-Sorted.xlsx> . It is designed for handling hundreds to thousands of requests per payload, however it optimized for requests that include up to 50 products.

CURL Request Example:

```
curl --request POST --url https://api.suredone.com/v1/editor/items \  
--header 'content-type: application/x-www-form-urlencoded' \  

```

```
--header 'x-auth-integration: partnename' \  
--header 'x-auth-user: testapixxx' \  
--header 'x-auth-token:  
4AFA32F39D6B7EC1F2589D7EF92CAE7D0A3A2B9846DA8E1C372C6CA0BF043669A227F  
C5F5D415EB41USL7PKJJB5LIJIPKUAVSYTM2YD7W0' \  
--data 'requests=[[{"action=add","guid","stock","price","title"},["add","TEST001",1,9.99,"Product  
1"],["add","TEST002",3,19.99,"Product 2"],["add","TEST003",5,39.99]]'
```

CURL Response Example:

```
{  
  "1": {  
    "result": "success",  
    "errors": "",  
    "codes": "",  
    "htmlcodes": "",  
    "messages": "",  
    "notices": "",  
    "action": "add",  
    "identifier": "TEST001",  
    "guid": "TEST001"  
  },  
  "2": {  
    "result": "success",  
    "errors": "",  
    "codes": "",  
    "htmlcodes": "",  
    "messages": "",  
    "notices": "",  
    "action": "add",  
    "identifier": "TEST002",  
    "guid": "TEST002"  
  },  
  "3": {  
    "result": "failure",  
    "errors": "Validation Errors (1)",  
    "codes": "Errors (1): title:62",  
    "htmlcodes": "Errors (1): <a class=\"errnum\" id=\"sd-errnum-1\"  
onmouseover=\"sdShowMessage('Error Code 62','Title is a required field for adding  
items');\">title:62</a>",  
    "messages": "title:62 - Title is a required field for adding items",  
    "notices": "",  
    "action": "add",  
    "identifier": "TEST003",
```

```
        "guid": "TEST003"
    },
    "result": "success",
    "message": null,
    "type": "items",
    "id": "guid",
    "time": "2016-06-27 22:02:14",
    "actions": 3,
    "email": null,
    "url": null
}
```

Response Fields:

Note - the response returns a list of result objects correlating to each of the uploaded requests in the payload.

result: API response; return values will be either “success”, “failure”, or “error”

message: api success or error messages

type: editor type, will always be “items” for product requests

id: field used as product identifier

time: timestamp of api request

actions: number of actions requested

email: IGNORE.

url: IGNORE.

Recommendation:

This api request is best utilized anywhere managing multiple products with similar attributes is desired.

Product Media API Requests

Description:

There are a few workflows for managing images via the api. The first of these workflows includes posting image urls to the fields media1, media2, media3, media4, media5, media6, media7, media8, media9, media10 and mediax using the product editor api endpoint <https://api.suredone.com/v1/editor/items> The mediax field accepts a list of image urls delimited by a “*”. In this first workflow, SureDone will store and use the image urls posted as is to create marketplace listings. The second workflow is nearly identical to the first workflow except the api endpoint includes the query string “importmedia” added to the end. This query string with a value of “true” instructs the platform to download images to SureDone for image processing, resizing, and storage. The third workflow involves uploading image files directly to the api for

image processing, resizing, and storage. The api request for this workflow requires a value of “multipart/form-data” for the content-type header.

CURL Request Example: (first workflow of storing image urls)

```
curl --request POST --url https://api.suredone.com/v1/editor/items \  
--header 'content-type: application/x-www-form-urlencoded' \  
--header 'x-auth-integration: partnername' \  
--header 'x-auth-user: testapixxx' \  
--header 'x-auth-token:  
4AFA32F39D6B7EC1F2589D7EF92CAE7D0A3A2B9846DA8E1C372C6CA0BF043669A227F  
C5F5D415EB41USL7PKJB5LIJIPKUA VSYTM2YD7W0' \  
--data  
'action=edit&identifier=guid&guid=TEST001&media1=http://assets.suredone.com/v6/logos/sure  
done-icon-164x164.png'
```

CURL Request Example: (second workflow of importing images by downloading from urls)

```
curl --request POST --url https://api.suredone.com/v1/editor/items?importmedia=true \  
--header 'content-type: application/x-www-form-urlencoded' \  
--header 'x-auth-integration: partnername' \  
--header 'x-auth-user: testapixxx' \  
--header 'x-auth-token:  
4AFA32F39D6B7EC1F2589D7EF92CAE7D0A3A2B9846DA8E1C372C6CA0BF043669A227F  
C5F5D415EB41USL7PKJB5LIJIPKUA VSYTM2YD7W0' \  
--data  
'action=edit&identifier=guid&guid=TEST001&media1=http://assets.suredone.com/v6/logos/sure  
done-icon-164x164.png'
```

CURL Request Example: (third workflow of uploading image files directly to api)

```
curl --request POST --url https://api.suredone.com/v1/editor/items \  
--header 'content-type: multipart/form-data' \  
--header 'x-auth-integration: partnername' \  
--header 'x-auth-user: testapixxx' \  
--header 'x-auth-token:  
4AFA32F39D6B7EC1F2589D7EF92CAE7D0A3A2B9846DA8E1C372C6CA0BF043669A227F  
C5F5D415EB41USL7PKJB5LIJIPKUA VSYTM2YD7W0' \  
--form 'action=edit' \  
--form 'identifier=guid' \  
--form 'guid=TEST001' \  
--form 'price=19.99' \  
--form 'media1=@/vagrant/sd-icon-testfix.png'
```

CURL Response Example: (all three workflows)

```

{
  "1": {
    "result": "success",
    "errors": "",
    "codes": "",
    "htmlcodes": "",
    "messages": "",
    "notices": "",
    "action": "edit",
    "identifier": "TEST001",
    "guid": "TEST001"
  },
  "result": "success",
  "message": "",
  "type": "items",
  "id": "guid",
  "time": "2016-06-28 01:01:13",
  "actions": 1,
  "email": null,
  "url": null
}

```

Recommendation:

With multiple workflows available for managing images via the SureDone API, a choice can be made based on business requirements. If there is media management already available through the platform utilizing the SureDone API, then the first workflow is recommended as it has the best performance and is the most efficient.

Logging API Requests

Description: The logging api endpoint exposes SureDone's sophisticated infrastructure for logging events related to products, orders, storefront, settings, and more. Data from this request can provide granular information for the state of each element going to the platform as well as the actual API calls the platform sends to the different marketplaces with varying degrees of detail. Each of the following parameters may be sent delimited by a "," to filter multiple values:

- context: name of channel such as "ebay" or "amazon".
- instance: instance id of the context which defaults to "0" and may be number 2 or greater.
- identifier: identifier of the element such as the guid for products and order number for orders.
- channel_id: channel specific identifier such as ebayid for eBay and amznsku for Amazon.

action: SureDone action which may be one of “add”, “edit”, “relist”, “end”, “delete”, “start”, “sold” and “skip” for products. Other actions include “import”, “export”, “acknowledge” and “tracking”.

operation: channel specific operation such as “ReviseFixedPriceItem” for eBay.

method: mode by which log was created and may be one of “api”, “ui”, “bulk”, “auto” and “sync”.

type: type of log record to return which may be one of “item” for general products, “inventory”, “price”, “image”, “variation” and “shipping” for feed specific logs related to Amazon, “order”, “setting”, “category” and “page” for orders, settings, and storefront types respectively.

integration: source log entry such as “ebay”, “amazon”, or “partnername”.

result: log result which may be one of “success”, “error”, “failure” and “warning” for general logs, and “queued”, “pending”, “submitted”, and “complete” for the queue states of Amazon.

job_id: identifier of job queue or other job type.

log_level: level of detail returned by the same log set and may be one of “1”, “3”, or “5”.

id: log record entry identifier.

timestamp: accepts start and end time requests of log entries and must sent in as two parameters “timestamp_start” and “timestamp_end” with date format Y-m-d H:i:s.

sort: order of logs returned which defaults to “timestamp desc” and accepts any field above.

start: starting index to pull logs from and accepts an integer value.

records: number of logs to return which defaults to “20” and accepts an integer from “1” to “100”.

CURL Request Example:

```
curl --request POST --url https://api.suredone.com/v1/logs \
--header 'content-type: application/x-www-form-urlencoded' \
--header 'x-auth-integration: partnername' \
--header 'x-auth-user: testapixxx' \
--header 'x-auth-token:
4AFA32F39D6B7EC1F2589D7EF92CAE7D0A3A2B9846DA8E1C372C6CA0BF043669A227F
C5F5D415EB41USL7PKJB5LIJIPKUA VSYTM2YD7W0' \
--data 'context=amazon&timestamp_start=2016-08-20 00:00:01&timestamp_end=2016-08-22
00:00:01&log_level=3&records=10'
```

CURL Response Example:

```
{
  "result": "success",
  "found": 2295726,
  "sort": "timestamp desc",
  "start": 0,
  "records": "10",
  "logs": {
    "0": {
      "id": "57ba3d9ab3149971218b8ddc",
      "sd_user_id": XXXX,
      "log_level": 3,
      "message": "{\"message\": \"successfully submitted inventory\"}"
    }
  }
}
```

```
"timestamp": "2016-08-21T23:47:35Z",
"context": "amazon",
"result": "submitted",
"operation": "Update",
"action": "edit",
"type": "inventory",
"identifier": "CTP6799",
"job_id": "243427017034",
"sd_acct_id": -1,
"integration": "",
"method": "sync",
"instance": 0,
"channel_id": "CTP6799",
"field_data": "",
"_version_": 1543318570284351490
},
"1": {
  "id": "57ba3d9ab3149971218b8ddd",
  "sd_user_id": XXXX,
  "log_level": 3,
  "message": "{\"message\": \"successfully submitted inventory\"}",
  "timestamp": "2016-08-21T23:47:35Z",
  "context": "amazon",
  "result": "submitted",
  "operation": "Update",
  "action": "edit",
  "type": "inventory",
  "identifier": "ZZZ00CTP6798",
  "job_id": "243427017034",
  "sd_acct_id": -1,
  "integration": "",
  "method": "sync",
  "instance": 0,
  "channel_id": "ZZZ00CTP6798",
  "field_data": "",
  "_version_": 1543318570285400066
},
"2": {
  "id": "57ba3d9ab3149971218b8de3",
  "sd_user_id": XXXX,
  "log_level": 3,
  "message": "{\"message\": \"successfully submitted inventory\"}",
  "timestamp": "2016-08-21T23:47:35Z",
```

```
"context": "amazon",
"result": "submitted",
"operation": "Update",
"action": "edit",
"type": "inventory",
"identifier": "CTP6787",
"job_id": "243427017034",
"sd_acct_id": -1,
"integration": "",
"method": "sync",
"instance": 0,
"channel_id": "CTP6787",
"field_data": "",
"_version_": 1543318570290642944
},
"3": {
  "id": "57ba3d9ab3149971218b8de7",
  "sd_user_id": XXXX,
  "log_level": 3,
  "message": "{\"message\": \"successfully submitted inventory\"}",
  "timestamp": "2016-08-21T23:47:35Z",
  "context": "amazon",
  "result": "submitted",
  "operation": "Update",
  "action": "edit",
  "type": "inventory",
  "identifier": "CTP6747",
  "job_id": "243427017034",
  "sd_acct_id": -1,
  "integration": "",
  "method": "sync",
  "instance": 0,
  "channel_id": "CTP6747",
  "field_data": "",
  "_version_": 1543318570290642947
},
"4": {
  "id": "57ba3d9ab3149971218b8dec",
  "sd_user_id": XXXX,
  "log_level": 3,
  "message": "{\"message\": \"successfully submitted inventory\"}",
  "timestamp": "2016-08-21T23:47:35Z",
  "context": "amazon",
```

```
    "result": "submitted",
    "operation": "Update",
    "action": "edit",
    "type": "inventory",
    "identifier": "CTP6699",
    "job_id": "243427017034",
    "sd_acct_id": -1,
    "integration": "",
    "method": "sync",
    "instance": 0,
    "channel_id": "CTP6699",
    "field_data": "",
    "_version_": 1543318570291691522
  },
  "5": {
    "id": "57ba3d9ab3149971218b8dfb",
    "sd_user_id": XXXX,
    "log_level": 3,
    "message": "{\"message\": \"successfully submitted inventory\"}",
    "timestamp": "2016-08-21T23:47:35Z",
    "context": "amazon",
    "result": "submitted",
    "operation": "Update",
    "action": "edit",
    "type": "inventory",
    "identifier": "CTP6501",
    "job_id": "243427017034",
    "sd_acct_id": -1,
    "integration": "",
    "method": "sync",
    "instance": 0,
    "channel_id": "CTP6501",
    "field_data": "",
    "_version_": 1543318570295885826
  },
  "6": {
    "id": "57ba3d9ab3149971218b8e07",
    "sd_user_id": XXXX,
    "log_level": 3,
    "message": "{\"message\": \"successfully submitted inventory\"}",
    "timestamp": "2016-08-21T23:47:35Z",
    "context": "amazon",
    "result": "submitted",
```

```
    "operation": "Update",
    "action": "edit",
    "type": "inventory",
    "identifier": "CTP6010",
    "job_id": "243427017034",
    "sd_acct_id": -1,
    "integration": "",
    "method": "sync",
    "instance": 0,
    "channel_id": "CTP6010",
    "field_data": "",
    "_version_": 1543318570302177282
  },
  "7": {
    "id": "57ba3d9ab3149971218b8e0a",
    "sd_user_id": XXXX,
    "log_level": 3,
    "message": "{\"message\": \"successfully submitted inventory\"}",
    "timestamp": "2016-08-21T23:47:35Z",
    "context": "amazon",
    "result": "submitted",
    "operation": "Update",
    "action": "edit",
    "type": "inventory",
    "identifier": "CTP6002",
    "job_id": "243427017034",
    "sd_acct_id": -1,
    "integration": "",
    "method": "sync",
    "instance": 0,
    "channel_id": "CTP6002",
    "field_data": "",
    "_version_": 1543318570304274434
  },
  "8": {
    "id": "57ba3d9ab3149971218b8e0b",
    "sd_user_id": XXXX,
    "log_level": 3,
    "message": "{\"message\": \"successfully submitted inventory\"}",
    "timestamp": "2016-08-21T23:47:35Z",
    "context": "amazon",
    "result": "submitted",
    "operation": "Update",
```

```

        "action": "edit",
        "type": "inventory",
        "identifier": "CTP5998",
        "job_id": "243427017034",
        "sd_acct_id": -1,
        "integration": "",
        "method": "sync",
        "instance": 0,
        "channel_id": "CTP5998",
        "field_data": "",
        "_version_": 1543318570305323010
    },
    "9": {
        "id": "57ba3d9ab3149971218b8e0e",
        "sd_user_id": XXXX,
        "log_level": 3,
        "message": "{\"message\": \"successfully submitted inventory\"}",
        "timestamp": "2016-08-21T23:47:35Z",
        "context": "amazon",
        "result": "submitted",
        "operation": "Update",
        "action": "edit",
        "type": "inventory",
        "identifier": "CTP5968",
        "job_id": "243427017034",
        "sd_acct_id": -1,
        "integration": "",
        "method": "sync",
        "instance": 0,
        "channel_id": "CTP5968",
        "field_data": "",
        "_version_": 1543318570307420162
    }
}
}
}

```

Response Fields:

result: api response; return values will be either "success" or "error".

found: number of logs available with the search criteria. Currently, logs are held for 20 days.

sort: field and direction returned log records are ordered by.

start: numerical index of the first record among the total "found" records.

records: number of log records returned.

logs: log entries.

logs > id: log entry identifier.
logs > sd_user_id: account user ID.
logs > log_level: log level of log entry.
logs > message: log message which may include raw api requests.
logs > timestamp: timestamp of log entry.
logs > context: channel or marketplace of log entry.
logs > result: log result.
logs > operation: channel or marketplace specific function or api call name.
logs > action: SureDone action.
logs > type: log entry type.
logs > identifier: identifier or product or order.
logs > job_id: job identifier.
logs > sd_acct_id: ignore, sub account ID of the SureDone user interface.
logs > integration: source integration generating log entry.
logs > method: mode of request generating log entry.
logs > instance: instance ID of channel integration.
logs > channel_id: channel or marketplace specific identifier.
logs > field_data: field data send in request.
logs > _version_: ignore.

Recommendation: Utilize the Logging API endpoint to discover issues with product and order requests and to understand the state of products awaiting status for integrations based on queue systems such as Amazon.