

eWarp 2.0 Specification

Table of Contents

| | |
|---|-----------|
| <i>eWarp 2.0 Specification</i> | 1 |
| <i>Introduction</i> | 2 |
| API updates disclaimer | 2 |
| <i>Endpoints</i> | 3 |
| JSON-RPC 2.0 | 3 |
| Example of JSON-RPC 2.0 request..... | 3 |
| Example of JSON-RPC 2.0 response..... | 3 |
| <i>Error handling</i> | 4 |
| Example of error response | 4 |
| Common error codes for the eWarp 2.0 methods | 4 |
| <i>Authentication</i> | 5 |
| <i>Formats</i> | 5 |
| Currency | 5 |
| Country..... | 5 |
| Dates | 6 |
| <i>eStinger (Notifications for Step 3 – Notify on materials added & Step 5 – Notify on delivery ready)</i> | 6 |
| Description | 6 |
| <i>eStinger HTTP callback</i> | 6 |
| Redirects support..... | 7 |
| Template URL formatting..... | 7 |
| Notify on materials added success | 9 |
| Notify on materials adding error | 9 |
| Request | 11 |
| Notification guarantee | 11 |
| <i>STEP 1 - eWarp 2.0 method: createOrder</i> | 12 |
| Description | 12 |
| Preconditions..... | 12 |
| Input Parameters | 12 |
| Output Parameters | 12 |
| JSON example | 13 |
| <i>STEP 2 - eWarp 2.0 method: addMaterialsForBatch</i> | 14 |
| Description | 14 |
| Preconditions..... | 15 |
| Input Parameters | 15 |
| Validations | 16 |
| Output Parameters | 16 |

| | |
|---|-----------|
| Related error codes..... | 17 |
| JSON example..... | 17 |
| STEP 4 - eWarp 2.0 method: batchReadyForProduction | 18 |
| Description | 18 |
| Preconditions..... | 19 |
| Input Parameters | 19 |
| Output Parameters | 19 |
| JSON example..... | 19 |
| STEP 6 - eWarp 2.0 method: getAssets | 20 |
| Description | 20 |
| ListOrder..... | 20 |
| Archival | 20 |
| Preconditions..... | 20 |
| Input Parameters | 20 |
| Output Parameters | 20 |
| JSON example..... | 22 |
| Asset types | 23 |
| <i>Note: in following section, we'll only describe Picture & Video asset types.....</i> | 23 |
| Picture | 23 |
| Video | 25 |
| Object types | 28 |
| Assets | 28 |
| EmbedAssetType..... | 29 |
| Profiles | 29 |
| Enumeration types..... | 30 |

Introduction

This document describes the eWarp 2.0 API that can be used by our clients to integrate with our systems.

The eWarp communication allows our clients and partners to access reference details and assets as well as perform operations like ordering and uploading of raw materials.

The intended audience for this document are developers and architects designing integrations with esoft systems.

API updates disclaimer

IMPORTANT

Please note that new fields might be added to the returned data structures over time, so systems integrating with eWarp 2.0 should ignore fields they do not know or need.

e.g. if originally we return

```
image :{
  width: 100,
  height: 100
}
```

we might modify it over time to return more information:

```
image :{
  width: 100,
  height: 100,
  dpi: 72,
  quality: 100
}
```

Systems integrating with eWarp 2.0 must be made so that the above change will not affect their functionality.

Endpoints

There are three available endpoints for development, testing and production.

Please note that data are not synchronised back from production to test and development so different dataset will be available on the three environments.

| Environment | Endpoint |
|-------------|---|
| Development | https://dev-m2.esoftsystems.com/ewarp/ |
| Test | https://test-m2.esoftsystems.com/ewarp/ |
| Production | https://m2.esoftsystems.com/ewarp/ |

JSON-RPC 2.0

JSON-RPC version 2.0 is used for the communication, see [json rpc 2.0 specification](#) for more details.

Example of JSON-RPC 2.0 request

```
{
  "id" : 694664211,
  "method" : "methodNameToCall",
  "jsonrpc" : "2.0",
  "params" : {
    ...
  }
}
```

Example of JSON-RPC 2.0 response

Please note: the **result** can be a string, an object, an array of objects depending on the method called.

Specific method documentation describes what value or objects is returned as result for that method.

```
{
```

```

    "id" : 17475107,
    "jsonrpc" : "2.0",
    "result" : {...}
}

```

Error handling

eWarp 2.0 uses the standard [json rpc 2.0 error object](#) for sending information on errors back. All methods will return a meaningful message and error code back. The data part is omitted.

Example of error response

```

{
  "jsonrpc": "2.0",
  "id": 57733179,
  "error": {
    "code": -32600,
    "message": "Sellers cannot be null."
  }
}

```

Common error codes for the eWarp 2.0 methods

| ERROR | Code | DESCRIPTION |
|--------------------------------|--------|---|
| Parse error | -32700 | JSON parse error |
| Invalid request | -32600 | Invalid JSON-RPC 2.0 Request |
| Method not found | -32601 | eWarp 2.0 method not found |
| Invalid parameters | -32602 | Invalid parameters for the method |
| Internal error | -32603 | Unhandled internal error |
| Unauthorized | -32001 | specified eWarp 2.0 user name and password not allowed, or EWS http header not valid |
| Forbidden | -32003 | specified eWarp 2.0 user does not have access to the method or the client/reference/batch |
| Not found | -32004 | part of the specified input could not be found |
| Not allowed | -32005 | HTTP post was not used |
| Unsupported media type | -32015 | HTTP content-type and accept must include application/json |
| External communication problem | -32018 | eWarp 2.0 had internal problems communicating with another system |
| External error | -32019 | eWarp 2.0 encountered an error return from another system |
| Maintenance mode | -32800 | Part of our system is currently down for maintenance. Please try again later. |

Authentication

HTTPS is used as the transport layer (TCP - transport protocol) and we will be using the HTTP Authorization request-header field with all the requests.

The Authorization field value is calculated on the base of your eWarp user name and password.

If you do not have an eWarp user name and password, please contact esoft systems at it@esoftsystems.com asking for an login by specifying "eWarp user name and password" request in the subject field.

An example of a typical header with authentication is shown here:

```
POST /ewarp/ HTTP/1.1
Host: dev.m2.esoftsystems.com
Authorization: EWS username:dGg8sDjxAjklsvzdA==
Accept: application/json
Content-Type: application/json
```

The Authorization header has EWS as the realm and the credentials are username:password where password is a base64 encoded version of the password you have received.

Frequently Seen Problems

This is a list of the most frequent issues we have seen when (eWarp) users are implementing HTTPS for eStinger receivers and material upload (the order does *not* imply frequency) :

1. The certificate has been issued to another domain than the one using it. Please check whether any of the following can be an issue:
 1. If a certificate has been issued to a specific host, then it must be used there. If the certificate has been issued to 'myhost.somedomain.com', then it must be used on 'myhost'
 2. If a certificate has been issued to a wildcard domain, ensure that the server using it is actually a part of that wildcard. If the certificate has been issued to '*. somedomain.com', then it can be used on 'myhost.somedomain.com', but 'mysub.myhost.somedomain.com' is not a part of that domain.
2. The issuer of the certificate is not a trusted provider. If the certificate is self-issued, then this *will* be a problem.
3. The certificate is no longer valid

Please check against this list if you are experiencing issues with implementing HTTPS.

Formats

Currency

Currency must follow ISO 4217 like "EUR" for Euro, "DKK" for danish kroner and "VND" for Vietnamese dong.

Country

Country names must follow ISO 3166-1 alpha-2 code which are two-letter names of countries, like "ES" for Spain and "VN" for Vietnam.

Dates

Dates are given as an integral number of seconds that have elapsed since 00:00:00 [Coordinated Universal Time](#) (UTC), Thursday, 1 January 1970.

See https://en.wikipedia.org/wiki/Unix_time for more details.

Please note that some programming libraries use milliseconds when converting time to number by default (e.g. default Java Date) and you might need to convert value in milliseconds to seconds whenever you send dates to eWarp 2.0.

eStinger (Notifications for Step 3 – Notify on materials added & Step 5 – Notify on delivery ready)

Description

eStinger is a notification service that informs client systems about changes on references.

This happens whenever data related to a reference has been updated. It is then up to the receiver of the call to investigate what needs to be updated and how.

Typically eStinger is used for notifying about updated contents to websites and external systems that uses the references and produced assets once they are ready or changed.

Types of Notifications include:

- DELIVERY - when we deliver assets (content) for the first time or due to a correction being done (after receiving and storing callback information [eWarp 2.0 method: getAssets](#) to refresh asset data)
- MATERIALS_ADDED - when materials have been added, and copied to our S3 buckets (to be processed) (*applies only for callbacks related to [eWarp 2.0 eStinger](#)*)
- MATERIALS_ADDING_ERROR - when materials have not been added because of an error (*applies only for callbacks related to [eWarp 2.0 eStinger](#)*)

eWarp 2.0 sends notification via HTTP callback: eStinger sends a simple GET request to a specified external URL.

The URL to which the callback is sent must respond quickly with a 200 HTTP Status.

eStinger HTTP callback

eStinger HTTP callback is a simple GET request to a specified external URL.

The call cannot take more than 10 seconds, after that time we'll consider the call as timed out.

eStinger do not support header based authentication methods.

Redirects support

eStinger callbacks provides limited support for up to 10 redirects, all of which cannot take more than 10 seconds in total to follow and execute.

eStinger will only follow the redirect using Location header returned by the client system: no parameters nor headers will be added or modified.

The client system must attach all relevant parameters from the original URI, and potentially rewrite them if needed to match the system it is redirecting to: e.g. if a redirect occurs due to a version change and parameter names have changed.

eStinger do not support header based authentication methods, so if any token or credentials are to be used they must be part of the parameter list specified by client system in the Location header returned during redirect.

Template URL formatting

The url used by eStinger is configured inside esoft systems. It is possible for the receiver to construct a url template that can take up to 5 parameters that will be replaced with appropriate values depending on the callback type.

| Parameter | Description | Example values |
|-----------|---|---------------------------------|
| clientId | The AX ID of the client/customer/company that the order belongs to. | 10033224 |
| reference | The reference number from the order. Reference usually is the same as case number when talking real estate orders. | HF345 |
| batchIds | <p>Comma separated list of batchIds. The batch ID can be linked to a specific production batch (needed if only parts of the order should be delivered).</p> <p>The batchIds param cannot be part of the URL path, it can only be included in the query string.</p> <p>Please note that the batchIds will only be sent for the following types of callbacks:</p> <ul style="list-style-type: none">• DELIVERY• MATERIALS_ADDED• MATERIALS_ADDING_ERROR | <p>98765,98766</p> <p>10002</p> |

| | | |
|------|--|--|
| type | <p>The type of the callback.</p> <p>This value will be always added to the callback even if it is not specified in the url template.</p> <p>e.g. &type=DELIVERY</p> | <p>DELIVERY</p> <p>MATERIALS_ADDED</p> <p>MATERIALS_ADDING_ERROR</p> |
|------|--|--|

Template examples:

- <http://www.example.org/updateCase.asp?clientId={clientId}&reference={reference}&batchIds={batchIds}>
- <http://www.example.org/updateCase.asp?clientId={clientId}&reference={reference}&batchIds={batchIds}&type={type}&error={error}>
- <http://www.example.org/updateCase.asp?c={clientId}&r={reference}&batchId={batchIds}&t={type}&e={error}>
- <http://www.example.org/updateCase.asp?companyId={clientId}&reference={reference}&batchId={batchIds}>
- <http://www.example.org/updateCase.asp/{clientId}/reference/{reference}?batchIds={batchIds}>
- <http://www.example.org/updateCase.asp/{type}/client/{clientId}/reference/{reference}?batchIds={batchIds}>

Callback examples

- <http://www.example.org/updateCase1.asp?clientId=123456&reference=HF345&batchIds=98766,98765&type=DELIVERY>
- http://www.example.org/updateCase1.asp?clientId=123456&reference=HF345&batchIds=98766&type=MATERIALS_ADDED
- http://www.example.org/updateCase1.asp?clientId=123456&reference=HF345&batchIds=98766&type=MATERIALS_ADDING_ERROR&error=001-Material%20was%20not%20found%20in%20the%20specified%20http%20location

Notify on materials added success

Once materials are copied by eWarp to its private store, eWarp will try to notify about this. The notification url is the one specified as the callbackUrl in the request [eWarp 2.0 method: addMaterialsForBatch](#).

On success MATERIALS_ADDED type will be sent without any error messages.

e.g. http://www.customer.com/materialsAdded/callback?clientId=123456&batchId=50000001234&type=MATERIALS_ADDED

Notify on materials adding error

On materials download failure (If errors happen in the copy to eWarp private storage), MATERIALS_ADDING_ERROR will be sent with error, errorCode and errorLogUrl.

e.g. http://www.customer.com/materialsAdded/callback?clientId=50100123&batchId=50000001234&type=MATERIALS_ADDING_ERROR&error=Resrouce%20Not%20Found&errorCode=-32604&errorLogUrl=https://cdn-m2.esoftsystems.com/error/50100123/50000001234/ssHHmm-ddMMYY.log

Note that

- The parameters error, errorCode and errorLogUrl will only be sent for the type of callback = MATERIALS_ADDING_ERROR
- These parameters and their corresponding values will be always added to the callback automatically.

| Parameter | Description | Example values |
|-----------|--|---------------------|
| error | <p>Error message describing what went wrong when adding materials. This parameter is formed as: error=error message summary.</p> <p>e.g. &error=Material%20was%20not%20found%20in%20the%20specified%20http%20location </p> | Resource Not Found. |

| | | |
|-------------|---|--|
| errorCode | <p>A technical code to show what was wrong with the action. Also see the section "Possible error codes on materials adding error".</p> <p>e.g. &errorCode=-32004</p> | -32604 |
| errorLogUrl | <p>The downloadable error log, this resource contain more details about the occurred error.</p> <p>e.g. &errorLogUrl=https://cdn-m2.esoftsystems.com/error/50100123/50000001234/HHmmss-ddMMYYYY.log</p> | <p>https://cdn-m2.esoftsystems.com/error/50100123/50000001234/HHmmss-ddMMYYYY.log</p> |

Possible error codes on materials adding error

| Error Code | Possible Error Message | Cause | Action |
|------------|---|-----------------------|---|
| -32602 | Resource URI malformed. Specified file size is different than the content length on the src location. Could not use HEAD to validate expected file size - got response {code} for url {url} | Customer Fault | Customer to fix materials and then call eWarp 2.0 to add materials for batch again. |
| -32603 | | Esoft - Unknown error | Customer to call either esoft customer service or IT administrator for help |
| -32004 | Resource Not Found. | Customer Fault | Customer to fix materials and then call eWarp 2.0 to add materials for batch again. |

Request

The request will be made to the URL(s) specified in the eWarp usage agreement.

If you want to add a new URL write an email to our customer service with the needed details and have the usage agreement updated.

When you receive the request, answer back with the correct response code right away (HTTP 200 OK) as the call will have a short time out (2-3s).

If you need to get updates from other services after receiving the eStinger request you must do so in the background (use queues or the like).

Do not attempt to get data from eWarp synchronously with an eStinger call!

Notification guarantee

eStinger expects to get back a HTTP 200 OK response for callback.

If eStinger does not get a successful response: it tries again up to 8 times (~ 9 hours) in various time intervals.

If eStinger does not get a successful response within those 8 tries, it will stop notification calls on that specific update.

eStinger would stop retrying immediately if it gets back HTTP 4xx from callbacks

STEP 1 - eWarp 2.0 method: createOrder

Description

This operation must be called to create an order. If the new order is created successful, the response will return the order id and the ordered order lines identifiers and the batch numbers to use for future reference.

Please note: If some of the order lines needs to have materials before processing can start, these must be added using the [eWarp 2.0 method: addMaterialsForBatch](#)

Also note that the production of any order line (also those without the need for materials) is not started before the [eWarp 2.0 method: batchReadyForProduction](#) is called.

Preconditions

eWarp user must have access to create orders for specified **clientId**.

Input Parameters

| Path | Type | Description | Required |
|-------------------------------|--------|---|----------|
| params.clientId | String | The client id | yes |
| params.reference | String | The reference id: max 128 characters | yes |
| params.receivingCompany | String | The esoft company id | yes |
| params.orderLines | Array | Array of order lines | yes |
| params.orderLines[].productId | Number | The product id Please contact esoft to get the possible values that can be used for each client | yes |
| params.orderLines[].quantity | Number | The quantity of products to be ordered | yes |
| params.orderLines[].unit | String | The product unit Please contact esoft to get the possible values that can be used for each client | yes |
| params.orderLines[].variant1 | String | The product variant Please contact esoft to get the possible values that can be used for each client | yes |
| params.orderLines[].comments | Array | Array of comments: string values | no |

Output Parameters

| Path | Type | Description |
|--------------------------------------|--------------------|--|
| result.clientId | String | The client id |
| result.reference | String | The reference id |
| result.receivingCompany | String | The esoft company id |
| result.placedDate | Number (Unix time) | The date of order creation |
| result.deliveredDate | Number (Unix time) | The date of order delivery |
| result.orderLines | Array | Array of order lines |
| result.orderLines[].productId | Number | The product id |
| result.orderLines[].quantity | Number | The quantity |
| result.orderLines[].unit | String | The product unit |
| result.orderLines[].variant1 | String | The product variant |
| result.orderLines[].variant2 | String | The product sub variant |
| result.orderLines[].batchId | Number | The batchId related to the created order line |
| result.orderLines[].comments | Array | Array of comments: string values |
| result.bookingAddress | Object | The booking address |
| result.bookingAddress.street | String | The booking address street |
| result.bookingAddress.buildingNumber | String | The booking address building number |
| result.bookingAddress.buildingFloor | String | The booking address building floor |
| result.bookingAddress.cityArea | String | The booking address city area (district) |
| result.bookingAddress.town | String | The booking address town (city) |
| result.bookingAddress.postalCode | String | The booking address postal code |
| result.bookingAddress.country | String | The booking address country (currently can only be DK or SE) |

JSON example

This is the a simple sample where there are just mandatory fields which need to be filled in and this could be used for the integration with the customer who has no need of booking address and order information in every single order.

```
{
  "id" : 194403908,
  "method" : "createOrder",
  "jsonrpc" : "2.0",
  "params" : {
    "clientId" : "clientId",
```

```

    "reference" : "referenceNumber",
    "receivingCompany" : "esvn",
    "orderLines" : [ {
        "productId" : 12345678,
        "quantity" : 1,
        "unit" : "Pcs",
        "variant1" : "Basic"
    } ]
}
}

```

Response with the requested info

```

{
  "id" : 194403908,
  "jsonrpc" : "2.0",
  "result" : {
    "clientId" : "clientId",
    "reference" : "referenceNumber",
    "receivingCompany" : "esvn",
    "placedDate" : 1372765119366,
    "orderLines" : [{
      "productId" : 12345678,
      "quantity" : 1,
      "unit" : "Pcs",
      "variant1" : "Basic"
      "batchId" : 12345678901
    } ]
  }
}

```

Error Response

```

{
  "jsonrpc": "2.0",
  "id": 194403908,
  "error": {
    "code": -32018,
    "message": "AX_ERR: Warning: The value '10101234' in field 'Customer account' is
not found in the related table 'Customers'."
  }
}

```

STEP 2 - eWarp 2.0 method: addMaterialsForBatch

Description

Most order lines for an order requires editing of materials provided by the client. In such cases once the order is successfully created, information of the materials to be edited for each order line needs to be added. There are two different ways of sending information about materials, which to use depends on where the materials are currently stored.

The actual copying of materials will happen asynchronously when resources in eWarp are available. This is usually done within a few minutes, but can take longer if the files to copy are big.

This method queues the materials for copying.

When eWarp has copied all files to its private storage, the specified callback url of the request will be used for notification. See [eWarp 2.0 eStinger: MATERIALS_ADDED](#) for more information.

If an error occurred during copy and our system cannot complete copying of the files (e.g. file was removed or is not accessible) then callbackUrl will be called with error message. See [eWarp 2.0 eStinger: MATERIALS_ADDING_ERROR](#) for more information.

1) If materials are already stored in Amazon S3, then the best solution is to send information using the bucket, the keys and the md5s of the files in the array of materials. If needed the bucket used can be protected. Then the requirements are that an eWarp AWS user have getObject access (using a bucket policy) and that the objects are available for copying when eWarp is ready to do so (read, do not delete the files until the materials added callback have been received).

2) If materials are located on servers and public accessible via http information about the files urls and sizes should be sent using the array of materialsHttp should be used. Files must not be deleted until the materials added callback have been received.

If materials already have been copied for a batch number, then invoking this method will **add** the specified files to the batch. If [eWarp 2.0 method: addMaterialsForBatch](#) have been called for this batch, then calling add materials will return an error.

Preconditions

eWarp user must have access to specified **clientId** resources.

Input Parameters

| Path | Type | Description | Required |
|---------------------------|--------|--|--|
| params.clientId | String | The client id | Yes |
| params.reference | String | Client's reference number for the order | Yes |
| params.callbackUrl | String | An http url to a server capable of receiving a http GET - used for notification of when files are copied to eWarps storage | Yes |
| params.batchId | Number | The batch id these files belongs to | Yes |
| params.materials | Array | A list of materials for this batch to be copied from amazon s3 | Either materials, materialsHttp or both must be present and have content |
| params.materials[].bucket | String | The bucket the object is located in | Yes for every specified material |

| | | | |
|-----------------------------|--------|---|--|
| params.materials[].key | String | The key to the object within the bucket | Yes for every specified material |
| params.materials[].md5 | String | The md5 sum of the objects content: hex-encoded, lowercase md5 hash | Yes for every specified material |
| params.materials[].name | String | The presentation name of the file - usually the original file name | Yes for every specified material |
| params.materialsHttp | Array | A list of materials for this batch to be copied from http | Either materials, materialsHttp or both must be present and have content |
| params.materialsHttp[].url | String | The absolute url of the file: must be url encoded | Yes for every specified materialHttp |
| params.materialsHttp[].size | Number | The size of the file in bytes | Yes for every specified materialHttp |
| params.materialsHttp[].name | String | The presentation name of the file - usually the original file name | Yes for every specified materialHttp |

Validations

General validations:

- Verify that batch ready for production has not been called yet.
- Verify that file names are not duplicated. The batch does not support the same file name twice.
- The callbackUrl must be RFC 2396 compliant.

Materials validations:

- If MD5 is in incorrect format this method will fail.
- If the specified MD5 does not match the MD5 of the file in AWS S3, an error callback will be made.

MaterialsHttp validations:

- If size is not specified this method will fail.
- If the copied file size does not match the specified size, an error callback will be made

Output Parameters

| Path | Type | Description |
|--------|--------|--|
| result | String | <p>"materialsAdded" value if saving the request succeeded.</p> <p>Please note that this does not mean that we've copied the materials to our system.</p> <p>After the materials are copied the callbackUrl will be called.</p> |

Related error codes

| Error code | Error Description |
|------------|---|
| -32600 | Materials cannot be null or empty! |
| -32600 | Materials contain duplicate name [filename.jpg] |
| -32600 | <p>Location/Name/Size cannot be null or empty</p> <p>Bucket/ObjectKey/Md5 cannot be null or empty</p> |
| -32600 | [Reference] contains unsupported characters (non BMP UTF-8) |
| -32610 | Batch ready for production was already called for batch id: [BatchId] |

JSON example

Response

```
{
  "id": 904630500,
  "method": "addMaterialsForBatch",
  "jsonrpc": "2.0",
  "params": {
    "clientId": "clientId",
    "reference": "referenceNumber",
    "callbackUrl":
"http://www.customer.com/?batchId=1234567890&copiedfilesFor=myUniqueIdentifier",
    "batchId": 1234567890,
    "materials": [
      {
        "bucket": "myownbucket",
        "key": "myfolder/and/file/at.aws",
        "md5": "46c4c16590a61fb3629fb2db079e6f11",
        "name": "img_0149.jpg"
      },
      {
        "bucket": "myownbucket",
        "key": "myfolder with *special %chars/and -90+50/file with #special'
chars\" at AWS.mp4",
        "md5": "1777705650cf8948715dbc28e943e6e7",
        "name": "file with #special' chars\" at AWS.mp4"
      }
    ],
    "materialsHttp": [
      {
```

```

        "url": "http://dc609.4shared.com/download/JrrDQacR/tsid20140603-071318-
f80e9752/Beautiful_Trees.jpg",
        "name": "img_0150.jpg",
        "size": 7362891
    },
    {
        "url": "http://green-
trees.weebly.com/uploads/2/3/9/8/23981799/my+special+folder+(-
90%2B50)/the%25%27*%3B%22/image003.jpg",
        "name": "image003.jpg",
        "size": 4325543
    }
]
}
}

```

Response

```

{
  "id" : 904630500,
  "jsonrpc" : "2.0",
  "result" : "materialsAdded"
}

```

Error Response

If errors happen when calling this method the normal error message will be returned as response:

```

{
  "id" : 904630500,
  "error": {
    "code": -32604,
    "message": "Batch id not found"
  }
}

```

Error duplicate file name

```

200 OK
{
  "jsonrpc": "2.0",
  "id": xxxxxxxx,
  "error": {
    "code": -32600,
    "message": "Materials cannot be null or empty!"
  }
}

```

STEP 4 - eWarp 2.0 method: batchReadyForProduction

Description

This operation must be used for all batches.

If a batch (order line) requires materials to be uploaded before the batch is ready for production, the [eWarp 2.0 method: addMaterialsForBatch](#) must be used before batchReadyForProduction is called for the first time, and only after matching [eWarp 2.0 eStinger: MATERIALS_ADDED](#) notification is received.

When this operation is successfully done the batch will now be queued in esoft systems for production.

It is therefore necessary to let esoftsystems know which company this batch is ready for. Contact esoftsystems to get information on which company to use.

If this operation have been called once, calling it again will have no effect. If materials have been added and they are not yet copied, or the copying failed this operation will return an error and not have any effect.

Preconditions

eWarp user must have access to specified **clientId** resources.

Input Parameters

| Path | Type | Description | Required |
|----------------------------|--------|---------------------------|----------|
| params.clientId | String | The client id | yes |
| params.batchId | Number | The batch id | yes |
| params.esoftsystemsCompany | String | The esoft systems company | yes |

Output Parameters

| Path | Type | Description |
|--------|--------|--------------------------------|
| result | String | "OK" value if update went well |

JSON example

Request

```
{
  "id" : 75574333,
  "method" : "batchReadyForProduction",
  "jsonrpc" : "2.0",
  "params" : {
    "clientId" : "clientId",
    "batchId" : 1234567890,
    "esoftsystemsCompany" : "ESVN"
  }
}
```

Response with the requested info

```
{
  "id" : 75574333,
  "jsonrpc" : "2.0",
```

```
"result" : "OK"  
}
```

STEP 6 - eWarp 2.0 method: getAssets

Description

This operation should be called to retrieve the details of the materials for the reference number (case) for the particular Client. All agreed configurations for the different processed assets are specified in the [eWarp Usage Agreement](#). It is not allowed in any way to modify the returned URL's as they can change in both format and location over time (if they change an automatic notification through eStinger will occur).

ListOrder

Please note that the field "**listOrder**" will always be present (except for **links** - as links cannot be ordered). It contains the order of which the materials should be shown. If the value is 0 then the material should be considered as inactive and not be shown. If listOrder is greater than 0 then the materials should be shown in ascending order (listOrder 1 first).

Archival

Assets for a reference can be archived and no longer available. Archiving will be for all assets for a reference so in effect it will be the reference which is archived. This is visible in a parameter in the returned result.

Preconditions

eWarp user must have access to specified **clientId** reference.

Input Parameters

| Path | Type | Description | Required |
|------------------|--------|------------------|----------|
| params.clientId | String | The client id | yes |
| params.reference | String | The reference id | yes |
| params.batchId | Long | The batchId | no |

Output Parameters

When the reference state is not ACTIVE no assets will be returned.

An error is returned if the clientId and reference could not be found.

| Path | Type | Description |
|-------------------------|--------|---|
| result.state | String | The state of assets Can be one of the following ACTIVE ARCHIVING RESTORING ARCHIVED. |
| result.pictures | Array | The list of picture assets for the reference See Asset types: Picture for more details on the picture assets structure |
| result.animatedPictures | Array | The list of animatedPicture assets for the reference See Asset types: Animated picture for more details on the animatedPicture assets structure |
| result.floorPlans | Array | The list of floorPlan assets for the reference See Asset types: Floor plan for more details on the floorPlan assets structure |
| result.panoramaHtml5 | Array | The list of panoramaHtml5 assets for the reference See Asset types: Panorama Html5 for more details on the panoramaHtml5 assets structure |
| result.panoramaFlash | Array | The list of panoramaFlash assets for the reference See Asset types: Panorama Flash for more details on the panoramaFlash assets structure |
| result.panoramaJpeg | Array | The list of panoramaJpeg assets for the reference See Asset types: Panorama Jpeg for more details on the panoramaJpeg assets structure |
| result.videos | Array | The list of video assets for the reference See Asset types: Video for more details on the video assets structure |
| result.slideShows | Array | The list of slideShow assets for the reference See Asset types: Slideshow for more details on the slideShow assets structure |
| result.links | Array | The list of link assets for the reference See Asset types: Link for more details on the link assets structure |
| result.copywriteTexts | Array | The list of copywriteText assets for the reference See Asset types: Copywrite texts for more details on the copywriteText assets structure |

| Path | Type | Description |
|--------------------------------------|--------|---|
| result.qrCodes @Deprecated | Array | <p>The list of QR code assets for the reference</p> <p>See Asset types: QR Code for more details on the QR code assets structure</p> <p>QR code is deprecated and will be removed in next releases</p> |
| result.beforeAfterPictures | Array | <p>the list of beforeAfterPicture assets for given reference</p> <p>See Asset types: BeforeAfterPicture for more details</p> |
| result.beforeAfterPanoramas | Array | <p>the list of beforeAfterPanoramas assets for given reference</p> <p>See Asset types: BeforeAfterPanorama for more details</p> |
| result.interactivePresentation | Object | <p>The interactive floor plan asset</p> <p>See Asset types: Interactive presentation for more details on the interactive floor plan asset structure</p> |
| result.poiPresentation | Object | <p>The points of interest asset</p> <p>See Asset types: POI presentation for more details on the points of interest asset structure</p> |

JSON example

Request

```
{
  "id" : 694664211,
  "method" : "getAssets",
  "jsonrpc" : "2.0",
  "params" : {
    "clientId" : "clientId",
    "reference" : "referenceNumber",
    "batchId" : batchIdNumber
  }
}
```

Response

```
{
  "jsonrpc":"2.0",
  "id":113578333,
  "result":{
    "state" : "ARCHIVED",
    "pictures":[],
    "animatedPictures":[],
    "floorPlans":[],
    "panoramasHtml5":[],
    "panoramasFlash":[],
    "panoramasJpeg":[],
    "videos":[],
    "slideShows":[]
  }
}
```

```

    "links": [],
    "copywriteTexts": [],
    "qrCodes": [],
    "interactivePresentation": null,
    "poiPresentation": null,
    "beforeAfterPictures" : [],
    "beforeAfterPanoramas" : []
  }
}

```

Full sample can be downloaded here: [full sample can be downloaded here](#)

Error Response

```

{
  "jsonrpc": "2.0",
  "id": 694664211,
  "error": {
    "code": -32004,
    "message": "The reference was not found."
  }
}

```

Asset types

Note: in following section, we'll only describe Picture & Video asset types

- [Picture](#)
- [Floor plan](#)
- [Animated picture](#)
- [Panorama Html5](#)
- [Panorama Flash](#)
- [Panorama Jpeg](#)
- [Video](#)
- [SlideShows](#)
- [QR Code](#)
- [Copywrite texts](#)
- [Links](#)
- [Interactive presentation](#)
- [POI presentation](#)
- [BeforeAfterPicture](#)
- [BeforeAfterPanorama](#)
- [Object types](#)
 - [AssetType](#)
 - [EmbedAssetType](#)
 - [ProfileType](#)
- [Enumeration types](#)

Picture

| Path | Type | description |
|----------------------|--------|-------------|
| result.pictures[].id | Number | Asset id |

| | | |
|---------------------------------|--------------------|--|
| result.pictures[].batchId | Number | BatchId the asset belongs to |
| result.pictures[].updatedAt | Number (Unix time) | Asset updatedAt updatedAt is only updated when the file has changed due to correction |
| result.pictures[].listOrder | Number | Asset listOrder |
| result.pictures[].description | String | Asset description |
| result.pictures[].tags[] | String | List of asset tags |
| result.pictures[].name | String | Asset file name |
| result.pictures[].referenceType | String | See ReferenceEnum below for more details |
| result.pictures[].orientation | String | See OrientationEnum below for more details |
| result.pictures[].assets | Object | See AssetType below for more details |

JSON example

Pictures Example

```

"pictures": [
  {
    "id": 153927002,
    "description": null,
    "updatedAt": 1625479068,
    "referenceType": "CASE",
    "hotspots": null,
    "batchId": 50000744943,
    "listOrder": 0,
    "tags": [
      "picture"
    ],
    "name": "DSC04945.JPG",
    "orientation": "HORIZONTAL",
    "assets": [
      {
        "url": "http://cdn-dev-
m2.esoftsystems.com/50100174/ESOFT+TEST+PHOTO+20210706/50000744943/ORIGINAL/DSC04945.JPG",
        "secureUrl": "https://cdn-dev-
m2.esoftsystems.com/50100174/ESOFT+TEST+PHOTO+20210706/50000744943/ORIGINAL/DSC04945.JPG",
        "md5": "a86bc9c72802be9dc3eb3b7eb6e812d4",
        "actualWidth": 3000,
        "actualHeight": 2004,
        "resizeRule": "ORIGINAL",

```



```

    "format": "JPG"
  },
  {
    "url": "http://cdn-dev-
m2.esoftsystems.com/50100174/ESOFT+TEST+PHOTO+20210706/50000744943/WHITE_FILL/170/113/DSC04945.jpg",
    "secureUrl": "https://cdn-dev-
m2.esoftsystems.com/50100174/ESOFT+TEST+PHOTO+20210706/50000744943/WHITE_FILL/170/113/DSC04945.jpg",
    "md5": "274e9079fc821f121f2ef32ea650987d",
    "actualWidth": 170,
    "actualHeight": 113,
    "resizeRule": "WHITE_FILL",
    "format": "JPG",
    "width": 170,
    "height": 113
  }
]
}
]

```

Video

| Path | Type | description |
|---------------------------------|--------------------|--|
| result.videos[].id | Number | Asset id |
| result.videos[].batchId | Number | BatchId the asset belongs to |
| result.videos[].updatedAt | Number (Unix time) | Asset updatedAt updatedAt is only updated when the file has changed due to correction |
| result.videos[].listOrder | Number | Asset listOrder |
| result.videos[].description | String | Asset description |
| result.videos[].tags[] | String | List of asset tags |
| result.videos[].posterUrl | String | Asset posterUrl (HTTP) |
| result.videos[].securePosterUrl | String | Asset securePosterUrl (HTTPS) |
| result.videos[].name | String | Asset name |
| result.videos[].referenceType | String | See ReferenceEnum below for more details |
| result.videos[].assets | Object | See AssetType below for more details |
| result.videos[].embedAssets | Object | See EmbedAssetType below for more details |
| result.videos[].profiles | Object | See ProfileType below for more details |

| | | |
|--------------------------|--------|---|
| result.videos[].duration | Number | Duration of the video in milliseconds This value will be null for: <ul style="list-style-type: none"> ▪ videos transcoded before October 2019 ▪ if only ORIGINAL profile is present |
|--------------------------|--------|---|

JSON example

Videos Example

```

"videos": [
{
  "id": 153927001,
  "description": "Video",
  "updatedAt": 1625572739,
  "referenceType": "CASE",
  "hotspots": null,
  "batchId": 50000744944,
  "listOrder": 0,
  "tags": [],
  "assets": [
    {
      "url": "http://cdn-dev-
m2.esoftsystems.com/50100174/ESOFT+TEST+VIDEO+ORDER+20210706/50000744944/1983938387/thumbnails/170/113/500
03563374.jpg",
      "secureUrl": "https://cdn-dev-
m2.esoftsystems.com/50100174/ESOFT+TEST+VIDEO+ORDER+20210706/50000744944/1983938387/thumbnails/170/113/500
03563374.jpg",
      "width": 170,
      "height": 113
    }
  ],
  "embedAssets": [
    {
      "width": "100%",
      "height": "100%",
      "url": "http://cdn-dev-
m2.esoftsystems.com/50100174/ESOFT+TEST+VIDEO+ORDER+20210706/50000744944/1983938387/embed_codes/100%25/1
00%25/50003563374_embed.html",
      "secureUrl": "https://cdn-dev-
m2.esoftsystems.com/50100174/ESOFT+TEST+VIDEO+ORDER+20210706/50000744944/1983938387/embed_codes/100%25/1
00%25/50003563374_embed.html",

```

```
"embedCode": "<script src=\"https://cdn-dev-m2.esoftsystems.com/video/esoftvideoplayer.js\"></script>\n<div  
id=\"esoft_video_1983938387\" style=\"width:100%;height:100%;\">\n  Loading video player...<noscript>ERROR:<br/>Javascript  
not activated</noscript>\n</div>\n<script type=\"text/javascript\">\n  loadEsoftPlayer(\"esoft_video_1983938387\", \"https://cdn-  
dev-  
m2.esoftsystems.com/50100174/ESOFT+TEST+VIDEO+ORDER+20210706/50000744944/1983938387/poster/50003563374-  
00002.jpg\", [{file: \"https://cdn-dev-  
m2.esoftsystems.com/50100174/ESOFT+TEST+VIDEO+ORDER+20210706/50000744944/1983938387/esoft-  
1080p/50003563374-esoft-1080p.mp4\", label: \"1080p\"}], {file: \"https://cdn-dev-  
m2.esoftsystems.com/50100174/ESOFT+TEST+VIDEO+ORDER+20210706/50000744944/1983938387/original/50003563374.m  
p4\", label: \"ORIGINAL\"}], \"100%\", \"100%\");</script>\"  
}  
],  
"name": "50003563374.mp4",  
"posterUrl": "http://cdn-dev-  
m2.esoftsystems.com/50100174/ESOFT+TEST+VIDEO+ORDER+20210706/50000744944/1983938387/poster/50003563374-  
00002.jpg",  
"posterSecureUrl": "https://cdn-dev-  
m2.esoftsystems.com/50100174/ESOFT+TEST+VIDEO+ORDER+20210706/50000744944/1983938387/poster/50003563374-  
00002.jpg",  
"profiles": [  
{  
  "url": "http://cdn-dev-  
m2.esoftsystems.com/50100174/ESOFT+TEST+VIDEO+ORDER+20210706/50000744944/1983938387/esoft-  
1080p/50003563374-esoft-1080p.mp4",  
  "secureUrl": "https://cdn-dev-  
m2.esoftsystems.com/50100174/ESOFT+TEST+VIDEO+ORDER+20210706/50000744944/1983938387/esoft-  
1080p/50003563374-esoft-1080p.mp4",  
  "md5": "1b8c53a57b25706c530166c0f99d30aa-13",  
  "profile": "esoft-1080p"  
},  
{  
  "url": "http://cdn-dev-  
m2.esoftsystems.com/50100174/ESOFT+TEST+VIDEO+ORDER+20210706/50000744944/1983938387/original/50003563374.m  
p4",  
  "secureUrl": "https://cdn-dev-  
m2.esoftsystems.com/50100174/ESOFT+TEST+VIDEO+ORDER+20210706/50000744944/1983938387/original/50003563374.m  
p4",  
  "md5": "433946a8c3b8275a3fc235a729ea0a64",  
  "profile": "ORIGINAL"  
}  
],
```

```
"duration": 90140
}
]
```

Object types

Assets

| Path | Type | description |
|--------------|--------|---|
| url | String | The file url (HTTP) |
| secureUrl | String | The file secureUrl (HTTPS) |
| md5 | String | The file md5 (<i>ommitted under Videos, Slideshows, Panoramas</i>) |
| resizeRule | String | The resize rule used for scaling the asset See ResizeRuleEnum (<i>ommitted under Videos, Slideshows, Panoramas</i>) |
| format | String | The format used for scaling the asset (<i>ommitted if none</i>) See FormatEnum |
| width | Number | The asset width used for scaling the asset (<i>ommitted if resizeRule=ORIGINAL</i>) |
| height | Number | The asset height used for scaling the asset (<i>ommitted if resizeRule=ORIGINAL</i>) |
| dpi | Number | The asset dpi used for scaling the asset (<i>ommitted if none</i>) |
| quality | Double | The asset quality used for scaling the asset (<i>ommitted if none</i>) |
| actualWidth | Number | The actual width of the asset (<i>ommitted under Videos, Slideshows, Panoramas</i>) - <i>this value might differ from width if resizeRule is BEST_FIT</i> |
| actualHeight | Number | The actual height of the asset (<i>ommitted under Videos, Slideshows, Panoramas</i>) - <i>this value might differ from height if resizeRule is BEST_FIT</i> |

JSON example

Asset - all fields example

```
{
  "url": "http://cdn.dev.m2.esoftsystems.com/50100010/ref1/09.jpg",
  "secureUrl": "https://cdn.dev.m2.esoftsystems.com/50100010/ref1/09.jpg",
  "md5": "76bb4cfd7ade36452580ce48c0a6ff09",
  "resizeRule": "BEST_FIT",
  "format": "JPG",
  "width": 906,
  "height": 1360,
  "dpi": 72,
  "quality": 50.0,
  "actualWidth":906,
  "actualHeight":1220
}
```

Asset - minimum fields example

```

{
  "url": "http://cdn.dev.m2.esoftsystems.com/50100010/ref1/09.tiff",
  "secureUrl": "https://cdn.dev.m2.esoftsystems.com/50100010/ref1/09.tiff",
  "md5": "76bb4cfd7ade36452580ce48c0a6ff09",
  "resizeRule": "ORIGINAL"
},
{
  "url":
"http://cdn.dev.m2.esoftsystems.com/examples/video/01/thumbnail/200/120/movie.png",
  "secureUrl":
"https://cdn.dev.m2.esoftsystems.com/examples/video/01/thumbnail/200/120/movie.png",
  "width": 200,
  "height": 120
}

```

EmbedAssetType

| Path | Type | description |
|-----------|--------|---|
| url | String | The embed code file url (HTTP) |
| secureUrl | String | The embed code file secureUrl (HTTPS) |
| embedCode | String | The embed code |
| width | String | The width of the embed content placeholder |
| height | String | The height of the embed content placeholder |

JSON example

EmbedAssetType Example

```

{
  "url": "http://cdn.dev.m2.esoftsystems.com/panorama/esoft_panorama_10.html",
  "secureUrl":
"https://cdn.dev.m2.esoftsystems.com/panorama/esoft_panorama_10.html",
  "embedCode": "<object><\object>",
  "width": "600px",
  "height": "400px"
}

```

Profiles

| Path | Type | description |
|-----------|--------|---|
| url | String | The embed code file url (HTTP) |
| secureUrl | String | The embed code file secureUrl (HTTPS) |
| md5 | String | The md5 of the profile file |
| profile | String | The profile used for the file transcoding |

JSON example

ProfileType Example

```
{
  "url": "http://cdn.dev.m2.esoftsystems.com/examples/video/01/supermovie.mp4",
  "secureUrl":
    "https://cdn.dev.m2.esoftsystems.com/examples/video/01/supermovie.mp4",
  "md5": "6a09051bed516bc67651eaa584ab543c",
  "profile": "esoft-720p"
}
```

Enumeration types

| Enum type | Enumeration values |
|------------------------|--|
| OrientationEnum | HORIZONTAL, VERTICAL, SQUARE |
| ResizeRuleEnum | ORIGINAL, BEST_CROP, BEST_FIT, WHITE_FILL |
| ReferenceEnum | CASE, LOCATION, PORTRAIT, SHOP, GROUP, INTRO_OUTRO |
| FormatEnum | PNG, JPG, SVF, PDF or none (not present) for other formats. <u>The url to the file will end with file extension matching the file format.</u> |