

# Case Study: Batch Update

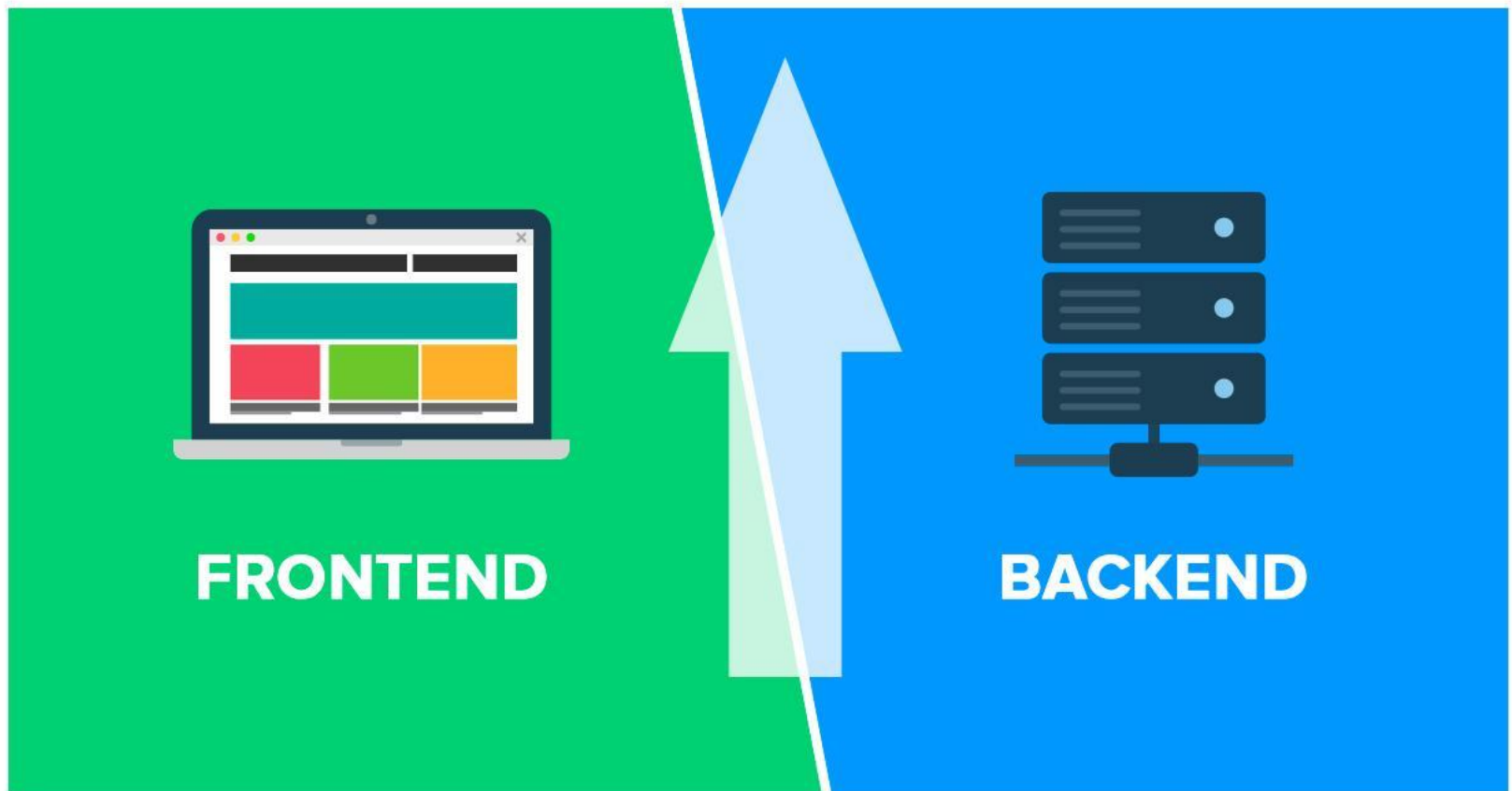
## A User Driven Approach

What we talk about when we talk about design thinking

Karen Shaw  
Adam Arling

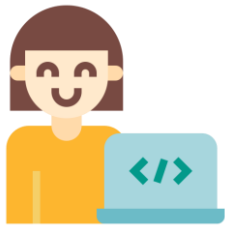
Northwestern

The NUL dev team is  
refining a new workflow



<https://medium.com/swlh/front-end-first-software-development-approach-1aa8b7f35447>

# What did we do before?



Northwestern

LIBRARIES | Alpha 1.0

Search

karen.didrickson@northwestern.edu

Home / Dashboard / Batches

### Batches

Status	Job ID	Submitter	Original Filename	Submitted
Complete	82a9d549-4874-4f7b-b4c7-f88062e8f1c	j-young2@northwestern.edu	box-34-broken-images.csv	4 days ago
Complete	34ff1ba7-ef64-46f9-8231-cd8f8464d79e	j-young2@northwestern.edu	yet_more_box_34_corrections.csv	4 days ago
Error	80843f11-d7f3-4ef9-a87a-8a57d308e948	j-young2@northwestern.edu	box_34_corrections.csv	4 days ago
Error	0e1c930a-adff-47b6-ad7d-d2df39ebc6e8	j-young2@northwestern.edu	Box_34_BFMP_Metadata_reconciled.csv	5 days ago
Complete	9cb3ef4f-2318-43cd-b03d-530f35804a08	erin.gilchrist@northwestern.edu	BFMF_batch_ingest_Box36_no_temps_final.csv	5 days ago
Complete	2c2ef3cc-dcb7-4321-8caf-b0c7291cde88	erin.gilchrist@northwestern.edu	BFMF_batch_ingest_Box36_no_temps_final.csv	6 days ago
Error	33960671-90cf-48a3-b5e8-ceecb80cdae	erin.gilchrist@northwestern.edu	BFMF_batch_ingest_Box36_no_temps_final.csv	6 days ago
Complete	895076b1-c0fc-4ccf-8620-6eff801163f9	erin.gilchrist@northwestern.edu	BFMF_batch_ingest_Box37_no_temps_final.csv	7 days ago
Complete	3ab54866-c23f-4f91-b94f-2c2c02ce68d1	erin.gilchrist@northwestern.edu	BFMF_batch_ingest_Box37_no_temps_final.csv	7 days ago

- Developers built and delivered finished product
- Not iterative, product was hard to change once delivered
- Resulting user experience lacking

# Design Thinking



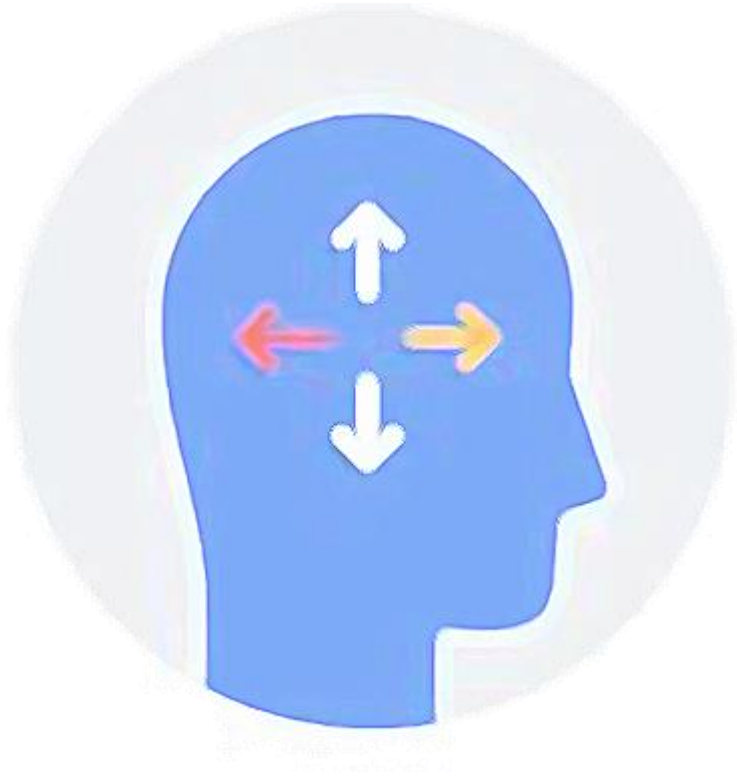
# Empathy



Users should be the *#1 focus*.

By hearing **needs**, **feelings** and **motivations**, one can create *meaningful solutions* to actual problems.

# Expansive Thinking



aka “Brainstorming”

~~One perfect solution~~

Look at problem from *all conceivable angles*

Not 10% better, 10X better. Go big.

# Experimentation



*Quickly* figure out which ideas **do** and **don't** work

Prototype *early-stage* versions

Feedback



# Building a better batch edit

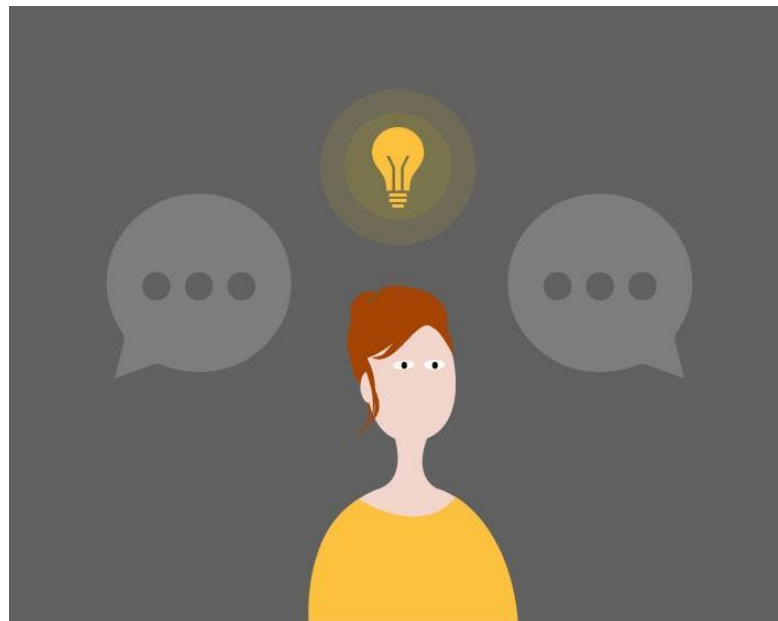


# What problem are we looking to solve?



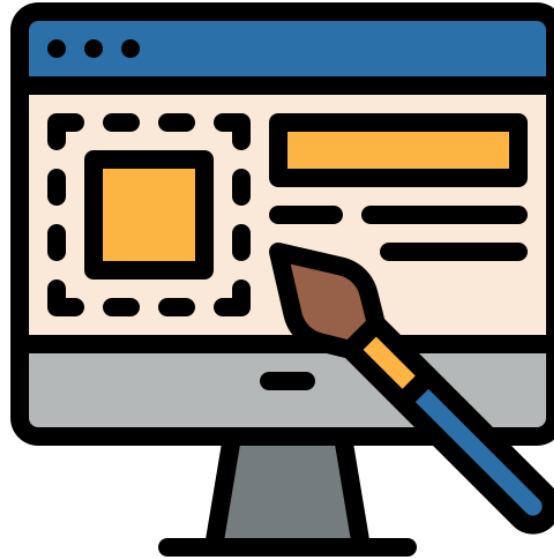


“We want to ingest our assets so that they are tracked in our preservation system, and then be able to describe them in large batches so that we can quickly publish them for end user and researcher consumption.”





“We want an application that ....”




- Has a simple, intuitive UI
- Scales well. (A batch based system by default)
- Is transparent - ability to queue jobs, monitor state, success, failure and progress, see meaningful errors



# Initial Prototypes (Wireframes)

## Batch Dashboard

 **Northwestern | Libraries**

### Batch Dashboard

Started	ID	User	Nickname	status	Works updated	Details
2020-02-02	12352	David Schober	remove bad genre from Gutner	done	1000	<a href="#">view details</a>
2020-02-02	12354	Laura Alagna	remove bad genre from Gutner	done	500	<a href="#">view details</a>
2020-02-02	12341	Michael Klein	Change Title	done	250	<a href="#">view details</a>
2020-02-02	12345	Michael Klein	Change description	error	0	<a href="#">view details</a>
2020-02-02	123415	Michael Klein	Change description	error	0	<a href="#">view details</a>
2020-02-02	342415	Laura Alagna	Change collection on works	In progress		<a href="#">view details</a>
2020-02-02	459392	David Schober		queued		<a href="#">view details</a>
2020-02-02	231231	David Schober		queued		<a href="#">view details</a>

[←](#) [1](#) [2](#) [3](#) [...](#) [8](#) [9](#) [10](#) [→](#)

Note:

The goal is to give users a quick overview of running and completed batches. Ideally we index the batch's unique id so that users can quickly link to a search page showing the works updated using the "XXX Works Updated" link. View details would link to a page with the original query, and number of works updated. We could also do this in a single table if we collapse/expand on the errors





# Initial Prototypes (Wireframes)

## Batch Edit Screens

Northwestern | Libraries

Search for items

Home / Resources

### Batch Edit

Editing 300 Works

Search Result Title

Remove Creator(s)

Search

☐ Apple  
☐ Banana  
☐ Papaya  
☐ Raspberry  
☐ Strawberry  
☐ Mango  
☐ Kiwi

Submit

Creator(s)

Creator Role

Select one...

+ Add Another Creator

- Find Creators to Re...

Contributors

Role

Select one...

+ Add Another Contribu...

- Find Contributors to

Physical Description

+ Add Another Physical Description

Physical Description Material

Northwestern | Libraries

Search for items

### Batch Edit Confirmation

#### Adding

- + Creator: Guitarist, George
- + Creator: Bassist, George
- + Contributor: Drummer, George Harrison
- + Physical Description Size: 12 inch Round

#### Removing

- Singer, Iggy Pop
- Punk Rock Music

NOTE: This will affect all currently affected works. Please batch edit with care. To execute this change type:

"I understand"

Cancel Submit

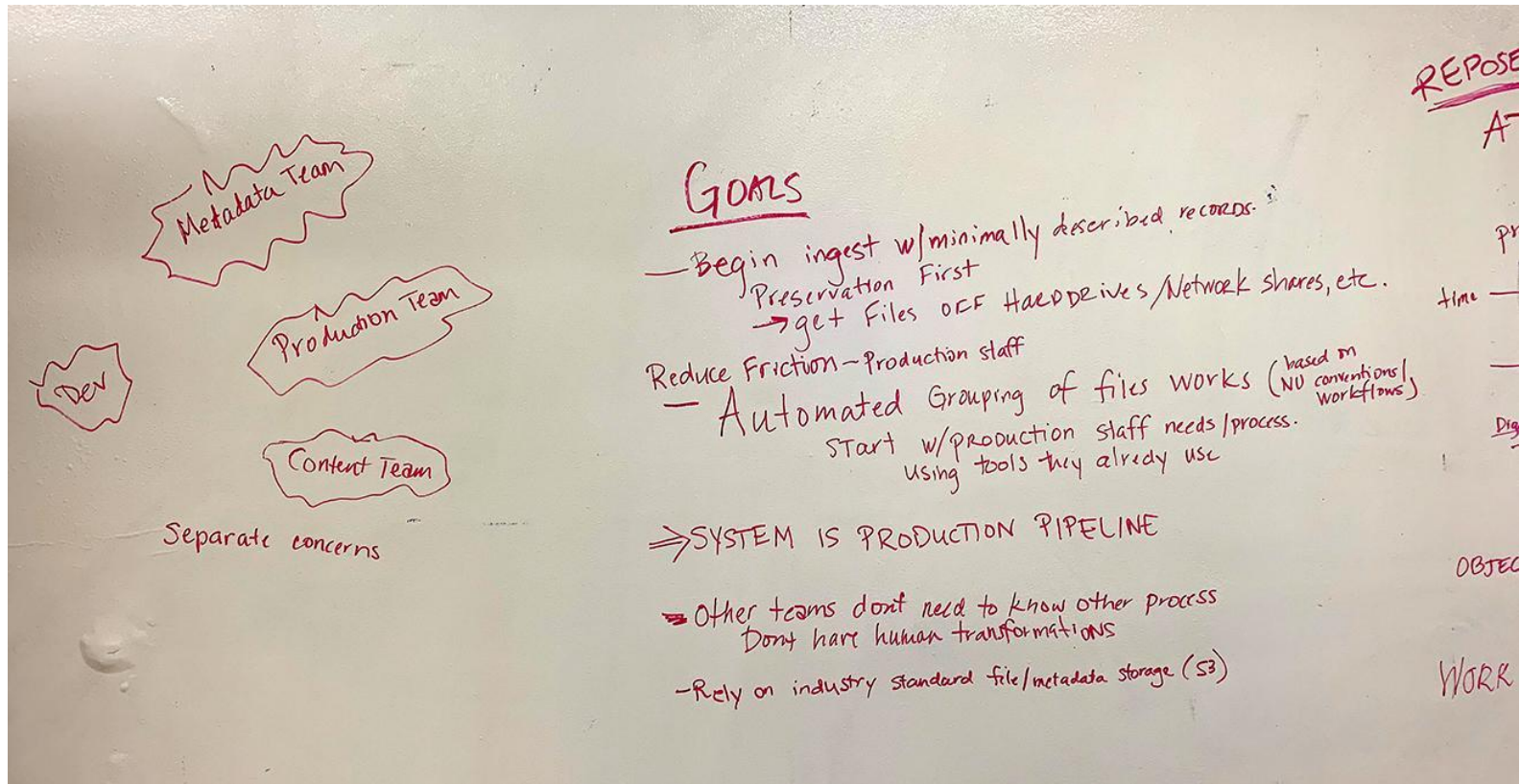
Meadow

Collection Work About Blog Contact

Copyright information goes down here Legal Notice Privacy Policy



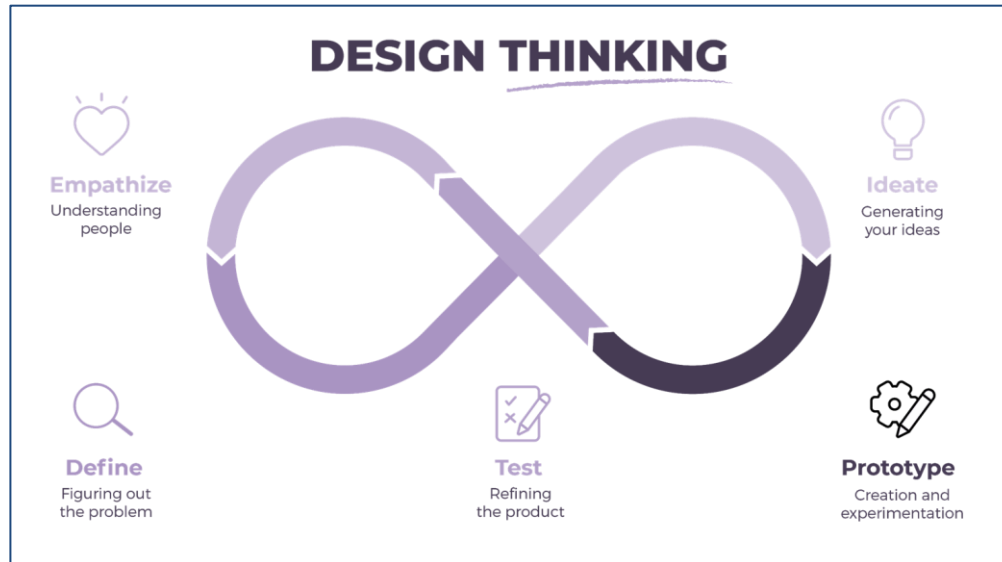
# White boarding



“How often do we ingest records? How often do we select batch records?”



# Prototyping - Early and Often



Wireframes &  
Whiteboarding



API Design



API Mocking





# API Design

How do we pass the search result set to the backend?  
What if there are thousands of id's?

Should we query through GraphQL or Elasticsearch?  
What about a lag between the two?

When someone updates a field, is that an append or replace operation?

How do we prevent race conditions?

Who is responsible for scrolling Elasticsearch results? The front or the back end?

Do we validate for required fields during a batch operation?



# API Mocking

Search the schema ...

SCHEMA

workByAccession(...): Work

works(...): [Work]

MUTATIONS

addGroupToRole(...): StatusMessage

addWorkToCollection(...): Work

addWorksToCollection(...): Collection

approveIngestSheet(...): StatusMessage

batchUpdate(...): Message

createCollection(...): Collection

createFileSet(...): FileSet

createIngestSheet(...): IngestSheet

createProject(...): Project

createSharedLink(...): SharedLink

createWork(...): Work

deleteCollection(...): Collection

deleteFileSet(...): FileSet

deleteIngestSheet(...): IngestSheet

deleteProject(...): Project

deleteWork(...): Work

removeWorksFromCollection(...):

batchUpdate(  
 add: BatchAddInput,  
 delete: BatchDeleteInput =  
 {contributor: [], creator: [], genre: [],  
 language: [], location: [],  
 stylePeriod: [], subject: [],  
 technique: []},  
 query: String!,  
 replace: BatchReplaceInput  
): Message

Start a batch update operation

TYPE DETAILS

type Message {  
 message: String

ARGUMENTS

add: BatchAddInput

delete: BatchDeleteInput =  
 {contributor: [], creator: [], genre: [],  
 language: [], location: [], stylePeriod: [],  
 subject: [], technique: []}

query: String!

replace: BatchReplaceInput

replace: BatchReplaceInput

replace replaces existing values  
(single and multi valued fields)

TYPE DETAILS

Input fields for batch replace  
operations

type BatchReplaceInput {  
 administrativeMetadata:  
 BatchReplaceAdministrativeMetadataInp  
 collectionId: ID  
 descriptiveMetadata:  
 BatchReplaceDescriptiveMetadataInput  
 published: Boolean  
 visibility: CodedTerminInput  
}

descriptiveMetadata:  
BatchReplaceDescriptiveMetadataInp  
ut

TYPE DETAILS

Input fields available for batch replace  
operations on works descriptive  
metadata

type BatchReplaceDescriptiveMetadataInp  
 abstract: [String]  
 alternateTitle: [String]  
 boxName: [String]  
 boxNumber: [String]  
 caption: [String]  
 catalogKey: [String]  
 description: [String]  
 folderName: [String]  
 folderNumber: [String]  
 keywords: [String]  
 license: CodedTerminInput  
 notes: [String]  
 physicalDescriptionMaterial: [String]  
 physicalDescriptionSize: [String]  
 provenance: [String]  
 publisher: [String]  
 relatedMaterial: [String]

# Keep on prototypin'

Iterate on modular prototypes

# Search driven selection?

**N Meadow** v1.x.x Projects Themes & Collections Dashboards

### Filter/Facet

#### Contributor

Search

- ☐ Schober, David, 1974- (Actor) 494
- ☐ Catholic Church (Abridger) 199
- ☐ Schober, David, 1974- (Adapter) 172
- ☐ Hopper, Grace Murray (Libellant) 157
- ☐ Valim, Jose (Voice actor) 156
- ☐ Mandela, Nelson, 1918-2013 (Voice actor) 155
- ☐ Dewey, Melvil, 1851-1931 (Voice) 153

#### Collection

Search

- ☐ usability test collection 1386
- ☐ cats should be dogs 172

#### Project

Search

- ☐ ux\_testing 1730
- ☐ Karen Validation Testing 1503

Ingest Sheet


Search all works

## Search Results

Edit All 3233 Items View and Edit 0 Items Export CSV


List View Grid View

3233 results found in 4ms




Dolorem commodi suscipit hic beatae quaerat!

Image Private




Voluptatum rerum facilis modi nemo odio facere nihil adipisci!

Image Private



Qui vero repudiandae in?

Image Private



Dolores omnis explicabo voluptatem quam ad nisi ut qui.

Image Private



# Manually select items?

## Search Results

Edit All 297 Items

View and Edit 4 Items

Deselect All

Export CSV

List View

Grid View

297 results found in 9ms

Sed voluptatem dolor molestiae dolore numquam veniam.

Image Private

Voluptate corporis ex quam!

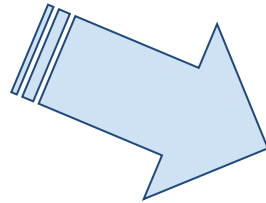
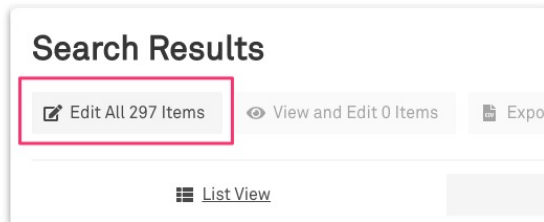
Image Private

Cum ut ut mollitia.

Image Private



# “Broad stroke” editing?



```
{ query: {  
  ...items in {  
    ... { "color":  
    "blue" }  
  }  
}
```

N Meadow v 1.x.x Projects Themes & Collections Dashboards 🔔 👤 🔍

About this item Administrative

Core and Descriptive Metadata Save Data for 297 Items [Clear Form](#)

Core Metadata ▾

Title

Alternate Title

+ Add

☐ Replace values

☐ Remove all values

Collection

-- Select -- ▾

Date Created Not Live

mm/dd/yyyy 📅

Display not yet supported Update not yet supported

Rights Statement

-- Select -- ▾

Description



# Add/replace/remove in batch?

Bookmark this tab | OW v 1.x.x | Projects | Themes & Collections | Dashboards | | |

## Core and Descriptive Metadata

[Save Data for 297 Items](#) [Clear Form](#)

### Description Information

#### Abstract

☒ Remove all values

#### Caption

Im the caption to replace all captions

And I'm their cousin tag-along

[+ Add another](#)

☒ Replace values

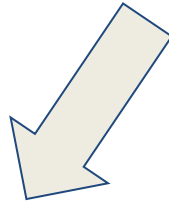
☐ Remove all values



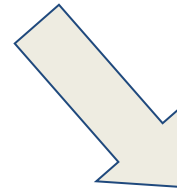
# Get UI in front of users

Is the flow intuitive?

What else might they want?

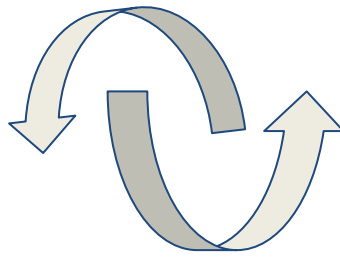


FINAL CONFIRMATION SCREEN?



PREVIEW ITEMS?

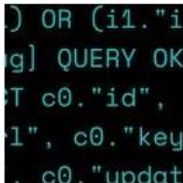




Search > Batch Edit

## Batch Edit

**⚠** You are batch editing the following 2663 items.



About this item

Administrative

Core and Descriptive Metadata

Save Data for 2663 Items

[Clear Form](#)

# Back end work in parallel

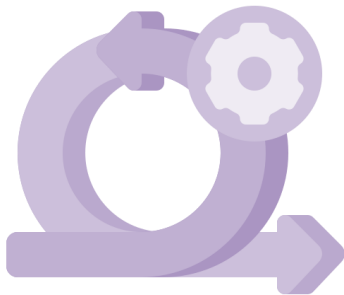
As the result of our process, we knew:

- Our batches would take place within a **transaction** - an entire batch succeeds or fails.
- We needed a simple batchUpdate **mutation** in the API with **replace**, **add** (append) and **delete** inputs, that also takes an Elasticsearch **query**.
- We needed to iterate over the Elasticsearch **scroll** and apply changes to each page of work IDs within one transaction.

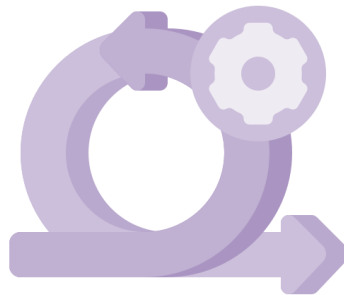
```
1 mutation BatchUpdate($add: BatchAddInput, $delete: BatchDeleteInput, $replace: BatchAddI
2   batchUpdate(add: $add, delete: $delete, replace: $replace, query: $query) {
3     message
4   }
5 }
HTTP HEADERS (0) PRETTIFY

QUERY VARIABLES
1 {
2   "query": "{\\\"query\\\":{\\\"bool\\\":{\\\"must\\\":[{\\\"bool\\\":{\\\"must\\\":[{\\\"bool\\\":{\\\"must\\\":[{\\\"
3   \"add\": {
4     \"collectionId\": \"c64719e3-32b0-465f-8539-ee9ffac36d21\",
5     \"descriptiveMetadata\": {
6   },
7 },
8   \"replace\": {
9     \"descriptiveMetadata\": {
10       \"boxNumber\": [\"3\"],
11       \"description\": [\"Whoops\"]
12     }
13   },
14   \"delete\": {
15     \"contributor\": [{
16       \"term\": \"http://id.loc.gov/authorities/names/n85153068\",
17       \"role\": {
18         \"id\": \"stl\",
19         \"scheme\": \"MARC_RELATOR\"
20       }
21     }, {
22       \"term\": \"http://id.loc.gov/authorities/names/n50053919\",
23       \"role\": {
24         \"id\": \"stl\",
25         \"scheme\": \"MARC_RELATOR\"
```

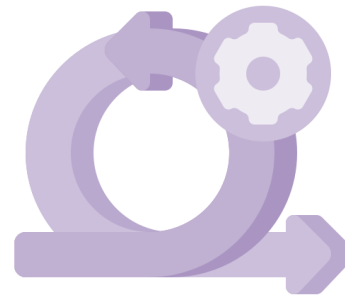
# Back end iterations



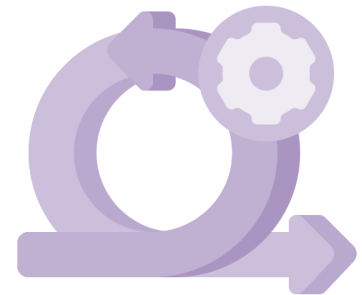
add/delete  
controlled terms



add/replace  
non-controlled  
terms



collection  
assignment



queueing/  
batch  
management

\*Replacing the mocked responses with live responses keeps the API stable and ***should be*** seamless for the front end.

icon: <https://www.flaticon.com/authors/mynamepong>

Ending thoughts...

Do Design Thinking techniques  
aid our team's workflow?