

📄🔍 Day 1: Pagination & Search

🕒 Agenda

- Objectives
- Welcome to Class
- Introduce Intro Project
- Pagination
- Search

🏆 Learning Outcomes

By the end of this lesson, you should be able to...

- Understand how Simple Search works, and how to implement it
- Understand how Pagination works, and how to implement it

👋 Welcome to Class

Instructor will walk through the [syllabus](#) and answer questions about the course.

Students, remember to join the following:

1. Course Slack channel, [##bew-2-1-web-patterns](#) !
2. Gradescope: [INSERT LINK HERE](#)

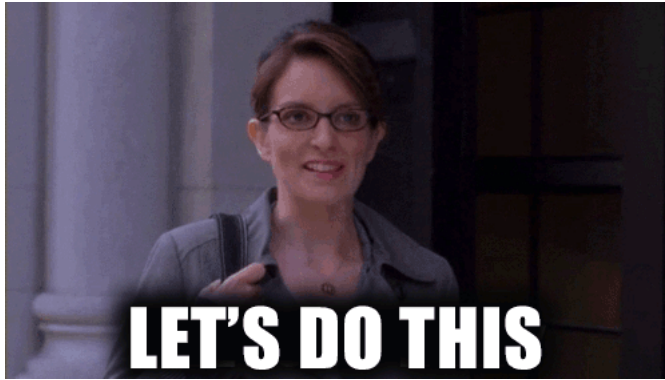
📁 Intro Project

Wait a minute, a project *already?!*

How Not To React:



Instead:



Project Overview

TODO:

- Add project description
- Where code will be written

Existing Vs New Projects

You may be asking, “why are we enhancing an existing project instead of creating a new one?”



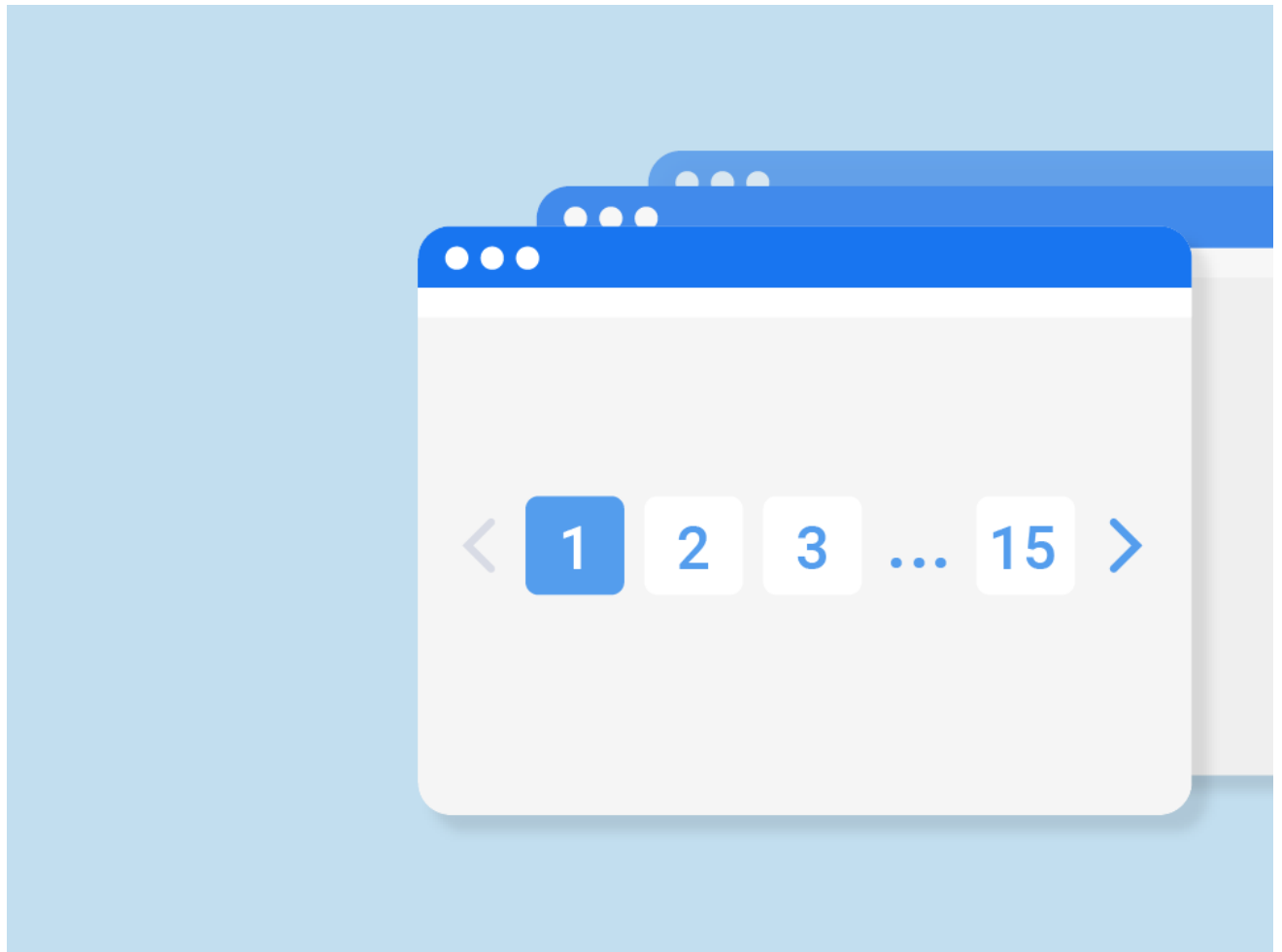
- More common as a junior engineer to be working on an existing project
- You will have to revamp/add features to an existing project at some point in your career
- Improve your existing portfolio projects to make them stand out more

Pagination

Why do we need pagination?

Question: What is the fastest way to speed up a query for 1,000,000,000 (1 Billion) records?

- ▶ The answer (do not click!)



If you look around almost every website is paginated. Why? Probably because pagination is **one of the easiest ways to speed** your index page, that will take 10 seconds to load. Pagination will speed it up by sending only the first 20 records.

Question

What are other benefits to pagination?

- Improved structure and readability: reduced chance of users getting lost
- Separate URLs for pages for ease of reference
- Positive effect on SEO, easier for crawlers to navigate

Activity: Technical Debate - Picking a Pagination Module



Throughout your career, you will have many technical debates with your teams. We're going to practice that here:

1. Compare and contrast these modules, list their pros and cons, and decide which one you would use and why.

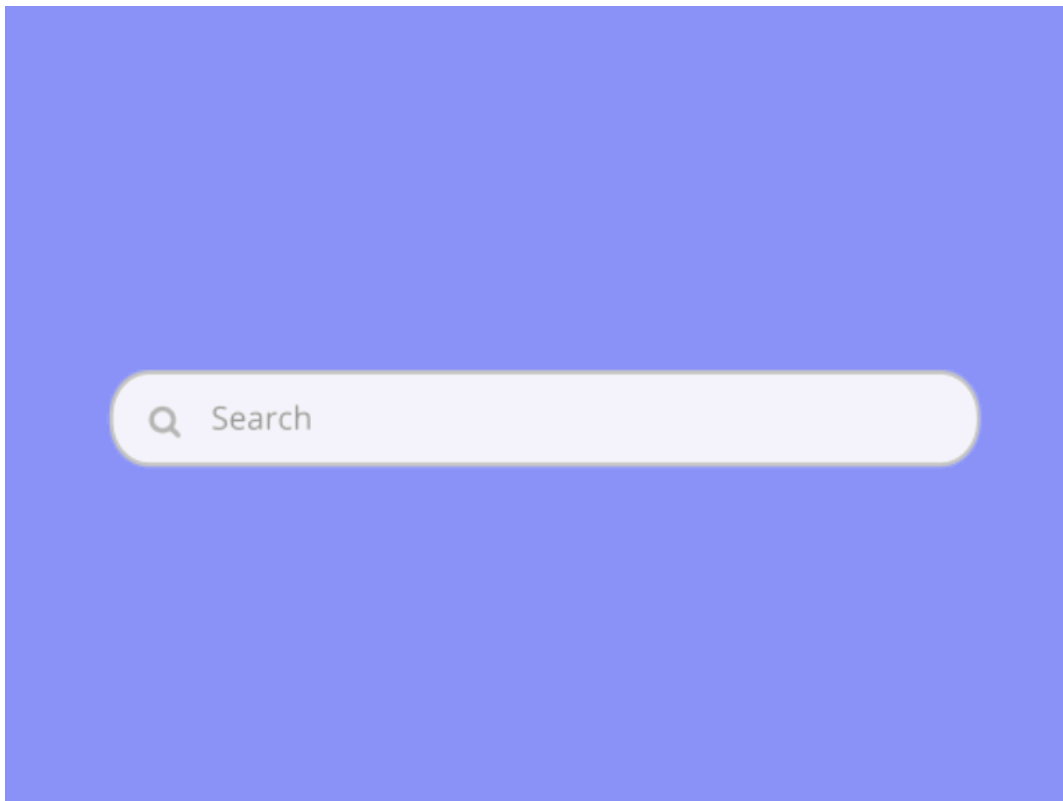
- [paginate](#)
 - [mongoose-paginate](#)
 - [express-paginate](#)
2. Divide into groups where everyone in the group agrees on which package they would use
 3. Now split your group into thirds, one third stay stays, the other two thirds go to another group and try to convince them to use the package you chose
 4. Could you convince anyone to change? What arguments are the most compelling for people? What arguments were most convincing?

Pagination Activity

TODO: add small coding activity for pagination. Something like this: <https://medium.com/javascript-in-plain-english/simple-pagination-with-mongoose-4942af479ab2>

[10m] 🌴 BREAK

Search



Think of how many websites you visit where a **Search Form** or an **Autocomplete** dropdown exists. This is a common web pattern. A **Simple Search** is a search based on the text of one or a few attributes, e.g. on words in a title or body of articles or commercial products. We're going to look at an implementation of Simple Search for Mongoose using RegEx's. (The SQL implementation uses the SCHEMA INDEX). Once we can search, we can then *paginate* the responses!

Simple Search

Sequelize



Sequelize

In order to conduct a simple search, we will use Sequelize's **query operators** **like** or **iLike** (PG only). These are basically Postgres databases only.

Activity

Go back and read through the [Sequelize Querying doc](#), and answer the following:

1. How would you use the **like** operator?
2. Write a query that would find all objects with a **title** of **dog**

Mongoose



elegant **mongodb** object modeling for **node.js**

For mongoose, we can search by passing a Regex (regular expression) for the term we want to search for.

```
User.find({ name: /john/i }, (err, docs) => { });
```

Remember to use the **RegExp** object in JavaScript to turn a string into a Regex pattern.

```
regex = new RegExp(`${req.query.term}/i`);
User.find({ name: regex }, (err, docs) => { });
```

RegEx

/regex/

Need a refresher? Read up on [this guide to RegEx](#), and then write RegExs for the following:

1. Return anything that has the letter “x”
2. Return anything that contains “ar”

Stretch Challenges

1. Return anything that starts with “the”
2. Return anything that ends with “ed”

Make sure to run it against test inputs!

Search Activity

TODO: a search activity, maybe something like this tutorial: <https://kb.objectrocket.com/mongo-db/mongoose-partial-text->