# Products Grid

This is an ecommerce site, where you can buy all sorts of ascii faces like `(□□□□)□` and `¯_(□)_/¯`, in a wide variety of font sizes. The homepage should display a list of products for people to browse.

Please read the instructions and FAQ below before beginning.

## Features

- products are displayed in a grid.
- give the user an option to sort the products in ascending order. Can sort by "size", "price" or "id". The products list should be reloaded when a new sorting option is chosen.
- each product has :
  - a "size" field, which is the font-size (in pixels). We should display the faces in their correct size, to give customers a realistic impression of what they're buying.
  - a "price" field, in cents. This should be formatted as dollars like `$3.51`.
  - a "date" field, which is the date the product was added to the catalog. Dates should be displayed in relative time (eg. "3 days ago") unless they are older than 1 week, in which case the full date should be displayed.
- the product grid should automatically load more items as you scroll down.
- display an animated "loading..." message while the user waits for the data to load.
- to improve the user's experience, we should always pre-emptively fetch the next batch of results in advance, making use of idle-time. But they still should not be displayed until the user has scrolled to the bottom of the product grid.
- when the user reaches the end and there are no more products to display, show the message "~ end of catalogue ~".

### Ads features

- after every 20 products we need to insert an advertisement from one of our sponsors. Use the same markup as the advertisement in the header shown in `public/index/html`, but make sure the `?r` query param is randomly generated each time an ad is displayed.
- Ads should be randomly selected, but a user must never see the same ad twice in a row.

## Products API

- The basic query looks like this: `/api/products`
- The response format is JSON.
- To paginate results use the `_page` parameter, eg: `/api/products?_page=10&_limit=15` (returns 15 results starting from the 10th page).
- To sort results use the `_sort` parameter, eg: `/api/products?_sort=price`. Valid sort values are `price`, `size` and `id`.

## FAQ

### How do I start the app?

Start with `npm start`. The server will look for any files you add to the `public/` directory.

### What libraries/frameworks, packages, tools can I use?

You need to use React.js as the main js library and Git for version control, but other than that you are free to work with any package of your choice with two exceptions - you should not use any package/plugin for the *loading of products on scroll* and *formatting of dates*. We want to see how you solve problems by writing your own JS code, so using any package/plugin for the above two features will immediately make us reject your submission.

### What about sort order (ascending / descending)?

We don't need to worry about alternate sort order for this project, we'll just use ascending-order for everything.

### Can I make changes to the backend or API?

No, your final solution should not include any changes to the server code.

### What should I do when I'm finished?

Please use Git and make regular commits while working on this app. After you have finished your work upload your app to a Github repo, and then send us an email with the link to that repo along with information about which features you have included in your solution, and provide clear instructions on how we can run your code and see the results in a browser.

### How is the exam graded?

We are looking for idiomatic use of javascript, and the ability to solve the problems with code that is clean and easy to read. Even though it's very small in scope, please show us how you would use the language and conventions to structure things in a clear and maintainable way. Please don't go overboard with using external packages/plugins and don't try to introduce extra complexity in your code just for the sake of showcasing your skills.

Try to create a simple and elegant UI for this. You are free to use SASS or any other CSS pre-processor of your choice, but plain CSS is fine too.

### This looks like it will take a while and I'm pretty busy

You're right! With something open-ended like this you could easily spend a week polishing and getting it just right. We don't expect you to do this, and we'll do our best to make sure you're not disadvantaged by this.

When we grade this exam we're not giving you a "score out of 100" for how many features you complete. We're trying to get some insight into your process, to see the way you work. So, by all means, spend more time if you want to. But you are also free to leave certain features out and give a written explanation of how you would approach it. The best approach is to spend your time on the features that you think is the best way to show us your strengths and experience.