Kilo Coding Challenge

The Enchantments is a popular backpacking area in Washington state that requires permits. To get permits for the summer, there is a lottery system. A huge amount of people enter the lottery for the chance to get only a few available permits. Luckily, the agency publishes the previous years permit statistics data which I have ingested into an API for your convenience. See https://www.fs.usda.gov/detail/okawen/passes-permits/recreation/?cid=fsbdev3_053607 for more details.

The goal of the project (if you had 100+ hours) is ultimately to determine, within a set of criteria of dates and zones, the right mix of conditions to apply based on the prior data that will be most likely to result in winning the lottery. Of course, I certainly don't expect you to do that in the time allotted! Instead you will write a front-end application to consume, filter, process, and display data from this API to provide valuable insights to the user.

Please feel free to use (or not) any libraries, frameworks, search engines, boilerplate, stack overflow, fancy IDE, google, and any other tools that you would normally have access to in your day job. I just ask that you submit your original work, though copying snippets of code or using boilerplate you've previously used / created is OK. You can write your app in one file or multiple files, whichever you'd prefer. You can make it as simple or as complex as you'd like, though we're not necessarily looking for you to use more complex technologies / libraries. Simple and straight javascript, or jquery, etc is also completely ok. We're not looking for styling, though if you have and want to use a UI library or have boilerplate for the components we're asking for, feel free to use it. Ultimately, focus on functionality and know that someone else will be reading your code.

The API is available at https://data.javin.io:5000/api/. There are 3 GET endpoints that support query parameters: zones, awards, and applications. Notes on the data model:

3 entities exist in the API:

- Zones: the places you can go which require a permit to be awarded
- Applications: groups applying for permits for a particular zone(s)/date
- Awards: awarded permits for a particular group / zone / date; only a subset of applications results in awards

Details:

- Each application can have up to 3 zones, linked by foreign key into the Zones table
- Each award only has one zone, and also links by foreign key into the application that was awarded

Check out the following endpoints.

- https://data.javin.io:5000/api/zones
- https://data.javin.io:5000/api/applications (award id will be -1 if application was unsuccessful)
- https://data.javin.io:5000/api/awards
- https://data.javin.io:5000/api/zones/1 (get a zone by id)
- https://data.javin.io:5000/api/applications/1 (get an application by id)
- https://data.javin.io:5000/api/awards/1 (get an award by id)
- https://data.javin.io:5000/api/awards?date=2020-07-20&zone=5 (matches any award for zone 5 on date - you may specify zone and date independently - date in YYYY-MM-DD format)
- https://data.javin.io:5000/api/applications?zone=1&date=2020-08-05 (matches any choice of zone 1 and date - you may specify zone and date independently - date in YYYY-MM-DD format)

The data will be in the 'data' field. The apis that return a lot of records have pagination. The 'pagination' field will contain the total number of pages and records in your query and may be somewhat compatible with table pagination or infinite scroll libraries. The max page size is 100, and will be reduced to 100 if you specify something larger. Also of note, I quickly wrote this API and there could be (are likely) bugs - contact me at 415.264.0616 if you find one you can't work around.

Here are some suggestions of how you might show off your skills. Your goal is to demonstrate some working code at the end of 3 hours which shows you can display / manipulate the API and it's data. It's pretty open ended, and successful candidates might finish two or three of these suggestions in the allotted time. The point is to produce some working code that we can later discuss, and be able to articulate how you would attack the rest.

- Retrieve the zones when your app loads and display them
- Create a form where you can select one of the zones (or multiple)
- Fetch and display awards data for the selected zone
- Make the zone selection and data fetch reactive - that is, if you change the zone selection, update the results without having to "submit" a form or reload the page

- Paginate the awards data, so that I can navigate through multiple pages (next / previous page buttons or infinite scroll)
- Create a search box that I can enter a date and filter the award results. Maybe make a second search box so I can search between 2 dates.
- Validate the data for the search box and display a toast if a valid date is input

- Make a button that would clear the search form (uncheck zones, clear dates / filters)
- Make an award clickable, and from an award view the applications that were submitted for that zone / date

Once 3 hours have passed, tag your repo wherever you are (optional, if using github) and email me a zip or tgz archive of your repository along with the (optional) repo link. Again I'm not expecting you to even come close to finishing the above, and whatever you've done in 3 hours is certainly enough. I'd like the code you send me to be working though, and tell me how far you got, and if/where you got stuck. If you enjoyed the problem and want to extend it after the time expires, feel free to do so but there is certainly no requirement to. At the debrief we will talk through how you would approach any tasks that were remaining.