

Flickr API Coding Challenge

Please write an Angular App that shows data from the Flickr API.

The app shall have following features:

- You will use the Flickr photo search API (flickr.photos.search) to search for photos with a specific tag, you will find the documentation [here](#). You can also play with the API in the browser [here](#) where you will also see example API calls (at the bottom of the page) with a necessary api key (the param api_sig can be left out).
- The App will have two pages, one start page where you can add tags that shall be looked up on Flickr and one detail page that loads/shows all the photos for this tag
- Please provide a simple and clean UI, nothing fancy. The functionality and efficient code is more important. Please use Twitter Bootstrap and Font Awesome as basic framework to help you structure the UI
- The start page will show a simple form at the top where the user can enter a search tag (mandatory field). Optionally one can also enter a user id. The form needs to be validated and error messages shall be shown if the validation fails. The form submits on enter and there is also a submit button. Additionally there is a cancel button that will clear the form.
- On form submit an API call to the Flickr API is made to load the most interesting photo for this tag. Following params shall be passed to the call:
 - tags: the tag that has been entered by the user
 - user_id: the user id that has been added by the user. If none is provided, this param shall be left out
 - sort: interestingness-desc
 - per_page: 1
 - extras: date_upload, date_taken, owner_name, views, url_q
- If the response is empty or an error occurred, there shall be an error message. If the request is successful, **the response data for this one photo is saved** (in memory, no persistent storage necessary) and displayed in the frontend. The form is then reset. Every subsequent form submit will add another data object to your array with the saved tag searches
- This data array shall be displayed underneath the form. For every entry there shall be a square that visualizes the stored data (like a tile or panel):
 - The square image (url_q)
 - The owners name
 - The date when the photo was uploaded
 - The date when the photo was taken
 - The number of views
- Above the squares (and underneath the form) there are some buttons for the sorting of the tiles. Please use the sorting features provided by angular. One can sort by:
 - Number of views (default)
 - Date when the photo was taken
 - Date when the photo was uploaded
 - The owners name

- When clicking on one of these tiles the second page opens. This page shall load and show **all the photos** that are found for this tag and user (if user id was set). This means that another API request has to be made on page load. This time with some other params. Please choose the appropriate params yourself
- On this page only the photos are displayed without any additional information.
- There shall be a pagination feature. You can freely choose an appropriate number of items per page. The Flickr API provides all required information to implement the pagination. Please add a pagination item below the photos that will provide buttons to load the other pages, including the first and last.

Please also keep this in mind:

- Please use Angular 1.5 or 2.0 and prefer to use components. You can use ES5, newer Versions of Javascript and/or type extension like Typescript or Flow
- It is perfectly fine to use an existing github repo/starter kit, so you can get started quickly. But make sure that you remove unnecessary code, dependencies and settings to not bloat your repo too much
- Create a public Github repo for the code. Commit your progress with good commit message and in sensible chunks
- Test the code in a way that seems appropriate to you in a professional environment
- Use standards and best practices as best as you can
- Pretend you have a big team and make it really easy for them to run the app and to maintain it for years to come. It is usually helpful to use some tool for task automation. Also think about the architecture and reusable components
- Please focus on functionality first and a clean coding style and worry about design later. The app does not need to be responsive but Bootstrap makes it very easy to build a responsive app if you think about it from the start. Try to incorporate this when you build the grid
- If you write custom css, think about the structure and reusability. Feel free to use a css preprocessor

Happy coding :-D