Ionic 2
Making REST HTTP requests

Simple but Bad Way

```
import { Http } from '@angular/http';
constructor(…public http: Http ) { ... }
getResults() {
    results: any;
    this.http.get(“URL”).subscribe (
            response=> results = response.json()
        );
    return results;
}
```

Create a Provider (Service)

- *ionic g provider MyProvider*
  - creates an *@injectable* class in providers/my-provider/my-provider.ts
- <my-provider>.ts

```
import { Injectable } from '@angular/core';
import { Http } from '@angular/http';

@Injectable()
export class MyProvider {
  constructor(private http: Http) {}
}
```

Make a query

- this.http.get(URL)
  .subscribe(
  response=> { this.result=response.json() }
  err => { <error handling> }
  () => { })

Get Observable

Get JSON

- Provider can return either
  - Observable
    - return this.http.get(URL)
    - Your component should subscribe to the object
  - ~~JSON data~~
    - return this.http.get(URL).subscribe(...); return result;
    - DO NO DO THIS !!

Using MyProvider

- Import
  - *import {MyProvider} from '…'*
- Add as provider
  - *@Component ({*
    templateURL: '…',
    providers: [MyProvider]
    })
- Inject, remember
  - *construct(public myprovider: MyProvider) {}*
- Access data
  - user whatever the provider returns

MovieService provider

- URL
  - http://netflixroulette.net/api/api.php?actor=<actor-name>
- getMoviesByActor(actor_name) {}
  - returns an Observable object

SearchBar

- <ion-searchbar
  [(ngModel)] = 'keyword'
  (ionInput) = 'onKeypress($event)'
  showCancelButton = 'true'>
  </ion-searchbar>

bind input value with a variable

bind an event handler to input-change event. This handles cancel event too

show cancel button
if false, only shows (x) button

Note on searchbar event handling

- Use (ionInput) event instead of (keyup) event
  - If use (keyup), the event handler fires before ngModel binding happens
    - e.g., inside handler onKeyPressed(), this.keyword variable does not reflect the current value of the searchbar. Instead, event.target.value should be used.
  - If use (ionInput), the event is fired after ngModel binding is finished. So, inside event handler, keyword is properly set. Therefore, with ionic 2, use this event.

How to filter the result

1. Prepare two member variables
    1. items_original: store original movie list obtained from a web query
    2. items: store a filtered list based on the searchbar text input
2. Catch the (ionInput) event
    1. in the handler, update this.items by filtering this.items_original using a filter method of an array
        1. <Array>.filter(x => <a condition about x>)
            1. This generates a new array that contains elements of the original array, which makes the condition true
3. Then, the list will be filtered everytime the user changes the searchbar text.

How to catch key event

- (ionInput) event
  - fires everytime the user changes the input.
  - But does not fire for Enter key
- (search) event
  - fires everytime the user presses Enter key
- In a searchbar, you can catch both the events by two separate handlers
  - <ion-searchbar (ionInput)='keyPressed($event)" (search)='enterPressed($event)'></ion-searchbar>

See it in action

- Check the app in action
  - app id: **62cf3237**