

# Angular Js RxJs Operators



# Agenda

- **Observables**
- **Operators**
  - interval
  - take
  - map
  - mergeMap
  - switchMap
- **fromEvent**
- **inputEvent**
- **Reddit Search Example**

# Operators

- **While using async pipe in angular, no need to subscribe.**
- **Then how do we wrap values and use.**
- **Operators are used to modify an observable midstream, without unwrapping it**

# Interval & take Operators

- **Interval**

- Millisecond interval

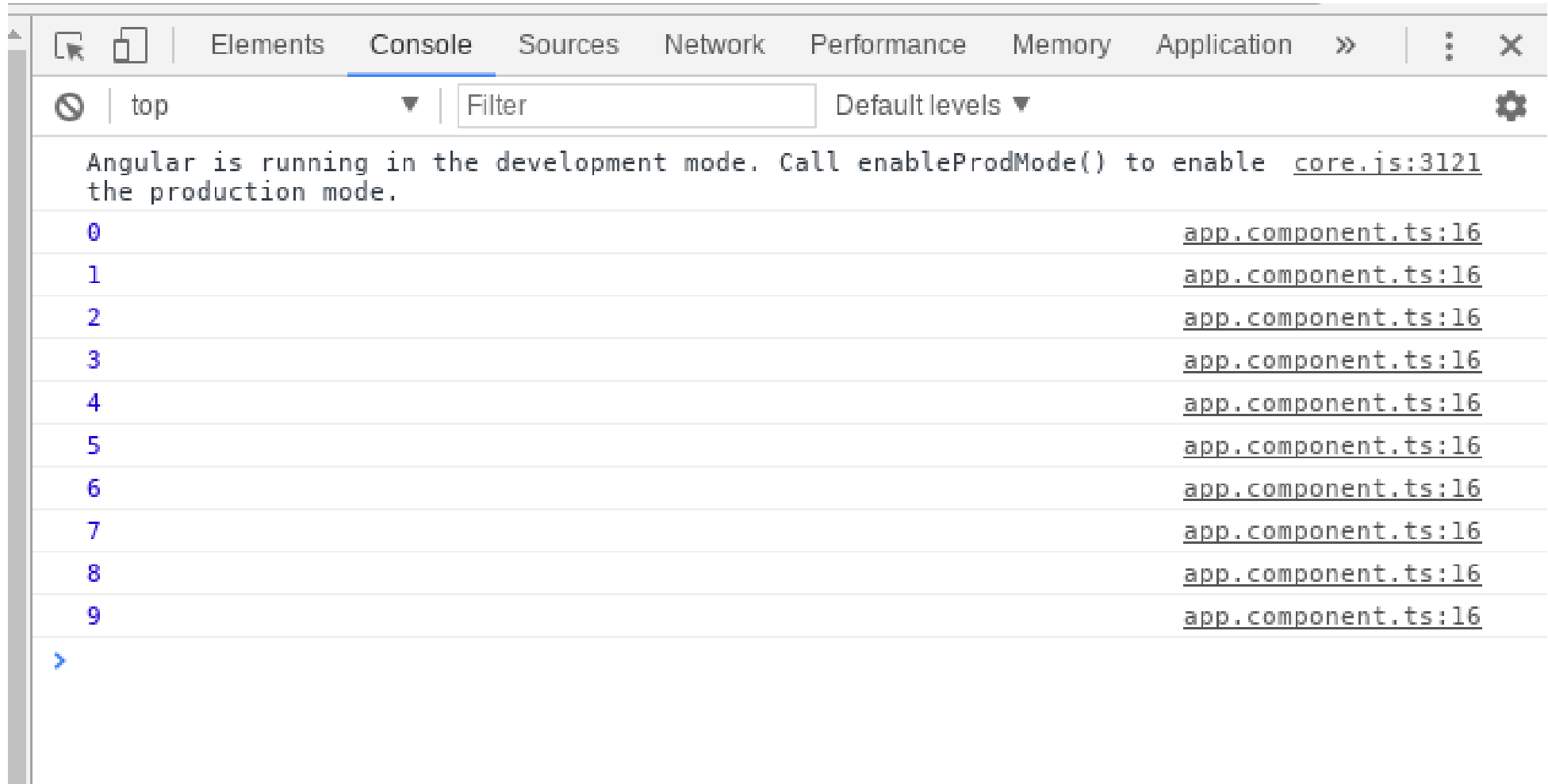
- **Pipe**

- allows chaining multiple pipeable operators.

- **Take**

- When we are interested in only the first set number of emission
  - Take first 10 values of observable from interval and stop
  - Where did user click first on when he/she first entered the page
    - `const numbers$ = interval(1000).pipe(take(10));`

# Interval & take



The screenshot shows a web browser's developer console with the 'Console' tab selected. At the top, there is a toolbar with icons for Elements, Console, Sources, Network, Performance, Memory, and Application. Below the toolbar, the console displays a message: 'Angular is running in the development mode. Call enableProdMode() to enable the production mode.' followed by a series of log entries. The log entries are numbered 0 through 9, and each entry shows the file path 'app.component.ts:16'. The console also features a 'Filter' input field and a 'Default levels' dropdown menu.

```
Angular is running in the development mode. Call enableProdMode() to enable the production mode.
0 app.component.ts:16
1 app.component.ts:16
2 app.component.ts:16
3 app.component.ts:16
4 app.component.ts:16
5 app.component.ts:16
6 app.component.ts:16
7 app.component.ts:16
8 app.component.ts:16
9 app.component.ts:16
>
```

# Map Operator

- **Map**

- Use map if you want to access value coming from subscribe and use it
- Take the payload and map to another value if you want to
  - `const numbers$ = interval(1000)`
  - `.pipe(`
  - `take(10),`
  - `map((x) => x * x)`
  - `);`

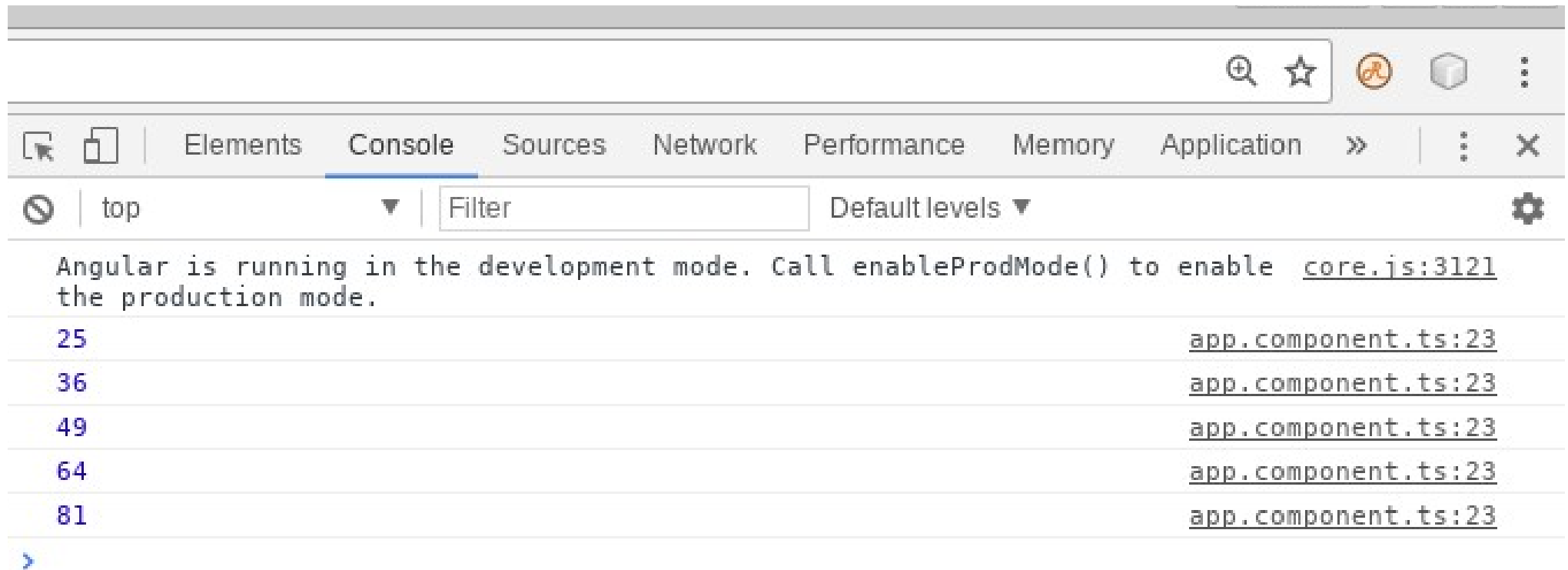
# Filter operator

- **Filter**

- Filters result base on function

- `const numbers$ = interval(1000)`
    - `.pipe(`
    - `take(10),`
    - `map((x) => x * x),`
    - `filter(x => x>20)`
    - `);`

# filter



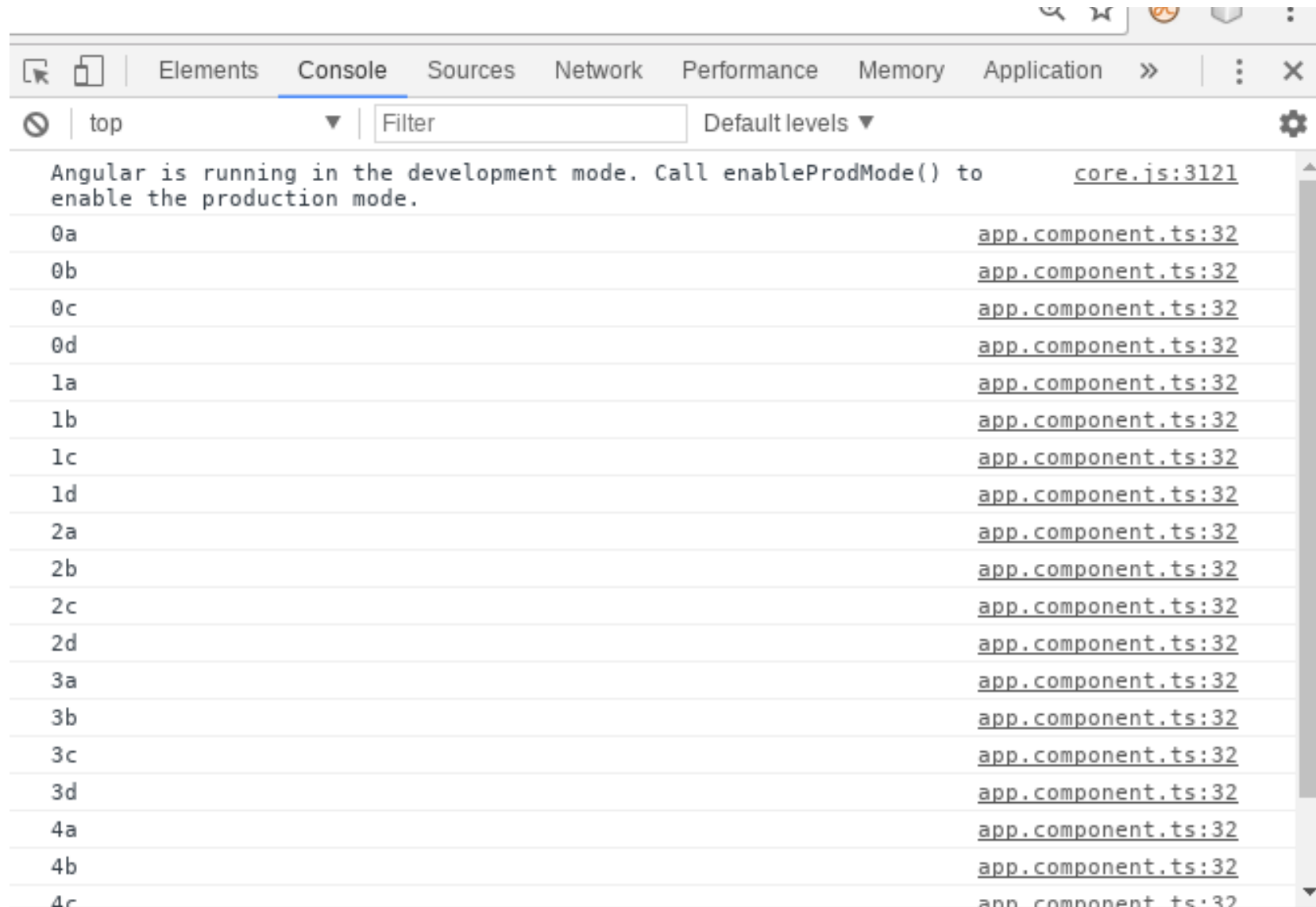


# mergeMap

- **MergeMap**

- allows for multiple inner subscriptions to be active at a time.
- When we do not want to cancel request
- `const numbers$ = interval(1000).pipe(take(5));`
- `const letters$ = of('a','b','c','d');`
- `letters$.pipe(mergeMap(x =>`
- `numbers$.pipe(take(5),map(i => i+x))`
- `)).subscribe(x => console.log(x));`

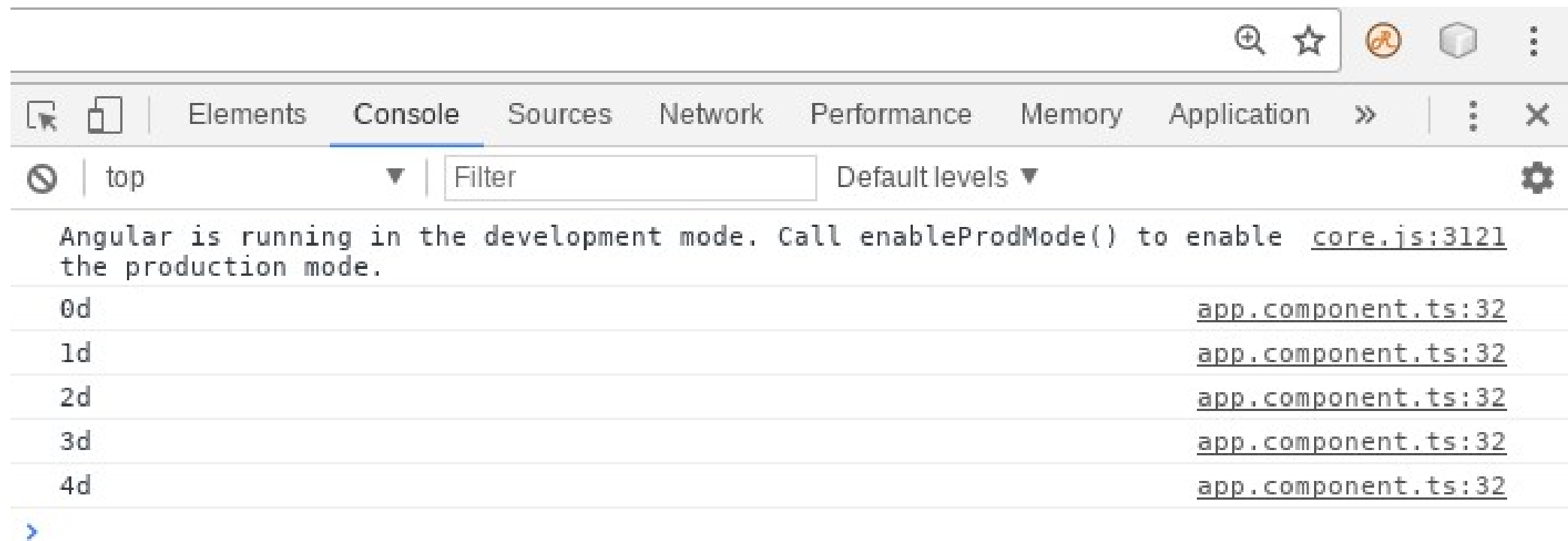
# mergeMap



# switchMap

- **SwitchMap**
- **Useful when we want to cancel request with new one**
- **Example-Search string change while waiting for query to bring result from database.**

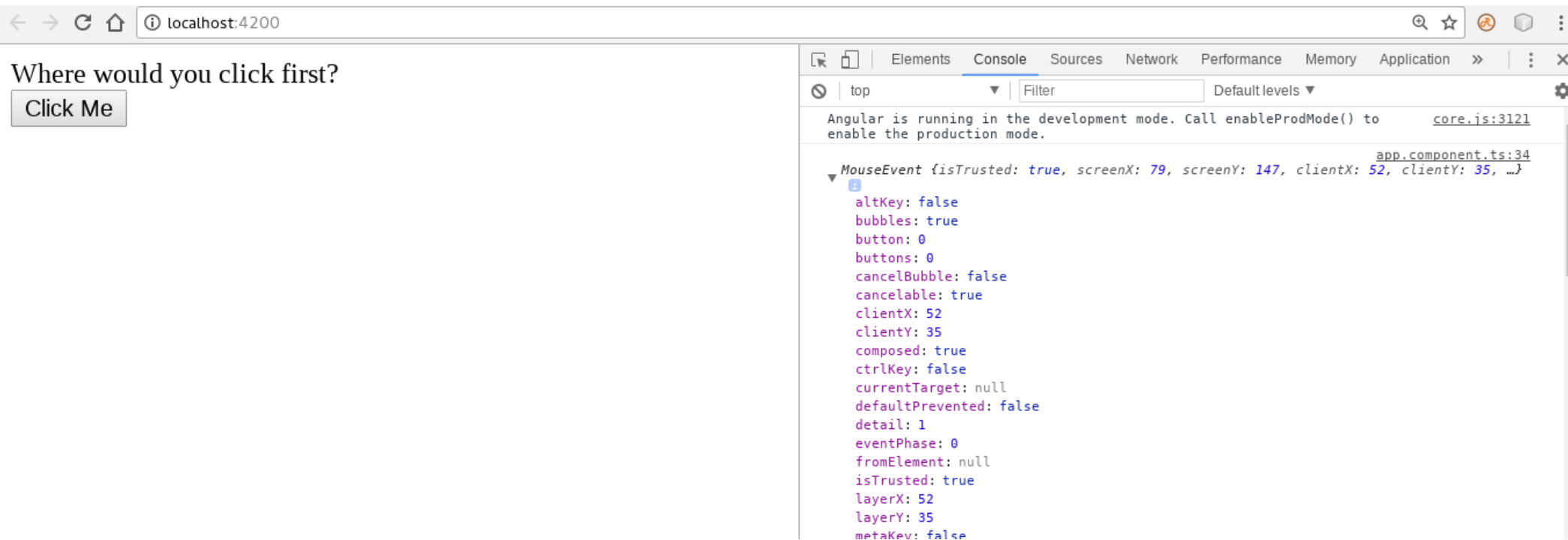
# switchMap



# fromEvent

- **Turn event into observable sequence.**
  - `const oneClickEvent = fromEvent(document, 'click').pipe().subscribe(x => console.log(x));`

# fromEvent



The screenshot shows a web browser at `localhost:4200` with the text "Where would you click first?" and a button labeled "Click Me". The browser's developer console is open, displaying a message from Angular about production mode and a detailed log of a `MouseEvent` object. The event object contains the following properties:

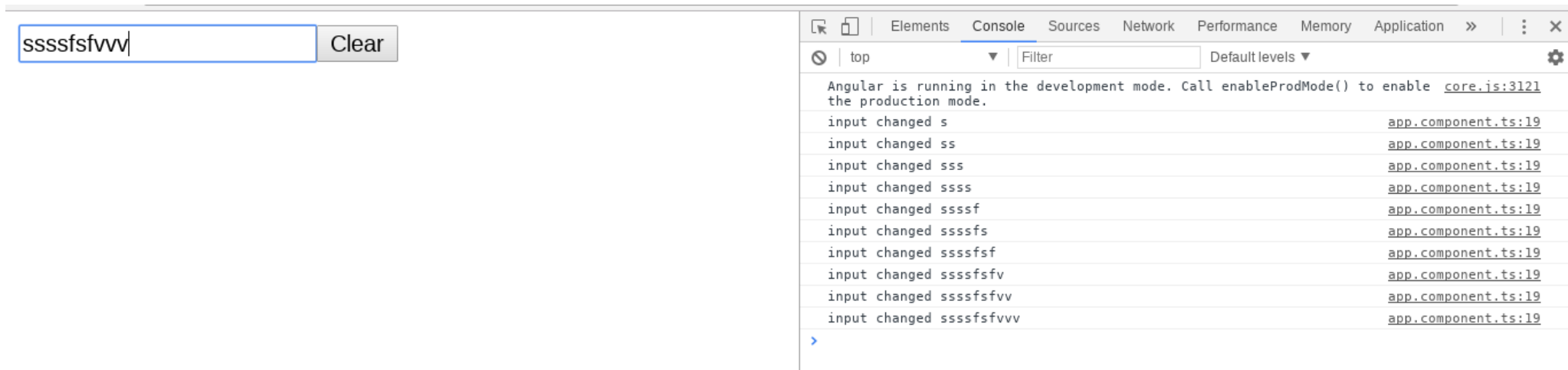
```
MouseEvent {isTrusted: true, screenX: 79, screenY: 147, clientX: 52, clientY: 35, ...}
  altKey: false
  bubbles: true
  button: 0
  buttons: 0
  cancelBubble: false
  cancelable: true
  clientX: 52
  clientY: 35
  composed: true
  ctrlKey: false
  currentTarget: null
  defaultPrevented: false
  detail: 1
  eventPhase: 0
  fromElement: null
  isTrusted: true
  layerX: 52
  layerY: 35
  metaKey: false
```

# Input Event

- **Capturing Event**

- `<input type="text" [(ngModel)]="searchString" (ngModelChange)="inputChanged($event)" placeholder="Search">`
- `<button (click)="searchString = ''>Clear</button>`
- 
- `inputChanged($event) {`
- `console.log('input changed', $event);`
- `}`

# Input Event



The screenshot displays a web application interface on the left and a browser's developer console on the right. The interface consists of a text input field containing the string "sssfsvv" and a "Clear" button to its right. The developer console, located on the right side, shows a series of log messages. The first log is a warning from Angular: "Angular is running in the development mode. Call enableProdMode() to enable the production mode." This is followed by ten log entries, each starting with "input changed" and the current value of the input field, ranging from "s" to "sssfsvv". All log entries are attributed to the source "app.component.ts:19".

sssfsvv Clear

Angular is running in the development mode. Call enableProdMode() to enable the production mode.

Log Message	Source
input changed s	app.component.ts:19
input changed ss	app.component.ts:19
input changed sss	app.component.ts:19
input changed ssss	app.component.ts:19
input changed ssssf	app.component.ts:19
input changed ssssfv	app.component.ts:19
input changed ssssfsv	app.component.ts:19
input changed ssssfsvv	app.component.ts:19
input changed ssssfsvvv	app.component.ts:19

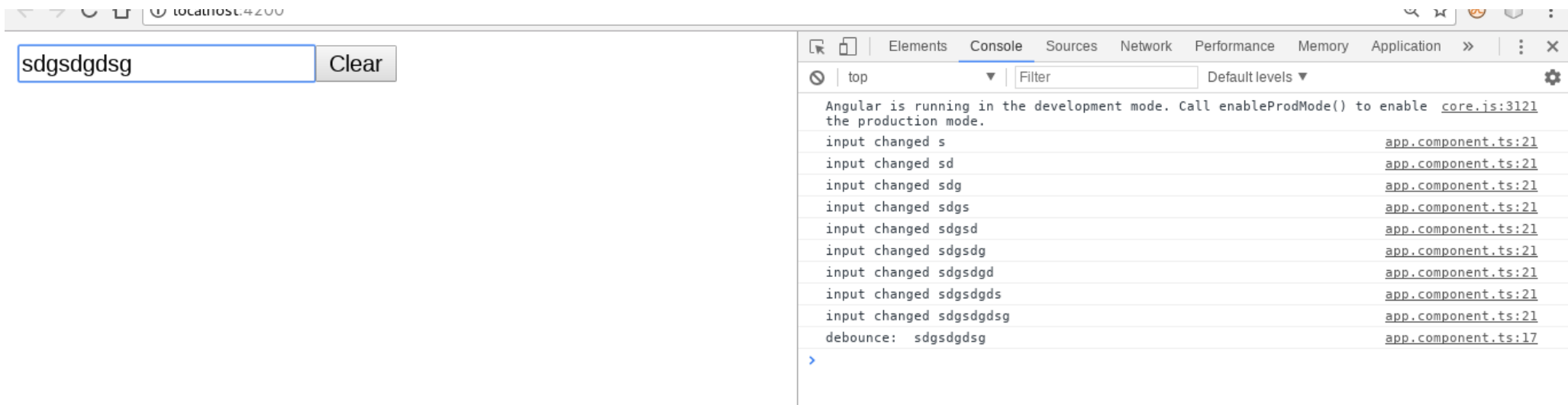


# debounceTime

- **No need to ping database ver and over again, use debounce item instead**

- searchSubject\$ = new Subject<string>();
- 
- ngOnInit() {
- this.searchSubject\$.pipe(debounceTime(200)).subscribe(x=> console.log('debounce: ',x));
- }
- 
- inputChanged(\$event) {
- console.log('input changed', \$event);
- this.searchSubject\$.next(\$event);
- }

# Debounce



The screenshot shows a web browser window with a text input field containing the text "sdgsdgdsg" and a "Clear" button. The browser's developer console is open, displaying a series of log messages. The messages show the input changing from "s" to "sdgsdgdsg" in rapid succession, followed by a "debounce" message. The console also shows a warning message from Angular about running in development mode.

Message	Source
Angular is running in the development mode. Call enableProdMode() to enable the production mode.	core.js:3121
input changed s	app.component.ts:21
input changed sd	app.component.ts:21
input changed sdg	app.component.ts:21
input changed sdgs	app.component.ts:21
input changed sdgsd	app.component.ts:21
input changed sdgsdgd	app.component.ts:21
input changed sdgsdgdg	app.component.ts:21
input changed sdgsdgdsg	app.component.ts:21
debounce: sdgsdgdsg	app.component.ts:17

# Async Pipe





- **Async pipe**
- **Allows an observable to send to DOM without using subscribe and wrap it.**
- **Async pipe automatically subscribe and unsubscribe when user leaves the page.**

# Reddit Search

```
constructor(private http: HttpClient){  
  searchSubject$ = new Subject<string>();  
  results$: Observable<any>;  
  
  ngOnInit() {  
    this.results$ = this.searchSubject$.pipe(debounceTime(400),  
      distinctUntilChanged(),  
      tap(x => console.log('do', x)),  
      switchMap(searchString => this.queryAPI(searchString))  
    );  
  
  }  
  
  queryAPI(searchString){  
    console.log('queryAPI', searchString);  
    return this.http.get(`https://www.reddit.com/r/aww/search.json?q=${searchString}`)  
      .pipe(map(result => result['data']['children']));  
  }  
  
  inputChanged($event) {  
    console.log('input changed', $event);  
    this.searchSubject$.next($event);  
  }  
  ngOnDestroy() {  
  }  
}
```

# Query String with reddit

Clear



top

Angular is running in the development mode. Call enableProdMode() to enable the production mode. [core.js:3121](#)

input changed d [app.component.ts:47](#)

input changed do [app.component.ts:47](#)

input changed dog [app.component.ts:47](#)

queryAPI dog [app.component.ts:41](#)

Console

What's New

Search

Highlights from the Chrome 60 update

New Audits panel, powered by Lighthouse  
Find out whether your site qualifies as a Progressive Web App, measure the accessibility and performance of a page, and discover best practices.

Third-party badges  
See what third-party entities are logging to the Console, making network requests, and causing work during performance recordings.

Progressive Web App

45

Performance

34

Rxjs - Google Ch...

prave@localhost...

app

app

[2 3.3-AJS6-RxJ...

app.component.t...

[prave@localhos...

3. Wrap an API...

1 / 4

5

# Resources

- [\*\*https://angular.io/\*\*](https://angular.io/)
- [\*\*https://rxjs-dev.firebaseapp.com/\*\*](https://rxjs-dev.firebaseapp.com/)
- [\*\*http://reactivex.io/\*\*](http://reactivex.io/)
- [\*\*https://www.learnrxjs.io\*\*](https://www.learnrxjs.io)

