

Custom Operators

7. Custom Operators

Operators

- Since Operators are simply pure functions you can easily create an operator of your own
- You can create your own operator that creates an Observable
- You can create your own operator that manipulates an Observable
- Whenever you see that you are copy pasting common operator code, it is probably time to encapsulate it in a custom operator instead

Create operator ex

- In the following ex you will have to create an operator that creates an Observable from a web socket
- The web socket server is located at the following url:
 - ▶ <https://academeez-websocket.herokuapp.com>

```
function fromWebSocket(socket: WebSocket): Observable<MessageEvent>
```

Custom manipulation operators

- You need to create a function which returns a function which gets the original observable and returns a new observable
- You can either subscribe to the original observable and create your own custom login
- But usually you will just want to combine existing operators together, in that case you can import the **pipe** function and return it

```
// custom operator for throttling
function throttling(time: number) {
  return pipe(
    map(event => (event as any).target.value),
    debounceTime(time),
    distinctUntilChanged()
  )
}
```

Summary

- Operators are extendable
- Whenever you are repeating your rxjs code you can often create a custom operator to avoid the code duplication

Thank You

Next Lesson: 8. Dealing with Errors