



```
const { initPublisher, initSubscriber } = (function () {
  const store = new Map();

  function generateUniqueId() {
    const keys = '123456ABCDEF';
    let id = '';
    for (let i = 0; i < 6; i++) {
      const index = Math.floor(Math.random() * keys.length);
      id += keys[index];
    }
    return id;
  }

  const initPublisher =
    (topic = 'global') =>
    (value = null) => {
      const registeredCallbacks = store.get(topic);
      if (registeredCallbacks) {
        [...registeredCallbacks.values()].forEach((cb) =>
          queueMicrotask(() => cb(value))
        );
      }
    };

  const initSubscriber = (topic = 'global') => {
    const funcIds = [];

    return {
      subscribe(callback) {
        if (typeof callback !== 'function') return;
        if (!store.has(topic)) store.set(topic, new Map());

        const funcId = generateUniqueId();
        store.get(topic).set(funcId, callback);
        funcIds.push(funcId);
      },
      unsubscribe() {
        const registeredCallbacks = store.get(topic);
        if (registeredCallbacks && funcIds.length) {
          funcIds.forEach((id) => registeredCallbacks.delete(id));
          funcIds.length = 0;
          if (!registeredCallbacks.size) store.delete(topic);
        }
      },
    };
  };

  return { initPublisher, initSubscriber };
})();
```




```
const publisher1 = initPublisher('topic1');
const subscriber1 = initSubscriber('topic1');

const publisher2 = initPublisher('topic2');
const subscriber2 = initSubscriber('topic2');

subscriber1.subscribe((value) => {
  console.log(`subscriber1: value => ${value}`);
});

subscriber2.subscribe((value) => {
  console.log(`subscriber2: value => ${value}`);
});

publisher1(100);
publisher2(100);

subscriber1.unsubscribe();

publisher1(200);
publisher2(200);

subscriber2.unsubscribe();

publisher1(300);
publisher2(300);
```

-----OUTPUT-----

```
subscriber1: value => 100
subscriber2: value => 100
```

```
// subscriber1 unsubscribed
```

```
subscriber2: value => 200
```

```
// subscriber2 unsubscribed
```

```
// nothing gets logged as no subscriber is available to consume published data
```