


```
let isRequesting = false;
let requestCount = 0;

const scrollHandler = function(event) {
  const DOM = event.target;
  if(hasScrolled(DOM) > 0.9 && !isRequesting) {
    isRequesting = true;
    fetch('url...')
      .then(res => {
        // do something change view
        isRequesting = false;
        requestCount = requestCount + 1;
        if (requestCount === 3) {
          scrollView.removeEventListener('scroll', scrollHandler)
        }
      })
  }
}

scrollView.addEventListener('scroll', scrollHandler);
```

```
fromEvent(scrollView, 'scroll')
    .pipe(
        map(event => event.target),
        map(hasScrolled),
        filter(p => p > 0.9),
        exhaustMap(() => ajax('url...')),
        take(3)
    )
    .subscribe(res => {
        // do something change view
    })
```

vanilla

auds

RxUs


```
let isRequesting = false;
let requestCount = 0;

const scrollHandler = function(event) {
  const DOM = event.target;
  if(hasScrolled(DOM) > 0.9 && !isRequesting) {
    isRequesting = true;
    fetch('url...')
      .then(res => {
        // do something change view
        isRequesting = false;
        requestCount = requestCount + 1;
        if (requestCount === 3) {
          scrollView.removeEventListener('scroll', scrollHandler)
        }
      })
  }
}

scrollView.addEventListener('scroll', scrollHandler);
```

VanillaJS

```
let isRequesting = false;
let requestCount = 0;

const scrollHandler = function(event) {
  const DOM = event.target;
  if(hasScrolled(DOM) > 0.9 && !isRequesting) {
    isRequesting = true;
    fetch('url...')
      .then(res => {
        // do something change view
        isRequesting = false;
        requestCount = requestCount + 1;
        if (requestCount === 3) {
          scrollView.removeEventListener('scroll', scrollHandler)
        }
      })
  }
}

scrollView.addEventListener('scroll', scrollHandler);
```

RxJS

```
fromEvent(scrollView, 'scroll')
  .pipe(
    map(event => event.target),
    map(hasScrolled),
    filter(p => p > 0.9),
    exhaustMap(() => ajax('url...')),
    take(3)
  )
  .subscribe(res => {
    // do something change view
  })
```

Observable 簡介

