

Pluck

Pluck : Like map, but meant only for picking one of the nested properties of every emitted object.

Consider you have a user details like the following and you want to get the city of user. You can get it using map or pluck operator.

```
src > app > pluck > TS pluck.component.ts > PluckComponent > getNames
30  ngAfterViewInit(): void {
31    fromEvent<any>(this.txt.nativeElement, 'input')
32    .pipe(
33      // map(event => event.target.value),
34      pluck('target', 'value')
35    )
36    .subscribe((val) => console.log(val));
37  }
38
39  getNames(users) {
40    //using map operator
41    // from(users).pipe(
42    //   map(user => user['name']),
43    //   toArray()
44    // ).subscribe(names => {
45    //   console.log('names => ', names);
46    // });
47    //using pluck operator
48    from(users).pipe(
49      pluck('name'),
50      toArray()
51    ).subscribe(names => {
52      console.log('names => ', names);
53    });
54  }
55
56  getCities(users) {
57    from(users).pipe(
58      pluck('address', 'city'),
59      // map(val => val['address']['city']),
60      toArray()
61    ).subscribe(cities => {
```

```
this.http.get('https://jsonplaceholder.typicode.com/users').pipe(
).subscribe((users) => {
console.log('users => ', users);
//users[0].address = null;
this.getNames(users);
this.getCities(users);
});
```

```

users => pluck.component.ts:32
▼ (10) [{...}, {...}, {...}, {...}, {...}, {...}, {...}, {...}, {...}, {...}] ⓘ
  ▼ 0:
    ▶ address: {street: 'Kulas Light', suite: 'Apt. 556', city: 'Gwenborough', zipcode: '92998-3874', geo: ...}
    ▶ company: {name: 'Romaguera-Crona', catchPhrase: 'Multi-layered client-server neural-net', bs: 'harnes...}
    email: "Sincere@april.biz"
    id: 1
    name: "Leanne Graham"
    phone: "1-770-736-8031 x56442"
    username: "Bret"
    website: "hildegard.org"
    ▶ [[Prototype]]: Object
  ▶ 1: {id: 2, name: 'Ervin Howell', username: 'Antonette', email: 'Shanna@melissa.tv', address: {...}, ...}
  ▶ 2: {id: 3, name: 'Clementine Bauch', username: 'Samantha', email: 'Nathan@yesenia.net', address: {...}, ...}
  ▶ 3: {id: 4, name: 'Patricia Lebsack', username: 'Karianne', email: 'Julianne.OConner@kory.org', address: {...}, ...}
  ▶ 4: {id: 5, name: 'Chelsey Dietrich', username: 'Kamren', email: 'Lucio_Hettinger@annie.ca', address: {...}, ...}
  ▶ 5: {id: 6, name: 'Mrs. Dennis Schulist', username: 'Leopoldo_Corkery', email: 'Karley_Dach@jasper.info', address: {...}, ...}
  ▶ 6: {id: 7, name: 'Kurtis Weissnat', username: 'Elwyn.Skiles', email: 'Telly.Hoeger@billy.biz', address: {...}, ...}
  ▶ 7: {id: 8, name: 'Nicholas Runolfsson', username: 'Maxime_Nienow', email: 'Sherwood@rosamond.me', address: {...}, ...}
  ▶ 8: {id: 9, name: 'Glenna Reichert', username: 'Delphine', email: 'Chaim_McDermott@dana.io', address: {...}, ...}
  ▶ 9: {id: 10, name: 'Clementina DuBuque', username: 'Moriah.Stanton', email: 'Rey.Padberg@karina.biz', address: {...}, ...}
  length: 10
  ▶ [[Prototype]]: Array(0)

```

Using map

```

getCities(users) {
  from(users).pipe(
    map(val => val['address']['city']),
    toArray()
  ).subscribe(cities => {
    console.log('cities => ', cities);
  });
}

```

Using pluck

```

getCities(users) {
  from(users).pipe(
    pluck('address', 'city'),
    toArray()
  ).subscribe(cities => {
    console.log('cities => ', cities);
  });
}

```

Both will give same result like the following image

```

cities => pluck.component.ts:64
▼ (10) [undefined, 'Wisokyburgh', 'McKenziehaven', 'South Elvis', 'Roscoeview', 'South Christy', 'Howemout h', 'Aliyaview', 'Bartholomebury', 'Lebsackbury']

```

Now if, user's address value will be null, then map operator will give error but pluck will not give any error, just print "undefined" in subscription value.

```
✖ ERROR TypeError: Cannot read properties of null (reading 'city') core.js:6210
    at MapSubscriber.project (pluck.component.ts:59:32)
    at MapSubscriber._next (map.js:29:1)
    at MapSubscriber.next (Subscriber.js:49:1)
    at Observable._subscribe (subscribeToArray.js:3:1)
    at Observable._trySubscribe (Observable.js:42:1)
    at Observable.subscribe (Observable.js:28:1)
    at MapOperator.call (map.js:16:1)
    at Observable.subscribe (Observable.js:23:1)
    at ScanOperator.call (scan.js:18:1)
    at Observable.subscribe (Observable.js:23:1)
```

```
>
cities =>
  (10) [undefined, 'Wisokyburgh', 'McKenziehaven', 'South Elvis', 'Roscoeview', 'South Christy', 'Howemout
    h', 'Aliyaview', 'Bartholomebury', 'Lebsackbury']
> |
```

Another example of pluck operator

```
fromEvent<any>(this.txt.nativeElement, 'input')
  .pipe(
    map(event => event.target.value)
  )
  .subscribe((val) => console.log(val));
```

Instead of map operator, you can use pluck operator

```
fromEvent<any>(this.txt.nativeElement, 'input')
  .pipe(
    pluck('target', 'value')
  )
  .subscribe((val) => console.log(val));
```