**[rxResource in Angular](https://www.bing.com/ck/a?!&&p=b5752236fcfd9edf4ef72afe8b9554a8d48df5b2a345501edf0bcdc2f56c72ebJmltdHM9MTc1NzExNjgwMA&ptn=3&ver=2&hsh=4&fclid=3f5da990-dec8-6a22-2828-bbf3df306b20&psq=rxresource+angular&u=a1aHR0cHM6Ly9hbmd1bGFyLmRldi9hcGkvY29yZS9yeGpzLWludGVyb3AvcnhSZXNvdXJjZQ&ntb=1" \t "_blank)**

[**1**](https://www.bing.com/ck/a?!&&p=b5752236fcfd9edf4ef72afe8b9554a8d48df5b2a345501edf0bcdc2f56c72ebJmltdHM9MTc1NzExNjgwMA&ptn=3&ver=2&hsh=4&fclid=3f5da990-dec8-6a22-2828-bbf3df306b20&psq=rxresource+angular&u=a1aHR0cHM6Ly9hbmd1bGFyLmRldi9hcGkvY29yZS9yeGpzLWludGVyb3AvcnhSZXNvdXJjZQ&ntb=1)[**2**](https://www.bing.com/ck/a?!&&p=4e82348d93de54e3e450e6e9837cd4792d224147f52d3d84a80b430663084b1dJmltdHM9MTc1NzExNjgwMA&ptn=3&ver=2&hsh=4&fclid=3f5da990-dec8-6a22-2828-bbf3df306b20&psq=rxresource+angular&u=a1aHR0cHM6Ly9kZXYudG8vdGhpcy1pcy1hbmd1bGFyL2FuZ3VsYXItMTktc3RyZWFtbGluaW5nLWRhdGEtcmV0cmlldmFsLXdpdGgtcmVzb3VyY2UtYW5kLXJ4cmVzb3VyY2UtYXBpcy0zbGIy&ntb=1)[**3**](https://www.bing.com/ck/a?!&&p=2afc6d5f3692c7becce03be4ec450771cb6ba5a1f260d44805b52332f5cfad59JmltdHM9MTc1NzExNjgwMA&ptn=3&ver=2&hsh=4&fclid=3f5da990-dec8-6a22-2828-bbf3df306b20&psq=rxresource+angular&u=a1aHR0cHM6Ly9hbmd1bGFyLmRldi9ndWlkZS9zaWduYWxzL3Jlc291cmNl&ntb=1)

**rxResource** is an experimental API in Angular that simplifies asynchronous data retrieval using RxJS Observables. It maps requests to an Observable of the resource's value, considering only the first emission.

**Example**

import { rxResource } from '@angular/core/rxjs-interop';

import { of } from 'rxjs';

const userResource = rxResource({

request: () => ({ id: '123' }),

loader: ({ request }) => of({ id: request.id, name: 'John Doe' })

});

userResource.value().subscribe(user => console.log(user));

**Key Features**

1. **Observable-Based**: Unlike the resource API that uses Promises, rxResource leverages Observables for more flexible and reactive data handling.
2. **First-Value-Only**: The current implementation considers only the first emitted value from the Observable.
3. **Writable Resources**: Allows updating the local state of a resource using Observables.

**Usage Considerations**

* **No abortSignal**: Since Observables can be easily unsubscribed, there's no need for an explicit abortSignal to cancel requests.
* **Reactive Data Fetching**: Ideal for scenarios where you need to manage asynchronous data operations reactively.

By leveraging rxResource, Angular developers can efficiently manage data flow and reactivity in their applications, enhancing both productivity and performance