

IApi

August 17, 2026

IApi

Kind: Interface**Source:** Pams/Web/Pams.WebApp/Client/app/services/core/base-api-service/i-api.interface.ts
(line 3)**Part of:** [Pams](#)

`IApi<T>` defines the contract for services that access a named resource and identify records by a key field. It exposes observable-based methods for loading collections, loading an item, deleting data, and saving an item.

Properties

Property	Type
<code>keyName</code>	<code>string</code>
<code>resourceName</code>	<code>string</code>

Diagram

```
graph LR
  Client --> ApiService[IApi<T>]
  ApiService --> Resource[resourceName]
  ApiService --> Key[keyName]
  ApiService --> Get[get(): Observable<T[]>]
  ApiService --> GetById[getById(): Observable<T>]
  ApiService --> Delete[delete(): Observable<{}>]
  ApiService --> Save[save(): Observable<T>]
```

Usage

```
import { IApi } from './i-api.interface';
```

```

interface Project {
  id: string;
  name: string;
}

function loadProjects(api: IApi<Project>): void {
  console.log(api.resourceName);
  console.log(api.keyName);

  api.get().subscribe((projects) => {
    console.log(projects);
  });

  api.getById().subscribe((project) => {
    console.log(project);
  });

  api.save().subscribe((project) => {
    console.log(project);
  });

  api.delete().subscribe(() => {
    console.log('Project deleted');
  });
}

```

AI Coding Instructions

- Keep API service implementations compatible with the `IApi<T>` observable return types.
- Set `resourceName` to the backend resource handled by the service.
- Set `keyName` to the field used to identify resource records.
- Subscribe to returned observables where the calling layer needs response data or error handling.
- Preserve the interface method signatures when adding service implementations.

Used by

1 reference from 1 file. Each is a place in this repository where the symbol is actually used — go read one rather than trusting an example.

Imported by (1)

- `BaseApiService` — `Pams/Web/Pams.WebApp/Client/app/services/core/base-api-service/base-api.service.ts :1`

