

IApi

August 17, 2026

IApi

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

`IApi<T>` defines the contract for services that perform API operations for a resource type. It exposes resource metadata through `keyName` and `resourceName`, and returns RxJS `Observable` values for retrieval, deletion, and saving.

Properties

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

Diagram

```
graph LR
  IApi["IApi<T>"]
  IApi --> KeyName["keyName: string"]
  IApi --> ResourceName["resourceName: string"]
  IApi --> Get["get(): Observable<T[]>"]
  IApi --> GetById["getById(): Observable<T>"]
  IApi --> Delete["delete(): Observable<{}>"]
  IApi --> Save["save(): Observable<T>"]
```

Usage

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

interface User {
```

```

    id: string;
    name: string;
  }

  declare const userApi: IApi<User>;

  console.log(userApi.resourceName);
  console.log(userApi.keyName);

  userApi.get().subscribe((users) => {
    console.log(users);
  });

  userApi.getById().subscribe((user) => {
    console.log(user);
  });

  userApi.save().subscribe((savedUser) => {
    console.log(savedUser);
  });

  userApi.delete().subscribe(() => {
    console.log('User deleted');
  });

```

AI Coding Instructions

- Keep API service implementations aligned with the `Observable` return types declared by `IApi<T>`.
- Use `resourceName` to identify the API resource and `keyName` to identify the resource key field.
- Do not add method parameters unless the interface contract is updated at the same time.
- Handle subscriptions and errors at the caller or through the application's existing RxJS patterns.

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)

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

