

Application Programming Interface (API)

[Go up to Terms and Definitions](#)

Discussion

An Application Programming Interface (API) provides an implementation interface through which software components, systems, services, applications, tools, or users request, provide, exchange, or control information or behavior.

In the FX Demo [Reference Architecture](#), API references usually identify implementation mechanisms that later profiles may select. Logical architecture terms such as [Communication Endpoint](#), [Node](#), [Data Structure Definition](#), and [Data Structure Instance](#) do not prescribe an API.

An API can implement, expose, or support a logical communication pattern, but it does not define the logical meaning of the information exchanged.

Definition

implementation interface through which software components, systems, services, or applications request, provide, exchange, or control information or behavior

Source

Generalised from common software architecture usage and specialized for use in the FX Demo [Reference Architecture](#).

Note

An API is not a logical [Communication Endpoint](#), [Node](#), data structure, runtime plane, or deployment artifact. An implementation profile may use an API to realize or expose those architectural elements.

Example

A later implementation profile may use a REST API to expose an FX Release Package, while the logical architecture continues to define the release interaction through the FX Release Package Endpoint.

Last
update:
2026/08/04 07:09 dido:99_annexes:annex-b-terms-and-definitions:a:api https://wiki.didosolutions.com/dido/99_annexes/annex-b-terms-and-definitions/a/api

From:
<https://wiki.didosolutions.com/> - **Dido Solutions, Inc Wiki**

Permanent link:
https://wiki.didosolutions.com/dido/99_annexes/annex-b-terms-and-definitions/a/api

Last update: **2026/08/04 07:09**

