

API - Rozhraní pro programování aplikací

API (zkratka z anglického **A**pplication **P**rogramming **I**nterface) představuje soubor definovaných pravidel, protokolů a nástrojů, které umožňují vzájemnou komunikaci a výměnu dat mezi různými softwarovými aplikacemi.

V praxi API funguje jako prostředník (poslíček), který přijímá požadavek od klienta, doručí jej systému, a následně vrátí odpověď zpět uživateli. Díky tomu vývojáři nemusí vědět, jak je cílový systém vnitřně naprogramován; stačí jim znát rozhraní, které jim systém poskytuje.

1. Jak API funguje?

Princip API lze přirovnat k menu v restauraci:

- **Zákazník (Uživatel/Aplikace):** Vybírá si z nabídky dostupných funkcí.
- **Menu (API):** Seznam operací, které lze provést, a formát, v jakém musí být požadovány.
- **Kuchyně (Server/Systém):** Vykoná požadovanou logiku a připraví data.

2. Hlavní typy API

Podle způsobu přístupu a účelu dělíme rozhraní do několika kategorií:

Webová API (Nejrozšířenější)

Umožňují komunikaci mezi webovým prohlížečem a serverem, nebo mezi dvěma servery přes internet (protokol HTTP).

- **REST (Representational State Transfer):** Dnes nejpopulárnější standard. Využívá HTTP metody (GET, POST, PUT, DELETE) a data přenáší nejčastěji ve formátu **JSON**.
- **SOAP (Simple Object Access Protocol):** Starší, robustnější protokol založený na XML. Často využíván v bankovníctví pro vysokou úroveň zabezpečení.
- **GraphQL:** Moderní dotazovací jazyk od Meta (Facebooku), který umožňuje klientovi přesně definovat, jaká data chce dostat (řeší problém over-fetchingu).

Knihovní a OS API

Umožňují programům komunikovat s operačním systémem nebo využívat funkce specifických knihoven.

- **Win32 API:** Pro aplikace v systému Windows.
- **POSIX:** Standard pro unixové systémy (Linux, macOS).

3. Formáty přenosu dat

Aby si dvě aplikace rozuměly, musí data posílat v dohodnutém formátu. Nejčastěji se setkáváme s:

Formát	Charakteristika
JSON	JavaScript Object Notation. Lehký, čitelný pro lidi i stroje. Standard pro REST.
XML	Extensible Markup Language. Hierarchický, přísně strukturovaný, ale datově objemnější.
YAML	Často využíván pro konfigurační API (např. Kubernetes, Docker).

4. Proč je API důležité?

- **Abstrakce:** Vývojář nemusí znát složitost backendu (např. složité SQL dotazy), stačí mu zavolat funkci `get_users()`.
- **Integrace:** Umožňuje propojit nesourodé systémy (např. platební brána Stripe na vašem e-shopu).
- **Škálovatelnost:** Mobilní aplikace i webová stránka mohou používat stejné API pro přístup k datům.
- **Automatizace:** API umožňuje programům ovládat jiné programy bez lidského zásahu.

5. Bezpečnost a autentizace

Vzhledem k tomu, že API otevírá přístup k datům, musí být chráněno:

- **API Key:** Unikátní řetězec identifikující volající aplikaci.
- **OAuth 2.0:** Standard pro delegovanou autorizaci (např. „Přihlásit se přes Google“).
- **Rate Limiting:** Omezení počtu požadavků za minutu, aby nedošlo k přetížení serveru (ochrana proti DoS útokům).

Závěr

API je „lepidlem“ moderního internetu. Bez něj by neexistovaly mapy v aplikacích pro rozvoz jídla, sdílení příspěvků mezi sociálními sítěmi ani propojení cloudových služeb. Výběr správného typu rozhraní (REST vs. GraphQL) je klíčovým rozhodnutím při návrhu jakéhokoli softwarového systému.

From:
<https://serviceit.cz/> - IT ENCYKLOPEDIE

Permanent link:
<https://serviceit.cz/doku.php?id=it:dev:api>

Last update: 2026/03/11 11:42



