

RESTful API (Representational State Transfer) je architektonický styl, který se používá k navrhování webových služeb. Principy RESTful API se zakládají na několika základních principech:

1. **Stateless (bezstavovost):** Každý požadavek od klienta na server musí obsahovat všechny potřebné informace pro jeho zpracování. Server nemá žádný stav o klientovi, každý požadavek je nezávislý.
2. **Zdroje a reprezentace:** Data na serveru jsou organizována jako zdroje (resources), které jsou identifikovány pomocí URI (Uniform Resource Identifier). Klient může s těmito zdroji interagovat pomocí standardních HTTP metod (GET, POST, PUT, DELETE) a získávat nebo manipulovat data v různých reprezentacích (např. JSON, XML).
3. **Jednotné rozhraní:** RESTful API poskytuje jednotné rozhraní pro komunikaci s různými zdroji. Klienti mohou používat stejné HTTP metody a standardní URI pro interakci se zdroji bez ohledu na jejich typ nebo umístění.
4. **Volitelná vrstva cachování:** Server může nastavit zdroje jako cachovatelné, což umožňuje klientům ukládat lokální kopie dat a snižovat tak počet požadavků na server.
5. **Klientské aplikace:** Klientské aplikace jsou oddělené od serveru a komunikují s ním pomocí standardních protokolů, jako je HTTP.

Komunikace s RESTful API pomocí Pythonu se provádí pomocí knihovny `requests`. Tato knihovna umožňuje provádět HTTP požadavky na server a zpracovávat odpovědi. Zde je základní postup, jak komunikovat s RESTful API pomocí Pythonu:

1. **Instalace knihovny requests:** Nejprve musíte nainstalovat knihovnu `requests`, pokud ji ještě nemáte. Můžete to udělat pomocí nástroje `pip`, spuštěním příkazu `pip install requests` v terminálu.
2. **Import knihovny requests:** V Pythonu musíte nejprve importovat knihovnu `requests`.
3. **Vytvoření HTTP požadavku:** Použijte funkce z knihovny `requests`, jako je `get`, `post`, `put`, nebo `delete`, pro vytvoření HTTP požadavku na server. Specifikujte URI zdroje a případně další parametry jako data nebo hlavičky.
4. **Zpracování odpovědi:** Po odeslání požadavku získáte odpověď od serveru. Odpověď může být ve formátu JSON, XML nebo jiném, podle toho, jak server data reprezentuje. Můžete použít metody z knihovny `requests` pro zpracování odpovědi, jako je `json()` pro zpracování JSON dat.
5. **Zpracování chyb:** Pamatujte na zpracování chyb, které mohou nastat při komunikaci s API, jako jsou chybné odpovědi nebo nedostupnost serveru.