

1. Úvod: Proč se učíme API? (Real-World Context)

Možná si říkáš: "API? To je jen nějaká Roblox věc, ne?" Omyl. API (Application Programming Interface) je základní stavební kámen celého moderního internetu a komerčních aplikací.

-  **Weby:** Když si na webu pizzerie zobrazíš mapu, ten web používá **Google Maps API**. Neumí kreslit mapy sám, „poprosí“ Google.
-  **Aplikace:** Když se podíváš na počasí v mobilu, aplikace nevypouští meteorologické balóny. Používá **Weather API**, aby stáhla data ze serverů meteorologů.
-  **Animace:** To, co dnes budeme dělat (**Tweening**), používají weboví vývojáři pro plynulé animace tlačítek a menu (v CSS/JavaScriptu) nebo vývojáři v Unity pro profesionální hry.

Dnes nebudeš jen „stavět v Robloxe“. Dnes se naučíš volat služby (**Services**) jako profesionální programátor.

2. Zadání Projektu: Duch Výtah

Tvým úkolem je vytvořit výtah, který se nepohybuje trhaně změnou pozice (teleportací), ale **plynule klouže** prostorem.

Použitá technologie: **TweenService** (Služba pro interpolaci pohybu).

3. Krok za krokem (Start to Finish)

Krok 1: Příprava scény

1. Vytvoř **Part** (blok), který bude sloužit jako výtah.
2. Pojmenuj ho **Elevator**.
3. **DŮLEŽITÉ:** Zapni mu vlastnost **Anchored** (jinak by spadl fyzikou dolů a Tween by se s ní "pral").
4. Vlož do něj nový **Script** (Server Script, ne LocalScript).

Krok 2: Volání API (Služby)

V kódu si musíme "objednat" službu, která umí počítat plynulý pohyb.

-  **Dokumentace:** [TweenService na Roblox Creator Hub](#)

Napiš do skriptu:

Lua

- -- 1. Získáme službu (API), která umí animovat
- local TweenService = game:GetService("TweenService")
-
- -- 2. Odkaz na náš výtah
- local part = script.Parent

Krok 3: Nastavení pohybu (TweenInfo)

Musíme počítači říct, *jak* má pohyb vypadat. Má být rychlý? Má na konci zpomalit (jako opravdový výtah)? Tomu se říká **Easing**.

-  **Vizualizace Easing Styles:** [Easing Style Cheat Sheet](#) (*Ukaž jim tohle! Je to super vizuální pomůcka*).

Přidej do skriptu:

Lua

- -- Nastavení animace (čas, styl pohybu)
- local tweenInfo = TweenInfo.new(
 - 4, -- Čas: Jak dlouho cesta trvá (v sekundách)
 - Enum.EasingStyle.Sine, -- Styl: Sine je plynulý rozjezd a dojezd
 - Enum.EasingDirection.InOut, -- Směr: Zpomalí na začátku i na konci
 - 0, -- Opakování: Kolikrát opakovat (0 = neopakovat hned)
 - false, -- Reverse: Má se vrátit zpět? (my to uděláme ručně)
 - 0 -- Zpoždění
 -)

Krok 4: Definice cílů (Target Properties)

Kam má výtah dojet? TweenService potřebuje vědět cílovou vlastnost.

Lua

- -- Pozor: Změň souřadnice podle toho, kde máš výtah ve hře!
- -- Cílová pozice NAHORE (např. o 20 studů výš)
- local targetUp = {Position = part.Position + Vector3.new(0, 20, 0)}
-
- -- Cílová pozice DOLE (původní pozice)
- local targetDown = {Position = part.Position}

Krok 5: Vytvoření a spuštění animace

Ted' spojíme vše dohromady. Vytvoříme "instrukce" pro pohyb (Tween) a řekneme API, aby je přehrálo (:Play()).

Lua

- -- Vytvoření samotných animací (zatím se nespustí, jen se připraví)
- local tweenUp = TweenService>Create(part, tweenInfo, targetUp)
- local tweenDown = TweenService>Create(part, tweenInfo, targetDown)
-
- -- Nekonečná smyčka výtahu
- while true do
- wait(2) -- Čeká dole
- print("Jedeme nahoru!")
- tweenUp:Play() -- Spusťí animaci nahoru
-
- tweenUp.Completed:Wait() -- DŮLEŽITÉ: Skript čeká, dokud animace nedojede
-
- wait(2) -- Čeká nahoře
- print("Jedeme dolů!")

- `tweenDown:Play()` -- Spustí animaci dolů
 -
 - `tweenDown.Completed:Wait()` -- Čeká, dokud animace nedojede
 - `end`
-

Extra výzvy pro rychlíky (Level Up)

Pokud to máš hotové a funguje to, zkus tohle:

1. **Změna barvy:** Uprav `targetUp` tak, aby se kromě `Position` měnila i `Color`. (TweenService umí měnit více vlastností najednou!).
 - *Nápočeda:* `local targetUp = {Position = ..., Color = Color3.new(1, 0, 0)}`
2. **Zvukový doprovod:** Najdi v **Toolboxu** zvuk výtahu, vlož ho do Partu a pomocí skriptu ho spusť (`:Play()`) když výtah jede a zastav (`:Stop()`), když stojí.
 -  **Info o zvuku:** [Sound API](#)
3. **Aktivace tlačítkem:** Předělej `while true do` smyčku tak, aby se výtah rozjel jen tehdy, když se hráč dotkne jiného tlačítka (použij ProximityPrompt z minula nebo event `.Touched`)
 -