Autentizace ASP.NET Web API pomocí API klíčů

Michal Altair Valášek

Altairis, s. r. o.

www.altair.blog

michal.valasek@altairis.cz

Tech·Ed GOPAS

Možnosti autentizace ASP.NET Web API

- **Cookie Authentication Middleware**
  - Naprosto nejjednodušší, funguje úplně stejně jako u ostatních stránek
  - Vhodné pro API konzumovaná webovou aplikací z prohlížeče
- **OAuth, JWT Bearer Token**
  - Relativně komplikované nastavení
  - Nabízí největší možnosti
  - Fakticky vyžaduje interakci uživatele
  - Vhodné pro mobilní a desktopové aplikace, kde se uživatel interaktivně přihlašuje
- **API Keys**
  - Vhodné pro M2M komunikaci, bez interakce uživatele
  - Bohužel neexistuje standardizovaná implementace

Co jsou API keys

- Bezvýznamový identifikátor, který slouží k autentizaci klienta
  - Buďto staticky, na základě hodnoty v konfiguraci
  - Nebo dotazem do databáze uživatelů
- Obvykle náhodný řetězec znaků
  - Doporučuji alespoň 256 bitů entropie
  - Nepoužívejte běžným způsobem generované GUIDy!
- Vždy mějte minimálně dva!
  - Jedině tak lze zajistit možnost jejich změny bez výpadku aplikace
  - Viz např. Azure Storage Account

Jak předávat API klíč?

- Jako **součást URL**
  - Cesta, query string, host name
  - Vhodné v případě opravdu primtivních klientů
- Jako **součást datové struktury** požadavku
  - Nepohodlné pro server i klienta
  - Obecně nedoporučuji
- Jako **vlastní HTTP hlavičku**
  - Např. `X-API-Token: {api-token}`
  - Funguje, ale jde mimo standardy
- V rámci **Basic autentizace**
  - `Authorization: Basic base64({username}:{password})`
  - Lepší, ale jde o jisté zneužití standardu
- Jako **Bearer token**
  - Moderní, formálně čisté

Bearer token authentication

- *„Give access to the bearer of this token.“*
- Součást specifikace OAuth 2.0
  - Tam se používá pro předávání JWT (Json Web Token)
  - Nicméně lze jej použít i samostatně
- Využívá standardní HTTP hlavičku `Authorization`
  - `Authorization: Bearer {api-token}`
  - Token může mít jakýkoliv formát

Napojení na ASP.NET Core infrastrukturu

- Nedělejte vlastní middleware nebo něco takového!
  - Low-level řešení, kterým si zadělováte na problémy s kompatibilitou
- Místo toho vytvořte vlastní autentizační schéma
  - Pak lze jednoduše kombinovat různé metody autentizace v jedné aplikaci
  - Umožňuje také snadno napojit na OpenAPI (Swagger a Swagger UI)

DEMO

Autentizace pomocí statického klíče

https://github.com/ridercz/Incitatus

Napojení na ASP.NET Core Identity

- Uložte k uživatelům informaci o jejich klíčích
  - Klíče by měly být nejméně dva
  - Jde o bezpečnostně kritickou informaci, ale nemůžete je zahashovat jako hesla
  - Nezapomeňte na ně dát v databázi indexy, budete podle nich hledat
- Vytvořte vlastní authentication scheme handler
  - Podle klíče najděte uživatele
  - Načtěte odpovídající claimy a vytvořte `ClaimsIdentity`
- V API použijte `[Authorize]` s vlastním schématem
- Uživatele rozeznáte jako obvykle, podle claimů

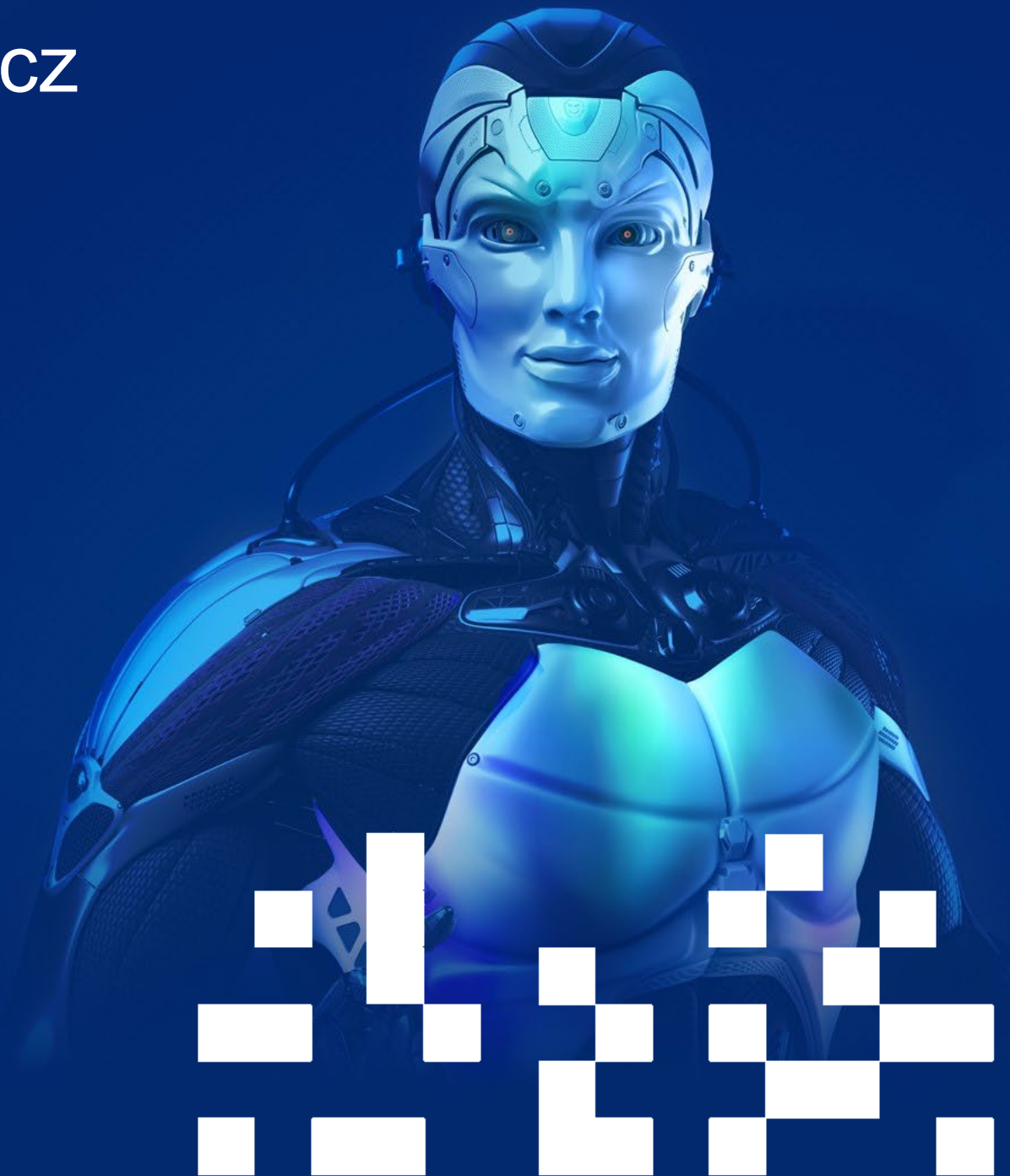
DEMO

Autentizace pomocí uživatelských klíčů

michal.valasek@altairis.cz

facebook.com/**rider.cz**
github.com/**ridercz**
twitter.com/**ridercz**
linkedin.com/in/**ridercz**

www.**altairis.cz**
www.**rider.cz**
www.**altair.blog**



Tech·Ed GOPAS