

Cucumber

Cucumber je open-source nástroj pro **behavior-driven development (BDD)**, který umožňuje psát testy v přirozeném jazyce (např. češtině nebo angličtině), srozumitelném nejen vývojářům, ale i testerům, produktovým manažerům nebo zákazníkům. Cucumber slouží k popisu **chování systému** prostřednictvím **uživatelských scénářů** a automaticky tyto scénáře spouští jako **funkční (akcepční) testy**.

Cucumber původně vznikl pro jazyk **Ruby**, dnes však existují oficiální i komunitní implementace pro řadu jazyků, včetně **Java, JavaScript, Python, C#** a dalších.

Princip BDD a Gherkin

Cucumber vychází z metodiky **BDD** (Behavior-Driven Development), která rozšiřuje **TDD** (Test-Driven Development) o spolupráci mezi technickými a netechnickými členy týmu. Chování aplikace je popsáno pomocí **doménového jazyka** (Domain-Specific Language, DSL), který v Cucumberu nese název **Gherkin**.

Gherkin - jazyk pro popis scénářů

Gherkin je **formální, jednoduchý a čitelný jazyk**, který používá klíčová slova jako:

- **Feature** - popisuje funkcionalitu systému.
- **Scenario** - konkrétní případ použití této funkality.
- **Given** - počáteční stav („předpoklady“).
- **When** - akce uživatele nebo systému.
- **Then** - očekávaný výsledek.
- **And / But** - pro spojení více kroků stejného typu.

Gherkin podporuje lokalizaci - kromě anglických klíčových slov lze použít i česká (např. **Požadavek, Scénář, Za předpokladu, Když, Pak**), ale **anglická varianta je nejrozšířenější** kvůli nástrojové podpoře.

Příklad .feature souboru

Soubor: `prihlaseni.feature`

```
# language: cs
Požadavek: Přihlášení uživatele do systému

Aby mohl uživatel přistupovat ke svým datům,
musí se nejprve přihlásit pomocí platných přihlašovacích údajů.

Scénář: Úspěšné přihlášení s platnými údaji
```

Základní předpoklad, že uživatel "jan" existuje s heslem "tajne_heslo"
Když uživatel zadá přihlašovací jméno "jan" a heslo "tajne_heslo"
Pak by měl být přihlášen a přesměrován na úvodní stránku

Scénář: Neúspěšné přihlášení – špatné heslo

Základní předpoklad, že uživatel "jan" existuje s heslem "tajne_heslo"
Když uživatel zadá přihlašovací jméno "jan" a heslo "špatné"
Pak by se měla zobrazit chybová zpráva "Neplatné přihlašovací údaje"

☐ **Poznámka:** I když je podpora češtiny technicky možná, v praxi se důrazně doporučuje používat **anglický Gherkin**, protože:

- většina IDE a nástrojů lépe podporuje angličtinu,
- testovací kroky (step definitions) jsou často psány v angličtině,
- zajišťuje lepší kompatibilitu v mezinárodních týmech.

Jak Cucumber funguje?

1. **Popis chování** – tým vytvoří `.feature`` soubor v Gherkinu. 2. **Implementace kroků** – vývojář napíše tzv. **step definitions**, což jsou funkce (v Java, JS, atd.), které mapují jednotlivé řádky Gherkinu na skutečný kód (např. ovládání webového prohlížeče přes Selenium). 3. **Spuštění testu** – Cucumber spustí scénáře a pro každý krok zavolá odpovídající step definition. 4. **Výstup** – Cucumber vygeneruje report (textový, HTML, atd.), který ukazuje, které scénáře prošly a které selhaly.

Příklad step definition v Java (s JUnit)

```
@Given("základní předpoklad, že uživatel {string} existuje s heslem {string}")
public void uzivatelExistuje(String jmeno, String heslo) {
    database.vytvorUzivatele(jmeno, heslo);
}

@When("uživatel zadá přihlašovací jméno {string} a heslo {string}")
public void uzivatelZadaUdaje(String jmeno, String heslo) {
    stranka.prihlaseni.zadejUdaje(jmeno, heslo);
    stranka.prihlaseni.odesli();
}

@Then("by se měla zobrazit chybová zpráva {string}")
public void zkontrolujChybu(String zprava) {
    assertTrue(stranka.obsahujeText(zprava));
}
```

Podpora jazyků a frameworků

Jazyk	Knihovna / Plugin	Poznámka
-----	-----	-----

Java	<code>`io.cucumber:cucumber-java`</code>	Nejrozšířenější verze, často s JUnit/TestNG
JavaScript	<code>`@cucumber/cucumber`</code> (dříve Cucumber.js)	Pro Node.js, podpora pro Playwright, Cypress
Python	<code>`behave`</code> (alternativa), <code>`pytest-bdd`</code>	Oficiální Cucumber pro Python není aktivně vyvíjen
C#	<code>`SpecFlow`</code>	Oficiální Cucumber pro .NET
Ruby	<code>`cucumber`</code> gem	Původní implementace

Výhody použití Cucumberu

- **Živá dokumentace** - `.feature`` soubory slouží jako aktuální popis systému.
- **Lepší komunikace** - most mezi business a technickým týmem.
- **Automatizace akceptčních testů** - snadná validace požadavků.
- **Regrese** - snadné odhalení chyb po změnách v kódu.

Nevýhody a rizika

- **Přetížení Gherkinem** - psaní příliš detailních nebo technických kroků porušuje BDD princip.
- **Údržba testů** - změna UI může vyžadovat úpravu mnoha step definitions.
- **Výkon** - Cucumber testy jsou obvykle **end-to-end**, tedy pomalejší než jednotkové testy.
- **Falešná bezpečnost** - „zelené testy“ neznamenají nutně kvalitní systém - pokud jsou scénáře špatně navrženy.

Doporučené postupy (Best Practices)

- Příkazy v Gherkinu by měly popisovat **co**, ne **jak** (např. „Když přidám produkt do košíku“, ne „Když kliknu na tlačítko X“).
- Vyhněte se podmínkám a cyklům v `.feature`` souborech - raději vytvořte samostatné scénáře.
- Používejte **parametry** (`{string}``, `{int}``) místo opakování podobných kroků.
- Udržujte step definitions **DRY** (Don't Repeat Yourself) - vytvářejte režné metody.
- Integrujte Cucumber do **CI/CD pipeline** (např. GitHub Actions, Jenkins).

Nástroje a ekosystém

- **Cucumber Reports** - generování vizuálních HTML reportů.
- **Pickles** - transformace `.feature`` souborů na živou dokumentaci.
- **Selenium WebDriver** - často používán s Cucumberem pro testování webových aplikací.
- **Allure, ExtentReports** - pokročilé reportovací nástroje kompatibilní s Cucumberem.

Související pojmy

- [BDD](#)
- [TDD](#)
- [Testování softwaru](#)

- [Gherkin](#)
- [Selenium](#)
- [Akcepční testování](#)

Externí odkazy

- Oficiální web: <https://cucumber.io>
- Dokumentace: <https://cucumber.io/docs>
- Gherkin reference: <https://cucumber.io/docs/gherkin/>
- Cucumber pro Java: <https://github.com/cucumber/cucumber-jvm>
- Cucumber pro JavaScript: <https://github.com/cucumber/cucumber-js>

Viz také

- [Testovací frameworky](#)
- [CI/CD](#)
- [End-to-end testování](#)
- [Živá dokumentace](#)

From:

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

Permanent link:

<https://serviceit.cz/doku.php?id=cucumber>

Last update: **2025/12/31 22:08**

