

VSCoDe OpenProject Rozšírenie

Technický Report (GitHub & GitLab Integrácia, PR Monitor)

Autor: Nykyta
Semester: Letný semester
Zameranie: Backend, API integrácia tretích strán & Automatizácia

1 CIEĽ PROJEKTU

V letnom semestri sme plynule nadviazali na predchádzajúci vývoj nášho rozšírenia pre vývojové prostredie Visual Studio Code. Hlavným cieľom bolo prepojiť toto prostredie s najpopulárnejšími platformami na správu zdrojového kódu a verzií – GitHub a GitLab. Touto integráciou sme chceli maximalizovať efektivitu práce vývojárov a automatizovať rutinné úkony spojené s manuálnym prepínaním stavov úloh. Moja časť práce sa sústredila výhradne na návrh a implementáciu komplexnej komunikačnej a integračnej vrstvy pre externé REST API a vytvorenie robustného monitorovacieho servisu na pozadí rozšírenia.

2 PREHLAD REALIZOVANEJ PRÁCE (BACKENDOVÁ INTEGRÁCIA & INTEGRÁCIA TRETÍCH STRÁN)

V rámci rozdelenia úloh v tíme som prebral plnú zodpovednosť za logiku na pozadí celého rozšírenia (backend), zatiaľ čo sa môj kolega venoval rozvoju používateľského rozhrania, formulárom a správe dátumov vo Webview komponentoch.

2.1 Integrácia GitHub a GitLab API

Prvým krokom bolo podrobné preštudovanie technickej dokumentácie a integrácia REST API platforiem GitHub a GitLab. Navrhol a implementoval som asynchrónne komunikačné moduly, ktoré bezpečne a efektívne sťahujú informácie o otvorených, aktualizovaných a zlúčených (merged) požiadavkách na zmenu (Pull Requests na GitHube a Merge Requests na GitLabe).

2.2 Implementácia prMonitorService.ts a Regex Parsovanie

Srdcom backendovej implementácie pre letný semester sa stala služba prMonitorService.ts, ktorú som kompletne navrhol a naprogramoval. Táto služba periodicky na pozadí dopytuje (polling) nakonfigurované repozitáre a vykonáva nasledujúce automatizované operácie:

- **Detekcia metadát pomocou regulárnych výrazov:** Na identifikáciu unikátnych ID úloh (Work Packages) z OpenProjectu v Git metadátach som implementoval flexibilnú parsovaciu logiku postavenú na dvoch hlavných regulárnych výrazoch (case-insensitive):

1. Vzor pre Názov a Popis MR/PR:

`/ (? : openproject | op ) # ( \d + ) / i`

Príklady zachytenia: OP#123, op#123, openproject#123 v textovom názve alebo popise požiadavky.

2. Vzor pre Vetvy, Názov a Popis:

`/ wp [ - / ] ( \d + ) / i`

Príklady zachytenia: wp-123 alebo wp/123 v názve zdrojovej vetvy (source branch), prípadne priamo v textoch MR.

- **Automatické uzatváranie úloh:** Ak monitor na základe vyextrahovaného ID úspešne spáruje vetvu alebo MR s konkrétnou úlohou a zistí, že daná požiadavka bola úspešne zlúčená (merged), automaticky vyvolá PATCH požiadavku prostredníctvom našej existujúcej triedy `ApiClient`. Systém následne zmení stav zodpovedajúcej úlohy v OpenProjecte na "Uzatvorená" (Closed), čím úplne odbúrava nutnosť manuálnej aktualizácie zo strany vývojára.

2.3 Architektonický návrh a Extenzibilita

Pri implementácii `prMonitorService.ts` som kládol veľký dôraz na čistú architektúru a princípy objektového návrhu. Komunikáciu s repozitármi som abstrahoval do jednotného rozhrania (interface). Vďaka tomuto prístupu je monitorovacia služba plne modulárna a **ľahko rozšíriteľná pre akúkoľvek ďalšiu platformu** (napr. Bitbucket, self-hosted Git riešenia), ktorá funguje na podobnom princípe ako GitHub, bez nutnosti akéhokoľvek zásahu do jadra parsovacej a synchronizačnej logiky.

3 TÍMOVÁ SPOLUPRÁCA A PREPOJENIE S FRONTENDOM

Úspešná realizácia si vyžadovala precíznu synchronizáciu v tíme. Spoločne s kolegom sme navrhli optimálny dátový tok, akým sa zmeny stavu z repozitárov automaticky prenášajú do prostredia editora. Moja backendová služba generuje presné informácie o stave monitorovania (čakanie, prebiehajúce dopytovanie, chyby spojenia), ktoré kolega úspešne namapoval na interaktívne prvky používateľského rozhrania – konkrétne na ikonu a tooltip v spodnej lište editora (Status Bar) a na vyskakovacie notifikácie, ktoré vývojára upozornia na zmenu stavu a umožnia mu jedným kliknutím otvoriť daný Pull Request v prehliadači.

4 ZÁVER

Práca v letnom semestri úspešne posunula naše rozšírenie do finálnej a plne funkčnej podoby. Vývoj backendu pre `prMonitorService.ts` mi priniesol cenné praktické skúsenosti s paralelným dopytovaním viacerých odlišných API tretích strán, pokročilým spracovaním textových dát pomocou regulárnych výrazov a návrhom flexibilnej, rozšíriteľnej architektúry v TypeScript. Výsledné riešenie tvorí stabilný most medzi správou zdrojového kódu a plánovaním úloh v OpenProjecte.