

VSCoDe OpenProject Rozšírenie

Zima Report

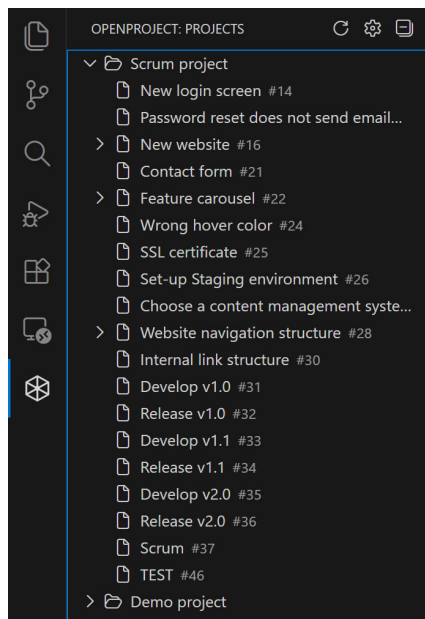
Cieľ projektu

Hlavným cieľom tohto projektu bolo navrhnúť a implementovať natívne rozšírenie pre vývojové prostredie Visual Studio Code, ktoré integruje nástroj na riadenie projektov OpenProject. Primárnym zámerom bolo zjednodušiť prácu vývojárov minimalizovaním potreby prepínania medzi editorom kódu a webovým prehliadačom. Rozšírenie umožňuje priamu správu úloh (Work Packages) priamo v prostredí IDE, čím prispieva k udržaniu plynulosti vývojového procesu.

1 Prehľad realizovanej práce (Tímová spolupráca)

Na projekte sme pracovali ako tím, pričom sme rozdelili zodpovednosti medzi backendovú integráciu a klientsku časť rozšírenia. Základom riešenia je jazyk TypeScript a VS Code Extension API.

Kolega sa primárne sústredil na jadro komunikácie s API a autentifikáciu. Spoločne sme však definovali štruktúru príkazov (Commands) a dátové modely potrebné pre korektné zobrazenie informácií v editore.



Obr. 1: Stromová štruktúra projektov v bočnom paneli rozšírenia.

2 Môj prínos k projektu

Môj prínos zahŕňal kompletnú implementáciu komponentov, ktoré tvoria vizuálnu a interaktívnu vrstvu rozšírenia. Zodpovedal som za logiku zobrazenia dát, používateľské rozhranie a vybrané integračné metódy.

2.1 Project Explorer a TreeDataProvider

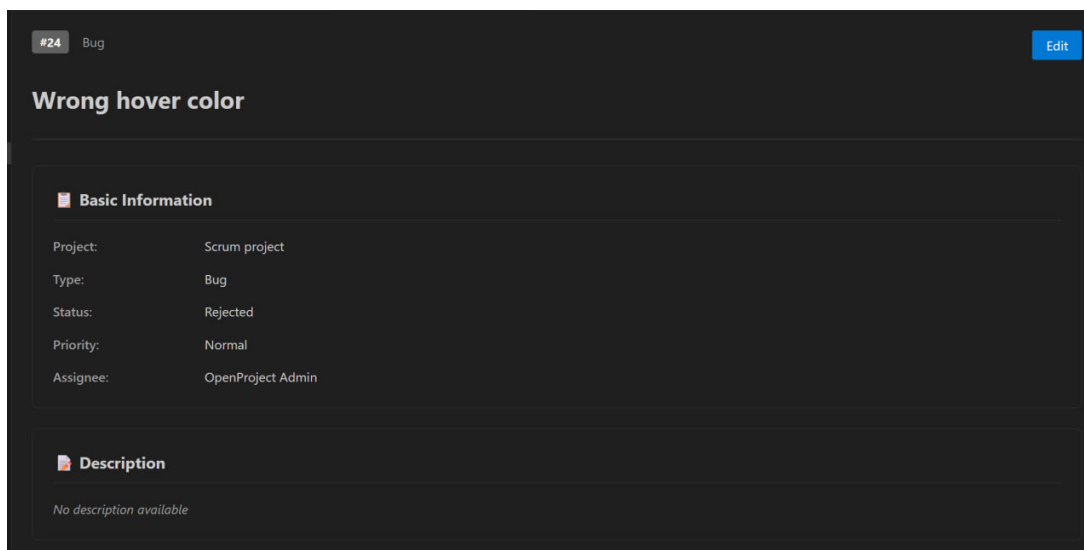
Implementoval som kľúčový komponent `workPackageTreeProvider.ts`, ktorý zabezpečuje hierarchické zobrazenie projektov a ich úloh.

- Navrhol som logiku spracovania surových dát do vizuálnej podoby stromu.
- Implementoval som metódy na dynamické rozbaľovanie uzlov (Lazy Loading), čo optimalizuje rýchlosť načítania pri veľkom počte úloh.

2.2 Implementácia Frontendu a Webview

Vytvoril som komplexné rozhranie v `workPackageWebview.ts`, ktoré slúži na detailnú prácu s úlohami.

- **Dynamické UI:** Navrhol som HTML/CSS šablóny pre zobrazenie detailov úlohy priamo v editore.
- **Interaktivita:** Implementoval som prepínanie medzi režimom prezerania a editácie a zabezpečil obojsmernú komunikáciu medzi Webview a logikou rozšírenia cez `postMessage`.
- **Hlavná logika:** V súbore `extension.ts` som integroval všetky súčasti tak, aby rozšírenie korektne reagovalo na interakcie používateľa.



Obr. 2: Detailný pohľad na úlohu v režime prezerania.

2.3 Integrácia číselníkov (Spoločná práca na API)

V rámci prípravy dátových podkladov pre používateľské rozhranie sme s kolegom spoločne pracovali na integrácii kľúčových metód pre prácu s metadátami.

- Spoločne sme navrhli a implementovali funkcie na dopytovanie číselníkov pre **Status**, **Priority** a **Typy** úloh z OpenProject API.
- Zabezpečili sme korektnú transformáciu týchto dát, aby boli okamžite dostupné pre dynamické prvky (dropdown menu) v editovacom rozhraní, čím sme zaručili konzistenciu dát medzi backendom a frontendom.



The screenshot shows a web form titled "Edit Work Package #24". It has a dark background. The form contains several input fields and dropdown menus. The "Subject" field has the text "Wrong hover color". The "Description (Markdown)" field has the text "Hello". There are three dropdown menus: "Type" with "Bug" selected, "Status" with "Rejected" selected, and "Priority" with "Normal" selected. At the bottom right, there are two buttons: "Cancel" and "Save Changes".

Obr. 3: Režim úpravy úlohy s interaktívnymi prvkami.

3 Záver

Práca na tomto projekte mi umožnila hlbšie pochopiť architektúru rozšírení pre Visual Studio Code. Úspešne som prepojil vizuálnu časť (Webview) s logikou stromového zobrazenia a zabezpečil integráciu dôležitých dátových komponentov z API. Práca v tíme ma naučila efektívne koordinovať vývoj frontendu s backendovými službami.