

PRACOVNÝ ZOŠIT PRE PRVORÁDOVÚ LOGIKU

Ročníkový projekt · Študent: Matej Mok · Vedúci: Ján Křuka

Report zo zimného semestra

V zimnom semestri sme sa sústredili na vylepšenia pracovného zošita, ktoré mali vplyv na jeho dobrú používateľnosť.

Globálne LaTeX definície

Obsah hárku pracovného zošita tvorí zoznam buniek. Z tohto dôvodu nebolo jednoduché používanie LaTeX makier, keďže definícia makra v jednej textovej bunke neumožňovala jeho používanie v iných textových bunkách. Implementovali sme teda funkcionality na definovanie globálnych LaTeX makier, ktoré je možné používať v každej textovej bunke.

Formátovanie textu pomocou HTML

Markdown v textových bunkách sme obohatili o možnosť používania HTML tagov na dodatočné formátovanie. Použili sme pri tom pluginy knižnice remark, ktorú používame na preklad markdown syntaxe do HTML. Na povolenie spracovania HTML sme použili plugin z balíčka `rehype-raw` a ďalej na ošetrovanie nebezpečných tagov sme použili plugin z balíčka `rehype-sanitize`.

Optimalizácia automatického ukladania

Jenou z požiadaviek na ročníkový projekt bolo zníženie počtu commitov vykonaných v repozitári, v ktorom je uložený hárk pracovného zošita. Avšak časté aktualizovanie repozitára rieši problém s automatickým ukladáním zmien, ktorý môže nastať ak používateľ otvorí jeden hárk viac krát. V tomto prípade, ak nastane zmena hárku v jednej inštancii aplikácie, ostatné sa o tejto zmene nedozvedia. Ak by teda používateľ začal robiť zmeny aj v inej inštancii, ktorá už neobsahuje aktuálny hárk, automatické ukladanie by zlyhalo. Takže čím častejšie sa aktualizuje repozitár, tým rýchlejšie sa používateľ dozvie o probléme s ukladáním a tým menej potenciálnych zmien nebude možné uložiť. Rozhodli sme sa teda že časté zmeny v aktualizácii repozitára zachováme, ale budeme v určitých intervaloch vykonávať zlučovanie väčšieho množstva commitov do jedného. Keďže v jednom repozitári môže byť viac hárkov, pred úpravou súboru je vytvorená nová vetva repozitára ktorá je zviazaná s konkrétnym súborom hárku a do ktorej sa pridávajú commity pre jeden konkrétny hárk. To nám umožňuje izolovať commity súvisiace s jedným súborom hárku. Túto vetvu môžeme následne zlúčiť do hlavnej vetvy repozitára a Github API nám umožní vykonať tzv. *squashing*, výsledkom čoho je, že všetky commity v tejto vetve sa zlúčia do jedného a ten je pridaný do hlavnej vetvy. Zlúčenie zmien vykonaných vo vytvorenej vetve do hlavnej robí používateľ manuálne kliknutím na tlačítko.

Pri práci na novom spôsobe aktualizácie repozitára sme refaktorovali kód hárku výsledkom čoho bolo oddelenie kódu zodpovedného za stav hárku a kódu zodpovedného za jeho ukladanie. Tým sa zlepšila jeho prehľadnosť a umožnilo nám to ľahšie implementovať ďalšiu funkcionality.

Offline režim

V pracovnom zošite nebolo vyriešené korektné fungovanie pri výpadku pripojenia na internet. Toto negatívne ovplyvňovalo jeho použiteľnosť napr. pri použití menej stabilného bezdrôtového pripojenia. Doplnili sme teda kód, ktorý po prechode prehliadača do offline režimu pozastaví automatické ukladanie zmien do repozitára a po návrate do online režimu sa ukladanie obnoví.

Undo/redo

Refaktoring kódu hárku a fakt, že jeho stav je spravovaný pomocou reduxu nám umožnil jednoduchú implementáciu undo/redo functionality použitím knižnice `redux-undo`.

Integrácia aplikácie Formalization checker

Testovali sme možnosti integrácie. Táto aplikácia kounikuje s backendom a vyžaduje prihlásenie pomocou účtu na službe github, teda rovnaký spôsob aký používame v našej aplikácii Workbook. Navrhli sme spôsob zdieľania prístupového tokenu, ktorý získa Workbook po prihlásení s integrovanou aplikáciou Formalization checker. Na tejto aplikácii však ešte pracujú iní študenti a jej integráciu sme zatiaľ nedokončili.