Slovenská technická univerzita v Bratislave
Fakulta informatiky a informačných technológií
Ilkovičova 2, 842 16 Bratislava 4

Modul – Visual Studio Extension

Akademický rok: 2016/2017
Vedúci práce: Ing. Karol Rástočný, Phd.
Členovia tímu: bc. Ondrej Čičkán, bc. Šimon Dekrét, bc. Dušan Javorník
bc. Dušan Jom, bc. Miroslav Laco, bc. Anton Ján Vrban
Dátum poslednej zmeny: 9.12.2016

1 Úvod

Tento dokument opisuje rozšírenie pre vývojové prostredie Visual Studio, ktoré slúži ako tenký klient integrujúci funkcionalitu CodeCrutches.

2 Komponenty

Hlavný komponent CodeCrutches Visual Studio Extension (skrátene VsExtension) predstavuje rozšírenie pre IDE Visual Studio, ktoré má každý používateľ nainštalovaný lokálne vo svojom prostredí. Rozšírenie nástroja Visual Studio, ktoré vyvíjame komunikuje s rôznymi komponentmi pri zabezpečení správnej funkcionality. Na obrázku Diagram 1: Komponenty je znázornený diagram, ktorý zahŕňa všetky relevantné komponenty.

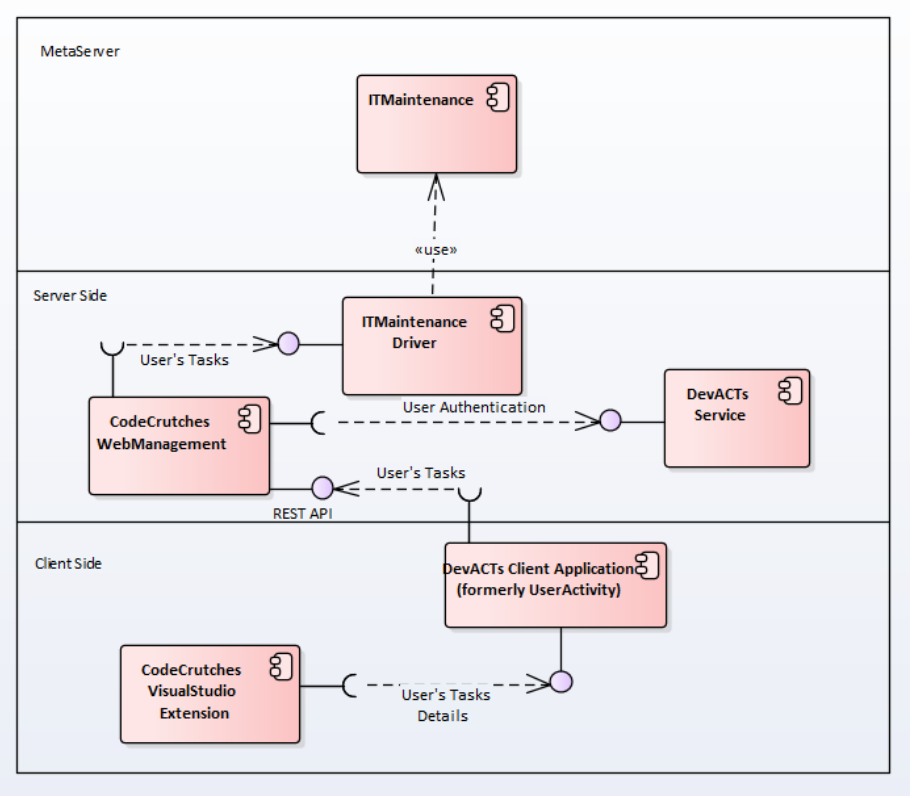

Diagram 1: Komponenty využívané rozšírením CodeCrutches VisualStudio Extension

Komentár od [OČ1]: Zrušiť komunikáciu s ITM - komunikuje cez DevACTs

Komentár od [KR2]: Čoho?

2.1 Komponent DevACTs Client Application

Rozšírenie komunikuje s aplikáciou DevACTs Client (pôvodne nazývaná UserAcitivity), ktorú musí mať používateľ nainštalovanú. Pri jej nainštalovaní tiež zadáva svoje meno a heslo do platformy DevACTs.

Meno aktuálne prihláseného používateľa sa získava z registra s názvom `user/Name` (umiestnenom v `HKEY_CURRENT_USER\SOFTWARE\JavaSoft\Prefs\com\gratex\perconik`

Komentár od [KR3]: Toto by som komunikáciou nenazýval.

`\useractivity\app)`. Druhý parameter, ktorý potrebuje VsExtension z registra je port, na ktorom DevACTs klient počúva jeho požiadavky. Názov registra z ktorého sa port číta je `code_crutches_port`.

Prostredníctvom tohto portu rozšírenie komunikuje s DevACTs aplikáciou, ktorá mu poskytuje metódu prostredníctvom REST API pre autentifikovanie správy a preposlanie do komponentu WebManagement.

2.2 Komponent CodeCrutches WebManagement

Tento komponent vystavuje REST API pre prácu s projektami a s úlohami. Volania na toto rozhrania musia byť autentifikované menom a heslom používateľa DevACT, ktoré sa pridáva v komponente DevACT Client. V súčasnosti sa využívajú operácie na:

Práca s projektami:

- Načítanie zoznamu projektom, ku ktorým má používateľ prístup

Práca s úlohami:

- Vytvorenie novej úlohy v ITM
- Aktualizácia stavu úlohy
- Načítanie úloh

Pre bližšie informácie o tomto module a práca ITM Maintenance viď. Modul - WebManagement

3 Dátový model

3.1 Úloha v ITM reprezentácií

Komentár od [OČ4]: Aktualizovať model pre úlohu

Model úlohy je uložený v *ProjectTask*, pre ktorý máme implementovaný vlastný konvertor ProjectTaskConvertor. Táto trieda nám poskytuje štruktúru na uloženie potrebných informácií o úlohe.

3.1.1 Opis polí

- Id – má len get metódu. Je to posledná časť TaskUri.
- Name – názov úlohy
- EstimatedEndDate = predpokladaný dátum ukončenia úlohy
- State - stav úlohy (bližšie opísaný v 3.1.2)
- ProjectId - identifikátor projektu, ku ktorému je úloha priradená
- AssignedTo – meno, komu je priradený task
- RemainingWork – koľko hodín treba do ukončenia úlohy
- CompletedWork – koľko hodín sa pracovalo na danej úlohe
- OriginalEstimateWork – odhadovaný počet hodín na začiatku
- TaskUri – identifikátor, ktorý linkuje issuetracker
- Author – meno programátora, ktorý má úlohu pridelenú

3.1.2 Životný cyklus

Úloha sa môže nachádzať v troch stavoch:

- *New* – úloha bola vytvorená a programátor na nej ešte nezačal pracovať
- *Active* – úloha bola vytvorená a programátor na nej pracuje
- *Paused* – programátor v súčasnosti nepracuje na danej úlohe
- *Done* – programátor označil danú úlohu za ukončenú

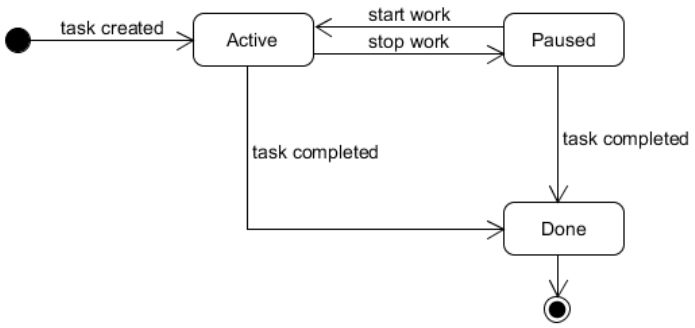

Diagram 2 Životný cyklus úlohy

4 Rozšírené používateľské rozhranie

4.1 Rozšírenie pre vzhľad okraja editora

Definovanie vlastného znaku na okraji editora v triede *CodeMarkerGlyphFactory*, ktorá implementuje rozhranie IGlyphFactory s metódou GenerateGlyph, ktorá vracia nadefinovaný výzor nového elementu používateľského rozhrania.

V triede CodeMarkerGlyphFactoryProvider zadefinujeme vzťah medzi naším vytvoreným znakom a značky zobrazujúcej sa na okraji editora.

CodeMarkerTag zaobaľuje model CodeTag, ktorý obsahuje informácie získané z ITM značky opisujúce kód v danom súbore.

CodeMarkerTagger zabezpečuje že sa dané značky pre daný súbor správne zobrazia na požadovanom riadku.

CodeMarkerTaggerProvider potom vystavuje novo vytvorený tagger.

4.2 Rozšírenie pre zvýraznenie textu editora

Toto rozšírenie spolupracuje s predošlým. Po vybraní zvolenej značky má za úlohu toto rozšírenie zvýrazniť text, na ktorý sa viaže daná značka.

CodeMarkerTagTextHighlighting obsahuje funkcionalitu tohto rozšírenia; výpočet oblasti na zvýraznenie, komunikáciu s taggerom a vykresľovanie zvýraznenia.

CodeMarkerTagTextHighlightingTextViewCreation slúži na vytváranie highlightingu pri každom otvorení dokumentu.

HighlightingManager vystavuje dostupné highlightre. Každý novovytvorený highlighter sa zaregistruje u HighlightingManagera. HighlighterManager si drží informácie o každom highlighteri a zároveň vie poskytnúť aktuálny highlighter k danému aktívnemu dokumentu.