

# Programátorská dokumentace

V repozitáři můžete najít 2 projekty, **CzeTex** a **CzeTex.Tests**, kde **CzeTex** je samotný program a **CzeTex.Tests** je projekt, který spouští testy.

## CzeTex

Projekt i zdrojový kód je rozdělen do dvou částí. První se zabývá zpracováním vstupních souborů, vystavěním pomocných datových struktur nebo poskytují pomocné metody. Druhá část se zabývá generováním výsledného .pdf souboru pomocí knihovny **iText**.

### První část

Do první části bychom mohl zahrnout následující soubory **Commander.cs**, **Files.cs**, **SetupLoader.cs**, **Trie.cs** a **Util.cs**. Pojďme si tyto soubory a jejich obsah projít jednotlivě.

#### Commander.cs

V souboru **Commander.cs** můžeme najít stejnojmennou třídu, která představuje centrum celého programu, volá dílčí funkce a třídy a udržuje koherentnost celého běhu. Zpracovává obsah načtený ze vstupního souboru, volá třídu pro zpracování setup .json souboru.

Pro zpracování vstupního souboru se využívá metoda **ReadContent**, která detekuje speciální znak, v základu / , a následně extrahuje funkce a jejich parametry z textu, vyhledá v pomocné datové struktuře **trie** odkaz na příslušnou funkci a zavolají.

Třída obsahuje několik funkcí jako **ReadRestForParameter**, **ReadParameter atd.**, které se navzájem doplňují v různých případech pro korektní vyhledání parametrů textových CzeTex funkcí.

#### Files.cs

Dalším souborem je **Files.cs** obsahující stejnojmennou třídu, která zabezpečuje vyhledání souboru podle poskytnutého parametru od uživatele a následné korektní otevření, načtení obsahu a uzavření.

Mimo těchto funkcionalit nabízí třída Files také možnosti generování cestu k umístění výsledného .pdf souboru v metodě **GetPdfPath** nebo metodu **GetBaseName**, která vrací jméno bez přípony vstupního souboru a další.

#### SetupLoader.cs

Dalším souborem je **SetupLoader.cs** obsahující 2 třídy **JsonEntry** a **SetupLoader**.

Jak už název napovídá **JsonEntry** reprezentuje jeden záznam v setup .json souboru. Má atributy pro **addFunction**, **getFunction** zda je má být funkce dynamicky generována a potom případně znak pro dynamicky generované funkce.

Třída **SetupLoader** načítá z setup JSON souboru mapování podle kterého bude následně přiřazovat

konkrétní textové CzeTex funkce k jejich kódovým ekvivalentům. Jak už bylo zmíněno, tak si funkce můžeme rozdělit na dynamicky generované a již předvytvořené. V metodě **AddFunctions** procházíme přes všechny záznamy v JSON souborů, tedy již vytvořené instance třídy `JsonEntry`. A následně rozhodujeme, zda má být funkce dynamicky generované nebo ne, dle příznaky v **dynamicallyGenerated**.

Když vyhodnotíme, že daná funkce musí být ještě vytvořená, tak přejdeme do větve programu v metodě **AddDynamicallyGeneratedFunction**, kde příslušné funkce pro vytvoření delegátů pro `add` a `get` funkce. A následně dané funkce přidáme to **trie** stromu. Pozn: Tyto dynamické funkce jsou využívány pouze pro funkce zobrazující nějaký znak, proto jsou silně spjaté s generací těchto funkcí. Ale možné rozšíření je jednoduché pouze by bylo potřeba volat jinou funkci pro delegáty.

Pro funkce, které se nacházejí v kódu, vyhodnotíme zda má daná funkce mít pouze svou `add` a `get` funkci nebo pouze `add`, následně už pomocí funkce **CreateDelegate** a potřebnými parametry vytvoříme delegáty pro tyto funkce a obdobně jako při dynamických funkcí je přidáme do **trie** stromu.