

Výroba nových DLL spolupracujúcich s našim klientom a serverom

Je potrebné stiahnuť najnovšiu verziu nášho nuggetu z devacts serveru.

Následne keď sa na ňu odkážeme, implementovaním JobExecution z daného nuggetu budeme mať základnú funkčnú verziu DLL.

JobExecution obsahuje niekoľko metód, ktoré je možné prekonať, ale ich základná implementácia je potrebná pre správnu komunikáciu s klientom alebo serverom. V prípade potrebných zmien na nich je dobré silno sa inšpirovať ich implementáciou.

Konkrétne ide o metódy:

`AfterStop()`

`SendTriples()`

Pre bežné úpravy v JobExecution existujú virtual metódy určené na to byť prekonávané. Konkrétne ide o metódy:

`BeforeStart()` - vykoná sa práve raz pred spustením samotnej funkcionality DLL

`AfterStart()` - vykoná sa práve raz po spustení DLL (nie synchrónne!)

`BeforeRestart()` - vykoná sa práve raz pred reštartovaním DLL (Zatiaľ nevyužitá funkcionality)

`BeforeStop()` - vykoná sa práve raz pred zastavením DLL

`BeforeStepProcessed()` - vykoná sa pravidelne pred každým vykonaným krokom sťahovania

`AfterStepProcessed()` - vykoná sa pravidelne po každom vykonanom kroku sťahovania

`DownloadStep()` - vykoná sa raz v každom kroku sťahovania. Ide o metódu samotného sťahovania, ktorá by mala vracať IGraph s trojicami. Hlavná funkcionality DLL sa očakáva práve tu.

Config items nastavované na serveri do DLL prichádzajú v premennej configuration, ktorá je typu `Dictionary<string, List<object>>`. Pod jedným value sa môže nachádzať viac ako jedna hodnota v zozname podľa potreby. Na začiatku spúšťania DLL je potrebné očakávané config itemy vyparsovať do lokálnych premenných.

V prípade zastavenia vykonávania z dôvodu prekorenia limitu na repozitári je potrebné nastaviť flag `stopExecution` na `false` a do `limitTime` premennej nastaviť očakávaný čas kedy už bude sťahovanie opäť možné.

V prípade akéhokolvek zastavenia je aj potrebné mať nastavenú hodnotu `downloadedStep` na posledný úspešne dokončený krok sťahovania.

V prípade potreby je možné zimplementovať priamo IJobExecution, ale je potrebné použiť metódy `AfterStop()` a `SendTriples()` z JobExecution pre validnú komunikáciu.

Po výrobe DLL a jeho zbuildovaní je potrebné stiahnuť zipper zo stránky a zabaliť DLL aj so všetkými referencovanými balíčkami za pomoci tohto zippera.