



Prvé kroky pri vývoji Pluginov

V tejto lekcii si povieme, aké sú prvé kroky vývojára pri vývoji Pluginov pre integráciu externých systémov.

Pluginy pre integráciu externých systémov sú komponenty prepájajúce Systém C4 s externým systémom. Tým môže byť akékoľvek zariadenie alebo softvér, ktorý umožňuje integráciu. Koncepcia týchto Pluginov umožňuje integráciu komunikačných protokolov prostredníctvom TCP/IP, UDP/IP, RS232 alebo vzdialených volaní Rest API, či volaní SQL do databázy.

V rámci integrácie externých systémov Systém C4 rozlišuje dva typy Pluginov. Prvý typ sa nazýva Driver. Komunikuje s externým systémom nepretržite od naštartovania komunikácie až po jej zastavenie. Driver slúži väčšinou na komunikáciu s bezpečnostnými systémami, ktoré je potrebné monitorovať a riadiť nepretržite.

Druhý typ sa nazýva IS Connector a je určený najmä na spracovanie a výmenu veľkého objemu dát medzi Systémom C4 a externým systémom. Využíva sa hlavne na plánovanú synchronizáciu personálnych dát s informačnými systémami podniku zákazníka, napríklad personálnym systémom Active Directory, mzdovým systémom, či HR systémom.

Teraz sa zameriame na vývoj Pluginu typu Driver.

Úlohou Drivera je monitorovať a ovládať jednotlivé časti externého zariadenia v prostredí Systému C4.

Externým systémom je v našich lekciách aplikácia, ktorá emuluje jednoduché zariadenia s jedným vstupom a jedným výstupom. Na tomto príklade budeme demonštrovať, ako takéto zariadenie integrovať do Systému C4. Typ komunikačného protokolu závisí od konkrétneho zariadenia, v tomto prípade ide o binárny protokol. Popis protokolu je súčasťou zdrojových kódov k Demo Driveru.

Každé zariadenie sa skladá z niekoľkých adresných prvkov (podsystémov), ktoré je možné jednoznačne identifikovať v rámci komunikačného protokolu. Prvým z nich je v našom prípade vstupný bod pre komunikáciu so zariadením. Druhým je centrálna jednotka, ktorá zabezpečuje logiku zariadenia. Ďalšími podsystémami sú





vstup a výstup. V tomto prípade pôjde o nevyvážený dvojstavový vstup. Keďže každý z podsystémov musí byť adresovateľný, v našom prípade vstup má adresu 1 a výstup adresu 1.

Teraz si povieme, v akom prostredí sa Driver vyvíja. V štandardnom operačnom režime sú Drivery hostované a spravované prostredníctvom aplikácie Plugin Runner. Každá instancia Plugin Runnera hostuje Drivery jedného typu. V rámci jedného Plugin Runnera môže bežať neobmedzený počet Driverov. Plugin Runnery sú manažované prostredníctvom aplikácie Plugin Manager, ktorá je v rámci architektúry Systému C4 spustená ako samostatný servis.

V aplikácii Plugin Runner je možné spúšťať len Drivery s licenciami. Keďže licencie sa vydávajú až na dovyvíjané Drivery, je vývoj Drivera v prostredí Plugin Runnera veľkou výzvou. Z tohto dôvodu Gamanet vyvinul aplikáciu Developer Console, ktorá pri vývoji Drivera emuluje aplikácie Plugin Runner a Plugin Manager. Je v nej umožnené spustenie len jednej instancie Drivera v danom čase, ktorý obsluhuje jedno integrované zariadenie. Výhodou je, že v Developer Console táto instancia Drivera nie je kontrolovaná z pohľadu licencií. Developer Console poskytuje vývojárom prostredie, kde môžu komfortne vyvíjať a ladiť Driver.

Vývoj Drivera pozostáva zo štyroch krokov: registrácia, inicializácia, vývoj a publikovanie.

V prvom kroku vývojár zaregistruje vývoj nového Drivera.

Táto registrácia sa vykonáva na Portálových službách spoločnosti Gamanet, v sekcii Plugin Store.

Vývojár musí následne zadefinovať štruktúru svojho Drivera v rámci sekcie mySDK.

Pre vytvorenie novej definície Drivera je nutné zadefinovať všetky kľúčové parametre Drivera. Je potrebné určiť spoločnosť, ktorá vlastní práva na daný Driver, a taktiež zadefinovať parametre prostredia, v rámci ktorého bude môcť Driver bezproblémovo pracovať.

Na adrese my.c4portal.com/sdk/devicetypes môžeme nájsť prehľad všetkých podsystémov, ktoré Gamanet poskytuje vývojárom na vyskladanie reprezentácie zariadenia, ktoré budú integrovať. Ak chcete integrovať podsystém, ktorý tu nie je uvedený, kontaktujte Gamanet na adrese sdk@gamanet.com a požiadajte o jeho





registráciu.

Každé zariadenie má svoju grafickú reprezentáciu vo forme ikoniek v rôznych veľkostiach a formátoch pre jednotlivé časti aplikácie.

Následne sa zadefinujú všetky podsystémy daného zariadenia. Každý podsystém, ktorý chceme samostatne monitorovať a ovládať, tu musí byť zaregistrovaný. Dôležitým krokom je identifikácia licenčnej politiky pre vyvíjaný Driver. Licencia sa definuje na jeden alebo viac podsystémov. V prípade viacerých licencovaných podsystémov Systém C4 považuje všetky za rovnocenné. Zákazník kupuje vždy jeden typ licencie pre jeden Driver, bez ohľadu na to, na koľko typov podsystémov bude licencia aplikovaná. Odporúča sa vždy zvoliť jeden licencovaný podsystém, najvhodnejším je riadiaca jednotka. Umožňuje to optimálnu škálovateľnosť licenčných poplatkov a ušetrí zákazníkovi čas a prácu v súvislosti s licenciami pri zmenách na inštalácii.

Keď je vytvorený zoznam podsystémov, v ďalšom kroku musíme zadefinovať vzťahy medzi jednotlivými podsystémami.

Štartovacím bodom je zvyčajne definovaná komunikačná jednotka, v našom prípade reprezentovaná ethernetovým vstupom. Pod ňou je v našom prípade centrálna jednotka a nasledujú vstup a výstup.

V tomto kroku vlastne určujeme, ako bude môcť byť definovaná topológia zariadenia na zákazníckej inštalácii. Na základe tejto definície sa v Systéme C4 následne objavujú položky v zozname po kliknutí pravým tlačidlom myši na zariadenie, v menu "Add".

V prípade, že potrebujeme zadefinovať možnosť pridávania podsystému pod viaceré prvky, najprv si zadefinujeme jeho reálnu reprezentáciu tam, kde bude registrovaný. Následne môže byť zavedený do ďalších skupín ako shortcut, a to stlačením a podržaním tlačidla Ctrl a pretiahnutím myškou do danej skupiny.

V ďalšom kroku sa špecifikujú vlastnosti pre každý podsystém. Gamanet preddefinoval pre každý typ zariadenia sadu parametrov, ktoré vývojár môže ďalej upravovať, rozširovať či redukovať. Niektoré parametre, ako napr. adresa, sú však povinné.





Keď vznikne na zariadení udalosť, načíta sa prostredníctvom Drivera do Systému C4. V Driveri sa takáto udalosť musí vždy spárovať s konkrétnym podsystémom, ktorého sa týka.

Každé zariadenie má svoj vlastný zoznam typov udalostí, ktoré následne generuje do denníka udalostí. Úlohou Drivera je pretransformovať udalosti zo zariadenia do jednotného zoznamu udalostí Systému C4, a doplniť prípadné parametre týkajúce sa zariadenia – napríklad adresa podsystému, osoba, ktorej sa udalosť týka, prípadne čas vzniku udalosti.

Takto vygenerovanú udalosť Driver uloží do databázy.

Preddefinovaný zoznam typov udalostí registrovaných v Systéme C4 si vývojár môže pozrieť na my.c4portal.com/sdk/Messages. Sú štruktúrované podľa typu a severity tak, aby následne mohla byť táto hierarchia použitá na jednoduché filtrovanie a hromadné spracovanie v rámci automatizácie. Ak chcete do zoznamu pridať novú udalosť, je potrebné kontaktovať Gamanet na sdk@gamanet.com.

Pri vývoji Drivera musí vývojár nadefinovať pre každý podsystém všetky typy udalostí, ktoré na ňom môžu nastať. V prípade, že sú udalosti v rámci Drivera správne priradené ku konkrétnemu podsystému, v Systéme C4 sa tieto zobrazujú užívateľovi v rámci vybranej položky (podsystému) v strome zariadení. Strom zariadení tu slúži ako filter pre zobrazovanie len príslušných udalostí.

V ďalšom kroku vývojár pre každý podsystém nadefinuje zoznam príkazov, ktorými bude môcť užívateľ daný podsystém ovládať. Kompletný zoznam dostupných príkazov nájdete na my.c4portal.com/sdk/Messages. Ak je potrebné pridať príkaz, ktorý sa v zozname nenachádza, vývojár kontaktuje Gamanet so žiadosťou o registráciu nového príkazu.

Každé zariadenie graficky reprezentuje užívateľovi možné reálne stavy zariadenia. Každý stav je reprezentovaný určitou farbou, prípadne zvukom či dokonca blikaním, a každý stav má svoju unifikovanú stavovú ikonu. Stav taktiež môže byť reprezentovaný zmenou základnej ikony (napr. otvorené a zatvorené dvere).





Stavy sú preddefinované Gamanetom a úlohou vývojára je previesť v Driveri reálne stavy zariadenia do ich jednotnej reprezentácie v Systéme C4. Preto musí vývojár v rámci Drivera špecifikovať, ktoré stavy zariadenia bude jeho Driver podporovať.

Po zadefinovaní potrebných informácií si vývojár vyexportuje Visual Studio projekt a stiahne si ho do svojho počítača.

Každý Driver musí v zdrojovom kóde obsahovať určité definície potrebné na to, aby mohol bežať v prostredí Systému C4.

Portál mySDK tieto parametre predgeneruje na základe definícií, ktoré vývojár zadal. Pre uľahčenie vývoja umožňuje portál mySDK exportovať rozšírenú verziu zdrojových kódov s preddefinovanými časťami a odporučeniami kódov, ktoré má následne vývojár dopracovať. V prípade opakovaného exportu, napríklad po úprave definície Drivera na portáli mySDK, je vhodné exportovať len časti projektu, ktoré sú trvalo pod správou web generátora. Tieto časti sú v pomenovaní vždy označené ako "generated".

Export "Driver Project Initial", vygeneruje projekt s doporučenými krokmi pre vývojára. Export "Driver Project Update" vygeneruje už len súbory s označením "generated".

Druhým krokom je inicializácia projektu vo vývojovom prostredí na počítači vývojára.

V rámci tohto kroku je potrebné aktualizovať potrebné knižnice z NuGet servera. Po skompilovaní zdrojových kódov Drivera vygenerovaných mySDK webom je potrebné ho zaregistrovať do Systému C4, pre potreby jeho konfigurácie v ďalších krokoch počas vývoja.

Po nahratí na C4 Server sa inštalačný balíček rozbalí a nainštaluje serverové časti Drivera. Taktiež poskytne inštaláciu klientskych častí do C4 Clienta.

Tento krok je možné vykonať po spustení v prostredí Developer Console, ktorá je v rámci projektu zadefinovaná ako debugovacia aplikácia pre vývoj Drivera.

Po zaregistrovaní balíka v Developer Console si môžete overiť, či bola registrácia vášho Drivera úspešná. Zoznam všetkých zaregistrovaných Driverov nájdete v C4 Clientovi, v záložkách "Settings" a "Packages".

Následne vývojár zadefinuje topológiu demo setu externého systému do Systému C4.

Demo set by mal obsahovať všetky podsystémy, aby vývojár mohol otestovať maximálne množstvo situácií, ktoré môžu pri reálnom nasadení systému nastať.





Ako sme už spomínali, každý Driver beží v preddefinovanom Plugin Manageri. Je dôležité nastaviť, aby Driver, ktorý budeme vyvíjať, bežal v prostredí Developer Consoly. Taktiež je potrebné nastaviť lokálnu IP adresu a port demo zariadenia. V rámci konfigurácie pridáme centrálnu jednotku, jeden vstup a jeden výstup, presne podľa konštrukcie daného zariadenia.

V ďalšom kroku si v Developer Console stiahneme túto konfiguráciu, a následne môžeme spustiť ladenie Drivera.

Ďalším krokom je samotný vývoj Drivera.

Hlavnou úlohou vývojára pri vývoji Drivera je implementácia komunikačného protokolu, ktorý zabezpečí komunikáciu Drivera s externým systémom. Vývojár naimplementuje všetky funkcionality požadované zákazníkom. Definuje rozsah príkazov na ovládanie externého systému, definuje a implementuje všetky typov stavov, ktoré pre externý systém môžu nastať, ako aj ich grafickú reprezentáciu v prostredí Systému C4. Taktiež implementuje zápis všetkých udalostí získaných z externého systému do centrálnej databázy Systému C4.

Tejto téme sa budeme bližšie venovať v samostatnej lekcii.

Keď je vývoj Drivera dokončený, vývojár ho publikuje pre používanie u koncových zákazníkov.

Driver je potrebné prostredníctvom Developer Consoly publikovať na Portálových službách, v sekcii Plugin Store. Driver môže následne testovať Gamanet QA oddelenie za účelom jeho verifikácie, pripadne výrobca externého systému za účelom certifikácie Drivera.

Takto otestovaný Driver je k dispozícii na implementáciu do projektov.