8.Zmeny verejného API medzi verziami 10.0.37 a 10.1.39

Verejné API sa medzi verziami 10.0.37 (`d2japi-37.jar`) a 10.1.39 (`d2japi-10.1.39.jar`) rozšírilo a v niektorých prípadoch aj zmenilo natoko, že sú pri prechode na novšiu verziu potrebné runé úpravy existujúceho kódu. (Pozn. verzia 10.1.38 bola vydaná len pre vnútorné potreby a nie je jej v tejto migranej príruke venovaná pozornos.) Nasledujúcich kapitolách je uvedený zoznam zmien, nových funkcií a návod, ako správne upravi aplikaný kód pri migrácii aplikácie z D2000 verzie 10.0.37 na 10.1.39.

8.1. Odlenenie balíka sk.ipesoft.d2000.async

Pre lepšie logické usporiadanie tried bol vytvorený balíek `sk.ipesoft.d2000.async` ktorý obsahuje niektoré rozhrania, ktoré sa predtým nachádzali v balíku `sk.ipesoft.d2000.d2japi.` Prehad zmien dokumentuje nasledujúca tabuka:

trieda/rozhranie	vo verzii 10.0.37	vo verzii 10.1.39
`FutureEvent`	`sk.ipesoft.d2000.d2japi`	`sk.ipesoft.d2000.async`
`FutureEventHandler`	`sk.ipesoft.d2000.d2japi`	`sk.ipesoft.d2000.async`
`AdjustableFuture`	`sk.ipesoft.d2000.d2japi.common`	`sk.ipesoft.d2000.async.base`

> Pri migrácii je potrebné upravi všetky príkazy import, ktoré odkazujú na spomínané triedy/rozhrania a všetky absolútne odkazy.

8.2. Zabezpeené a reverzné pripojenie JAPI

Trieda D2Japi predstavuje vstupný bod do používania knižnice JAPI. Používa sa na pripojenie sa k bežiacemu D2000 kernelu. Pripojenie sprostredkuje proces `D2Connector.exe`. Vo verzii 10.0.37 bolo možné pripojenie sa iba jednou metódou `createConnector (String, D2ConnectorEventsListener)`. Pripojenie inicioval klient JAPI a pripájal sa na poúvajúci socket procesu.

Verzia 10.1.39 pridala metódu `startListeningForConnection(String , int, ServerSocketEventsListener)`. Metóda otvorí poúvajúci socket na strane JAPI klienta, ktorý aká na pripojenie iniciované procesom `D2Connector.exe`.

Od verzie 10.1.39 je možné zabezpei komunikáciu medzi JAPI klientom a `D2Connector.exe` protokolom TLS verzie 1.2. Slúžia na to preaženia metód `createConnector` a `startListeningForConnection` s parametrom `String certificatePath`, ktorý uvádza cestu k súboru s certifikátom, ktorý je použitý na zabezpeenie spojenia.

8.3. Zdieané zdroje systému D2000

Zdieané zdroje systému D2000 sú

- stavové texty,
- systémové texty (pomenovania konštánt),
- transformaná paleta,
- slovník,

- definície štruktúr.

Od verzie 10.1.39 je možné získa ich aktuálnu konfiguráciu a notifikácie o zmenách hodnôt. Slúži na to preaženie metódy `D2Connector.createSession` s parametrom `SharedResourcesListener sharedResourcesListener`.

8.4. Použitie konkrétnych Table tried namiesto AnonymousTable

Štruktúrovaná UNIVAL hodnota je v knižnici JAPI reprezentovaná objektom typu `UnivalRecord`. Atribúty UNIVAL-u sú v tomto prípade málo zaujímavé, hodnota je alej zabalená objektom triedy `Table`, ktorý je sprístupnený volaním `getValue`. Trieda `sk.ipesoft.d2000.datatable.Table` je abstraktná a pre jednoduchú tvorbu inštancií slúži odvodená trieda `AnonymousTable`, ktorú v základnej konfigurácii používa aj knižnica JAPI pri dekódovaní. Nevýhodou tejto triedy je, že neponúka prístup k jednotlivým bunkám pomocou identifikátora stpca a neumožuje ich statickú typovú kontrolu.

Od verzie 10.1.39 je možné volaním `D2Connector.registerTableClass` zaregistrova vlastnú implementáciu triedy `Table` (odporúané je dedi z triedy `TableBase`). Konkrétna implementácia vždy reprezentuje jednu konkrétnu definíciu štruktúry s ktorou je spojená anotáciou `@StructureTableDefinition`. Po takejto registrácii bude JAPI dekódova príslušné štruktúrované hodnoty ako inštancie registrovanej triedy namiesto inštancií `AnonymousTable`.

> **Pozor**: Po takejto registrácii sa JAPI pri dekódovaní spolieha na to, že poet a typy stpcov registrovanej triedy zodpovedajú definícii štruktúry. Ak sa odlišujú, dekódovanie pravdepodobne skoní výnimkou. K odlišnosti môže dôjs napríklad aj neskoršou zmenou konfigurácie definície štruktúry v nástroji CNF. V takom prípade je potrebné implementáciu príslušnej triedy upravi a aplikáciu nanovo skompilova.

8.5. Podpora odahenej verzie štruktúrovanej UNIVAL hodnoty

V súvislosti s vývojom projektu *D2000 Web Suite* bola knižnine podporená konverzia inštancie triedy `Table` na `List<TableRowBean>` a spä. `TableRowBean` je abstraktná trieda predstavujúca odahenú štruktúrovanú hodnotu s rozhraním zodpovedajúcim konvencii Java Beans. Pre dosiahnutie dobrej statickej typovej kontroly kódu bol rozšírený zoznam generických parametrov viacerých tried. Vo všetkých uvedených triedach ako aj v ich potomkoch je nový generický parameter definovaný ako `B extends TableRowBean`.

- `Table<T, R, B>`
- `TableRow<T, R, B>`
- `Column<I, S, T, R, B, C, E>`
- `TableCell<I, S, T, R, B, C, E>`

> Pri migrácii oznaí všetky miesta s nesprávnym potom generických parametrov. Oprava spoíva v doplnení chýbajúceho generického parametra:
>
> - Pri získaní odkazu na existujúci objekt bez známeho typu pridaním „wildcard" parametra: `Column<?, ?, ?, ?, ?, ?, ?> column = record.getValue().getColumn(1);`
> - Pri vytváraní objektu špecifikovaním všeobecného typu `TableRowBean: UnivalRecord<AnonymousTable, AnonymousTableRow, TableRowBean> record = new UnivalRecordValue<>(...);`

8.6. Rozšírenie ítania dát z archívu o ArchiveObjectDescription

ítanie dát z archívu bolo rozšírené o konfiguráciu príslušného archívneho bodu. Konfigurácia je zapuzdrená rozhraním `ArchiveObjectDescription`. Z objektu je možné preíta informácie o názve objektu, jeho popise, technických jednotkách, type hodnoty, periodicite a iných informáciách potrebných pre zobrazenie hodnoty pravidlami systému D2000.

8.6.1. Rozšírenie rozhrania ArchiveDataListener

Pribudla metóda `void onArchiveObjectDescription(ArchiveObjectDescription)`, ktorou implementátor získa konfiguráciu. Ak nenastala chyba pri prístupe do archívu, metóda je zavolaná vždy ako prvá. Samotné dáta z archívu sú volaním metódy `onArchiveData` sprístupnené až následne.

> Pri migrácii je potrebné rozšíri všetky implementácie rozhrania `ArchiveDataListener` o spomínanú metódu. Implementácia metódy môže by prázdna.

8.6.2. Rozšírenie rozhrania ArchiveDataProvider

Rozhranie `ArchiveDataProvider` umožuje alternatívny prístup k dátam z archívu. Bolo rozšírené o metódu `getDescription()`, ktorá sprístupuje konfiguráciu príslušného archívneho bodu. Táto zmena je spätne kompatibilná.

Okrem toho boli obidve preaženia metódy `waitForData()` doplnené o kontrolované výnimky `InterruptedException`, `D2JapiException`, ktoré umožujú zachyti vínimoné stavy, ktoré pri akaní na dáta mohli nasta. Je to zmena správania oproti predošlej verzii, kde boli tieto výnimoné stavy v tichosti zamlané.

Pri migrácii je potrebné blok kódu, ktorý tieto dáta spracúva obali do try... catch... alebo deklarova kontrolované výnimky v metóde, ktorá k dátam pristupuje.

8.7. Výpoet archívnej štatistickej funkcie

Od verzie 10.1.39 je podporené volanie archívnych štatistických funkcií z JAPI. Ide o ekvivalent ESL akcie `CALCSTATFUNCARR`. Slúži na to volanie `D2Session.getArchiveValuesThroughStatisticalFunction`.

8.8. Rozšírenie rozhrania AuditDataListener

Rozhranie `AuditDataListener` bolo rozšírené o metódu void `onError(String comment)`, ktorá je zavolaná, ak pri prístupe k dátam došlo k chybe.

Pri migrácii je potrebné doplni všetky implementácie o danú metódu. Implementácia metódy môže by prázdna, ale jej zavolaním sa signalizuje ukonenie zasielania samotných dát.

8.9. Úprava logiky operácie close v rozhraniach `D2Object`, `D2Cono`, `D2ProcessList`, `D2Session`

Tieto rozhrania sú si navzájom podobné v tom, že reprezentujú uritý zdroj platformy D2000, ktorý v ase mení svoj stav a registrovaným klientom o tom v reálnom ase posiela notifikácie. Posiela ich tak dlho, kým klient nepožiada o ukonenie odberu notifikácií, alebo kým daný zdroj nezanikne. Za distribúciu notifikácií ako aj udržiavanie zoznamu registrovaných odberateov je zodpovedný Kernel. (V prípade CONO je zodpovedný samotný proces produkujúci správy). V knižnici JAPI slúži na zrušenie registrácie v týchto rozhraniach metóda `close()`.

Vo verzii 10.0.37 sa kanál považoval za uzatvorený zavolaním metódy `close()` okamžite. To však nezodpovedá stavu v systéme D2000, pretože zavolaním metódy sa len odosiela správa so žiadosou o zrušenie registrácie. V ase medzi zavolaním `close()` a spracovaním správy kernelom ešte môžu prichádza notifikácie o zmene hodnoty.

Z toho dôvodu je signatúra metódy vo verzii 10.1.39 `FutureEvent<Void> close()`. Volaním operácie sa kanál neuzatvorí automaticky, ale až vtedy, ke je doruená odpove potvrdzujúca uzatvorenie. Dovtedy ešte stále môžu prichádza notifikácie. Doruenie potvrdenia je signalizované tým, že návratový `Future` objekt nadobudne hodnotu.

Pri migrácii je nutné sa uisti, že po zavolaní close()sa následné príkazy nespoliehajú, že bol kanál okamžite uzatvorený. Pokia je v kóde takéto miesto, je potrebné pred ním spravi nasledovné:

```
FutureEvent<Void> futureCloseResult = openedObject.close();


... // tento kód sa na uzatvorenie ešte nespolieha


futureResult.get(); // akanie na potvrdenie uzatvorenia

... // nasleduje miesto, na ktorom už musí plati, že bol kanál
skutone uzatvorený
```

8.10. Rozšírenie rozhrania D2SessionEventsListener

Rozhranie bolo rozšírené o 2 nové metódy:

`void onTerminateRequest()` – je zavolaná v prípade, že niekto prostredníctvom systému D2000 žiada o ukonenie procesu aktuálnej JAPI Session. Napríklad cez TELL príkaz. Korektná implementácia metódy zariadi, aby došlo k uvoneniu zdrojov a následne zavolá `session.close()`.

`void onRedundancyStateChanged(RedundancyStateType redundancyState)` – je zavolaná v prípade prepnutia redundancie D2000 kernela. Hodnota parametra signalizuje aktuálny stav kernela, ku ktorému je pripojená aktuálna session. Korektná implementácia zabezpeí, že sa na komunikáciu so systémom D2000 bude používa session, ktorá je pripojená na HOT kernel.

Pri migrácii je potrebné všetky implementácie rozšíri o príslušné metódy.

8.11. Volanie RPC implementujúcej ESL rozhranie

Od verzie 10.1.39 pribudla možnos vola RPC, ktorá je implementáciou ESL rozhrania. Slúžia na to preažené metódy `D2Session.callRPC` a `D2Session.callRPCNoReply` s parametrom `interfaceHobj`.

8.12. Test existencie RPC a SBA volania

Vzhadom k dynamickej povahe konfigurácie D2000 je od verzie 10.1.39 možné za behu overi, i sa predpokladaná RPC a SBA nachádza na predpokladanom mieste. Slúžia na to metódy `D2Session.testRPC a D2Session.testSBA`.

9.13. Informácie o prihlásenom používateovi a zmena hesla používatea

Aby klient knižnice JAPI mohol komunikova s D2000 kernelom, je potrebné, aby mal k dispozícii session, v ktorej je prihlásený platným D2000 používateským kontom. Pri prihlasovaní sa cez JAPI podliehajú používateské kontá rovnakým pravidlám, ako pri prihlasovaní sa cez iné D2000 procesy. Napríklad povinnosti pravidelne meni svoje prihlasovacie heslo. Z toho dôvodu od verzie 10.1.39 pribudlo volanie D2Session.getLoggedUserInformation, ktoré sprístupní informácie o prihlásenom používateovi a stave jeho prihlasovacieho hesla.

Pre možnos zmeny prihlasovacieho hesla prihláseného používatea pribudla od verzie 10.1.39 metóda `D2Session.changePassword`, ktorá zmenu hesla umožuje. Nové heslo musí spa všetky pravidlá, ktoré sú pre prihlasovacie heslá v systéme D2000 zapnuté.

> Pri migrácii je odporúané implementova funkcionalitu, ktorá skontroluje stav prihlasovacieho hesla. V prípade, že kernel požaduje zmenu hesla, upozorní používatea na túto skutonos a umožní mu heslo zmeni. V opanom prípade hrozí zablokovanie používateského útu.

9.14. Zmena správania pri neúspešnom vytvorení `D2Session`

Vytvorenie objektu typu D2Session volaním `D2Connector.createSession` môže zlyha z niekokých príin. Od verzie 10.1.39 sú tieto príiny lepšie rozlíšené rôznymi typmi výnimiek, ktoré nastanú pri volaní `createSession`, alebo pri volaní `Future.get`.

- Ak inštancia `D2Connector`-a už nemá aktívne spojenie: `IllegalStateException`
- Ak kernel vyhodnotí parametre `sessionName` alebo `hostName` ako invalidné: `ConnectSessionException`
- Ak kernel odmietne prihlási session s uvedenými prihlasovacími údajmi: `LoginFailedException`
  - Ak bolo prihlasovacie meno správne, obsahuje výnimka `PasswordInformation`, v ktorom je uvedený dôvod odmietnutia.
  - V opanom prípade je dôvod `null`.

> Pri migrácii je odporúané obali získanie session do try ... catch ... so zachytením príslušných výnimiek.

9.15. Volanie `D2Session.getIndirectInfo`

Volanie `D2Session.getObjectInfo` umožuje získa prístup ku konfigurácii objektu, ktorá je užitoná zvláš pri zobrazovaní hodnoty objektu používateovi. V prípade štruktúrovanej premennej s hodnotami získanými z odkazov na iné objekty je potrebné nasledova „*indirect*" odkazy. Od verzie 10.1.39 je k dispozícii volanie `D2Session.getIndirectInfo`, ktoré požiada kernel o dohadanie informácií na jedno zavolanie.

9.16. Volanie `D2Session.getObjectInfo` vráti aktuálnu hodnotu objektu

Od verzie 10.1.39 je súasou výsledku volania `D2Session.getObjectInfo` aj aktuálna hodnota objektu.

9.17. Volanie `D2Session.openObject` vráti informácie o objekte

Od verzie 10.1.39 sú súasou výsledku volania `D2Session.openObject` aj informácie o objekte.

9.18. Zmena rozhrania `RPCReturnContext`

Inštancie objektu `RPCReturnContext` sa používajú na zaslanie návratovej hodnoty prichádzajúceho RPC volania (ke ESL zavolá CALL RPC do JAPI). Vo verzii 10.1.39 bolo jej rozhranie a logika ovládania zásadne prepracovaná.

- Metóda `doReturnError` berie ako parameter hodnotu vymenovaného typu `RpcReturnContextErrorType` namiesto íselného kódu (ktorý nebol dokumentovaný). Použitím metódy sa signalizuje, že RPC bola zavolaná nesprávnym spôsobom – nesprávny poet alebo typy parametrov volania.

- Pribudla metóda `doReturnException` ktorou sa signalizuje, že poas vykonávania RPC nastala výnimka.
- Metóda `doReturn`, ktorej parametrom bolo pole hodnôt typu `UnivalValue` zanikla.
- Nahrádza ju metóda `getResponseDataSet`, ktorá sprístupní inštanciu typu `RpcResponseDataSet`.

Vo verzii 10.0.37 bola návratová hodnota zasielaná ako pole hodnôt typu `UnivalValue`. Nevýhodou tohto prístupu bolo, že džka poa ani typy jednotlivých Unival hodnôt v poli neboli nijak kontrolované, ale ESL interpreter pritom spoliehal na ich korentnos.

Z toho dôvodu sa od verzie 10.1.39 návratová hodnota zadáva do objektu typu `RpcResponseDataSet`, ktorý kontroluje typy hodnôt ako aj ich poet.

> Pri migrácii je potrebné upravi volanie `RPCReturnContext.doReturn(values)` na `RPCReturnContext.getResponseDataSet().setValues(values)`, priom je pred tým vhodné skontrolova, i je zadané pole v premennej values správnej vekosti a obsahuje hodnoty požadovaných typov. Taktiež je potrebné upravi parameter volania `RPCReturnContext.doReturnError` z ísla na hodnoty vymenovaného typu `RpcReturnContextErrorType`.

9.19. Zmena rozhrania `SBAReturnContext`

Inštancie objektu `SBAReturnContext` sa používajú na zaslanie návratovej hodnoty prichádzajúceho SBA volania z *D2000 Java Runtime* do JAPI:

- Metóda `doReturnError` berie ako parameter hodnotu vymenovaného typu `SBAErrorCode` namiesto íselného kódu (ktorý nebol dokumentovaný). Použitím metódy sa signalizuje, že SBA bola zavolaná nesprávnym spôsobom a nebude ani vykonaná.
- Pribudla metóda `doReturnException` ktorou sa signalizuje, že poas vykonávania SBA nastala výnimka.

> Pri migrácii je potrebné upravi parameter volania `SBAReturnContext.doReturnError` z íselnej hodnoty na hodnotu vymenovaného typu `SBAErrorCode`

9.20. Zastaraná `D2SessionUtils.getObjectHOBJByName`

Od verzie 10.1.39 bola zavedená metóda `D2SessionUtils.getFutureObjectHobjByName`, ktorá namiesto hotového výsledku vráti `FutureEvent`, ktorého výsledok je požadované HOBJ. Typicky je potrebné získa viac HOBJ a použi ich spolu. Pôvodná metóda vrátila výsledok až ke získala odpove z *Kernel*-a, o znane spomauje beh programu. Nová metóda umožuje rýchlo zasla viac žiadostí a poka na odpovede spolu.

> Pri migrácii je odporúané upravi kód na používanie novej metódy.