

# NOVINKY ABRA BI 13.0.15–13.0.20

Datum: 15. července 2020

## OBSAH

1.	DŮLEŽITÁ UPOZORNĚNÍ.....	3
1.1	OD VERZE 13.0.17 NEJSOU SOUČÁSTÍ DISTRIBUCE ABRA BI JDBC OVLADAČE PRO PLATFORMY ORACLE, MYSQL A DB2.....	3
1.1.1	SYMPTOMY .....	3
1.1.2	ŘEŠENÍ .....	4
1.1.2.1	ORACLE, DB2.....	4
1.1.2.2	MYSQL.....	4
2.	SEZNAM ZMĚN VE VERZÍCH 13.0.15 – 13.0.20.....	5
2.1	PODPORA PRÁCE SE SELDAT .....	5
2.2	NOTIFIKACE PRO AKTIVNÍ UŽIVATELE ABRA BI .....	7
2.3	BAREVNÉ ODLIŠENÍ ZÁPORNÝCH ČÍSEL V DXPIVOT.....	8
2.4	NOVÉ FUNKCE STRÁNEK APLIKACÍ V ABRA BI.....	9
2.5	OBLÍBENÉ STRÁNKY APLIKACÍ V ABRA BI .....	11
2.6	ROZŠÍŘENÍ VOLÁNÍ AKCÍ O MOŽNOST PŘIDAT VLASTNÍ PARAMETRY .....	13
2.7	UNIVERZÁLNÍ SQL PŘÍKAZY NEZÁVISLÉ NA DATABÁZOVÉM SERVERU .....	15
2.8	ZMĚNY V PLÁNOVANÉ ÚLOZE TYPU ODESLÁNÍ E-MAILU S PŘÍLOHOU.....	19
2.9	DXDATAGRID – NOVÁ GENERACE TABULKY .....	22
2.9.1	RŮZNÉ MOŽNOSTI ZOBRAZENÍ SLOUPCŮ.....	25
2.9.1.1	PROSTÉ PŘEVZETÍ SLOUPCE Z DEFINICE FIELDŮ V DATOVÉM ZDROJI .....	25
2.9.1.2	PŘEVZETÍ SLOUPCE Z DEFINICE FIELDŮ V DATOVÉM ZDROJI A ÚPRAVA VLASTNOSTÍ 25	
2.9.1.3	MOŽNOSTI DODATEČNÉ KONFIGURACE.....	25
2.9.1.4	OBJECT DATA.....	26
2.9.1.5	OBJECT ROWDATA.....	26
2.9.2	SPECIÁLNÍ TYPY ZOBRAZENÍ SLOUPCŮ.....	27
2.9.2.1	TYPY DIFF A DIFFWITHPERCENTS .....	27
2.9.2.2	TYP PICTURE.....	27
2.9.2.3	TYPY PERCENTS A PERCENTSWITHTEXT .....	28
2.9.2.4	TYPY DATE, TIME, DATETIME, DATEFULL A DATETIMEFULL .....	29
2.9.2.5	TYPY DATEINTERVAL, DURATION A DURATIONFULL .....	29
2.9.2.6	TYPY ISO8601DATE A ISO8601DATEFULL .....	29
2.9.2.7	TYP CUSTOM.....	30

## 1. DŮLEŽITÁ UPOZORNĚNÍ

### 1.1 OD VERZE 13.0.17 NEJSOU SOUČÁSTÍ DISTRIBUCE ABRA BI JDBC OVLADAČE PRO PLATFORMY ORACLE, MYSQL A DB2

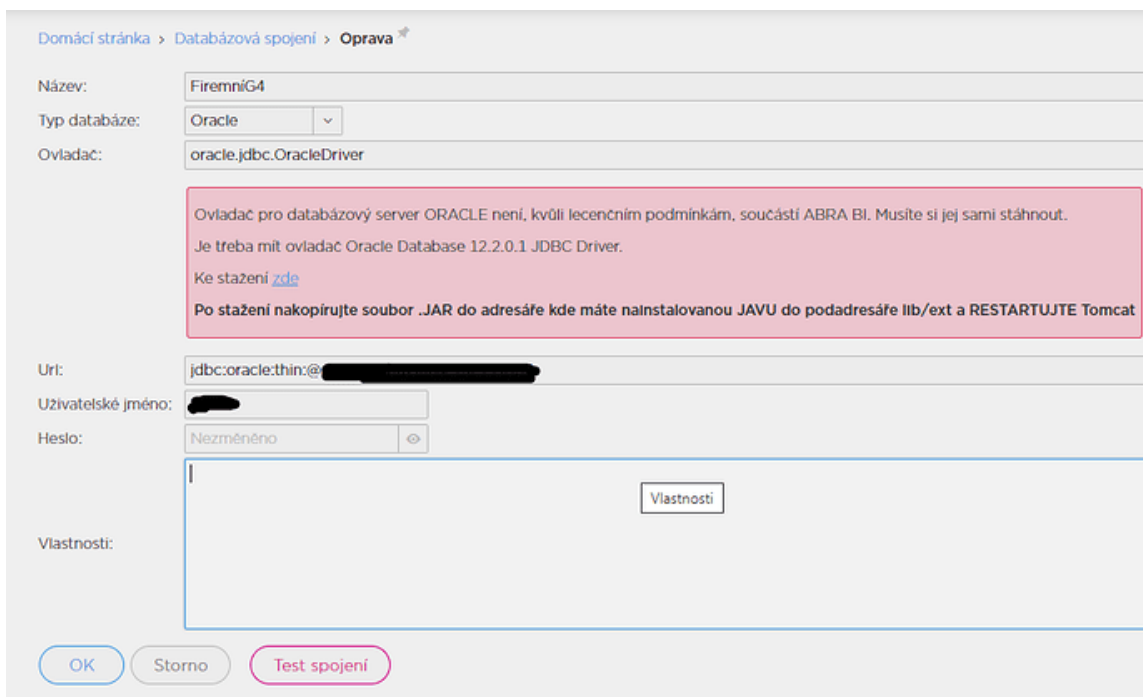
Z důvodu restriktivní licenční politiky společností Oracle a IBM není možné distribuovat jejich JDBC ovladače společně s ABRA BI.

- **Oracle:** každý uživatel musí před stažením ovladače odsouhlasit licenční podmínky. Soubor ke stažení naleznete na webových stránkách společnosti Oracle [ZDE](#) ([ojdbc8.jar](#)).
- **MySQL:** licence, pod kterou je distribuován JDBC ovladač MySQL, není kompatibilní s licencí, pod kterou je distribuován produkt ABRA BI. (Ovladač je distribuován pod licencí [GPL](#) a tudíž jej není možné distribuovat s produkty s uzavřeným kódem.)
- **DB2:** není zřejmé, pod jakou licencí je JDBC ovladač distribuován a proto implicitně předpokládáme, že jej s ABRA BI není možné distribuovat. Soubor ke stažení naleznete na webových stránkách společnosti IBM zde: [DB2 Version 11.1 JDBC 4.0 Driver verze 4.23.42](#).

#### 1.1.1 SYMPTOMY

Problém se projevuje následovně:

- Začne se objevovat chyba **ClassNotFoundException**.
- V agendě Databázová spojení naleznete informaci, že ovladač není k dispozici.



Domácí stránka > Databázová spojení > Oprava <sup>\*</sup>

Název: FiremníG4

Typ databáze: Oracle

Ovladač: oracle.jdbc.OracleDriver

Ovladač pro databázový server ORACLE není, kvůli licenčním podmínkám, součástí ABRA BI. Musíte si jej sami stáhnout.  
 Je třeba mít ovladač Oracle Database 12.2.0.1 JDBC Driver.  
 Ke stažení [zde](#)  
 Po stažení nakopírujte soubor .JAR do adresáře kde máte nainstalovanou JAVU do podadresáře lib/ext a RESTARTUJTE Tomcat

Uri: jdbc:oracle:thin:@[redacted]

Uživatelské jméno: [redacted]

Heslo: Nezměněno

Vlastnosti: [Vlastnosti]

OK Storno Test spojení

## 1.1.2 ŘEŠENÍ

### 1.1.2.1 ORACLE, DB2

Podle instrukcí si stáhněte JDBC ovladač (jeden nebo více souborů s příponou .jar), nakopírujte jej do umístění, kde je nainstalovaná Java (do složky lib/ext) a restartujte Tomcat.

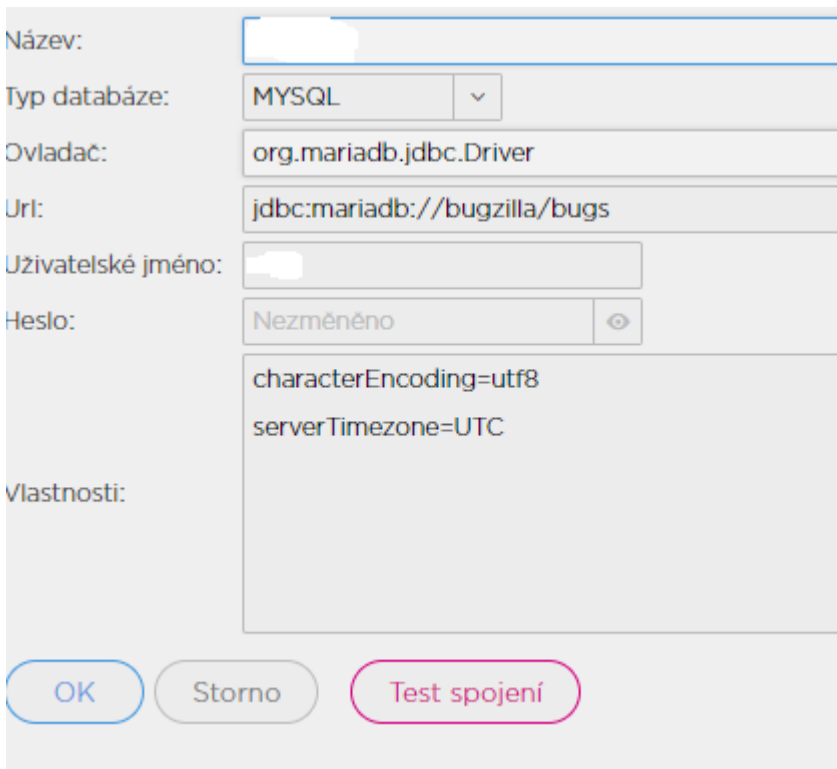
Pokud složka lib/ext neexistuje (používá se v distribuci JRE, pokud používáte JDK, složka existovat nemusí), můžete tyto .jar soubory nahrát do umístění, kde je nainstalovaný Tomcat (do složky lib) a opět provést restart Tomcatu.

### 1.1.2.2 MYSQL

U MySQL je situace jednodušší. JDBC MySQL byl sice z distribuce odstraněn, ale byl nahrazen alternativním MariaDB ovladačem.

U nových spojení se automaticky předvyplňuje nový MariaDB ovladač. U stávajících spojení stačí v poli Ovladač uvést místo původní MySQL třídy novou třídu pro MariaDB, viz obrázek (org.mariadb.jdbc.Driver) a opravit URL.

Případně si je možné stáhnout JDBC ovladač pro MySQL podobně jako pro Oracle a nakopírovat ho do stejného umístění.



The image shows a configuration dialog box for a database connection. The fields are as follows:

- Název:** [Redacted]
- Typ databáze:** MySQL (dropdown menu)
- Ovladač:** org.mariadb.jdbc.Driver
- Uri:** jdbc:mariadb://bugzilla/bugs
- Uživatelské jméno:** [Redacted]
- Heslo:** Nezměněno (password field with eye icon)
- Vlastnosti:** characterEncoding=utf8  
serverTimezone=UTC

At the bottom, there are three buttons: OK, Storno, and Test spojení.

## 2. SEZNAM ZMĚN VE VERZÍCH 13.0.15 – 13.0.20

### 2.1 PODPORA PRÁCE SE SELDAT

Abra BI nově umožňuje pracovat se SELDAT (seznam SELDAT je umístěn na každé SQL komponentě). Zde je možné přidávat komponenty, které jsou poskytovateli SELDAT informací.

Zatím se jedná o dvě komponenty/poskytovatele:

- **SQLSeldat**
- **ScriptedSeldat**

Obě komponenty mají společné vlastnosti:

- **SelDatName** – jedná se o jméno Seldat definice, které bude použito v SQL textu jako místo, kam se vloží vygenerované ID pro vybrané záznamy.
- **Namespace** – specifikuje tabulkový prostor, kde se pracuje s tabulkami SELDAT. Zatím je podpora pouze pro Gx.

Komponenty se liší pouze způsobem, jakým získávají ID záznamů pro SELDAT:

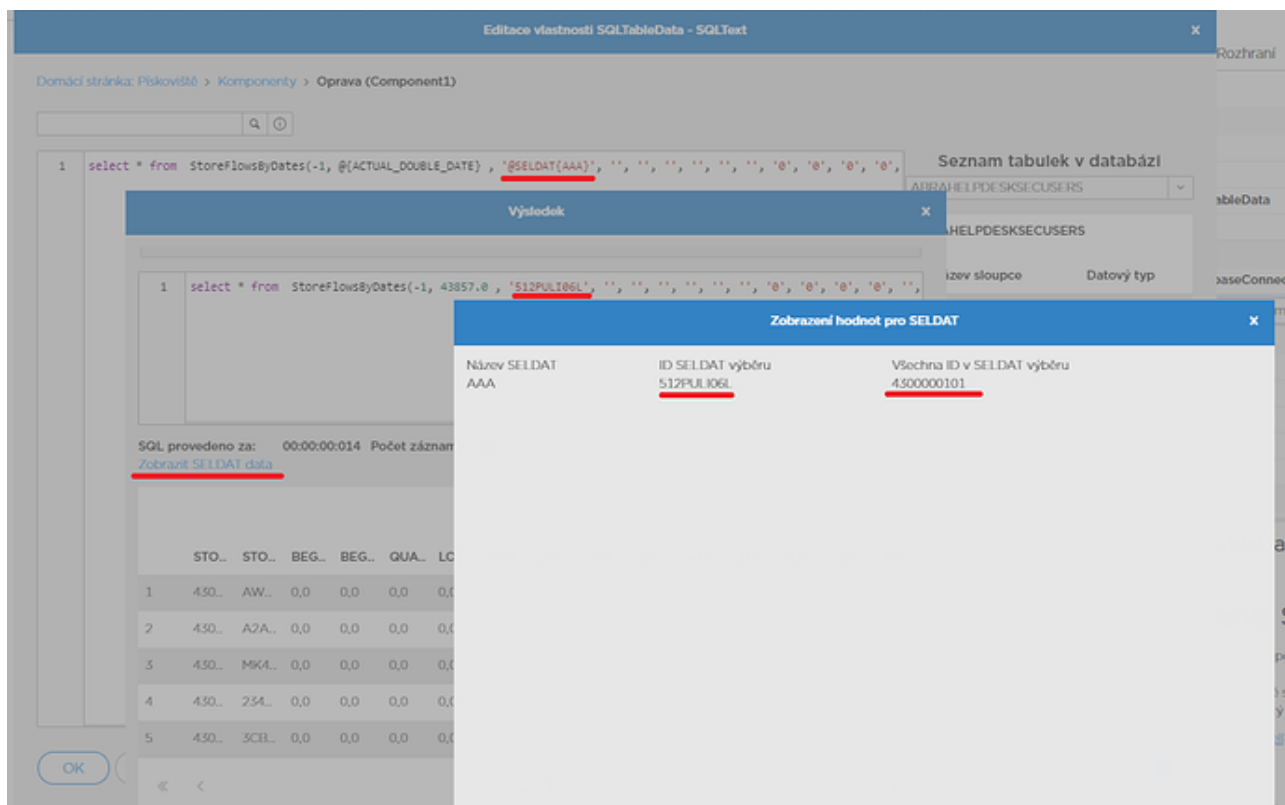
- **SQLSeldat** je načte pomocí SQL dotazu do databáze, který musí vrátit všechna požadovaná ID.
- **ScriptedSeldat** odkazuje na skript, který musí vrátit instanci třídy List obsahující všechna požadovaná ID.

SQL pak může vypadat např. následovně:

```
select * from StoreFlowsByDates(-1, @{ACTUAL_DOUBLE_DATE} ,
'@SELDAT{AAA}', '', '', '', '', '', '', '0', '0', '0', '0', '', '') A
```

kde část @SELDAT{AAA} odkazuje na komponentu, jejíž SelDatName je „AAA“. Právě na toto místo bude později vloženo příslušné ID pro SELDAT. Celý kód je v příkladu obalen apostrofy, protože ID zde musí být v apostrofech, ale to záleží na konkrétním způsobu použití.

Zároveň došlo k rozšíření funkčnosti SQL Editoru, který umí takovéto SQL provést a také zobrazit ID, která byla pro jednotlivé SELDAT definice načtena a uložena do databáze. Viz obrázek.



The screenshot shows the SQL Editor interface with the following elements:

- SQL Editor:** A query is entered: `select * from StoreFlowsByDates(-1, @{ACTUAL_DOUBLE_DATE} , '@SELDAT{AAA}', '', '', '', '', '', '0', '0', '0', '0', '', '') A`. The `@SELDAT{AAA}` part is highlighted in red.
- Seznam tabulek v databázi:** A dropdown menu showing database tables like `ADBAHELPDESKSECUSERS`.
- Výsledek:** A table showing the execution results of the query.
- Zobrazení hodnot pro SELDAT:** A dialog box with the following data:
 

Název SELDAT	ID SELDAT výběru	Všechna ID v SELDAT výběru
AAA	<u>512PUL106L</u>	<u>4300000101</u>
- SQL provedeno za:** 00:00:00:014 Počet záznamů: 5
- Zobrazit SELDAT data:** A button to view the SELDAT data.
- Table Data:** A table with columns `STO..`, `STO..`, `BEG..`, `BEG..`, `QUA..`, `LC` and rows of data.

## 2.2 NOTIFIKACE PRO AKTIVNÍ UŽIVATELE ABRA BI

Od verze 13.0.15 je možné zaslat všem aktivním uživatelům aplikace ABRA BI hromadnou notifikaci. Notifikace se následně zobrazí v podobě lišty se zprávou pod hlavní lištou aplikace v horní části obrazovky.

Administrátor tak může uživatelům např. sdělit, že se chystá restart systému a že by si měli uložit práci:



Pro nastavení této notifikace přibyla v sekci **Základní nastavení** položka **Notifikace uživatelům**. V této agendě je možné **zapnout/vypnout** zobrazení notifikace, nastavit **text**, **typ** (barva pozadí lišty) a **rychlost zobrazení** notifikace.



Nastavení **Rychlost zobrazení** notifikace určuje, jak často se bude webový klient dotazovat na server pro obdržení notifikace. Čím kratší čas, tím dříve se uživatelé o notifikaci dozvědí. Zároveň se ale více zatěžuje server, protože každá BI záložka každého webového prohlížeče zasílá na server po každém uplynutí intervalu nový dotaz. Hodnota se uvádí v sekundách, výchozí hodnota je **30**, povolené rozmezí **10 – 120 sekund**.

**Typ** notifikace může v současné době nabývat čtyř různých hodnot: INFO (modrá), SUCCESS (zelená), WARNING (žlutá), ERROR (červená).

## 2.3 BAREVNÉ ODLIŠENÍ ZÁPORNÝCH ČÍSEL V DXPIVOT

Od verze 13.0.15 jsou v DXPivot číselné buňky rozlišeny CSS třídami podle toho, jestli obsahují kladné nebo záporné číslo (positive-number/negative-number). Na základě této CSS třídy pak lze číslo například obarvit červeně, viz uvedený CSS styl a obrázek.

```
.dx-pivotgrid-container .dx-pivotgrid-area-data td.negative-number span {
    color: red;
}
```

CSS styl lze zadat přímo do CSS v aplikaci (pak bude styl aplikován na všechny pivoty), nebo lze použít komponentu **ComponentCSSStyleSheet** a tu nasměrovat na Layout, který obsahuje pivot. Pak bude styl uplatněn pouze na tento konkrétní pivot.

Složení										
Firma										
Třída										
Skupina										
Systémka										
Období										
Číslo										
Kvartál										
Měsíc										
Den										
Typ účtu	Typ zaúčtování	Účet	Číslo dokladu	* 2015 - Fiskální rok 2015	* 2016 - Fiskální rok 2016	* 2017 - Fiskální rok 2017	* 2018 - Fiskální rok 2018	* 2019 - Fiskální rok 2019	* 2020 - Fiskální rok 2020	Celkem
*									-605 292,68	-605 292,68
* Aktivi				21 557 783,77	20 288 566,09	8 048 005,97	5 923 466,39	16 491 642,11	16 396 729,95	128 706 584,28
* Nákladov				87 658 652,26	171 967 120,09	182 647 219,74	200 496 560,63	399 734 272,38	121 291 119,42	965 794 944,32
* Pasivní				-19 357 945,06	-9 471 731,26	4 525 552,48	7 615 529,52	-6 359 411,57	-48 086 562,18	-71 134 587,07
* Podrozvahov				0,00	0,00	-0,00	-0,00	0,00	-15 600,00	-15 600,00
* Výnosov				-89 858 490,97	-184 781 954,90	-195 220 758,19	-214 035 556,54	-209 886 502,72	-128 902 385,51	-1 022 747 648,85
Celkem				0,00	-0,00	0,00	0,00	0,00	0,00	-0,00

**POZOR:** Komponentu **ComponentCSSStyleSheet** nelze nasměrovat přímo na DXPivot, neboť pivot se v prohlížeči vytváří se zpožděním a styl by pak nefungoval.



## 2.4 NOVÉ FUNKCE STRÁNEK APLIKACÍ V ABRA BI

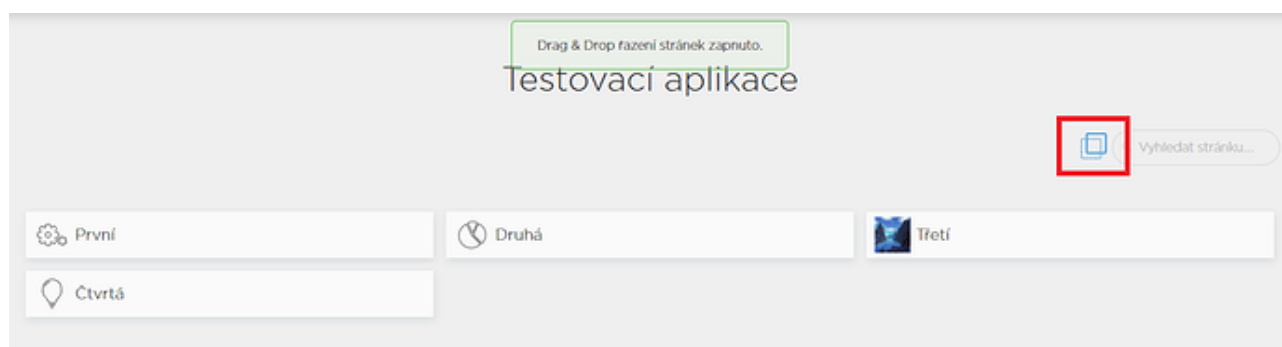
Od verze 13.0.15 lze v ABRA BI stránkám přiřadit také obrázek – stejně, jako je tomu v případě aplikací:

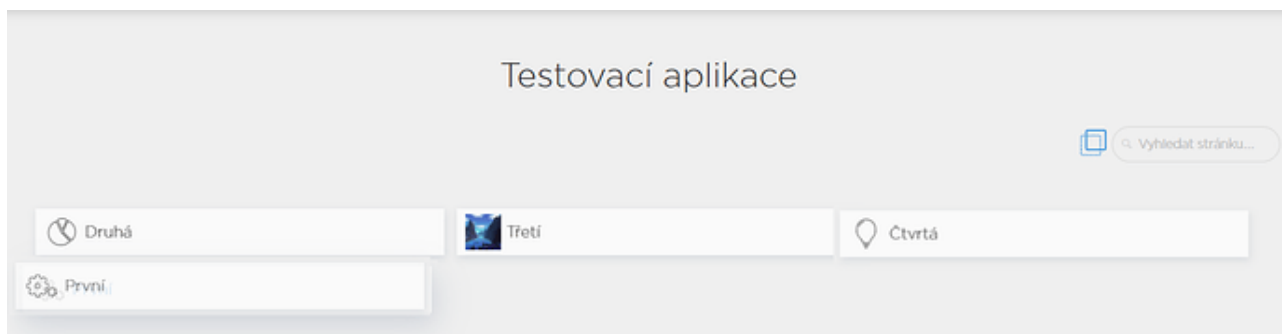


Tyto obrázky by neměly být příliš velké, protože jsou zatím vidět pouze v listu stránek aplikace, kde mají velikost ikony:

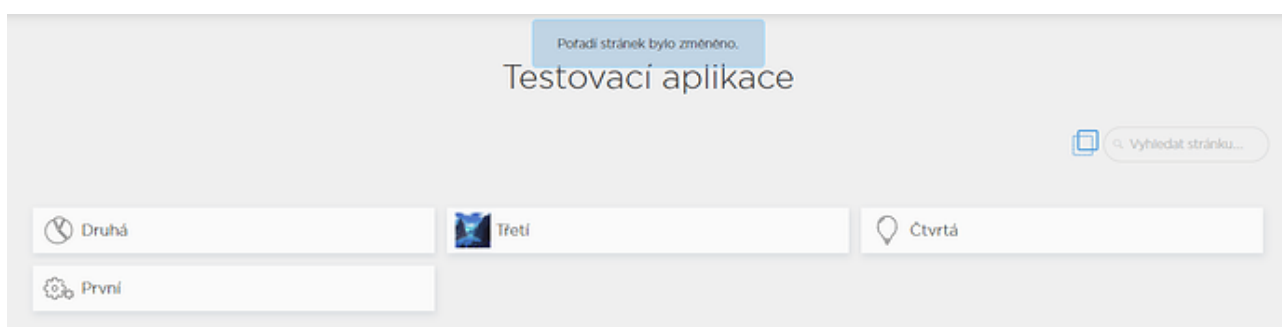


Na této stránce je také možné jednotlivé stránky aplikace seřadit pomocí **Drag and Drop** v případě, že přihlášený uživatel je admin, nebo má admin práva. Toto řazení je nejdříve nutné zapnout pomocí DnD ikonky:





V případě úspěšného přesunutí stránky na jinou pozici vyskočí popup notifikace, která o události informuje:



Změna pořadí je tímto zachována pro všechny uživatele ABRA BI.

## 2.5 OBLÍBENÉ STRÁNKY APLIKACÍ V ABRA BI

Od verze 13.0.16 je možné označit libovolnou stránku jakékoli aplikace jako oblíbenou. Všechny tyto stránky se následně zobrazují ve fiktivní aplikaci **Oblíbené**. Tato aplikace se automaticky zobrazí v případě existence alespoň jedné oblíbené stránky.

Oblíbené stránky se vztahují ke konkrétnímu uživateli. Admin, nebo nepřihlášený uživatel možnost nastavování oblíbených stránek a „aplikací“ **Oblíbené** vůbec nevidí.

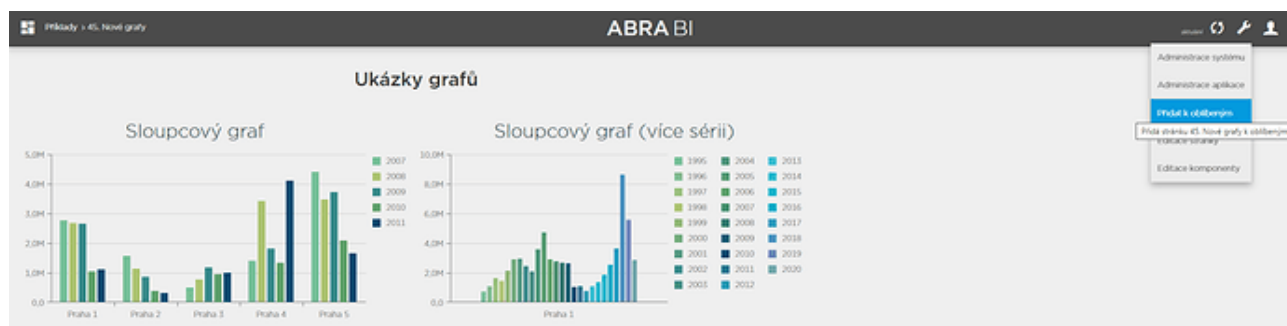
Přidání do oblíbených / odebrání stránek z oblíbených je možné provádět na dvou místech.

1. V seznamu stránek konkrétní aplikace:



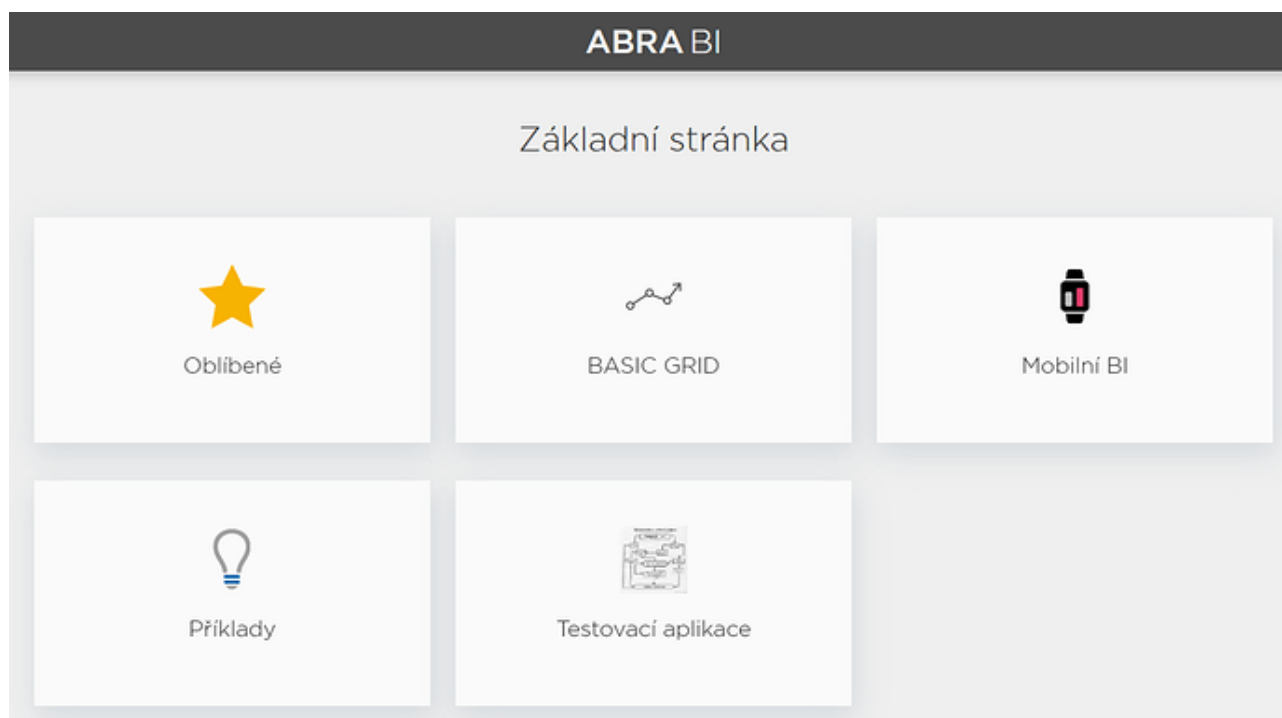
Již oblíbené stránky mají příznak žluté hvězdičky. Šedá hvězdička pro přidání stránky se zobrazí až po najetí myši na konkrétní stránku.

2. Přímou na zobrazené stránce:



Příslušnou volbu naleznete pod ikonkou maticového klíče.

Do fiktivní aplikace **Oblíbené** se pak dostanete z přehledu dlaždic všech aplikací:



Dlaždice oblíbených stránek se zobrazuje vždy na prvním místě.

V této „aplikaci“ už vidíte pouze naše oblíbené stránky:



I zde je samozřejmě možné stránky vyhledávat, nebo z oblíbených odstranit, jak je vidět z obrázku (uživatel kliknul na hvězdičku u stránky **04. Report s obrázky**).

## 2.6 ROZŠÍŘENÍ VOLÁNÍ AKCÍ O MOŽNOST PŘIDAT VLASTNÍ PARAMETRY

V ABRA BI se často objevuje potřeba vykonávat „podobné“ akce. Příklad – komponenta obsahuje pět tlačítek a každé otevírá stránku Bugzilly (nástroj používaný vývojářskými týmy pro tzv. bugtracking) ovšem každé s mírně odlišnými parametry v URL.

Dosud bylo jediným možným řešením vytvoření pět tlačítek, přičemž každé tlačítko odkazovalo na svůj vlastní Multiaction a příslušná multiaction prováděla vždy mírně odlišný skript, který nastavil požadované URL parametry a následně použil společnou akci showPage.

Celé řešení tedy zahrnovalo:

- 5 × tlačítko
- 5 × multiakci
- 5 × skript
- 1 × showPage

– celkem 16 komponent.

Přitom by stačilo, aby skript dokázal na základě parametru určit, které tlačítko uživatel stiskl a podle toho se zachovat. V tom případě by stačila jedna společná multiakce, v ní jeden společný skript a jedna showPage.

Proto byla místa, kde se odkazujeme na komponentu typu akce, rozšířena o možnost zadat parametr, který se pošle do odkazované akce. Takový parametr je nově možné zadat např. v tlačítku v události onClick. Jak je vidět na obrázku, při volání onClick budou dodatečně předány dva parametry. Parametr **tlacitko** s hodnotu **A** a parametr **value** s vyhodnocenou hodnotou **ACTUAL\_DATE**.

Parametry se předají do multiakce, která je následně předá všem svým akcím, takže se dostanou i do skriptu, který se volá právě z multiakce.

Events	
OnClick	<div style="border: 1px solid #ccc; padding: 2px;"> <span style="background-color: #e0e0e0; padding: 2px;">MultiAction</span> <span style="float: right;">→ ▾</span> </div> <div style="border: 1px solid #ccc; padding: 2px; margin-top: 2px;">           tlačitko=A;value=@{ACTUAL_DATE}           <span style="float: right;">✎</span> </div>
OnFinish	[none]
OnInit	[none]

[ImageButton](#) > Action

## Typ vlastnosti Action

Vlastnost typu Action je potomkem vlastnosti typu [Instance](#) a interně obsahuje objekt třídy [Identity](#).

Oproti vlastnosti typu [Instance](#) nedovoluje odkazovat na libovolnou vnitřní komponentu, ale pouze na komponentu, která implementuje rozhraní [Action](#).

Pokud nastavíme null hodnotu, nebude odkazovat na žádnou instanci komponenty.

Také je možné specifikovat dodatečné parametry, které budou předány do volané akce.

Parametry se zadávají jako [Template](#) ve formátu:

```
paramName1=paramValue1;paramName2=@{výraz}
```

Jako oddělovač lze použít `,` nebo `;` Parametry jsou ve volané akci k dispozici pod zvoleným jménem a vždy jsou typu String.

Interně je vlastnost implementována třídou [ActionPropertyValue](#).

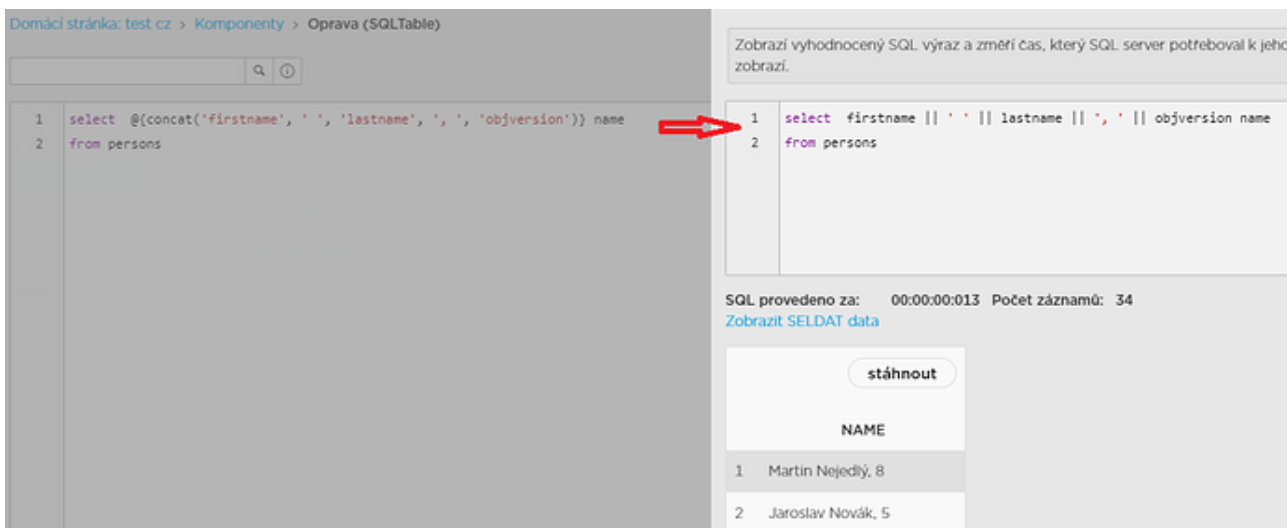
## 2.7 UNIVERZÁLNÍ SQL PŘÍKAZY NEZÁVISLÉ NA DATABÁZOVÉM SERVERU

Od verze 13.0.17 je možné ve vlastnostech typu **SQLText** sjednotit SQL syntaxi určitých funkcí pro všechny (8) typy databází, které ABRA BI podporuje.

Tyto společné funkce je zapotřebí v SQL textu vyhodnotit jako samostatný skript, tedy uzavřít je do standardního obalovače **@{ ... }**.

### 1. Funkce **concat(Object... objects)**

Tato funkce očekává 1–N libovolných parametrů, které chceme ve výsledném SELECTu řetězcově spojit do jednoho sloupce. Každá databáze používá pro spojování řetězců jiný operátor, nově však stačí použít funkci **concat()**, která bude např. pro Firebird převedena následovně:



Domácí stránka: test cz > Komponenty > Oprava (SQLTable)

Zobrazí vyhodnocený SQL výraz a změní čas, který SQL server potřeboval k jeho zobrazení.

```

1 select @{concat('firstname', ' ', 'lastname', ' ', 'objversion')} name
2 from persons

```

```

1 select firstname || ' ' || lastname || ' ' || objversion name
2 from persons

```

SQL provedeno za: 00:00:00:013 Počet záznamů: 34  
[Zobrazit SELDAT data](#)

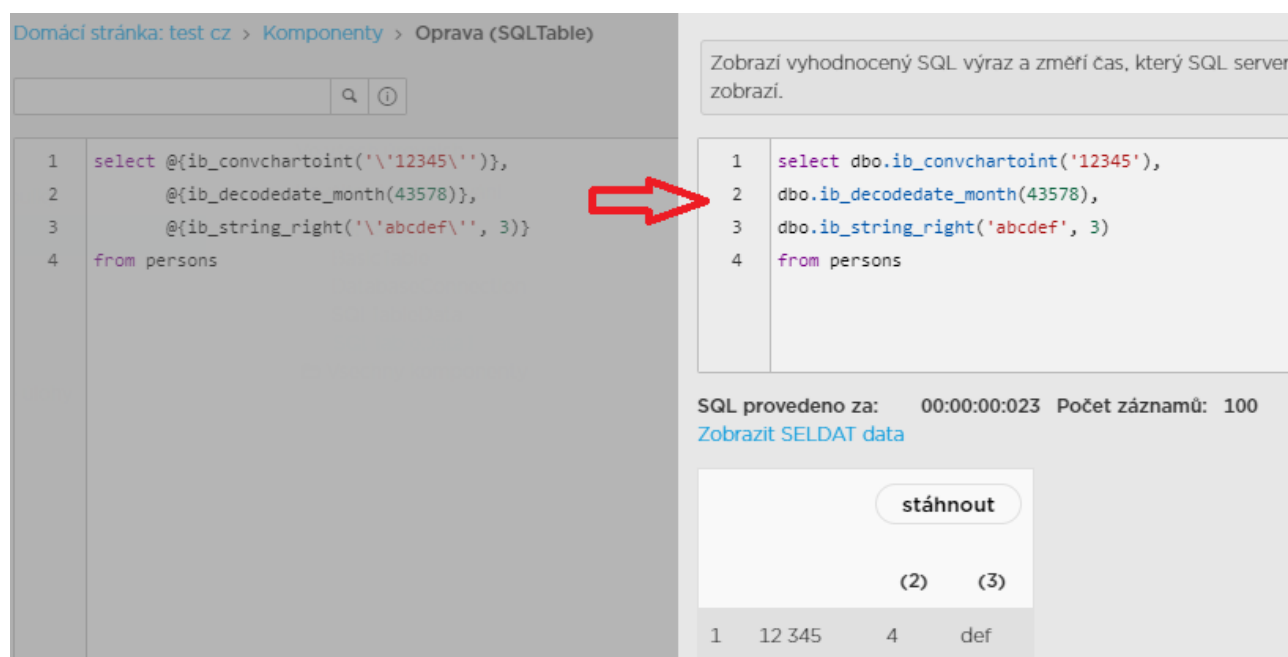
stáhnout

NAME	
1	Martin Nejedlý, 8
2	Jaroslav Novák, 5

Některé databáze používají místo operátoru **||** např. **+**. Jiné používají stejnojmennou funkci **CONCAT(Object... objects)**, jako např. MySQL nebo MS SQL.

## 2. UDF **ib\_()** funkce

**ib\_** funkce se např. na platformách Firebird a MS SQL volají jinak. Nově je možné libovolnou **ib\_** funkci zavolat zaobalením do skriptu. Toto zaobalení se opět postará o konverzi do syntaxe podporované konkrétním databázovým strojem:



The screenshot shows a SQL IDE interface. On the left, the original code uses placeholders: `@{ib_convchartoint('\12345\')}`, `@{ib_decodedate_month(43578)}`, and `@{ib_string_right('\abcdef\', 3)}`. A red arrow points to the right, where the transformed code is shown: `select dbo.ib_convchartoint('12345'),`, `dbo.ib_decodedate_month(43578),`, and `dbo.ib_string_right('abcdef', 3)`. Below the code, the execution time is 00:00:00:023 and the number of records is 100. A 'stáhnout' button and a table with 2 columns and 3 rows are also visible.

Z obrázku je patrné, že při zaobalení do skriptu je potřeba pro CHAR parametry používat dvoje uvozovky. Vnějšími uvozovkami říkáme, že budeme volat skript s parametrem datového typu String (Java) a vnitřními escapovanými uvozovkami `\ '`  docílíme toho, že se uvozovky doplní také ke stringu předávanému k vyhodnocení databázovému stroji.

## 3. Funkce **stringToDate()** a **dateToString()**

Protože je velmi často potřeba pracovat s položkami typu datum a čas v různých formátech, byly vytvořeny dvě funkce, které dovedou převést řetězec na objekt datumu a času konkrétní databáze a opačně, tedy převést datumový objekt konkrétní databáze na libovolný formátovaný řetězec.

Tyto funkce pro datumové operace se využijí nejspíše v případech čtení dat třetích stran, jako např. API Google Analytics, kde se datum a čas vrací ve formátu **YYYYMMDDHH**, nebo pro případ napojení do databáze, která používá klasické datumové sloupce. V GENU se pro hodnoty DOUBLE asi nejlépe hodí standardní **ib\_** funkce.

**a)** Funkce **stringToDate(String date, String format: Optional)** se snaží převést zadaný řetězec v prvním parametru na datumový objekt databáze. Druhý parametr **format** je volitelný, protože ne všechny databáze umožňují převést řetězec podle formátu na datum. Např. Firebird a MS SQL pouze očekávají hodnotu datumu v některém z podporovaných formátů. Pro ostatní databáze je druhý parametr (vstupní formát data) **povinný**. Je tak lepší ho pro jistotu uvádět a držet se nějakého standardního tvaru, např.: **stringToDate('2019.01.31 22:02:45', 'yyyy.MM.dd HH:mm:ss')**

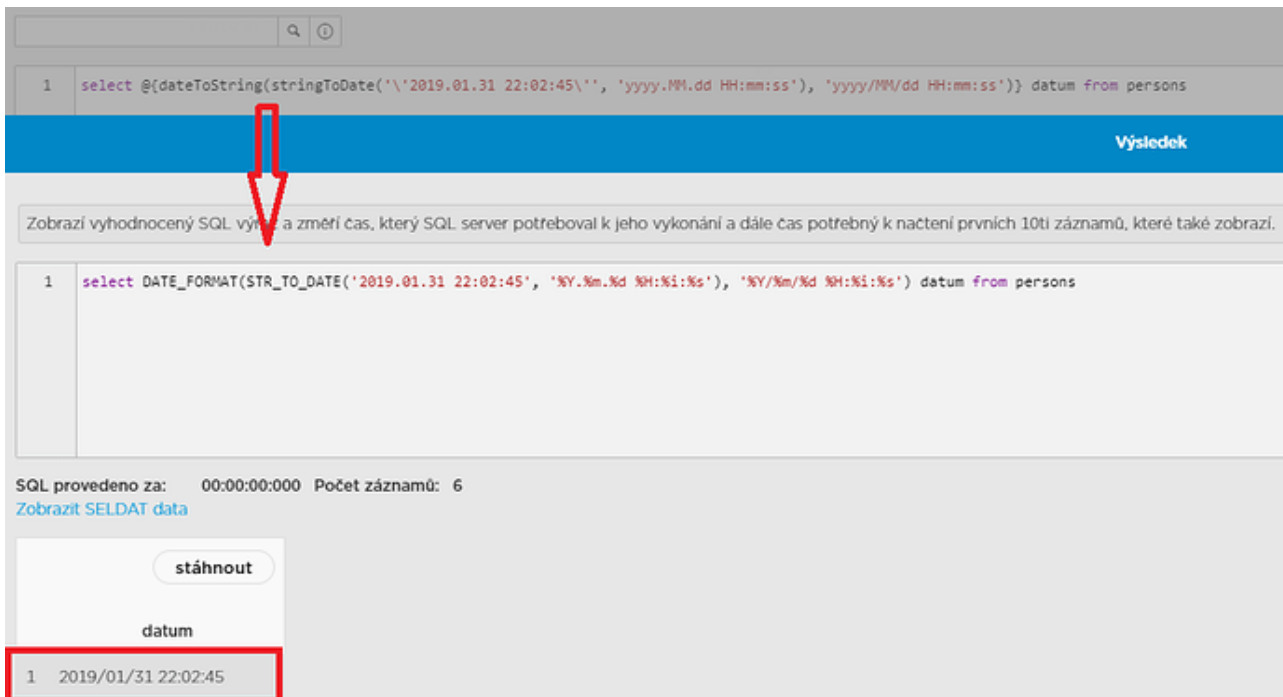


**b)** Funkce `dateToString(Date date, String format)` očekává vždy dva parametry. Prvním je datumový objekt databáze a druhým je formát výstupního řetězce.

Např.: `dateToString('date_column', 'dd/MM/yyyy HH:mm:ss')`. Výsledkem bude řetězec (CHAR) ve formátu **dd/MM/yyyy HH:mm:ss**.

Zástupné znaky jednotlivých částí datumu a času jsou standardní znaky pro formátování v jazyce **Java** a stejně tak databáze **MS SQL** a **H2**, u kterých **záleží na velikosti písmen**.

Příklad složeného volání pro **MySQL**:



The screenshot shows a SQL query execution interface. At the top, a query is entered: `select @(dateToString(stringToDate('\2019.01.31 22:02:45\'), 'yyyy.MM.dd HH:mm:ss'), 'yyyy/MM/dd HH:mm:ss'}) datum from persons`. Below the query, a blue bar indicates the result. A red arrow points from the `dateToString` function in the query to the result table. The result table has a header 'datum' and one row with the value '2019/01/31 22:02:45'. Below the table, there is a 'stáhnout' button and a 'datum' label. The execution statistics show 'SQL provedeno za: 00:00:00:000 Počet záznamů: 6' and a link 'Zobrazit SELDAT data'.

	datum
1	2019/01/31 22:02:45

a stejný příklad pro **MS SQL**:



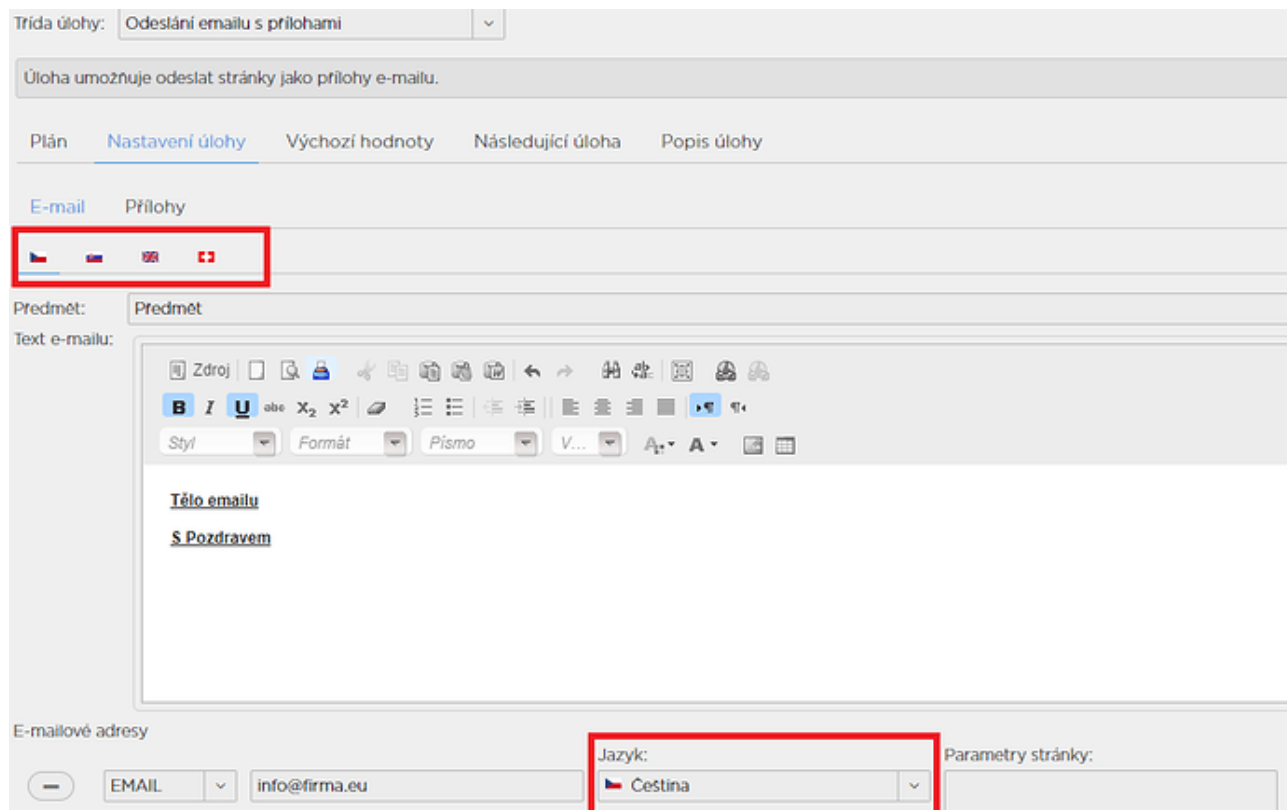
The screenshot shows a SQL query execution interface. The top part shows a failed query: `select @dateToString(stringToDate('\2019.01.31 22:02:45\'', 'yyyy.MM.dd HH:mm:ss'), 'yyyy/MM/dd HH:mm:ss')} datum from persons`. Below it, a message states: "Zobrazí vyhodnocený SQL výraz a změní čas, který SQL server potřeboval k jeho vykonání a dále čas potřebný k načtení prvních 10ti záznamů, které také zobrazí." A red arrow points from this message to the successful query below: `select FORMAT(CAST('2019.01.31 22:02:45' AS DATETIME), 'yyyy/MM/dd HH:mm:ss') datum from persons`. Below the successful query, the execution statistics are shown: "SQL provedeno za: 00:00:00:009 Počet záznamů: 100". A "stáhnout" button is visible. Below the button, the column name "datum" is shown. A red box highlights the result of the query: "1 2019/01/31 22:02:45".

Zde stojí za povšimnutí, že MS SQL se pokusí použít formát „odhadnout“ a vytvořit z řetězcového vstupního parametru datumu DB objekt DATETIME. V tomto případě úspěšně, pak jej převede na tvar **yyyy/MM/dd HH:mm:ss**.

Také se opět musí přidat dvojí „zauvozování“ prvního parametru funkce `stringToDate()`, protože skript neví, zda se jedná o název sloupce, či řetězcovou konstantu určenou pro použití ve výsledném SQL dotazu.

## 2.8 ZMĚNY V PLÁNOVANÉ ÚLOZE TYPU ODESLÁNÍ E-MAILU S PŘÍLOHOU

1. Předmět e-mailu i jeho tělo je nově možné definovat pro různé jazyky. Jazyky se přepínají pomocí záložek v levé horní části editoru obsahu e-mailu. Každému příjemci pak bude zaslán e-mail v jazyce, který má nastavený u své e-mailové adresy.



2. V předmětu i v těle e-mailu je možné využívat vyhodnocování výrazů (Template).
3. Nově je možné definovat místo e-mailové adresy SQL výraz, který načte e-mailové adresy z databáze. Pro využití SQL **je nutné vybrat konektor**, který aplikace definuje a na jehož napojení na databázi bude SQL provedeno. SQL výraz může načíst i více e-mailových adres a pak bude chování obdobné jako při výběru ROLE.

Jedná se v podstatě o skupinu emailových adres. Rozdíl je v tom, že SQL výraz může kromě **sloupce s názvem email, který je povinný**, obsahovat i sloupec s názvem **lang**, který má vyšší prioritu než jazykové nastavení celé skupiny. Pokud sloupec lang ve výrazu není, použije se jazyk definovaný pro celou skupinu. SQL výraz může vypadat např. takto:

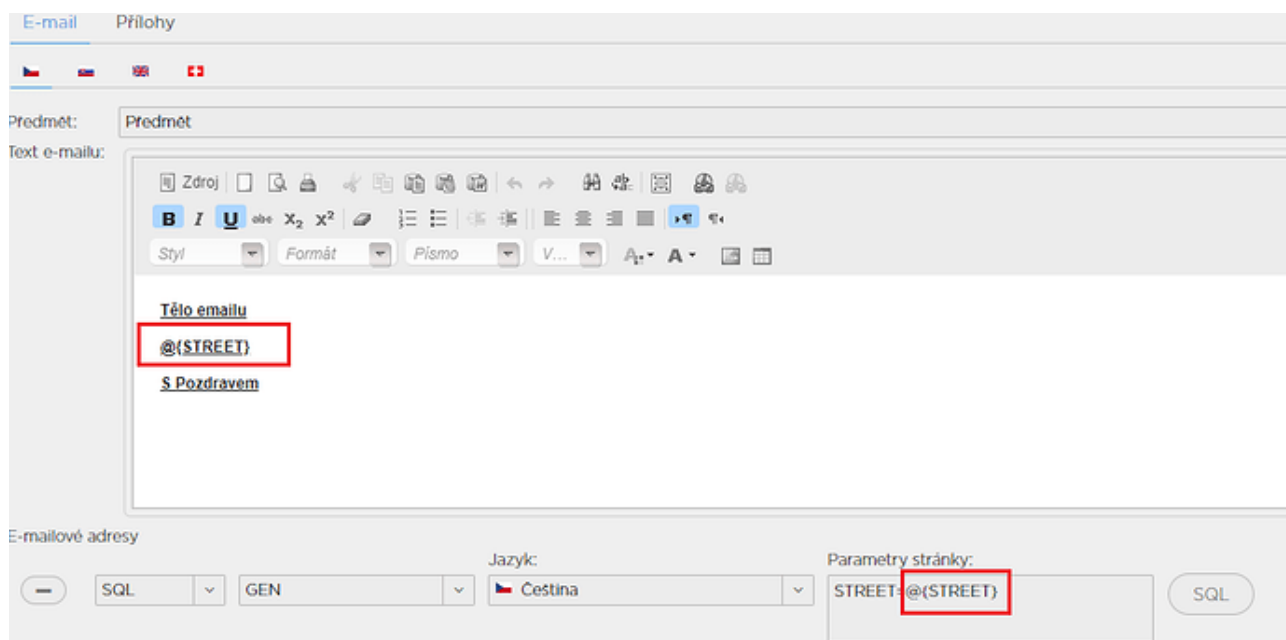
```
select email as email, 'cz_CZ' as lang from table
```

Načítání adres pomocí SQL přináší možnost využít hodnot načtených pomocí SQL k vyhodnocování výrazů v předmětu, těle emailu a i v parametrech, které se zašlou stránce při generování příloh.

Např. pokud SQL výraz vypadá takto:

```
select email, 'cs_CZ' as lang, STREET from Addresses
```

Ize při vyhodnocování výrazů použít hodnoty ve všech třech sloupcích. Např. sloupec **STREET** lze využít způsobem zachyceném na obrázku:



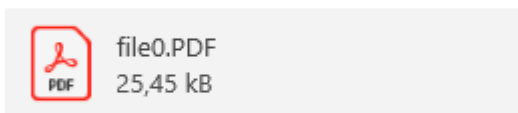
Výsledný e-mail pak vypadá následovně:

## Předmět



karel. [REDACTED]@ [REDACTED]  
15:20

Komu: Karel [REDACTED]



## Tělo emailu

**Pod Náspem 959**

**S Pozdravem**

Parametr **STREET** byl zároveň předán i do stránky a ta jej použila k zobrazení jako text. Vygenerovaná příloha pak vypadá následovně:



## 2.9 DXDATAGRID – NOVÁ GENERACE TABULKY

Preneste hlavičku sloupce zde pro sloučení ☐

▼	Priorita	▼ ↑	Datum 2	Předmět	▼	▼	Vlastní
	0		22.9.2006 0:00:00	dotaz na pozáruční ..			HIGH
	1		22.9.2006 0:00:00	dotaz na cenu u skla..			MID
	0		22.9.2006 0:00:00	dotaz na cenu skl. k..			HIGH
	3		22.9.2006 14:45:00	zájem o nákup jerábu			LOW
	3		22.9.2006 14:45:00	zájem o nákup jerábu			LOW
	3		22.9.2006 14:48:00	zaslány propagační ..			LOW
	3		22.9.2006 14:55:00	zaslány propagační ..			LOW
	3		23.9.2006 9:00:00	návrh na prezentaci ..			LOW
	3		27.9.2006 14:00:00	provedena prezenta..			LOW
	3		28.9.2006 10:00:00	zaslání smlouvy			LOW
	3		28.9.2006 10:20:00	akceptování smlouvy			LOW
▼			28.9.2006 10:30:00	akceptování smlouvy			LOW

DataGrid je nová komponenta pro zobrazování dat formou tabulky s některými pokročilými možnostmi, jako jsou např.:

- Skupinování řádků přes vybrané sloupce

Priorita ↑ ▼	Datum 2 ↑ ▼	Předmět
▼	▼	dotaz na pozáruční servis
	▼ Datum 2: 22.9.2006 0:00:00	dotaz na cenu skl. karty MB100
	▶ Datum 2: 22.9.2009 8:00:00	
	▶ Datum 2: 9.9.2009 8:00:00	
▶	Priorita: 1	
▶	Priorita: 3	

- Fixování sloupců vlevo při scrollování

Datum 2	Datum 1
22.9.2006 14:55:00	22.9.2006 Pátek, 22
22.9.2006 14:55:00	22.9.2006 Pátek, 22
22.9.2006 14:55:00	22.9.2006 Pátek, 22
22.9.2006 14:55:00	22.9.2006 Pátek, 22
22.9.2006 14:55:00	22.9.2006 Pátek, 22
22.9.2006 14:55:00	22.9.2006 Pátek, 22
22.9.2006 14:55:00	22.9.2006 Pátek, 22

Srovnat vzestupně  
 Srovnat sestupně  


---

 Zrušit rovnání  


---

 Sloučit sloupce  


---

 Oddělit vše  


---

 Uchytit ▶

Vlevo  
 Vpravo

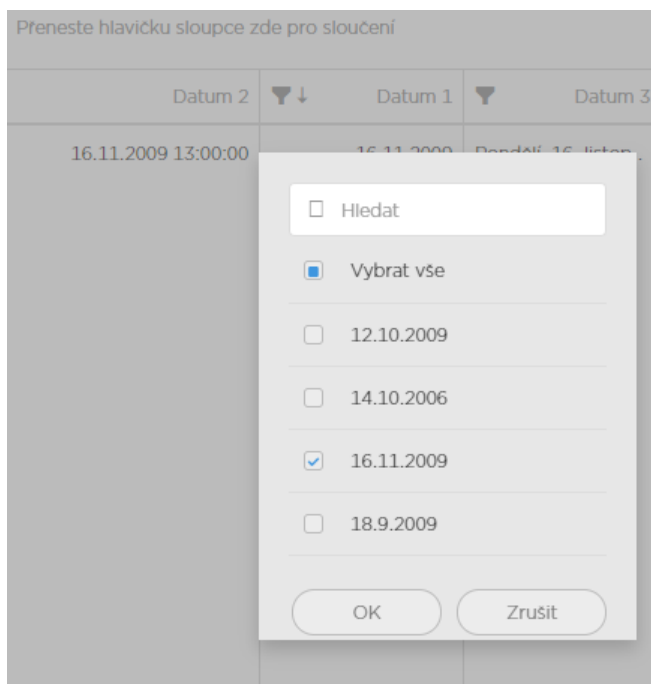
 Uvolnit

- Řazení

Prenešte niavicku sloupce dole pro sloučení

Datum 2	Datum 1
21.9.2009 8:00:00	21.9.2009
18.9.2009 8:00:00	18.9.2009
9.9.2009 8:00:00	9.9.2009
9.9.2009 8:00:00	9.9.2009
9.9.2009 8:00:00	9.9.2009
8.9.2009 13:00:00	8.9.2009
14.10.2006 13:10:00	14.10.2006
14.10.2006 13:00:00	14.10.2006
28.9.2006 10:20:00	28.9.2006
28.9.2006 10:20:00	28.9.2006
28.9.2006 10:00:00	28.9.2006

- Filtrování podle hodnot ve sloupcích



Jednou z nejdůležitějších vlastností gridu je VisibleFields. Popisuje (definuje), jaké sloupce budou v gridu použity a jaké budou mít vlastnosti.

Datový zdroj InputData poskytne seznam všech dostupných fieldů a z nich se následně sestaví seznam sloupců, které budou v gridu obsaženy.

**Pokud tuto vlastnost necháme prázdnou, použijí se všechny fieldy, které definuje datový zdroj a jejich vlastnosti se převezmou z těchto definic. Jakmile je ale definice sloupců zadána, pak se použijí pouze sloupce z této definice.**

Grid umožňuje definovat vlastnosti pro jednotlivé sloupce a dokonce i definovat sloupce, které nejsou obsaženy přímo v datovém zdroji, ale jsou výsledkem výpočtu.



## 2.9.1 RŮZNÉ MOŽNOSTI ZOBRAZENÍ SLOUPCŮ

### 2.9.1.1 PROSTÉ PŘEVZETÍ SLOUPCE Z DEFINICE FIELDŮ V DATOVÉM ZDROJI

Field z datového zdroje, s názvem ID, bude zobrazen v gridu a jeho vlastnosti budou převzaty z definice v datovém zdroji:

```
{"valueFieldName":"ID"}
```

### 2.9.1.2 PŘEVZETÍ SLOUPCE Z DEFINICE FIELDŮ V DATOVÉM ZDROJI A ÚPRAVA VLASTNOSTÍ

Field z datového zdroje, s názvem Complete (stav dokončení), bude zobrazen v gridu jako progress bar (grafické vyjádření stavu dokončení):

```
{"valueFieldName":"complete", "displayType":"Percents"}
```

### 2.9.1.3 MOŽNOSTI DODATEČNÉ KONFIGURACE

- String **columnName** - Název sloupce v gridu. Pokud název odpovídá názvu fieldu z datového zdroje, pak lze ostatní vlastnosti vynechat a nastavení sloupce se určí podle fieldu v datovém zdroji.
- String **valueFieldName** - Název fieldu v datovém zdroji, který obsahuje hodnotu pro zobrazení v tomto sloupci.
- String **value2FieldName** - Některé typy zobrazení potřebují definovat dva datové sloupce (např. DIFF, DURATION atd...). Zde se definuje jméno druhého fieldu.
- String **valueFormat** - Formát pro zobrazení hodnoty z hlavního fieldu.
- String **value2Format** - Formát pro zobrazení hodnoty z druhého fieldu.
- String **displayType** - Speciální typ zobrazení - jednotlivé typy budou popsány níže.
- String **totalSummary** - Zobrazit celkovou agregaci sloupce ve spodní části gridu (count, min, max, avg, sum).
- String **groupSummary** - Zobrazit agregaci sloupce pro skupinu - ve spodní části skupiny (count, min, max, avg, sum).
- String **width** - Šířka sloupce - např. „150px“, „80%“ nebo „auto“.
- String **height** - Výška je použita pouze pro zobrazení obrázků, např. „150px“.
- String **pictureType** - V případě, že **displayType** je **picture** a nejedná se o obrázek z ABRA Gen, pak je nutné specifikovat MIME type obrázku, např. **image/jpeg**.
- Integer **hidingPriority** - Určuje pořadí automatického skrývání sloupců při nedostatku místa. Od 0 výše, přičemž 0 se skrývá jako první.
- Boolean **visible** = true. Určuje, zda bude sloupec viditelný v gridu, nebo bude ve skrytých odkud je možné je zobrazit.
- Boolean **allowSorting** = true - Určuje, zda bude možné podle sloupce řadit. (Řazení provádí grid sám, není nutné ho implementovat v datovém zdroji.)
- String **sortOrder** - Řazení sloupce asc/desc.
- Boolean **allowFiltering** = true - Určuje, zda bude možné hodnoty ve sloupci filtrovat. (Filtrování provádí grid sám, není nutné ho implementovat v datovém zdroji.)
- Boolean **allowSearch** = true - Určuje, zda je povoleno vyhledávání ve sloupci.

- Boolean **allowResizing** = true - Určuje, zda je povoleno měnit šířku sloupce.
- Boolean **allowReordering** = true - Určuje, zda je možné sloupec přesouvat.
- Boolean **allowHiding** = true - Určuje, zda je možné sloupec skrýt.
- Boolean **allowHeaderFiltering** = true
- Boolean **allowGrouping** = true - Určuje, zda je možné sloupec použít pro skupinování.
- Integer **groupIndex** - Určuje pořadí pro seskupování. 0-seskupení na první úrovni, 1-seskupení na druhé úrovni...
- String **groupExpression** - Javascriptový výraz, který vrací hodnotu pro skupinování. Lze použít např. pro sloupce typu CUSTOM, které zobrazují obrázek. Tato hodnota se pak použije pro vytváření skupin. Např. může vracet popis obrázku. Pokud není zadáno, je pro skupinování použitá hodnota sloupce.  
Pro vyhodnocení výrazu jsou k dispozici předdefinované proměnné **data** a **rowData**.  
Objekt **data** odkazuje na objekt obsahující informace o aktuální buňce, je popsán níže.  
Objekt **rowData** odkazuje na celý aktuální řádek, rovněž je popsán níže.  
Např. `data.value == 0 ? 'Nárůst' : 'Pokles'`  
vrátí text Nárůst / Pokles, který bude použitý pro skupinování.
- String **sortExpression** - Velmi podobné `groupExpression`, s tím rozdílem, že hodnota z tohoto výrazu je použita pro třídění. Pokud není zadáno, pak je pro třídění přednostně použita formátovaná hodnota (pokud je použit `valueFieldFormat`), jinak je při třídění použita prostá hodnota sloupce. U datumových sloupců je správné řazení řešeno automaticky i když je zobrazena formátovaná hodnota.  
Pro vyhodnocení výrazu jsou k dispozici předdefinované proměnné **data** a **rowData**.  
Objekt **data** odkazuje na objekt obsahující informace o aktuální buňce, je popsán níže.  
Objekt **rowData** odkazuje na celý aktuální řádek, rovněž je popsán níže.  
Např. `data.value == 0 ? 1 : 0`  
Pokud je prostá hodnota sloupce `0 = Nárůst`, bude pro třídění použita hodnota `1`. Pokud se jedná o pokles, použije se pro třídění hodnota `0`. Tím se poklesy zatřídí před nárůsty i přes to, že prostá hodnota fieldu je definována obráceně.
- String **expression** - Výraz pro vytvoření obsahu buňky u typu CUSTOM. Viz samostatný odstavec věnovaný typu CUSTOM.
- String **cssClass** - Umožní zadat CSS třídu, která bude umístěna na všechny HTML elementy typu které představují jednotlivé buňky v řádcích tohoto sloupce.

#### 2.9.1.4 OBJECT DATA

Jedná se o Javascriptový objekt nesoucí informace o aktuální buňce. jeho struktura je následující:

- **value** - prostá neformátovaná hodnota načtená z datového zdroje.
- **formattedValue** (nemusí být definována) - textová hodnota naformátovaná pomocí `valueFieldFormat`.

#### 2.9.1.5 OBJECT ROWDATA

Jedná se o Javascriptový objekt obsahující data celého řádku. Pro každý sloupec obsahuje objekt typu `data`. Např. `rowData.Aktuální` vrátí objekt `data` pro sloupec Aktuální.

## 2.9.2 SPECIÁLNÍ TYPY ZOBRAZENÍ SLOUPCŮ

### 2.9.2.1 TYPY DIFF A DIFFWITHPERCENTS

Sloupec typu DIFF zobrazuje aktuální hodnotu a na druhém řádku přírůstek/úbytek včetně grafického znázornění (šipka nahoru/dolů + barevné zvýraznění). Jak je vidět v definici sloupce níže, **zadávají se dvě hodnoty. Aktuální hodnota (valueFieldName) a předchozí hodnota (value2FieldName)**. Z těchto hodnot je pak vypočten rozdíl a ten je zobrazen, včetně grafického znázornění a barevného rozlišení. U typu **DIFFWITHPERCENTS** je navíc menším písmem vpravo dole vyčíslen procentuální rozdíl.

```
{
  "columnName": "rozdil",
  "displayType": "diff",
  "valueFieldName": "amount",
  "value2FieldName": "amount2",
  "valueFormat": "@{ '%', .2f
'+getCurrencySymbol()+'' }"}
}
```

	DIFF
	0,00
	19,00 AA ↑
	0,00
	0,00 AA ↓

DIFF s procentem

2 462 200,00
2 462 200,00 ↑ 100,00 %

### 2.9.2.2 TYP PICTURE

Sloupec typu PICTURE zobrazuje obrázek. Existuje několik možností, jak obrázek zobrazit:

1. Použít dataset pro BusinessObjekty `[[DBObjectRecordSet]]`. BusinessObject musí obsahovat definici fieldu, který má ve vlastnostech příznak, že se jedná o Gx obrázek. Pak stačí použít prostou definici sloupce. Typ obrázku je rozpoznán automaticky. Je možné dodatečně specifikovat velikost pro zobrazení v gridu:
 

```
{
  "valueFieldName": "picturedata",
  "width": "125px",
  "height": "150px"
}
```
2. Obrázek načtený pomocí SQL za použití `[[TableDataRecordSet]]`.
3. `Select picturedata from table`




Pokud je select prováděn z tabulky ABRA Gen, pak je situace stejná jako v bodu 1.

Pokud select načítá „čistá“ data obrázku, pak je nutné specifikovat typ obrázku **pictureType**:

```
{
  "valueFieldName": "picturedata",
  "width": "125px",
  "height": "150px",
  "pictureType": "image/jpeg"
}
```

4. Další možností je použít jako datový zdroj například CSVTable data. Data obrázku musí být v CSV uložena ve formátu BASE64. Definice tohoto fieldu v datovém zdroji pak **není typu BINARYBLOB, ale musí být typu STRING** a situace je stejná jako v bodě 2. Je nutné specifikovat **pictureType**.

Přenešte hlavičku sloupce zde pro sloučení □

Titulek ▼	Obrázek ▼
TV Panasonic TX-14B4TP	
Videorecordér PANASONIC NV-HD680	
Palm Tungsten T3	

### 2.9.2.3 TYPY PERCENTS A PERCENTSWITHTEXT

Tento typ sloupce zobrazí progress bar, který reprezentuje procentuální hodnotu. Podkladová data musejí být číselné hodnoty v rozsahu 0–100. PERCENTSWITHTEXT navíc uvnitř progress baru zobrazí text (procentní údaj).

▼ Procento	▼ Procento s textem
	

#### 2.9.2.4 TYPY DATE, TIME, DATETIME, DATEFULL A DATETIMEFULL

Zdrojová data musejí být z datového zdroje čtena jako DATUM s prefixem \$DATE. Zobrazení je pak formátováno podle národního prostředí přihlášeného uživatele.

Formáty pro češtinu jsou následující:

- DATE 22.9.2006
- TIME 14:45:00
- DATETIME 22.9.2006 14:45:00
- DATEFULL Pátek, 22. září 2006
- DATETIMEFULL Pátek, 22. září 2006 14:45:00

27.9.2006	27.9.2006 14:00:...	Streda, 27. září 2006	Streda, 27. září 2006 14:00:00	14:00:00	Streda, 27. září 2006 16:00:00
-----------	---------------------	-----------------------	--------------------------------	----------	--------------------------------

#### 2.9.2.5 TYPY DATEINTERVAL, DURATION A DURATIONFULL

Tyto typy zobrazení sloupců umožňují zobrazovat formátovaný rozdíl mezi dvěma hodnotami typu DATE.

Je nutné zadat valueFieldName (začátek intervalu) i value2FieldName (konec). Pokud je zadána pouze hodnota valueFieldName, pak je zobrazen interval od datumu 0 (EMPTYDATE - 30.12.1899).

Tuto variantu lze použít např. pokud mám v databázi uložené pouze číslo reprezentující informaci, jak dlouho jsem na něčem pracoval (třeba kumulovaný součet).

Formáty pro češtinu jsou následující:

- DATEINTERVAL 09.09.2009 - 24.09.2009
- DURATION 15 dní
- DURATIONFULL 15d 08:00:00,000

2 hodin	2:00:00,000
---------	-------------

#### 2.9.2.6 TYPY ISO8601DATE A ISO8601DATEFULL

Umožní zobrazit text ve formátu ISO8601 [https://cs.wikipedia.org/wiki/ISO\\_8601](https://cs.wikipedia.org/wiki/ISO_8601) (např. 2020-05-11T07:29Z) jako datum.

### 2.9.2.7 TYP CUSTOM

```
{"columnName":"Vlastní", "displayType":"custom",  
"valueFieldName":"PRIORITY", "expression":"$('<div>'+(data.value == 0 ?  
'HIGH' : data.value == 1 ? 'MID' : 'LOW') +'</div>')"} }
```

U typu CUSTOM je nutné v **expression** specifikovat javascriptový výraz, který vrátí HTML obsah buňky.

Jak je vidět, ve výrazu lze použít:

- proměnou **data**, která umožňuje přístup k hodnotě buňky `data.value` nebo `data.formatedValue` (pokud buňku formátuji)
- nebo vlastnost **rowData**, která zpřístupňuje data řádku formou `rowData[jmeno sloupce].value` nebo `rowData[jmeno sloupce].formatedValue`
- nebo vlastnost **options**, která zpřístupňuje další možnosti (**options.column** - vlastnosti aktuálního sloupce)

Např. **options.data.Priorita** obsahuje data pro sloupec Priorita.

Data sloupce jsou vždy objekt {value:hodnota, formatedValue:hodnota přeformátovaná pomocí valueFieldFormat}

Naše ukázka čte hodnotu našeho sloupce Priority a pokud je

- 0, vrátí HIGH
- 1, vrátí MID
- jinak vrátí LOW