

# RAD Studio a StellarDS.io

Rychlý vývoj mobilní aplikace s využitím RAD Studia a cloudových služeb společnosti TMS Software.

## Část 1 – Server



## Obsah

Úvod .....	3
Embarcadero RAD Studio .....	3
Co je StellarDS.io .....	3
Popis projektu .....	3
Serverová část aplikace .....	3
Zřízení účtu StellarDS.io .....	3
Vytvoření projektu .....	4
Definování struktury dat (datové schéma) .....	4
Správa uživatelů .....	6
Nastavení oprávnění .....	7
Import dat .....	7
Přihlašování ve webových aplikacích .....	10
Logování operací v rámci TMS StellarDS.io .....	10
Testování s Embarcadero REST Debugger .....	11
Načtení dat z tabulky Dodavatel .....	12
Načtení dat z tabulky Produkt .....	15
Vkládání dat .....	15
Aktualizace dat .....	16
Mazání dat .....	16
Save / Load Request .....	17
Volba „Copy Components“ .....	17
Závěr .....	17

# Úvod

Dokument představuje možnosti vývoje mobilních aplikací s využitím cloudové platformy StellarDS.io vybudované a provozované společností TMS Software a vývojového prostředí RAD Studio společnosti Embarcadero.

Cílem bylo popsat základní principy. Jedná se proto o maximálně zjednodušený příklad, který ani zdaleka nepostihuje všechny možnosti obou technologií. ani se nesnaží prezentovat jediné správné řešení. Vývoj aplikací je (alespoň zatím) tvůrčí záležitostí a cest k cíli je vždy více 😊.

## Embarcadero RAD Studio

Embarcadero RAD Studio je vývojové prostředí, jehož cílem je co nejvíce usnadnit a urychlit vývoj aplikačního software. Odtud zkratka RAD pro Rapid Application Development. Kromě rychlosti vývoje patří mezi další přednosti nástroje překlad zdrojového kódu pro více platform (Windows, MacOS, Android, iOS nebo Linux) a otevřenost. Kromě vestavěných aplikačních rámců VCL (Visual Component Library) a FMX (FireMonkey) lze využívat i obrovské množství komponent od nezávislých dodavatelů.

## Co je StellarDS.io

StellarDS.io je soubor cloudových služeb poskytovaná společností TMS Software, které nahrazují databázový, aplikační i webový server. Zcela bez programování můžete svá data sdílet prostřednictvím REST API doslova během minut. Nemusíte řešit jednotlivé REST pointy a převod dat do a z formátu JSON. Neřešíte doménu, správu serveru, zálohování databáze ani škálování nebo dostupnost. Samostatně je třeba vypíchnout řešení bezpečnosti. Za zabezpečení serverové části ručí TMS Software. Součástí služby je také správa uživatelů a oprávnění, včetně autentizace.

## Popis projektu

Popisuje realizaci jednoduchého systému, který splňuje následující požadavky:

- Centrální úložiště dat pro snadné sdílení dat
- API rozhraní pro tvorbu standardních, mobilních i webových klientů
- Možnost vzdálené zprávy
- Zabezpečený přístup (autentizace, řízení přístupu a uživatelských oprávnění)
- Klientskou aplikaci pro přístup z mobilních zařízení
- Minimální požadavky na správu

## Serverová část aplikace

Použití StellarDS.io umožňuje „přeskočit“ řadu kroků. Není třeba řešit server (ať již fyzický stroj, nebo hosting VPS) pro provoz řešení. Dále odpadá výběr, instalace a konfigurace databázového, aplikačního a webového serveru.

Předpokládejme, že máme hotovou analýzu požadavků a návrh datového modelu. V takovém případě se lze pustit přímo do realizace řešení.

## Zřízení účtu StellarDS.io

K platformě StellarDS.io se přihlásíte na stránce <https://stellards.io/account/sign-up>. V době přípravy tohoto dokumentu bylo možné využít kromě nabízených plánů i účast na beta testování. I po jeho ukončení bude ale k dispozici bezplatný přístup, díky kterému je možné si nejen službu vyzkoušet, ale i reálně využít pro menší nebo PoC (Proof of Concept) projekty. Ceny a rozsah služeb naleznete zde: <https://stellards.io/#pricing>.

stellards.io Features Pricing Libraries Documentation Login

### Register for Stellar DataStore

Thanks for your interest in Stellar DataStore.  
Don't forget to [read our terms and privacy policy](#).

**Username**  
username

**Email**  
email

**Password**  
password

**Confirm password**  
password

I agree to the [service level agreement](#)

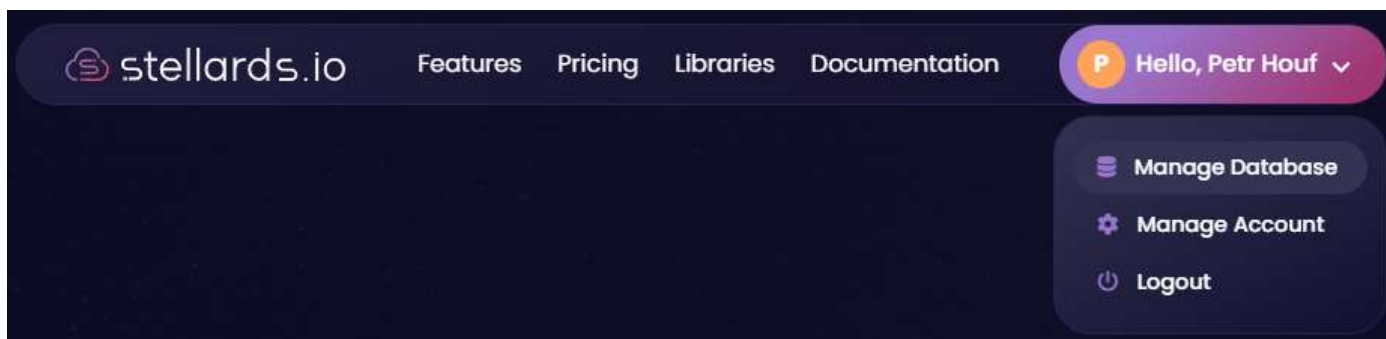
Nejsem robot reCAPTCHA  
Ochrana soukromí - Smluvní podmínky

Register

Obrázek 1 - Registrace do StellarDS.io

## Vytvoření projektu

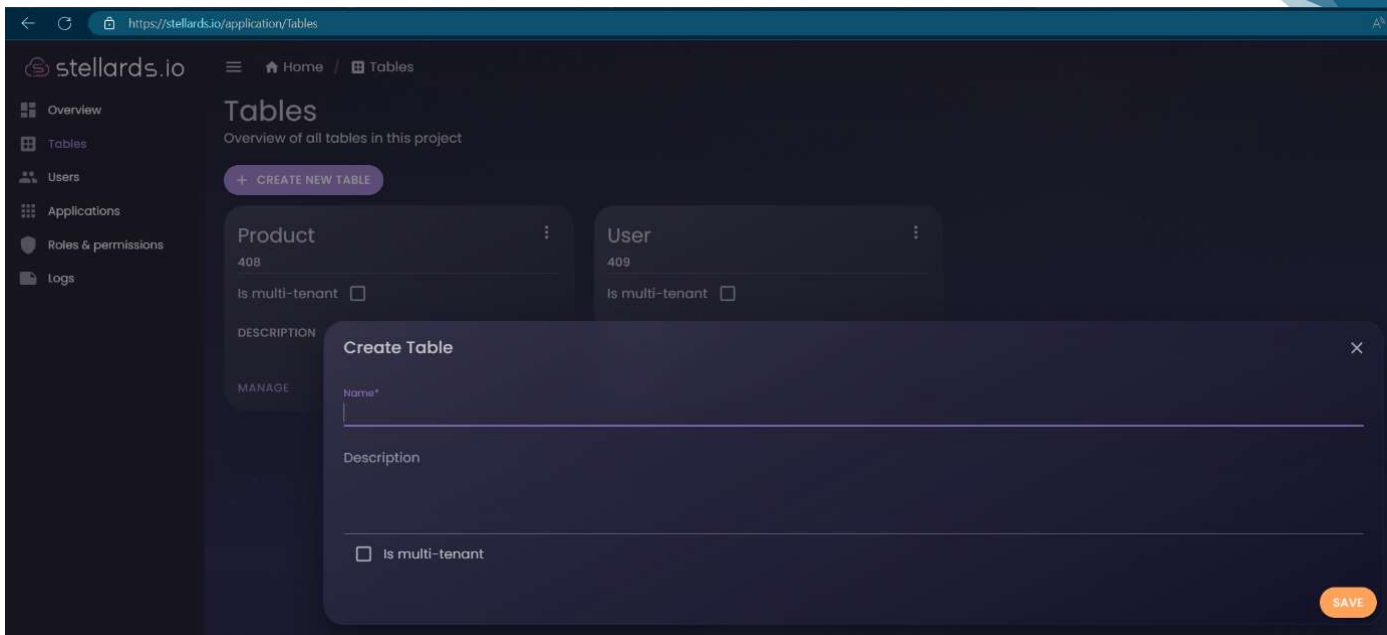
V dalším kroku již přistoupíme k definování datového úložiště. Po registraci získáme přístup do rozhraní správce (dále jen konzole správce). Zde po přihlášení přejdeme do sekce pro správu databáze („Manage Database“):



Nejdříve je třeba založit projekt. To v praxi znamená zvolit název pro datové úložiště, které hodláme definovat. Můžeme také určit, zda bude možné využívat „Invitations“ (StellarDS.io poskytuje aparát pro přizvání dalších uživatelů) a zda má být projekt „Multi-tenant“. To je výhodné v situaci, kdy má projekt sloužit více zákazníkům, ale zároveň potřebujete zajistit, že každý z nich bude mít přístup výhradně ke svým datům. Vlastnost multi-tenant můžeme nastavit pro celý projekt, nebo ji můžeme povolit jen pro vybrané tabulky. Nakonec platforma StellarDS.io vygeneruje pro projekt unikátní identifikátor **ProjectID**.

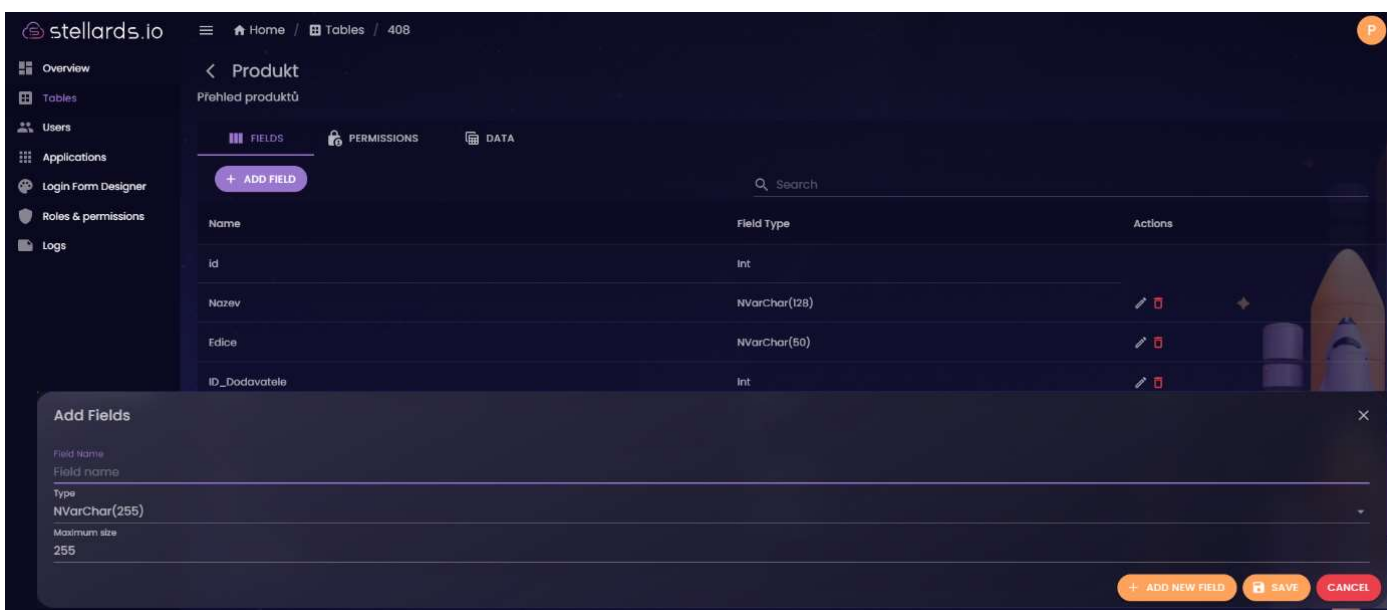
## Definování struktury dat (datové schéma)

Vlastní návrh struktury dat je naprosto intuitivní, protože se nijak neliší od postupu v běžně používaných databázových nástrojích. Nadefinujeme jednotlivé tabulky (Tables) a jejich pole (Fields):



Obrázek 2 - Definování tabulek

Jediné, na co je třeba při vytváření tabulky dát pozor, je identifikátor. Prostředí jej vytváří automaticky s defaultním názvem „ID“ a celočíselným datovým typem Integer. Pole je „autoincrement“, což znamená, že při vkládání dat hodnotu ID nezadávejte a systém jej přiřadí automaticky. Toto pole (viz obrázek níže) není možné odstranit.



Obrázek 3 - Definování sloupců tabulky

V sekci „Tables“ si můžete (pokud k tomu máte oprávnění) také zobrazit data, která tabulka obsahuje:

id	Název	Edice	Dodavatel
21	TMS VCL GUIActions	Single Developer	27
22	TMS VCL MultiTouch SDK	Single Developer	27

Obrázek 4 - Zobrazení dat vč. vyhledávání

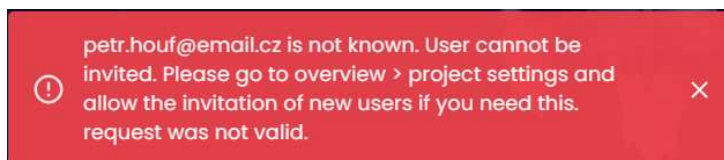
## Správa uživatelů

Správa uživatelů je důležitou součástí prakticky každého většího systému. StellarDS.io umožňuje nejen standardní přidávání a rušení uživatelských účtů (sekce „Users“), ale nabízí navíc kompletní aparát pro registraci uživatelů prostřednictvím „pozvánky“.

Username	Email	Role	Is Owner	Has application access	Actions
Petr Houf	petr.houf@codesecondure.eu	Admin	True	True	
	petr.houf@email.cz	Admin	False	True	

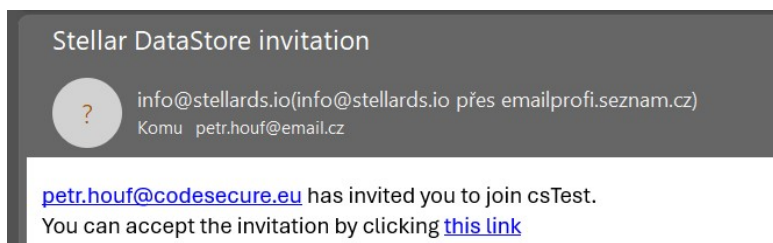
Obrázek 5 - Správa uživatelů

Posílání pozvánek musí být povoleno v nastavení projektu, jinak na to budete upozorněni chybovým hlášením:



Obrázek 6 - Chyba při odeslání pozvánky

StellarDS.io zajistí odeslání pozvánky na požadovanou emailovou adresu. Zpráva obsahuje mimo jiné také link pro ověření adresy.



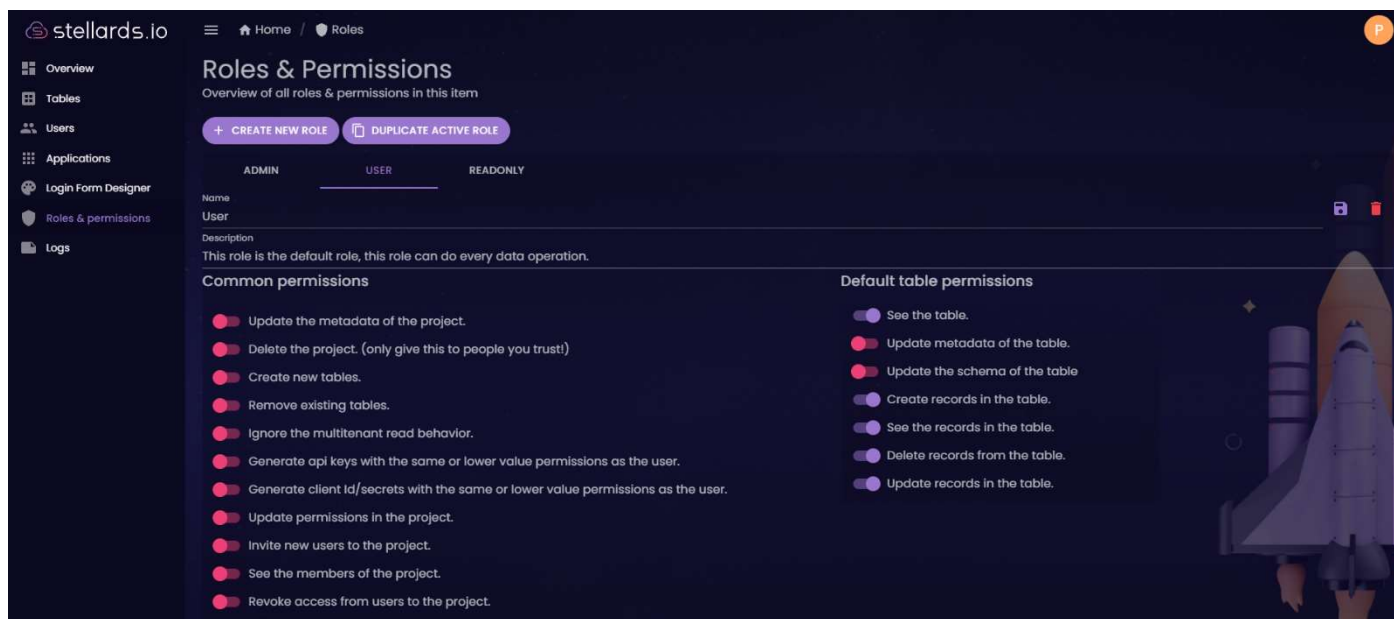
Obrázek 7 - Zobrazení "Invitation" uživateli

Do doby, než je pozvánka přijata a uživatel v systému založen, je uživatel uveden v seznamu „čekající pozvánky“ (Pending Invitations).

## Nastavení oprávnění

Společnost TMS Software připravila tři přednastavené role. Jedná se o „ADMIN“, „USER“ a „READONLY“. Pro každou definovanou tabulku lze nastavit oprávnění, která jsou jednotlivým rolím přiřazena. Tedy jestli mohou uživatelé s danou rolí data číst, upravovat nebo dokonce měnit jejich strukturu.

V záložce „Roles & permissions“ můžete vytvářet nové role, nastavit jednotlivá oprávnění a dokonce oprávnění, která budou uživateli dané role automaticky přiřazeny při vytvoření nové tabulky:



Obrázek 8 - Nastavení oprávnění

Nyní tedy máme definovanou strukturu dat, vytvořené uživatelské účty s přiřazenými oprávněními.

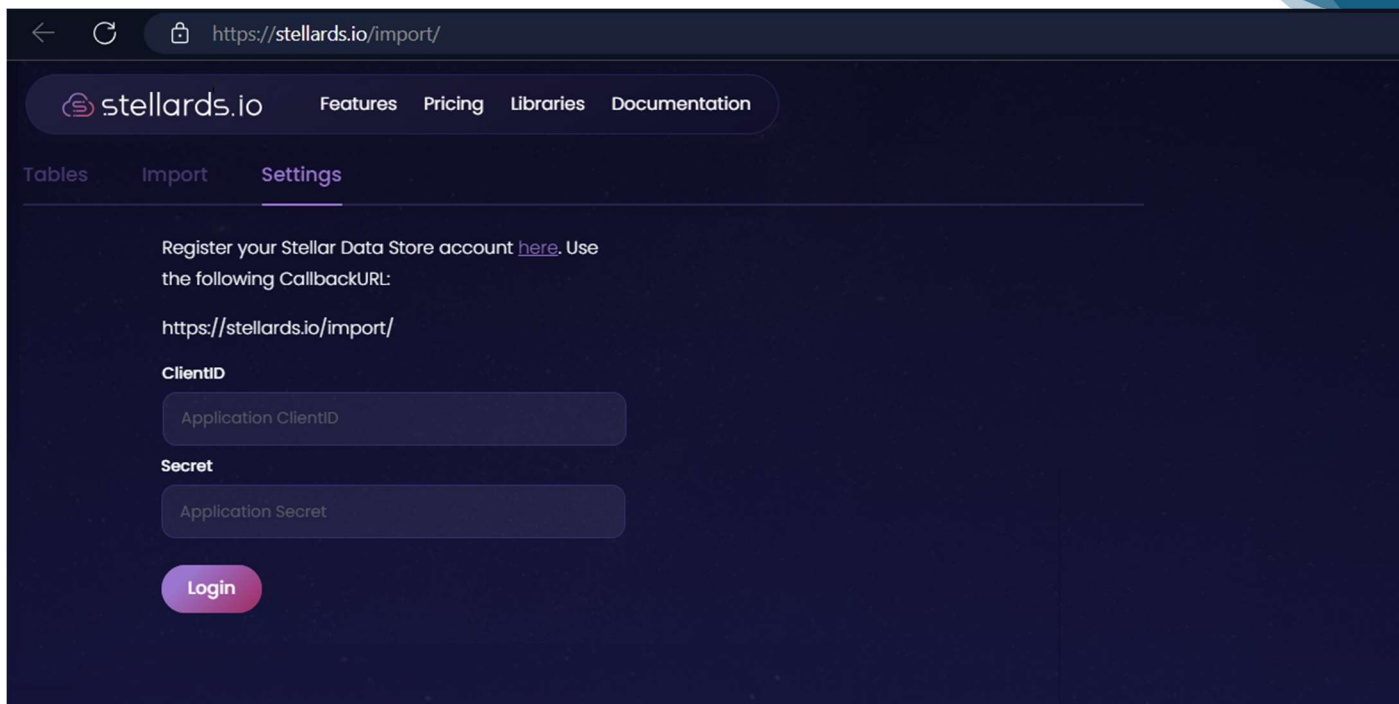
DODAVATEL		PRODUKT	
ID <PK>	Integer	ID <PK>	Integer
Nazev	NVarchar(50)	Dodavatel <FK>	Integer
Mena	NVarchar(3)	Nazev	NVarchar(50)
Aktualizace	DateTime	Edice	NVarchar(25)

Obrázek 9 - Definované tabulky

Pro vkládání a aktualizaci dat se primárně předpokládá použití API rozhraní. Přesto je často užitečné některé tabulky naplnit rovnou (například číselníky). I na to bylo myšleno.

## Import dat

Společnost TMS Software neopomněla ani možnost importu dat z již existujících datových zdrojů. Import se provádí přes webové rozhraní na adrese <https://stellards.io/import/>. Pro přihlášení je třeba se přihlásit prostřednictvím ClientID a hesla.

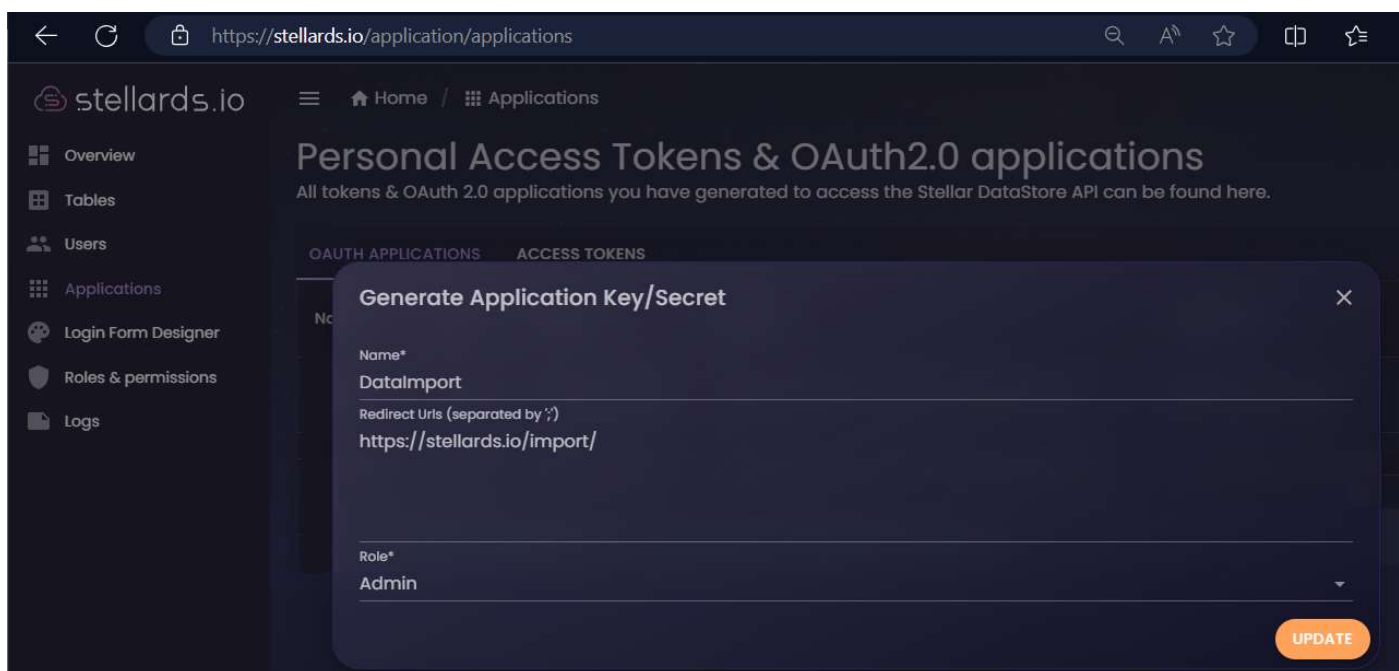


Obrázek 10 - Přihlášení do rozhraní pro import dat

Oba údaje získáte v konzoli správce. V nabídce umístěné v levé části obrazovky zvolíte „Applications“. Zde musí být uvedena každá aplikace, které chcete umožnit přístup k vašemu datovému úložišti. Zde tak musíme povolit i rozhraní pro import dat. Kliknete tak na tlačítko „Add Application“ a vyplníme požadované údaje. Jako jméno aplikace použijeme například „DataImport“, ale může být samozřejmě zcela libovolné.

Do pole „Redirect URL“ vložíme odkaz na stránky importu, tedy <https://stellards.io/import/>. Přístupové údaje mohou být použity pro více URL. V takovém případě je třeba jednotlivé odkazy oddělit středníkem.

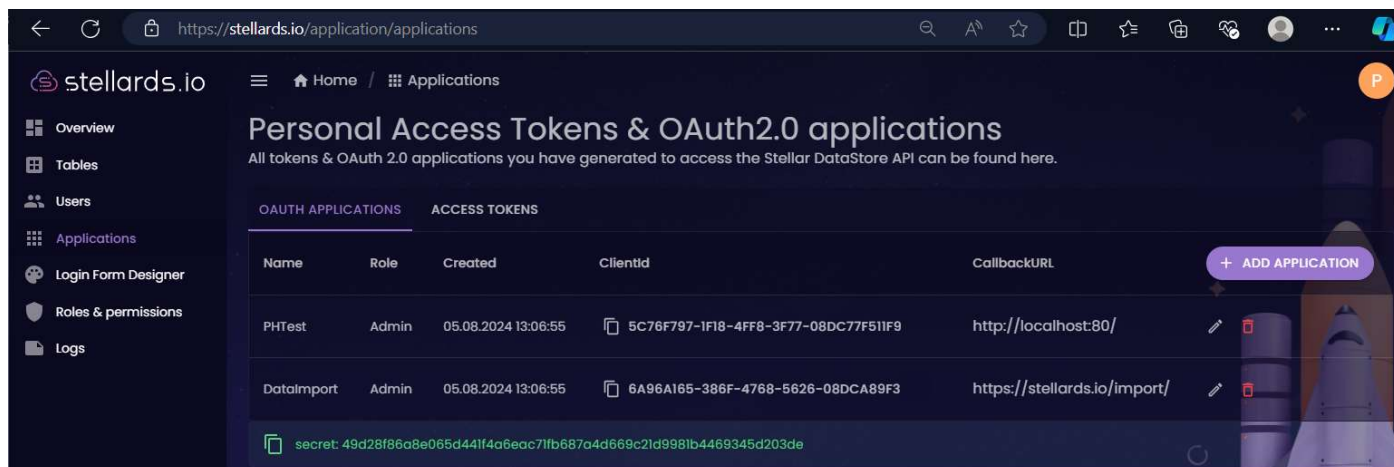
Nakonec určíme, která role bude mít oprávnění data importovat. Pokud nastavíme například roli „Admin“, bude tato funkce dostupná všem uživatelům, kterým jsme tuto roli přiřadili.



Obrázek 11 - Získání přístupových údajů pro import dat



Po odeslání formuláře tlačítkem „UPDATE“ se aplikace objeví v přehledu a je u ní uvedeno „ClientID“. Pod seznamem aplikací se také zobrazí heslo „secret“. Oba údaje si můžeme zkopírovat přímo z formuláře:

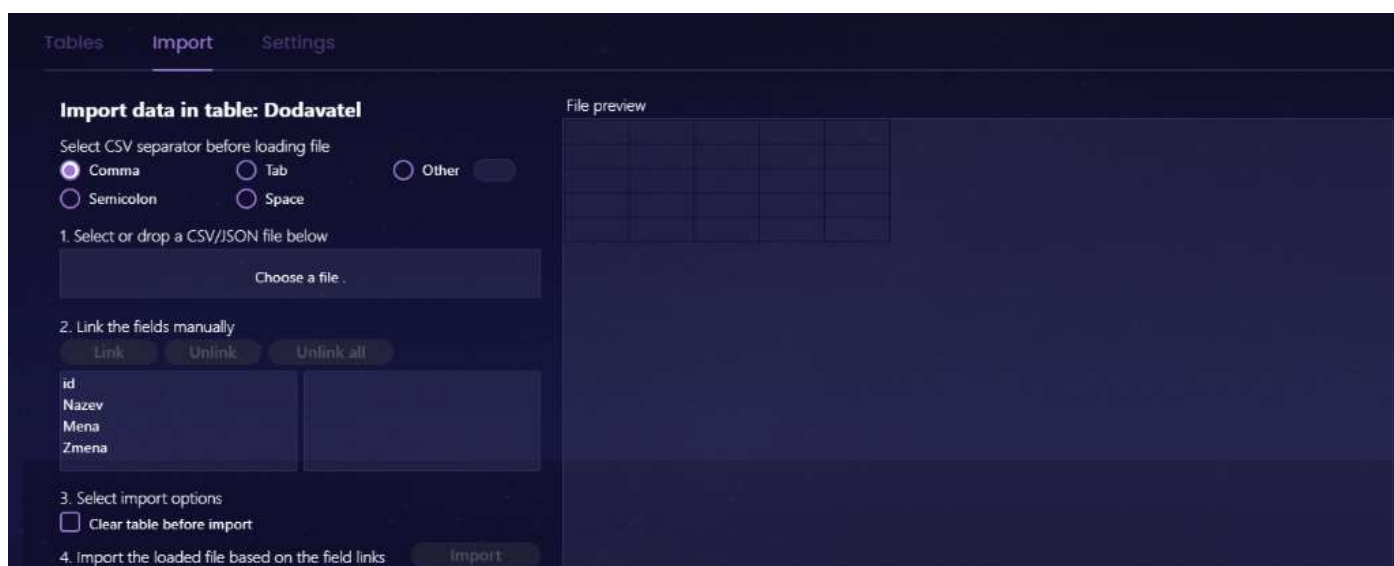


Obrázek 12 - ClientID & Secret

Data lze importovat ze souboru ve formátu \*.json, nebo \*.csv. V nastaveních můžete vybrat kódování, oddělovače a provést mapování dat mezi cílem a zdrojem.

Při vytváření souboru pro import doporučuji dodržet následující:

- Použijte kódování utf-8. Je to standard a vyhnete se tak problémům s diakritikou a speciálními znaky.
- Pole typu Datum se převedte do formátu „rrrr-mm-dd“. Pokud se jedná o DateTime, pak „rrrr-mm-dd:hh.mm.ss“. Jedná se o preferovaný formát pro JSON. Více viz: [https://en.wikipedia.org/wiki/ISO\\_8601](https://en.wikipedia.org/wiki/ISO_8601).
- Pozor na primární klíče! Platforma StellarDS.io je, jak jsem již zmiňoval generuje automaticky. Buď je tedy z importu vypustíte, nebo pro ně vytvořte samostatné pole. Týká se to především případů, když importujete data do tabulek, které jsou mezi sebou provázány prostřednictvím primárního a cizího klíče.



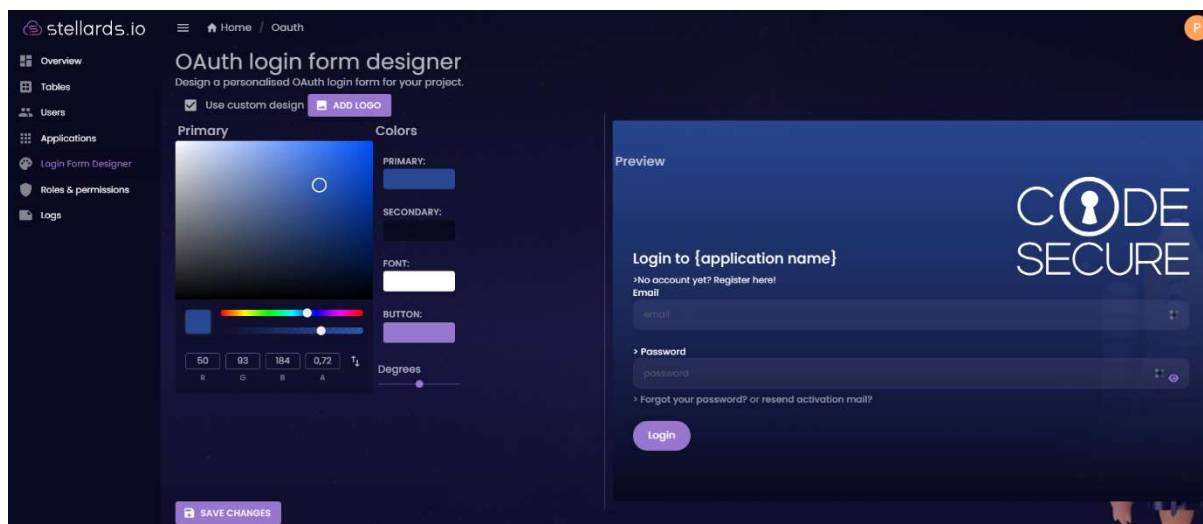
Obrázek 13 - Výběr tabulky pro import

Při vlastním importu pak určíte použitý oddělovač a vyberete příslušný soubor. V dalším kroku je třeba provést linkování mezi poli tabulky a poli v souboru. A to i přesto, že se počet sloupců i jejich jména shodují a data se správně zobrazila v náhledu „File preview“. Dokud tento krok nedokončíte, tlačítko „Import“ zůstává neaktivní.

Pod číslem 3 najdete volbu „Clear table before import“. Pokud není tato volba zaškrtnuta, budou data ze souboru přidána již k existujícím datům. V opačném případě budou existující data odstraněna a vložena nová. ID odstraněných záznamů již nebudou znovu použita.

## Přihlašování ve webových aplikacích

TMS Software vybavil StellarDS.io návrhářem, aby bylo možné personalizovat přihlašovací dialog:

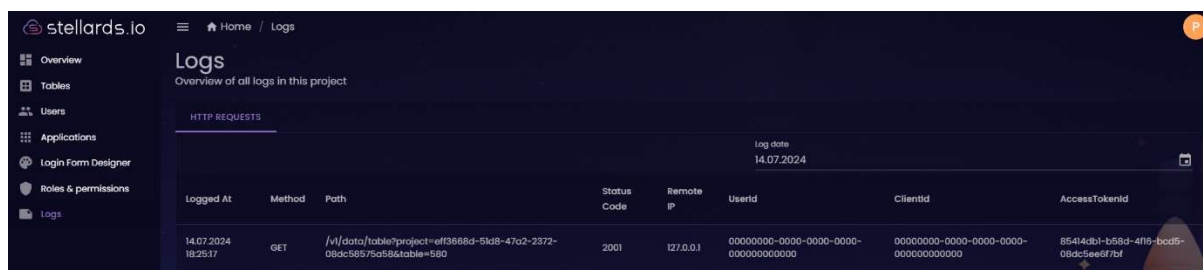


Obrázek 14 - Designer přihlašovacích dialogů

Můžete tak použít firemní grafiku vaší společnosti nebo zákazníka, pro kterého aplikaci vyvíjíte.

## Logování operací v rámci TMS StellarDS.io

Během testování ale i později při vlastním provozu aplikace jistě oceníte možnost získat informace o serverem zpracovávaných požadavcích. Ten najdete ve webové konzoli v sekci „logs“:



Obrázek 15 - Přehled operací (http požadavků)

Tím se serverová část připravuje. Data jsou nyní dostupná prostřednictvím REST API z libovolného operačního systému či aplikace.

Než se začneme zabývat vývojem klientské aplikace, podíváme se, jak formulovat požadavky pro StellarDS.io a jak vypadají odpovědi. Vývoj aplikace je v reálu výrazně jednodušší (TMS Software nabízí pro RAD Studio komponenty, které udělají prakticky vše za Vás), ale je dobré vědět jak věci fungují „pod kapotou“.

Použití REST Debuggeru je důležité nejen pro lepší pochopení komunikace mezi serverem a klientem, ale například také pro rychlé testování REST služeb obecně.

# Testování s Embarcadero REST Debugger

Podporu komunikaci prostřednictvím REST API dnes nabízí snad všechna prostředí a programovací jazyky. Často je ale výhodné si sestavit a otestovat REST požadavky izolovaně. Případné problémy se nám tak nebudou míchat s chybami ve vyvíjené aplikaci.

Použít můžeme nástroj příkazového řádku curl (<https://curl.se/download.html>), nebo některý z vizuálních rozhraní. Jedním z nich je Embarcadero REST Debugger, který je všem vývojářům k dispozici zcela zdarma. Je součástí vývojových nástrojů Embarcadero, nebo si jej můžete stáhnout na následující adrese <https://www.embarcadero.com/free-tools/rest-debugger>.

Dokumentace [http://docwiki.embarcadero.com/RADStudio/Athens//en/REST\\_Debugger\\_Tool](http://docwiki.embarcadero.com/RADStudio/Athens//en/REST_Debugger_Tool)

Uživatelské rozhraní REST Debuggeru má dvě oblasti. První „Request“ slouží pro sestavení požadavku, který chceme odeslat na zvolený server. Druhá oblast „Response“ pak umožňuje zobrazení získané odpovědi. Obě oblasti jsou pak dále rozděleny na záložky. Pro definování požadavku jsou to záložky:

**Request** – Zde je možné zvolit metodu (GET, POST, PUT, DELETE), která se použít pro odeslání dotazu. Dále zde zadáváme adresu serveru, na který má být požadavek odeslán. V roletce „Content-Type“ vybíráme formát pro komunikaci. V případě, že použijeme některou z metod, které na server odesílají data, můžeme jej do zprávy vložit jako „Custom body“.

**Parameters** – Jak už název napovídá, zde je možné definovat parametry, které chceme serveru spolu s požadavkem odeslat. Při definování parametru je důležitý výběr jeho typu („Kind“). Ten totiž určuje jak bude serveru předán a tedy jak jej server bude zpracovávat.

**Authentication** – Zde se můžeme zvolit způsob autentizace uživatele (možné hodnoty jsou "none", "basic", "simple", "oauth" nebo "oauth2"). Pro protokoly "oauth" a "oauth2" je k dispozici vestavěný průvodce konfigurací.

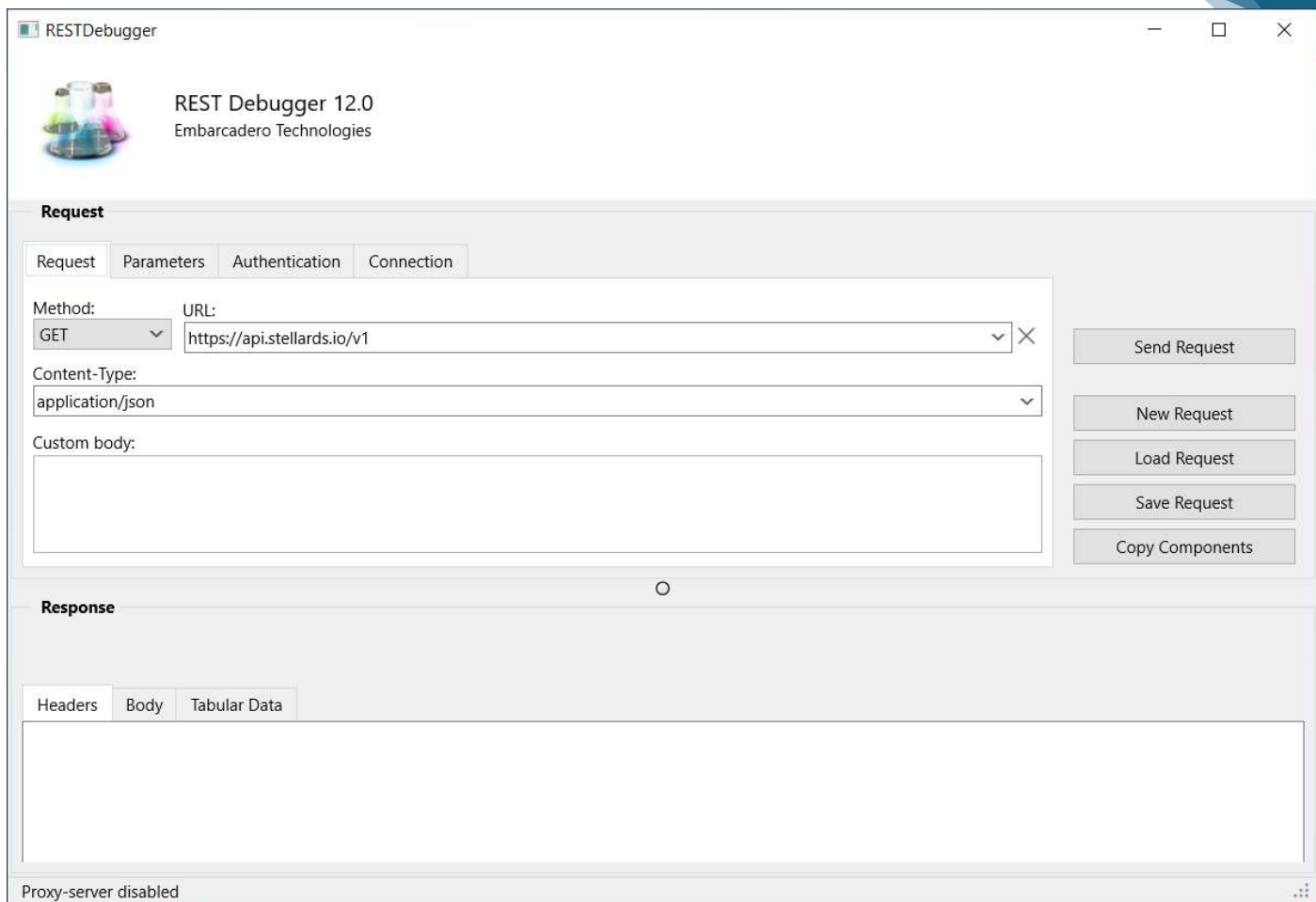
**Connection** – Na této záložce lze nastavit vše co se týká připojení. Určit lze verzi SSL (Secure Sockets Layer) nebo TLS (Transport Layer Security), nastavení týkající se případného použití proxy serveru, nebo specifikovat čas pro vypršení relace (Request Timeout).

Zpracování odpovědi (oblast „Response“) je rozděleno do třech záložek:

**Headers** – Odpověď serveru se skládá ze dvou částí. Hlavičky („header“) a těla zprávy („body“). V záložce „Headers“ tak nalezneme informace týkající se zpracování. Například návratový kód, počet vložených záznamů a podobně. V případě chyby zde můžeme najít informace, které nám mohou pomoci chybu identifikovat a odstranit.

**Body** – Zde jsou vlastní data, které server v reakci na požadavek odeslal. Podle zvoleného typu se může jednat o text, JSON, XML a podobně.

**TabularData** – Pokud je tělo zprávy ve formátu JSON, můžete v této záložce určit element, který obsahuje tabulková data („Root element“). Nastavit zde také můžeme interpretaci datových typů získané JSON zprávy.



Obrázek 16 - Prostředí nástroje REST Debugger

REST Debugger použijeme pro otestování přístupu k tabulkám uloženým ve službě StellarDS.io a ke vkládání nových záznamů

## Načtení dat z tabulky Dodavatel

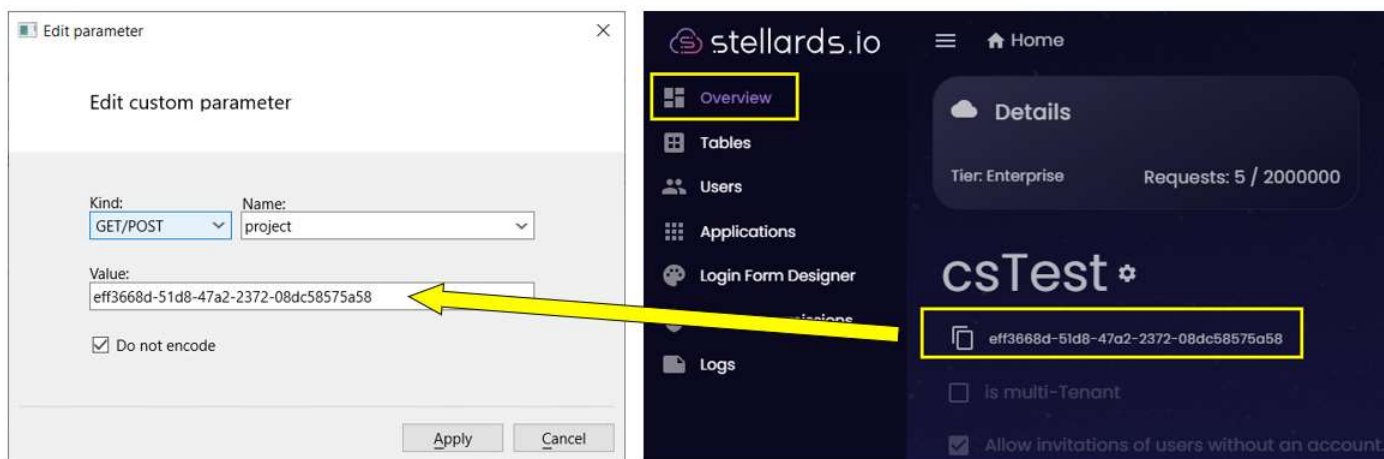
V záložce „Request“ zvolíme metodu „GET“ pro načtení dat. URL služby TMS StellarDS.io je „https://api.stellards.io/v1“.



Obrázek 17 - Nastavení REST požadavku

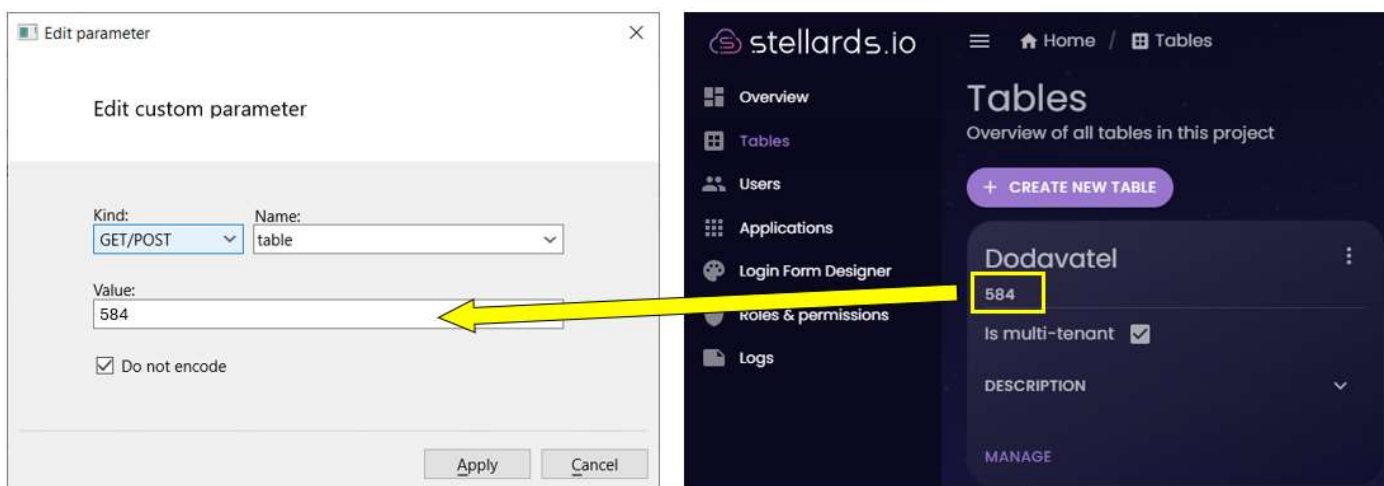
Nyní se přepneme do záložky „Parameters“. Do pole „Resource“ zapíšeme cestu k požadované funkci („RESTpointu“), v tomto případě tedy „data/table“ (služba pro získání dat z tabulky). Nyní ale potřebujeme předat serveru informace, které zajistí, že nám budou odeslána naše data a z tabulky, kterou určíme. K tomu budeme potřebovat vytvořit tři parametry:

**Projekt** – Parametr „project“ nám umožní předat službě „ProjectID“, které identifikuje námi vytvořený projekt. Pod účtem StellarDS.io jej naleznete v sekci „Overview“ pod názvem projektu:



Obrázek 18- Nastavení parametrů (Project ID)

**Tabulka** – Pomocí parametru „table“ sdělíme službě z které tabulky má data číst. Hodnotou parametru bude „TableID“ (sekce „Tables“, pod názvem tabulky).



Obrázek 19 - Nastavení parametrů (Table ID)

**Autorizace** – Posledním parametrem je „Authorization“. Pozor, tento parametr musíme serveru předat v hlavičce požadavku („HEADER“) a **před vlastním autorizačním klíčem musí být prefix „Bearer“**. V účtu StellarDS.io jej naleznete v sekci „Applications“. Token získáte při přidání aplikace. Pokud již byla aplikace přidána a token jste si nepoznámali, stačí kliknout na editační tlačítko.



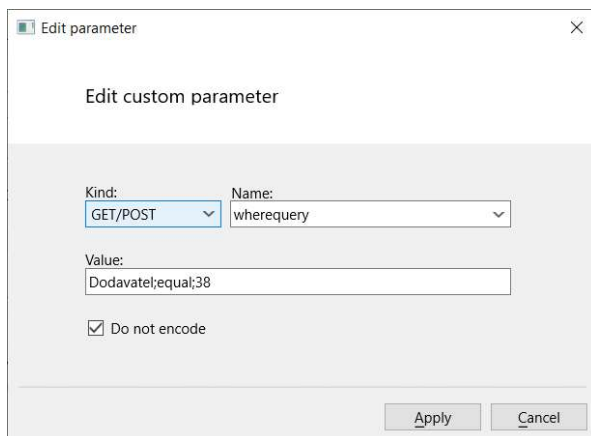
## Načtení dat z tabulky Produkt

Postup je naprosto totožný se získáním dat o dodavatelích. Jediným rozdílem bude to, že budeme chtít získat pouze informace o produktech vybraného dodavatele. V SQL bychom využili klauzuli WHERE. Podobnou možnost nám nabízí i StellarDS.io.

Nebudeme-li počítat autorizaci, použili jsme doposud parametry „project“ a „table“. Použít však můžeme i další. Zde je jejich úplný výčet:

- Projekt – ID vašeho projektu (viz obr. 16)
- Tabulka – ID tabulky, ze které chcete data (viz obr. 17)
- Offset (nepovinný) - Celočíslná hodnota určující, kolik záznamů od prvního se má z odpovědi vyloučit
- Take (nepovinný) - Celočíslná hodnota určující, kolik záznamů má server vrátit
- sortQuery (nepovinný) - Řetězec pro zakódování do URL. Dodržet je třeba následující tvar: JMÉNO\_POLE;asc. Můžete třídit podle více polí oddělením dotazu znakem '&' (FIELD\_A;asc&FIELD\_B;desc). Počet polí, podle kterých můžete třídit, není omezen, ale každé pole lze použít pouze jednou.
- whereQuery (nepovinný) - Řetězec pro zakódování do URL. Je třeba použít následující tvar: FIELD\_NAME;OPERÁTOR;HODNOTA. Filtruje data dle zadaných kritérií.
- joinQuery (nepovinný) - Řetězec pro zakódování do URL. Požadovaný tvar: TABLE;ID\_CIZIHO\_KEY\_TABLE\_TO\_JOIN;ID. Umožňuje vám spojit 2 tabulky a načíst data v 1 požadavku.

Pro získání produktů vybraného dodavatele tedy použijeme parametr „whereQuery“ v kombinaci s operátorem „equal“. Kromě „equal“ pro rovnost jsou k dispozici operátory „like“, „in“, „largerthan“, „smallerthan“, „or“ a „and“.



Obrázek 23 - Definice parametru "wherequery"

Nyní máme odladěné požadavky pro zobrazení dat se vztahem Master/Detail. V aplikaci budeme samozřejmě hodnoty parametrů zadávat dynamicky. Nyní se ještě pojďme podívat na další CRUD operace, jako je vkládání, aktualizace nebo mazání záznamů.

## Vkládání dat

Pro vložení nového záznamu jsou použity stejné parametry, tedy „project“, „table“. Autorizace je shodná pro všechny operace. Co je třeba změnit:

- http metodu: Pro vkládání záznamů je třeba použít metodu „POST“
- Custom body: Při vkládání, nebo aktualizaci dat je třeba serveru předat data v odpovídajícím formátu. V našem případě je to JSON:

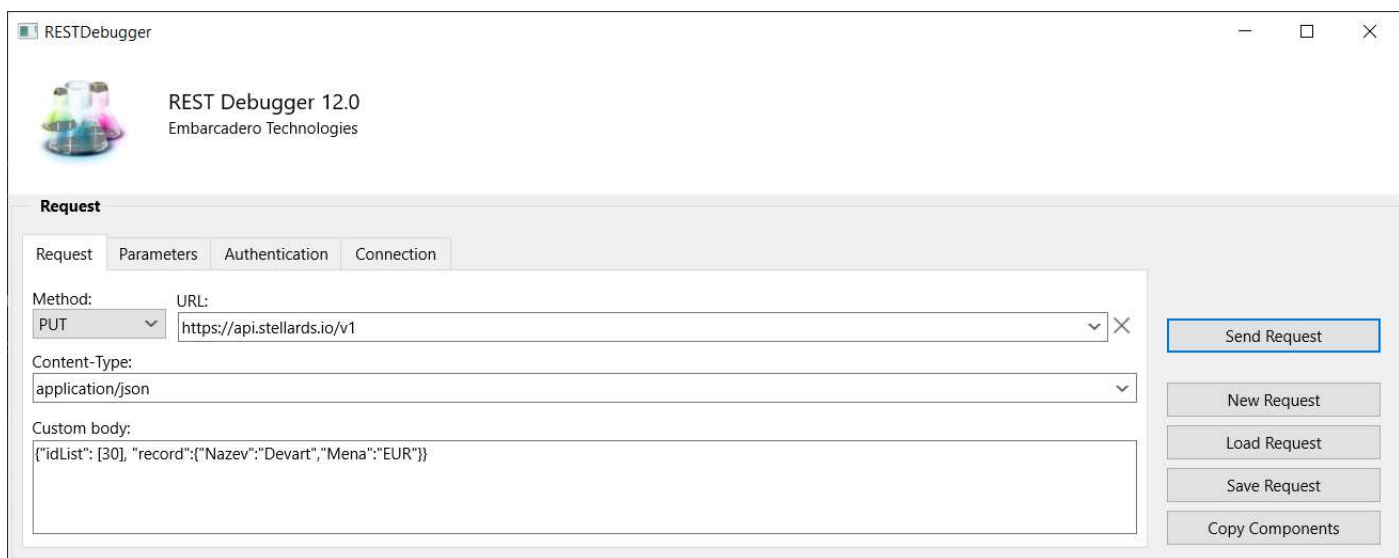


Obrázek 24 - Požadavek s daty pro vložení

Požadavek odešleme pomocí tlačítka „Send Request“. Jako odpověď získáme návratový kód (200 pro úspěšném volání) a rekapitulaci vložených dat. Pokud se něco nepodaří (chybové kódy 4xx nebo 5xx), obdržíme stručný popis chyby. Přehled chybových kódů naleznete například zde: [https://profitserver.net/knowledge-base/codes\\_http\\_error/](https://profitserver.net/knowledge-base/codes_http_error/). Při vkládání stejně jako během importu nezadáváme hodnoty ID, protože je server přidělí automaticky.

## Aktualizace dat

Aktualizace (Update) existujících záznamů se liší použitou metodou (pro update je to „PUT“) a předanými daty. Nyní je třeba serveru sdělit, které záznamy mají být změněny. Kromě nových dat je tak třeba serveru zaslat také primární klíč (ID záznamu), v kterém mají být změny provedeny:



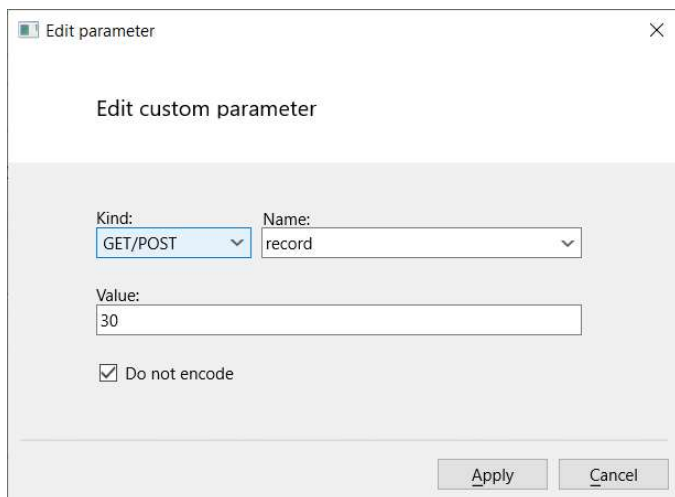
Obrázek 25 - požadavek s daty pro aktualizaci záznamu

Odeslání požadavku a odpověď serveru je stejná jako u vkládání dat.

## Mazání dat

Pro odstranění dat použijeme metodu „DELETE“. Když ji v prostředí REST Debuggeru vyberete, znepřístupní se pole pro vložení dat. Protože však potřebujeme serveru předat informaci, který záznam nebo záznamy má z databáze odstranit, musíme to udělat prostřednictvím parametru:





Obrázek 26 - Definice parametru "record" pro předání ID záznamu ke smazání



Obrázek 27 - Rekapitulace zadaných parametrů

## Save / Load Request

Všechny odesílané požadavky si můžete uložit pomocí tlačítka „Save Request“ a kdykoliv později se k nim vrátit a načíst je do prostředí REST Debuggeru pomocí kliknutí na „Load Request“.

## Volba „Copy Components“

REST Debugger má také jednu funkci navrženou speciálně pro RAD Studio. Volba „Copy components“ vygeneruje a vloží do schránky (clipboardu) kód pro vložení potřebných komponent a nastavení jejich vlastností, které lze následně vložit na formulář. Tuto možnost můžeme využít při tvorbě klientských aplikací v Delphi.

## Závěr

Jak rychle vytvořit aplikaci pro práci s daty spravovanými cloudovým řešením StellarDS.io si ukážeme v připravované, druhé části tohoto dokumentu.