# Zadání úkolu

Vytvořte webovou aplikaci, která se bude připojovat na REST API ABRA Flexi a umožní uživateli pohodlné rozúčtování dokladů na základě pravidel, které si sám bude moci spravovat. Aplikace bude sloužit jako doplněk systému ABRA Flexi.

Rozúčtováním se rozumí uložení dokladu s nový rozložením položek (řádek) dokladu tak, aby celková částka zůstala původní a aby na jednotlivých položkách dokladu bylo uvedeno jiné středisko. **Základní pravidla** jsou:

* celková částka na jedno středisko
* poměrná procentní část celkové částky na jedno středisko
* absolutní částka na jedno středisko
* zbytek celkové částky na jedno středisko

Pravidla je možné kombinovat, např. faktura přijatá za měsíční nájem obchodních prostor se rozúčtuje vždy 40 % na středisko A, 5000 Kč na středisko B a zbytek na středisko C. Tuto **sadu pravidel rozúčtování** bude možné uložit volitelně **s podmínkou** určující doklad, např. doklad je vystaven na firmu XYZ a v popisu dokladu je slovo “nájem”, tak aby bylo možné sadu uplatnit automaticky na doklad vystavený příští měsíc.

Aplikace se rovněž může snažit **navrhnout pravidla rozúčtování automaticky** na základě položek podobných dokladů, zaúčtovaných v minulosti. Například při zpracování faktury na firmu XYZ s částkou 1000 Kč se najde v minulosti několik podobných dokladů na stejnou firmu a stejnou částku, jejichž položky budou shodně zaúčtované 400 Kč na středisko B a 600 Kč na středisko C. Uživateli se navrhne automaticky sada pravidel 40 % na středisko B a zbytek na středisko C.

# Krátce o problematice

Firmy bývají interně rozdělené na střediska, která představují “firmy ve firmě” tj. samostatné jednotky, které si mimo jiné sledují náklady a výnosy. Na všech dokladech, které vstupují do účetnictví, je možné určit středisko na úrovni položky (řádku) každého dokladu.

Nákupy a zpracování přijatých faktur bývá centralizované, kdy se doklad zanese do systému často jednou částkou a následně je třeba tuto částku rozdělit na různá střediska. To jak se náklady rozdělují bývá dohodnuto v rámci firmy podle typu nákladu a příliš často se nemění, např. na fakturu za měsíční nájem se uplatňuje poměrné rozdělení podle velikosti využité plochy každého střediska, nebo měsíční platba za internet podle počtu zaměstnanců středisek. Rozúčtování se zpět do systému může zapsat několika způsoby, pro náš případ plně vyhoví modifikace původního dokladu s požadovaným rozpadem původní částky do více položek (řádků) pod různými středisky.

# Co očekáváme

## MUST HAVE

* aplikace si přečte z URL identifikaci dokladu, načte jej a zobrazí základní údaje: celková částka, firma a popis
* aplikace najde **vhodnou sadu pravidel** na základě podmínek uložených sad a uplatní jí
* pokud se nenajde žádná vhodná sada pravidel, nabídnou se k uplatnění existující sady
* pokud se neuplatní žádná sada pravidel (nebo žádná ještě neexistuje), bude moci uživatel doklad rozúčtovat ručně; **ručním rozúčtováním** se rozumí rozdělení částky postupně podle základních pravidel a výběr střediska pro každé použité pravidlo
* po ručním rozúčtování se navrhne uživateli **uložení použitých pravidel do pojmenované sady** a volitelně určení podmínky pro automatické uplatnění; uživatel by měl pouze zaškrtat navrhnuté podmínky za firmu z dokladu, celkovou částku a text v popisu (analogie pro pravidla přijaté pošty v emailových klientech)
* pokud již má doklad položky, tak do rozúčtování vstupuje jen celkovou částkou a při uložení jsou **původní položky nahrazeny novými**
* nesmí dojít ke změně **celkové částky** na dokladu
* nesmí dojít k uložení **záporné částky** na položce dokladu (např. uplatněním základního pravidla s vysokou absolutní částkou na doklad s nižší částkou)
* důraz je na uživatelskou přívětivost UI, přehlednost a rychlost ovládání

## SHOULD HAVE

* při ručním rozúčtování **aplikace dokáže navrhnout pravidla** pro rozúčtování sama podle rozúčtování podobných dokladů v minulosti; najde doklady se shodnou firmou a částkou nebo popisem a z jejích položek navrhne pravidla automaticky; upřednostňuje návrh pravidla s procentní částí, pokud procento vychází na celé číslo
* uživatel může jednoduše **spravovat sady pravidel**; může je mazat a přejmenovávat

## NICE TO HAVE

* vstupem do rozúčtování není pouze jeden ale **více dokladů**, opět jsou jejich identifikace v URL, tyto doklady se pak zpracovávají postupně
* pokud je sada pravidel uplatňována automaticky, uživatel volitelně může před uložením dokladu **upravit ručně výsledné částky**; tyto změny mají pouze vliv na toto uložení dokladu; stále platí že se nesmí celková částka změnit
* ve správě sad pravidel je uživatel může duplikovat, měnit podmínky, měnit samotná pravidla
* v rámci rozúčtování uživatel bude moci určovat více atributů, nejen středisko ale i zakázku, činnost, případně konkrétní čísla účtů

## WILL NOT HAVE

* není třeba řešit přihlašování, v URL dostane aplikace autentizační token pro komunikaci s API

# Co je ABRA Flexi

Jedná se o informační systém pro vedení firemní agendy pro střední a menší firmy, je možné ho používat on-premise, ale také v cloudu, což bude i náš případ. Uživatelé se k němu mohou připojit buď pomocí desktopové aplikace nebo webového rozhraní, které najdete na adrese:

<https://demo.flexibee.eu/flexi/demo/>

Přihlašovací údaje dostanete po odstartování soutěže. Kromě UI je možné připojit se, číst a zapisovat data také pomocí REST API, jehož dokumentaci naleznete na:

<https://demo.flexibee.eu/devdoc>

ABRA Flexi umožňuje uživatelům ze svého UI vyvolat tzv. doplňky (označované v desktopovém klientu jako **uživatelská tlačítka** a ve webovém jako **uživatelské akce**), což jsou samostatné webové aplikace, do kterých se předají informace o právě zobrazeném záznamu a přihlášeném uživateli. Doplněk pak s využitím REST API realizuje nějakou rozšiřující funkci.

# Poznámky k technickému řešení

Každý doplněk je třeba registrovat do uživatelského rozhraní ABRA Flexi. Uživatel pak doplněk vyvolá uživatelským způsobem v příslušné agendě nad aktuálním záznamem nebo nad více označenými záznamy. Vaše aplikace pak typicky dostane jako parametry v URL tyto údaje:

* authSessionId
  + autentizační token pro komunikaci s REST API
* companyUrl
  + začátek URL adresy pro komunikaci s REST API
* objectId
  + identifikace dokladu, použije se v případě vyvolání doplňku nad jedním dokladem
* objectIds
  + seznam identifikací dokladů oddělených čárkou, použije se v případě vyvolání doplňku nad seznamem označených dokladů

Na základě předaných údajů můžete získat celý doklad z REST API po sestavení URL adresy požadavku. Příklad takové adresy pro fakturu s identifikací 123 může vypadat takto:

[https://demo.flexibee.eu/v2/c/demo/faktura-prijata/**123**.json](https://demo.flexibee.eu/v2/c/demo/faktura-prijata/123.json)

Důležité: pro přístup do REST API je třeba se přihlásit, vaše aplikace dostanete v URL přihlašovací token, který je třeba předávat do API s každým requestem. Uděláte to buď uvedením query parametru authSessionId.

Ve výstupu tohoto požadavku nebude úplný výčet polí dokladu, to lze ovlivnit pomocí query parametru detail, viz dokumentace <https://demo.flexibee.eu/devdoc/detail-levels> např. pro úplný výčet všech polí to bude:

[https://demo.flexibee.eu/v2/c/demo/faktura-prijata/123.json?**detail=full**](https://demo.flexibee.eu/v2/c/demo/faktura-prijata/123.json?detail=full)

Dokumentace k jednotlivým polím najdete na samodokumentačním zdroji properties:

<https://demo.flexibee.eu/v2/c/demo/faktura-prijata/properties>

[https://demo.flexibee.eu/v2/c/demo/faktura-prijata-polozka/properties](https://demo.flexibee.eu/v2/c/demo/faktura-prijata-polozky/properties)

Kromě úrovně detailu lze ovlivnit výstup REST API v mnoha dalších směrech, například pro seznam prvních 10 faktur, vystavených po 1. 1. 2023 na konkrétní firmu to bude:

[https://demo.flexibee.eu/v2/c/demo/faktura-prijata/**(firma eq 8712 and datVyst gte 2023-01-01)**.json?&**order**=datVyst&**start**=0&**limit**=10](https://demo.flexibee.eu/v2/c/demo/faktura-prijata/(firma%20eq%208712%20and%20datVyst%20gte%202023-01-01).json?detail=full&top=)

Důležité: pokud neuvedete parametr limit, bude výstup automaticky omezen za 20 záznamů, pokud chcete celý seznam, který vyhovuje podmínce, vždy uvádějte **limit=0**.

Pokud budete chtít rozvinout relace a získat v rámci jednoho požadavku data z odkazovaných entit, můžete použít query parametr includes a v parametru detail specifikovat která pole vás zajímají. Například pro rozvinutí kódu a názvu střediska na faktuře:

[https://demo.flexibee.eu/v2/c/demo/faktura-prijata/123.json?**includes**=**/faktura-prijata/stredisko** &detail=custom:id,kod,datVyst,**stredisko(id,kod,nazev)**](https://demo.flexibee.eu/v2/c/demo/faktura-prijata/123.json?includes=/faktura-prijata/stredisko%20&detail=custom:id,kod,datVyst,stredisko(id,kod,nazev))

Více informací k sestavování požadavků, filtrování, řazení apod. najdete v dokumentaci <https://demo.flexibee.eu/devdoc> a dobrým zdrojem může být i sledování sítě pomocí Chrome DevTools provozu webového klienta ABRA Flexi

# Zápis do REST API

Odešle se požadavek metodou POST na URL zdroje REST API serveru, například pro fakturu přijatou vypadá požadavek takto:

{  
 "winstrom": {  
 "@version": "1.0",  
 "faktura-prijata": [  
 {  
 "id": "123",  
 "bezPolozek": false,  
 "polozkyFaktury@removeAll": true,  
 "polozkyFaktury": [  
 {  
 "mnozMj": 1,  
 "cenaMj": 10000,  
 "typSzbDphK": "typSzbDph.osvobozeno",  
 "kopStred": false,  
 "stredisko": "1"  
 },  
 {  
 "mnozMj": 1,  
 "cenaMj": 2000,  
 "typSzbDphK": "typSzbDph.osvobozeno",  
 "kopStred": false,  
 "stredisko": "2"  
 }  
 ]  
 }  
 ]  
 }  
}

Není třeba uvádět všechna pole, zápis inkrementální, tj. je možné zapsat pouze ta pole, která se změnila. Toto platí i pro kolekce, tj. objekty posílané do kolekce ji za normálního režimu rozšiřují, pokud potřebujete kolekci zcela nahradit, je nutné zároveň poslat příznak polozkyFaktury@removeAll a v kolekci poslat všechny položky položek znovu.

Více o aktualizacích viz <https://demo.flexibee.eu/devdoc/partial-updates>

V případě polí typu relace je možné kromě interního identifikátoru uvést přirozený, např. v případě kdy chcete uložit relaci na středisko A, můžete identifikaci střediska zapsat jako "code:A"

Více o identifikátorech záznamů viz <https://demo.flexibee.eu/devdoc/identifiers>

# Další zdroje REST API

Práce s dalšími zdroji je obdobná jako v případě faktur, pravděpodobně budete potřebovat nabídnout uživateli seznam středisek:

[https://demo.flexibee.eu/v2/c/demo/stredisko.json](https://demo.flexibee.eu/v2/c/demo/faktura-prijata/123.json?detail=full)

Dále budete zřejmě ukládat pracovní data vaší aplikace, jako jsou sady pravidel. Na to se hodí obecné úložiště dat, které je součástí REST API. Máte dvě možnosti, ukládat si data do společného úložiště (global-store).

Zápis se provádí obdobně jako v případě dokladů, jen je možné použít pro identifikaci záznamu klíč:

{  
 "winstrom": {  
 "@version": "1.0",  
 "global-store": [   
 {  
 "id":"key:moje-aplikace/pravidla",  
 "hodnota": "moje-hodnota”  
 }  
 ]  
 }  
}

Dokumentaci s úplným seznamem všech zdrojů najdete na:

<https://demo.flexibee.eu/v2/c/demo/evidence-list>