

IS CESTa

Obsah

Zadejte název kapitoly (úroveň 1).	1
Zadejte název kapitoly (úroveň 2)	2
Zadejte název kapitoly (úroveň 3)	3
Zadejte název kapitoly (úroveň 1).	4
Zadejte název kapitoly (úroveň 2)	5
Zadejte název kapitoly (úroveň 3)	6

Bezpečnostní kategorie: Veřejné

1. Požadavky na úpravy

1. Propojení na Aplikaci změnových listů

PROCES STAHOVÁNÍ DAT Z AZL

Popis kroků, které by měly proběhnout během přenosu dat z AZL do IS CESTa.

1. IS CESTa bude stahovat data v dohodnutém intervalu. Může být nastaveno např. na 1x denně, nebo 1x za hodinu dle potřeby. Alternativně lze funkci zpřístupnit např. administrátorovi IS CESTa, aby mohl import dle potřeby spustit ručně (tlačítkem v aplikaci). Případně správci ZBV, aby měl možnost ručně aktualizovat data za danou stavbu.
2. Každý požadavek bude na straně IS CESTa auditován, kde bude uloženo:
 - a. Datum, čas a payload požadavku (jednoznačná specifikace požadavku poslaná do AZL), případně ID uživatele, pokud byl import spuštěn ručně
 - b. Kompletní odpověď, která byla poskytnuta ze systému AZL
3. Proběhne validace dat a data budou uložena do databáze ke zpracování

- a. Každý úkon bude reportován (úspěšný i neúspěšný)
 - b. Zaslání e-mailové notifikace s popisem problému (?)
4. Zpracování a vytvoření / aktualizace ZL
- a. U stavby musí být vložena smlouva na realizace SOD, na kterou je možné navázat ZL. Bez této podmínky není možné ZL vytvořit / aktualizovat.
 - b. Budou převzaty všechny ZL vystavené systémem AZL bez ohledu na stadium ZL ve stavu „New“ (případně „Cancelled“ viz. dále).
 - c. Aktualizace ZL bude provedena automaticky.
 - d. Automaticky načtená data ZL nebudou uživatelsky editovatelná.
 - e. Ručně pořízené ZL nebudou aktualizovány daty z AZL
 - f. V jedné stavbě může být uveden mix ZL a to ručně pořízených v IS C.E.Sta a zároveň automaticky načtených z AZL. Po načtení prvního ZL z AZL do stavby bude ruční pořízení nového ZL na této stavbě zakázáno.
 - g. V případě, že AZL bude ve stavu „Cancelled“ bude ZL v IS CESTa označen jako smazaný.
5. Stahování příloh k ZL
- a. U ZL načtených z AZL budou přiloženy pouze identifikátory případných příloh. Přílohy se nebudou automaticky stahovat, nebo ukládat do IS CESTa. Uživatel bude mít možnost si přílohu stáhnout odkazem do AZL. Tento odkaz vyvolá zavolání WS AZL, která přílohu poskytne.
6. Potvrzení o převzetí a zpracování dat z dané dávky zpět do AZL
- a. Potvrzují se jednotlivé ZL. Po korektním zpracování dat ZL poskytnutých WS AZL, bude prostřednictvím WS AZL potvrzeno zpracování na straně IS C.E.Sta. V AZL bude daný ZL označen, že byl zpracován.

POTŘEBNÉ ÚKONY V IS CESTa

Kroky, které bude potřeba upravit na straně IS CESTa

1. Vytvoření konektoru mezi AZL a IS CESTa na základě WS AZL
2. Vytvoření procesu zpracování a aktualizace dat
3. Zajištění editovatelnosti dat (data získaná z AZL nebude možné v IS CESTa editovat)
4. Synchronizace číselníků, které se neshodují mezi AZL a IS CESTa (viz. spárování číselníků)
5. Aktualizace dat na základě změn v číselnících
6. Vytvoření reportu, který bude přístupný vybraným uživatelům

7. Úprava sestav a výstupů, kde se provádí výpočty na základě způsobu zatřídění § (např. shrnutí ZBV a claimů, procesované ZL).
8. Úprava hlavičky ZBV, kde se provádí výpočty na základě způsobu zatřídění §.

SPÁROVÁNÍ ČÍSELNÍKŮ

Tabulky porovnávají rozdíly mezi číselníky v AZL a IS CESTa.

- Červeně označené jsou řádky, které jsou na straně IS CESTa, ale nejsou na straně AZL.
- Oranžové jsou na straně AZL, ale nejsou na straně IS CESTa.
- Zeleně jsou shodné.

Způsob zatřídění (§)

ID	Položka v IS CESTa	Ekvivalent v AZL	
1	JŘBU		
5	§222/5 nezbytné dodatečné práce	§ 222/5 nezbytné dodatečné práce	§222/5
6	§222/6 nepředvídané změny	§ 222/6 nepředvídané změny	§222/6
7	§222/7 záměny v položkovém rozpočtu	§ 222/7 záměny v položkovém rozpočtu	§222/7
8	§222/4 de minimis	§ 222/4 de minimis	§222/4
9	§222/3 změny ad hoc	§ 222/3 změny ad hoc	§222/3
10	§64 písm. b dodatečné dodávky řešené JŘBU	§ 64 písm. b	§64b
11	Dle článku 10.2 Směrnice č. 105 ZBV (měření)		
12	Veřejná zakázka dle Směrnice SŽDC č. 53 o zadávání veřejných zakázek	VZ dle Směrnice SŽ SM053	SM053
13	Otevřené řízení dle ZZVZ	Otevřené řízení dle ZZVZ	OtevŘizeníZZVZ
14	§100 odst. 1 ZZVZ vyhrazené změny	100 odst. 1 ZZVZ vyhrazené změny	100odst1ZVZZ
15	Claim		
16	Opce		
17	Covid		
18	Valorizace dle 13.8		
19	Valorizace komodity		
		Paragraf 1	par 226/1
		Paragraf 2	par 226/2
		Paragraf 3	par 226/3

Kategorizace ZBV

Tento číselník vnímáme jako podrobnější členění stávajícího číselníku. Pokud by byl dodán kompletní číselník, měl by být na straně IS CESTa doplněn a data by měla být přemapována na dané položky. Položky se nyní neshodují.

ID	Položka v IS CESTa	Ekvivalent v AZL	
132	1.1 - Mosty, tunely, propustky, podchody		
		Mosty a propustky	1.1.1
		Tunely	1.1.2
		Podchody	1.1.3
		Opěrné a zárubní zdi	1.1.4
133	1.2 - Kabelovody, horkovody, kolektory, kanalizace, vodovody		
		Kabelovody	1.2.1
		Horkovody	1.2.2
		Kolektory	1.2.3
		Kanalizace	1.2.4
		Vodovody	1.2.5
134	2 - Pozemní stavby/budovy, přístřešky, nástupiště, výtahy, eskalátory		
		Tunely	2.0.0
		Budovy/přístřešky	2.1
		Nástupiště vč. zastřešení	2.2
		Oplocení, zpevněné plochy	2.3
		Výtahy, eskalátory	2.4
135	3.1 - Železniční spodek	Železniční spodek	3.1
136	3.2 - Železniční svršek	Železniční svršek	3.2
137	4 - Sdělovací a zabezpečovací technika		
		Sdělovací technika	4.1
		Zabezpečovací technika	4.2
138	5 - Energetika a elektrotechnika		
		Energetika	5.1
		Elektrotechnika	5.2
139	6 - Ekologie, odpady, kácení		
		Ekologie	6.1
		Odpady	6.2
		Kácení	6.3
140	7 - Majetek 3. stran, přeložky sítí, vyvolané investice		

		Majetek 3. stran	7.1
		Přeložky sítí	7.2
		Vyvolané investice	7.3
141	8 - Pozemní a přístupové komunikace, PHS, DIO		
		Pozemní a přístupové komunikace	8.1
		HSP	8.2
		DIO	8.3
142	9 - Claimy, akcelerace, zpoždění		
		Claimy	9.1
		Akcelerace	9.2
		Zpoždění	9.3
143	10 - Průzkumy, výzkumy	Průzkumy, výzkumy	10
144	11 - Jiné/více/nelze předem určit	Jiné/více/nelze předem určit	11
		Mosty a propustky	1.1.1
		Tunely	1.1.2
		Podchody	1.1.3
		Opěrné a zárubní zdi	1.1.4

Požadavek od

Prosím o upřesnění, o co se v kapitole „Požadavek od“ v dokumentaci AZL jedná.

V dokumentaci k AZL je uvedeno, že je zde rozdíl mezi AZL a IS CESTa.

Číselník A/N:

Stav ZL je na straně AZL rozdělen do 8 fází, kde např. fáze 1–3 jsou v IS CESTa pod jednou položkou „Připravuje se“. V dodávaných datech by mělo být poskytováno o jakou fázi se konkrétně jedná jako další položka.

ID	Položka v IS CESTa	Ekvivalent v AZL	
1	ANO	Fáze 6	Podepisování
	ANO	Fáze 7	Příprava dodatku
	ANO	Fáze 8	Dokončeno
2	NE		
3	ANO - v I. kole		
4	V podpisovém kole	Fáze 4	Textová kontrola
	V podpisovém kole	Fáze 5	Schvalování
5	Připravuje se	Fáze 1	Cenová příprava
	Připravuje se	Fáze 2	Cenová kontrola
	Připravuje se	Fáze 3	Textová příprava

PŘIPOMÍNKY A PODNĚTY K WS AZL

- Ve výsledných datech z WS AZL poskytovat i konkrétní identifikaci fáze ZL jako další hodnotu.
- Bylo by vhodnější, aby základní metoda, která vrací hlavičky prvních 50 zpráv ve stavu „New“, aby vracela i ZL ve stavu „Cancelled“. Tím by bylo zajištěno, že dostaneme i zprávu o tom, že ZL na straně IS CESTa by měl být označen za smazaný. Bez nutnosti službu volat znovu jen za účelem stornovaných ZL.

2. Optimalizace SQL dotazů

Na základě požadavků uživatelů optimalizovat front-end aplikace a zrychlit komunikaci se serverem pro tři modulů:

1. **Plán**
2. **Požadavky**
3. **Administrace**

Postup:

2.1. Analýza výkonu

- Změřit aktuální dobu načítání jednotlivých modulů a identifikovat nejpomalejší části.
- Analyzovat síťovou komunikaci mezi front-endem a serverem (velikost požadavků, opakované dotazy, složité vyhledávání).
- Prozkoumat výkon renderování komponent a modulů (např. velké tabulky, složité grafy, opakované re-rendery).

2,2. Optimalizace front-endu bude provedena následujícím způsobem

- Minimalizace a optimalizace JavaScriptu a CSS (Tree Shaking, Code Splitting).
- Implementace lazy loading pro moduly a velké komponenty.
- Efektivní práce se stavem aplikace (omezit nadměrné re-renderování).

- Redukce nepotřebných DOM operací a optimalizace práce s velkými datovými sadami.
- Zlepšení využití cachování na úrovni prohlížeče (např. Cache-Control, Service Workers).

3. Optimalizace komunikace se serverem

- Redukce zbytečných API volání ve všech modulech (optimalizace načítání dat).
- Implementace server-side paginace a efektivnějšího načítání velkých datasetů.
- Komprese odpovědí ze serveru u modulů, kde lze provést (Gzip, Brotli).
- Zavedení HTTP/2 nebo HTTP/3 pro efektivnější přenos dat.
- Možnost využití GraphQL nebo jiného mechanismu pro minimalizaci zbytečného přenosu dat.
- Implementace WebSocketů pro real-time aktualizace tam, kde je to vhodné.

4. Testování a nasazení

- Provést zátěžové testy po optimalizacích.
- Ověřit snížení doby odezvy a načítání pomocí měření (Lighthouse, Chrome DevTools, WebPageTest).
- Zajistit kompatibilitu optimalizací s existujícími funkcemi modulů.

Očekávaný výstup:

- Výrazné zrychlení načítání a odezvy modulů.
- Snížení zatížení serveru a klientských zařízení.
- Potvrzení ze strany uživatelů - plynulejší práci s aplikací.

3. Přejít stávajícího modulu tiskových sestav na PHP 8.3.+

Stávající modul tiskových sestav je napsán ve starší verzi PHP 7. Cílem je přeprogramovat jej do PHP 8+ s důrazem na modernizaci kódu, zlepšení výkonu, bezpečnosti a snadnější údržbu.

3.1 Technické požadavky

- Použití PHP 8.3+.
- Odstranění zastaralých funkcí a nekompatibilních prvků.
- Optimalizace výkonu – odstranění nadbytečných SQL dotazů, efektivní využití cache.
- Zvýšení bezpečnosti – ochrana proti SQL injection, XSS, CSRF.

3.2 Funkční požadavky

- Zachování původní funkčnosti tiskových sestav.
- Možnost generování výstupů ve formátech požadovaných uživateli.
- Podpora více jazyků pomocí lokalizačních souborů (např. gettext nebo jiný systém).

- Možnost exportu a odeslání sestav e-mailem.
- Přehledné uživatelské rozhraní s možností filtrování a nastavení parametrů tisku.
- Podpora připojení k databázi (MySQL/MariaDB) a možnost získávání dat z více tabulek.

3.3 Technologie a knihovny

- **PHP 8+**
- **Framework:** Laravel (pokud je možné přejít na framework), jinak modernizace existujícího kódu.
- **PDF generátor:** DomPDF, TCPDF nebo FPDF.
- **Excel/CSV export:** PhpSpreadsheet.
- **Databázová vrstva:** PDO nebo použití ORM (Eloquent, Doctrine).
- **Šablonovací systém:** Blade (pokud Laravel) nebo Twig.

3.3 Refaktoring kódu

- Nahrazení zastaralého procedurálního kódu objektově orientovaným přístupem.
- Použití moderních PHP vlastností (typové deklarace, pojmenované argumenty, atributy, match místo switch).
- Oddělení logiky od prezentace (MVC architektura, pokud to projekt umožní).
- Optimalizace dotazů do databáze (použití indexů, JOIN operací místo více dotazů).

3.4 Testování

- Jednotkové testy (PHPUnit).
- Integrační testy pro ověření propojení s databází a generování výstupů.
- Zátěžové testy pro ověření výkonu při generování velkých sestav.
- Testy zabezpečení (SQL injection, XSS, CSRF).

3.5 Nasazení

- Příprava prostředí pro testování.
- Převod uživatelů na nový systém bez výpadku.

4. Optimalizace rychlosti jednotlivých modulů při přístupu k datům

Na základě požadavků uživatelů optimalizovat front-end aplikace a zrychlit komunikaci se serverem pro čtyři moduly:

5. **InvestDokument**
6. **SUD**
7. **IS CESTa**
8. **IS CESTa_OUA**

Postup:

4.1. Analýza výkonu

- Změřit aktuální dobu načítání jednotlivých modulů a identifikovat nejpomalejší části.
- Analyzovat síťovou komunikaci mezi front-endem a serverem (velikost požadavků, opakované dotazy, složité vyhledávání).
- Prozkoumat výkon renderování komponent a modulů (např. velké tabulky, složité grafy, opakované re-rendery).

4.2. Optimalizace front-endu bude provedena následujícím způsobem

- Minimalizace a optimalizace JavaScriptu a CSS (Tree Shaking, Code Splitting).
- Implementace lazy loading pro moduly a velké komponenty.
- Efektivní práce se stavem aplikace (omezit nadměrné re-renderování, využít memoizaci).
- Redukce nepotřebných DOM operací a optimalizace práce s velkými datovými sadami.
- Zlepšení využití cachování na úrovni prohlížeče (např. Cache-Control, Service Workers).

4.3. Optimalizace komunikace se serverem

- Redukce zbytečných API volání ve všech modulech (optimalizace načítání dat).
- Implementace server-side pagination a efektivnějšího načítání velkých datasetů.
- Komprese odpovědí ze serveru u modulů, kde lze provést (Gzip, Brotli).
- Zavedení HTTP/2 nebo HTTP/3 pro efektivnější přenos dat.
- Možnost využití GraphQL nebo jiného mechanismu pro minimalizaci zbytečného přenosu dat.
- Implementace WebSocketů pro real-time aktualizace tam, kde je to vhodné.

4.4. Testování a nasazení

- Provést zátěžové testy po optimalizacích.
- Ověřit snížení doby odezvy a načítání pomocí měření (Lighthouse, Chrome DevTools, WebPageTest).
- Zajistit kompatibilitu optimalizací s existujícími funkcemi modulů.

Očekávaný výstup:

- Výrazné zrychlení načítání a odezvy modulů.
- Snížení zatížení serveru a klientských zařízení.

- Potvrzení ze strany uživatelů - plynulejší práci s aplikací.

1.1. Popis současného stavu

Funkčnosti nejsou v současnosti v IS CESTa zapracovány

1.2. Návrh řešení

Viz popis požadavku

—