

Číslo nabídky: 202412084OSC

Nabídka: „Intermodální plánovač trasy – Rozvojový požadavek č. 2“

oict.

K rukám:
Ing. Lenka Řezáčová
vedoucí oddělení
Oddělení koncepce odbavení, projektů a projektového řízení
Integrovaná doprava středočeského kraje
Sokolovská 100/94
186 00 Karlín

Nabízející: Operátor ICT, a.s.

Operátor ICT, a.s. (dále jen „OICT“), je městskou společností, která pro Hlavní město Prahu zajišťuje odborné služby ICT a realizaci ICT projektů pro městské části, další městské společnosti, případně další subjekty.

Předmět nabídky

Předmětem nabídky, připravené na základě poptávky objednatele, je plnění dle Smlouvy o poskytování služeb multikanálového odbavovacího systému“, uzavřené 15. 1. 2018. Konkrétně jde o Rozvojový požadavek BE IPT: „Intermodální plánovač trasy – Rozvojový požadavek č. 2“.

ZNĚNÍ ROZVOJOVÉHO POŽADAVKU:

1. BE ukládá response (včetně requestů)

Předmětem tohoto zadání je **první fáze** implementace změn IPT. Cílem je provést základní změny architektury řešení které zefektivní datové toky FE <-> BE pro plnění návazných funkčních požadavků a optimalizuje náročnost implementace logiky na FE.

- V rámci tohoto zadání požadujeme řešit ukládání nalezených tras tak, aby dle potřeby na FE bylo možné načítat jak přehledová data o nalezených trasách, tak i jejich plný detail.
- Zároveň požadujeme implementovat ukládání tras tak, aby při dalších operacích nad nalezenými trasami bylo možné na BE IPT zasílat pouze identifikátor příslušných objektů (seznam tras, trasa, segment) a ne celý jeho obsah, jak je tomu nyní
- Zároveň v rámci toho požadavku požadujeme implementaci nových základních EP pro vyhledání, načtení a přenačtení vyhledaných tras.
- Pro zachování zpětné kompatibility s FE Lítačky požadujeme zachovat i stávající EP v rozsahu uvedeném níže.

Detailní funkční požadavky v rámci tohoto zadání:

Ukládání tras:

- Do databáze budou ukládány trasy tak, aby bylo optimalizováno množství přenášovaných a ukládaných dat
 - v databázi se budou ukládat pouze informace, které jsou nezbytně nutné k rekonstrukci trasy v plném detailu, bez nutnosti spustit vyhledávání trasy.
 - Počátek a konec každého segmentu
 - Dopravní mód
 - Identifikátor počáteční a koncové zastávky pro VHD segmenty
 - Trajektorii trasy v silniční, cyklistické a chodecké síti
 - Identifikátory využitých prostředků (linky, spoje, vozidla VHD, sdíleného vozidla, parkoviště kola, auta)
 - Výjimky z výše uvedeného pravidla:
 - v rámci požadavku na zobrazení dat objektů které mohou v realtime vstupních datech zmizet (např z důvodu vypůjčení sdíleného prostředku) požadujeme ukládat jejich plný detail
 - aktuálně požadujeme ukládat plný detail u sdílených prostředků mobility – kola, auta, koloběžky, motocykly
 - řešení musí počítat s potenciálním ukládáním dalších detailů navíc v rámci dalších změnových požadavků
- Stávající ukládání do databáze Mongo bude zrušeno včetně nasazení Mongo databáze.

Doba uložení tras a jejich automatické odmazávání:

- Požadujeme, aby trasy byly dle zadaných pravidel z databáze automaticky odmazávány dle následujících pravidel:
 - Trasa je odmazána 30 minut po čase jejího ukončení (arrival) nebo 30 minut od posledního načtení přes EP načtení trasy nebo EP načtení segmentu trasy (čas načtení je uložený v timestamp) dle toho, co nastane později.
 - Trasa ale nesmí být vymazána dříve, než 30 minut po jejím vyhledání.
 - RoutingResponse bude odmazán po odmazání poslední trasy k němu náležející
 - Session bude odmazána po odmazání poslední RoutingResponse k němu náležející

Endpointy:

Obecné požadavky:

- Požadujeme rozdělení do 2 úrovní detailu:
 - Přehled
 - obsahuje pouze redukováné informace, neposílají se zejména údaje o lokalizaci a geometrii a další objekty, detailní specifikace viz dále
 - Plný detail
 - obsahově zůstává stejný jako doposud se tím, že struktura se může v zájmu zpřehlednění a zjednodušení implementace změnit
 - viz k tomu také Implementační poznámky
- Všechny endpointy v rámci tohoto zadání i doplněné v dalších fázích, které provádí plánování a vrací seznam (List) tras vrací RoutingResponse tak, aby se mohl stahovat přehledový i detailní pohled přes GET endpointy uvedené dále.
- Všechny endpointy, které provádí plánování, ale vrací pouze trasu (t.j nikoliv seznam tras i když s jedním prvkem) vrací detailní data trasy.
- Všechny endpointy pro plánování a načtení uvedené dále obsahují dodatečný parametr removeRoutingRequest ovládající přítomnost routingRequestu ve stahovaných odpovědích.
- Obsah routingRequest pro vyhledání tras zůstává ve stávající podobě
- Všechny EP pro načtení uvedené dále, zasílají aktualizovaná real time data trasy

EP plánování trasy:**POST directions/**

RoutingResponse getDirections(RoutingRequest routingRequest)

- Slouží pro plánování tras, které probíhá zasláním požadavku RoutingRequest (ponechán ve stávající podobě) na plánovací endpoint. Po naplánování tras dojde k uložení tras do databáze a endpoint odpovídá pomocí objektu RoutingResponse.
- Objekt *RoutingResponse* je v základní struktuře zůstává stejný jako stávající *RoutingResponse* s tím, že se upravují/nahrazují zejména následující atributy/objekty:

Objekt RoutingResponse

Název sloupce	typ	popis
response_id	String	Identifikátor nalezené RoutingResponse
session_id	String	Identifikátor session
plannedRoutes	List<Planned Route>	Seznam nalezených tras v přehledovém formátu, viz Specifikace přehledových dat

Objekt PlannedRoute

Název sloupce	typ	popis
response_id	String	Identifikátor nalezené RoutingResponse
route_id	String	Identifikátor trasy v dané RoutingResponse
segments	List<Segment>	Seznam segmentů nalezené trasy v přehledovém formátu, viz Specifikace přehledových dat
departure	Timestamp	Čas odjezdu
arrival	Timestamp	Čas příjezdu
price	Float	Cena trasy

Objekt Segment

viz Specifikace přehledových dat

EP načtení Session:**GET directions/sessions/{sessionID}**

RoutingSession getSession(String sessionID)

- EP slouží pro načtení přehledového detailu tras z původní response (originalResponse) a tras která byli doplněny voláním EP pro vyhledání následujících/předcházejících tras. Jelikož v rámci tohoto zadání EP pro vyhledání následujících/předcházejících tras nebude implementován očekáváme že PreviousRoutes, NextRoutes budou vráceny jako prázdné seznamy.

Objekt RoutingSession

Název sloupce	typ	popis
sessionId	String	Identifikátor celé session
originalResponse	RoutingResponse	Původní odpověď na uživatelský plánovací požadavek
responses PreviousRoutes	List<RoutingResponse>	Seznam odpovědí na plánovací požadavky pro předcházející trasy

responses NextRoutes	List<RoutingResponse>	Seznam odpovědí na plánovací požadavky pro následující trasy
----------------------	-----------------------	--

GET directions/sessions/{sessionID}/detail
RoutingSessionDetail getSession(String sessionID)

- EP slouží pro načtení plného detailu tras z původní response (originalResponse) a tras která byli doplněny voláním EP pro vyhledání následujících/předcházejících tras. Jelikož v rámci tohoto zadání EP pro vyhledání následujících/předcházejících tras nebude implementován očekáváme že PreviousRoutes, NextRoutes budou vráceny jako prázdné seznamy.

Objekt RoutingSessionDetail		
Název sloupce	typ	popis
sessionId	String	Identifikátor celé session
originalResponse	RoutingResponseDetail	Původní odpověď na uživatelský plánovací požadavek
responses PreviousRoutes	List<RoutingResponseDetail>	Seznam odpovědí na plánovací požadavky pro předcházející trasy
responses NextRoutes	List<RoutingResponseDetail>	Seznam odpovědí na plánovací požadavky pro následující trasy

EP načtení RoutingResponse:

GET directions/{responseID}
RoutingResponse getDirections(String responseId)

- EP slouží pro načtení tras ze zadané response v přehledovém detailu

GET directions/{responseID}/detail
RoutingResponseDetail getDirections(String responseId)

- EP slouží pro načtení tras ze zadané response v plném detailu PlannedRouteDetail

EP načtení trasy:

GET directions/route/{routeID}
PlannedRouteDetail getPlannedRoute(String routeId)

- EP slouží pro načtení trasy v plném detailu

EP načtení segmentu:

GET directions/route/segment/{segmentID}
RouteSegmentDetail getRouteSegment(String segmentId)

- EP slouží pro načtení segmentu trasy v plném detailu

Požadavky na zachování stávajících EP:

- Budou dodány 2 verze API (a 2 swagger specifikace):

- stávající verze plně zachovávající funkčnost všech stávajících EP
- nová verze V2 obsahující EP z tohoto požadavku a vyjmenované EP funkčně stejné se starou verzí.
- Tabulka níže uvádí EP ve staré a nové verzi:

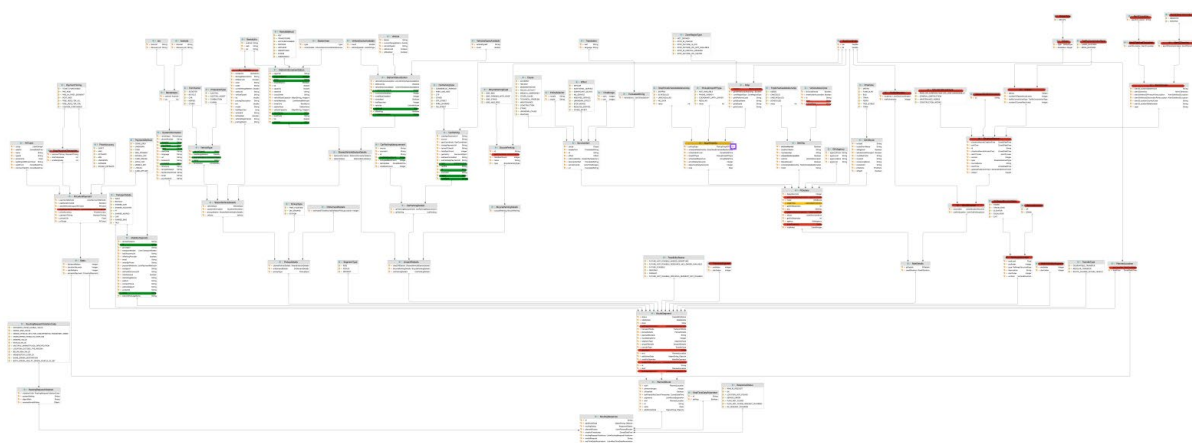
Stávající EP, které musí být zachované ve stávající verzi API	EP v nové verzi API V2	Doplňující komentář
	/*/V2/directions/	Nově doplněný EP v rámci tohoto požadavku
	/*/V2/directions/sessions/{sessionID}	Nově doplněný EP v rámci tohoto požadavku
	/*/V2/directions/sessions/{sessionID}/detail	Nově doplněný EP v rámci tohoto požadavku
	/*/V2/directions/{responseID}	Nově doplněný EP v rámci tohoto požadavku
	/*/V2/directions/{responseID}/detail	Nově doplněný EP v rámci tohoto požadavku
	/*/V2/directions/route/{routeID}	Nově doplněný EP v rámci tohoto požadavku
	/*/V2/directions/route/segment/{segmentID}	Nově doplněný EP v rámci tohoto požadavku
/*/isalive	/*/V2/isalive	
/*/intermodalrouting		ve V2 není potřeba
/*/intermodalrouting/ptonly		ve V2 není potřeba
/*/intermodalrouting/plannedroute/update		ve V2 není potřeba
/*/intermodalrouting/routesegment/update		ve V2 není potřeba
/*/intermodalrouting/trip/update	/*/V2/intermodalrouting/trip/update	parametry, funkčnost a struktura výstupních dat ve verzi V2 stejná jako ve stávající verzi API
/*/mobilityoperators	/*/V2/mobilityoperators	parametry, funkčnost a struktura výstupních dat ve verzi V2 stejná jako ve stávající verzi API
/*/settings	/*/V2/settings	parametry, funkčnost a struktura výstupních dat ve verzi V2 stejná jako ve stávající verzi API
/*/stations/bikesharing	/*/V2/stations/bikesharing	parametry, funkčnost a struktura výstupních dat ve verzi V2 stejná jako ve stávající verzi API
/*/stations/publictransport	/*/V2/stations/publictransport	parametry, funkčnost a struktura výstupních dat ve verzi V2 stejná jako ve stávající verzi API
/*/vehiclelocations/freefloating	/*/V2/vehiclelocations/freefloating	parametry, funkčnost a struktura výstupních dat ve verzi V2 stejná jako ve stávající verzi API
/*/vehiclelocations/publictransport	/*/V2/vehiclelocations/publictransport	parametry, funkčnost a struktura výstupních dat ve verzi V2 stejná jako ve stávající verzi API
/*/intermodalrouting/executeplannedroute		EP musí být zrušen i ve staré verzi

		protože vyžaduje ukládání do Mongo databáze kterou požadujeme zrušit
/*/intermodalrouting/next		ve V2 není potřeba
/*/intermodalrouting/previous		ve V2 není potřeba

- Požadujeme, aby řešení počítalo s tím, že v budoucnu budou staré EP, vyjmenované ve stávající verzi API výše, úplně zrušeny

Specifikace přehledových dat:

- pro specifikaci přehledového obsahu (nikoliv struktury) jsme zvolili proškrtání stávajícího datového modelu kde
 - červená – atribut/objekt se v přehledu neposílá
 - žlutá – u seznamu objektů se posílá pouze jeho část (konkrétně nástupní a výstupní zastávka)
 - zelená – v objektu se posílají atributy označené zeleně
 - navíc bude jak v přehledové, tak v detailní verzi přidán atribut obsahující údaj o počtu projížděných zastávek v rámci spojení.



Testovací aplikace:

- v rámci tohoto požadavku požadujeme vytvoření nové testovací aplikace kterou bude možné nasadit paralelně se stávající verzí testovací aplikace
- požadujeme v ní v plném rozsahu zachovat stávající funkčnost testovací aplikace s tím že:
 - nová testovací aplikace bude používat nové EP V2
 - funkce pro následující a předcházející spoje bude "zaslepena" do implementace nového EP pro next/prev spoje
 - testovací aplikace bude počítat s doplněním o nově implementované funkčnosti v dalších fázích viz Implementační poznámky.
- v rámci dodání nové testovací aplikace požadujeme dodání všech potřebných konfiguračních souborů pro běh a nastavení přístupů testovací aplikace
- Stará testovací aplikace bude dodána pouze pokud v ní budou změny z důvodu nahlášených chyb. V takovém případě zdrojové kódy staré a nové aplikace budou dodány odděleně.

Administrátorská aplikace:

- v rámci tohoto požadavku nepředpokládáme žádné změny v administrátorské aplikaci

Nefunkční požadavky:

- Struktura databáze pro ukládání a vyhledávání dat bude navržena tak, aby optimalizovala jejich rychlé ukládání a vyhledávání pro naplnění specifikovaných EP
- Pro ukládání tras bude použita relační databáze PostgreSQL

Implementační poznámky:

- Objednatel připouští, pokud to přispěje k lepší strukturovanosti a zpřehlednění návrhu API, v novém návrhu využít polymorfismus a kompozici (dědičnost) a to zejména u:
 - rozlišení přehled vs. detail
 - rozlišení typů segmentů trasy dle dle typů dopravních módů
 - podmínkou je vždy jasné kritérium pro určení typu objektu (použití discriminator, mapping)
- Názvy objektů, atributů, endpointů a cest ve V2 z tohoto zadání se můžou v detailním návrhu API změnit, pokud to přispěje k zpřehlednění návrhu
- Požadujeme dodat Swagger specifikaci k vyjádření do 6 týdnů od data objednání. Objednatel se pak nejpozději do 8 týdnů od data objednání k specifikaci vyjádří.
- Řešení v první fázi musí přihlížet k požadavkům, které můžou být zadány v dalších fázích, jedná se zejména o tyto funkce:
 - nová funkcionality vyhodnocování realizovatelnosti trasy
 - dohledávání předcházejících/následujících spojů pro zvolený segment dříve naplánované jedné trasy a jeho nahrazení nově nalezeným spojem
 - spočítání předcházejících/následujících tras pro dříve naplánované trasy a jejich doplnění do session
 - plánování tras na základě statických dat o VHD s tím, že informace z realtime dat budou pouze propagovány do naplánované trasy
 - umožní integrovat mezivrstvu, která bude zajišťovat prezentační logiku nad nalezenými RoutingResponse, Trasami, Segmenty (např. zaokrouhlování)
- Základní struktura objektů session, response, trasy, segmenty je daná tímto zadáním. Na úrovni jednotlivých vnořených objektů/ atributů nebo parametrů volání může v dalších iteracích dojít k požadavkům na změny a řešení s tím musí počítat.
- Struktura požadavku RoutingRequest bude ponechána ve stávající podobě. Objednatel pouze připouští přidání nepovinných atributů, pokud si to bude vyžadovat implementace nové funkce.
- Pokud to vyplývá z požadované funkcionality (čas posledního načtení detailu trasy nebo segmentu pro vyhodnocení odmazání trasy) tak EP pro načtení při jejich použití ukládají timestamp posledního načtení objektu, což vyžaduje zápis do databáze i v případě pouhého načtení objektu.

Akceptační kritéria:

- Dodanou verzi IPT lze nasadit a je jako celek funkční
- Je dodaná dokumentace (Swagger + popsaná struktura databáze)
- Nalezené objekty jsou ukládány do databáze PostgreSQL v rozsahu dle zadání a lze je načíst s využitím jejich id prostřednictvím zde specifikovaných EP
- Nalezené objekty jsou automaticky odmazávány dle popsaných pravidel
- Ukládání do Mongo databáze je odstraněno

- Nové EP zasílají korektní data v rozsahu a struktuře dle zadání
- Jsou zachovány a plně funkční stávající EP dle zadání
- Nová testovací aplikace poskytuje funkčnost dle požadavku

Poznámka: vychází z issue: <https://gitlab.com/operator-ict/ipt/umotional/-/issues/703>

2. Bikeparking – změna formátu + přidání adresy

Z důvodu sjednocení formátu mezi bike parking a car parking a ulehčení frontendu chceme změnit formát, v kterém poskytujeme bike parking. Nově je to validní geojson.

nový zdroj:

- TEST <https://deviptoict.blob.core.windows.net/storage/Bikeparking/Bikeparking.geojson>
- PROD <https://iptoict.blob.core.windows.net/storage/Bikeparking/Bikeparking.geojson>

ukázka:

```
{
  "type": "FeatureCollection",
  "features": [
    {
      "type": "Feature",
      "geometry": {
        "type": "Point",
        "coordinates": [
          13.369076,
          49.739747
        ]
      },
      "properties": {
        "id": "osm_n_296777015",
        "name": "Borská",
        "totalSpotCount": 8,
        "type": "BIKE_STAND",
        "addressFormatted": "Borská 1041/15, Jižní Předměstí, Plzeň"
      }
    }
  ]
}
```

Z důvodu ulehčení na FE jsme nově k bike parkingům přepočítali adresy. Ty by měly být nově součástí routingResponse, obdobně jako jsou u car parking. routingResponse bude tedy v patřičné části vypadat takto:

```
bicycleParkingDetails:{
  bicycleParking:{
    id:"osm_n_296777015",
    name:"Borská",
    type:"BIKE_STAND",
    geoCoordinate:{...},
    addressFormatted:"Borská 1041/15, Jižní Předměstí, Plzeň",
    totalSpotCount:8
  }
}
```

}

Poznámka: vychází z issue: <https://gitlab.com/operator-ict/ipt/umotional/-/issues/672>

Cenová nabídka

Název položky	Sazba / J bez DPH v Kč	Jednotka (J)	Počet J	Sazba celkem bez DPH v Kč
<i>Bod č. 1</i>	13.400,- Kč	MD	36,3	486.420,- Kč
<i>Bod č. 2</i>	13.400,- Kč	MD	1,9	25.460,- Kč
Celková cena v Kč bez DPH			33,4	511.880,- Kč

Celková cena je mezi organizátory hromadné dopravy rozdělena následovně:

Poměr	Cena
IDSK 32,15%	164 569,42 Kč
ROPID 67,85%	347 310,58 Kč
Cena celkem	511 880 Kč
Náklad pro IDSK činí	164 569,42 Kč bez DPH.

Platební podmínky

Cenová nabídka nezahrnuje úpravu související s případným umístěním aplikace v AppGallery.

Podmínky akceptace

Akceptace bude probíhat na základě článku 5 Smlouvy.

Doba plnění

Termín realizace požadavků dle zadání a dodání upravených aplikací k ověření a akceptaci je nejpozději do 16 týdnů od objednání.

Kontaktní a další informace

Tato nabídka má pouze informativní charakter a není závazným návrhem k uzavření smlouvy.

Operátor ICT, a.s.

www.operatorict.cz

IČO: 02795281

DIČ: CZ02795281

V Praze dne dle el. podpisu

Ing. Luboš Kratochvíl, MBA
předseda představenstva

