

REST API - Rezervace

Tento dokument popisuje, jak pomocí REST API KASA FIK vytvořit a spravovat **rezervace** (tabulka `reservations`) a **kapacitní sloty** (tabulka `reservation_capacity`).

1. Základní informace

Položka	Hodnota
Base URL	<code>https://m6vadtaz1h.execute-api.eu-west-1.amazonaws.com/prod/data/</code>
Staging URL	<code>https://m6vadtaz1h.execute-api.eu-west-1.amazonaws.com/staging/data/</code>
Endpoint pro rezervace	<code>POST /data/reservations</code> (vytvoření / aktualizace)
Endpoint pro kapacitní sloty	<code>POST /data/reservation_capacity</code>
Endpoint pro čtení	<code>GET /data/reservations</code> , <code>GET /data/reservation_capacity</code>
Autentizace	HTTP hlavička <code>`Authorization: A</code>
Content-Type	<code>application/json</code>
Formát ID	BigInt – v JSONu posílejte jako číslo nebo string , ale konzistentně v celé integraci
Formát času	Unix timestamp v milisekundách (např. <code>1735689600000</code>)

“[Generování nových ID](#) je popsáno v hlavní dokumentaci REST API (sekce *Funkce pro práci s id*). Pro vytvoření nového záznamu můžete vždy poslat `id = 0` a server přidělí nové ID sám.

1.1 Společná pravidla

- **Verze** (`_v`) – pokaždé, když záznam posíláte, musíte poslat **novější** `_v` (např. aktuální `Date.now()` v ms). Server záznam aktualizuje jen tehdy, pokud je příchozí `_v` vyšší než stávající. **Čas nesmí být v budoucnosti.**
- **Smazání** – nastavte `_d = 1` (místo mazání fyzického).
- **Viditelnost** – `visible = false` skryje záznam bez smazání.
- **Peníze** – všechny monetární hodnoty v JSONu jsou vynásobené 1000 (cena 23,50 Kč → `23500`).
- **Globální ID zákazníka** (`id_c`) – všechny objekty patří do tenanta identifikovaného `id_c`. Toto ID dostanete při registraci v Backoffice.
- **User-Agent** – vkládejte identifikaci své aplikace (název, verze, kontaktní e-mail), aby KASA FIK mohl v případě potřeby kontaktovat vývojáře.

- **Rate limit** – při chybě 429 snižte frekvenci požadavků.

2. Objekt `ReservationsModel` (rezervace)

2.1 Struktura JSON

Pole	Typ	Povinné	Výchozí	Popis
<code>id</code>	int64 (BigInt)	ano	-	ID záznamu. Pro nový záznam nastavte na <code>0</code> – server přidělí nové ID. Pro aktualizaci stávajícího zadejte jeho ID.
<code>_t</code>	string	ano	<code>"reservations"</code>	Identifikátor tabulky. Vždy odešlete <code>reservations</code>.
<code>_v</code>	int64	ano	-	Verze (Unix ms). Musí být vyšší než předchozí verze. Nesmí být v budoucnosti.
<code>_d</code>	int (0/1)	ano	<code>0</code>	Příznak smazání. <code>0</code> = aktivní, <code>1</code> = smazáno.
<code>id_c</code>	int64	ne	-	Globální ID zákazníka (tenanta). Doporučujeme vyplnit.
<code>id_shop</code>	int64	ano	<code>0</code>	ID pobočky. <code>0</code> = bez vazby na pobočku (pro všechny).
<code>id_customer</code>	int64	ne	<code>null</code>	ID zákazníka (z tabulky <code>customers</code>).
<code>id_product</code>	int64	ne	<code>null</code>	ID produktu (z tabulky <code>products</code>), ke kterému se rezervace vztahuje.
<code>id_park_location</code>	int64	ne	<code>null</code>	ID parkovacího místa (z tabulky <code>park_locations</code>), pokud rezervace zahrnuje parkování.

Pole	Typ	Povinné	Výchozí	Popis
id_employee_added	int64	ne	null	ID zaměstnance, který rezervaci vytvořil.
id_employee_accepted	int64	ne	null	ID zaměstnance, který rezervaci přijal.
id_reservation_capacity	int64	ne	null	ID kapacitního slotu (viz sekce 3), ke kterému rezervace patří.
date_start	int64	ne	null	Začátek rezervace (Unix ms).
date_end	int64	ne	null	Konec rezervace (Unix ms).
date_expires	int64	ne	null	Platnost rezervace do (Unix ms). Po tomto čase server rezervaci automaticky odmítne.
date_updated	int64	ne	-	Čas poslední aktualizace (Unix ms).
quantity_request	double	ne	null	Požadovaný počet (osob, kusů, míst...).
reservation_type	int (smallint)	ano	0	Typ rezervace (číselník - viz sekce 2.3).
reservation_bitmask	int (int32)	ano	0	Bitová maska příznaků rezervace (viz sekce 2.4).
accepted	bool	ano	false	true = rezervace byla přijata personálem.
visible	bool	ano	true	false = skryto, ale nesmazáno.
note_request	string	ne	null	Poznámka od zákazníka k rezervaci (max 128 znaků).
note_external	string	ne	null	Poznámka viditelná zákazníkem (max 128 znaků).
note_internal	string	ne	null	Interní poznámka personálu (max 128 znaků).

Pole	Typ	Povinné	Výchozí	Popis
tags	string	ne	null	Štítky (čárkou oddělený seznam, max 128 znaků).
url	string (URI)	ne	null	URL odkazující na detail rezervace (např. ve vašem systému).

2.2 Minimální payload pro vytvoření rezervace

```
{
  "id": 0,
  "_t": "reservations",
  "_v": 1735689600000,
  "_d": 0,
  "id_shop": 492696696397774,
  "id_customer": 3665795354036044,
  "id_product": 6555921856010421,
  "date_start": 1735689600000,
  "date_end": 1735693200000,
  "quantity_request": 2.0,
  "reservation_type": 0,
  "reservation_bitmask": 0,
  "accepted": false,
  "visible": true,
  "note_request": "Dvě osoby, sauna + whirlpool"
}
```

2.3 ?íselník reservation_type

Hodnota je malé celé číslo (smallint). Sada hodnot není v současné verzi plně formalizovaná, nicméně z praxe:

Hodnota	Význam
-1	RESERVATION_TYPE_NOT_SET – nedefinováno (interní konstanta, na API neposílejte)
0	výchozí / obecná rezervace
1	rezervace stolu / místa
2	rezervace slotu / časového okna
3	rezervace produktu (vstupenka, vstup, služba)

Hodnota	Význam
4+	další typy dle konfigurace tenanta

“ Konkrétní číselník pro vašeho tenanta si ověřte u KASA FIK podpory.

2.4 Bitová maska reservation_bitmask

Bitové pozice (lowest bit = 1) pro rezervace:

Bit	Hodnota	Význam
0	1	rezervace vyžaduje potvrzení
1	2	rezervace je opakující se (rekurzentní)
2	4	rezervace je blokována (drží místo)
3	8	rezervace s platbou předem
4	16	rezervace s upomínkou (SMS / e-mail)
5+	32+	další příznaky dle konfigurace tenanta

Příklady:

- 0 – běžná rezervace bez zvláštních příznaků
- 1 – vyžaduje potvrzení
- 3 – vyžaduje potvrzení a je opakující se
- 8 – s platbou předem
- 24 – 8 | 16 – s platbou předem a upomínkou

3. Objekt ReservationCapacityModel (kapacitní sloty)

Kapacitní slot definuje **opakovatelný blok** (např. "sauna pro 6 osob každý pátek 18:00 – 19:30"), ke kterému se pak vážou jednotlivé rezervace přes `id_reservation_capacity`.

3.1 Struktura JSON

Pole	Typ	Povinné	Výchozí	Popis
id	int64	ano	-	ID záznamu. 0 pro nový.

Pole	Typ	Povinné	Výchozí	Popis
<code>_t</code>	string	ano	<code>"reservation_capacity"</code>	Vždy <code>"reservation_capacity"</code> .
<code>_v</code>	int64	ano	–	Verze (Unix ms).
<code>_d</code>	int (0/1)	ano	<code>0</code>	Příznak smazání.
<code>id_c</code>	int64	ne	–	Globální ID tenanta.
<code>id_shop</code>	int64	ano	–	ID pobočky.
<code>id_cash_register</code>	int64	ano	–	ID pokladny, na které se slot spravuje.
<code>id_product</code>	int64	ne	<code>null</code>	ID produktu, se kterým slot souvisí.
<code>id_park_location</code>	int64	ne	<code>null</code>	ID parkovacího místa, pokud slot popisuje parkování.
<code>id_employee_req</code>	int64	ne	<code>null</code>	ID zaměstnance odpovědného za slot.
<code>name</code>	string	ne	<code>null</code>	Název slotu (např. "Sauna 18:00 pátek").
<code>date_start</code>	int64	ne	<code>null</code>	Začátek prvního výskytu slotu (Unix ms).
<code>date_end</code>	int64	ne	<code>null</code>	Konec prvního výskytu slotu (Unix ms).
<code>date_updated</code>	int64	ne	–	Čas poslední aktualizace.
<code>quantity_max</code>	double	ne	<code>null</code>	Maximální počet jednotek, které slot pojme (např. 6 osob).
<code>quantity_units</code>	double	ne	<code>null</code>	Počet jednotek na jednu rezervaci (např. 1 osoba).
<code>unit</code>	int (smallint)	ano	–	Jednotka (<code>0</code> = osoby, <code>1</code> = místa, <code>2</code> = kusy...).
<code>capacity_type</code>	int (smallint)	ano	<code>0</code>	Typ kapacity (číselník – viz 3.2).
<code>capacity_bitmask</code>	int (int32)	ano	<code>0</code>	Bitová maska příznaků slotu.

Pole	Typ	Povinné	Výchozí	Popis
dow_bitmask	int (int32)	ano	0	Bitová maska dnů v týdnu, kdy se slot opakuje (viz 3.3).
visible	bool	ano	true	Viditelnost.
note_external	string	ne	null	Poznámka pro zákazníka.
note_internal	string	ne	null	Interní poznámka.
tags	string	ne	null	Štítky.
internal_extra	string	ne	null	Interní dodatečná data (libovolný JSON string).

3.2 ?íselník capacity_type

Hodnota	Význam
0	obecný / výchozí slot
1	časový slot (opakovaný v čase)
2	místo / stůl (např. konkrétní sauna)
3	zdroj (např. whirlpool, lehátko)
4+	další typy dle konfigurace tenanta

3.3 Bitová maska dow_bitmask (den v týdnu)

Bit	Den
0	neděle
1	pondělí
2	úterý
3	středa
4	čtvrtek
5	pátek
6	sobota

Příklady:

- 0 – žádný den (jednorázový slot, řídí se jen date_start / date_end)
- 2 – pouze pondělí
- 96 – 32 | 64 – pouze pátek a sobota
- 127 – 1+2+4+8+16+32+64 – každý den

3.4 Příklad vytvoření slotu

```
{
  "id": 0,
  "_t": "reservation_capacity",
  "_v": 1735689600000,
  "_d": 0,
  "id_shop": 492696696397774,
  "id_cash_register": 49269669639111,
  "id_product": 6555921856010421,
  "name": "Sauna A – 18:00",
  "date_start": 1735689600000,
  "date_end": 1735693200000,
  "quantity_max": 6.0,
  "quantity_units": 1.0,
  "unit": 0,
  "capacity_type": 1,
  "capacity_bitmask": 0,
  "dow_bitmask": 32,
  "visible": true
}
```

4. Endpointy

4.1 Vytvoření nebo aktualizace jedné rezervace

POST /data/reservations

Tělo může být buď jeden objekt, nebo pole objektů pro dávkové vložení.

cURL - jedna rezervace:

```
curl -X POST 'https://m6vadtaz1h.execute-api.eu-west-1.amazonaws.com/prod/data/reservations' \
-H 'Authorization: A|xxxxxxxxxxxxxxxxxxxxxxxxxxxx' \
-H 'Content-Type: application/json' \
-H 'User-Agent: MojAplikace/1.0 (kontakt@firma.cz)' \
-d '{
  "id": 0,
  "_t": "reservations",
```

```
"_v": 1735689600000,
"_d": 0,
"id_shop": 492696696397774,
"id_customer": 3665795354036044,
"id_product": 6555921856010421,
"id_reservation_capacity": 9007199254740992,
"date_start": 1735689600000,
"date_end": 1735693200000,
"date_expires": 1735603200000,
"quantity_request": 2.0,
"reservation_type": 3,
"reservation_bitmask": 1,
"accepted": false,
"visible": true,
"note_request": "2 osoby, sauna + whirlpool"
}'
```

cURL - dávka (array):

```
curl -X POST 'https://m6vadtazlh.execute-api.eu-west-1.amazonaws.com/prod/data/reservations' \
-H 'Authorization: A|xxxxxxxxxxxxxxxxxxxxxxxxxxxx' \
-H 'Content-Type: application/json' \
-d '[
  { "id": 0, "_t": "reservations", "_v": 1735689600000, ... },
  { "id": 0, "_t": "reservations", "_v": 1735689601000, ... }
]'
```

Odpověď 200 OK:

Server vrací pole uložených objektů včetně serverem přiděleného `id`:

```
[
  {
    "id": 11154058202189370,
    "_t": "reservations",
    "_v": 1735689600000,
    "_d": 0,
    "id_shop": 492696696397774,
    "id_customer": 3665795354036044,
    "id_product": 6555921856010421,
    "date_start": 1735689600000,
```

```

    "date_end": 1735693200000,
    "quantity_request": 2.0,
    "reservation_type": 3,
    "reservation_bitmask": 1,
    "accepted": false,
    "visible": true,
    "note_request": "2 osoby, sauna + whirlpool",
    "date_updated": 1735689600050,
    "id_c": 111111
  }
]

```

4.2 Vytvoření nebo aktualizace kapacitního slotu

```
POST /data/reservation_capacity
```

Payload odpovídá struktuře v sekci 3.4.

4.3 Přístup k seznamu rezervací

```

GET /data/reservations
GET /data/reservations/{id}

```

S parametrem `id_start` můžete stránkovat (viz sekce 5).

4.4 Přístup k seznamu kapacitních slotů

```

GET /data/reservation_capacity
GET /data/reservation_capacity/{id}

```

4.5 Zrušení rezervace

Rezervace se **nemaze fyzicky** - nastavte `_d = 1`:

```

curl -X POST 'https://m6vadtaz1h.execute-api.eu-west-1.amazonaws.com/prod/data/reservations' \
-H 'Authorization: A|xxxxxxxxxxxxxxxxxxxxxxxxxxxx' \
-H 'Content-Type: application/json' \
-d '{
  "id": 11154058202189370,
  "_t": "reservations",
  "_v": 1735776000000,
  "_d": 1
}'

```

}'

“Poznámka: `reservation_bitmask` a `reservation_type` jsou v `_Fields` enumu definované jako `canBeNull = false` – při mazání je bezpečnější poslat je beze změny (tj. stejné jako u původního záznamu), aby server záznam skutečně přijal.

5. Stránkování

API vrací maximálně `limit` (1–400, výchozí 100) záznamů na stránku. Pro další stránku použijte `id_start` z posledního záznamu předchozí odpovědi.

Parametr	Typ	Výchozí	Popis
<code>limit</code>	int	<code>100</code>	Počet záznamů (1–400).
<code>id_start</code>	int64 (string)	<code>0</code>	ID posledního záznamu z předchozí strany.
<code>direction</code>	string	<code>backward</code>	<code>forward</code> (od nejstarších) / <code>backward</code> (od nejnovějších).
<code>version_start</code>	int64	<code>0</code>	Filtr na <code>_v >= version_start</code> (změnové stránkování).

Příklad sekvenčního čtení:

```
# 1. strana – 100 nejnovějších rezervací
curl 'https://m6vadtazlh.execute-api.eu-west-1.amazonaws.com/prod/data/reservations?limit=100' \
-H 'Authorization: A|xxxxxxxxxxxxxxxxxxxxxxxxxxxx'

# 2. strana – použijte id z posledního záznamu předchozí odpovědi
curl 'https://m6vadtazlh.execute-api.eu-west-1.amazonaws.com/prod/data/reservations?limit=100&id_start=11154058202189370' \
-H 'Authorization: A|xxxxxxxxxxxxxxxxxxxxxxxxxxxx'
```

“**Pozor na přesnost:** `id` je BigInt, může přesáhnout 53bit limit JavaScriptu. Při předávání do `id_start` ho **vždy posílejte jako string**, jinak hrozí ztráta přesnosti.

6. Kompletní příklad – vytvoření rezervace krok za krokem

Níže je ucelený příklad, jak vytvořit rezervaci pro zákazníka na konkrétní slot. Předpokládáme, že již máte:

- API token z Backoffice (A|...),
- ID pobočky (id_shop),
- ID produktu (id_product),
- ID zákazníka (id_customer) – případně nejprve vytvořte přes POST /data/customers,
- ID kapacitního slotu (id_reservation_capacity) – případně nejprve vytvořte přes POST /data/reservation_capacity.

```
# --- 1. Vytvoření rezervace ---
curl -X POST 'https://m6vadtaz1h.execute-api.eu-west-1.amazonaws.com/prod/data/reservations' \
-H 'Authorization: A|VAS_API_TOKEN' \
-H 'Content-Type: application/json' \
-H 'User-Agent: SaunofIntegrace/1.0 (dev@firma.cz)' \
-d '{
  "id": 0,
  "_t": "reservations",
  "_v": '$(date +%s%3N)',
  "_d": 0,
  "id_shop": 492696696397774,
  "id_customer": 3665795354036044,
  "id_product": 6555921856010421,
  "id_reservation_capacity": 9007199254740992,
  "date_start": 1735689600000,
  "date_end": 1735693200000,
  "date_expires": 1735603200000,
  "quantity_request": 2,
  "reservation_type": 3,
  "reservation_bitmask": 1,
  "accepted": false,
  "visible": true,
  "note_request": "2 osoby, prosím připravit ručníky",
  "tags": "online,wellness"
}'
# --- 2. Server odpoví 200 OK s uloženým objektem (včetně přiděleného id) ---
```

```
# --- 3. Později – potvrzení rezervace personálem ---
curl -X POST 'https://m6vadtaz1h.execute-api.eu-west-1.amazonaws.com/prod/data/reservations' \
-H 'Authorization: A|VAS_API_TOKEN' \
-H 'Content-Type: application/json' \
-d '{
  "id": 11154058202189370,
  "_t": "reservations",
  "_v": '$(date +%s%3N)',
  "_d": 0,
  "id_shop": 492696696397774,
  "id_customer": 3665795354036044,
  "id_product": 6555921856010421,
  "id_reservation_capacity": 9007199254740992,
  "date_start": 1735689600000,
  "date_end": 1735693200000,
  "date_expires": 1735603200000,
  "quantity_request": 2,
  "reservation_type": 3,
  "reservation_bitmask": 1,
  "accepted": true,
  "visible": true,
  "id_employee_accepted": 9876543210123456
}'

# --- 4. Zrušení rezervace (místo mazání) ---
curl -X POST 'https://m6vadtaz1h.execute-api.eu-west-1.amazonaws.com/prod/data/reservations' \
-H 'Authorization: A|VAS_API_TOKEN' \
-H 'Content-Type: application/json' \
-d '{
  "id": 11154058202189370,
  "_t": "reservations",
  "_v": '$(date +%s%3N)',
  "_d": 1
}'
```

7. Chybové stavy

Kód	Význam	Řešení
-----	--------	--------

200	OK - záznam vytvořen / aktualizován	pokračujte podle odpovědi
400	Neplatný <code>id_start</code> , chybějící povinné pole, apod.	zkontrolujte payload
403	Neplatný nebo chybějící API token	ověřte <code>Authorization: A</code>
404	Záznam s daným id neexistuje	ověřte <code>id</code> (případně přes <code>GET</code> seznamu)
429	Překročen rate limit	snižte frekvenci požadavků
500	Interní chyba serveru	opakujte požadavek po chvíli

8. Nejčastější chyby integrace

- Zapomenuté `_v`** - server záznam odmítne. Vždy posílejte `Date.now()` (nebo jiné aktuální `now()` v ms).
- Budoucí `_v`** - timestamp `_v` nesmí být v budoucnosti. Server takové záznamy odmítá.
- `id = 0` pro update** - pokud chcete aktualizovat existující záznam, musíte poslat jeho skutečné `id`. `id = 0` vždy vytvoří nový.
- Smazání nastavením `visible = false`** - to záznam pouze skryje, ale nesmaže. Pro smazání použijte `_d = 1`.
- BigInt v JavaScriptu** - `id` přesahuje 53 bitů. Vždy ho posílejte jako string v URL (`id_start=...`) a parsujte s `BigInt` na straně klienta.
- Špatné ID pobočky** - `id_shop` musí existovat v tabulce `shops`. Nejdříve získejte seznam poboček přes `GET /data/shops`.
- Chybějící záznam zákazníka** - `id_customer` musí existovat v tabulce `customers`. Nejdříve vytvořte zákazníka přes `POST /data/customers` (viz dokumentace REST API).

9. Související endpointy

Pro úplnou integraci budete pravděpodobně potřebovat i tyto endpointy (podrobnosti v hlavní dokumentaci REST API):

- `GET /data/shops` - seznam poboček
- `GET /data/cash_registers` - seznam pokladen
- `GET /data/customers` / `POST /data/customers` - zákazníci
- `GET /data/products` - produkty (vstupenky, služby)
- `GET /data/park_locations` - parkovací místa

10. Kontakt a podpora

- E-mail podpory KASA FIK: podpora@kasafik.cz

- Dokumentace REST API: <https://manual.kasafik.cz/books/kasa-fik-pokladna/page/rest-api>
- Webhooky (doporučený způsob synchronizace): <https://manual.kasafik.cz/books/kasa-fik-pokladna/page/webhook-integrace-api>

11. Wellness / rezervační systém – kompletní příklad

Tento příklad ukazuje typický tok z pohledu wellness aplikace, která posílá do KASA FIK rezervaci sauny pro 2 osoby v konkrétním časovém okně na konkrétním místě.

Předpoklad: v KASA FIK již existuje produkt "Sauna 60 minut" (`id_product`) a záznam zákazníka (`id_customer`).

11.1 Krok 1 – ujistěte se, že existuje kapacitní slot

Pokud máte v KASA FIK definované opakující se sloty (např. "Sauna A, 18:00-19:00, každý pátek"), použijte jejich `id` v kroku 2. Pokud ne, vytvořte slot jednorázově:

```
curl -X POST 'https://m6vadtaz1h.execute-api.eu-west-1.amazonaws.com/prod/data/reservation_capacity' \
-H 'Authorization: A|VAS_API_TOKEN' \
-H 'Content-Type: application/json' \
-H 'User-Agent: WellnessApp/1.0 (dev@wellness.cz)' \
-d '{
  "id": 0,
  "_t": "reservation_capacity",
  "_v": 1735689000000,
  "_d": 0,
  "id_shop": 492696696397774,
  "id_cash_register": 49269669639111,
  "id_product": 6555921856010421,
  "name": "Sauna A – 18:00",
  "date_start": 1735689600000,
  "date_end": 1735693200000,
  "quantity_max": 6,
  "quantity_units": 1,
  "unit": 0,
  "capacity_type": 1,
  "capacity_bitmask": 0,
```

```
"dow_bitmask": 32,  
"visible": true  
}'
```

“ Všimněte si, že `id_product` je vyplněn i na slotu – díky tomu nemusíte při vytváření každé rezervace uvádět produkt znovu, stačí `id_reservation_capacity`. Viz 11.2 B).

Server odpoví pole s přiděleným `id` slotu, např. `9007199254740992`.

11.2 Krok 2 – vytvořte rezervaci

A) Kompletní varianta – doporučeno

V této variantě uvádíte **všechny tři vazby**: co, kdy/kde, kolik. Je to nejpřehlednější a nejméně chybové.

```
curl -X POST 'https://m6vadtaz1h.execute-api.eu-west-1.amazonaws.com/prod/data/reservations' \  
-H 'Authorization: A|VAS_API_TOKEN' \  
-H 'Content-Type: application/json' \  
-H 'User-Agent: WellnessApp/1.0 (dev@wellness.cz)' \  
-d '{  
  "id": 0,  
  "_t": "reservations",  
  "_v": 1735689600000,  
  "_d": 0,  
  "id_shop": 492696696397774,  
  "id_customer": 3665795354036044,  
  "id_product": 6555921856010421,  
  "id_reservation_capacity": 9007199254740992,  
  "date_start": 1735689600000,  
  "date_end": 1735693200000,  
  "quantity_request": 2,  
  "reservation_type": 3,  
  "reservation_bitmask": 0,  
  "accepted": false,  
  "visible": true,  
  "note_request": "Výročí – prosím tiché prostředí",  
  "url": "https://wellness.example.com/bookings/42",  
  "tags": "wellness,sauna,online"
```

```
}'
```

Co zde jednotlivá pole znamenají v kontextu wellness:

Pole	V tomto příkladu
<code>id_product</code>	Produkt "Sauna 60 minut" z vašeho ceníku – toto se bude účtovat
<code>id_reservation_capacity</code>	Konkrétní slot "Sauna A, 18:00–19:00, 6 míst"
<code>date_start</code> / <code>date_end</code>	Začátek a konec konkrétní návštěvy (zde 18:00–19:00)
<code>quantity_request</code>	Počet osob (2)
<code>url</code>	Odkaz zpět do vašeho wellness systému na detail této rezervace
<code>note_request</code>	Poznámka od zákazníka (zobrazí se personálu)

B) Stručná varianta – když má slot vyplněný produkt

Pokud jste v kroku 1 vyplnili `id_product` na slotu, stačí poslat jen `id_reservation_capacity`. KASA FIK si produkt sám doplní ze slotu.

```
curl -X POST 'https://m6vadtazlh.execute-api.eu-west-1.amazonaws.com/prod/data/reservations' \
-H 'Authorization: A|VAS_API_TOKEN' \
-H 'Content-Type: application/json' \
-H 'User-Agent: WellnessApp/1.0 (dev@wellness.cz)' \
-d '{
  "id": 0,
  "_t": "reservations",
  "_v": 1735689600000,
  "_d": 0,
  "id_shop": 492696696397774,
  "id_customer": 3665795354036044,
  "id_reservation_capacity": 9007199254740992,
  "date_start": 1735689600000,
  "date_end": 1735693200000,
  "quantity_request": 2,
  "reservation_type": 3,
  "reservation_bitmask": 0,
  "accepted": false,
  "visible": true
}'
```

C) Minimální varianta – bez slotu

Pokud vám stačí **časová vazba** (rezervace na určitý časový úsek, ale bez konkrétního místa/zdroje), můžete `id_reservation_capacity` vynechat. Produkt a časové okno ale musíte vyplnit.

```
curl -X POST 'https://m6vadtaz1h.execute-api.eu-west-1.amazonaws.com/prod/data/reservations' \
-H 'Authorization: A|VAS_API_TOKEN' \
-H 'Content-Type: application/json' \
-H 'User-Agent: WellnessApp/1.0 (dev@wellness.cz)' \
-d '{
  "id": 0,
  "_t": "reservations",
  "_v": 1735689600000,
  "_d": 0,
  "id_shop": 492696696397774,
  "id_customer": 3665795354036044,
  "id_product": 6555921856010421,
  "date_start": 1735689600000,
  "date_end": 1735693200000,
  "quantity_request": 2,
  "reservation_type": 3,
  "reservation_bitmask": 0,
  "accepted": false,
  "visible": true
}'
```

11.3 Krok 3 – server odpoví

Server vrací pole s uloženým objektem včetně přiděleného `id` rezervace (např. `11154058202189370`). Toto `id` si uložte – budete ho potřebovat pro pozdější aktualizace a zrušení.

11.4 Krok 4 – personál potvrdí rezervaci

Při příchodu zákazníka personál na pokladně změní `accepted` na `true`. Vy (wellness systém) o tom můžete vědět díky webhookům (viz hlavní dokumentace).

11.5 Krok 5 – zrušení rezervace

Rezervace se **nemaze fyzicky** – nastavte `_d = 1`:

```
curl -X POST 'https://m6vadtaz1h.execute-api.eu-west-1.amazonaws.com/prod/data/reservations' \
-H 'Authorization: A|VAS_API_TOKEN' \
-H 'Content-Type: application/json' \
-d '{
```

```
"id": 11154058202189370,  
"_t": "reservations",  
"_v": 1735776000000,  
"_d": 1  
'
```

11.6 Nejčastější chyby v tomto scénáři

1. **Produkt neexistuje v KASA FIK** – ujistěte se, že `id_product` odpovídá produktu v ceníku pokladny. Pokud váš wellness systém používá vlastní ID, musíte si je předem namapovat na ID v KASA FIK. Koordinaci ID řešte s provozovatelem pokladny.
2. **Zákazník neexistuje** – před vytvořením rezervace musí zákazník existovat v tabulce `customers` (viz `POST /data/customers` v hlavní dokumentaci).
3. **Slot nemá kapacitu** – pokud `quantity_request` překročí zbývajících `quantity_max` slotu, server může rezervaci odmítnout. Kapacitu slotu je třeba hlídat na straně wellness systému.
4. `reservation_type` **musíte zvolit správně** – typ `3` je "rezervace produktu (vstupenka, vstup, služba)", což odpovídá wellness. Jiné typy konzultujte s podporou.
5. **Vždy posílejte aktuální `_v`** – pro update použijte nové `Date.now()` v ms, starší `_v` server odmítne.

Revision #6

Created 2026-09-01 06:41:01 UTC by Admin

Updated 2026-10-06 04:22:05 UTC by Admin