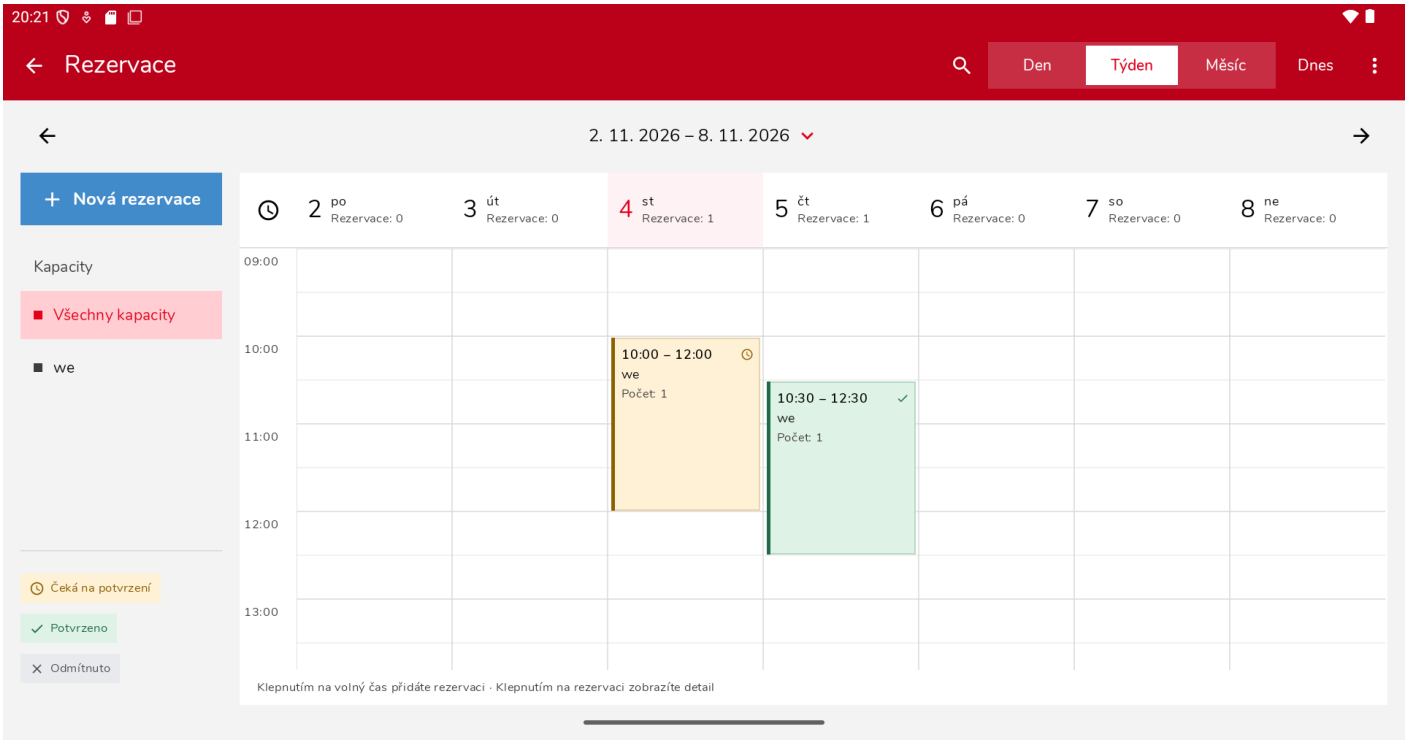
Rezervace

- Rezervační kniha (KASA FIK Pokladna)
- REST API - Rezervace

Rezerva?ní kniha (KASA FIK Pokladna)

Co je rezerva?ní kniha

Rezervační kniha je **kalendář, do kterého zapisujete obsazení stolu, místnosti nebo služby**. Slouží k tomu, aby se na stejný stůl (nebo místnost) nepřekrývaly dvě rezervace s větším počtem osob, než kapacita snese.



Do kalendáře se dostanete z **domovské obrazovky** — dlaždice **„Rezervační kniha"** (ikona kalendáře). Obrazovka se jmenuje **„Rezervace"**.

> **Nejdřív kapacity, pak rezervace.** Dokud nezaložíte žádnou kapacitu, kalendář je prázdný a ukáže výzvu „Nastavte rezervační kalendář". Bez kapacity není možné vytvořit rezervaci.

Základní pojmy

Kapacita

Kapacita je jedno reservovatelné místo — stůl, místnost nebo služba. Každá kapacita má vlastní kalendář, vlastní maximální kapacitu a vlastní dostupné dny.

Pole	Význam
Název	Jak se kapacita jmenuje v kalendáři (např. „Stůl 5", „Místnost Terasa"). Volitelné? **Ne — název je povinný.**
Počet osob na slot	Deklarovaná velikost slotu — kolik osob je slot určený pro. Zobrazuje se v kalendáři jako „x3".
Maximální kapacita	**Tvrdý strop:** kolik osob smí být na této kapacitě současně obsazeno (součet všech rezervací, které se v čase překrývají).
Dostupnost (dny v týdnu)	V kterých dnech je kapacita k dispozici. Pondělí je první den v řadě.
Datum začátku / Datum konce	Volitelný interval platnosti kapacity. **Prázdné = bez omezení data.**

Rezervace

Rezervace je konkrétní obsazení kapacity v konkrétním čase.

Pole	Význam
Kapacita	Na kterou kapacitu rezervace patří.
Začátek	Datum a čas začátku.
Konec	Datum a čas konce. Musí být **po** začátku.
Délka	Rychlá volba konce (30 min / 60 min / 90 min / 2 hodiny / 1 den).
Počet	Kolik osob přijde. Musí být **větší než 0**.
Poznámka (viditelná zákazníkovi)	Poznámka pro hosta.
Interní poznámka	Poznámka pouze pro personál.

„Po?et osob na slot" vs. „Maximální kapacita"

Toto jsou **dvě různé věci** a snadno se zamění:

	„Počet osob na slot"	**„Maximální kapacita"**
Co to je	Velikost **jednoho** slotu (stolu).	Strop **celkově** na kapacitě.
Co se kontroluje	Jen to, že je **1 až 32767**.	Sčítá se s počtem osob v překrývajících se rezervacích.
Co se stane, když je překročený	Nic — hodnota je pouze informativní.	Aplikace **upozorní** při uložení.

Pravidlo při zakládání kapacity: Maximální kapacita musí být **alespoň rovna** Počtu osob na slot. Aplikace to při uložení vyžaduje.

Důležité: Aplikace **neumí** uložit minimální počet osob. Rezervace na kapacitu „Stůl 5" (5 osob na slot, max 5) tedy může klidně obsahovat jedinou osobu — minimum je dáno tím, že Počet musí být větší než 0. Pokud potřebujete rezervovat stůl **výhradně** pro skupiny od 3 osob, tuto podmínku je nutné hlídat ručně.

Příklad — stůl pro 1 až 3 osoby, maximálně 3:

- Počet osob na slot = **3** (stůl je určený pro tři osoby)
- Maximální kapacita = **3** (víc než tři osoby u stolu nebude)

Pak platí: rezervace na 1, 2 i 3 osoby se uloží; rezervace na 4 osoby se uloží jen po výslovném potvrzení; dvě překrývající se rezervace po 2 osobách už překročí strop a aplikace na to upozorní.

> **Pozor:** Hodnota „Maximální kapacita = 0" znamená **bez omezení** — aplikace pak nikdy nevypíše upozornění na překročení. Přes formulář ale kapacitu s maximem 0 vytvořit nejde, protože aplikace vyžaduje alespoň Počet osob na slot.

Ovládání kalendá?e

Režimy zobrazení

V horní liště (na širokých displejích) nebo pod obdobím (na telefonech) přepínáte mezi třemi režimy:

Režim	Co ukazuje
Den	Časová osa jednoho dne od 00:00 do 24:00. Na širokém displeji jeden sloupec na každou kapacitu.
Týden	Sedm dnů vedle sebe; na širokém displeji řádek na každou kapacitu.
Měsíc	Měsíční mřížka; vpravo (na širokém displeji) nebo pod ní (na telefonu) agenda vybraného dne.

Období

- **Šipky vlevo / vpravo** — předchozí / následující den, týden nebo měsíc.
- **Klepnutí na nadpis období** (např. „úterý 4. listopadu 2026") otevře **výběr data**.
- **Tlačítko „Dnes"** v horní liště vrátí na dnešek.

Filtrování podle kapacity

- Na telefonu a tabletu je nahoře řada **filtrů** (štěptek): „Všechny kapacity" a pak jedna pro každou kapacitu.
- Na širokém displeji je kapacity v **postranním panelu** vlevo; dole je legenda stavů.

Vyhledávání

Klepněte na **ikonu lupy** v horní liště. Hledá se podle:

- jména hosta v poznámkách,
- obou poznámek (i interní),
- **názvu kapacity**,
- data rezervace (i ve tvaru `2026-11-04` nebo `4.11`).

Výsledky se zobrazí v řadě pod lištou. Klepnutím na výsledek se kalendář přesune na daný den a otevře detail. Hledání prochází **rok do obou stran** od zobrazeného data, takže najdete i rezervaci mimo viditelný týden.

Nabídka ⋮

Vpravo nahoře (tři tečky) najdete:

Položka	Co udělá
Správa kapacit	Seznam kapacit — přidat, upravit, smazat.
Nastavení kalendáře	Výchozí délka rezervace a druh správy rezervací.
Obnovit kalendář	Ruční načtení ze serveru.

Správa kapacit

Založení nové kapacity

1. Otevřete ⋮ → **Správa kapacit**.
2. Klepněte na **„Přidat kapacitu"**.
3. Vyplňte:
   - **Název** — např. „Stůl 5". *Povinné.*
   - **Počet osob na slot** — např. `3`.
   - **Maximální kapacita** — např. `3`. *Musí být alespoň Počet osob na slot.*
4. V sekci **Dostupnost** klepněte na dny v týdnu, kdy je kapacita otevřená. Vybrat lze libovolnou kombinaci, ale **alespoň jeden den**.
5. Volitelně nastavte **Datum začátku** a **Datum konce** (formát `dd.MM.yyyy`). Když necháte pole prázdné, je pole **„Bez omezení data"** a kapacita platí trvale.
6. Klepněte na **„Uložit"**.

Úprava kapacity

V seznamu kapacit klepněte na **„Upravit"** u dané kapacity. Formulář se otevře s vyplněnými hodnotami. Po úpravě klepněte na **„Uložit"**.

> Přejmenování kapacity funguje a hned se propíše i do kalendáře — stačí změnit pole **Název** a uložit.

Smazání kapacity

V seznamu kapacit klepněte na **ikonu koše** u dané kapacity a potvrďte **„Smazat"**.

Smazat lze pouze prázdnou kapacitu. Pokud na kapacitě existuje alespoň jedna rezervace, aplikace smazání odmítne a zobrazí: *„Kapacita obsahuje rezervace. Před smazáním je přesuňte nebo odstraňte."* Existující rezervace se **nikdy** neodstraní automaticky.

Práce s rezervacemi

Vytvo?ení rezervace

Způsob A — klepnutím do kalendáře:

1. Klepněte na volný čas v denním režimu. Čas se zarovná na **čtvrt hodiny** (00:00, 00:15, 00:30, …).
2. Otevře se okno **„Nová rezervace"** předvyplněné kapacitou a časem.

Způsob B — tlačítkem:

1. Klepněte na modré tlačítko **„Nová"** (na telefonu plovoucí tlačítko **+**).
2. Nová rezervace se předvyplní:
   - **na dnešek** — na nejbližší čtvrt hodiny, která ještě nenastala,
   - **na jiný den** — na **09:00**.
3. V okně **„Nová rezervace"** zkontrolujte nebo upravte:
   - **Kapacita** — z rozbalovacího menu; u každé kapacity je uvedeno její maximum.
   - **Začátek** a **Konec** — datum i čas (datum i čas se otevírají samostatnými ovládacími prvky).
   - **Délka** — rychlé chipsy **30 min | 60 min | 90 min | 2 hodiny | 1 den** nastaví konec vzhledem k začátku.
   - **Počet** — počet osob.
   - **Poznámka (viditelná zákazníkovi)** a **Interní poznámka**.
4. Klepněte na **„Uložit"**.

Když aplikace odmítne uložit

Hlášení	Příčina	Řešení
„Vyberte kapacitu"	Není zvolena žádná kapacita.	Zvolte kapacitu v rozbalovacím menu.
„Počet musí být větší než 0"	Pole Počet je prázdné nebo nula.	Zadejte počet osob.
„Konec musí být po začátku"	Konec není po začátku.	Změňte konec.
„Tato kapacita není ve vybraném termínu dostupná"	Kapacita není dostupná v daný den (dnem v týdnu nebo mimo interval platnosti).	Změňte den nebo kapacitu.

P?ekro?ení kapacity

Před uložením se aplikace nejdřív zeptá back office, jestli v daném čase nevznikla rezervace na jiné pokladně. Teprve pak porovná součet osob ve všech překrývajících se rezervacích (na téže kapacitě) s **Maximální kapacitou**.

Když součet kapacitu přesáhne, objeví se dialog:

> **Dostupná kapacita je překročena** — V tomto čase se překrývají jiné rezervace. Vyberte jiný čas nebo kapacitu, nebo překročení kapacity výslovně potvrďte.

Máte dvě možnosti:

- **„Přesto uložit"** — rezervace se uloží i přes překročení (například když víte, že hosté nakonec nepřijdou).
- **„Pokračovat v úpravách"** — vrátíte se do formuláře a čas nebo kapacitu upravíte.

> **Bez připojení k internetu** se rezervace uloží bez čekání na back office a kontrola proběhne pouze podle dat v zařízení. Rezervace vytvořená mezitím na jiné pokladně se tak nemusí odhalit — po obnovení spojení se data sloučí.

Odmítnuté rezervace (viz níže) **kapacitu neobsazují**.

Stav rezervace

Stav	Vzhled	Význam
Čeká na potvrzení	oranžový	Nově vytvořená rezervace, ještě nepotvrzená.
Potvrzeno	zelený	Přijatá rezervace.

Stav	Vzhled	Význam
Odmítnuto	červený	Odmítnutá rezervace. **Neobsazuje kapacitu.**

Detail rezervace a akce

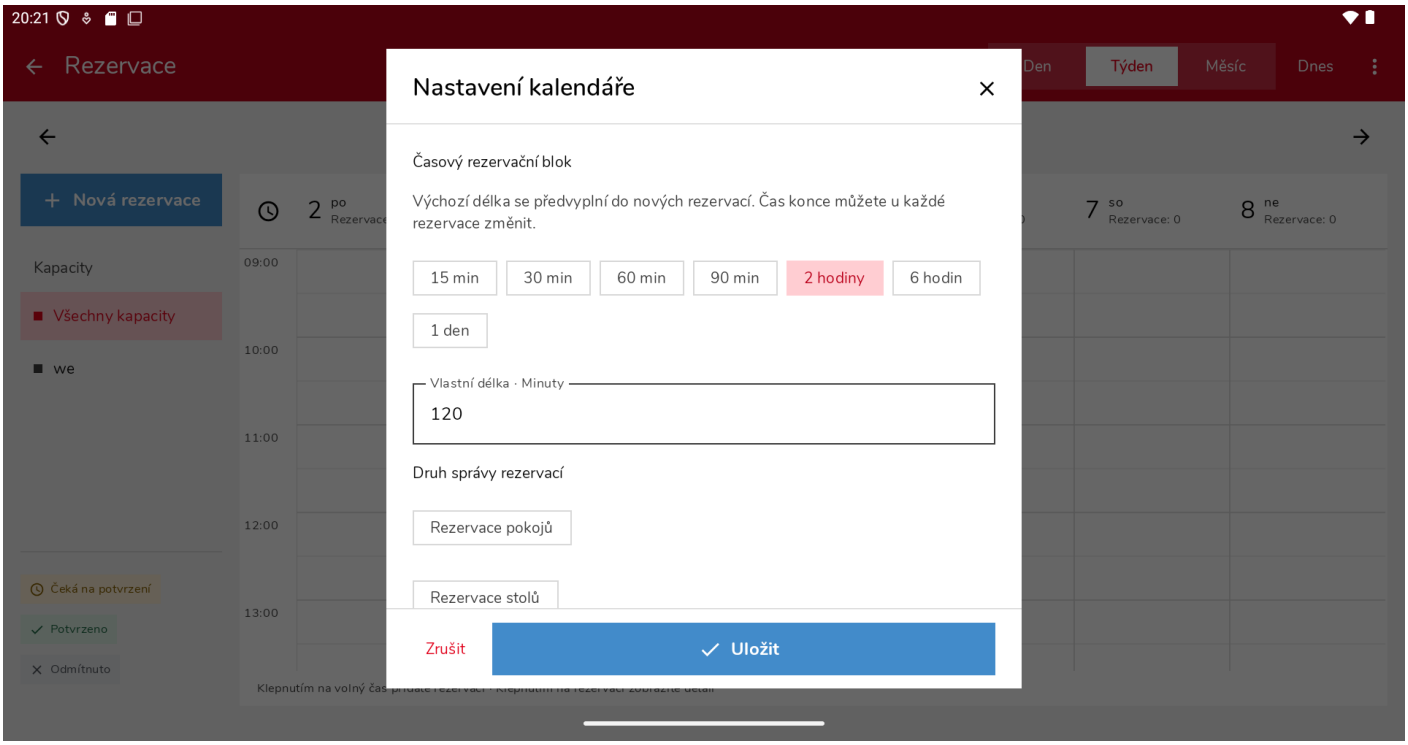
Klepněte na rezervaci v kalendáři. Otevře se panel (na širokém displeji vpravo, jinak jako spodní panel) s **detailem rezervace** a tlačítky:

Akce	Co udělá
Přijmout	Nastaví stav „Potvrzeno". Nabízí se jen u stavu jiném než Potvrzeno.
Odmítnout	Nastaví stav „Odmítnuto" — rezervace přestane obsazovat kapacitu.
Upravit	Otevře formulář rezervace k úpravám.
Přesunout / změnit termín	Otevře stejný formulář — změňte Začátek a Konec.
Smazat	Po potvrzení smaže rezervaci z kalendáře.

Do detailu se zobrazuje i upozornění **„Překročena kapacita"**, pokud rezervace v daném okamžiku přesahuje maximum.

Nastavení kalendá?e

Otevřete ⋮ → **Nastavení kalendáře**.



?asový rezerva?ní blok (výchozí délka)

Výchozí délka se **předvyplní do nových rezervací**. Konkrétní rezervaci můžete kdykoli ukončit jinak (pole Konec nebo chipsy Délka).

Zvolená délka	Uloží se jako
15 minut	`15m`
30 minut	`30m`
60 minut	`60m`
90 minut	`90m`
2 hodiny	`2h`
6 hodin	`6h`
1 den	`1d`
Vlastní délka	libovolný počet minut (1 – 143 998 560)

Výchozí délka je **globální** — platí pro všechny kapacity.

Druh správy rezervací

Určuje, o jaký typ rezervací jde (slouží jako štítek uložený u kapacity):

Volba	Význam
Rezervace pokojů	Pokoje / ubytování.
Rezervace stolů	Stoly v restauraci.
Rezervace služeb	Služby (např. kosmetika, servis).

> Nastavení je **součástí rezervační knihy** — najdete ho jen přes ⋮ → **Nastavení kalendáře**. V Nastavení aplikace sekce rezervací není.

Synchronizace s back officem

Kapacity i rezervace se **synchronizují automaticky** — nemusíte nic potvrzovat ani hlídat.

Co se synchronizuje

Směr	Kdy	Co

Odeslání	Ihned po uložení změny a při pravidelné synchronizaci.	Nové a upravené kapacity a rezervace se nahrají do back office.
Načtení	Při otevření obrazovky Rezervace (na pozadí) a při pravidelné synchronizaci.	Kapacity a rezervace vytvořené jinde se zobrazí zde.

Která verze vyhrává

- **Neposlaná místní změna vždy vyhrává** nad serverovou kopií. Pokud kapacitu právě měníte a v back office se mezitím změní, vyhrává vaše verze a po následném odeslání se propíše i tam.
- Novější serverová kopie přepíše starší místní kopii.

Kdy se zm?na propíše

Změna se odesílá **na pozadí** — kalendář zůstává použitelný. Pokud se připojení nezdaří, změna zůstane uložená v zařízení a odešle se při nejbližší synchronizaci.

Omezení

- V **režimu SQL klienta** se rezervace na server **neodesílají** (kapacity se odesílají). Tento režim je určen pro pobočky, které pracují jen s databází.
- Rezervace, jejíž kapacita na zařízení zatím není, se zobrazí jako **„Nepřiřazená kapacita"** . Jakmile se kapacita dohraje, zobrazí se správně.

Omezení a ?asté dotazy

Omezení

- **Rezervace musí patřit kapacitě.** Rezervace bez kapacity nelze vytvořit.
- **Počet musí být větší než 0.** Záporné hodnoty ani nula nejsou přípustné.
- **Maximální kapacita musí být ≥ Počtu osob na slot**, jinak kapacitu nelze uložit.
- **Kapacitu nelze smazat, pokud na ní jsou rezervace.**
- **Minimální počet osob nelze nastavit.** Rezervace na 1 osobu je možná i na kapacitě s maximem 3.
- **Časovou dostupnost nelze nastavit na samotné kapacitě.** Termín 14:00–16:00 není vlastností kapacity, ale jednotlivé rezervace. Dostupné dny a interval platnosti kapacity ano.
- **Datum vypršení rezervace se nevyhodnocuje** — aplikace toto pole zatím nepoužívá.

?asté dotazy

Q: Kde najdu nastavení kalendáře? A: Přímo v rezervační knize: ⋮ → **Nastavení kalendáře**. V Nastavení aplikace sekce rezervací není.

Q: Jak smažu kapacitu, na které jsou rezervace? A: Nelze. Nejprve přesuňte nebo smažte všechny její rezervace, pak kapacitu smažte. Aplikace existující rezervace nikdy neodstraní automaticky.

Q: Co znamená „x3" u kapacity v kalendáři? A: Počet osob na slot — deklarovaná velikost slotu. **Nejde o limit.** Skutečný limit je Maximální kapacita.

Q: Rezervace se uložila, ale kalendář hlásí „Překročena kapacita". A: Součet osob v překrývajících se rezervacích na téže kapacitě je vyšší než maximum. Překročení lze potvrdit tlačítkem „Přesto uložit", nebo upravit čas či kapacitu. Samotné potvrzení nevytvoří nový limit.

Q: Proč se mi rezervace neobjeví v kalendáři? A: Zkontrolujte filtr kapacit — je zapnutá jiná kapacita než ta, na které je rezervace. Dále zkontrolujte stav „Odmítnuto" (odmítnuté rezervace se v kalendáři stále zobrazují, ale neobsazují kapacitu) a případně i datum — rezervace mimo zobrazené období se zobrazí jen po posunutí období nebo přes hledání.

Q: Změnil jsem kapacitu, ale změna se nevidí na jiné pokladně. A: Změna se odesílá na pozadí. Zkontrolujte připojení k internetu a zkuste ⋮ → **Obnovit kalendář** na druhé pokladně.

Q: Lze mít jednu kapacitu pro tři stoly po dvou osobách? A: Ano — vytvořte kapacitu s Počtem osob na slot = 2 a Maximální kapacitou = 6. Pak lze současně obsadit tři stoly po dvou osobách (6 osob celkem). Rezervace se ale nepoznají podle stolu — patří všechny jedné kapacitě.

Q: Lze rezervaci vytvořit na 4 osoby, když maximum je 3? A: Ano, ale aplikace při uložení vyžádá výslovné potvrzení („Přesto uložit"). Bez potvrzení uložení nepokračuje.

Q: Co znamená prázdné Datum začátku / Datum konce u kapacity? A: „Bez omezení data" — kapacita platí od založení napočud.

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

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

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
`_t`	string	**ano**	`"reservation_capacity"`	Vždy `"reservation_capacity"`.
`_v`	int64	**ano**	–	Verze (Unix ms).
`_d`	int (0/1)	**ano**	`0`	Příznak smazání.
`id_c`	int64	ne	–	Globální ID tenanta.
`id_shop`	int64	**ano**	–	ID pobočky.
`id_cash_register`	int64	**ano**	–	ID pokladny, na které se slot spravuje.
`id_product`	int64	ne	`null`	ID produktu, se kterým slot souvisí.
`id_park_location`	int64	ne	`null`	ID parkovacího místa, pokud slot popisuje parkování.
`id_employee_req`	int64	ne	`null`	ID zaměstnance odpovědného za slot.
`name`	string	ne	`null`	Název slotu (např. "Sauna 18:00 pátek").
`date_start`	int64	ne	`null`	Začátek prvního výskytu slotu (Unix ms).
`date_end`	int64	ne	`null`	Konec prvního výskytu slotu (Unix ms).
`date_updated`	int64	ne	–	Čas poslední aktualizace.
`quantity_max`	double	ne	`null`	Maximální počet jednotek, které slot pojme (např. 6 osob).
`quantity_units`	double	ne	`null`	Počet jednotek na jednu rezervaci (např. 1 osoba).
`unit`	int (smallint)	**ano**	–	Jednotka (`0` = osoby, `1` = místa, `2` = kusy…).
`capacity_type`	int (smallint)	**ano**	`0`	Typ kapacity (číselník – viz 3.2).
`capacity_bitmask`	int (int32)	**ano**	`0`	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. whirpool, 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 P?íklad vytvo?ení 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 Vytvo?ení 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|xxxxxxxxxxxxxxxxxxxxxxxx' \
  -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 + whirpool"
  }'
```

cURL – dávka (array):

```
curl -X POST 'https://m6vadtaz1h.execute-api.eu-west-1.amazonaws.com/prod/data/reservations' \
  -H 'Authorization: A|xxxxxxxxxxxxxxxxxxxxxxxx' \
  -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 + whirpool",
    "date_updated": 1735689600050,
    "id_c": 111111
  }
]
```

4.2 Vytvo?ení nebo aktualizace kapacitního slotu

```
POST /data/reservation_capacity
```

Payload odpovídá struktuře v sekci 3.4.

4.3 ?tení seznamu rezervací

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

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

4.4 ?tení seznamu kapacitních slot?

```
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|xxxxxxxxxxxxxxxxxxxxxxxx' \
  -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
`limit`	int	`100`	Počet záznamů (1–400).
`id_start`	int64 (string)	`0`	ID posledního záznamu z předchozí strany.
`direction`	string	`backward`	`forward` (od nejstarších) / `backward` (od nejnovějších).
`version_start`	int64	`0`	Filtr na `_v >= version_start` (změnové stránkování).

Příklad sekvenčního čtení:

```
# 1. strana – 100 nejnovějších rezervací
curl 'https://m6vadtaz1h.execute-api.eu-west-1.amazonaws.com/prod/data/reservations?limit=100'
\
  -H 'Authorization: A|xxxxxxxxxxxxxxxxxxxxxxxx'

# 2. strana – použijte id z posledního záznamu předchozí odpovědi
curl 'https://m6vadtaz1h.execute-api.eu-west-
1.amazonaws.com/prod/data/reservations?limit=100&id_start=11154058202189370' \
  -H 'Authorization: A|xxxxxxxxxxxxxxxxxxxxxxxx'
```

> **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 p?íklad – vytvo?ení 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 `id_start`, chybějící povinné pole, apod.	zkontrolujte payload
`403`	Neplatný nebo chybějící API token	ověřte `Authorization: A
`404`	Záznam s daným id neexistuje	ověřte `id` (případně přes `GET` seznamu)
`429`	Překročen rate limit	snižte frekvenci požadavků
`500`	Interní chyba serveru	opakujte požadavek po chvíli

---

8. Nej?ast?jší chyby integrace

1. **Zapomenuté** `_v` – server záznam odmítne. Vždy posílejte `Date.now()` (nebo jiné aktuální `now()` v ms).
2. **Budoucí** `_v` – timestamp `_v` nesmí být v budoucnosti. Server takové záznamy odmítá.
3. `id = 0` **pro update** – pokud chcete aktualizovat existující záznam, musíte poslat jeho skutečné `id`. `id = 0` vždy vytvoří nový.
4. **Smazání nastavením** `visible = false` – to záznam pouze skryje, ale nesmaže. Pro smazání použijte `_d = 1`.
5. **BigInt v JavaScriptu** – `id` přesahuje 53 bitů. Vždy ho posílejte jako string v URL (`id_start=…`) a parsujte s `BigInt` na straně klienta.
6. **Špatné ID pobočky** – `id_shop` musí existovat v tabulce `shops`. Nejdříve získejte seznam poboček přes `GET /data/shops`.
7. **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 / rezerva?ní systém – kompletní p ?íklad

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 – ujist?te 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 – vytvo?te rezervaci

A) Kompletní varianta – doporu?eno

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

B) Stru?ná varianta – když má slot vypln?ný 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://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_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 Nej?ast?jší chyby v tomto scéná?i

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 zbývající `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.