

API SPRÁVA ALERTŮ

Dokumentace

ver. 7.0

(Červenec 2023)

OBSAH

1. ÚVOD	3
1.1. PŘÍSTUP K API A K WEBOVÉMU ROZHRANÍ	3
1.2. E-MAILOVÉ NOTIFIKACE	4
2. POPIS API	5
2.1. VERZE API	5
2.1.1. API v1.0	5
2.1.2. API v2.0 a vyšší	6
2.2. OBECNÉ	7
2.2.1. Nastavení jazyka odpovědi	7
2.3. PŘEDÁVÁNÍ PARAMETRŮ U METOD GET A DELETE	7
2.4. TYP VRÁCENÝCH HODNOT	8
2.5. OVĚŘENÍ PŘIPOJENÍ	8
2.6. PŘEHLED CHYBOVÝCH STAVŮ – POLE „CODE“	9
2.7. FUNKCE ALERTY	11
2.7.1. Metoda GET	11
2.7.2. Metoda POST	19
2.7.3. Metoda PUT	20
2.7.4. Metoda DELETE	22
2.8. FUNKCE VÝJIMKY	23
2.8.1. Metoda GET	23
2.8.2. Metoda POST	24
2.8.3. Metoda DELETE	25
3. PŘEHLED ZMĚN	25
3.1 ZMĚNY OPROTI AMS R6.0	25
3.2 ZMĚNY OPROTI AMS R5.0	25
3.3 ZMĚNY OPROTI AMS R4.0	26
3.4 ZMĚNY OPROTI AMS R2.0	26

1. ÚVOD

Systém pro správu alertů (Alert management system – **AMS**) je nadstavbou **Národního systému pro ověřování pravosti léčiv** (NSOL). AMS usnadňuje správu alertů (podezření na možný padělek). AMS byl vytvořen a je spravován **Národní organizací pro ověřování pravosti léčiv, z.s** (NOOL,z.s.).

API CZMVO.CZ je jednoduché **REST API**, které umožňuje napojení vlastního systému pro správu alertů MAH nebo koncového uživatele na systém pro správu alertů NOOL. Tato integrace dovolí automatizaci činností spojených se řešením alertů, a to na straně MAH i koncového uživatele (lékárna, distributor).

Pro uživatele, kteří nemají vlastní systém pro správu alertů, nebo jim politika společnosti integraci přes API nedovoluje, je dostupné webové rozhraní s ekvivalentní funkcionalitou.

Popis webového rozhraní je v samostatných uživatelských příručkách, a to zvlášť pro MAH a zvlášť pro koncové uživatele. V těchto příručkách je také podrobněji popsána logika systému a podpora vzájemné komunikace mezi MAH, koncovým uživatelem a NOOL.

1.1. PŘÍSTUP K API A K WEBOVÉMU ROZHRAŇÍ

Přístupové linky:

Prostředí	Webové rozhraní	API
Produkční - PROD	https://portal.czmvo.cz/	https://api.czmvo.cz/
Testovací - IQE	https://sandbox.czmvo.cz/	https://api.czmvo.cz/t/
Vývojové - ITE	https://beta.czmvo.cz/	https://betaapi.czmvo.cz/

API 2.0 OAuth2 - autorizační linky:

Prostředí	Link
Produkční - PROD	https://api.czmvo.cz/auth/token/
Testovací - IQE	https://api.czmvo.cz/t/auth/token/
Vývojové - ITE	https://betaapi.czmvo.cz/auth/token/

Do **testovacího a vývojového prostředí** se pravidelně **každou noc** nahrává **kopie produkčních dat**. Testovat lze tedy na reálných datech uživatele. Změny v testovacím a vývojovém prostředí nijak neovlivňují data v produkční databázi, lze je tedy libovolně měnit. Každou noc jsou ale data přepsána, s tím je nutné v rámci testování a vývoje počítat.

Testovací prostředí slouží k testování a vývoji software pro stávající produkční prostředí (tj. funkčně je shodné jako produkční prostředí).

Vývojové prostředí slouží k testování a vývoji aplikačního software pro připravovanou verzi AMS, která **bude teprve nasazena na produkční prostředí**.

Přihlašovací údaje do webového rozhraní a API 1.0 jsou:

- a) **Login a heslo**, které jsou generovány a zaslány systémem **na vyžádání na adrese registrace@czmvo.cz**
- b) Jednorázově (pouze pro koncové uživatele) lze použít **ID alertu** (UPRC) jako **login a ID lokace** (provozovny) jako **heslo**.
- c) Jednorázově (pouze pro koncové uživatele) lze použít **ID lokace** (provozovny) jako login a stejné **ID lokace** (provozovny) jako heslo pro ověření, zda je produktový kód (nebo šarže) v seznamu výjimek MZ (žádné jiné oprávnění nemá).
- d) Pro společnosti, vytvářející software pro koncové uživatele, je možné zřídit **vývojový účet**. S tímto účtem se lze přihlásit pouze a výhradně do testovacího a vývojového prostředí (týká se to jak webového rozhraní, tak samotného API). Po přihlášení si mohou (resp. musí) zvolit za jakého koncového uživatele v rámci testovacího prostředí budou vystupovat (pokud nezvolí, pak nebudou mít oprávnění v přístupu k žádnému alertu ani zprávě). Toto nastavení koncového uživatele lze kdykoliv změnit ve webovém rozhraní API – výběrem (selectboxem) umístěným vedle nastavení jazykové mutace v hlavičce.

Přístupové údaje do API 2.0 jsou:

- a) **Client ID a Secret ID** – přístupové údaje pro OAuth2. **Generují se výhradně ve webovém rozhraní** a slouží k získání **přístupového tokenu**.
Upozornění: **Uživatel musí být nejprve registrován ve webovém rozhraní, teprve poté si může vygenerovat přístupové údaje.**
Klíče „Client ID“ a „Secret ID“ jsou různé pro jednotlivá prostředí (produkční/testovací), je proto nutné je **generovat v odpovídající verzi webového rozhraní**.
- b) Pro získání tokenu (pouze pro koncové uživatele) lze použít **ID alertu** (UPRC) jako „Client ID“ a **ID Lokace** jako „Secret ID“. Takový uživatel má přístup **výhradně k jednomu konkrétnímu alertu**, a nemůže provádět hromadné operace a operace nad skupinami.
- c) Pro získání tokenu (pouze pro koncové uživatele) lze použít **ID Lokace** jako „Client ID“ a **ID Lokace** jako „Secret ID“. Takovýto uživatel nemá přístup k žádným alertům (ani k nahlášení). Přístupy lze použít pouze k ověření, **zda je produktový kód (nebo šarže) v seznamu výjimek Ministerstva zdravotnictví**.

1.2. E-MAILOVÉ NOTIFIKACE

Systém v závislosti na svém nastavení využívá celou řadu **automatických e-mailových notifikací**. Automatické notifikace jsou používány při automatických eskalacích stavů, při uzavírání alertů koncovým uživatelem nebo NOOL (konfirmasi požadované změny). Také po každém vložení nové zprávy/souboru systém generuje e-mailovou notifikaci.

Notifikace je odesílána v čistě textové formě a v kódování **UTF-8**. Pro snadnější automatické zpracování je pak ke zprávě ještě připojena **patička**, která je určena pro automatické zpracování a je ve tvaru:

****EVENT:NEWMESSAGE*ID:20 ****

Samotná pole se mohou do budoucna měnit, ale formát zůstane zachován takto:

****PARAMETR1:HODNOTA1*PARAMETR2:HODNOTA2* ... *PARAMETR X :HODNOTA X ****

Možné parametry/hodnoty jsou následující:

EVENT – hodnota: v současné době pouze NEWMESSAGE

ID – hodnota: ID nově vložené zprávy (pro EVENT: NEWMESSAGE)

2. POPIS API

2.1. VERZE API

Počínaje verzí AMS 6.0 je zavedeno verzování API. Vše před tímto releasem je považováno za **API 1.0**. API 1.0 se zveřejněním release 6.0 stává nepodporovanou a nedoporučovanou verzí a **bude ze systému AMS nejpozději během podzimu 2023 odstraněna**.

Ve všech odpovědích ze systému jsou přítomny hlavičky s informací o stavu verzí:

Příklad:

```
amscz-version: 2.1
amscz-supported-versions: 2.0,2.1
amscz-deprecated-versions: 1.0
```

amscz-version - použitá verze API při zpracování požadavku.

supported-versions – seznam aktuálně podporovaných verzí, verze jsou oddělené čárkou

deprecated-versions – seznam nepodporovaných verzí, které budou ze systému odstraněny

2.1.1. API 1.0

Pro autorizaci se používá **basic auth**.

Jednoduchý vzorový kód požadavku v PHP a CURL: (pro ostatní jazyky bude obdobné).

```
$ch = curl_init();
curl_setopt($ch, CURLOPT_URL, URL);
curl_setopt($ch, CURLOPT_RETURNTRANSFER, true);
curl_setopt($ch, CURLOPT_USERPWD, „LOGIN:PASSWORD“);
curl_setopt($ch, CURLOPT_HTTPAUTH, CURLAUTH_BASIC);
curl_setopt($ch, CURLOPT_CUSTOMREQUEST, http_METHOD);
curl_setopt($ch, CURLOPT_POSTFIELDS, „REQUEST“ );
curl_setopt($ch, CURLOPT_HTTPHEADER, array('Content-
Type:application/json`));
$result = curl_exec($ch);
```

Kde:

- URL je adresa funkce.
- LOGIN, PASSWORD jsou přihlašovací údaje.

Přihlašovací údaje jsou buď generovány systémem na vyžádání, nebo je možné pro přístup ke konkrétnímu alertu použít (pro koncové uživatele) jako login „ID alertu“ – UPRC a jako heslo ID lokace.

- http_METHOD je jedna z metod „GET“, „POST“, „DELETE“, „PUT“.
- REQUEST – JSON formátovaný požadavek.

2.1.2. API 2.0 a vyšší

Pro autorizaci se používá **OAuth2**. Oproti API 1.0 jsou navíc povinné některé doplňující údaje v hlavičce požadavku.

2.1.2.1. Obecný požadavek do API

Příklad požadavku pro API 2.0:

```
GET /alerts/?list=messages&changedFrom=2021-07-06+12%3A00%3A00 HTTP/1.1
Host: api.czmvo.cz
User-Agent: Client 1.0
amscz-version: 2.0
Authorization: Bearer eyJtc2ciOiJUYWR5IGplIG7Em2tkbyB6dsSbZGF2w70i...
Accept-Language: en
Accept: application/json
```

User-Agent – povinný údaj – identifikuje použitý software a jeho verzi (název je volitelný, jeho maximální délka by neměla přesáhnout 100 znaků).

amscz-version – povinný údaj – specifikuje jakou verzi API má systém použít zpracování požadavku. Aktuální hodnota je 2.0.

Authorization – autorizační token (povinný).

Accept-Language – nepovinné – určuje jazyk odpovědi.

Accept – povinný údaj – požadovaný výstup odpovědi (obvykle `application/json` nebo `application/csv`).

2.1.2.2. Získání autorizačního tokenu

Pro získání autorizačního tokenu je potřeba **Client ID** a **Secret ID** (viz bod 1.1 PŘÍSTUP K API A K WEBOVÉMU ROZHRANÍ).

Požadavek (pro produkční prostředí):

```
POST /auth/token/ http/1.1
Host: api.czmvo.cz
Content-Type: application/x-www-form-urlencoded
User-Agent: Client 1.0
Cache-Control: no-store
grant_type=client_credentials&client_id=id&client_secret=secret
```

Odpověď je pak:

```
http/1.1 200 OK
Content-Type: application/json
{
  „access_token“:„eyJtc2ciOiJUYWR5IGplIG7Em2tkbyB6dsSbZGF2w70i...“,
  „expires_in“:1800,
  „token_type“:„Bearer“
```

```
}
```

kde **access token** se používá při provádění požadavků.

Expires in – platnost autorizačního tokenu v sekundách.

Pokud se autorizace nezdaří, pak API vrátí chybu **http/1.1 400 Bad Request** s obsahem:

```
{„error“:„invalid_client“}
```

2.2. OBECNÉ

Jedná se o standardní **REST API**. Výměna dat probíhá přes formát JSON (pokud není stanoveno jinak).

Výstupem je JSON, formát se liší dle použité funkce nebo metody. Základní struktura je následující:

```
{„status“:„ok“,„code“:0,„message“:„OK“,„result“:{ ... výsledek ...}}
```

Pokud „code“ je různý od nuly, pak v „message“ je popis chyby. Pole „status“ je buď „ok“ – požadavek proběhl nebo „error“ – požadavek neproběhl (obvykle to znamená – pokud nenastala interní chyba API, že „code“ – číslo chyby – je nenulové).

2.2.1. Nastavení jazyka odpovědi

Některé hodnoty v odpovědích můžou být lokalizovatelné – např. chybové hlášení API, stavy alertů v číselníku. Pro nastavení požadovaného jazyka lze použít http hlavičku „Accept-Language“.

Podporované jazyky jsou v současné době čeština (**cs**) a angličtina (**en**).

```
Příklady: Accept-Language: en
```

nebo

```
Accept-Language: cs-CZ
```

2.3. PŘEDÁVÁNÍ PARAMETRŮ U METOD GET A DELETE

U metod GET a DELETE některé systémy nebo knihovny neumožňují přenos dat v těle požadavku (POST data). V tomto případě je potřeba parametry požadavku zadat přímo v url. Parametry předávané v url musí plně korespondovat s parametry z požadovaného JSON dotazu.

Příklad:

Pokud je JSON požadavek „Zjištění všech zpráv od 6.7.2022 12:00:00“

```
{„list“:„messages“,„changedFrom“:„2022-07-06 12:00:00“}
```

pak předání parametrů přes url (pro testovací prostředí) bude:

<https://api.czmvo.cz/alerts/?list=messages&changedFrom=2022-07-06+12%3A00%3A00>

2.4. TYP VRÁCENÝCH HODNOT

Parametr požadavků „*resultAs*“ je nahrazen odpovídající http hlavičkou „*Accept*“. Parametr „*resultAs*“ je zachován pro zpětnou kompatibilitu.

Pro „*resultAs*“: „*json*“:

```
Accept: application/json
```

Pro „*resultAs*“: „*csv*“:

```
Accept: text/csv
```

V API 2.0 a vyšší je parametr „*resultAs*“ zrušen a ignorován, a je zcela nahrazen povinnou hlavičkou „*Accept*“.

2.5. OVĚŘENÍ PŘIHOJENÍ

Pro ověření správnosti připojení k API a jeho funkčnosti lze využít doplňující parametr „*connection*“. Pokud k jakémukoliv požadavku na API je navíc připojen GET parametr „*connection*“ s nastavenou hodnotou „**verify**“, pak se samotný požadavek neprovede, ale provede se pouze ověření připojení a autentizace (url požadavku bude např. <https://api.czmvo.cz/t/?connection=verify>)

Pro ověření připojení není potřeba zasílat ani žádný požadavek, stačí jen parametr „*connection*“ s odpovídající hodnotou „**verify**“.

Pro účely ověření (pouze pro něj) lze alternativně pro koncového uživatele jako přihlašovací údaje použít jako login a heslo **pouze ID lokace** (shodné pro login i heslo).

Pokud je odeslán požadavek:

```
{ „list“: „enumState“ }
```

na url <https://api.czmvo.cz/alerts/?connection=verify>

pak odpověď je:

```
{ „status“: „ok“, „code“: 0, „message“: „OK“, „result“: { „method“: „GET“, „module“: „alerts“, „Environment“: „production“, „auth“: „Regular“, „userrole“: „Enduser“, „state“: true } }
```

kde jednotlivé pole mohou nabývat následujících hodnot:

method – metoda požadavku – jedna z GET, POST, PUT, DELETE.

Module – volaná funkce API – aktuálně může nabývat hodnot „*alerts*“ nebo „*filter*“.

Environment – informace, zda se jedná o volání na sandbox nebo produkční server, možné hodnoty: „*production*“, „*sandbox*“.

auth – způsob autentifikace uživatele.

Možné návratové hodnoty jsou:

- „No authorization“ – přihlašovací údaje jsou neplatné.
- „Regular“ – standardní přihlášení (přes login a heslo).
- „Enduser alert based“ – přihlášení Koncového uživatele pomozí lokace a UPRC.
- „Verify only“ – přihlášení pouze pro ověření připojení (login i heslo je id lokality) – toto přihlášení nelze použít pro provádění požadavků.

Userrole – role uživatele (pokud je úspěšně přihlášen) – možné hodnoty: „N/A“ – pokud neproběhlo přihlášení, „Enduser“ – koncový uživatel, „MAH/OBP“ – MAH.

State – bool hodnota – „true“ – pokud je připojení v pořádku, „false“ – pokud nastal při připojení jakýkoliv problém.

2.6. PŘEHLED CHYBOVÝCH STAVŮ – POLE „CODE“

code (kód chyby)	http response	Popis
0	200	Požadavek proběhl v pořádku.
1	404	Neznámá funkce <i>Pravděpodobně chybné url požadavku.</i>
2	401	Neověřený přístup. <i>Ověření uživatele neproběhlo v pořádku, uživatele nelze přihlásit.</i>
3	401	Funkce nepovolena. <i>Pravděpodobně chybné url požadavku, nebo uživatel nemá oprávnění k přístupu do API.</i>
4	405	Nepovolené volání metody. <i>Neznámá http metoda (povolené jsou GET, POST, PUT, DELETE).</i>
5	400	Nepovolená hodnota pro parametr. <i>Vypíše i konkrétní parametr.</i>
11	400	Nevyplněna hodnota pro parametr. <i>Pravděpodobně není vyplněn povinný parametr.</i>
12	404	Alert nebyl nalezen.
13	405	Není oprávnění k zápisu do alertu.
14	400	Soubor se nepodařilo dekodovat. Chybně zakódovaný soubor v JSON požadavku
15	400	Velikost souboru překročila limit MB. <i>Uvede aktuální maximální velikost souboru (16MB).</i>
16	500	Zprávu se nepodařilo uložit. <i>Interní chyba API.</i>
17	401	Není oprávnění k úpravě zprávy.
18	401	Na zprávu nelze odpovědět. <i>Buď již neexistuje nebo je uzavřená.</i>
19	401	Zprávu nelze smazat. <i>Ke zprávě je přiřazená odpověď, nelze ji proto smazat.</i>
20	400	Musí být zadán alespoň jeden parametr UPRC, ID.
21	404	Soubor ID %s nebyl nalezen.
22	401	Není oprávnění ke čtení souboru.

23	415	Soubor není podporovaného typu. Podporované typy jsou <i>txt, pdf, csv, jpg, png, tiff</i>
24	500	Interní chyba API. <i>Dále nespecifikovaná chyba softwarová části API. Případný požadavek opakujte později.</i>
25	405	Alert nelze upravit, je již archivovaný. <i>Alerty se archivují 90 dní po uzavření alertu.</i>
26	405	Alert nelze upravit, je přiřazený jinému MAH. <i>Chyba vzniká při hromadném importu při pokusu nastavit stav u alertu cizího MAH.</i>
27	401	Nelze nastavit stav pro alert. Neočekávaný nový stav. <i>Požadovaný stav alertu nelze nastavit, protože neodpovídá procesnímu workflow.</i>
28	401	Nelze nastavit stav pro alert. Není oprávnění. <i>Uživatel není oprávněn provést požadovanou změnu stavu.</i>
29	401	Nelze nastavit stav pro alert. Alert je uzavřen.
30	401	Nelze nastavit stav pro alert. Nejsou splněny všechny podmínky. <i>Nastavení stavu je možné, ale nebyly splněny všechny doplňující podmínky (např. nebyl zadán „Důvod znovuotevření“).</i>
31	401	Zprávu k alertu nelze zaslat. Alert není ve správném stavu pro zaslání zprávy. <i>Nelze nastavit požadovaný stav alertu po odeslání zprávy, protože nekoresponduje s alertním workflow.</i>
32	400	Požadována specifikace chyby NSOL.
33	400	Byl zadán nepodporovaný typ hlavičky požadavku Accept, nebo nebyl zadán. <i>Je požadováno zasílání hlavičky Accept v hlavičce požadavku na API.</i>
34	405	Alert nelze upravit, je přiřazený jinému Koncovému uživateli
35	405	Alert nelze upravit, požadavek neodpovídá alertnímu workflow.
36	404	ID Lokace nebylo nalezeno. <i>Požadovaná lokace nebyla nalezena v databázi.</i>
37	401	Není oprávnění přístupu k lokaci. <i>Uživatel nemá oprávnění k přístupu k požadované lokaci.</i>
38	400	Neplatný nebo expirovaný Autorizační token. <i>Autorizační token není platný. Je potřeba vygenerovat nový.</i>
39	400	Neplatný požadavek. <i>Při požadavku na API v2+ není vyplněna některá z povinných http hlaviček.</i>
40	401	Akce nad skupinou nelze provést. <i>Jeden nebo více alertů ve skupině je ve stavu, ve kterém nelze provést požadavek. Seznam dotčených alertů je v odpovědi.</i>

2.7. FUNKCE ALERTY

url funkce: <https://api.czmvo.cz/alerts/>

url funkce pro testovací prostředí: <https://api.czmvo.cz/t/alerts/>

url funkce pro vývojové prostředí: <https://betaapi.czmvo.cz/alerts/>

2.7.1. Metoda GET

2.7.1.1. Načtení Stavů alertů

Příklad požadavku (zjistění stavů alertů):

```
{„list“:„state“,„uprc“:„CZ-0VG-ZZW-5BU-LZ0“,„latest“:true,„createdFrom“:„2022-08-06 00:00:00“,„createdTo“:„2022-08-13 00:00:00“,„changedFrom“:„2022-08-04 00:00:00“,„state“:1,„page“:1}
```

Příklad odpovědi:

```
{„status“:„ok“,„code“:0,„message“:„OK“,„result“:{„alerts“:[{„uprc“:„CZ-0VR-Y94-KK5-6FJ“,„created“:„2022-07-16 07:50:04“,„productcode“:„08595116521485“,„stateid“:1,„state“:„Nov\u00fd“,„lastmessageid“:„20“,„statedescription“:„Nov\u00fd“ } ]}}
```

Příklad požadavku (zjistění počtu stránek subsetů):

```
{„list“:„state“,„page“:-1}
```

Příklad odpovědi:

```
{„status“:„ok“,„code“:0,„message“:„OK“,„result“:{„pages“:3,„currentPage“:0}}
```

Rozšíření odpovědi pro koncové uživatele:

Pokud má přihlášený uživatel roli „Koncový uživatel“, pak jsou v odpovědi navíc pole *typestate* a *pepatedescription*, které indikují požadovanou akci.

Příklad odpovědi:

```
{„status“:„ok“,„code“:0,„message“:„OK“,„result“:{„pages“:1,„currentPage“:1,„alerts“:[{„uprc“:„CZ-KSR-RLB-6MF-E8C-8RT“,„created“:„2022-05-05 11:07:00“,„productcode“:„08594175410327“,„stateid“:1,„state“:„01a - Nov\u00fd - transakce KU“,„lastmessageid“:0,„statedescription“:„Nov\u00e1 v\u00fdstra\u00fdha - v\u00fdsledk investigace NOOL p\u00e1dn\u00e1 v poli „P\u00e1dn\u00e1 - automatick\u00e1“. Balen\u00e1 m\u00e1te v karant\u00e9n\u00e1. Pokud se ale domn\u00e1te, \u00e1te se jedn\u00e1 o odstranitelnou chybu na va\u00ed stran\u00e1, po odstran\u00e1n\u00e1 probl\u00e9mu se m\u00e1te pokusit o op\u00edt\u00edverifikaci. Pokud bylo n\u00e1sledn\u00e9 ov\u00e1n\u00e1n\u00e1 \u00e1sp\u00e1n\u00e1, lze LP vydat, a alert m\u00e1te v AMS uzav\u00e1t pomoc\u00e1 p\u00e1n\u00e1 stavu. „“,„typestate“:„Informace
```

```
MAH", "tpestatedescription": "Po\u017eadov\u00e1ny dodate\u010dn\u00e9  
informace od u\u017eivatele" ] ] ] }
```

Poznámky:

Pokud je parametr nepovinný, pak se do požadavku nemusí zadávat.

resultAs (pouze API 1.0) může nabývat hodnot „json“ nebo „csv“. Pokud je nastaveno „csv“, pak výstupem je tabulka ve formátu csv.

Uprc – jednoznačný identifikátor alertu.

List – vždy „state“.

Latest – true/false – řazení od nejnovějšího alertu.

createdFrom – zobrazit pouze alerty novější než zadaný čas. Všechny časy se do systému vkládají ve formátu „RRRR-MM-DD HH:MM:SS“. Veškeré časy jsou uváděné v UTC.

createdTo – zobrazit pouze alerty starší než zadaný čas.

changedFrom – zobrazit pouze alerty, u kterých je evidovaná změna stavu od zadaného času.

State – zobrazit pouze alerty s odpovídající stavem (dle číselníku stavu).

Page – celé číslo – určuje, která stránka výsledků (subset) se vrátí v odpovědi. Maximální počet alertů (tj. jedna stránka) v jednom subsetu je **500** (v závislosti na nastavení konkrétních oprávnění může mít vybraný subset počet alertů nižší). Pokud se v požadavku parametr *page* nastaví jako záporné číslo (např. -1), pak v odpovědi bude pouze celkový počet stran subsetů.

V json odpovědi jsou pak vždy vráceny hodnoty **pages** a **currentPage**.

V odpovědi pak (kromě zřejmých polí):

alerts je vždy pole – i v případě vrácení jednoho výsledku.

Stateid – ID statusu alertu.

State – textové vyjádření ID statusu.

Statedescription – podrobný popis stavu v závislosti na roli přihlášeného uživatele.

Lastmessageid – poslední vložená zpráva.

Pages – počet stránek záznamů (subsetů).

currentPage – aktuální vrácená stránka výsledků v poli alerts (pokud je parametr *page* kladný, pak se jedná o tuto hodnotu).

2.7.1.2. Načtení Zpráv

Příklad požadavku:

Zjištění všech zpráv od 6.7.2022 12:00:00

```
{ „list“: „messages“, „changedFrom“: „2022-07-06 12:00:00“ }
```

Zjištění všech zpráv k alertu CZ-0VG-ZZW-5BU-LZP

```
{ "list": "messages", "uprc": "CZ-0VG-ZZW-5BU-LZP" }
```

Zjištění/načtení zprávy s ID 20

```
{ "list": "messages", "id": "20" }
```

Příklad odpovědi:

```
{ "status": "ok", "code": 0, "message": "OK", "result": { "messages": [ { "id": "19",  
  "parent": "0", "uprc": "CZ-0VR-Y94-KK5-6FJ", "created": "2022-07-06  
10:45:59", "changed": "2022-07-06  
10:45:59", "subject": "info", "message": "Uplne  
ok", "isfile": false, "public": false, "fromme": true }, { "id": "20", "parent": "1  
9", "uprc": "CZ-0VR-Y94-KK5-6FJ", "created": "2022-07-06  
10:59:06", "changed": "2022-07-06 10:59:06", "subject": "Re:  
info", "message": "Fajn", "isfile": false, "public": true, "fromme": false } ] } }
```

Poznámky:

uprc – jednoznačný identifikátor alertu, jedná se o povinný parametr, pokud není zadán parametr **id** nebo **changedFrom**.

list – vždy „messages“.

id – ID Zprávy.

changedFrom – čas ve formátu „RRRR-MM-DD HH:MM:SS“ – zobrazit jen Zprávy, které jsou novější než zadaný čas. Pokud není zadán parametr **uprc** ani **id**, pak hodnota **changedFrom** by neměl být datum starší než **1 měsíc**.

V odpovědi pak (kromě zřejmých polí):

messages – je vždy pole i v případě jediného výsledku.

id – ID Zprávy.

parent – ID Zprávy, na kterou se odpovídá (tj. ID dotazy/požadavku).

uprc – jednoznačný identifikátor Alertu.

created – čas vytvoření Zprávy.

changed – čas poslední úpravy Zprávy.

subject – předmět Zprávy.

message – text samotné Zprávy.

isfile – true/false – zpráva obsahuje soubor.

public – true/false – pokud je „true“, pak je Zpráva viditelná všem zainteresovaným stranám u alertu. Pokud je „false“, pak je Zpráva viditelná pouze tomu, kdo ji vložil (a NOOLu).

fromme – true/false – příznak, že Zprávu vytvořil aktuálně přihlášený Uživatel.

id_request – ID požadavku/zprávy z číselníku zpráv (viz GET Načtení číselníku zpráv), nebo 0.

2.7.1.3. Načtení Souboru

Příklad požadavku pro API 1.0:

```
{ "resultAs": "json", "list": "file", "id": "21" }
```

nebo (přímé zaslání souboru) pro API 1.0

```
{ "resultAs": "csv", "list": "file", "id": "21" }
```

Příklad požadavku pro API 2.0 a vyšší:

```
{ "list": "file", "id": "21" }
```

V hlavičce požadavku pro API 2.0 je možné nastavit:

```
Accept: application/json
```

nebo (pro získání binárních dat)

```
Accept: application/octet-stream
```

Příklad odpovědi:

```
{ "status": "ok", "code": 0, "message": "OK", "result": { "filename": "xxx.pdf", "filedata": "JVBERi0xLjcKCjQgMCBvYmoKKElkZW50aXR5KQplbmRvYmoKNSAwIG9iagooQWRvYmUpCmVuZG9iag4IDAgb2JqCjw8Ci9GaWx0ZXIgL0ZsYXRlRGVjb2RlCi9MZW5ndGgODk1NjAKL1R5cGUGL1N0cmVhbQo+PgpzdHJlYW0KeJzsfQlgVMX9\3fese+9Pd\em80mu5vNRUIOkpADAtlwiVIkAmKiRsOlYD2CcqkV8ATBA6sith7xQsWDJfEiIBWPHmqtWK2irTWtt5Wf2FJqlez+vzO7yW7kaNJ\apn19z678515M\OdmTfzne+b77y3+4AAQBoSAdZPmHH8cYYfjn4MhDMAbx\OG7CxElb6zruAe63XQB8xXEN02acdCDjb8C97QCSPu+4GSeP+\zOs\4E\ ...
```

Poznámky:

resultAs (pouze API 1.0) – *json* – pak vrátí soubor ve formátu který je naznačen v příkladu výše.

resultAs (pouze API 1.0) – *csv* – pak pošle soubor přímo (s odpovídajícím mimetypem).

list – vždy „file“.

2.7.1.4. Načtení číselníku Stavů alertů

Příklad požadavku:

```
{ "list": "enumState" }
```

Příklad odpovědi:

```
{ "status": "ok", "code": 0, "message": "OK", "result": { "states": [ { "id": 1, "name": "Nov\u00fd", "externalcode": "01", "finalstate": false }, { "id": "5", "name": "V\u0159e\u0161en\u00ed", "externalcode": "#", "finalstate": false }, { "id": "3", "name": "Uzav\u0159en\u00fd", "externalcode": "06a,06b,06c", "finalstate": true }, { "id": "6", "name": "Odlo\u017een\u00fd", "externalcode": "", "finalstate": false }, { "id": "7", "name": "Chyba import na callcentrum", "externalcode": "CALLFAIL", "finalstate": false } ] } }
```

Rozšíření odpovědi pro koncové uživatele:

Pokud má přihlášený Uživatel roli „Koncový uživatel“, pak jsou v odpovědi navíc textová pole *typestate* a *typestatedescription*, které indikují požadovanou akci.

Příklad odpovědi:

```
{ "status": "ok", "code": 0, "message": "OK", "result": { "states": [ { "id": 1, "name": "01a - Nov\u00fd - transakce KU", "externalcode": "01a", "finalstate": false, "settingallowed": false, "description": " ... ", "typestate": "Informace MAH", "typestatedescription": "Po\u017eadav\u00edny dodate\u010dn\u00e9 informace od u\u017eivatele", ...
```

Poznámky:

resultAs (pouze API 1.0) může nabývat hodnot „json“ nebo „csv“ – pokud je nastaveno csv, pak výstupem je tabulka ve formátu csv.

states – vždy pole.

name – název Stavů.

externalcode – kód dle kódů číselníku Stavů.

finalstate – true/false – příznak, zda Stav je konečný (tj. zda je alert vyřešen).

settingallowed – true/false – příznak, zda je možné tento Stav alertu nastavit přes API při splnění podmínek workflow.

description – podrobný popis Stavů v závislosti na Roli přihlášeného Uživatele.

2.7.1.5. Načtení číselníku Zpráv

Příklad požadavku:

```
{ "list": "enumRequest" }
```

Příklad odpovědi:

```
{ "status": "ok", "code": 0, "message": "OK", "result": { "requests": [ { "id": 1, "name": "Fotka", "text": "\u017d\u00e1d\u00e1me o zasl\u00e1n\u00ed fota obalu LP, s \u010diteln\u00fdm 2D k\u00f3dem" }, { "id": 2, "name": "Fotka_EAN", "text": "\u017d\u00e1d\u00e1me o zasl\u00e1n\u00ed fota obalu LP, s \u010diteln\u00fdm 2D k\u00f3dem. Nafotte pros\u00edme i vizu\u00e1ln\u00ed \u011b \u010diteln\u00e9 \u00fadaje (EAN, \u011bar\u017ee, SN, datum expirace, apod.)" } ] }
```

Poznámky:

resultAs (pouze API 1.0) může nabývat hodnot „json“ nebo „csv“ – pokud je nastaveno csv, pak výstupem je tabulka ve formátu csv.

requests – vždy pole, kde:

id – jednoznačný identifikátor požadavku/zprávy, uvádí se při zaslání Zprávy

name – název Zprávy

text – předdefinovaný text Zprávy

forStates – pole celých čísel (int), seznam ID stavů alertů (viz. kap. 2.7.1.4), pro které lze Zprávu zaslat. (Alert musí být v jednom z vyjmenovaných Stavů, aby bylo možné Zprávu zaslat).

name – název Stavů.

text – text samotné Zprávy.

2.7.1.6. Načtení skupiny alertů

Pouze pro MAH.

Umožňuje načíst seznam všech alertů, které se nachází ve stejné skupině jako výchozí alert. Systém do skupin zařazuje podle systémově definované podobnosti jednotlivých alertů.

Příklad požadavku:

```
{ "list": "group", "uprc": "CZ-0VR-YE5-C1N-KLM" }
```


Příklad odpovědi:

```
{ „status“: „ok“, „code“: 0, „message“: „OK“, „result“: { „uprc“: [ „CZ-0VR-YE5-C1N-KLM“, „CZ-0VR-YE5-VS7-BXP“ ] } }
```

Poznámky:

uprc – v dotazu – číslo výchozího (jednoho) Alertu.

V odpovědi:

uprc – seznam všech alertů, které patří do stejné Skupiny. Pokud alert do žádné skupiny nepatří, pak je v odpovědi prázdné pole.

2.7.1.7. Načtení číselníku „Důvody znovuotevření“

Při pokusu o změnu stavu alertu, který je již uzavřený, je vyžadován „**Důvod jeho znovuotevření**“ (pokud je možné ho opětovně otevřít). Důvod otevření se nastavuje parametrem při nastavení stavu.

Příklad požadavku:

```
{ „list“: „enumReopenReason“ }
```

Příklad odpovědi:

```
{ „status“: „ok“, „code“: 0, „message“: „OK“, „result“: { „reasons“: [ { „id“: 1, „name“: „Chybn\u011b uzav\u0159eno“ } ] } }
```

V odpovědi:

reasons – aktuální číselník. Při nastavení stavu alertu se pak v příslušném parametru předává ID důvodu.

2.7.1.8. Načtení vlastní Poznámky

Slouží k načtení existující Poznámky a jejích případných parametrů.

Příklad požadavku: (načtení vlastní Poznámky)

```
{ „list“: „note“, „uprc“: „CZ-LD8-F79-ABY-PFC-5J0“ }
```

Příklad požadavku: (načtení Poznámky jiného uživatele, který dal k tomu povolení)

```
{ „list“: „note“, „uprc“: „CZ-LD8-F79-ABY-PFC-5J0“, „from“: „mah“ }
```

Příklad odpovědi:

```
{ „status“: „ok“, „code“: 0, „message“: „OK“, „result“: { „note“: „Test note“, „uprc“: „CZ-LD8-F79-ABY-PFC-5J0“, „private“: false, „changed“: „2022-01-12 11:13:42“ } }
```

Poznámky:

uprc – v dotazu – číslo alertu, ke kterému je Poznámka přiřazena.

From – nepovinný parametr, obsahuje jednu z hodnot „mah“, „enduser“, „nool“ (pokud je zadána jiná hodnota, pak se API chová jako by zadána žádná nebyla). Specifikuje, či

Poznámka se má načíst. Aby bylo možné načíst cizí Poznámku, pak tato nesmí být označena jako „interní“.

V odpovědi:

uprc – číslo Alertu (pokud je alert, ke kterému je požadována Poznámka, nalezen).

Note – text Poznámky (nebo prázdný řetězec, pokud Poznámka nebyla dosud vložena).

Private – true/false – pokud je „true“, pak Poznámka je interní a v rámci systému není přístupná žádné třetí straně. Pokud je „false“, pak Poznámku mohou vidět ostatní účastníci.

Changed – datum a čas poslední úpravy Poznámky. Pokud Poznámka neexistuje, pak je hodnota „null“.

2.7.1.9. Načtení číselníku „Typ stav“

Tento číselník může načíst pouze koncový uživatel. Jedná se o číselník indikací pro Uživatele, co mají v dané situaci provést s daným balením léčivého přípravku. (viz sekce 2.7.1.1. a 2.7.1.4). Ke každému stavu alertu je přiřazen právě jeden „Typ stav“.

Příklad požadavku:

```
{ „list“: „enumTypeState“ }
```

Příklad odpovědi:

```
{ „status“: „ok“, „code“: 0, „message“: „OK“, „result“: { „typestates“: [ { „name“: „N“, „description“: „Neprov\u00e1d\u011bt nic“ } ] } }
```

V odpovědi:

typestates – aktuální číselník.

2.7.1.10. Načtení čítače pro level 3 alerty

Slouží pro zjištění aktuálního inkrementálního čítače pro generování čísel level 3 alertů. Funkce je použitelná pouze pro koncového uživatele.

Příklad požadavku:

```
{ „list“: „alcounter“, „location“: „858d085f-324a-4938-a796-333bfac94f05“ }
```

Příklad odpovědi:

```
{ „status“: „ok“, „code“: 0, „message“: „OK“, „result“: { „counter“: 2 } }
```

Poznámky:

location – kód Provozovny. Povinné pole.

V odpovědi:

counter – celé číslo – aktuální stav čítače pro vybranou Provozovnu.

2.7.1.11. Načtení level 3 alertů

Slouží pro načtení aktuálního seznamu level 3 alertů. Funkce je použitelná pouze pro koncového uživatele.

Příklad požadavku:

```
{"list": "a1alerts", "location": "858d085f-324a-4938-a796-333bfac94f05"}
```

Příklad požadavku: (omezení dle čítače)

```
{"list": "a1alerts", "location": "858d085f-324a-4938-a796-333bfac94f05", "from": 2}
```

Příklad odpovědi:

```
{"status": "ok", "code": 0, "message": "OK", "result": {"a1alerts": [{"uprc": "CZ-858d085f-324a-4938-a796-333bfac94f05-000001", "created": "2022-10-19 07:48:04", "productcode": "18901138057340", "stateid": 6}, {"uprc": "CZ-858d085f-324a-4938-a796-333bfac94f05-000002", "created": "2022-10-19 07:48:38", "productcode": "18901138057340", "stateid": 6}]}}
```

Poznámky:

location – kód lokality/provozovny. Povinné pole.

from – celé číslo – omezí seznam level 3 alertů pouze na alerty s čítačem větším nebo rovno zadané hodnotě. Parametr je nepovinný (pokud není zadán, funkce vrátí všechny alerty)

V odpovědi:

a1alerts – pole – seznam alertů

stateid – ID stavu alertu dle číselníku stavů

Důležité: body 2.7.1.10. a 2.7.1.11. jsou platné jen pro API 1.0, pro API 2.0 bude volání nahrazeno bodem 2.7.2.2.

2.7.1.12. Zjištění možných akcí nad alertem

Funkce slouží pro získání seznamu možných operací nad konkrétním alertem, které lze aktuálně provést.

Příklad požadavku:

```
{"list": "allowedActions", "uprc": "CZ-LD8-F79-ABY-PFC-5J0"}
```

Odpověď:

```
{"status": "ok", "code": 0, "message": "OK", "result": {"sendMessage": [16, 14], "setState": [39, 40, 41], "group": true, "group_a": false}}
```

Poznámka:

uprc – číslo existujícího alertu. Pokud alert neexistuje, pak je vrácena chyba 12.

V odpovědi:

sendMessage – pole celých čísel (nebo prázdné pole – pokud nelze zaslat žádná zpráva) – seznam ID zpráv (viz bod 2.7.1.5. Načtení číselníku Zpráv), které lze aktuálně zaslat.

setState – pole celých čísel (nebo prázdné pole – pokud nelze změnit stav alertu) – seznam ID stavů alertu (viz bod 2.7.1.4. Načtení číselníku Stavů alertů), které lze aktuálně nastavit u alertu.

group – bool hodnota – „true“ – pokud je alertu přiřazen v některé skupině (a předpokládá se provádět operaci nad skupinou).

group_a – bool hodnota – „true“ – pokud je alertu přiřazen v některé anonymní skupině (a předpokládá se provádět operaci nad skupinou).

2.7.2. Metoda POST

Požadavek – vložení zprávy pomocí číselníku zpráv (jediná možná metoda pro komunikaci mezi MAH a Koncovým uživatelem):

```
{"uprc":"CZ-0VR-Y94-KK5-6FJ","public":true,"id_request":1}
```

Požadavek (jednoduché vložení):

```
{"uprc":"CZ-0VR-Y94-KK5-6FJ","public":true,"subject":"test","message":"test"}
```

Požadavek (vložení jako odpověď na jinou zprávu):

```
{"uprc":"CZ-0VR-Y94-KK5-6FJ","public":true,"id_parent":20,"subject":"Re: Re: info","message":"test"}
```

Požadavek (vložení souboru):

```
{"uprc":"CZ-0VR-Y94-KK5-6FJ","public":false,"subject":"Re: Re: info","message":"test","file":"iVBORw0KGgoAAAANSUgAAABAAAAQCAIAAAACQkWg2AAAACXBIWXMAAAsTAAALEwEAmpwYAAAAIGNIUk0AAHolAAAgwAA+f8AAIDpAAAB1MAAA6mAAADqYAAAXb5JfxUYAAADjSURBVHjaYnwaJcSABBhZ2RlQwf\ /fP5G5THiUwsWRpZiQV ePSgyzFhF8dJmBB43PahAmkTEAWeRYtDLf\ /\ /++fjE+jhCB88cmXmXiFcRn8PEEK6iRGVn ZOmzDJBc9eZKihGfw8QQquTnLBM4Qfvh9Z9WFOgdTSt8+iETZILX0LCVBkQUQo4dGDFiQIT 38\ /sgqi6FmkgNTyD3A9EMaHOQVYQgmqZ\ /mHZ9HCcKUQ1RApBgYGFohD4fYi7IHpQVb9\ \ /dPFswQhOtBVooz4pD1oKmGugU5teJJI\ /A0y4QnJcMFkcUBAwCuU3b1BVKmxQAAAABJRU 5ErkJggg==" ,"filename":"test.png"}
```

Odpověď:

```
{"status":"ok","code":0,"message":"OK","result":{"id":22}}
```

API 2.0 a vyšší:

Pokud se provádí operace nad celou skupinou (jsou nastavené parametry „group“ nebo „group_a“), a není možné operaci provést nad jedním nebo více alerty, pak se operace neprovede a systém vrátí seznam alertů, které blokují z důvodu svého stavu dokončení operace:

Příklad odpovědi:

```
{"status":"error","code":40,"message":"Akce nad skupinou nelze provést","result":{"uprc":["CZ-0VR-Y94-KK5-6FJ","CZ-1VR-Y94-KK5-6FI"]}}
```

Poznámka:

ID zprávy je vždy celé číslo, je unikátní v rámci celého systému a platí, že novější zpráva bude mít vždy vyšší ID.

uprc – jednoznačný identifikátor alertu, je to povinné pole, pokud není vyplněno pole *id_parent*.

public – *true/false* – příznak, že zpráva je veřejná ve smyslu, že k ní má přístup MAH i koncový uživatel (pokud je nastaveno na *false*, pak ke zprávě má přístup pouze autor a administrátor systému).

id_parent – ID zprávy, na kterou se odpovídá. Pokud se odpovídá na zprávu, pak parametry *group* a *group_a* jsou ignorovány a zpráva je přiřazena ke všem alertům, na které byla zaslána původní zpráva, a ke kterým má uživatel oprávnění.

subject – povinné pole, pokud není vložen soubor - předmět zprávy.

message – text zprávy. Pole je povinné, pokud není vložen soubor.

file – base64 zakódovaný binární soubor (bez omezení typu).

filename – jméno souboru (pokud se vkládá soubor, pak se jedná o povinné pole).

id_request – ID požadavku (zprávy) z číselníku zpráv – viz GET Načtení číselníku Zpráv.

group – *true/false* – pokud je *“true”*, pak zpráva/soubor bude zaslán na všechny alerty ve skupině, ke kterým má uživatel oprávnění. Pokud parametr není zadán, pak se bere jeho hodnota jako *„false“*.

group_a – *true/false* – pokud je *“true”*, pak zpráva/soubor bude zaslán na všechny alerty v anonymní skupině, ke kterým má uživatel oprávnění. Pokud parametr není zadán, pak se bere jeho hodnota jako *„false“*.

only_file – *true/false* – Pokud je parametr nastaven na *„true“*, pak se systém ke zprávě chová pouze jako k samostatnému souboru, a nikoliv jako k regulérní zprávě (není změněn stav alertu, nejsou odeslány notifikace). Pokud je tento parametr nastaven na *„true“*, pak musí být vložen soubor.

2.7.3. Metoda PUT

2.7.3.1. Upravit existující zprávu

Umožňuje upravit pole *public*, *subject* a *message* u existující Zprávy.

Pozn.: Upravit data může pouze autor, nemůže být upravována Zpráva, na kterou již existuje odpověď).

Příklad požadavku:

```
{"id":22,"public":true,"subject":"TEST"}
```

Odpověď:

```
{"status": "ok", "code": 0, "message": "OK", "result": {"id": 22, "changed": "2019-08-06 15:36:35"}}
```

Poznámky:

id – ID zprávy

Upravit lze pouze pole *public*, *subject*, *message* a *id_request* (viz. method POST).

2.7.3.2. Upravit stav alertu

Umožňuje upravit stav jednoho alertu nebo více alertů.

Požadavek: (Natavení stavu alertu)

```
{"uprc": "CZ-0VR-YE5-VS7-BXP", "state": 5, "group": false}
```

Požadavek: (Hromadné nastavení stavu)

```
{"uprc": "CZ-0VR-YE5-VS7-BXP", "state": 5, "group": true}
```

Požadavek: (Alternativní hromadné nastavení stavu)

```
{"uprc": ["CZ-0VR-YE5-VS7-BXP", "CZ-0VR-YE5-C1N-KLM"], "state": 5}
```

Odpověď:

```
{"status": "ok", "code": 0, "message": "OK", "result": {"uprc": ["CZ-0VR-YE5-C1N-KLM", "CZ-0VR-YE5-VS7-BXP"]}}
```

API 2.0 a vyšší:

Pokud se provádí operace nad celou skupinou (jsou nastavené parametry „*group*“ nebo „*group_a*“) a není možné operaci provést nad jedním nebo více alerty, **pak se operace neprovede** a systém vrátí seznam alertů, které blokují z důvodu svého stavu dokončení operace:

Příklad odpovědi:

```
{"status": "error", "code": 40, "message": "Akce nad skupinou nelze provést", "result": {"uprc": ["CZ-0VR-Y94-KK5-6FJ", "CZ-1VR-Y94-KK5-6FI"]}}
```

Poznámka:

uprc – jednoznačný identifikátor alertu, je to povinné pole

state – ID nového stavu alertu. Seznam možných stavů lze získat viz kapitola **Načtení číselníku Stavů**. U stavu, který je možné nastavit přes API, je uveden příznak „**settingallowed**“.

group – true/false – Pokud je „*true*“, pak je stav nastaven pro všechny alerty ze stejné skupiny (viz metoda GET, získání alertů ze skupiny) jako vstupní alert (parametr „**uprc**“).

Pokud je nastaven na „*false*“, pak se změna stavu týká pouze zadaného alertu.

group_a – true/false – shodné jako „**group**“, ale pro anonymní skupinu.

id_request – ID zprávy z číselníku zpráv – viz GET Načtení číselníku Zpráv.

V odpovědi pak:
v parametru „**uprc**“ je vždy seznam dotčených (upravených) alertů.

2.7.3.3. Upravit vlastní Poznámku

Umožňuje vložit, resp. upravit, existující vlastní Poznámku k alertu.

Požadavek:

```
{"uprc": "CZ-LD8-F79-YBY-PFC-5J0", "private": true, "note": "TEST NOTE" }
```

Odpověď:

```
{"status": "ok", "code": 0, "message": "OK", "result": {"uprc": "CZ-LD8-F79-YBY-PFC-5J0", "changed": "2022-01-12 10:43:06"}}
```

Poznámka:

uprc – jednoznačný identifikátor alertu, je to povinné pole

note – povinné pole (pokud jde o prázdný řetězec, pak dojde k vymazání Poznámky), maximální délka Poznámky je 60 000 znaků.

private – *true/false* – nepovinné pole. Pokud není zadáno pak se použije výchozí hodnota „*true*“. Pokud je hodnota tohoto parametru „*true*“, pak se jedná o **interní** Poznámku a není viditelná pro nikoho v rámci systému. Pokud je hodnota „*false*“ – pak je Poznámka viditelná pro všechny strany.

V odpovědi pak:

changed – čas poslední změny Poznámky – obvykle aktuální čas (pokud se však při požadavku na úpravu Poznámky nemění žádná hodnota, pak jde o čas poslední skutečné změny).

2.7.4. Metoda DELETE

2.7.4.1. Odstranění Zprávy

Požadavek:

```
{"id": 22}
```

Odpověď:

```
{"status": "ok", "code": 0, "message": "OK", "result": {"id": 22}}
```

Poznámka:

Parametr „**id**“ (ID Zprávy) je jediným možným parametrem. Podmínky smazání Zprávy jsou shodné jako podmínky při její úpravě.

2.7.4.2. Odstranění vlastní Poznámky

Požadavek:

```
{"note": "CZ-LD8-F79-YBY-PFC-5J0"}
```

Odpověď:

```
{"status": "ok", "code": 0, "message": "OK", "result": {"uprc": "CZ-LD8-F79-YBY-PFC-5J0"}}
```

Poznámka:

note – povinné pole. Jedná se o „**uprc**“, u kterého se má odstranit Zpráva. V odpovědi pak hodnota „**uprc**“ slouží pro kontrolu.

2.8. FUNKCE VÝJIMKY

url funkce: <https://api.czmvo.cz/filter/>

url funkce pro testovací prostředí: <https://api.czmvo.cz/t/filter/>

url funkce pro vývojové prostředí: <https://betaapi.czmvo.cz/filter/>

2.8.1. Metoda GET

2.8.1.1. Načtení číselníku Stavů

Požadavek:

```
{"list": "enumState"}
```

Odpověď:

```
{"status": "ok", "code": 0, "message": "OK", "result": {"states": [{"code": "NO", "name": "Uzav\u0159eno - MAH - nelze opravit"}, {"code": "OP", "name": "Uzav\u0159eno - MAH - opraveno"}]}}
```

Poznámka:

resultAs (pouze API 1.0) může nabývat hodnot „json“ nebo „csv“ – pokud je nastaveno csv, pak výstupem je tabulka ve formátu csv.

2.8.1.2. Načtení existujících Výjimek

Požadavek:

```
{"list": "product", "productCode": "kod", "batch": "", "id": [1, 2, 3]}
```

Odpověď:

```
{"status": "ok", "code": 0, "message": "OK", "result": {"products": [], "count": 0}}
```

Poznámka:

„**id**“ je pole id, která jsou filtru přiřazena při vložení, parametry pro filtrování „**id**“. Parametry „**productCode**“, „**batch**“ jsou nepovinné.

Tato funkce vrátí pouze Výjimky, které sám uživatel vložil nebo jsou k němu přiřazeny (platí pro MAH).

2.8.1.3. Ověření Výjimky

Požadavek:

```
{"list":"verify","productCode":"08594158891136","batch":"1"}
```

Odpověď:

```
{"status":"ok","code":0,"message":"OK","result":{"isException":true,"info":{"id":41,"productCode":"08594158891136","batch":null,"stateId":1,"state":"Nov\u00fd"}}
```

Poznámka:

Alespoň jedno z polí „productCode“, „batch“ musí být vyplněno.

V odpovědi pak:

isException - true/false – produkt/balení je v seznamu Výjimek.

info – pokud **isException** je „true“, pak v tomto poli je popis samotné Výjimky.

2.8.2. Metoda POST

Požadavek: (jednoduché vložení)

```
{"validity":"2019-04-30","state":"OP","productCode":"0123456789","batch":"123456"}
```

Požadavek: (hromadné vložení přes csv soubor)

```
{"validity":null,"state":"NO","csv":"zCIsIsSMbM0hbmVrIDU3IGtvZGV4dSIsIk1EIGRyxb5pdGVsZSBYZWdpc3RyYWNlICChNQUGgSUQpIiRyxb5pdGVsZSBYZWdpc3RyYWNlICChNQUGpIiwiQWRyZXNhIGRyxb5pdGVsZSBYZWdpc3RyYWNlIiwIc3RvIGRyxb5pdGVsZSBYZWdpc3RyYWNlIiwIUFEjCBkcsW+aXRlbGU0cmFjZSIsIlplbcSbIGRyxb5gaahjfkleuHGAFlgheghv4fdb2xxYVNS5fAS5gdrgpdGVsZSBYZWdpc3RyYWNlIiw5pkYWplIG8gZGlzdHJpYnV0b3JvdmkiDQo="}
```

Odpověď:

```
{"status":"ok","code":0,"message":"OK","result":{"products":[{"lineNo":1,"productCode":"0123456789","batch":"123456","validity":null,"state":1,"ID":1,"errorCode":0,"errorText":""}], "count":0}}
```

Poznámka:

Parametr „validity“ (platnost) je datum ve formátu [YYYY]-[MM]-[DD].

Kód produktu, šarže a platnost jsou povinné parametry. Pokud nejsou zadány, pak Výjimka nebude do systému uložena.

Parametr „state“ (Stav slertu) slouží jako výchozí hodnota pro nastavení Stavů (tj. pokud není v csv definována).

Pro hromadné vkládání je v odpovědi více řádků, kde je vždy přiřazeno ID, nebo popis důvodu nevložení do databáze pro každý řádek.

Parametr „csv“ je base64 csv souboru ve formátu popsaném v sekce „Vložit data“ ve webovém rozhraní.

2.8.3. Metoda DELETE

Požadavek:

```
{"productCode":"0123456789","batch":"123456","id":[1,2]}
```

Odpověď:

```
{"status":"ok","code":0,"message":"OK","result":{"affected":1,"deleted": [{"id":"2","productCode":"0123456789","batch":"123456"}]}}
```

Poznámka:

Parametry „productCode“, „batch“ a pole „id“ jsou nepovinné, ale musí být zadán alespoň jeden parametr (nelze odstranit všechny záznamy jedním požadavkem).

3. PŘEHLED ZMĚN

3.1. ZMĚNY OPROTI AMS R6.0

1. Zjištění možných akcí nad alertem

Přidána funkce pro získání seznamu možných operací nad konkrétním alertem, které lze v daný moment provést.

3.2. ZMĚNY OPROTI AMS R5.0

1. API nastavení

Rozhraní AMS API má omezení na počet dotazů za daný čas. Klientským systémům, které překročí svou kvótu, je vrácen stavový kód HTTP (429).

Pozn.: aktuální nastavení je 800 žádostí za 5 minut na IP adresu nebo 400 žádostí za 5 minut na id client. Klient může znovu ověřit a získat token pouze jednou za 60 minut.

2. API ověření pomocí OAuth 2.0

Pro přístup přes API je implementován protokol OAuth 2.0.

3. Verzování API a doplňující informace

Zavedení verzování API. Informace o podporovaných, nepodporovaných a ukončených verzích v odpovědích.

a) AMS API verze

Verze AMS API, která se má použít, je specifikována pomocí „amsapi-version“ jako hlavičky HTTP. (např. „ams-api-version“ 2.0). Vynechání hlavičky má za následek směrování požadavků do AMS API 1.0.

b) Http hlavička „User-agent“

Použití HTTP hlavičky „User-Agent“ je povinné. Klientské systémy musí poskytnout podrobnosti o verzi svého klientského softwaru.

4. Byly přidány chybové stavy – ID 38-40.

5. Změněna tvorba jednoznačného identifikátoru pro level 3 alerty (po dohodě se SÚKL).

3.3. ZMĚNY OPROTI AMS R4.0

1. **Nový číselník „Typ stav“**

Pouze pro roli „Koncový uživatel“. Je to číselník indikací pro uživatele, co má v dané situaci provést s balením. Ke každému stavu alertu je přiřazen právě jeden „*Typ_stav*“.

2. **Rozšíření číselníku „Stav Alertů“**

Pro roli „koncový uživatel“ jsou přidána pole „*typstate*“ a „*typstatedescription*“ (vazba na nový číselník „*Typ_stav*“).

3. Byly přidány **chybové stavy – ID 34-37**.

4. **Načtení čítače pro level 3 alerty**

Přidána funkce pro koncové uživatele pro zjištění **hodnoty aktuálního čítače** pro generování jednoznačné identifikace level 3 výjimek.

5. **Načtení level 3 alertů**

Přidána funkce pro koncové uživatele pro načtení aktuálního **seznamu výjimek level 3**.

3.4 ZMĚNY OPROTI AMS R2.0

1. **Zřízeno Vývojové prostředí.**

Slouží k testování a vývoji aplikačního software pro verzi AMS, která bude teprve nasazena na produkčního prostředí.

2. **Alertní workflow.**

AMS využívá nové **alertní workflow** (úprava možných stavů pro jednotlivé uživatele, rozšíření a úpravy automatických eskalací, konfirmační e-maily).

3. Parametr požadavků „**resultAS**“ nahrazen http hlavičkou „**Accept**“. Parametr „**resultAs**“ je zachován pro zpětnou kompatibilitu.

4. Nový číselník „**Důvody znovuotevření**“.

5. **Doplňeny chybové stavy** (pole „*code*“) – **ID 25 – 33**.

6. **Přepřepočována práce s grupami a anonymními grupami** (hromadné odpovědi, hromadné změny stavů).

7. Možnost **přidat soubor k alertu** (i bez požadavku MAH).