

API SPRÁVA ALERTŮ

Dokumentace

ver. 4.1

(červenec 2021)

OBSAH

1. ÚVOD.....	3
1.1. PŘÍSTUP K API A K WEBOVÉMU ROZHRANÍ.....	3
1.2. EMAILOVÉ NOTIFIKACE	4
2. POPIS API	4
2.1. OBECNÉ.....	4
2.1.1. Nastavení jazyka odpovědi.....	5
2.2. PŘEDÁVÁNÍ PARAMETRŮ U METOD GET A DELETE	5
2.3. TYP VRÁCENÝCH HODNOT.....	6
2.4. OVĚŘENÍ PŘIPOJENÍ.....	6
2.5. PŘEHLED CHYBOVÝCH STAVŮ – POLE „CODE“.....	7
2.6. FUNKCE ALERTY	8
2.6.1. Metoda GET	8
2.6.2. Metoda POST	12
2.6.3. Metoda PUT	14
2.6.4. Metoda DELETE.....	15
2.7. FUNKCE VÝJIMKY	15
2.7.1. Metoda GET	16
2.7.2. Metoda POST	17
2.7.3. Metoda DELETE.....	17
3. PŘEHLED ZMĚN OPROTI AMS 2.2.	18

1. ÚVOD

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

Pro uživatele, kteří nemají vlastní systém pro správu alertů, nebo jim politika společnosti integraci přes API nedovoluje, je k API rovněž dostupné webové rozhraní s ekvivalentní funkcionalitou. Popis webového rozhraní je v samostatných uživatelských příručkách, a to zvlášť pro MAHy a zvlášť pro koncové uživatele. V těchto příručkách je i podrobněji popsána logika systému a podpora komunikace mezi MAHem a koncovým uživatelem.

1.1. PŘÍSTUP K API A K WEBOVÉMU ROZHRAŇÍ

Přístupové linky:

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

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 pro testování a vývoj softwaru pro stávající produkční prostředí (tj. funkčně je shodné jako produkční prostředí).

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

Přihlašovací údaje jsou:

- Login a heslo**, které jsou generovány a zasílány systémem **na vyžádání na adrese registrace@czmvo.cz**
- Jednorázově (pouze pro koncové uživatele) lze použít **ID alertu** (UPRC) jako **login** a **ID Lokace** (provozovny) jako heslo.
- 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).
- 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.

Přihlašovací údaje slouží k přístupu do API i do webového rozhraní systému.

1.2. EMAILOVÉ 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 NOOLEM (konfirmasi požadované změny). Také po každém vložení nové zprávy/souboru systém generuje e-mailovou notifikaci.

Notifikace je odesílána v čistě textové formě a 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* ... *PARAMETRX:HODNOTAX**
```

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. OBECNÉ

Funkčně je ekvivalentní k webovému rozhraní. Jedná se o REST API, autorizace se používá basic auth. Výměna dat probíhá přes formát JSON (pokud není stanoveno jinak).

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. V případě potřeby lze generovat i ve webovém rozhraní.

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).

Všechny metody mají svůj ekvivalent ve webovém rozhraní. Vše, co platí tam, platí i při přímém použití API. Doporučujeme proto získávat dodatečné informace i ze sekce „Webové rozhraní“.

2.1.1. Nastavení jazyka odpovědi

Některé hodnoty v odpovědích mohou 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
```

2.2. 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 6.7.2021 12:00:00“

```
{"resultAs":"json","list":"messages","changedFrom":"2021-07-06 12:00:00"}
```

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

<https://api.czmvo.cz/alerts/?resultAs=json&list=messages&changedFrom=2021-07-06+12%3A00%3A00>

2.3. TYP VRÁCENÝCH HODNOT

Parametr požadavků „resultAs“ je nahrazen odpovídající http hlavičkou Accept.
Parametr resultAs je zachován pro zpětnou kompatibilitu. Do budoucna bude zrušen.

Pro "resultAs": "json":

```
Accept: application/json
```

Pro "resultAs": "csv":

```
Accept: text/csv
```

2.4. OVĚŘENÍ PŘIPOJENÍ

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 pomůže 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.

2.5. 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 NMB. <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 MAHa.</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 splněné 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 procesním 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>

2.6. 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/>

Generování konkrétních dotazů a zobrazení odpovědí API ve formátu JSON je k dispozici i ve webovém rozhraní API systému.

2.6.1. Metoda GET

Odpovídá operaci „Načíst data“ z webového rozhraní.

2.6.1.1. Načtení Stavů alertů

Příklad požadavku (zjistění stavů alertů):


```
{"resultAs":"json","list":"state","uprc":"CZ-0VG-ZZW-5BU-LZ0","latest":true,"createdFrom":"2019-08-06 00:00:00","createdTo":"2019-08-13 00:00:00","changedFrom":"2019-08-04 00:00:00","state":1,"page":1}
```

Příklad odpovědi:

```
{"status":"ok","code":0,"message":"OK","result":{"alerts":[{"uprc":"CZ-0VR-Y94-KK5-6FJ","created":"2019-07-16 07:50:04","productcode":"08595116521485","stateid":1,"state":"Nov\u00fd","lastmessageid":"20","statedescription":"Nov\u00fd" }]}}
```

Příklad požadavku (zjistění počtu stránek subsetů):

```
{"resultAs":"json","list":"state","page":-1}
```

Příklad odpovědi:

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

Poznámky:

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

resultAs 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 alery 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 alery starší než zadaný čas.

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

state - zobrazit pouze alery 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).

2.6.1.2. Načtení Zpráv

Příklad požadavku:

Zjištění všech zpráv od 6.7.2021 12:00:00

```
{"resultAs":"json","list":"messages","changedFrom":"2021-07-06 12:00:00"}
```

Zjištění všech zpráv k alertu CZ-OVG-ZZW-5BU-LZP

```
{"resultAs":"json","list":"messages","uprc":"CZ-OVG-ZZW-5BU-LZP"}
```

Zjištění/načtení zprávy s ID 20

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

Příklad odpovědi:

```
{"status":"ok","code":0,"message":"OK","result":{"messages":[{"id":"19","parent":"0","uprc":"CZ-OVR-Y94-KK5-6FJ","created":"2021-07-06 10:45:59","changed":"2021-07-06 10:45:59","subject":"info","message":"Uplne ok","isfile":false,"public":false,"fromme":true},{id":"20","parent":"19","uprc":"CZ-OVR-Y94-KK5-6FJ","created":"2021-07-06 10:59:06","changed":"2021-07-06 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.

2.6.1.3. Načtení Souboru

Příklad požadavku:

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

nebo (přímé zaslání souboru)

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

Příklad odpovědi:

```
{"status":"ok","code":0,"message":"OK","result":{"filename":"xxx.pdf","file data":"JVBERi0xLjcKcKjQgMCMCBvYmoKKElkZW50aXR5KQplbmRvYmoKNSAwIG9iagooQWRvYmUp CmVuZG9iagoo4IDAgb2JqCjw8Ci9GaWx0ZXIgL0ZsYXRlRGVjb2RlCi9MZW5ndGggODk1NjAKLlR 5cGUgL1N0cmVhbQo+PgpzdHJlYW0KeJzsfQlgVMX9\3fese+9Pd\em80mu5vNRUIOkpADAtlw iVIkAmKiRsOlYD2CcqkV8ATBA6sith7xQsWDJfEiIBWPHmqtWK2irTWtt5Wf2FJqlez+vzO7yW7 kaNJ\apn19z678515M\OdmTfzne+b77y3+4AAQBoSAdZPmHH8cYYfjn4MhDMAAbx\OG7CxE1 b6zruAe63XQB8xXEN02acdCDjb8C97QCSpu+4GSeP+\zOs\4E\ ...
```

Poznámky:

resultAs – json – pak vrátí soubor ve formátu který je naznačen v příkladu výše.

resultAs – csv – pak pošle soubor přímo (s odpovídajícím mimetypem).

list – vždy „file“.

2.6.1.4. Načtení číselníku Stavů alertů

Příklad požadavku:

```
{"resultAs":"csv","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,"na me":"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}]}}
```

Poznámky:

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

states – vždy pole.

name – název stavu.

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 slertu nastavit přes API při splnění podmínek workflow.

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

2.6.1.5. Načtení číselníku Zpráv

Příklad požadavku:

```
{"resultAs":"csv","list":"enumRequest"}
```

Příklad odpovědi:

```
{"status":"ok","code":0,"message":"OK","result":{"requests":[{"id":1,"n ame":"Fotka","text":"\u017d\u00e1d\u00e1me o zasl\u00e9ln\u00e9 fota obalu LP, s \u010diteln\u00fdm 2D k\u00f3dem"}, {"id":2,"name":"Fotka_EAN","text":"\u017d\u00e1d\u00e1me o
```

```
zasl\u00e1n\u00e1 fota obalu LP, s \u010diteln\u00fdm 2D k\u00f3dem.  
Nafotte pros\u00edm i vizu\u00e1ln\u011b \u010diteln\u00e9 \u00fadaje  
(EAN, \u0161ar\u017ee, SN, datum expirace, apod.)"]}]}
```

Poznámky:

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

requests – vždy pole.

name – název stavu.

text – text samotné zprávy.

2.6.1.6. 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 podobnosti jednotlivých alertů.

Příklad požadavku:

```
{"resultAs":"json","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.

2.6.1.7. Načtení číselníku „Důvody znovuootevření“

Při pokusu o změnu stavu alertu, který je již uzavřený je vyžadován Důvod jeho znovuootevření (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.

2.6.2. Metoda POST

Požadavek – vložení zprávy pomocí číselníku zpráv (preferovaná varianta):

```
{"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": "iVBORw0KGgoAAAANSUgAAABAAAAQCAIAAACQkWG2AAAACXBIWXMAAAAsTAAALEwEAmpwYAAAAIGNIUk0AAHolAAAgwAA+f8AAIDpAAAB1MAAA6mAAADqYAAAXb5JfxUYAAADjSURBVHjaYnwaJcSABBhZ2RlQwF\ /fp5G5THiUwsWRpZiQVePSgyzFhF8dJmBB43PahAmkTEAWeRYtDLf\ / \ / ++fjE+jhCB88cmXmXiFcRn8PEEK6iRGVnZOmzDJBC9eZKihGf w8QQquTnLBM4QfVh9Z9WFOgdTSt8+iETZILX0LCVBkQUQo4dGDFiQIT38\ /sgqi6FmkgNTyD3A9EMaHOQVYQgmqZ\ /mHZ9HCcKUQ1RApBgYGFohD4fYi7IHpQVb9\ / \ /dPFswQhOtBVooz4pD1oKmGugU5teJJI\ /A0y4QnJcMFkcUBAwCuU3b1BVKmxQAAAABJRU5ErkJggg==", "filename": "test.png"}
```

Odpověď:

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

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 regulární zprávě (není změněn stav

alertu, nejsou odeslány notifikace). Pokud je tento parametr nastaven na true, pak musí být vložen soubor.

2.6.3. Metoda PUT

2.6.3.1. 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).

2.6.3.2. Upravit stav alertu

Umožňuje upravit stav alertu nebo více alertů.

Požadavek: (Nastavení 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}
```

Požadavek: (Nastavení alertu pro standardní zpracování)

```
{"uprc":"CZ-0VR-YE5-VS7-BXP","state":-1,"group":false}
```

Odpověď:

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

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.

2.6.4. Metoda DELETE

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. FUNKCE VÝJIMKY

Jedná se o REST API, autorizace se používá basic auth. Výměna dat probíhá přes formát JSON (pokud není stanoveno jinak).

Jednoduchý vzorový kód požadavku v PHP a CURL:

```
$ch = curl_init();  
curl_setopt($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.
- HTTP_METHOD je jedna z metod „GET“, „POST“, „DELETE“.
- REQUEST – JSON formátovaný požadavek.

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.

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/>

2.7.1. Metoda GET

2.7.1.1. Načtení číselníku stavů

Požadavek:

```
{"resultAs": "json", "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 může nabývat hodnot „json“ nebo „csv“ – pokud je nastaveno csv, pak výstupem je tabulka ve formátu csv.

2.7.1.2. Načtení existujících výjimek

Požadavek:

```
{"resultAs": "json", "list": "product", "productCode": "kod", "batch": "", "serialNumber": "", "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“, „serialNumber“ ... jsou nepovinné.

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

2.7.1.3. Ověření výjimky

Požadavek:

```
{"resultAs": "json", "list": "verify", "productCode": "08594158891136", "batch": "1", "serialNumber": "00000045287"}
```

Odpověď:

```
{"status": "ok", "code": 0, "message": "OK", "result": {"isException": true, "info": {"id": 41, "productCode": "08594158891136", "batch": null, "serialNumber": "00000045287", "stateId": 1, "state": "Nov\u00fdd"}}
```

Poznámka:

Alespoň jedno z polí productCode, batch nebo serialNumber 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

2.7.2. Metoda POST

Požadavek: (jednoduché vložení)

```
{"validity":"2019-04-30","state":"OP","productCode":"0123456789","batch":"123456","serialNumber":"0123"}
```

Požadavek: (hromadné vložení přes csv soubor)

```
{"validity":null,"state":"NO","csv":"zCIsIsSMbMOhbmVrIDU3IGtvZGV4dSIsIk  
lEIGRyxb5pdGVsZSByZWdpc3RyYWNlICChNQUggSUQpIiRyxb5pdGVsZSByZWdpc3RyYWNlI  
ChNQUgpIiwiQWRyZXNhIGRyxb5pdGVsZSByZWdpc3RyYWNlIiwiTc3RvIGRyxb5pdGVsZSB  
yZWdpc3RyYWNlIiwiUFPEjCBkcsW+aXRlbGU0cmFjZSIsIlplbcSbIGRyxb5gaahjfkleuH  
GAF1gheghv4fdb2xxYVNS5fAS5gdrgrpGVsZSByZWdpc3RyYWNlIiwiw5pkYWplIG8gZGlz  
dHJpYnV0b3JvdmkiDQo="}
```

Odpověď:

```
{"status":"ok","code":0,"message":"OK","result":{"products":[{"lineNo":1,"productCode":"0123456789","batch":"123456","serialNumber":"0123","validity":null,"state":1,"ID":1,"errorCode":0,"errorText":""}], "count":0}}
```

Poznámka:

Parametr „validity“ (platnost) je datum zadán ve formátu [YYYY]-[MM]-[DD], pokud formát neodpovídá, pak je bere jako, že není zadán.

Parametr „state“ (Stav slertu) slouží jako výchozí hodnota pro nastavení stavu (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.

2.7.3. Metoda DELETE

Požadavek:

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

Odpověď:

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

Poznámka:

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

3. PŘEHLED ZMĚN OPROTI AMS 2.2.

1. Zřízeno **vývojové prostředí** - slouží k testování a vývoji aplikačního software pro verzi AMS, která bude teprve nasazena na produkčního prostředí.
2. AMS využívá nové **alerní 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. V další verzi AMS **bude zrušen**.
4. **Doplňeny chybové stavy** (pole „code“) – ID 25 – 33.
5. **Nový číselník** „Důvody znovuotevření“.
6. **Přepřacována práce s grupami a anonymními grupami** (hromadné odpovědi, hromadné změny stavů).
7. Možnost **přidat soubor k alertu** (i bez požadavku MAH).