Nemzeti Adó- és Vámhivatal

NAV API Gateway
M2M REST API autentikáció leírás és fejlesztői dokumentáció

Nemzeti Adó- és Vámhivatal

Tartalomjegyzék

1 BEVEZETÉS 1

1.1 CÉL 1
1.2 ADÓZÓKRA VONATKOZÓ HASZNÁLATI KÖVETELMÉNYEK 1
1.3 A KAPCSOLÓDÁSHOZ IMPLEMENTÁLANDÓ TECHNOLÓGIÁK 2
1.4 ÜGYVITELI PROGRAMOKRA VONATKOZÓ TECHNIKAI KÖVETELMÉNYEK 2

2 API GATEWAY AUTENTIKÁCIÓ ISMERTETÉSE 3

2.1 XML ÜZENETEK ÁLTALÁNOS FELÉPÍTÉSE 3
2.2 BASICREQUESTTYPE 3
2.2.1 BasicHeaderType 3
2.2.2 UserHeaderType 4
2.3 BASICRESPONSETYPE 4
2.4 A REQUESTSIGNATURE SZÁMÍTÁSA 5
2.4.1 Számítás fájlfeltöltés esetén 6
2.4.2 Számítás egyéb operáció esetén 7
2.5 ÁLTALÁNOS TECHNIKAI ADATOK 7
2.5.1 HTTP fejlécek 7
2.5.2 HTTP státuszkódok 7
2.5.3 Méretkorlát 8
2.5.4 Válaszidő, timeout 8
2.5.5 Szerveróra, NTP 8
2.5.6 Helyi idő konvertálása UTC időre 8
2.5.7 Karbantartási mód 8
2.5.8 Verziókezelés 8
2.5.9 Karakterkonverzió 8
2.5.10 Forgalomkorlátozás 8

3 HIBAKEZELÉS 9

3.1 ÁLTALÁNOS HIBAKÓDOK 9
3.1.1 GeneralExceptionResponseType 9
3.1.2 GeneralErrorResponseType 9
3.2 TECHNIKAI HIBAKÓDOK 11

4 HELPDESK ÉS TECHNIKAI SEGÍTSÉGNYÚJTÁS 16

4.1 IMPLEMENTÁCIÓ ELLENŐRZÉSÉT SEGÍTŐ ESZKÖZÖK 16
4.2 HELPDESK ELÉRHETŐSÉG 16
4.3 GITHUB ELÉRHETŐSÉG 16
4.3.1 Common repository 16
4.3.2 eÁFA repository 16

Nemzeti Adó- és Vámhivatal

Kifejezések, rövidítések

Kifejezés	Leírás
Adózó	Az a Magyarországon nyilvántartásba vett áfaalany, aki nevében a felhasználók eljárnak.
Aláírókulcs	Jelen dokumentum értelmében egy karaktersorozat, mely segítségével más karakter vagy jelsorozat kiegészítésre, “aláírásra” kerül.
API	Alkalmazásprogramozási interfész.
BASE64	64 karakterből álló ábécén alapuló tartalomkódolási forma, melynek segítségével bináris, illetve speciális karaktereket tartalmazó adatokból ASCII karaktersorozat állítható elő (Binary-to-text encoding, RFC 3548).
M2M	Gép-gép kapcsolat.
NAV	Nemzeti Adó- és Vámhivatal.
Online Számla rendszer	Az online számlaadat-szolgáltatások feldolgozását végző rendszer. https://onlineszamla.nav.gov.hu/
Gyártó	Az API kommunikációt megvalósító programot fejlesztő természetes vagy jogi személy.
SHA-512	512 bites Biztonságos HASH algoritmus (Secure Hash Algorithm 3, RFC 6234).
SHA3-512	512 bites Keccak titkosítású biztonságos HASH algoritmus (FIPS 202).)
Technikai felhasználó	A REST API-n keresztüli kommunikációhoz szükséges user, melyet az Elsődleges felhasználó hozhat létre az Online Számla rendszerben.
XML	Kiterjeszthető Jelölő Nyelv (eXtensible Markup Language, W3C standard https://www.w3.org/TR/xml/).
XSD	XML-séma definíciós fájl (XML Schema Definition, W3C standard https://www.w3.org/TR/xmlschema11-1/).
REST	Representational state transfer (REST) vagy másnéven RESTful webszolgáltatás.
Elsődleges felhasználó	A rendszer azon felhasználója, aki az adózó maga vagy törvényes képviselője vagy állandó meghatalmazottja, és ezáltal teljes körű jogosultsággal rendelkezik a rendszer használata tekintetében. (Ez alól csak a REST API-n keresztüli adatforgalom a kivétel, mely az elsődleges felhasználó által létrehozott technikai felhasználóval teljesíthető.)
Endpoint	Olyan elérési út, amelyen keresztül az operáció által nyújtott szolgáltatás elérhető.
Single Sign-On (SSO)	Webes egyszeri bejelentkezési módszer, amely olyan speciális formája a szoftveres azonosításnak, ami lehetővé teszi a felhasználó számára, hogy egy adott rendszerbe való belépéskor mindössze csak egyszer azonosítsa magát és ezután a rendszer minden erőforrásához és szolgáltatásához további autentikáció nélkül hozzáfér.
Operáció	Azon informatikai eljárások, szolgáltatások, amelyek meghívhatók a kiajánlott REST webszolgáltatáson keresztül.
Webszolgáltatás	Alkalmazások közötti adatcserére szolgáló protokollok és szabványok gyűjteménye.

Dokumentum történet

Dátum	Szerző	Verzió	Változtatás
2023.05.04.	N.Á.	0.1	Kezdeti verzió
2023.06.06	N.Á.	0.2	Első kiadásra szánt változat

Nemzeti Adó- és Vámhivatal

1 BEVEZETÉS

A generikus API kommunikációt megvalósító típusok és elemek a korábbiakban kiemelésre kerültek egy közös XSD-be (common.xsd). Ez teszi lehetővé, hogy ezen sémaleírót felhasználva több NAV-os projekt által használt API kommunikációja egységesíthető legyen.

Ezen céllal került létrehozásra egy központi API Gateway alkalmazás, amely előtétrendszerként a kérések autentikációját és autorizációját, illetve a kérések elosztását végzi.

A fejlesztés első fázisában a Gateway kizárólag az eÁFA M2M rendszert szolgálja ki. A dokumentum az eÁFA gépi interfészen alkalmazandó általános autentikációs metódust írja le. A gépi interfészen alkalmazott autentikációs és autorizációs mechanizmusok a fejlesztők számára transzparensen a common sémában definiált típusokra épülnek, és ugyanazon autentikációs hibakódokat alkalmazzák, mint amelyek már korábban az Online Számla rendszerben kialakításra kerültek.

1.1 Cél

A dokumentum célja a gép-gép kapcsolaton keresztüli kommunikáció hitelesítés működésének, illetve az általa használt XML-üzenetstruktúráknak bemutatása, valamint az adózók saját rendszereinek interfészeihez történő integráció támogatása.

Jelen dokumentum a következő sémaleírók üzleti és műszaki tartalmát foglalja magában:

Séma	Tartalom
common.xsd	Generikus, NAV kommunikációt leíró típusok, katalógus elemek és primitívek

1.2 Adózókra vonatkozó használati követelmények

A gép-gép kommunikáció megvalósításához az adózónak rendelkeznie kell egy technikai felhasználóval a megfelelő jogosultságokkal.

A technikai felhasználó létrehozása és beállítása az alábbi lépések mentén teljesíthető:

1) Az áfaalanynak, törvényes képviselőnek vagy állandó meghatalmazottnak legalább egyszer be kell jelentkeznie az Online Számla rendszerbe.

2) A szolgáltatást használni kívánó áfaalanynak előzetesen az Online Számla rendszer felhasználó szerkesztő felületén technikai felhasználót szükséges létrehoznia vagy egy meglévő technikai felhasználónak megfelelő jogosultságot beállítania, tehát használható akár az Online Számla rendszer interfész kommunikációjához létrehozott technikai felhasználó is. Az adatkommunikáció kizárólag gép-gép kapcsolattal, technikai felhasználó útján lehetséges. Az adózó tetszőlegesen megválaszthatja, hogy a gép-gép kommunikációhoz mennyi technikai felhasználót hoz létre. A technikai felhasználó létrehozása és jogosultságainak beállítása csak elsődleges felhasználók által tehető meg.

3) A technikai felhasználó számára aláírókulcsot és cserekulcsot kell generáltatni a kialakításra került felületen. Az aláírókulcs az üzenetek aláírására szolgáló requestSignature számításában játszik szerepet, míg a cserekulcs az esetleges adatküldési token szerveroldali elkódolásához és a kliensoldali dekódolásához szükséges.

Jogosultságok kiosztása:

Az eÁFA esetén egy többszintű jogosultságrendszer került kialakításra, ahol az első szintet az interfész elérési jogosultság jelenti. A második szinten az eÁFA interfész specifikus jogosultságai szerepelnek, amelyeket a használni kívánt operációknak megfelelően szintén az elsődleges felhasználóknak szükséges beállítaniuk. A megfelelő jogosultságok nélkül a kliens összes kérése visszautasításra kerül a megfelelő hibakóddal, amely a „**HIBAKEZELÉS”** fejezetben található.

A jogosultságok az alábbi módon állíthatóak be az egyes technikai felhasználókhoz:

1) A technikai felhasználók számára az elsődleges felhasználóknak a szükséges jogosultságokat ki kell osztaniuk az erre a célra kialakításra került webes felületen. Az Online számla webes felületére bejelentkezve a Felhasználók menüponton belül szabályozhatóak az adott technikai felhasználó jogosultságai, így az is, hogy elérheti-e az eÁFA gépi interfészét.

2) Emellett az eÁFA gépi interfészen belül gyakorolható jogosultságokat az eÁFA webes felületére átlépve szükséges beállítani. A webes felületeken az egyszeri bejelentkezési módszer (SSO) segítségével elegendő egyszer hitelesítenie magát a felhasználónak.

A felsorolt követelmények rendszersíkonként értendők. A tesztkörnyezetben létrehozott technikai felhasználók és kulcsok nem használhatók az éles környezetben, hasonlóan az Online Számla rendszerhez.

1.3 A kapcsolódáshoz implementálandó technológiák

- HTTPS – Biztonságos HTTP
- Webservice - Webszolgáltatás
- REST API – Adatszolgáltatáshoz szükséges REST interfész
- XML – Kiterjeszthető Jelölő Nyelv
- Kódolási és titkosítási algoritmusok

A fenti elvárások a gyakorlatban azt jelentik, hogy ha egy ügyviteli szoftver képes volt az Online Számla rendszerrel API kapcsolatot kialakítani, akkor az eÁFA M2M rendszerhez is képesnek kell lennie kialakítani a kapcsolatot, mivel nem tervezünk az Online Számla rendszertől eltérő technológiát alkalmazni.

1.4 Ügyviteli programokra vonatkozó technikai követelmények

1) A gépi interfészt bármely, az adóalany által alkalmazni kívánt olyan program (továbbiakban: ügyviteli program) igénybe veheti, amely képes az adott szakrendszerben kialakításra kerülő specifikációban meghatározott HTTP üzenet küldésére és séma-konform XML összeállítására.

2) Az ügyviteli programnak minden adatbeküldésnél és adatlekérdezésnél az adózó technikai felhasználójának hitelesítési adatait is küldenie kell. Az ehhez szükséges implementációt az ügyviteli program szabadon meghatározhatja. Nem szükséges kliens oldalon automatizmusokat kialakítani, az adózói igényeknek megfelelő folyamatok szabadon kialakíthatók.

3) Az ügyviteli programnak a sikeres autentikáció elvégzéséhez a következő kódolási és titkosítási algoritmusokat kell implementálnia:

- BASE64 encode/decode (RFC 3548)
- SHA-512 encode (RFC 6234)
- SHA3-512 encode (FIPS 202)

Az ügyviteli programokra vonatkozó technikai követelmények nagyban hasonlók, mint az Online Számla API kapcsolat esetében. Az alapvető folyamatok nagymértékben azonosak a számlaadat-szolgáltatásnál kialakított folyamatokkal. Eltérések természetesen vannak, hiszen itt nem egy valós idejű adatszolgáltatás került kialakításra.

2 API GATEWAY AUTENTIKÁCIÓ ISMERTETÉSE

Az API Gateway az autentikációt és autorizációt az alábbi fejezetekben leírt típusokban szereplő adatok alapján végzi. Minden kérésben szükséges szerepeltetni a technikai felhasználó autentikációs adatait, illetve minden kéréshez ki kell számolni a kéréshez tartozó requestSignature értékét. A requestSignature, mint aláírás biztosítja az üzenetek hitelességét, integritását és letagadhatatlanságát.

2.1 XML üzenetek általános felépítése

A szakrendszereknél, amelyek az API Gateway hitelesítését veszik igénybe, az üzleti operációkat úgy kell kialakítani, hogy a request-response elemek kiterjesszék a common.xsd-ben definiált BasicRequestType és BasicResponseType elementeket.

2.2 BasicRequestType

Minden request element kötelező része kell legyen a BasicRequestType-ban szereplő header és user csomópont. A típuson belül a header az üzenetváltással kapcsolatos általános technikai adatokat, a user pedig az autentikációval kapcsolatos adatokat tartalmazza.

2.2.1 BasicHeaderType

A kérésekben a header elementet a BasicHeaderType implementálja.

A BasicHeaderType felépítése:

Tag	Típus	Kötelező	Tartalma
requestId	xs:string	igen	A kérés egyedi azonosítója
timestamp	xs:dateTime	igen	A kérés kliensoldali időpontja UTC-ben
requestVersion	xs:string	igen	A kérés verziószáma
headerVersion	xs:string	nem	A header verziószáma

Facetek és leírók:

Tag	SimpleType	Pattern	Enum	Default
requestId	EntityIdType	[+a-zA-Z0-9_]{1,30}	-	-
timestamp	GenericTimestampType	\d{4}-\d{2}-\d{2}T\d{2}:\d{2}:\d{2}(.\d{1,3})?Z	-	-
requestVersion	AtomicStringType15	-	-	-
headerVersion	AtomicStringType15	-	-	-

Leírás és kapcsolódó követelmények

1) A requestId a kérés azonosítója. Értéke bármi lehet, ami a pattern szerint érvényes és az egyediséget nem sérti. A requestId-nak - az adott adózó vonatkozásában és a timestamp tűrési toleranciáján belül - kérésenként egyedinek kell lennie. Az egyediségbe minden sémavalid kérés beleszámít, azaz minden sikeres kérés és minden olyan kérés, amely szerver oldalon értelmezhető, tehát nem INVALID_REQUEST hibakóddal kerül visszautasításra. A tag értéke beleszámít a requestSignature értékébe.

A timestamp a kérés beküldésének időpontja a kliens órája szerint. A tag értéke beleszámít a requestSignature értékébe. A timestamp-nek a kérésben UTC időben és megfelelő formátum szerint kell érkeznie.

Ez magyarországi időzóna esetén:

- DT (téli időszámítás) idején GMT+1 órát
- DST (nyári időszámítás) esetén GMT+2 órát jelent.

A timestamp értékének megengedett toleranciája a szerveridőhöz képest +- 1 nap.

A dátumokkal kapcsolatosan a „**Helyi idő konvertálása UTC időre**” fejezet ad felvilágosítást.

2) A requestVersion a kérés struktúráját azonosítja. A későbbi interfészváltozások erre a tagra lesznek visszavezetve, így a requestVersion a kérés és a válasz struktúráját, az ahhoz kapcsolódó validációkat, ellenőrzéseket is meghatározza. Értéke a támogatott verzió értékének megfelelően töltendő. Üzleti validáció vizsgálja az értékét, nincs XSD-szintű enum értékkészlete. Minden szakrendszerre vonatkozóan külön értelmezendő. Interfészverzió-váltások esetén egyszerre több támogatott verzió is lehetséges.
   - Az eÁFA esetén az induláskor támogatott érték: 1.0
3) A headerVersion opcionális elem a kérésben. Arra szolgál, hogy ha a jövőben a kérések struktúrája is alapvetően megváltozna, akkor a különböző struktúrák és az azokhoz kapcsolódó ellenőrzések erre a tagra lesznek visszavezetve. Üzleti validáció vizsgálja az értékét, nincs XSD-szintű enum értékkészlete.

Jelenleg egyetlen elfogadott értéke van, amennyiben megadásra kerül: 1.0

2.2.2 UserHeaderType

A kérésekben a user elementet a UserHeaderType implementálja.

A UserHeaderType felépítése:

Tag	Típus	Kötelező	Tartalma
login	xs:string	igen	A technikai felhasználó login neve
passwordHash	xs:complexType	igen	A technikai felhasználó jelszóhash értéke
taxNumber	xs:string	igen	Azon adózó adószámának első 8 jegye, aki az interfész szolgáltatását igénybe veszi, és akihez a technikai felhasználó tartozik
requestSignature	xs:complexType	igen	A kérés aláírásának hash értéke

Facetek és leírók

Tag	SimpleType	Pattern	Enum	Default
login	LoginType	[a-zA-Z0-9]{6,15}	-	-
passwordHash	CryptoType	-	-	-
taxNumber	TaxpayerIdType	[0-9]{8}	-	-
requestSignature	CryptoType	-	-	-

2.3 BasicResponseType

Minden szakrendszeri operáció úgy került definiálásra, hogy a response element a BasicResponseType-ot egészítse ki. A komplex típus, valamint a benne definiált header és result csomópontjai a common.xsd-ből származnak. A típuson belül a header a válasz tranzakcionális adatait, a result a feldolgozás eredményét tartalmazza.

A válaszban adott header csomópont szerkezetileg és tartalmilag mindig meg fog egyezni a kérésben szereplő header tagek adataival.

A válaszokban a feldolgozási eredményt a BasicResultType implementálja.

A BasicResultType felépítése:

Tag	Típus	Kötelező	Tartalma
funcCode	xs:string	igen	A feldolgozás eredménye
errorCode	xs:string	nem	A feldolgozás hibakódja
message	xs:string	nem	A feldolgozási eredményhez vagy hibakódhoz tartozó szöveges üzenet
notification/notificationCode	xs:string	nem	Értesítés kód
notification/notificationText	xs:string	nem	Értesítés szöveg

Facetek és leírók

Tag	SimpleType	Pattern	Enum	Default
funcCode	FunctionCodeType	-	OK ERROR	-
errorCode	SimpleText50NotBlankType	.*[^\s].*	-	-
message	SimpleText1024NotBlankType	.*[^\s].*	-	-
notification/ notificationCode	SimpleText100NotBlankType	.*[^\s].*	-	-
notification/notificationText	SimpleText1024NotBlankType	.*[^\s].*	-	-

Leírás és kapcsolódó követelmények

1) A funcCode a szerver által adott státusz a requestben szereplő művelet végrehajtására. Az értelmezése az üzleti operációk szerint eltérő lehet, mindig a teljes válasszal együtt értelmezendő!
2) Az errorCode akkor kerül visszaadásra, ha a funcCode értéke ERROR volt. A hiba egyedi kódját tartalmazza, a kliens oldalon ezt a taget lehet használni a hibaüzenet mappelésére. Az errorCode értékkészletéről a „**HIBAKEZELÉS”** fejezetben lévő hibakód táblázat tájékoztat.
3) A message opcionális szöveges üzenet, ami a funcCode-ot vagy az errorCode-ot kíséri. Az emberi megértést segíti olvasható üzenet közvetítésével.
4) A notification csomópontot a NAV a jövőben egyéb, API hívásokban értelmezhető tájékoztató üzenetek közlésére fogja használni, kulcs-érték struktúrában.

2.4 A requestSignature számítása

A requestSignature az interfész-autentikáció egyik fő eleme. A szerepe, hogy illetéktelenek ne tudjanak a rendszerben változtatásokat végrehajtani. A hash értéket a szerver oldal minden operáció minden kérésénél ellenőrzi, és csak akkor hajtja végre a műveletet, ha a tárolt és kapott adatokból a helyes érték ténylegesen előállítható. A requestSignature-nek minden esetben **nagybetűs** lenyomatnak kell lennie.

A requestSignature tag a típusából adódóan rendelkezik egy kötelező attribútummal: cryptoType néven.

Az eÁFA esetén egyetlen elfogadott értéke: **SHA3-512**

2.4.1 Számítás fájlfeltöltés esetén

A fájlfeltöltést végző operációk esetén, amelyek multipart/form-data média típussal rendelkeznek, a requestSignature-t az alábbi módszer alapján lehet kiszámítani:

A requestSignature egy parciális hitelesítésből, illetve a multipart/form-data kérésben szereplő feltöltendő fájl (octet-stream) SHA3-512 lenyomatának értékének összefűzéséből és egy további SHA3-512 hash műveletből számítódik. A parciális hitelesítést a következő értékek összefűzéséből lehet megállapítani:

- a requestId értéke
- a timestamp tag értéke YYYYMMDDhhmmss maszkkal, UTC időben
- a technikai felhasználó aláírókulcsának literál értéke

Az összefűzéskor a timestamp maszkoláshoz ki kell venni a dátum- és időpontszeparátorokat, valamint az időzónát.

- Az eÁFA esetén multipart/form-data média típussal rendelkeznek és a fent leírt metódus alapján számolandó a requestSignature értéke az alábbi operációk esetén:
    - manageDeclarationPartition – partíció feltöltés
    - manageAttachmentUpload – melléklet feltöltés

Egy fiktív példa request adatai:

- requestId = TSTKFT1222564
- timestamp = 2017-12-30T18:25:45.000Z
    - maszkolással: 20171230182545
- a technikai felhasználó aláírókulcsa = ce-8f5e-215119fa7dd621DLMRHRLH2S
- a parciális hitelesítés értéke = TSTKFT12225642017123018254 5ce-8f5e-215119fa7dd621DLMRHRLH2S

- a kérésben szereplő octet-stream part nagybetűs SHA3-512 lenyomata =
    - 797EB337CB3FD673976F67DE36230DFEEB3A7BC62F68423DEB3607BB211EED7E57E8515A5B8C865B97799E16961EE83FE13D5A82A4951ADF4BB42C779832883B

A teljes requestSignature alapja így:

TSTKFT12225642017123018254 5ce-8f5e-215119fa7dd621DLMRHRLH2S797EB337CB3FD673976F67DE36230DFEEB3A7BC62F68423DEB3607BB211EED7E57E8515A5B8C865B97799E16961EE83FE13D5A82A4951ADF4BB42C779832883B

A requestSignature értéke SHA3-512 hashelést és nagybetűsítést követően az alábbi lesz:

- BBC670463D11CFE8428F492807CA9086243B13015DA41605E077830EC37459543DE1C0965C2BD1A9D8811FAFAED0D465107A93D8EA0E9BBC2ECB8DCA18FB2F17

2.4.2 Számítás egyéb operáció esetén

A manageDeclarationPartition és manageAttachmentUpload operáció kivételével az eÁFA API-jához intézett kérések esetében minden más operációban – mivel ezekben nem merül fel fájlfeltöltés – a requestSignature egyenlő a parciális hitelesítés SHA3-512 hash értékével, amelyet a következő értékek összefűzéséből lehet megállapítani:

- a requestId értéke
- a timestamp tag értéke yyyyMMddHHmmss maszkkal, UTC időben
- a technikai felhasználó aláírókulcsának literál értéke

Az így és sorrendben konkatenált string nagybetűs SHA3-512 hash eredménye lesz a requestSignature értéke.

2.5 Általános technikai adatok

A szolgáltatásokra HTTP POST metódussal kell a body-ban a megfelelő XML vagy multipart kérést elküldeni, melyre a szerver XML-t vagy multipart/form-data választ ad vissza. Az operációk többségében XML request-response média típussal rendelkeznek, az ettől való eltérésekről részletesen az adott szakrendszer interfészspecifikációjában tájékozódhat.

A kérésben az elvégzendő műveletet a hívó a megfelelő endpoint címzésével és a megfelelő struktúrájú kérés összeállításával definiálja. A kérés helyességétől függően a szerver vagy üzleti XML választ, vagy csupán standard HTTP választ ad vissza.

Az eÁFA esetében

- Az általános elemek definíciója a common.xsd-ben, illetve az earBase.xsd-ben található. A kommunikációhoz használt elemek definíciója az earApi sémaleíróban, a bevallások üzleti modellje és elemei definíciója az earData sémaleíróban található.
- Az XML request alól kivételt képeznek a fájlfeltöltést végző operációk, amelyeket multipart/form-data média típussal szükséges küldeni: ManageDeclarationPartition, ManageAttachmentUpload
- Az XML response alól pedig kivétel a QueryDeclarationData operáció, amely multipart/form-data válasszal is rendelkezik.

2.5.1 HTTP fejlécek

A kérésben a következő HTTP fejléc mezőket kötelező megadni:

- content-type=application/xml
- accept=application/xml

Ez alól kivételt képeznek multipart/form-data request-tel és / vagy response-zal rendelkező operációk.

Az adatbázisba mentés és a válasz a kérésben megadott encodingtól függetlenül mindig UTF-8 lesz, ezért javasolt a kérésben is ennek a kódolásnak a használata.

2.5.2 HTTP státuszkódok

A szolgáltatás a hívónak helyes kérés esetén minden esetben HTTP 200-as választ ad. Ez nem feltétlenül jelzi, hogy a megfogalmazott kérés tartalmán az üzleti végrehajtás sikeresen lefutott, csak azt, hogy a kérés informatikai tekintetben jól formázott volt, a hívott erőforrás el tudta olvasni, be tudta fogadni. Mivel a szolgáltatás által kezelt hibakódok fel vannak mappelve, így a visszaadott hibakód is sikeres válasznak számít. Tehát egy HTTP 200-as válaszban is lehet hibakódokat tartalmazó üzenet.

A helytelen kérés vagy egyéb technikai hiba esetén visszaadott eredményekről a „**Hibakezelés”** fejezetben lévő hibakód táblázat tájékoztat.

2.5.3 Méretkorlát

A szolgáltatásnak küldött HTTP POST bodyban szereplő XML mérete nem fájlfeltöltés művelet esetén nem haladhatja meg a 10 megabájtot. Ez alól kivételt jelentenek a multipart/form-data content-type-pal rendelkező fájlfeltöltést végző műveletek.

- A fájlfeltöltés során, az érvényben lévő limitről az eÁFA interfészspecifikációjában tájékozódhat.

2.5.4 Válaszidő, timeout

A szerver jellemzően 200 ms alatti válaszidőkkel szolgál ki. A szinkronhívások blokkoló timeout értéke 5000 ms. Kérjük, hogy kliens oldalon a fenti értéket meghaladó válaszidőt kezeljék csak időtúllépésként! Az abszolút timeout értéke 60 sec. Ha egy műveletre nem érkezik válasz a 60 másodperces timeout miatt, még nem jelenti a művelet sikertelenségét.

2.5.5 Szerveróra, NTP

A szerver az időbeállításokat egy zárt, a külvilág számára nem hozzáférhető NTP szervertől kapja. Kliens oldalon a szerveridőhöz szinkronizálás nem követelmény, azonban opcionálisan a következő időszinkronizáció lehetséges: http://www.pool.ntp.org/zone/hu (a csatlakozáshoz NTP kliensre van szükség).

2.5.6 Helyi idő konvertálása UTC időre

A helyes kliensoldali requestSignature előállításához a helyi időt UTC időre kell konvertálni. Ez úgy tehető meg, hogy a kliensnél érvényes időzóna szerinti helyi idő értékéhez hozzá kell adni vagy ki kell vonni annyi egész órát, amennyivel az időzóna az UTC középidőtől eltér. Amelyik időzónában van téli/nyári időszámítás, ott a kivonásnál/összeadásnál erre is figyelni kell.

2.5.7 Karbantartási mód

Karbantartási módban a „**Hibakezelés”** fejezetben jelzett szabványos HTTP 503 státuszkód kerül visszaadásra.

2.5.8 Verziókezelés

A szolgáltatás szempontjából a verziót az URL, míg az üzleti adatmodell szempontjából a verziót a HTTP body-ban megadott requestVersion tag értéke definiálja.

Főverziónak azon verziókat nevezzük, amelyekben az üzleti adatmodell visszafelé kompatibilitása az egyes verziók között nem biztosítható. Kisverziónak azok a verziók számítanak, amelyekben adott főverzión belül az üzleti adatok kompatibilisek maradnak.

Új főverzióhoz minden esetben új URL és új XML namespace tartozik. A kisverziók öröklik annak a főverziónak az URL és namespace adatait, amelynek a részét képezik.

2.5.9 Karakterkonverzió

A beküldött adatokon szerver oldali karakterkonverzió nem történik semmilyen esetben sem.

2.5.10 Forgalomkorlátozás

A NAV - mivel az eÁFA M2M szolgáltatásként jelenik meg - az interfészspecifikációtól lényegesen eltérő és a rendszer működését zavaró vagy akadályozó kommunikáció megakadályozása érdekében a jövőben rate limiting megoldást vezethet be. A rate limiting azt jelenti, hogy a szerver oldali erőforrások védelmének érdekében az API Gateway képes lesz limitálni az adózónként adott idő alatt beküldhető kérések számát, és amennyiben egy adózó túllépi a limitben meghatározott kérések számát, akkor a HTTP szabványnak megfelelő HTTP 429 Too Many Request hibakóddal elutasításra kerülnek a kérései. A limit túllépése esetén a kérések forgalmazása az időablak leteltét követően folytatható.

A visszaadott hibáról bővebben a „**HIBAKEZELÉS**” fejezetben tájékozódhat.

3 HIBAKEZELÉS

A szolgáltatás egy közös, a szolgáltatás oldalán enumerált értékkészletből vett eredmény és hibakód listával működik. Az eredménykódoktól eltérően a hibakódok szándékosan nem jelennek meg a sémaleíró enumerációiban, hogy azok esetleges változása vagy bővülése ne keletkeztessen implementációs függőséget a kliensek oldalán. Az eredménykódok a BasicResultType node funcCode tagjában, míg a hibakódok az errorCode tagban kerülhetnek visszaadásra a válaszüzenetben. A visszakapott funcCode értékeket a hívott üzleti folyamatnak megfelelően kell értelmezni.

3.1 Általános hibakódok

3.1.1 GeneralExceptionResponseType

A szolgáltatás minden operációjában a technikailag feldolgozhatatlan üzenetre (rosszul formázott XML, helytelen namespace, vagy helytelen context root) a GeneralExceptionResponse hibatípus kerül visszaadásra.

A típus a BasicResultType-ot terjeszti ki, az abban foglalt elemeken kívül más elemet nem tartalmaz.

A notification listatípus a sémaleíróhoz tartozó minden sémasértést tételesen tartalmaz, ha a kérésben volt legalább 1 nem sémavalid tag. Ilyen esetben a notificationCode = SCHEMA_VIOLATION lesz.

3.1.2 GeneralErrorResponseType

A szolgáltatás minden operációjának általános hibatípus üzenetét a GeneralErrorResponseType implementálja. A típus a GeneralErrorHeaderResponseType-ot terjeszti ki, így az abban foglalt elemeken kívül egy technikai validációs listatípust tartalmaz.

A GeneralErrorResponseType felépítése:

Tag	Típus	Kötelező	Tartalma
validationResultCode	xs:string	Igen	A technikai validáció eredménye
validationErrorCode	xs:string	Nem	A technikai validáció hibakódja
message	xs:string	Nem	A technikai validáció eredményéhez tartozó szöveges üzenet

Facetek és leírók

Tag	SimpleType	Pattern	Enum	Default
validationResultCode	TechnicalResultCodeType	-	CRITICAL ERROR	-
validationErrorCode	SimpleText100NotBlankType	.*[^\s].*	-	-
message	SimpleText1024NotBlankType	.*[^\s].*	-	-

Leírás és kapcsolódó követelmények

1) Ha a technicalValidationMessages tag képződik, akkor a validationResultCode jelenleg csak ERROR értéket vehet fel (a CRITICAL ebben a típusban fenntartott érték az esetleges jövőbeni validációk számára).
2) A validationErrorCode tag a hibatípus kódját tartalmazza.
3) A message tag a technikai validáción fennakadt hibás tag nevét, értékét, egyéb esetben pedig a validationErrorCode taghoz tartozó szöveges hibaüzenetet tartalmazza.

A következő fejezetben leírt technikai hibakódokat a rendszer minden esetben vagy a GeneralExceptionResponse, vagy a GeneralErrorResponse válaszelemben adja vissza, a „**Hibakezelés”** fejezetben leírt response elemek csak és kizárólag akkor képződnek, ha a szinkron feldolgozás üzletileg és technikailag is sikeres volt! Így a HTTP response body-ban visszakapott valamely error tag mindig valamilyen hibát fog jelezni.

3.2 Technikai hibakódok

A szinkronhívások errorCode értékkészletét a következő táblázat tartalmazza.

Technikai és autentikációs hibák

#	HTTP válasz	Response body	funcCode	errorCode	requestVersion
1	HTTP 404 NOT_FOUND	-	-	-	1.0
2	HTTP 405 METHOD_NOT_ALLOWED	GeneralExceptionResponse XML tag	ERROR	NOT_ALLOWED_EXCEPTION	1.0
3	HTTP 400 BAD_REQUEST	GeneralExceptionResponse XML tag	ERROR	INVALID_REQUEST	1.0
4	HTTP 400 BAD_REQUEST	GeneralExceptionResponse XML tag	ERROR	INVALID_REQUEST	1.0
5	HTTP 401 UNAUTHORIZED	GeneralErrorResponse XML tag	ERROR	INVALID_SECURITY_USER	1.0
6	HTTP 500 INTERNAL_SERVER_ERROR	GeneralErrorResponse XML tag	ERROR	NOT_REGISTERED_CUSTOMER	1.0
7	HTTP 500 INTERNAL_SERVER_ERROR	GeneralErrorResponse XML tag	ERROR	INVALID_CUSTOMER	1.0
8	HTTP 500 INTERNAL_SERVER_ERROR	GeneralErrorResponse XML tag	ERROR	INVALID_USER_RELATION	1.0
9	HTTP 500 INTERNAL_SERVER_ERROR	GeneralErrorResponse XML tag	ERROR	FORBIDDEN	1.0
10	HTTP 400 BAD_REQUEST	GeneralErrorResponse XML tag	ERROR	REQUEST_ID_NOT_UNIQUE	1.0
11	HTTP 400 BAD_REQUEST	GeneralErrorResponse XML tag	ERROR	INVALID_REQUEST_SIGNATURE	1.0
12	HTTP 503 SERVICE_UNAVAILABLE	GeneralErrorResponse XML tag	ERROR	SERVICE_UNAVAILABLE	1.0
13	HTTP 400 BAD_REQUEST	GeneralErrorResponse XML tag	ERROR	INVALID_TIMESTAMP	1.0
14	HTTP 400 BAD_REQUEST	GeneralErrorResponse XML tag	ERROR	INVALID_PASSWORD_HASH_CRYPTO	1.0
15	HTTP 400 BAD_REQUEST	GeneralErrorResponse XML tag	ERROR	INVALID_REQUEST_SIGNATURE_HASH_CRYPTO	1.0
16	HTTP 400 BAD_REQUEST	GeneralErrorResponse XML tag	ERROR	INVALID_REQUEST_VERSION	1.0

17	HTTP 400 BAD_REQUEST	GeneralErrorResponse XML tag	ERROR	INVALID_HEADER_VERSION	1.0
18	HTTP 415 UNSUPPORTED_MEDIA_TYPE	GeneralExceptionResponse XML tag	ERROR	INVALID_REQUEST	1.0
19	HTTP 416 NOT_ACCEPTABLE	GeneralExceptionResponse XML tag	ERROR	INVALID_REQUEST	1.0
20 *	HTTP 400 BAD_REQUEST	GeneralErrorResponse XML tag	ERROR	REQUEST_VERSION_NOT_ALLOWED	2.0
21 *	HTTP 429 TOO_MANY_REQUESTS	GeneralErrorResponse XML tag	ERROR	TOO_MANY_REQUESTS	1.0

*Jövőben bevezetendő validációk hibakódjai.

Hibaeset, teendők

#	errorCode	Hiba oka	Teendő
1	-	hibás a szolgáltatás endpoint a kérésben	Az egyes környezetekben megcímzendő endpointokról az eÁFA_M2M_rendszer_interfész_specifikáció „Környezetek elérhetőségei” című fejezet tartalmaz információkat, ellenőrizni kell az URL-t.
2	NOT_ALLOWED_EXCEPTION	hibás a HTTP metódus a kérésben	Az URL helyes, de a HTTP metódus nem POST. Az interfész minden operációját POST metódussal kell küldeni!
3	INVALID_REQUEST	rosszul formázott az XML a request body-ban	A szintaktikailag helytelen XML-üzenetet az XML-szabvány szerint tilos XML-nek tekinteni és feldolgozni, javítani kell.
4	INVALID_REQUEST	nem sémavalid XML a request body-ban	A beküldött XML - válaszban felsorolt - elemei sértik az earApi.xsd megkötéseit, javítani kell. (A hibakód továbbá akkor is jelentkezhet, ha egy request nem a megfelelő operáció végpontjára van beküldve)
5	INVALID_SECURITY_USER	a kérésben hibás login + passwordHash pár	Számos esetben jelentkezhet a hibaüzenet. Lehetséges okok: a megadott login névvel nem létezik felhasználó, vagy nem helyes a jelszava, vagy a login + passwordHash pár szemantikailag helyes, de a jelszóhash rosszul kerül kiszámításra a kliens oldalán. Meg kell győződni az adatok és a hashelés helyességéről, szükség esetén fel kell venni a kapcsolatot a technikai felhasználót birtokló adózóval.
6	NOT_REGISTERED_CUSTOMER	a kérésben megadott adózó nem található	A user tagban megadott adószámmal nem található a rendszert használó adózó.
7	INVALID_CUSTOMER	a kérésben hibás a taxNumber	A user tagban megadott adószám vagy nem létezik, vagy a státusza nem engedi a műveletek elvégzését. Meg kell győződni az adatok helyességéről, szükség esetén fel kell venni a kapcsolatot az érintett adózóval.
8	INVALID_USER_RELATION	a kérésben szereplő entitások között nincs kapcsolat	A megadott adószámhoz nem tartozik a megadott login névvel technikai felhasználó, vagy a felhasználó státusza már nem engedélyezi a művelet elvégzését. Meg kell győződni az adatok helyességéről, szükség esetén fel kell venni a kapcsolatot az érintett adózóval.
9	FORBIDDEN	a kérésben szereplő technikai felhasználó nem jogosult az endpoint szolgáltatását hívni	A technikai felhasználók jogosultságait az adózó elsődleges felhasználói osztják ki. Szükség esetén fel kell venni a kapcsolatot az érintett adózóval.

10	REQUEST_ID_NOT_UNIQUE	a kérésben szereplő requestId nem egyedi	A kérésben szereplő adószámra a megadott requestId-t már felhasználták. Az egyediség miatt új Id megadása szükséges.
11	INVALID_REQUEST_SIGNATURE	a kérésben szereplő requestSignature hibás	A szerver oldalon elvégzett requestSignature számítás nem egyezik meg a kliens oldalon kiszámított értékkel. A számítás módjáról lásd a „requestSignature számítása” című fejezetet.
12	SERVICE_UNAVAILABLE	karbantartás van folyamatban	A hívott operáció karbantartás miatt átmenetileg nem szolgál ki. Továbbá a hiba akkor is visszaadható, amennyiben a Gateway vagy a mögötte lévő szakrendszer karbantartása van folyamatban. Kísérje figyelemmel a felületen elhelyezett tájékoztatót és ismételje meg a kérést egy későbbi időpontban!
13	INVALID_TIMESTAMP	kérésben megadott időbélyeg 1 napon kívül esik	A hibaüzenet az összes autentikációt tartalmazó operációban visszaadható akkor, ha a header csomópontban megadott timestamp értéke a szerveridő +- 1 napos intervallumán kívül esik.
14	INVALID_PASSWORD_HASH_CRYPTO_TYPE	technikai felhasználó jelszavának hash-képző algoritmusa helytelen	A hibaüzenet akkor adható vissza, ha a user/passwordHash/cryptoType értéke nem SHA-512.
15	INVALID_REQUEST_SIGNATURE_HASH_CRYPTO	kérés aláírásának hash-képző algoritmusa helytelen	A hibaüzenet akkor adható vissza, ha a user/ requestSignature/cryptoType értéke nem SHA3-512.
16	INVALID_REQUEST_VERSION	kérés verziója érvénytelen	A hibaüzenet akkor adható vissza, ha a header/requestVersion értéke nem 1.0.
17	INVALID_HEADER_VERSION	header verziója érvénytelen	A hibaüzenet akkor adható vissza, ha a /header/headerVersion értéke meg van adva és nem 1.0.
18	INVALID_REQUEST	a kérés nem a megfelelő Content-type headerrel került beküldésre	A kérés nem az operációnak megfelelő médiatípussal került beküldésre. Javítani kell a kérés média típusát és / vagy a Content-type header értékét.
19	INVALID_REQUEST	a kérés nem a megfelelő Accept headerrel került beküldésre	A kérésben szereplő Accept headernek megfelelő választ a szerver nem képes visszaadni. Javítani kell az Accept header értékét.
20	REQUEST_VERSION_NOT_ALLOWED	a kérésben szereplő requestVersion tag értéke már nem megengedett	A kérés requestVersion értéke már nem támogatott verzió. Ez akkor fordulhat elő, amikor valamilyen szabály változása miatt át kell állni az interfész egy újabb verziójára úgy, hogy adott időponttól kezdve a korábbi verzió már nem használható. Javítani kell!

21	TOO_MANY_REQUESTS	túl sok kérés került beküldésre rövid idő alatt	A hibaüzenet akkor adható vissza, ha az adózó adott időablakon belül túl sok kérést küldött be. Az erőforrások védelmének érdekében ilyenkor a kérések visszautasításra kerülnek. Az időablak letelte után a kérések forgalmazása folytatható.

Feldolgozási hibák

#	HTTP válasz	Response body	funcCode	errorCode	requestVersion
1	HTTP 500 INTERNAL_SERVER_ERROR	GeneralErrorResponse	ERROR	OPERATION_FAILED	1.0

Hibaeset, teendők

#	errorCode	Hiba oka	Teendő
1	OPERATION_FAILED	váratlan feldolgozási hiba	Szerver oldali általános hibakód. A szóban forgó hiba csak szinkronhívásoknál jelentkezhet, ilyenkor a műveletet kis idő elteltével meg kell ismételni. Ha az éles rendszerben többszöri próbálkozásra sem sikerül a művelet, fel kell venni a kapcsolatot a NAV helpdeskkel, azonban célszerű előtte tájékozódni, hogy a portál oldalon nincs-e üzemszünettel, üzemzavarral kapcsolatos tájékoztatás. Felhívjuk a figyelmet, hogy a felhasználói teszt rendszerben nincs garantált rendelkezésre állás, ezért kérjük, hogy a tesztrendszer hibáit ne jelentsék be!

4 HELPDESK ÉS TECHNIKAI SEGÍTSÉGNYÚJTÁS

A fejezet a hibaelhárításhoz és további segítség igénybevételéhez nyújt támpontokat.

4.1 Implementáció ellenőrzését segítő eszközök

Az egyes kódolások, hashelések helyességének ellenőrzéséhez, valamint az XML-formátum általános szintaxisának ellenőrzéséhez a következő weboldalakon található információ.

Aktuális UTC középidő: https://www.timeanddate.com/worldclock/timezone/utc

BASE64 online encode/decode: https://www.base64decode.org/

CRC számítás online: https://www.functions-online.com/crc32.html (az online konverterek jellemzően hexadecimális értékben számolnak, ezek is használhatók, de ekkor az outputot felhasználás előtt át kell váltani decimálisra)

SHA-512 online encode: http://www.convertstring.com/Hash/SHA512

SHA3-512 online encode: https://codebeautify.org/sha3-512-hash-generator

XML jól formázottság és séma konformitás ellenőrző online: https://www.xmlvalidation.com/

Regex ellenőrzés: https://regex101.com/

XML szintaxis információk: https://www.w3schools.com/xml/xml_syntax.asp

XML-séma információk: https://www.w3schools.com/xml/schema_intro.asp

4.2 Helpdesk elérhetőség

Az eÁFA rendszerben felmerülő hibák megoldására és kérdések megválaszolására két különálló helpdesk vehető igénybe. Minden éles rendszerrel kapcsolatos kérdéssel és problémával a ***https://nav.gov.hu/ugyfeliranytu/keressen_minket/levelkuldes*** funkción keresztül „***eÁFA, informatikai problémák***” tárggyal küldött megkereséssel lehet fordulni. A levélküldő űrlap angol nyelven is elérhető.

Kizárólag a teszt rendszerre vonatkozó, fejlesztőknek szóló technikai segítségnyújtás az ***init.eafa_teszt_support@nav.gov.hu*** címre küldött e-mailen keresztül vehető igénybe.

Kérjük, hogy ha az interfész használatához kapcsolódóan technikai segítséget igényel, a megkeresésben a teljes HTTP request (header és body) tartalmát és a beküldés pontos időpontját tüntesse fel!

4.3 Github elérhetőség

4.3.1 Common repository

A repository elérése: https://github.com/nav-gov-hu/Common

A tárhely abból a célból jött létre, hogy az atomi típusokat, üzleti katalógus jellegű elemeket, valamint a generikus API kommunikációt megvalósító típusokat egy külön, közös XSD-ben (common.xsd) lehessen verziókezelni. Ez lehetővé teszi azt, hogy ezeket a séma elemeket több NAV-os projekt is felhasználhassa, ezáltal az API kommunikáció egységesíthető.

4.3.2 eÁFA repository

eÁFA repository elérése: https://github.com/nav-gov-hu/eVAT