

Dokumentacja API Bazy Wiedzy

Historia dokumentacji

Data	Opis
20-12-2013	Utworzenie dokumentu i opisanie aktualnych metod
10-01-2014	Aktualizacja dokumentacji o nowe funkcje
13-01-2014	Opisanie funkcji pobierającej szczegóły procedury
22-01-2014	Aktualizacja dokumentacji dla metody getPkdWithHints
22-01-2014	Opisanie nowej metody getProcedureCountByPkdid
24-01-2014	Zmiana metody getProcedureCountByPkdid na getProcedureCountByPkdCode
27-01-2014	getPKDWithHints zaktualizowano tabele kodów błędów.
28-01-2014	Dodanie metody linkSessionWithUser związanej z sytuacjami życiowymi Modyfikacja parametrów wyjściowych dla funkcji getListToPerformElectronicProcedure
21-02-2014	Dodanie department_id dla ApiProcedures dla metody getList, Dodanie informacji o znaczniku w aktualnych czynnościach
03-03-2014	Dodanie pól zwracanych przez metodę getDetails ApiProcedures: name_abbreviation, keywords, keywords_additional Dodanie rodzaju grupy do parametrów wejściowych metody getList ApiInstitutions: UMG (urzędy miast i gmin)
06-03-2014	Sortowanie listy procedur po dacie publikacji i dacie modyfikacji Dodanie pola modification_date do wyników getList ApiProcedures
02-04-2014	Zaktualizowanie kodów błędów dla voteOnLifeSituation. Dodanie informacji o parametrze Content-Type przy wysyłaniu metodą POST.
07-04-2014	Dodanie metody revertToStage umożliwiającej cofnięcie sytuacji życiowej do wybranego etapu, aktualizacja parametrów wyjściowych dla getList i getDescription dla Sytuacji życiowych
10-04-2014	Dodanie nowych pól do metod getDescription i getList dla API LifeSituation
25-04-2014	Dodanie nowego kodu błędu do funkcji startLifeSituation
06-05-2014	ApiLifesituation GetList dodanie możliwości sortowania po name,category, importance
14-05-2014	ApiComons – zaktualizowano listę kodów odpowiedzi dla metody getPDF
28-05-2014	ApiLifeSituations -> getDescription - aktualizacja listy kodów związanych z funkcjonalnością uruchomienia testowego sytuacji życiowej (ERR_5019, ERR_5020)
05-06-2014	Aktualizacja dla ApiLifeSituation dla metody startLifeSituation z funkcjonalnością uruchomienia testowego sytuacji życiowej (ERR_5019, ERR_5020)
18-06-2014	ApiProcedures GetDetails – dodanie nazwy instytucji do listy instytucji wykonujących procedurę, Dla ApiInstitution dla metody getList dodano nowe pole voivodeship w odpowiedzi na zgłoszenie 0000091
01-07-2014	Dla metody ApiProcedures getList poprawiono filtrację wyniku dla parametrów electronic_only i institution_id. Struktura odpowiedzi nie uległa zmianie
23-07-2014	Dla ApiProcedures dla metody getDetail uzupełniono wzór odpowiedzi o elementy electronic_service_count, electronic_services
20-10-2014	Zmieniono sposób odwoływania do procedur (poprzez alias). Zmieniono odpowiedź dla getDetails procedury
22-10-2014	Aktualizacja getDetails procedury – dodatkowe opisy, zmiana update_date na confirmation_date, dodanie oprócz opisów kodów słownikowych również wartości kodów (code, code_description)
27-10-2014	Aktualizacja getDetails instytucji – dodatkowe sekcje: InstitutionPkd oraz InstitutionTerritoryRange
28-10-2014	Zmiana odpowiedzi API dla metody ApiLifeSituations->getPublication (zostały usunięte pola pub_tytul, pub_alias, fulltext; pole alias zmieniło strukturę z postaci [string] na [int-string]) Dodanie bloku „forms” – form dokumentów w szczegółach dokumentu na kroku – ApiProcedures, getDetails
05-11-2014	Dokumentacja ApiPnaInstitutions – metody getHintsByCity oraz getListByLocation

06-11-2014	Poprawiono strukturę statutory_duration na procedurze oraz na kroku, poprawiono kody typów aktów prawnych
06-11-2014	Dokumentacji ApiPnaAddress metoda getListByPostCode,
07-11-2014	Dokumentacja ApiPnaAddress metoda getStreetHints, Poprawka dla opisu procedury->procedure_steps->details->deadline code/ code_description
14.11.2014	Poprawa ApiInstitutions->getDetails. Zostało usunięte pole contact z element InstitutionDepartments[DepartmentDetails]
19.11.2014	Zmodyfikowano strukturę wyjściową dla ApiProcedures -> getDetails. Zmieniono strukturę eservices dla procedury oraz dla kroków Zmieniono strukturę opłat na kroku
02-12-2014	Uzupełniono opisy dla ApiLifeSituations->startLifeSituation
05-12-2014	Dodano informacje o planowanych zmianach w ramach sprintu 27 (19-12-2014) <ul style="list-style-type: none"> • Dodano do struktury wyników: ApiLifeSituations: getPublication -> „noindex” • Dodano do parametru wejściowego dodatkowy element słownikowy „INSTYTUCJA” ApiOpinions: saveOpinionForContent -> „group” • Zmieniono strukturę wynikową dla ApiInstitutions: getDetails dodając: „city”, „number”, „post_code”, „street”, „general_eservice” • Zmieniono strukturę wynikową dla ApiInstitutions: getList dodając gałąź „departments” • Dodano parametr wejściowy : ApiInstitutions: getList -> „with_departments” • Dodano do struktury wyników: ApiProcedures: getDetails -> „is_expired”, „institutions_implementing_hint” • Dodano parametr wejściowy: ApiPnaInstitutions: getHintsByCity -> „esp_only” • Dodano parametr wejściowy: ApiPnaInstitutions: getListByLocation -> „esp_only” • Dodano nową grupę API: ApiPublications, a w niej metody getList i getDetails
12-12-2014	Dodatek do Sprint 27 <ul style="list-style-type: none"> • Zmieniono strukturę wynikową dla ApiInstitutions: getDetails dodając: „esp”, „epuap_id”
18-12-2014	Ujednolicono struktura aliasu testowego dla publikacji w ApiPublications i ApiLifeSituations.
29-12-2014	Uzupełniono ApiPnaInstitutions: getListByLocation o informacje o wydziale instytucji.
16-01-2015	<ul style="list-style-type: none"> • Dodano do struktury wyników: ApiProcedures: getDetails -> procedure_general pole „purpose” (cel procedury) • Dodano metodę ApiProcedures: getListForRespAuthority
29-01-2015	<ul style="list-style-type: none"> • Dodano parametr wejściowy „range_teryt” do ApiProcedures: getList
06-02-2015	<ul style="list-style-type: none"> • Dodano ApiInstitutions: getDepartmentDetails
11-02-2015	<ul style="list-style-type: none"> • Usunięto szczegóły wydziału z ApiInstitutions: getDetails
13-02-2015	<ul style="list-style-type: none"> • Uzupełniono ApiLifeSituations->startLifeSituation i ApiLifeSituations->getDescription o dane katalogów tematycznych
17-02-2015	<ul style="list-style-type: none"> • Uzupełniono parametr „group” metody ApiInstitutions->getList o element „POZOSTALE”
09-03-2015	<ul style="list-style-type: none"> • Uzupełniono ApiLifeSituations->getDescription o dane publikacji powiązanych
11-03-2015	<ul style="list-style-type: none"> • Dodano parameter epk_only do ApiProcedures/getList
17-03-2015	<ul style="list-style-type: none"> • Uzupełniono dane opłat o kod waluty w ApiProcedures->getDetails
19-03-2015	<ul style="list-style-type: none"> • Uzupełniono dane publikacji o powiązane poradniki w ApiPublications->getDetails
24-03-2015	<ul style="list-style-type: none"> • Dodano kod kontekstu (context_code) dla dokumentu do ApiProcedures->getDetails
25-03-2015	<ul style="list-style-type: none"> • Gdy podana instytucja nie znajduje się w obszarze podanego zasięgu terytorialnego jest zgłaszany odpowiedni wyjątek dla ApiProcedures->getDetails
26-03-2015	<ul style="list-style-type: none"> • Dodanie danych instytucji akceptującej do ApiProcedures->getDetails • Wyszukiwanie w getListForRespAuthority również po instytucji akceptującej
01-04-2015	<ul style="list-style-type: none"> • Usunięcie ApiLifeSituations->getPublication, zamienionej na ApiPublications->getDetails
09-04-2015	<ul style="list-style-type: none"> • Uporządkowanie opisu pól time_unit w ApiPrcedures->getDetails (wartości słownikowe po polsku) • Dodanie ApiInstitutions->getDetails->InstitutionDetails->responsible_procedures_count
13-04-2015	<ul style="list-style-type: none"> • Dodanie informacji o procedurach powiązanych z procedurą poprzez domnent wejściowy w ApiPrcedures->getDetails jako related_procedures dla każdego dokumentu
17-04-2015	<ul style="list-style-type: none"> • Zmiana struktury ApiPrcedures->getDetails, umieszczenie pól publication_text i accepting_institution poza obiektami.
30-04-2015	<ul style="list-style-type: none"> • Wydłużono pole wyszukiwania procedur do 300 znaków

06-05-2015	<ul style="list-style-type: none"> Filtrowanie zawodów parametrem <code>with_assigned_procedures_only</code>
19-05-2015	<ul style="list-style-type: none"> Zmiana nazwy pola w odpowiedzi "RowCount na "rows_count" dla: <code>ApiLifeSituations->getList</code>, <code>ApiPublications->getList</code>
27-05-2015	<ul style="list-style-type: none"> Dodanie właściwości "business_lifecycle_name" i "business_lifecycle" do obszaru "references" w <code>ApiProcedures->getDetails</code>
22-06-2015	<ul style="list-style-type: none"> Usunięto metody <code>ApiCommons: searchPKD</code>, <code>getProcedureCountByPkdCode</code>, <code>getPKDWithHints</code> Usunięto metodę <code>ApiProcedures: getHintsQuestionPKD</code> Dodano metody <code>ApiCommons: getPkdList</code>, <code>getPkdHints</code> Dodano kod błędu ERR_0035 do metod: <code>ApiInstitutions->getList</code>, <code>ApiLifeSituations->getList</code>, <code>ApiProcedures->getElectronicList</code>, <code>ApiProcedures->getList</code>, <code>ApiProcedures->getListForRespAuthority</code>
01-07-2015	<ul style="list-style-type: none"> Rozbudowano metody <code>ApiInstitutions->getDetails</code> i <code>ApiInstitutions->getDepartmentDetails</code> o dane dotyczące systemu kolejkowego
24-07-2015	<ul style="list-style-type: none"> Modyfikacja akronimu sytuacji życiowych. Pole acronym może przyjmować wartości „id-akronim” lub „id” lub „akronim”. Wartość akronimu jest zwracana w odpowiedziach w postaci „id- akronim”
05-08-2015	<ul style="list-style-type: none"> Modyfikacja danych wejściowych metody <code>ApiOpinions/saveOpinionForContent</code>
21-08-2015	<ul style="list-style-type: none"> Dodanie pola <code>institution_verification_date</code> w szczegółach procedury
26-08-2015	<ul style="list-style-type: none"> Dla <code>ApiLifeSytuations/GetDescription</code> dodano w odpowiedzi pole acronym
29-09-2015	<ul style="list-style-type: none"> Dodanie kodu teryt do zasięgu terytorialnego wydziałów i instytucji.
30-09-2015	<ul style="list-style-type: none"> Dodanie nowej metody <code>ApiMail/sendEmail</code>
07-10-2015	<ul style="list-style-type: none"> Dodanie parametru PKD w liście Poradników Dodanie <code>anyone_has_euap</code> do szczegółów Procedury
21-10-2015	<ul style="list-style-type: none"> Dodanie pól <code>related_procedures_count</code> i <code>related_life_situations_count</code> do metody <code>api ApiCommons/getPkdList</code> Dodanie powiązanych aktów prawnych do funkcji <code>ApiLifeSituation/getDescription</code>
23-10-2015	<ul style="list-style-type: none"> Dodanie sekcji <code>pkd</code> do metody <code>api ApiLifeSituations/getDescription</code>
04-11-2015	<ul style="list-style-type: none"> Dodanie do sekcji "procedure_general" pola "description_code" zawierającego informacje na temat typu opisu procedury dla <code>ApiProcedures/getDetails</code>
18-11-2015	<ul style="list-style-type: none"> Zmiana wartości pola <code>has_fees</code> w elemencie <code>procedures_general</code> gdy pole <code>description_code = OPIS_SKROCONY</code> dla <code>ApiProcedures/getDetails</code>
15-12-2015	<ul style="list-style-type: none"> Dodanie nowej metody <code>ApiUsers/getDeletedUsers</code>
17-05-2016	<ul style="list-style-type: none"> Nowe wartości słownikowe w polu <code>procedure_general/local_jurisdiction</code>
05-07-2016	<ul style="list-style-type: none"> Dodanie nowej metody <code>ApiMail/getMessages</code>
08-07-2016	<ul style="list-style-type: none"> Zmiany struktury odpowiedzi oraz opcjonalność czasu w <code>ApiMail/getMessages</code>
28-07-2016	<ul style="list-style-type: none"> Zwracany błąd ERR_0021 w metodzie <code>ApiOpinions/saveOpinionForContent</code>, gdy przekazany parametr <code>user_ip</code> nie jest poprawnym adresem IP
16-09-2016	<ul style="list-style-type: none"> Dodanie nowej wartości parametru <code>procedure_execution->eservices->category</code> w <code>ApiProcedures/getDetails - „general_eservice”</code>
04-11-2016	<ul style="list-style-type: none"> Modyfikacja ustawy czas realizacji procedury odpowiedzi dla <code>Statutory_duration - usunięcie opcji "time" i "time_unit"</code>
04-11-2016	<ul style="list-style-type: none"> Wygaszenie pola przebieg alternatywny dla <code>APIProcedures/getDetails</code> Wygaszenie pola rezultat procedury dla <code>APIProcedures/getDetails</code>
18-11-2016	<ul style="list-style-type: none"> Pole miejscowość w obiektach instytucji oraz wydziałów zawiera teraz dodatkowo kraj jeśli jest on inny niż Polska
23-11-2016	<ul style="list-style-type: none"> <code>ApiProcedures/getDetails</code> <ul style="list-style-type: none"> <code>procedure_steps->type_code</code> – zmiana kodów (usunięcie kodu U) <code>procedure_steps->type_description</code> – zmiana tekstów <code>procedure_steps->details->deadline->code</code> – zmiana kodów (usunięcie kodu 'czas') <code>procedure_steps->details->deadline->code_description</code> – zmiana tekstów <code>procedure_steps->details->deadline</code> – usunięcie właściwości 'time' i 'time_unit' <code>procedure_steps->details->participants</code> – właściwość nieobligatoryjna
	<ul style="list-style-type: none"> <code>ApiProcedures/getListForRespAuthority</code> – wartość „status” – usunięto kod Br
21-12-2016	<ul style="list-style-type: none"> <code>ApiProcedures/getDetails</code> <ul style="list-style-type: none"> <code>procedure_general-> local_jurisdiction -> for_entrepreneur</code> – dodano: <ul style="list-style-type: none"> ZASP_KORZ_SRO - ze względu na miejsce korzystania ze środowiska ZASP_ODP_KOM - ze względu na miejsce powstawania odpadów

	<p>komunalnych</p> <ul style="list-style-type: none"> ○ ZASP_MIE_ZAM - ze względu na miejsce zamieszkania • ApiProcedures/getDetails <ul style="list-style-type: none"> procedurę_general-> local_jurisdiction -> for_citizen – dodano: <ul style="list-style-type: none"> ○ ZASO_KORZ_SRO - ze względu na miejsce korzystania ze środowiska ○ ZASO_ODP_KOM - ze względu na miejsce powstawania odpadów komunalnych • ApiProcedures/getDetails <ul style="list-style-type: none"> procedurę_general-> statutory_duration – zmieniono: <ul style="list-style-type: none"> code_description ->null description -> tekst w zależności od pola 'code'
11-01-2017	<p>Modyfikacja i dodanie nowych pól dla metody getDetails dla APIProcedures. Dodanie pól: - procedure_execution->how_to_execute</p> <ul style="list-style-type: none"> • by_institution • by_letter • by_other • by_internet • other_description • barrier_description • eservices <p>przeniesienie gałęzi eservice do nowej gałęzi how_to_execute usunięcie pola z opisu kroku: "is_conditional_step" : < czy krok jest warunkowy (true/false)>,</p>
19-01-2017	<p>Dodanie nowych pól dla metody getCurrentQuestion:</p> <pre>"content" : { "question" : { "answer_type" : <typ odpowiedzi, jednokrotnego lub wielokrotnego wyboru: checkbox/radio></pre> <p>Oraz</p> <pre>"content" : { "answers" : [{ < tablicaodpowiedzi > "is_exclusive": <dodatkowy parametr, który określa czy odpowiedź wyłącza możliwość wyboru innych (może wystąpić 1 na dane pytanie o typie 'checkbox')>: false/true</pre> <p>Dodanie nowych błędów dla metody saveCurrentQuestion:</p> <p>ERR_5024 Dla pytania typu RADIO odpowiedź musi przyjmować jedną wartość ERR_5023 Zaznaczono w odpowiedzi wiele wartości, spośród których jedna posiada atrybut EXCLUSIVE</p>
25-01-2017	<p>Utworzono wersję v2 metody getList:</p> <ol style="list-style-type: none"> 1. ApiPublication, 2. ApiLifeSituation, 3. ApiProcedures. <p>Zmiana polega na dodaniu pola 'short_introduction'.</p> <p>Cel – ujednoczenie sposobu działania wyszukiwarek.</p>
09-02-2017	<p>Zmodyfikowano wersję V3 metody getDetail dla APIProcedures Modyfikacja gałęzi how_to_execute:</p> <pre>how_to_execute : { < w jaki sposób można zrealizować usługę > options: [<lista opcji sposobów realizacji usługi – podzbiór INSTITUTION, MAIL, ONLINE>],</pre>

	<p>additional_description: <Dodatkowy opis który jeśli niepusty to można wykorzystać do opisanie innych sposobów załatwienia sprawy lub barier w powyższych sposobach></p> <pre>eservices : [] }</pre>
28-02-2017	<p>Zmodyfikowano wersję V3 metody getDetails dla ApiProcedures Gałąź procedure_execution: Dodano pole 'room_person'</p> <p>Zmodyfikowano metodę getDetails dla ApiInstitutions Gałąź Institution: Dodano pole 'executive_name'</p> <p>Zmodyfikowano metodę getList dla ApiInstitutions Dodano pole 'executive_name'</p>
01-06-2017	<p>Dodano nowe właściwości miejscowe w sekcji Zasięg realizowania usługi ze względu na właściwość miejscową – ApiProcedures/getDetails for_entrepreneur: ZASP_MIE_INW_BUD; ZASP_MIE_ROZ; ZASP_POL_BUD; ZASP_LOK_ZJA; ZASP_LOK_ZAJ_PDROG; ZASP_POL_NIE; ZASP_POL_DRZE; ZASP_LOK_INST; ZASP_MIE_POL_ZAB; ZASP_MIE_WYZBPR_ODP; ZASP_MIE_IMP_MAS; ZASP_URZ_SKR_PRZE; ZASP_MIE_ODP_CEL for_citizen: ZASO_MIE_INW_BUD; ZASO_MIE_ROZ; ZASO_POL_BUD; ZASO_LOK_ZJA; ZASO_LOK_ZAJ_PDROG; ZASO_POL_NIE; ZASO_POL_DRZE; ZASO_LOK_INST; ZASO_MIE_POL_ZAB; ZASO_MIE_IMP_MAS</p>
07-06-2017	<p>ApiProcedures/getDetails – w przypadku wybrania typu specjalnego instytucji realizującej dla usługi "procedure_execution_type" = CENTRAL, "institutions_implementing_in_range" = null "institutions_implementing_all":{ "units": null LUB [<lista instytucji realizujących>] "types": null },</p>
08-06-2017	<p>ApiProcedures/getSYM - pozwala na wyszukanie prawidłowej miejscowości w oparciu o bazę SIMC</p>
13-06-2017	<p>ApiProcedures/getDetails – dodano wersję V4 metody, która pobiera dodatkowy opcjonalny parametr „sym” i na podstawie terytu lub sym zwraca wartość „localization_name”. Metoda zwraca dodatkowo 3 rodzaje błędów związanych z parametrem „sym”</p>
01-08-2017	<p>Dodano nowe właściwości miejscowe w sekcji Zasięg realizowania usługi ze względu na właściwość miejscową – ApiProcedures/getDetails for_entrepreneur: ZASP_MIE_PRO_KSI_RACH for_citizen: ZASO_MIE_POL_OBI; ZASO_MIE_ODP_CEL</p>
17-08-2017	<p>Zmodyfikowano kody błędów dla ApiCommons/getSYM</p>

Spis treści

Wstęp	8
Format odpowiedzi	8
Opis funkcji	9
1. ApiCommons	9
1.1. getVoivodeships - Słownik województwa	9
1.2. getPKD - Słownik PKD	9
1.3. getPKDDetails - Pobranie szczegółów PKD	10
1.4. getPKDList - wyszukiwanie PKD	11
1.5. getPKDHints – Pobranie podpowiedzi dla wyszukiwarki procedur po pkd	12
1.6. getSYM – wyszukanie miejscowości w oparciu o tekst lub kod pocztowy	14
2. ApiLifeSituations	16
2.1. Algorytm	16
2.2. getPDF - Pobranie PDF-a	16
2.3. startLifeSituation - start sytuacji życiowej	17
2.4. getCurrentQuestion - pobranie pytania	18
2.5. saveCurrentQuestion – zapisanie bieżącego pytania	20
2.6. revertToPreviousQuestion - cofnięcie do poprzedniego pytania	20
2.7. revertToStage – powrót do wybranego etapu	21
2.8. getEndDescription - pobranie końcowego opisu	22
2.9. getSummary - pobranie podsumowania	22
2.10. getAnswers - Metoda pobiera odpowiedzi sytuacji życiowej	23
2.11. getList – Pobranie listy sytuacji życiowych	24
2.12. getDescription Pobranie opisu sytuacji życiowej	25
2.13. linkSessionWithUser Powiązanie sesji sytuacji życiowej z użytkownikiem	27
2.14. getVisibleLifeSituationsPropositions Pobranie listy widocznych propozycji sytuacji życiowych	28
2.15. voteOnLifeSituationProposition Głosowanie na propozycję sytuacji życiowej	29
3. ApiOpinions	31
3.1. saveOpinionForContent - Dodawanie opinii	31
4. ApiInstitutions	33
4.1. saveInstitutionProposition - Zapisywanie propozycji nowej instytucji	33
4.2. getListToPerformElectronicProcedure – Pobranie list instytucji wykonującej elektronicznie procedurę	35
4.3. getCategories - Pobranie listy kategorii instytucji	36

4.4.	getHints Pobieranie podpowiedzi nazw instytucji	36
4.5.	getList - Pobieranie listy instytucji.....	37
4.6.	getDetails - Pobieranie szczegółów instytucji	39
4.7.	getDepartmentDetails - Pobieranie szczegółów wydziału instytucji.....	41
4.8.	getListByTeryt - Pobieranie listy jednostek samorządu terytorialnego	42
5.	ApiProcedures	44
5.1.	Wprowadzenie	44
5.1.1.	Zastosowanie parametru ALIAS zamiast ID w wywołaniach	44
5.2.	getHints - Pobieranie podpowiedzi nazw procedur	44
5.3.	getList - Pobieranie listy procedur.....	45
5.4.	getElectronicList - Pobieranie listy procedur elektronicznych	48
5.5.	getDetails - Pobieranie szczegółów procedury.....	49
	Znaczenia „executing_institution”	56
5.6.	getThematicCategories - Pobieranie katalogu tematycznego.....	57
5.7.	getProfessionCategories - Pobieranie katalogu zawodów.....	57
5.8.	getListForRespAuthority - Pobieranie listy procedur dla Instytucji Centralnych / Akceptujących	58
6.	ApiPnaInstitutions	60
6.1.	getHintsByCity – pobieranie podpowiedzi po nazwie miejscowości.....	60
6.2.	getListByLocation – pobranie instytucji na podstawie lokalizacji	61
7.	ApiPnaAddress.....	63
7.1.	getListByPostCode – lista miejscowości po kodzie pocztowym	63
7.2.	getStreetHints - pobranie podpowiedzi ulic.....	63
8.	ApiPublications.....	65
8.1.	getDetails – Pobranie publikacji	65
8.2.	getList – Pobranie listy publikacji	66
9.	ApiMail	68
9.1.	sendMail – wysyłka wiadomości e-mail	68
9.2.	getMessages – pobieranie wiadomości	70
10.	ApiUsers.....	72
10.1.	getDeletedUsers – pobranie usuniętych użytkowników.....	72

Wstęp

Do komunikacji z zewnętrznymi partnerami w projekcie Baza Wiedzy EPK wykorzystano wzorzec architektury REST, Representational State Transfer.

Parametry w zależności od metody przesyłane są metodą POST lub GET. Jeżeli dane są wysyłane metodą POST to w nagłówku wysyłanej wiadomości parametr Content-Type musi być ustawiony na „application/x-www-form-urlencoded”. Jest to standardowy format przesyłania parametrów metodą POST.

Wszystkie wyniki reprezentowane są przy użyciu zapisu JSON. Pełen opis tego formatu danych znajduje się w dokumencie [RFC 4627](http://tools.ietf.org/html/rfc4627) <http://tools.ietf.org/html/rfc4627> .

Format odpowiedzi

Struktura każdej odpowiedzi wygląda następująco:

```
{
  "content" : < rezultat zapytania >,
  "language" : < kod tłumaczenia >,
  "status" : {
    "code" : < kod odpowiedzi/błędu >,
    "message" : < opis odpowiedzi/błędu >
  }
}
```


Opis funkcji

1. ApiCommons

1.1. getVoivodeships - Słownik województwa

Opis

Metoda pobierająca słownik województw wraz z przyporządkowanym kodem teryt.

Nazwa funkcji

getVoivodeships

Parametry wejściowe

Parametry wysyłane metodą POST					
Nazwa zmiennej	Typ	obowiązkowy	domyślna wartość	warunki	opis
Lang	String	Nie	pl	poprawny kod lang	kod języka w standardzie ISO 639-1

Struktura danych wyjściowych

```
{
  "content" : [ {
    "Territory" : {
      "teryt" : < identyfikator terytu >,
      "name" : < nazwa województwa >
    }
  } ],
  "language" : < kod języka >,
  "status" : {
    "code" : < status_odpowiedzi >,
    "message" : < komunikat >
  }
}
```

Przykład wywołania

```
http://[adres_usługi]/ApiCommons/getVoivodeships
```

Opis kodów

```
OK_0000 poprawna odpowiedź
```

1.2. getPKD - Słownik PKD

Opis

Metoda pobierająca określony fragment drzewa PKD na podstawie przesłanego kodu PKD. W przypadku nie wysłania kodu PKD metoda zwraca główny kategorię PKD

Nazwa funkcji

getPKD

Parametry wejściowe

Parametry wysyłane metodą POST					
Nazwa zmiennej	Typ	obowiązkowy	domyślna wartość	warunki	opis
code	string	nie		max 10 znaków	kod PKD
lang	string	nie	pl	poprawny kod lang	kod języka w standardzie ISO 639-1

Struktura danych wyjściowych

```
{
  "content" : {
    "PkdTreeFragment" : [ {
      "id" : < identyfikator PKD >,
      "code" : < kod PKD >,
      "title" : < nazwa PKD >,
      "hasChildren" : < informacja czy dany element posiada podgałęzie >
    } ]
  },
  "language" : < kod języka >,
  "status" : {
    "code" : < status odpowiedzi >,
    "message" : < komunikat >
  }
}
```

Przykład wywołania

```
http://[adres_usługi]/ApiCommons/getPkd
http://[adres_usługi]/ApiCommons/getPkd?code=A
http://[adres_usługi]/ApiCommons/getPkd?lang=en
http://[adres_usługi]/ApiCommons/getPkd?code=A&lang=en
```

Opis kodów

```
OK_0000 poprawna odpowiedź
OK_0002 poprawna odpowiedź brak wyników

ERR_0006 Pole [nazwa pola] ma większą długość niż [max] znaków
ERR_0005 Pole lang posiada niepoprawny kod języka.
```

1.3. getPKDDetails - Pobranie szczegółów PKD

Opis

Metoda pobierająca szczegóły PKD. W odpowiedzi dostajemy pełne informacje o PKD

Nazwa funkcji

getPKDDetails

Parametry wejściowe

Parametry wysyłane metodą POST					
Nazwa zmiennej	Typ	obowiązkowy	domyślna wartość	warunki	opis
id	int	tak			identyfikator PKD
lang	string	nie	pl	poprawny kod lang	kod języka w standardzie ISO 639-1

Struktura danych wyjściowych

```
{
  "content" : {
    "PkdDetails" : {
      "id" : < identyfikator PKD >,
      "code" : < kod PKD >,
      "title" : < nazwa PKD >,
      "includes_description" : < podklasa obejmuje >,
      "not_includes_description" : < podklasa nie obejmuje >
    }
  },
  "language" : < kod języka >,
  "status" : {
    "code" : < status_odpowiedzi >,
    "message" : < komunikat >
  }
}
```

Przykład wywołania

```
http://[adres_uslugi]/ApiCommons/getPKDDetails?id=8242
http://[adres_uslugi]/ApiCommons/getPKDDetails?id=8242&lang=en
```

Opis kodów

OK_0000 poprawna odpowiedź

ERR_0005 Pole lang posiada niepoprawny kod języka.

ERR_0011 Pole [nazwa pola] jest wymagane.

ERR_0014 Pole [nazwa pola] nie posiada prawidłowej wartości dla ID.

1.4. getPKDList - wyszukiwanie PKD

Metoda wyszukuje PKD na podstawie kodu, opisu lub słów kluczowych (użyty mechanizm SPHINX)

Nazwa funkcji

getPKDList

Parametry wejściowe

Parametry wysyłane metodą GET					
Nazwa zmiennej	Typ	obowiązkowy	domyślna wartość	warunki	opis
search	string	nie		max 255 znaków	wyszukiwana fraza – część słowa, pełna fraza lub pełen kod (z literą lub bez) podany w formacie scalonym lub z kropkami

sort	string	nie	weight	weight, code, name, for_natural_person.	wartość z listy, przy czym „weight” sortuje dwustopniowo – po wadze, a potem po nazwie (jeśli nie ma search, waga jest stała, sortowanie jest po nazwie)
order	string	nie	asc	asc, desc	ustawia kierunek sortowanie listy po nazwie sytuacji życiowej, przyjmuje dwie możliwe wartości: <i>asc</i> lub <i>desc</i>
lang	string	nie	pl	poprawny kod lang	kod języka w standardzie ISO 639-1

Struktura danych wyjściowych

```
{
  "content" : {
    "hint" : < fraza, która zastąpiła przypuszczalnie błędnie wpisane słowa >,
    "pkd_list" : [ {
      "id" : < wewnętrzny identyfikator PKD >,
      "code" : < kod PKD w formacie AA.BB.C >,
      "section" : < sekcja PKD >,
      "name" : < nazwa PKD >,
      "for_natural_person" : < dla osoby fizycznej: true/false >,
      "related_procedures" : < czy istnieją powiązane procedury: true/false >,
      "related_procedures_count" : < ilość powiązanych procedur >,
      "related_life_situations_count" : < ilość powiązanych poradników >,
      "weight" : < waga >,
    } ]
  },
  "language" : < kod języka >,
  "status" : {
    "code" : < status odpowiedzi >,
    "message" : < komunikat >
  }
}
```

Przykład wywołania

```
http://[adres_usługi]/ApiCommons/getPKDList
http://[adres_usługi]/ApiCommons/getPKDList?search=produkcja
```

Opis kodów

```
OK_0000 - poprawna odpowiedź
OK_0002 - poprawna odpowiedź, brak wyników

ERR_0005 - Pole lang posiada niepoprawny kod języka.
ERR_0006 - Pole [nazwa pola] ma większą długość niż [max] znaków.
ERR_0024 - wartość pola z poza listy
```

1.5. getPKDHints – Pobranie podpowiedzi dla wyszukiwarki procedur po pkd

Metoda pobierająca podpowiedzi dla wyszukiwarki procedur po PKD na podstawie wpisanej frazy. Podpowiadają się pełne nazwy PKD

Nazwa funkcji getPKDHints

Parametry wejściowe

Parametry wysłane metodą GET					
Nazwa zmiennej	Typ	obowiązkowy	domyślnie	warunki	opis
search	string	tak		max 100 znaków	początkowy fragment słowa znajdującego się w nazwach PKD
limit	int	nie	10	min wartość 1, max wartość 99999	liczba wyników na stronie
lang	string	nie	pl	poprawny kod lang	kod języka w standardzie ISO 639-1

Struktura danych wyjściowych

```
{
  "content" : [ {
    "name" : < nazwa PKD >
  } ],
  "language" : < kod języka >,
  "status" : {
    "code" : < status odpowiedzi >,
    "message" : < komunikat >
  }
}
```

Przykład wywołania

[http://\[adres_usługi\]/ApiCommons/getPKDHints?search=przetwórstwo](http://[adres_usługi]/ApiCommons/getPKDHints?search=przetwórstwo)

Opis kodów

OK_0000 poprawna odpowiedź
 OK_0002 poprawna odpowiedź nie odnaleziono wyników spełniających podane kryterium
 ERR_0004 - Niepoprawny typ pola
 ERR_0005 - Pole lang posiada niepoprawny kod języka.
 ERR_0006 - Pole [nazwa pola] ma większą długość niż [max] znaków.
 ERR_0010 - Pole wykracza poza dopuszczalny zakres

1.6. getSYM – wyszukanie miejscowości w oparciu o tekst lub kod pocztowy

Metoda pozwala na wyszukanie prawidłowej miejscowości w oparciu o tekst wpisany przez użytkownika. Może być nim fragment nazwy miejscowości, kompletna nazwa miejscowości wraz z fragmentem gminy po przecinku, lub kompletny kod pocztowy.

System wyszukuje jedynie pełne miejscowości z bazy SIMC (bez miejscowości będących częściami innych miejscowości)

Wynikowo użytkownik dostaje listę miejscowości posortowaną wg preferencji miasto/wieś a następnie posortowaną alfabetycznie.

System wyszukuje nazwy wg następujących kryteriów:

- jeśli fraza wejściowa to kod pocztowy TRIM([0-9]{2}[-]*[0-9]{3}) system pyta o kod pocztowy, w innym wypadku o miejscowość
- system jest niezależny od ilości spacji i myślników w nazwach miejscowości i traktuje je zamiennie
- system zwraca uwagę na polskie znaki
- system nie zwraca uwagi na wielkość znaków
- jeśli we frazie jest przecinek, to rozdzielamy zapytanie na pytanie o miejscowość (przed przecinkiem) i pytanie o gminę (po przecinku)
- fraza przed przecinkiem - czyli miejscowość wyodrębniona z frazy podlega takim samym zasadom jak dla frazy tylko z samą miejscowością
- miejscowość nie może mieć mniej niż 2 znaków znaczących. Jeśli tak się stanie - system zgłosi wyjątek zbyt krótkiej frazy i zakończy proces
- pytanie o gminę zawsze jest nazwa_gminy% niezależnie od liczby znaków
- jeśli fraza miejscowości <5 znaków a więcej niż 1 znak, wówczas zapytanie kierowane jest o dokładnie taką nazwę miejscowości jako podano
- jeśli fraza miejscowości >=5 znaków, to zapytanie kierowane jest do bazy o nazwa%
- jeśli nie ma odpowiedzi w wyniku zapytania, wówczas zanim odpowiemy użytkownikowi, system uruchamia zapytanie z eliminacją znaków diakrytycznych w tekście przy wyszukiwaniu, czyli zwróci np. „Poznań” po wpisaniu „Poznan”
- jeśli nadal nie ma odpowiedzi a fraza jest mniejsza niż 5 znaków to generowany jest wyjątek typu "brak wyników"
- jeśli fraza miejscowości zawiera spację lub myślnik w środku (jest wielocłonowa) również generuj wyjątek typu "brak wyników"
- jeśli nadal nie ma odpowiedzi w wyniku zapytania to system korzysta z algorytmów wykrywających literówki w tekście wg zaszytego słownika i próbuje zadać pytanie bazę o sugerowany poprawny wynik
- jeśli i to nie pomogło - system zwraca wyjątek typu "brak wyników"

Nazwa funkcji

getSYM

Parametry wejściowe

Parametry wysłane metodą GET					
Nazwa zmiennej	Typ	obowiązkowy	domyślnie	warunki	opis
search	string	tak		Kod pocztowy w formacie xx-xxx lub ciąg tekstu długości 2-255 znaków	fragment nazwy miejscowości, kompletna nazwa miejscowości wraz z fragmentem gminy po przecinku, lub kompletny kod pocztowy

Struktura danych wyjściowych

```
{
"content" :
[
[
{"localities" : <lista najbardziej pasujących miejscowości/gmin>
[
{"name" : <nazwa miejscowości wyświetlana użytkownikowi w standardzie: miejscowość (gmina, województwo), jeśli SYM to miasto to nie wyświetla się gmina, tylko samo woj. >,
"sym" : <kod SYM (baza SIMC) dla danej miejscowości np. 0879340>
}],
"rows_count" : < liczba rekordów ogółem >,
}],
"language" : < kod języka, zawsze PL >,
"status" : {
"code" : < status odpowiedzi >,
"message" : < komunikat >,
}
}
]
```

Przykłady wywołania

```
http://[adres_usługi]/ApiCommons/getSYM?search=Warszawa
```

Opis kodów

OK_0000 poprawna odpowiedź

OK_0002 poprawna odpowiedź nie odnaleziono wyników spełniających podane kryterium

ERR_7001 Podana fraza jest zbyt krótka. Podaj co najmniej dwa znaki dla miejscowości lub gminy.

ERR_7002 Nie wprowadzono tekstu.

ERR_7003 Nieprawidłowy kod pocztowy, wprowadź kod w formacie 'XX-XXX'.

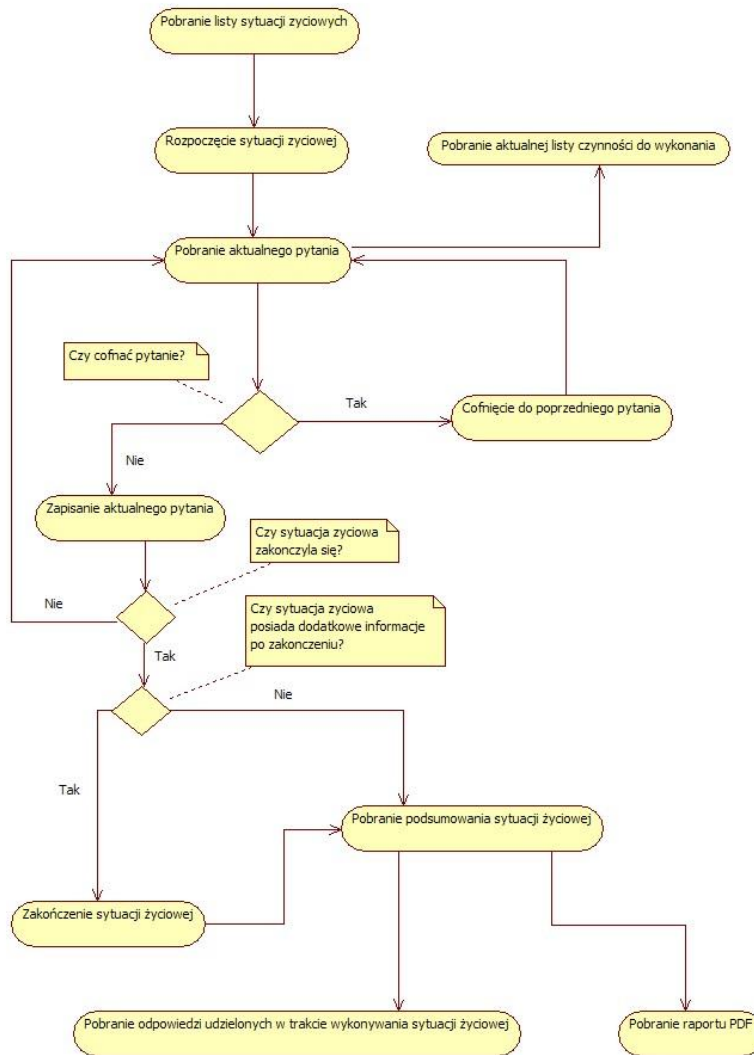
ERR_7004 Nieprawidłowy format kodu pocztowego. Wprowadź kod w formacie liczbowym 'XX-XXX'.

ERR_7005 Nazwa gminy nie powinna zawierać znaku ','.

2. ApiLifeSituations

2.1. Algorytm

Poniższy algorytm opisuje sposób poruszania się po metodach api sytuacji życiowych.



2.2. getPDF - Pobranie PDF-a

Metoda pobiera PDF-a dla ukończonej sytuacji życiowej

Nazwa funkcji

getPDF

Parametry wejściowe

Parametry wysłane metodą POST

Nazwa zmiennej	Typ	obowiązkowy	domyślna wartość	warunki	opis
----------------	-----	-------------	------------------	---------	------

hash	string	tak			identyfikator wykonanej sytuacji życiowej
steps	int	nie	0	wartość 0 lub 1	flaga umożliwiająca wyświetlenie dokładnych informacji na temat szczegółów procedury (przebieg)
summary	int	nie	0	wartość 0 lub 1	flaga umożliwiająca dodanie podsumowania oraz listę procedur do dokumentu PDF

Struktura danych wyjściowych

Jeżeli zostały poprawnie przekazane parametry wejściowe, po wykonaniu metody przekazany zostanie pdf

Przykład wywołania

```
http://[adres_usługi]/ApiLifeSituations/getPDF?hash=2164d0be48f216ce514ec76eda7f53c779c14ff4&steps=0
```

Opis kodów

```
ERR_0013 Wartość pola nie znajduje się na liście (0,1)
ERR_0004 Pole jest niepoprawnego typu
ERR_0011 Pole jest wymagane
ERR_5006 Sytuacja życiowa nie została ukończona.
```

2.3. startLifeSituation - start sytuacji życiowej

Metoda startuje sytuację życiową

Nazwa funkcji

startLifeSituation

Parametry wejściowe

Parametry wysyłane metodą POST					
Nazwa zmiennej	Typ	obowiązkowy	domyślna wartość	warunki	opis
acronym	string	tak		max 255 znaków	Akronim, dla testowego wywołania może przyjąć postać *TEST-xxx1-xxx2*yyy gdzie xxx1 to id sytuacji życiowej xxx2 to timestamp yyy to akronim sytuacji życiowej w postaci „id-akronim” lub „id” lub „akronim”
lang	string	nie	PL	poprawny kod lang	kod języka w standardzie ISO 639-1
user_id	int	nie		unikatowy identyfikator	identyfikator użytkownika

Struktura danych wyjściowych

```
{
  "content" : {
    "description" : < opis startowy sytuacji życiowej >,
    "keywords" : < keywords >,
    "id" : < identyfikator sytuacji życiowej >,
    "name" : < nazwa sytuacji życiowej >,
    "category" : < kategoria >,
    "importance" : < ważność sytuacji życiowej z przedziału od 0 do 999 >,
    "icon" : < link do ikony sytuacji życiowej >,
    "smallDescription" : < skrócony opis >,
    "catalogues" : [ { < katalogi tematyczne powiązane z sytuacją życiową >
      "id" : < identyfikator katalogu >,
      "name" : < nazwa katalogu >
    } ],
    "language" : < język w którym może być opis >,
    "hash" : < hash >,
    "stages" : [ { < drzewo etapów >
      "stage_id" : < identyfikator etapu >,
      "process_id" : < identyfikator sytuacji życiowej związanej z etapem >,
      "title" : < tytuł etapu >,
      "substages" : [ { < drzewo podetapów >
        } ]
    } ]
  },
  "language" : < kod języka >,
  "status" : {
    "code" : < status odpowiedzi >,
    "message" : < komunikat >
  }
}
```

Przykład wywołania

```
http://[adres_usługi]/ApiLifeSituations/startLifeSituation?acronym=GABINET_PIELEGNIARSKI_R
http://[adres_usługi]/ApiLifeSituations/startLifeSituation?acronym=205-
GABINET_PIELEGNIARSKI_R
http://[adres_usługi]/ApiLifeSituations/startLifeSituation?acronym=205
http://[adres_usługi]/ApiLifeSituations/startLifeSituation?acronym=*TEST-12-9999999*GABINET
_PIELEGNIARSKI_R
```

Opis kodów

OK_0000 poprawna odpowiedź
 ERR_0006 Pole acronym ma większą długość niż 255 znaków
 ERR_0011 Pole jest wymagane
 ERR_0005 Pole lang posiada niepoprawny kod języka.
 ERR_5016 Jeden lub więcej podprocesów wchodzących w skład sytuacji życiowej nie jest opublikowany. Nie można rozpocząć sytuacji życiowej.
 ERR_5018 - Pole [nazwa pola] posiada niepoprawną strukturę.
 ERR_5019 - Przekroczono czas uruchomienia testowej sytuacji życiowej. Należy ponownie uruchomić podgląd sytuacji życiowej z zaplecza
 ERR_5020 - Nieodnaleziono sytuacji życiowej o podanym ID

2.4. getCurrentQuestion - pobranie pytania

Metoda pobiera bieżące pytanie

Nazwa funkcji

getCurrentQuestion

Parametry wejściowe

Parametry wysyłane metodą POST					
Nazwa zmiennej	Typ	obowiązkowy	Wartość domyślna	warunki	opis
hash	string	tak		max 40 znaków	hash
lang	string	nie	PL	poprawny kod lang	kod języka w standardzie ISO 639-1

Struktura danych wyjściowych

```
{
  "content" : {
    "question" : {
      "question" : < pytanie >,
      "description" : < dodatkowy opis pytania >
      "answer_type" : <typ odpowiedzi, jednokrotnego lub wielokrotnego wyboru:
checkbox/radio>
    },
    "answers" : [ { < tablicaodpowiedzi >
      "answer_id" : < identyfikator odpowiedzi przekazywany do metody saveCurrentQuestion
>,
      "answer" : < treść odpowiedzi >
      "is_exclusive": <dodatkowy parametr, który określa czy odpowiedź wyłącza możliwość
wyboru innych (może wystąpić 1 na dane pytanie o typie 'checkbox')>: false/true>
    } ],
    "last_answer" : {
      "answer_id" : < ostatnia odpowiedz udzielona przez uzytkownika na to pytanie >
    },
    "actions" : [ { < tablica czynności >
      "url" : < adres URL >,
      "url_description" : < opis co znajduje się pod URL >,
      "description" : < opis czynności >,
      "title" : < nazwa czynności >,
      "type" : < typ czynności >
    } ],
    "stages" : [ { < drzewo etapów >
      "stage_id" : < identyfikator etapu >,
      "process_id" : < identyfikator sytuacji życiowej związanej z etapem >,
      "title" : < tytuł etapu >,
      "substages" : {
        } ]
    },
    "current_stage" : { < aktualny etap >
      "stage_id" : < identyfikator etapu >,
      "process_id" : < identyfikator procesu >
    }
  },
  "language" : pl,
  "status" : {
    "code" : < status odpowiedzi >,
    "message" : < komunikat >
  }
}
```

Przykład wywołania

```
http://[adres_uslugi]/ApiLifeSituations/getCurrentQuestion?hash=2164d0be48f216ce514ec76eda7f53c779c14ff4
```

Opis kodów

```
OK_0000 Poprawna odpowiedź
OK_0001 Zakończono wykonywanie sytuacji życiowej
```

ERR_0011 Pole hash jest wymagane
 ERR_0005 Pole lang posiada niepoprawny kod języka
 ERR_0022 Pole hash posiada nieprawidłową wartość hash.
 ERR_5008 Brak sesji o podanym identyfikatorze.
 ERR_5009 Sytuacja życiowa posiada nieprawidłową strukturę.

2.5. saveCurrentQuestion – zapisanie bieżącego pytania

Metoda zapisuje odpowiedź na bieżące pytanie

Nazwa funkcji

saveCurrentQuestion

Parametry wejściowe

Parametry wysyłane metodą POST				
Nazwa zmiennej	Typ	obowiązkowy	warunki	opis
hash	string	tak	max 40 znaków	hash
answer_id	int	tak	weryfikacji id	identyfikator odpowiedzi

Struktura danych wyjściowych

```
{
  content : {
    status : < status >
  },
  "language" : < kod języka >,
  "status" : {
    "code" : < status_odpowiedzi >,
    "message" : < komunikat >
  }
}
```

Przykład wywołania

[http://\[adres_usługi\]/saveCurrentQuestion?hash=2164d0be48f216ce514ec76eda7f53c779c14ff4](http://[adres_usługi]/saveCurrentQuestion?hash=2164d0be48f216ce514ec76eda7f53c779c14ff4)

Opis kodów

OK_0000 poprawna odpowiedź
 ERR_0004 Pole answer_id posiada niepoprawny typ. Wymagany typ to INT
 ERR_0011 Pole jest wymagane
 ERR_0022 Pole hash posiada nieprawidłową wartość hash
 ERR_5003 Brak odpowiedzi dla aktualnego stanu sesji
 ERR_5004 Taka odpowiedź nie istnieje dla tego pytania
 ERR_5024 Dla pytania typu RADIO odpowiedź musi przyjmować jedną wartość
 ERR_5023 Zaznaczono w odpowiedzi wiele wartości, spośród których jedna posiada atrybut EXCLUSIVE

2.6. revertToPreviousQuestion - cofnięcie do poprzedniego pytania

Metoda cofa do poprzedniego pytania

Nazwa funkcji

revertToPreviousQuestion

Parametry wejściowe

Parametry wysyłane metodą POST				
Nazwa zmiennej	Typ	obowiązkowy	warunki	opis
hash	string	tak	max 40 znaków	hash

Struktura danych wyjściowych

```
{
  "content" : {
    "status" : < status >
  },
  "language" : < kod języka >,
  "status" : {
    "code" : < status odpowiedzi >,
    "message" : < komunikat >
  }
}
```

Przykład wywołania

```
http://[adres_usługi]/ApiLifeSituations/revertToPreviousQuestion?hash=2164d0be48f216ce514ec76eda7f53c779c14ff4
```

Opis kodów

OK_0000 poprawna odpowiedź

ERR_0011 Pole jest wymagane

ERR_5007 Sytuacja życiowa została zakończona. Nie ma możliwości cofnięcia się

ERR_5001 Wystąpił błąd podczas wycofywania odpowiedzi

2.7. revertToStage – powrót do wybranego etapu

Metoda umożliwia powrót do pierwszego pytania wybranego etapu

Nazwa funkcji

revertToStage

Parametry wejściowe

Parametry wysyłane metodą POST				
Nazwa zmiennej	Typ	obowiązkowy	warunki	opis
hash	string	tak	max 40 znaków	Hash
stage_id	Int	Tak		Identyfikator etapu
process_id	Int	Tak		Identyfikator sytuacji życiowej

Struktura danych wyjściowych

```
{
  "content" : {
    "status" : < status >
  },
  "language" : < kod języka >,
  "status" : {
    "code" : < status odpowiedzi >,
    "message" : < komunikat >
  }
}
```

Przykład wywołania

[http://\[adres_uslugi\]/ApiLifeSituations/revertToStage?hash=af1ec08554448b55d2a1b2757b725da89c672f84&process_id=1244&stage_id=0](http://[adres_uslugi]/ApiLifeSituations/revertToStage?hash=af1ec08554448b55d2a1b2757b725da89c672f84&process_id=1244&stage_id=0)

Opis kodów

OK_0000 poprawna odpowiedź

ERR_0011 Pole jest wymagane

ERR_5007 Sytuacja życiowa została zakończona. Nie ma możliwości cofnięcia się

ERR_5001 Wystąpił błąd podczas wycofywania odpowiedzi

ERR_5008 Sesja nie istnieje

ERR_5012 Taka sytuacja życiowa nie istnieje

ERR_5013 Taki etap dla podanej sytuacji życiowej nie istnieje

ERR_5014 W tej sesji nie występuje sytuacja życiowa o podanej strukturze

ERR_5015 Nie można cofnąć się do etapu który nie został jeszcze wykonany

2.8. getEndDescription - pobranie końcowego opisu

Metoda pobiera opis zakończonej sytuacji życiowej

Nazwa funkcji

getEndDescription

Parametry wejściowe

Parametry wysyłane metodą POST				
Nazwa zmiennej	Typ	obowiązkowy	warunki	opis
hash	string	tak	max 40 znaków	hash

Struktura danych wyjściowych

```
{
  "content" : {
    "title" : < tytuł opisu >,
    "description" : < tresc opisu >
  },
  "language" : < kod języka >,
  "status" : {
    "code" : < status odpowiedzi >,
    "message" : < komunikat >
  }
}
```

Przykład wywołania

[http://\[adres_uslugi\]/ApiLifeSituations/getEndDescription?hash=2164d0be48f216ce514ec76eda7f53c779c14ff4](http://[adres_uslugi]/ApiLifeSituations/getEndDescription?hash=2164d0be48f216ce514ec76eda7f53c779c14ff4)

Opis kodów

OK_0000 poprawna odpowiedź

ERR_0011 Pole jest wymagany

ERR_5006 Sytuacja życiowa nie została zakończona

2.9. getSummary - pobranie podsumowania

Metoda pobiera podsumowanie sytuacji życiowej

Nazwa funkcji

getSummary

Parametry wejściowe

Parametry wysyłane metodą POST				
Nazwa zmiennej	Typ	obowiązkowy	warunki	opis
hash	string	tak	max 40 znaków	hash

Struktura danych wyjściowych

```
{
  "content" : { < lista czynności do wykonania >
    "url" : < odnośnik do artykułu lub procedury >,
    "url_description" : < opis odnośnika >,
    "description" : < opis czynności >,
    "title" : < tytuł/nazwa czynności >,
    "type" : < typ czynności >
  },
  "language" : < kod języka >,
  "status" : {
    "code" : < status odpowiedzi >,
    "message" : < komunikat >
  }
}
```

Przykład wywołania

```
http://[adres_usługi]/ApiLifeSituations/getSummary?hash=2164d0be48f216ce514ec76eda7f53c779c14ff4
```

Opis kodów

OK_0000 poprawna odpowiedź

ERR_0011 Pole jest wymagany

ERR_5006 Sytuacja życiowa nie została zakończona

2.10. getAnswers - Metoda pobiera odpowiedzi sytuacji życiowej

Metoda zwraca listę wszystkich pytań wraz z udzielonymi odpowiedziami

Nazwa funkcji

getAnswers

Parametry wejściowe

Parametry wysyłane metodą POST				
Nazwa zmiennej	Typ	obowiązkowy	warunki	opis
hash	string	tak	max 40 znaków	hash

Struktura danych wyjściowych

```
{
  "content" : { < lista udzielonych odpowiedzi >
    {
      "question" : < pytanie >,
      "answer" : < udzielona odpowiedź >
    }
  },
  "language" : < kod języka >,
}
```

```

"status" : {
  "code" : < status odpowiedzi >,
  "message" : < komunikat >
}
}

```

Przykład wywołania

```

http://[adres_uslugi]/ApiLifeSituations/getAnswers?hash=2164d0be48f216ce514ec76eda7f53c779c14ff4

```

Opis kodów

OK_0000 poprawna odpowiedź

ERR_0011 Pole jest wymagany

ERR_5006 Sytuacja życiowa nie została zakończona

2.11. getList – Pobranie listy sytuacji życiowych

Metoda zwraca listę wszystkich sytuacji życiowych posortowanych po nazwie sytuacji życiowej. Możliwe jest podanie parametru **search**, **lang**, **limit**, **page**, **sort**. W przypadku pobierania listy w języku innym niż język polski, jeśli nie została znaleziona sytuacja życiowa w żądany języku zwracana jest sytuacja życiowa w języku polskim

Nazwa funkcji

getList

Wersja

V2

Parametry wejściowe

Parametry wysyłane metodą GET					
Nazwa zmiennej	Typ	obowiązkowy	domyślnie	warunki	opis
search	string	nie		max 100 znaków	umożliwia zawężenie zwracanej listy do elementów wyszukanych po nazwie lub po słowach kluczowych na podstawie przekazanego parametru
lang	string	nie	PL	poprawny kod lang	kod języka w standardzie ISO 639-1
limit	int	nie	10	max wartość 9999	ustawia ilość zwracanych jednorazowo elementów listy
page	int	nie	1	max wartość 999	ustawia numer strony która ma zostać zwrócona, współpracuje z parametrem limit
order	string	nie	asc	na liście (asc, desc), max 4 znaki	ustawia kierunek sortowanie listy po nazwie sytuacji życiowej, przyjmuje dwie możliwe wartości: <i>asc</i> lub <i>desc</i>
category	array	nie		Wartość z listy musi być zgodna ze słownikiem	Lista kategorii podana po przecinku z zakresu : 'ZAK' => 'Zakładam firmę', 'PRO' => 'Prowadzę firmę',

					'ZAM' => 'Zamykam firmę', 'USL' => 'Tymczasowo świadczę usługi na terenie Polski'
importance	Int	Nie		Z zakresu od 0 do 999	Określa ważność z przedziału od 0 do 999
sort	String	Nie	Name	Na liście, max 25 znaków	Wartość z listy name, category,importanc
pkd	string	nie			Poprawny kod PKD, z kropkami lub bez

Struktura danych wyjściowych

```
{
  "content" : {
    "rows_count" : < liczba wszystkich wierszy >,
    "LifeSituations" : [ {
      "LifeSituation" : {
        "id" : < identyfikator sytuacji życiowej >,
        "acronym" : < akronim sytuacji życiowej >,
        "category" : < kod kategorii >,
        "importance" : < ważność poradnika >,
        "icon" : < link do ikonki sytuacji życiowych >,
        "name" : < nazwa sytuacji życiowej >,
        "language" : < język w którym został zwrócony element listy >,
        "short_introduction" : < zawartość pola „Krótkie wprowadzenie” >
      }
    } ]
  },
  "language" : < kod języka >,
  "status" : {
    "code" : < status odpowiedzi >,
    "message" : < komunikat >
  }
}
```

Przykład wywołania

```
http://[adres_usługi]/ApiLifeSituations/getList
http://[adres_usługi]/ApiLifeSituations/getList?limit=500
http://[adres_usługi]/ApiLifeSituations/getList?limit=500&search=otwieram
```

Opis kodów

OK_0000 - poprawna odpowiedź

ERR_0004 - pole posiada [nazwa_pola] niepoprawny typ. Wymagany typ to [typ]
 ERR_0005 - pole [nazwa_pola] posiada niepoprawny kod języka
 ERR_0006 - pole [nazwa_pola] ma większą długość niż [wymagana_długość] znaków.
 ERR_0007 - pole [nazwa_pola] większą wartość niż [maksymalna wartość].
 ERR_0007 - pole [nazwa_pola] mniejsza wartość niż [maksymalna wartość].
 ERR_0013 - pole [nazwa_pola] nie znajduje się na liście [wymagana lista parametrów].
 ERR_0024 - wartość pola z poza listy
 ERR_0025 - wartość pola pkd jest niepoprawnym kodem PKD
 ERR_0035 - Żądana strona wykracza poza listę wyników

2.12. getDescription Pobranie opisu sytuacji życiowej

Metoda zwraca opis sytuacji życiowej. Wymagane jest podanie parametru **acronym**, możliwe jest podanie parametru **lang**. Wartość pola **acronym** to złączenie „id-akronim” lub „id” lub „akronim”. W przypadku pobierania opisu w języku innym niż język polski, jeśli nie został znaleziony opis sytuacji życiowej w żądany języku zwracany jest opis sytuacji życiowej w języku polskim

Nazwa funkcji

getDescription

Parametry wejściowe

Parametry wysyłane metodą GET					
Nazwa zmiennej	Typ	obowiązkowy	domyślnie	warunki	opis
acronym	string	tak		max 255 znaków	akronim sytuacji życiowej, otrzymany po wywołaniu funkcji getList Uwaga: akronim może przyjmować również format: *TEST-[id sytuacji życiowej]- [timestamp]*[Akronim sytuacji życiowej]
lang	string	nie	PL	poprawny kod lang	kod języka w standardzie ISO 639-1

Struktura danych wyjściowych

```
{
  "content" : {
    "LifeSituationAction" : {
      "acronym" : < akronim sytuacji życiowej >,
      "description" : < zwrócony opis sytuacji życiowej >,
      "keywords" : < słowa kluczowe >,
      "smallDescription" : < skrócony opis >,
      "id" : < identyfikator sytuacji życiowej >,
      "name" : < nazwa sytuacji życiowej >,
      "category" : < kod kategorii sytuacji życiowej >,
      "importance" : < ważność sytuacji życiowej z przedziału od 0 do 999 >,
      "icon" : < link do ikony >,
      "description" : < opis sytuacji życiowej >,
      "catalogues" : [ { < katalogi tematyczne powiązane z sytuacją życiową >
        "id" : < identyfikator katalogu >,
        "name" : < nazwa katalogu >
      } ],
      "relatedPublications" : [ { < publikacje powiązane z sytuacją życiową >
        "alias" : < alias publikacji >,
        "title" : < tytuł publikacji >,
        "language" : < język, w którym zwrócony został tytuł publikacji >
      } ],
      "acts" : [ { < tablica aktów prawnych powiązanych z poradnikiem >
        "id" : < id aktu prawnego >,
        "name" : < nazwa aktu prawnego >,
        "link" : < link do aktu prawnego >,
        "type_code" : < kod aktu prawnego Us; Ro; Uch; Pu; In >,
        "type_description" : < opis kodu aktu prawnego (Ustawa; Rozporządzenie; Uchwała; Przepisy unijne; Inny >,
        "language" : < język, w którym zwrócony został tytuł aktu prawnego >,
        "references" : [ { < lista nawiązań do innych aktów prawnych >
          "id" : < id aktu prawnego >,
          "name" : < nazwa aktu prawnego >,
          "link" : < link do aktu prawnego >,
          "type_code" : < kod aktu prawnego Us; Ro; Uch; Pu; In >,
          "type_description" : < opis kodu aktu prawnego (Ustawa; Rozporządzenie; Uchwała; Przepisy unijne; Inny >,
          "language" : < język, w którym zwrócony został tytuł nawiązania do aktu prawnego >,
        } ]
      } ],
      "pkd" : {
        "status_code" : < kod statusu PKD (NIE_DOTYCZY; WYBRANE >,
        "status_description" : < opis statusu pobrany ze słownika (np. Poradnik dotyczy wybranych gałęzi PKD) >,
      }
    }
  }
}
```



```

"list" : [ { < tablica obiektów PKD >
  "code" : < kod PKD >,
  "title" : < nazwa gałęzi PKD >,
  "includes_description" : < opis informujący czego dotyczy PKD >,
  "not_includes_description" : < opis informujący czego nie dotyczy PKD >
} ],
},
"language" : < kod języka w którym została zwrócona odpowiedź >
}
},
"language" : < kod języka >,
"status" : {
  "code" : < status odpowiedzi >,
  "message" : < komunikat >
}
}

```

Przykład wywołania

```

http://[adres_usługi]/ApiLifeSituations/getDescription
http://[adres_usługi]/ApiLifeSituations/getDescription?acronym=GABINET_PIELEGNIARSKI_R
http://[adres_usługi]/ApiLifeSituations/getDescription?acronym=205-GABINET_PIELEGNIARSKI_R
http://[adres_usługi]/ApiLifeSituations/getDescription?acronym=205
http://[adres_usługi]/ApiLifeSituations/getDescription?acronym=GABINET_PIELEGNIARSKI_R&lang=en

```

Opis kodów

OK_0000 poprawna odpowiedź
 ERR_0011 - pole [nazwa_pola] jest wymagane.
 ERR_0005 - pole [nazwa_pola] posiada niepoprawny kod języka
 ERR_0006 - pole [nazwa_pola] ma większą długość niż [wymagana_długość] znaków
 ERR_0100 - znaleziono więcej niż jedna sytuację życiową o podanym akronimie
 ERR_5012 - Taka sytuacja życiowa nie istnieje.
 ERR_5017 - Brak opisu dla podanej sytuacji życiowej
 ERR_5018 - niepoprawna struktura akronimu(zawiera niepoprawne znaki)
 ERR_5019 - Przekroczono czas uruchomienia testowej sytuacji życiowej. Należy ponownie uruchomić podgląd sytuacji życiowej z zaplecza
 ERR_5020 - Nieodnaleziono sytuacji życiowej o podanym ID

2.13. linkSessionWithUser Powiązanie sesji sytuacji życiowej z użytkownikiem

Metoda umożliwia połączenie sytuacji życiowej z identyfikatorem użytkownika. Metoda wywoływana w POST.

Nazwa funkcji

linkSessionWithUser

Parametry wejściowe

Parametry wysyłane metodą GET					
Nazwa zmiennej	Typ	obowiązkowy	domyślnie	warunki	opis
hash	string	tak		max 255 znaków	hash
user_id	int	tak			Identyfikator użytkownika

Struktura danych wyjściowych

```

{
  "content" : {
    "status" : "OK"
  },
  "language" : < kod języka >,
  "status" : {
    "code" : < status odpowiedzi >,
    "message" : < komunikat >
  }
}

```

```
}

```

Przykład wywołania

```
http://[adres_uslugi]/ApiLifeSituations/linkSessionWithUser
```

Opis kodów

```
OK_0000 poprawna odpowiedź  
ERR_5011 - Sesja o podanym identyfikatorze jest już powiązana z użytkownikiem
```

2.14. getVisibleLifeSituationsPropositions Pobranie listy widocznych propozycji sytuacji życiowych

Metoda listę wszystkich propozycji sytuacji życiowych o statusie „WIDOCZNY”. Możliwe jest podanie parametru **lang** (domyślnie *pl*), aby zwrócić listę w innym niż domyślnie języku (TODO).

Nazwa funkcji

getVisibleLifeSituationsPropositions

Parametry wejściowe

Parametry wysyłane metodą GET					
Nazwa zmiennej	Typ	obowiązkowy	domyślnie	warunki	opis
lang	string	nie	PL	poprawny kod lang	kod języka w standardzie ISO 639-1

Struktura danych wyjściowych

```
[ {
  "LifeSituationsProposition" : {
    "szp_id" : < status propozycji >,
    "szp_status" : < status propozycji >,
    "szp_liczba_glosow" : < liczba oddanych głosów na propozycję >,
    "szp_temat" : < nazwa propozycji >,
    "szp_widoczny_od" : < data rozpoczęcia wyświetlania >,
    "szp_data_utworzenia" : < data utworzenia >,
    "szp_utworzyl" : < nazwa użytkownika tworzącego propozycję >,
    "szp_data_modyfikacji" : < data modyfikacji propozycji >,
    "szp_modyfikowal" : < nazwa użytkownika modyfikującego >,
    "szp_widoczny_do" : < data graniczna widoczności propozycji >,
    "szp_utworzyl_id" : < identyfikator użytkownika tworzącego propozycję >,
    "szp_modyfikowal_id" : < identyfikator użytkownika modyfikującego propozycję >,
    "id" : < identyfikator propozycji >,
    "szp_widoczny_dni" : < liczba >,
    "created" : < data utworzenia >,
    "created_by" : < osoba która utworzyła propozycję >,
    "modified" : < data modyfikacji >,
    "modified_by" : < osoba która zmodyfikowała propozycję >
  }
} ]
```

Przykład wywołania

```
http://[adres_uslugi]/ApiLifeSituations/getVisibleLifeSituationsPropositions
```

Opis kodów

```
OK_0000 poprawna odpowiedź  
ERR_0005 Parametr „lang” niezgodny ze standardem ISO 639-1
```

2.15. **voteOnLifeSituationProposition** Głosowanie na propozycję sytuacji życiowej

Metoda umożliwiająca oddanie głosu na wybraną propozycję sytuacji życiowej lub dodanie własnej propozycji sytuacji życiowej.

Nazwa funkcji

voteOnLifeSituationProposition

Parametry wejściowe

Parametry wysyłane metodą POST					
Nazwa zmiennej	Typ	obowiązkowy	domyślnie	warunki	opis
id	int	nie			identyfikator propozycji sytuacji życiowej
text	string	nie		max 2000 znaków	temat propozycji sytuacji życiowej
user	string	nie		max 150 znaków	imię i nazwisko osoby zgłaszającej propozycję

Struktura danych wyjściowych

```
{
  content : {
    voteStatus : < status głosu >
  },
  "language" : < kod języka >,
  "status" : {
    "code" : < status_odpowiedzi >,
    "message" : < komunikat >
  }
}
```

Przykład wywołania

Głosowanie na istniejącą propozycję

```
http://[adres_usługi]/ApiLifeSituations/voteOnLifeSituationProposition
W POST
id=12
```

Dodanie własnej propozycji

```
http://[adres_usługi]/ApiLifeSituations/voteOnLifeSituationProposition
W POST
text=To jest moja propozycja
```

Dodanie własnej propozycji przez użytkownika zalogowanego

```
http://[adres_usługi]/ApiLifeSituations/voteOnLifeSituationProposition
W POST
text=Moja propozycja&user=Jan Kowalski
```

Opis kodów

OK_0000 poprawna odpowiedź

ERR_0004 Pole posiada niepoprawny typ. Wymagany typ to

ERR_0006 Pole ma większą długość niż znaków

ERR_2001 Brak propozycji o podanym ID.

ERR_2002 Nie udało się zaaktualizować licznika głosów dla podanej propozycji sytuacji życiowej.

ERR_2003 Nie podano parametrów wejściowych

ERR_0001 Dane należy wysłać metodą POST

3. ApiOpinions

3.1. saveOpinionForContent - Dodawanie opinii

Metoda umożliwiająca dodanie opinii odnośnie artykułu, sytuacji życiowej, publikacji, procedury

Nazwa funkcji

saveOpinionForContent

Parametry wejściowe

Parametry wysyłane metodą POST					
Nazwa zmiennej	Typ	obowiązkowy	domyślnie	warunki	opis
rate	string	tak		na liście (TAK, NIE, TROCHE)	wartość oceny
comment	string	nie		max 5000 znaków	komentarz do oceny
group	string	tak		na liście (ARTYKULY, PUBLIKACJE, SYTUACJE, PROCEDURY, INSTYTUCJE, INNE)	kod określający czego dotyczy opinia
context_id	int	tak, jeśli group != INNE			identyfikator ocenianego elementu
context_title	string	tak, jeśli group == INNE		max 255 znaków; jeśli group != {INNE, ARTYKULY}, to parametr nie jest uwzględniany	tytuł ocenianego elementu
context_link	string	tak, jeśli group == INNE		max 255 znaków; jeśli group != {INNE, ARTYKULY}, to parametr nie jest uwzględniany	link do ocenianego elementu (URL absolutny)
user_id	int	nie			identyfikator użytkownika, który utworzył opinię
user_ip	string	tak		max 255 znaków	adres IP użytkownika dodającego opinię

Struktura danych wyjściowych

```
{
  content : {
    voteStatus : < status głosu >,
    voteCount : < liczba głosów >
  },
  "language" : < kod języka >,
  "status" : {
    "code" : < status_odpowiedzi >,
    "message" : < komunikat >
  }
}
```

Przykład wywołania

Wysłanie opinii z komentarzem

[http://\[adres_usługi\]/ApiOpinions/saveOpinionForContent](http://[adres_usługi]/ApiOpinions/saveOpinionForContent)

W POST

rate=TAK&comment=[Tresc komentarza]&group=ARTYKULY&context_id=1&context_title=Tytuł artykułu&context_link=http://eu-go.gov.pl/artykuly/1&user_ip=234.231.121.2

Wysłanie opinii bez komentarza

[http://\[adres_usługi\]/ApiOpinions/saveOpinionForContent](http://[adres_usługi]/ApiOpinions/saveOpinionForContent)

W POST rate=TAK&group=ARTYKULY&context_id=1&context_title=Tytuł
artykulu&context_link=http://eu-go.gov.pl/artykuly/1&user_ip=234.231.121.2

Opis kodów

OK_0000 poprawna odpowiedź

ERR_0001 Dane należy wysłać metodą POST
ERR_3001 Nie udało się dodać nowej opinii
ERR_0004 - Niepoprawny typ pola
ERR_0006 - Przekroczona dopuszczalna długość pola
ERR_0010 - Pole wykracza poza dopuszczalny zakres
ERR_0011 - Brak pola obligatoryjnego
ERR_0013 - Wartość nie znajduje się na liście
ERR_0021 - Wartość pola jest niepoprawnym adresem IP

4. ApiInstitutions

4.1. saveInstitutionProposition - Zapisywanie propozycji nowej instytucji

Metoda umożliwiająca wysłanie przez użytkownika portalu propozycji nowej instytucji

Nazwa funkcji

saveInstitutionProposition

Parametry wejściowe

Parametry wysyłane metodą POST					
Nazwa zmiennej	Typ	obowiązkowy	domyślnie	warunki	opis
name	string	tak		max 255 znaków	nazwa instytucji
type_id	int	tak			id typu instytucji
nip	string	nie		walidacja NIP	NIP
regon	string	tak		walidacja REGON	REGON
street	string	tak		max 100 znaków	ulica
number	string	nie		max 50 znaków	numer
post_code	string	tak		walidacja kodu pocztowego	kod pocztowy
city	string	tak		max 100 znaków	mięscowość
voivodeship	string	tak		walidacja id województwa	województwo
longitude	string	nie		walidacja współrzędnych	długość geograficzna
latitude	string	nie		walidacja współrzędnych	szerokość geograficzna
email	string	tak		walidacja email, max 256 znaków	email kontaktowy
www	string	tak (jeśli brak pola bip)		walidacja www, max 256 znaków	strona www
bip	string	tak (jeśli brak pola www)		walidacja www, max 256 znaków	strona bip
logo	string	nie		tekst base64, max długość 5*1024*1024 znaków	obraz zakodowany przy użyciu base64
logo_ext	string	nie		max 256 znaków	rozszerzenie pliku logo
activities	string	nie		max 256 znaków	zakres działalności
reported_id	int	tak			id użytkownika wysyłającego zgłoszenie
reported_name	string	nie		max 100 znaków	imię i nazwisko użytkownika wysyłającego zgłoszenie

reported_email	string	nie		walidacja email, max 256 znaków	e-mail użytkownika wysyłającego zgłoszenie
reported_ip	string	nie		walidacja adresu ip	adres IP użytkownika wysyłającego zgłoszenie
workingHours	string	nie		max 256 znaków	godziny otwarcia
phone	string	nie		walidacja numeru telefonu, max 50 znaków	telefon kontaktowy
fax	string	nie		walidacja numeru telefonu, max 50 znaków	fax
other_info	string	nie		max 256 znaków	inne informacje

Struktura danych wyjściowych

```
{
  "content" : {
    "institutionPropositionSaveStatus" : < status >
  },
  "language" : < kod języka >,
  "status" : {
    "code" : < status odpowiedzi >,
    "message" : < komunikat >
  }
}
```

Przykład wywołania

[http://\[adres_usługi\]/ApiInstitutions/saveInstitutionProposition](http://[adres_usługi]/ApiInstitutions/saveInstitutionProposition)

W POST:

```
name=Przykładowy Urząd&type_id=74&nip=9717789134&regon=950056002&street=Powstańców  
Warszawskich&number=5&  
post_code=60-  
343&city=Poznań&voivodeship=30&longitude=16.94091796875&latitude=52.42922227795134&www=htt  
p://przyklad.www.pl&bip=http://przyklad.bip.pl&logo=[logo_base64]&logo_ext=png  
activities=zakres działalności&reported_id=10485&reported_name=Jan  
Kowalski&reported_email=jan@kowalski.pl&  
reported_ip=43.16.1.76&email=przyklad@email.pl&workingHours=>6-  
16&phone=554434442&fax=111111111&  
other_info=inne
```

Opis kodów

OK_0000 poprawna odpowiedź

```
ERR_0001 Dane należy wysłać metodą POST
ERR_0002 Pole %s jest niepoprawnym ciągiem base64.
ERR_0003 Pole %s ma niepoprawny adres e-mail.
ERR_0004 Pole %s posiada niepoprawny typ. Wymagany typ to %s.
ERR_0006 Pole %s ma większą długość niż %s znaków.
ERR_0009 Pole %s posiada niepoprawny numer telefonu.
ERR_0011 Pole %s jest wymagane.
ERR_0012 Pole %s posiada niepoprawny adres URL.
ERR_0013 Pole %s nie znajduje się na liście (%s).
ERR_0014 Pole %s nie posiada prawidłowej wartości dla ID.
ERR_0015 Pole %s posiada nieprawidłowe znaki
ERR_0016 Wartość pola %s jest niepoprawnym ciągiem NIP
ERR_0017 Wartość pola %s jest niepoprawnym ciągiem REGON
```

ERR_0018 Pole %s posiada nieprawidłową sumą kontrolną
 ERR_0019 Wartość pola %s jest niepoprawnym ciągiem kodu pocztowego
 ERR_0020 Wartość pola %s jest niepoprawnym koordynatem Latitude lub Longitude
 ERR_0021 Wartość pola %s jest niepoprawnym adresem IP
 ERR_4002 Nie udało się dodać nowej propozycji instytucji
 ERR_4006 Brak parametru www/bip
 ERR_4008 Przekroczony dopuszczalny rozmiar logo

4.2. getListToPerformElectronicProcedure – Pobranie list instytucji wykonującej elektronicznie procedurę

Metoda umożliwia pobranie listy instytucji wykonujących elektronicznie procedurę

Nazwa funkcji

getListToPerformElectronicProcedure

Parametry wejściowe

Parametry wysyłane metodą GET					
Nazwa zmiennej	Typ	obowiązkowy	domyślnie	warunki	opis
procedure_alias	int	tak			Identyfikator procedury

Struktura danych wyjściowych

```
{
  "content" : {
    "rows_count" : < liczba wszystkich rekordów >,
    "institutions" : [ { < drzewo instytucji >
      "id" : < identyfikator instytucji >,
      "latitude" : < szerokość geograficzna >,
      "longitude" : < długość geograficzna >,
      "name" : < nazwa instytucji >,
      "street" : < ulica >,
      "number" : < numer >,
      "post_code" : < kod pocztowy >,
      "city" : < miasto, kraj jeśli inny niż Polska >
    } ]
  },
  "language" : < kod języka >,
  "status" : {
    "code" : < status odpowiedzi >,
    "message" : < komunikat >
  }
}
```

Przykład wywołania

http://[adres_usługi]/
 ApiInstitutions/getListToPerformElectronicProcedure?procedure_id=1048

Opis kodów

OK_0000 poprawna odpowiedź
 OK_0002 poprawna odpowiedź nie odnaleziono wyników spełniających podane kryterium

ERR_0004 - Niepoprawny typ pola
 ERR_0011 - pole [nazwa_pola] jest wymagane.

4.3. getCategories - Pobranie listy kategorii instytucji

Metoda umożliwiająca pobranie listy kategorii instytucji, które nie są typem specjalnym

Nazwa funkcji

getCategories

Parametry wejściowe

brak

Struktura danych wyjściowych

```
{
  "content" : {
    "UrządTyp" : < typ instytucji >,
    "id" : < id kategorii >,
    "name" : < nazwa kategorii >
  },
  "language" : < kod języka >,
  "status" : {
    "code" : < status odpowiedzi >,
    "message" : < komunikat >
  }
}
```

Przykład wywołania

Pobranie pełnej listy kategorii instytucji

```
http://[adres_usługi]/ApiInstitutions/getCategories
```

Opis kodów

OK_0000 poprawna odpowiedź

4.4. getHints Pobieranie podpowiedzi nazw instytucji

Metoda umożliwiająca pobranie nazw instytucji rozpoczynających się od podanego w parametrze słowa

Nazwa funkcji

getHints

Parametry wejściowe

Parametry wysyłane metodą GET					
Nazwa zmiennej	Typ	obowiązkowy	domyślnie	warunki	opis
search	string	tak		max 100 znaków	początkowy fragment nazwy instytucji
limit	int	nie	10	min wartość 1, max wartość 99999	liczba wyników na stronie

Struktura danych wyjściowych

```
{
```

```

"content" : {
  "name" : < nazwa instytucji >
},
"language" : < kod języka >,
"status" : {
  "code" : < status odpowiedzi >,
  "message" : < komunikat >
}
}

```

Przykład wywołania

Pobranie podpowiedzi 5 instytucji, których nazwa rozpoczyna się od „Izba”

```
http://[adres_usługi]/ApiInstitutions/getHints?search=Izba&limit=5
```

Pobranie 10 (domyślna liczba) instytucji, których nazwa rozpoczyna się od „Izba”

```
http://[adres_usługi]/ApiInstitutions/getHints?search=Izba
```

Opis kodów

OK_0000 - poprawna odpowiedź

ERR_0010 - Pole wykracza poza dopuszczalny zakres

ERR_0004 - Niepoprawny typ pola

ERR_0011 - Brak pola obowiązkowego

ERR_0006 - Przekroczona dopuszczalna długość pola

4.5. getList - Pobieranie listy instytucji

Metoda umożliwiająca pobranie listy instytucji odpowiadających zadanym kryteriom. Nie zwraca instytucji o typie określonym jako typ specjalny. Jeżeli zdefiniowano parametr with_departments jako AUTO, to jeżeli system w odpowiedzi na wprowadzony parametr search nie wyszuka żadnej pasującej instytucji, automatycznie przełączy się na wyszukiwanie po wydziałach instytucji tworząc wynikowo strukturę instytucja -> wydziały. Jeśli parametr with_departments przyjmie wartość 1, pobrane zostaną instytucje posiadające wydziały w przypadku, kiedy nazwa instytucji lub wydziału pasuje do wyszukiwanej frazy. Jeśli dla instytucji odnalezionych zostanie więcej niż 1 wydział (dla with_departments = 1), weight przyjmie wartość najlepiej dopasowanej pary instytucja-wydział na podstawie wyszukiwanej frazy. Gdy parametr zostanie ustawiony na 0, metoda zwróci zawsze tylko urzędy, bez struktury drzewiastej odpowiedzi.

Nazwa funkcji

getList

Parametry wejściowe

Parametry wysłane metodą GET					
Nazwa zmiennej	Typ	obowiązkowy	domyślnie	warunki	opis
search	string	nie		max 100 znaków	wyszukiwana fraza
group	string	nie		na liście (CENTRALNE, SKARBOWE, SADY, ZUS, CECHY, UMG[=urzędy miast i gmin), POZOSTALE	grupa kategorii instytucji
category_id	int	nie			id kategorii instytucji

voivodeship_teryt	string	nie		max 2 znaki	teryt województwa w zakresie którego znajduje się instytucja
city	string	nie		max 100 znaków	miejscowość instytucji
with_departments	String	nie	0	na liście (0, <brak wydziałów> 1, <do listy dołączone będą wydziały pasujące do parametru wyszukiwania serach> AUTO <pobieranie listy bez wydziałów, a jeśli nie znaleziono żadnego, nastąpi wyszukanie wyników z uwzględnieniem nazw wydziałów >)	czy dołączyć wydziały do wyników wyszukiwania
page	int	nie	1	min wartość 1, max wartość 999	numer strony wyszukiwania
limit	int	nie	10	min wartość 1, max wartość 99999	liczba wyników na stronie
sort	string	nie	NAME gdy brak parametru „search”, WEIGHT gdy wprowadzono "search"	na liście ('ID', 'ELECTRONIC_PROCEDURES_COUNT', 'EXECUTABLE_PROCEDURES_COUNT', 'CITY', 'NAME', 'NUMBER', 'POST_CODE', 'STREET', 'WEIGHT') pole sortowania	
order	string	nie	asc	na liście (asc, desc)	kierunek sortowania
range_teryt	string	nie		max 7 znaków	zasięg terytorialny

Struktura danych wyjściowych

```
{
  "content" : {
    "rows_count" : < liczba rekordów ogółem (bez limitów) >,
    "institutions" : [ {
      "electronic_procedures_count" : < liczba procedur elektronicznych powiązanych z instytucją >,
      "executable_procedures_count" : < liczba procedur wykonywanych powiązanych z instytucją >,
      "id" : < id instytucji >,
      "name" : < nazwa instytucji >,
      "executive_name" : < nazwa organu wykonawczego >,
    }
  ]
}
```

```

"city" : < nazwa miejscowości, kraj jeśli inny niż Polska >,
"number" : < numer lokalu >,
"post_code" : < kod pocztowy >,
"street" : < ulica >,
"voivodeship" : < województwo >,
"weight" : < waga sphinx (gdy podano parametr "search") >,
"departments" : [ { < lista znalezionych wydziałów instytucji >
  "id" : < id wydziału >,
  "city" : < nazwa miejscowości, kraj jeśli inny niż Polska >,
  "name" : < nazwa wydziału >,
  "number" : < numer lokalu >,
  "post_code" : < kod pocztowy >,
  "street" : < ulica >,
  "voivodeship" : < województwo >
} ],
} ],
},
"language" : < kod języka >,
"status" : {
  "code" : < status odpowiedzi >,
  "message" : < komunikat >
}
}

```

Przykłady wywołania

```

http://[adres_usługi]/ApiInstitutions/getList?search=izba&group=cechy&category_id=9&voivodeship_teryt=28&city=Leszno&page=2&limit=5&sort=weight&order=asc
http://[adres_usługi]/ApiInstitutions/getList?search=Ministerstwo

```

Opis kodów

OK_0000 - poprawna odpowiedź

ERR_0004 - Niepoprawny typ pola

ERR_0006 - Przekroczona dopuszczalna długość pola

ERR_0010 - Pole wykracza poza dopuszczalny zakres

ERR_0011 - Brak pola obowiązkowego

ERR_0013 - Wartość nie znajduje się na liście

ERR_0014 - Pole nie będące ID

ERR_0035 - Żądana strona wykracza poza listę wyników

4.6. getDetails - Pobieranie szczegółów instytucji

Metoda umożliwiająca pobranie szczegółów instytucji odpowiadających zadanym kryteriom.

Nazwa funkcji

getDetails

Parametry wejściowe

Parametry wysłane metodą GET					
Nazwa zmiennej	Typ	obowiązkowy	domyślnie	warunki	opis
id	int	tak			identyfikator instytucji

Struktura danych wyjściowych

```

{
  "content" : {
    "Institution" : {

```

```
"electronic_procedures_count" : < liczba procedur elektronicznych powiązanych z
instytucją >,
"executable_procedures_count" : < liczba procedur wykonywanych powiązanych z
instytucją >,
"id" : < id instytucji >,
"name" : < nazwa instytucji >,
"executive_name" : < nazwa organu wykonawczego >
},
"InstitutionType" : {
  "type_id" : < identyfikator typu instytucji >,
  "type_name" : < nazwa typu instytucji >
},
"InstitutionDetails" : {
  "shortcut" : < skrót instytucji >,
  "nip" : < NIP >,
  "regon" : < REGON >,
  "street" : < ulica >,
  "number" : < numer lokalu >,
  "post_code" : < kod pocztowy >,
  "city" : < miejscowość, kraj jeśli inny niż Polska >,
  "voivodeship_teryt_id" : < identyfikator terytu >,
  "working_hours" : < godziny pracy instytucji >,
  "additional_info" : < dodatkowe informacje; m.in. zakres działalności >,
  "email" : < email >,
  "phone" : < telefon >,
  "fax" : < fax >,
  "www" : < strona www >,
  "bip" : < strona bip >,
  "logo" : < herb/logo instytucji >,
  "latitude" : < szerokość geograficzna >,
  "longitude" : < długość geograficzna >,
  "esp" : < nazwa skrzynki podawczej; tymczasowo nazwa epuap >,
  "epuap_id" : < identyfikator instytucji w epuap >,
  "general_eservice" : < adres e-usługi pisma ogólnego >,
  "responsible_procedures_count" : < liczba procedur za które instytucja jest
odpowiedzialna >,
  "queue_system_link" : < link do systemu kolejkowego >,
  "queue_system_description" : < opis systemu kolejkowego >
},
"InstitutionDepartments" : [ { < wydziały instytucji >
  "Department" : {
    "id" : < identyfikator wydziału >,
    "name" : < nazwa wydziału >
  }
} ],
"InstitutionPkd" : [ {
  "pkd_kod" : < kod pkd >,
  "pkd_opis" : < opis pkd >
} ],
"InstitutionTerritoryRange" : [ {
  "TerritoryRange" : {
    "voivodeship" : < województwo >,
    "district" : < powiat >,
    "community" : < gmina >,
    "community_kind" : < rodzaj gminy >,
    "teryt" : < kod teryt >
  }
} ]
},
"language" : < kod języka >,
"status" : {
  "code" : < status odpowiedzi >,
  "message" : < komunikat >
}
}
```


Przykłady wywołania

[http://\[adres_usługi\]/ApiInstitutions/getDetails?id=1322](http://[adres_usługi]/ApiInstitutions/getDetails?id=1322)

Opis kodów

OK_0000 - poprawna odpowiedź

ERR_0011 - Brak pola obligatoryjnego

ERR_0014 - Pole nie będące ID

ERR_0033 - Instytucja o podanym ID nie istnieje

4.7. getDepartmentDetails - Pobieranie szczegółów wydziału instytucji

Metoda umożliwiająca pobranie szczegółów wydziału danej instytucji

Nazwa funkcji

getDepartmentDetails

Parametry wejściowe

Parametry wysyłane metodą GET					
Nazwa zmiennej	Typ	obowiązkowy	domyślnie	warunki	opis
id	int	tak			identyfikator wydziału instytucji

Struktura danych wyjściowych

```
{
  "content" : {
    "Department" : {
      "electronic_procedures_count" : < liczba procedur elektronicznych powiązanych z
      wydziałem >,
      "executable_procedures_count" : < liczba procedur wykonywanych powiązanych z
      wydziałem >,
      "id" : < id wydziału instytucji >,
      "name" : < nazwa wydziału instytucji >
    },
    "DepartmentDetails" : {
      "shortcut" : < skrót >,
      "street" : < ulica >,
      "number" : < numer lokalu >,
      "post_code" : < kod pocztowy >,
      "city" : < miejscowość, kraj jeśli inny niż Polska >,
      "voivodeship_teryt_id" : < identyfikator terytu >,
      "working_hours" : < godziny pracy wydziału >,
      "additional_info" : < dodatkowe informacje, m.in. zakres działalności >,
      "email" : < email >,
      "phone" : < telefon >,
      "fax" : < fax >,
      "www" : < strona www >,
      "bip" : < strona bip >,
      "latitude" : < szerokość geograficzna >,
      "longitude" : < długość geograficzna >,
      "is_queue_system_central" : < czy wydział korzysta z systemu kolejkowego instytucji,
      do której należy >,
      "queue_system_link" : < link do systemu kolejkowego >,
      "queue_system_description" : < opis systemu kolejkowego >
    },
    "ParentInstitution" : {
      "id" : < id instytucji >,

```

```

"name" : < nazwa instytucji >,
"general_eservice" : < adres e-usługi pisma ogólnego instytucji >,
"logo" : < herb/logo instytucji >,
},
" InstitutionTerritoryRange" : [ {
  "TerritoryRange" : {
    "voivodeship" : < województwo >,
    "district" : < powiat >,
    "community" : < gmina >,
    "community_kind" : < rodzaj gminy >,
    "teryt" : < kod teryt >
  }
} ]
},
"language" : < kod języka >,
"status" : {
  "code" : < status odpowiedzi >,
  "message" : < komunikat >
}
}

```

Przykłady wywołania

[http://\[adres_usługi\]/ApiInstitutions/getDepartmentDetails?id=1322](http://[adres_usługi]/ApiInstitutions/getDepartmentDetails?id=1322)

Opis kodów

OK_0000 - poprawna odpowiedź

ERR_0011 - Brak pola obligatoryjnego

ERR_0014 - Pole nie będące ID

ERR_0032 - Wydział o podanym ID nie istnieje

4.8. getListByTeryt - Pobieranie listy jednostek samorządu terytorialnego

Metoda umożliwiająca pobranie listy jednostek samorządu terytorialnego (starostwa powiatowe, urzędy marszałkowskie, urzędy wojewódzkie, urzędy miast i gmin) dla wybranego obszaru teryt

Nazwa funkcji

getListByTeryt

Parametry wejściowe

Parametry wysyłane metodą GET					
Nazwa zmiennej	Typ	obowiązkowy	domyślnie	warunki	opis
teryt	string	tak		max 6 znaków	kod teryt

Struktura danych wyjściowych

```

{
  "content" : {
    "electronic_procedures_count" : < liczba procedur elektronicznych powiązanych z
instytucją >,
    "executable_procedures_count" : < liczba procedur wykonywanych powiązanych z instytucją
>,
    "id" : < id instytucji >,
    "city" : < nazwa miejscowości >,
    "name" : < nazwa instytucji >,
    "number" : < numer lokalu >,
    "post_code" : < kod pocztowy >,

```

```
"street" : < ulica >,  
},  
"language" : < kod języka >,  
"status" : {  
  "code" : < status odpowiedzi >,  
  "message" : < komunikat >  
}  
}
```

Przykłady wywołania

```
http://[adres_usługi]/ApiInstitutions/getListByTeryt?teryt=306402
```

Opis kodów

OK_0000 - poprawna odpowiedź
ERR_0006 - Przekroczona dopuszczalna długość pola
ERR_0011 - Brak pola obligatoryjnego

5. ApiProcedures

5.1. Wprowadzenie

5.1.1. Zastosowanie parametru ALIAS zamiast ID w wywołaniach

Główne wywołanie procedur opiera się o tzw. **ALIAS** czyli parametr składający się standardowo z głównego, statycznego identyfikatora procedury, dawniej identyfikatora UEPA, niezależnego od wersji, oraz nazwy skróconej procedury pozbawionej znaków specjalnych, spacji i znaków diakrytycznych. Nazwa może zmieniać się w kolejnych wersjach publikowanej procedury. Jeżeli alias jest parametrem wejściowym dla metod, system dopuszcza podanie jedynie ID nie badając pozostałego ciągu znaków. Wyjątkiem od tej reguły jest wywołanie niestandardowe dla `getDetails`, gdzie podobnie jak dla innych części systemu (sytuacje życiowe, publikacje) można wymusić wyświetlenie konkretnej wersji procedury niezależnie od jej statusu. Alias wówczas zawiera prefix „TEST”, ID wersji procedury, znacznik czasu, prawidłowy alias.

Portal w celu zachowania poprawnych linków, oraz co za tym idzie poprawności indeksowania w wyszukiwarkach powinien w przypadku wywołania procedury z niekompletnym lub niewłaściwym aliasem automatycznie przesyłać do przeglądarki żądanie zmiany 301 (Move Permanently) na właściwy alias. Bieżący, właściwy alias zawsze znajduje się w odpowiedziach API.

Przykład wywołania API `getDetails` z różnym zastosowaniem parametru ALIAS

Wywołanie standardowe

```
http://[adres_usługi]/ ApiProcedures/getDetails?alias=111-wpis-do-rejestru-operatorow-pocztowych
```

Wywołanie bez nazwy aliasu

```
http://[adres_usługi]/ ApiProcedures/getDetails?alias=111
```

Wywołanie elementu o konkretnym id wersji

```
http://[adres_usługi]/ApiProcedures/getDetails?alias=*TEST-339-1413981229*111-wpis-do-rejestru-operatorow-pocztowych
```

5.2. `getHints` - Pobieranie odpowiedzi nazw procedur

Metoda umożliwiająca pobranie nazw procedur zawierających podany w parametrze ciąg znaków

Nazwa funkcji

`getHints`

Parametry wejściowe

Parametry wysłane metodą GET					
Nazwa zmiennej	Typ	obowiązkowy	domyślnie	warunki	opis
search	string	tak		max 100 znaków	fragment nazwy procedury
limit	int	nie	10	min wartość 1, max	maksymalna liczba zwróconych

				wartość 99999	wyników
lang	string	nie	PL	kod języka	język, w jakim mają być zwrócone procedury
electronic_only	int	nie	0	na liście (0, 1)	czy mają być zwrócone tylko procedury elektroniczne

Struktura danych wyjściowych

```
{
  "content" : {
    "name" : < nazwa procedury >
  },
  "language" : < kod języka >,
  "status" : {
    "code" : < status odpowiedzi >,
    "message" : < komunikat >
  }
}
```

Przykład wywołania

Pobranie podpowiedzi 5 procedur elektronicznych, zawierających słowo "zezwolenie"

[http://\[adres_usługi\]/ApiProcedures/getHints?search=zezwolenie&limit=5](http://[adres_usługi]/ApiProcedures/getHints?search=zezwolenie&limit=5)

Pobranie 10 (domyślna liczba) procedur w języku angielskim, zawierających frazę "trans"

[http://\[adres_usługi\]/ApiInstitutions/getHints?search=transport&lang=EN](http://[adres_usługi]/ApiInstitutions/getHints?search=transport&lang=EN)

Opis kodów

OK_0000 - poprawna odpowiedź
 OK_0002 - poprawna odpowiedź, brak wyników
 ERR_0004 - Niepoprawny typ pola
 ERR_0005 - Niepoprawny kod języka
 ERR_0006 - Przekroczona dopuszczalna długość pola
 ERR_0010 - Pole wykracza poza dopuszczalny zakres
 ERR_0011 - Brak pola obowiązkowego
 ERR_0013 - Element nie znajduje się na liście

5.3. getList - Pobieranie listy procedur

Metoda umożliwiająca pobranie listy procedur wg zadanych kryteriów wyszukiwania. W przypadku podania parametru lang innego niż 'PL' oraz braku parametrów 'search' i 'letter' pobierana jest lista procedur w języku obcym odpowiadająca kryteriom, a w przypadku braku opublikowanej wersji tłumaczenia zwracana jest wersja polska procedury. W sytuacji, gdy podany jest parametr electronic_only i parametr institution_id, otrzymamy w wyniku procedury, które są elektroniczne w kontekście podanego urzędu.

Nazwa funkcji

getList

Wersja

V2

Parametry wejściowe

Parametry wysyłane metodą GET					
Nazwa zmiennej	Typ	obowiązkowy	domyślnie	warunki	opis
search	string	nie		max 300 znaków	nazwa procedury
pkd	string	nie			lista kodów pkd powiązanych z procedurami, oddzielona przecinkami
type	string	nie		na liście (TYP_UZ_KW_ZA, TYP_SW_US_TR, TYP_ZA_REG, TYP_DZ_REG, TYP_ZE_PO_FO_REGL, TYP_INN)	lista typów procedury oddzielona przecinkami. Tłumaczenie kodów: TYP_UZ_KW_ZA - Uznawanie kwalifikacji zawodowych; TYP_SW_US_TR - Świadczenie usług transgranicznych; TYP_ZA_REG - Zawody, w tym regulowane; TYP_DZ_REG - Działalność regulowana; TYP_ZE_PO_FO_REGL - Zezwolenia i pozostałe formy reglamentacji; TYP_INN - Inne
starting_business_only	int	nie	0	na liście (0, 1)	czy wyszukiwać tylko procedury związane z zakładaniem działalności gospodarczej
electronic_only	int	nie	0	na liście (0, 1)	czy wyszukiwać tylko procedury elektroniczne
category_id	int	nie			id kategorii procedury
profession_id	int	nie			id zawodu
letter	string	nie			litera, od której rozpoczyna się nazwa procedury
lang	string	nie	PL	kod języka	język tłumaczenia procedury
institution_id	int	nie			id instytucji

					wykonywanej powiązanej z procedurami
page	int	nie	1	min wartość 1, max wartość 999	numer strony wyszukiwania
limit	int	nie	10	min wartość 1, max wartość 99999	liczba wyników na stronie
sort	string	nie	NAME	na liście (ID, ELECTRONIC, NAME, WEIGHT, PUBLICATION_DATE, MODIFICATION_DATE)	pole sortowania
order	string	nie	ASC	na liście (ASC, DESC)	kierunek sortowania
department_id	int	nie			Id wydziału powiązanego z procedurami
epk_only	int	nie	0	na liście (0, 1)	czy wyszukiwać tylko procedury, które posiadają e-usługę typu EPK (zawierającą w adresie url fragment biznes.gov.pl)

Struktura danych wyjściowych

```
{
  "content" : {
    "procedures" : [ {
      "electronic" : < czy procedura elektroniczna >,
      "id" : < id procedury >,
      "alias" : < alias procedury >,
      "name" : < nazwa procedury >,
      "weight" : < waga >,
      "language" : < kod języka procedury >,
      "modification_date" : < data modyfikacji >,
      "short_introduction" : < zawartość pola „Krótkie wprowadzenie” >
    } ],
    "rows_count" : < liczba rekordów ogółem >
  },
  "language" : < kod języka >,
  "status" : {
    "code" : < status odpowiedzi >,
    "message" : < komunikat >
  }
}
```

Przykład wywołania

Pobranie listy procedur bez kryteriów, z domyślną liczbą wyników 10

[http://\[adres_uslugi\]/ApiProcedures/getList](http://[adres_uslugi]/ApiProcedures/getList)

Pobranie listy 50 procedur elektronicznych

[http://\[adres_uslugi\]/ApiProcedures/getList?limit=50&electronic_only=1](http://[adres_uslugi]/ApiProcedures/getList?limit=50&electronic_only=1)

Pobranie listy procedur powiązanych z zawodem o id 8 rozpoczynających się na literę A

[http://\[adres usługi\]/ApiProcedures/getList?profession_id=8&letter=a](http://[adres usługi]/ApiProcedures/getList?profession_id=8&letter=a)

Opis kodów

OK_0000 - poprawna odpowiedź
 OK_0002 - poprawna odpowiedź, brak wyników

 ERR_0004 - Niepoprawny typ pola
 ERR_0005 - Niepoprawny kod języka
 ERR_0006 - Przekroczona dopuszczalna długość pola
 ERR_0010 - Pole wykracza poza dopuszczalny zakres
 ERR_0013 - Element nie znajduje się na liście
 ERR_0014 - Pole nie jest ID
 ERR_0023 - Niepoprawna lista PKD
 ERR_0024 - Niepoprawna lista typów procedury
 ERR_0035 - Żądana strona wykracza poza listę wyników

5.4. getElectronicList - Pobieranie listy procedur elektronicznych

Metoda umożliwiająca pobranie listy procedur elektronicznych wg zadanych kryteriów wyszukiwania wraz z powiązanymi instytucjami

Nazwa funkcji

getElectronicList

Parametry wejściowe

Parametry wysyłane metodą GET					
Nazwa zmiennej	Typ	obowiązkowy	domyślnie	warunki	opis
search	string	nie		max 100 znaków	nazwa procedury
pkd	string	nie			lista kodów pkd powiązanych z procedurami, oddzielona przecinkami
type	string	nie		na liście (DEF_CSU, DEF_DR, DEF_SUTG, DEF_UKZ, DEF_ZIIFR, DEF_ZR)	lista typów procedury oddzielona przecinkami. Tłumaczenie kodów: DEF_CSU - Czasowe Świadczenie Usług; DEF_DR - Działalność Regulowana; DEF_SUTG - Świadczenie Usług Transgranicznych; DEF_UKZ - Uznawanie Kwalifikacji Zawodowych; DEF_ZIIFR - Zezwolenia I Inne Formy Reglamentacji; DEF_ZR - Zawody, w tym Regulowane
lang	string	nie	PL	kod języka	język tłumaczenia procedury
page	int	nie	1	min wartość 1, max wartość 999	numer strony wyszukiwania
limit	int	nie	10	min wartość 1, max wartość 99999	liczba wyników na stronie
sort	string	nie	NAME	na liście (ID, ELECTRONIC, NAME, WEIGHT)	pole sortowania

order	string	nie	ASC	na liście (ASC, DESC)	kierunek sortowania
-------	--------	-----	-----	-----------------------	---------------------

Struktura danych wyjściowych

```
{
  "content" : {
    "procedures" : [ {
      "electronic" : < czy procedura elektroniczna >,
      "id" : < id procedury >,
      "alias" : < alias procedury >,
      "name" : < nazwa procedury >,
      "weight" : < waga >,
      "language" : < kod języka >,
      "Institutions" : [ {
        "inst_id" : < id instytucji >,
        "inst_nazwa" : < nazwa instytucji >
      } ]
    } ]
  },
  "rows_count" : < liczba rekordów ogółem >
},
"language" : < kod języka >,
"status" : {
  "code" : < status odpowiedzi >,
  "message" : < komunikat >
}
}
```

Przykład wywołania

Pobranie listy procedur elektronicznych bez kryteriów, z domyślną liczbą wyników 10

[http://\[adres_usługi\]/ApiProcedures/getElectronicList](http://[adres_usługi]/ApiProcedures/getElectronicList)

Pobranie listy 50 procedur elektronicznych zawierających frazę "zezwo"

[http://\[adres_usługi\]/ApiProcedures/getElectronicList?limit=50&search=zezwo](http://[adres_usługi]/ApiProcedures/getElectronicList?limit=50&search=zezwo)

Opis kodów

OK_0000 - poprawna odpowiedź
 OK_0002 - poprawna odpowiedź, brak wyników

ERR_0004 - Niepoprawny typ pola
 ERR_0005 - Niepoprawny kod języka
 ERR_0006 - Przekroczona dopuszczalna długość pola
 ERR_0010 - Pole wykracza poza dopuszczalny zakres
 ERR_0013 - Element nie znajduje się na liście
 ERR_0023 - Niepoprawna lista PKD
 ERR_0024 - Niepoprawna lista typów procedury
 ERR_0035 - Żądana strona wykracza poza listę wyników

5.5. getDetails - Pobieranie szczegółów procedury

Metoda pobierająca szczegóły procedury. W wyniku otrzymamy szczegóły procedury wraz z informacją czy procedurę można wykonać elektronicznie oraz ewentualnie urzędem wykonującym procedurę.

Nazwa funkcji

getDetails

Parametry wejściowe

Parametry wysłane metodą GET					
Nazwa zmiennej	Typ	obowiązkowy	domyślnie	warunki	opis
alias	string	tak		max 255 znaków	alias procedury (id-name_abbreviation)
institution_id	int	nie		?	identyfikator instytucji
teryt	string	nie		?	kod TERYT miejscowości, dla której ma wyświetlany być opis procedury
lang	string	nie	PL	?	język opisu procedury
type		nie		?	dla kogo ta procedura
check_if_anyone_has_epuap	int	nie	0	na liście (0, 1)	Metoda przy włączonym parametrze sprawdza, czy jakkolwiek instytucja w obrębie instytucji wykonujących dla tej procedury posiada wprowadzony identyfikator podmiotu ePUAP. Informacja wyświetlana jest w wyniku w polu "anyone_has_epuap"
sym	string	nie		7 cyfr	Kod SYM z bazy SIMC miejscowości, dla której ma być wyświetlony opis procedury

Struktura danych wyjściowych

```
{
  "content" : {
    "procedure_name" : {
      "name" : < nazwa procedury >,
      "name_abbreviation" : < skrót nazwy procedury >,
      "keywords" : < słowa kluczowe >,
      "keywords_additional" : < słowa kluczowe dodatkowe >,
      "alias" : < poprawny alias procedury >,
      "id" : < id procedury >,
      "version_id" : < id wersji procedury >,
      "first_publication_date" : < data pierwszej publikacji >,
      "last_modification_date" : < data ostatniej modyfikacji >,
      "is_expired" : < czy procedura jest oznaczona jako nieaktualna (true/false) >,
      "confirmation_date" : < data zatwierdzenia opisu procedury przez Instytucję Centralną >,
      "verification_date" : < data świadomego potwierdzenia aktualności całego opisu procedury przez Instytucję Centralną >,
      "recall_date" : < data planowanego wycofania >,
      "responsible_institution_id" : < identyfikator instytucji odpowiedzialnej za opis procedury >,
      "responsible_institution_name" : < nazwa instytucji >,
      "publication_status_code" : < kod procedury (ETA; EZA; IZA) >,
      "publication_status_code_description" : < opis do pola code >,
      "accepting_institution_id" : < identyfikator instytucji odpowiedzialnej za opis procedury >,
      "accepting_institution_name" : < nazwa instytucji >,
      "institution_verification_date" : < data przekazania do weryfikacji instytucji >
    },
  },
}
```

```
"procedure_execution_type" : < rodzaj procedury ze względu na instytucję wykonującą  
(CENTRAL-procedura wykonywana przez jedną instytucję centralną; RANGE-procedura wykonywana  
przez instytucję przypisaną do wybranego obszaru >,  
"procedure_execution" : {  
  "institution" : { < dane wskazanej instytucji w której można zrealizować procedurę  
>  
    "id" : < id instytucji >,  
    "name" : < nazwa instytucji >,  
    "street" : < ulica >,  
    "number" : < numer domu >,  
    "post_code" : < kod pocztowy >,  
    "city" : < miasto, kraj jeśli inny niż Polska >,  
    "voivodeship_teryt" : < teryt województwa >,  
    "voivodeship" : < nazwa województwa >,  
    "department" : { < wydział instytucji >  
      "id" : < id wydziału >,  
      "name" : < nazwa wydziału >,  
      "street" : < ulica >,  
      "number" : < numer domu >,  
      "post_code" : < kod pocztowy >,  
      "city" : < miasto, kraj jeśli inny niż Polska >,  
      "voivodeship_teryt" : < teryt województwa >  
    }  
  }  
},  
===== WERSJA V2 =====  
"eservices" : [ { < usługi elektroniczne >  
  "link" : < link do usługi >,  
  "description" : < opis usługi >,  
  "category" : < kod określający typ usługi (epk-usługa specjalna;  
executing_institution-usługa zdefiniowana przez instytucję realizującą; central-usługa  
centralna; general_eservice - usługa oparta o pismo ogólne >  
} ]  
===== WERSJA V3 =====  
"how_to_execute" : { < w jaki sposób można zrealizować usługę >  
  "options" : [ <lista opcji sposobów realizacji usługi - podzbiór INSTITUTION,  
MAIL, ONLINE> ],  
  "additional_description" : <dotatkowy opis który jeśli niepusty to można  
wykorzystać do opisanie innych sposobów załatwienia sprawy lub barier w powyższych  
sposobach>  
  "eservices" : [ { < usługi elektroniczne >  
    "link" : < link do usługi >,  
    "description" : < opis usługi >,  
    "category" : < kod określający typ usługi (epk-usługa specjalna;  
executing_institution-usługa zdefiniowana przez instytucję realizującą; central-usługa  
centralna; general_eservice - usługa oparta o pismo ogólne >  
  } ]  
},  
  "room_person" : < pokój / osoba >  
===== KONIEC V3 =====  
"anyone_has_epuap" : <czy istnieje min. jedna instytucja z podmiotem ePUAP (TRUE,  
FALSE, NULL - jeśli metoda wywołana bez parametru 'check_if_anyone_has_epuap') >  
},  
"procedure_general" : { < dane ogólne o procedurze >  
  "description_code" : < typ opisu procedury (OPIS_PELNY, OPIS_SKROCONY) >  
  "type" : < typ procedury >,  
  "purpose" : < cel procedury >,  
  "statutory_duration" : { < ustawowy czas realizacji procedury >  
    "code" : < kod typu czasu realizacji >,  
    "code_description" : < opis kodu >,  
    "description" : < opis gdy kod typu ustawiony na opis >  
  }  
},
```

```
"has_fees" : < czy procedura posiada opłaty (true/false), wartość pola uzależniona
jest od wartości pola description_code; gdy description_code == OPIS_SKROCONY to has_fees =
null; gdy description_code == OPIS_PELNY to has_fees = (true/false) >,
"pkd" : {
  "status_code" : < kod statusu PKD (NIE_DOTYCZY; WYBRANE; WSZYSTKIE >,
  "status_description" : < opis statusu pobrany ze słownika (np. Procedura dotyczy
wybranych gałęzi PKD) >,
  "list" : [ { < tablica obiektów PKD >
    "code" : < kod PKD >,
    "title" : < nazwa gałęzi PKD >,
    "hint" : < odpowiedź do PKD >,
    "includes_description" : < opis informujący czego dotyczy PKD >,
    "not_includes_description" : < opis informujący czego nie dotyczy PKD >
  } ],
},
"participants" : { < tablica uczestników >
  "name" : < wnioskodawca; nazwa >,
  "name_additional" : < nazwa dodatkowa >
},
"local_jurisdiction" : { < właściwość miejscowa >
  "for_entrepreneur" : < kod właściwości miejscowej
(ZASP_NIE_DOTYCZY; ZASP_MIE_WYK; ZASP_SIEDZ; ZASP_DOW_URZ; ZASP_LOK_PRZEDS;
ZASP_MIE_WYK_SIEDZ; ZASP_SIE_LUB_MIE; ZASP_LOK_SPRZED; ZASP_PORT_ZALAD; ZASP_MIE_POL_OB;
ZASP_MIE_WYK_RO; ZASP_MIE_POL_UPR; ZASP_POL_PLA_NAS; ZASP_PORT_MAC; ZASP_MIE_PROW_POL;
ZASP_KORZ_SRO; ZASP_ODP_KOM; ZASP_MIE_ZAM; ZASP_MIE_INW_BUD; ZASP_MIE_ROZ; ZASP_POL_BUD;
ZASP_LOK_ZJA; ZASP_LOK_ZAJ_PDROG; ZASP_POL_NIE; ZASP_POL_DRZE; ZASP_LOK_INST;
ZASP_MIE_POL_ZAB; ZASP_MIE_WYZBPR_ODP; ZASP_MIE_IMP_MAS; ZASP_URZ_SKR_PRZE;
ZASP_MIE_ODP_CEL; ZASP_MIE_PRO_KSI_RACH) >,
  "for_citizen" : < kod właściwości miejscowej
(ZASO_NIE_DOTYCZY; ZASO_MIE_ZAMEL; ZASO_MIE_ZAMIE; ZASO_DOW_URZ; ZASO_LOK_PRZEDS;
ZASO_MIE_ZAMEL_ZAMIE; ZASO_MIE_WYK_ZAW; ZASO_MIE_POB_CUDZ; ZASO_MIE_WYK_ROB;
ZASO_MIE_POL_UPR; ZASO_POL_PLA_NAS; ZASO_PORT_MAC; ZASO_MIE_PRO_POL; ZASO_KORZ_SRO;
ZASO_ODP_KOM; ZASO_MIE_INW_BUD; ZASO_MIE_ROZ; ZASO_POL_BUD; ZASO_LOK_ZJA;
ZASO_LOK_ZAJ_PDROG; ZASO_POL_NIE; ZASO_POL_DRZE; ZASO_LOK_INST; ZASO_MIE_POL_ZAB;
ZASO_MIE_IMP_MAS; ZASO_MIE_ODP_CEL; ZASO_MIE_POL_OBI) >
},
  "institutions_implementing_hint" : < odpowiedź dla użytkownika dotycząca wyboru
instytucji >,
  "institutions_implementing_in_range" : < { < lista id instytucji wykonujących
procedurę dla wskazanego obszaru (wymaga podania kodu teryt w parametrach wejściowych) lub
null gdy przypisana jest instytucja o typie specjalnym >
    "id" : < id instytucji >,
    "name" : < nazwa instytucji >
  } ],
  "institutions_implementing_all" : { < lista id instytucji i typów instytucji
realizujących procedurę wskazanych na formularzu opisu procedury >
    "units" : [ {
      "id" : < id instytucji >,
      "name" : < nazwa instytucji >
    } ],
    "types" : [ { < null gdy została przypisana instytucja o typie specjalnym >
      "id" : < id typu instytucji >,
      "name" : < nazwa typu instytucji (np. urzędy miast i gmin) >
    } ],
  },
"procedure_steps" : { < tablica kroki procedury >
  "id" : < identyfikator >,
  "description" : < opis >,
  "number" : < numer kroku >,
  "type_code" : < kod typu kroku (W - złożenie wniosku; B - inny) >,
  "type_description" : < typ kroku (złożenie wniosku; inny) >,
  "title" : , < tytuł >
  "details" : {
    "documents" : {
```

```

"documents_description" : < opis dotyczący dokumentów podpiętych pod procedurę
>,
  "list" : [ { < lista dokumentów >
    "title" : < tytuł dokumentu >,
    "description" : < opis dokumentu >,
    "id" : < identyfikator dokumentu >,
    "context_code" : < kontekst dokumentu: WE-wejściowy, WY-wyjściowy, ZA-
załącznik >,
    "related_procedures" : [ { < lista procedur powiązanych poprzez dokument >
      "id" : < identyfikator procedury >,
      "alias" : < alias powiązanej procedury >,
      "name" : < nazwa procedury >
    } ],
    "attachments" : {
      "central" : [ { < lista załączników centralnych >
        "name" : < nazwa załącznika >,
        "type_code" : < typ załącznika (WZOR_PAPIEROWY; WZOR_ELEKTRONICZNY;
PRZYKŁAD_ELEKTRONICZNY; PRZYKŁAD_PAPIEROWY; INSTRUKCJA) >,
        "type_description" : < opis kodu załącznika (Wzór papierowy; Wzór
elektroniczny; Przykład elektroniczny; Przykład papierowy; Instrukcja) >,
        "link" : < link do załącznika >,
        "visible" : < true/false >,
      } ],
      "executing_institution" : [ {
        "name" : < nazwa załącznika >,
        "type_code" : < typ załącznika, opis typów j.w. >,
        "type_description" : < typ opisu >,
        "link" : < link do załącznika >,
        "visible" : < true/false >,
      } ],
    },
    "forms" : [ {
      "form_code" : < kod formy (Ko; Ut; Or; Pe; Dw; Uk) >,
      "form_description" : < opis formy >
    } ],
  } ],
  "fees" : {
    "value" : < wartość >,
    "currency" : < waluta (PLN, EUR) >,
    "description" : < opis >,
    "type_code" : < kod określający status uzupełnienia opłaty (central-opłata
ustalona przez instytucję centralną; waiting_for_exec_inst-opłata ustalona przez instytucję
realizującą ale jeszcze nie uzupełniona przez tę instytucję; executing_institution-opłata
ustalona przez instytucję realizującą już wypełniona przez instytucję >,
    "bank_account_number" : < numer rachunku bankowego instytucji realizującej >,
  },
  "deadline" : {
    "code" : < opis; nie_dotyczy >,
    "code_description" : < opis kodu typu (opis; nie_dotyczy) >,
    "description" : < działa dla typu opis >
  },
  "participants" : [ < pole nie występuje w przypadku braku uczestników > {
    "name" : < nazwa >,
    "name_additional" : < nazwa typu; nazwa urzędu; null >
  } ],
  "eservices" : [ {
    "link" : < link do usługi >,
    "description" : < opis usługi >,
    "type_code" : < kod określający typ usługi (epk-usługa specjalna;
executing_institution-usługa zdefiniowana przez instytucję realizującą; central-usługa
centralna) >
  } ]
}
},
"acts" : [ { < tablica aktów >

```



```

"id" : ,
"name" : ,
"link" : ,
"type_code" : < kod aktu prawnego Us; Ro; Uch; Pu; In >,
"type_description" : < opis kodu aktu prawnego (Ustawa; Rozporządzenie; Uchwała;
Przepisyunijne; Inny >,
"references" : [ { < lista nawiązań do innych aktów prawnych >
  "id" : < id aktu prawnego >,
  "name" : < link > : ,
  "type_code" : < kod aktu prawnego >,
  "type_description" : < opis kodu aktu prawnego >
} ]
} ],
"conditions" : {
  "input_conditions" : < warunki wejściowe >,
  "result" : < rezultat procedury !!! Uwaga to pole jest nieużywane >
},
"additional_information" : { < dodatkowe informacje >
  "alternative_process" : < przebieg alternatywny !!! Uwaga to pole jest nieużywane >,
  "additional_information" : < dodatkowe informacje >,
  "prevention_means" : < środki odwoławcze >,
},
"references" : { < Odwołania do innych procedur; publikacji; poradników >
  "with_procedures" : [ { < lista procedur zbliżonych tematycznie >
    "id" : < id powiązanej procedury >,
    "alias" : < alias powiązanej procedury >,
    "name" : < nazwa powiązanej procedury >
  } ],
  "with_publications" : [ { < lista publikacji w których mowa o tej procedurze >
    "id" : < id powiązanej publikacji >,
    "alias" : < alias powiązanej publikacji >,
    "name" : < nazwa powiązanej publikacji >,
  } ],
  "with_lifesituations" : [ { < lista poradników w których mowa o tej procedurze >
    "id" : < id powiązanego poradnika >,
    "alias" : < alias powiązanego poradnika >,
    "name" : < nazwa powiązanego poradnika >
  } ],
  "business_lifecycle_name" : < nazwa działalności dla cyklu życia, np.: "Cykl życia
działalności związanej [np.: z wpisem do rejestru działalności szkoleniowej]" >,
  "business_lifecycle" : [ { < lista procedur związana z cyklem życia działalności >
    "type" : < kod podtypu na procedurze: rozpoczęcie, prowadzenie, zakończenie
działalności (ROZ; PRO; ZAK) >,
    "id" : < id powiązanej procedury >,
    "alias" : < alias powiązanej procedury >,
    "name" : < nazwa powiązanej procedury >
  } ]
},
===== WERSJA V4 =====
"localization_name" : <mięscowość dla której wyświetlana jest procedura, nazwa
miejsowości wyświetlana użytkownikowi w standardzie: miejscowość (gmina <nazwa>, woj.
<NAZWA_WOJEWODZTWA>) lub gmina: <nazwa> (powiat <nazwa>, woj. <NAZWA_WOJEWODZTWA>) jeśli
nie podano SYM >
===== KONIEC WERSJI V4 =====
=====
},
"language" : < kod języka >,
"status" : {
  "code" : < status odpowiedzi >,
  "message" : < komunikat > :
}
}

```

Opis kodów

procedure_name->publication_text->code

- ETA - opis przygotowany przez EPK i w trakcie akceptacji przez podmiot akceptujący
- EZA - opis przygotowany przez EPK i zaakceptowany przez podmiot akceptujący
- IZA - opis przygotowany przez podmiot akceptujący

procedure_general->local_jurisdiction->for_entrepreneur

- ZASP_NIE_DOTYCZY - nie dotyczy
- ZASP_MIE_WYK - ze względu na miejsce wykonywania działalności
- ZASP_SIEDZ - ze względu na siedzibę
- ZASP_DOW_URZ – dowolny urząd
- ZASP_LOK_PRZEDS - ze względu na lokalizację przedsięwzięcia
- ZASP_MIE_WYK_SIEDZ - ze względu na miejsce wykonywania działalności lub siedzibę firmy
- ZASP_SIE_LUB_MIE - ze względu na siedzibę lub miejsce zamieszkania
- ZASP_LOK_SPRZED - ze względu na lokalizację punktu sprzedaży
- ZASP_PORT_ZALAD - ze względu na port załadunku statku
- ZASP_MIE_POL_OB - ze względu na miejsce położenia obiektu
- ZASP_MIE_WYK_RO - ze względu na miejsce wykonywania robót
- ZASP_MIE_POL_UPR - ze względu na miejsce położenia upraw
- ZASP_POL_PLA_NAS - ze względu na położenie plantacji nasiennej
- ZASP_PORT_MAC - ze względu na port macierzysty statku
- ZASP_MIE_PROW_POL - ze względu na miejsce prowadzenia połowu
- ZASP_KORZ_SRO - ze względu na miejsce korzystania ze środowiska
- ZASP_ODP_KOM - ze względu na miejsce powstawania odpadów komunalnych
- ZASP_MIE_ZAM - ze względu na miejsce zamieszkania
- ZASP_MIE_INW_BUD - ze względu na miejsce inwestycji budowlanej
- ZASP_MIE_ROZ - ze względu na miejsce rozbiórki
- ZASP_POL_BUD - ze względu na położenie budynku
- ZASP_LOK_ZJA - ze względu na lokalizację zjazdu
- ZASP_LOK_ZAJ_PDROG - ze względu na lokalizację zajmowanego pasa drogowego
- ZASP_POL_NIE - ze względu na położenie nieruchomości
- ZASP_POL_DRZE - ze względu na położenie drzewa
- ZASP_LOK_INST - ze względu na lokalizację instalacji
- ZASP_MIE_POL_ZAB - ze względu na miejsce położenia zabytku
- ZASP_MIE_WYZBPR_ODP - ze względu na miejsce wytwarzania, zbierania lub przetwarzania odpadów
- ZASP_MIE_IMP_MAS - ze względu na miejsce imprezy masowej
- ZASP_URZ_SKR_PRZE - ze względu na urząd skarbowy właściwy dla przedsiębiorcy (podatnika)
- ZASP_MIE_ODP_CEL - ze względu na miejsce odprawy celnej
- ZASP_MIE_PRO_KSI_RACH - ze względu na miejsce prowadzenia ksiąg rachunkowych

procedure_general->local_jurisdiction->for_citizen

- ZASO_NIE_DOTYCZY - nie dotyczy
- ZASO_MIE_ZAMEL - ze względu na miejsce zameldowania
- ZASO_MIE_ZAMIE - ze względu na miejsce zamieszkania

- ZASO_DOW_URZ – dowolny urząd
- ZASO_LOK_PRZEDS - ze względu na lokalizację przedsięwzięcia
- ZASO_MIE_ZAMEL_ZAMIE - ze względu na miejsce zameldowania lub zamieszkania
- ZASO_MIE_WYK_ZAW - ze względu na miejsce wykonywania zawodu
- ZASO_MIE_POB_CUDZ - ze względu na miejsce pobytu cudzoziemca
- ZASO_MIE_WYK_ROB - ze względu na miejsce wykonywania robót
- ZASO_MIE_POL_UPR - ze względu na miejsce położenia upraw
- ZASO_POL_PLA_NAS - ze względu na położenie plantacji nasiennej
- ZASO_PORT_MAC - ze względu na port macierzysty statku
- ZASO_MIE_PRO_POL - ze względu na miejsce prowadzenia połowu
- ZASO_KORZ_SRO - ze względu na miejsce korzystania ze środowiska
- ZASO_ODP_KOM - ze względu na miejsce powstawania odpadów komunalnych
- ZASO_MIE_INW_BUD - ze względu na miejsce inwestycji budowlanej
- ZASO_MIE_ROZ - ze względu na miejsce rozbiórki
- ZASO_POL_BUD - ze względu na położenie budynku
- ZASO_LOK_ZJA - ze względu na lokalizację zjazdu
- ZASO_LOK_ZAJ_PDROG - ze względu na lokalizację zajmowanego pasa drogowego
- ZASO_POL_NIE - ze względu na położenie nieruchomości
- ZASO_POL_DRZE - ze względu na położenie drzewa
- ZASO_LOK_INST - ze względu na lokalizację instalacji
- ZASO_MIE_POL_ZAB - ze względu na miejsce położenia zabytku
- ZASO_MIE_IMP_MAS - ze względu na miejsce imprezy masowej
- ZASO_MIE_ODP_CEL - ze względu na miejsce odprawy celnej
- ZASO_MIE_POL_OBI - ze względu na miejsce położenia obiektu

Przykład wywołania

Pobranie szczegółów procedury

[http://\[adres_uslugi\]/ApiProcedures/getDetails?alias=10](http://[adres_uslugi]/ApiProcedures/getDetails?alias=10)

Opis kodów

OK_0000 - poprawna odpowiedź

ERR_0004 - Niepoprawny typ pola

ERR_0005 - Niepoprawny kod języka

ERR_0006 - Przekroczona dopuszczalna długość pola

ERR_0010 - Pole wykracza poza dopuszczalny zakres

ERR_0013 - Element nie znajduje się na liście

ERR_0014 - Pole nie jest ID

ERR_0034 - Podana instytucja nie znajduje się w obszarze podanego zasięgu terytorialnego

ERR_5108 - Parametry TERYT i SYM nie mogą wystąpić jednocześnie

ERR_5109 - Podano nieprawidłowy format numeru SYM

ERR_5110 - Podany numer SYM nie istnieje

Znaczenia „executing_institution”

W ramach tej metody często pojawiają się elementy list typu „executing_institution” (e-usługi, dokumenty, itp.). Są to elementy, które mogą wystąpić tylko w przypadku podania zasięgu terytorialnego lub konkretnego identyfikatora instytucji w parametrach wejściowych metody. Dane

zawarte w odpowiedzi nie wynikają z opisu procedury, ale z dokumentów uszczegóławiających tworzonych przez daną instytucję.

5.6. getThematicCategories - Pobieranie katalogu tematycznego

Funkcja zwraca katalog tematyczne dla procedur

Nazwa funkcji

getThematicCategories

Parametry wejściowe

Parametry wysyłane metodą GET					
Nazwa zmiennej	Typ	obowiązkowy	domyślnie	warunki	opis
lang	string	nie	PL	kod języka	język tłumaczenia procedury

Struktura danych wyjściowych

```
{
  "content" : {
    "ThematicCategories" : [ {
      "id" : < identyfikator katalogu tematycznego >,
      "name" : < nazwa katalogu tematycznego >
    } ]
  },
  "language" : < kod języka >,
  "status" : {
    "code" : < status odpowiedzi >,
    "message" : < komunikat > :
  }
}
```

Przykład wywołania

```
http://[adres_usługi]/ApiProcedures/getThematicCategories
http://[adres_usługi]/ApiProcedures/getThematicCategories?lang=en
```

Opis kodów

```
OK_0000 - poprawna odpowiedź
ERR_0005 - Niepoprawny kod języka
```

5.7. getProfessionCategories - Pobieranie katalogu zawodów

Funkcja zwraca katalog zawodów powiązany z procedurami

Nazwa funkcji

getProfessionCategories

Parametry wejściowe

Parametry wysyłane metodą GET

Nazwa zmiennej	Typ	obowiązkowy	domyślnie	warunki	opis
lang	string	nie	PL	kod języka	język tłumaczenia
with_assigned_procedures_only	int	nie	0	na liście (0, 1)	tylko powiązane z opublikowanymi procedurami

Struktura danych wyjściowych

```
{
  "content" : {
    "ProfessionCategories" : [ {
      "id" : < identyfikator katalogu zawodu >,
      "name" : < nazwa katalogu zawodu >,
      "is_regulated" : < czy zawód regulowany >
    } ]
  },
  "language" : < kod języka >,
  "status" : {
    "code" : < status odpowiedzi >,
    "message" : < komunikat >
  }
}
```

Przykład wywołania

```
http://[adres_usługi]/ApiProcedures/getProfessionCategories
http://[adres_usługi]/ApiProcedures/getProfessionCategories?lang=en
http://[adres_usługi]/ApiProcedures/getProfessionCategories?with_assigned_procedures_only=1
```

Opis kodów

```
OK_0000 - poprawna odpowiedź
ERR_0005 - Niepoprawny kod języka
ERR_0013 - Element nie znajduje się na liście
```

5.8. getListForRespAuthority - Pobieranie listy procedur dla Instytucji Centralnych / Akceptujących

Metoda która umożliwi wyświetlanie wszystkich procedur znajdujących się na liście opisów procedur instytucji centralnej lub akceptującej w zakładce „Procedury przypisane” w widoku szczegółów instytucji na portalu.

Nazwa funkcji

getListForRespAuthority

Parametry wejściowe

Parametry wysłane metodą GET					
Nazwa zmiennej	Typ	obowiązkowy	domyślnie	warunki	opis
lang	string	nie	PL	kod języka	język tłumaczenia procedury
institution_id	int	tak			id instytucji centralnej lub akceptującej

page	int	nie	1	min wartość 1, max wartość 999	numer strony wyszukiwania
limit	int	nie	10	min wartość 1, max wartość 99999	liczba wyników na stronie
search	string	nie		max 100 znaków	nazwa procedury

Struktura danych wyjściowych

```
{
  "content" : {
    "procedures" : [ {
      "id" : < id procedury >,
      "alias" < alias procedury >,
      "name" : < nazwa procedury >,
      "language" : < kod języka procedury >,
      "modification_date" : < data modyfikacji >,
      "status" : < status opublikowania procedury na portalu (Ni, Op, Pr) >
    } ],
    "rows_count" : < liczba rekordów ogółem >
  },
  "language" : < kod języka >,
  "status" : {
    "code" : < status odpowiedzi >,
    "message" : < komunikat >
  }
}
```

Opis słownika „status”

- Ni – Opis opublikowano na portalu, ale jest nieaktualny
- Op – Opis opublikowano na portalu
- Pr – W trakcie przygotowania

Przykład wywołania

Pobranie listy procedur dla Ministerstwa Gospodarki, z domyślną liczbą wyników 10

[http://\[adres_usługi\]/ApiProcedures/getListForRespAuthority?institution_id=170](http://[adres_usługi]/ApiProcedures/getListForRespAuthority?institution_id=170)

Opis kodów

OK_0000 - poprawna odpowiedź
 OK_0002 - poprawna odpowiedź, brak wyników
 ERR_0004 - Niepoprawny typ pola
 ERR_0005 - Niepoprawny kod języka
 ERR_0006 - Przekroczona dopuszczalna długość pola
 ERR_0010 - Pole wykracza poza dopuszczalny zakres
 ERR_0014 - Pole nie posiada prawidłowej wartości dla ID
 ERR_0031 - Instytucja o podanym id nie istnieje
 ERR_0035 - Żądana strona wykracza poza listę wyników

6. ApiPnaInstitutions

6.1. getHintsByCity – pobieranie podpowiedzi po nazwie miejscowości

Opis

Metoda pobierająca listę instytucji po nazwie miejscowości

Nazwa funkcji

getHintsByCity

Parametry wejściowe

Parametry wysyłane metodą GET					
nazwa zmiennej	typ	obowiązkowy	domyślna wartość	warunki	opis
category_ids	array	Nie			Lista kategorii instytucji
city_prefix	string	Tak		Minimum 3 znaki	miejscowość danego urzędu
limit	int	Nie	10		Limit wyświetlania elementów na stronie
procedure_id	string	Nie			id procedury w kontekście której wyświetlana jest instytucja. Jeśli podany, wyniki zawężane są jedynie do instytucji realizujących daną procedurę
esp_only	int	Nie	0	1 lub 0	Jeżeli esp_only=1, funkcja zwraca tylko te instytucje, które mają niepusty adres Elektronicznej Skrzynki Podawczej.

Struktura danych wyjściowych

```
{
  "content" : [ {
    "Institution" : {
      "id" : < id instytucji >,
      "name" : < nazwa instytucji >,
      "esp" : < nazwa skrzynki podawczej; tymczasowo nazwa epuap >,
      "epuap_id" : < identyfikator instytucji w epuap >
    }
  } ],
  "language" : < kod języka >,
  "status" : {
    "code" : < status_odpowiedzi >,
    "message" : < komunikat >
  }
}
```

Przykład wywołania

[http://\[adres_usługi\]/ApiPnaInstitutions/getHintsByCity?city_prefix=Pozna](http://[adres_usługi]/ApiPnaInstitutions/getHintsByCity?city_prefix=Pozna)

Opis kodów

OK_0000 poprawna odpowiedź

ERR_0011 Pole wymagane

ERR_0027 Pole ma mniejszą długość niż wymagana liczba znaków

ERR_0004 Pole posiada niepoprawny typ

6.2. getListByLocation – pobranie instytucji na podstawie lokalizacji

Opis

Odpowiada listą instytucji dla lokalizacji podanej przez użytkownika. Niezbędne jest ID procedury. Opcjonalne są: kategoria, oraz miasto lub kod pocztowy (miasto – rozumiane tutaj jako miasto wnioskującego, nie miasto w którym zlokalizowana jest instytucja). Parametr ulica i numer budynku są elementami dodatkowymi - filtracyjnymi. Gdy Procedura jest wykonywana przez wydział, funkcja zwróci dane adresowe wydziału oraz nazwę tego wydziału.

Nazwa funkcji

getListByLocation

Parametry wejściowe

Parametry wysyłane metodą GET					
Nazwa zmiennej	Typ	obowiązkowy	domyślna wartość	warunki	opis
category_ids	array	nie			Kategoria instytucji
city	string	nie			miasto <u>wnioskującego</u> (zameldowania, zamieszkania, prowadzenia działalności ...), dla którego chce znaleźć urząd przynależny danego typu/typów
post_code	string	nie			kod <u>wnioskującego</u> (zameldowania, zamieszkania, prowadzenia działalności ...), dla którego chce znaleźć urząd przynależny danego typu/typów
street	string	nie			ulica wnioskującego (zameldowania, zamieszkania, prowadzenia działalności ...), dla którego chce znaleźć urząd przynależny danego typu/typów
number	string	nie			numer budynku wnioskującego (zameldowania, zamieszkania, prowadzenia działalności ...), dla którego chce znaleźć urząd przynależny danego typu/typów
procedure_id	string	tak			id procedury (obowiązkowe) w kontekście której wyświetlany jest urząd
esp_only	int	nie	0	1 lub 0	Jeżeli esp_only=1, funkcja zwraca tylko te instytucje, które mają niepusty adres Elektronicznej Skrzynki Podawczej.

Struktura danych wyjściowych

```
{
```



```
"content" : [ {  
  "Institution" : {  
    "id" : < id instytucji >,  
    "city" : < nazwa miejscowości, kraj jeśli inny niż Polska >,  
    "name" : < nazwa instytucji >,  
    "department_name" : < nazwa wydziału instytucji lub null >  
    "number" : < numer lokalu >,  
    "post_code" : < kod pocztowy >,  
    "street" : < ulica >,  
    "voivodeship" : < nazwa województwa >,  
    "voivodeship_teryt_id" : < identyfikator teryt >,  
    "esp" : < skrzynka podawcza >,  
    "epuap_id" : < identyfikator epuap >,  
  }  
} ],  
"language" : < kod języka >,  
"status" : {  
  "code" : < status_odpowiedzi >,  
  "message" : < komunikat >  
}  
}
```

Przykład wywołania

```
http://[adres_usługi]/ApiPnaInstitutions/getListByLocation?procedure_id=538  
http://[adres_usługi]/ApiPnaInstitutions/getListByLocation?procedure_id=158&city=Poz
```

Opis kodów

OK_0000 poprawna odpowiedź

ERR_0011 Pole wymagane

ERR_0004 Pole posiada niepoprawny typ

ERR_0019 Wartość pola jest niepoprawnym ciągiem kodu pocztowego

7. ApiPnaAddress

7.1. getListByPostCode – lista miejscowości po kodzie pocztowym

Nazwa funkcji

getListByPostCode

Parametry wejściowe

Parametry wysłane metodą GET					
Nazwa zmiennej	Typ	obowiązkowy	domyślna wartość	warunki	opis
post_code	string	tak		Kod pocztowy w formacie xx-xxx	Kod pocztowy

Struktura danych wyjściowych

```
{
  "content" : {
    "voivodeship" : [ { < województwa >
      "name" : < nazwa >,
      "province" : [ { < powiaty >
        "name" : < nazwa >,
        "community" : [ { < gminy >
          "name" : < nazwa >,
          "city" : [ { < miasta >
            "name" : < nazwa >,
            "postbox" : < czy skrzynka pocztowa >
          } ]
        } ]
      } ]
    } ]
  },
  "language" : < kod języka >,
  "status" : {
    "code" : < status_odpowiedzi >,
    "message" : < komunikat >
  }
}
```

Przykład wywołania

```
http://[adres_usługi]/ApiPnaAddress/getListByPostCode?post_code=62-020
```

Opis kodów

OK_0000 poprawna odpowiedź
 OK_0002 brak rekordów do wyświetlenia

ERR_0011 Pole wymagane
 ERR_0004 Pole posiada niepoprawny typ
 ERR_0019 Wartość pola jest niepoprawnym ciągiem kodu pocztowego

7.2. getStreetHints - pobranie podpowiedzi ulic

Metoda pobiera listę ulic pasujących do parametrów wejściowych

Nazwa funkcji

getStreetHints

Parametry wejściowe

Parametry wysyłane metodą GET					
Nazwa zmiennej	Typ	obowiązkowy	domyślna wartość	warunki	opis
post_code	string	tak		Poprawny kod pocztowy	Kod pocztowy
city	string	nie			miejsowość
street_prefix	int	tak		Min. 3 znaki	ulica

Struktura danych wyjściowych

```
{  
  "content" : [ < lista nazw ulic > ],  
  "language" : < kod języka > ,  
  "status" : {  
    "code" : < status_odpowiedzi > ,  
    "message" : < komunikat >  
  }  
}
```

Przykład wywołania

```
http://[adres_usługi]/ApiPnaAddress/getStreetHints?street_prefix=kosa
```

Opis kodów

OK_0000 poprawna odpowiedź
OK_0002 brak rekordów do wyświetlenia

ERR_0011 Pole jest wymagane
ERR_0019 Wartość pola jest niepoprawnym ciągiem kodu pocztowego
ERR_0004 Pole posiada niepoprawny typ
ERR_0027 Pole ma mniejszą długość niż wymagana liczba znaków

8. ApiPublications

8.1. getDetails – Pobranie publikacji

Metoda zwraca obiekt publikacji na podstawie podanego aliasu, odpowiednie tłumaczenie zgodne z podanym parametrem **Lang** oraz poradnikami powiązаныmi z publikacją w języku zgodnym z parametrem **Lang**. W przypadku braku tłumaczenia domyślnie jeżeli istnieje to zwracana jest wersja w języku angielskim, a gdy nie istnieje to wersja w języku polskim.

Nazwa funkcji

getDetails

Parametry wejściowe

Parametry wysyłane metodą GET					
Nazwa zmiennej	Typ	obowiązkowy	domyślnie	warunki	opis
alias	string	tak		max 255 znaków	alias publikacji, *TEST-xxx1-xxx2*yyy gdzie xxx1 to id publikacji xxx2 to timestamp yyy to alias publikacji pobiera publikację nawet jeżeli nie jest opublikowana
lang	string	nie	PL	poprawny kod lang	kod języka w standardzie ISO 639-1

Struktura danych wyjściowych

```
{
  "content" : {
    "Publication" : {
      "LifeSituations": [ { < powiązane sytuacje życiowe >
        "acronym" : < akronim >,
        "name" : < nazwa poradnika >,
        "language" : < język >
      } ],
      "alias" : < alias >,
      "text" : < treść >,
      "id" : < id >,
      "title" : < tytuł publikacji >,
      "created" : < data utworzenia >,
      "created_by" : < imię i nazwisko autora >,
      "modified" : < data ostatniej modyfikacji >,
      "modified_by" : < imię i nazwisko osoby modyfikującej >,
      "metadata_description" : < dodatkowy opis >,
      "metadata_keywords" : < słowa kluczowe >,
      "publication" : < wprowadzona data opublikowania >,
      "state" : < status publikacji >,
      "noindex" : < publikacja ukryta >
    }
  },
  "language" : < kod języka >,
  "status" : {
    "code" : < status_odpowiedzi >,
    "message" : < komunikat >
  }
}
```

```
}  
}
```

Przykład wywołania

```
http://[adres_usługi]/ApiPublications/getDetails?alias=2103-testowa-publikacja&lang=de
```

Opis kodów

OK_0000 poprawna odpowiedź

ERR_0005 Parametr 'lang' niezgodna ze standardem ISO 639-1

ERR_0006 Niepoprawna długość

ERR_0011 Pole jest wymagane

ERR_0101 Nie odnaleziono publikacji

ERR_0015 Użyto niepoprawnych znaków

8.2. getList – Pobranie listy publikacji

Pobiera listę opublikowanych publikacji, które nie są ukryte. Lista jest tworzona z podziałem na strony, na podstawie podanej frazy.

Nazwa funkcji

getList

Wersja

V2

Parametry wejściowe

Parametry wysyłane metodą GET

Nazwa zmiennej	Typ	obowiązkowy	domyślnie	warunki	opis
page	int	nie	1	max wartość 999	numer strony
limit	int	nie	10	max wartość 9999	liczba publikacji na stronie
lang	string	nie	PL	poprawny kod lang	kod języka w standardzie ISO 639-1
search	string	nie		max 100 znaków	wyszukiwana fraza

Struktura danych wyjściowych

```
{  
  "content" : {  
    "rows_count" : < liczba wszystkich wierszy >,  
    "Publications" : [ {  
      "Publication" : {  
        "alias" : < alias >,  
        "id" : < id >,  
        "title" : < tytuł publikacji >,  
        "lang" : < język publikacji >,  
        "weight" : < waga sphinx >,  
        "short_introduction" : < zawartość pola „Krótkie wprowadzenie” >  
      }  
    } ]  
  },  
}
```

```
"language" : < kod języka >,  
"status" : {  
  "code" : < status_odpowiedzi >,  
  "message" : < komunikat >  
}  
}
```

Przykład wywołania

```
http://[adres_usługi]/ApiPublications/getList
```

Opis kodów

OK_0000 poprawna odpowiedź

OK_0002 brak rekordów do wyświetlenia

ERR_0005 Parametr 'lang' niezgodny ze standardem ISO 639-1

ERR_0006 Przekroczona dopuszczalna długość pola

9. ApiMail

9.1. sendMail – wysyłka wiadomości e-mail

Metoda umożliwia wysyłkę wiadomości email.

Nazwa funkcji

sendMail

Parametry wejściowe

Parametry wysyłane metodą GET					
Nazwa zmiennej	Typ	obowiązkowy	domyślnie	warunki	opis
email	string	tak		max 250 znaków	Adres email odbiorcy
reply_to	String	nie		max 250 znaków	Adres email nadawcy
template_code	string	tak		poprawny kod szablonu, max 250 znaków	kod szablonu dostępny z poziomu zaplecza
parameters	tablica	nie		zgodność przesłanych elementów z szablonem	Parametry zdefiniowane z szablonie są zastępowane przesłanymi wartościami
attachments	Tablica	Nie		pojedynczy załącznik nie większy niż 50 MB, obsługiwane typy plików to rtf, txt, pdf, xls, xlsx, doc, docx, png, jpg, xml	Zakodowane w Base64 pliki

Struktura parametru parameters

```
array(
  [nazwa parametru 1] = [wartość parametru 1],
  [nazwa parametru 2] = [wartość parametru 2],
  [nazwa parametru 3] = [wartość parametru 3],
)
```

Struktura parametru attachments

```
array( (int) 0 =
  array(
    "data" = [treść załącznika w Base64],
    "name" = [nazwa załącznika],
    "mime" = [typ MIME pliku]
  )
)
```

Struktura danych wyjściowych

```
{
  "content" : {
```



```
"Mail" : {  
  "message" : < treść >,  
}  
}  
"language" : < kod języka >,  
"status" : {  
  "code" : < status_odpowiedzi >,  
  "message" : < komunikat >  
}  
}
```

Przykład wywołania

[http://\[adres_uslugi\]/ApiMail/sendMail](http://[adres_uslugi]/ApiMail/sendMail)

W POST

Nagłówek: Content-Type: application/x-www-form-urlencoded

Body:

1. [email=test%40test.pl&template_code=wysylka](#)
2. email= test%40test.pl&template_code=wysylka
¶meters%5Btest%5D=test¶meters%5Btest2%5D=test2

Opis kodów

OK_0000 poprawna odpowiedź

ERR_0001 Niepoprawna metoda protokołu. Wymagany POST.
ERR_0003 Pole ma niepoprawny adres e-mail.
ERR_0004 Pole posiada niepoprawny typ.
ERR_0006 Niepoprawna długość.
ERR_0011 Pole jest wymagane.
ERR_0015 Użyto niepoprawnych znaków.
ERR_0041 Pole ma niepoprawny format. Pole może zawierać tylko litery, cyfry, znak _.
ERR_0042 Zbyt duży rozmiar załącznika.
ERR_0043 Niepoprawny typ załącznika.
ERR_0044 Brak szablonu wiadomości o podanym kodzie.
ERR_0045 Brak wymaganych parametrów dla szablonu wiadomości.
ERR_0046 Podano nadmiarowe parametry dla szablonu.
ERR_0047 Załącznik jest niekompletny.

9.2. getMessages – pobieranie wiadomości

Funkcja zwracająca wiadomości e-mail. Wykonanie tej funkcji jest możliwe tylko z poziomu środowiska QA.

Nazwa funkcji

getMessages

Parametry wejściowe

Parametry wysyłane metodą GET					
Nazwa zmiennej	Typ	obowiązkowy	domyślnie	warunki	opis
from	string	tak		poprawna data z czasem	Data z czasem w formacie RRRR-MM-DD GG:MM:SS od której mają zostać zwrócone wiadomości, podanie czasu jest opcjonalne
subject	String	nie			Fragment tematu
message	string	nie			Fragment wiadomości
to_email	string	nie		poprawny adres e-mail	E-mail adresata
to_name	string	nie			Nazwisko adresata

Struktura danych wyjściowych

```
{
  "content" : {
    "rows_count" : < liczba wszystkich wierszy >,
    "Messages" : [ {
      "Message" : {
        "subject" : < treść >,
        "message" : < treść >,
        "message_html" : < treść w formacie HTML >,
        "to_name" : < adresat >,
        "to_email" : < e-mail adresata >,
        "creation_date" : < data utworzenia >,
        "processing_date" : < data przetworzenia >,
        "is_processed" : < status przetworzenia >,
        "reply_to" : < odpowiedź do >,
        "processing_error_code" : < kod błędu przetwarzania >,
        "processing_error_description" : < opis błędu przetwarzania >
      }
    } ]
  },
  "language" : < kod języka >,
  "status" : {
    "code" : < status_odpowiedzi >,
    "message" : < komunikat >
  }
}
```

Przykład wywołania

[http://\[adres usługi\]/ApiMail/getMessages?from=2016-07-05 12:10](http://[adres usługi]/ApiMail/getMessages?from=2016-07-05 12:10)

Opis kodów

OK_0000 Poprawna odpowiedź
OK_0002 Poprawna odpowiedź, bez wyników
ERR_0003 Pole ma niepoprawny adres e-mail.
ERR_0011 Pole jest wymagane.
ERR_0015 Użyto niepoprawnych znaków.
ERR_0049 Format podanej daty z czasem jest nieprawidłowy
ERR_5022 Nieprawidłowe środowisko

10. ApiUsers

10.1. getDeletedUsers – pobranie usuniętych użytkowników

Funkcja zwraca identyfikatory użytkowników którzy zostali usunięci z systemu

Nazwa funkcji

getDeletedUsers

Parametry wejściowe

Parametry wysyłane metodą GET					
Nazwa zmiennej	Typ	obowiązkowy	domyślnie	warunki	opis
date_from	string	nie	1 tydzień od data_do w postaci RRRR-MM-DD		Data od
date_to	String	nie	Data aktualna w postaci RRR-MM-DD		Data do

Struktura danych wyjściowych

```
{
  "content" : {
    "users" : [ { <użytkownicy>
      "id": <identyfikator użytkownika>
    } ]
  }
  "language" : < kod języka >,
  "status" : {
    "code" : < status_odpowiedzi >,
    "message" : < komunikat >
  }
}
```

Przykład wywołania

```
http://[adres_usługi]/ApiUsers/getDeletedUsers
http://[adres_usługi]/ApiUsers/getDeletedUsers?date_from=2012-10-10
http://[adres_usługi]/ApiUsers/getDeletedUsers?date_from=2012-10-10&date_to=2015-10-10
```

Opis kodów

OK_0000 poprawna odpowiedź

ERR_0026 Wartość pola [nazwa pola] nie jest datą w postaci RRRR-MM-DD.

ERR_0048 Data [nazwa pola] nie może być większa od daty [nazwa pola].