**Описание серсвисов REST ISP WebAPI**

Система серсвисов REST ISP WebAPI предоставляет возможность внешним по отношению к ISP приложениям выполнять различные запросы в базе данных ISP, а также различные функции с файлами, директориями, средствами коммуникации.

Все сервисы разделены по группам, в зависимости от объекта, на который они направлены. Например, группа сервисов webusers объединяет в себе сервисы, предназначенные для работы со списком web user. Среди этих сервисов есть сервис для ввод нового user, сервис для поиска всех users, сервис для поиска данных конкретного user и т.д.

**Группы сервисов**

1. Сервисные функции (4)
2. Сервисы для работы со списком пользователей мобил. приложения WebUsers (7)
3. Сервисы для работы со списком устройств пользователей мобильного приложения WebUsersDevices (9)
4. Сервисы для работы со списком клиентов компании Clients (7)
5. Сервисы для работы со списком стран Countries (1)
6. Сервисы для работы со списком валют Currencies (1)
7. Сервисы для работы со списком банков Banks (1)
8. Сервисы для работы со списком специальностей Specials (1)
9. Сервисы для работы со списком отраслей деятельности клиентов Brunches (1)
10. Сервисы для работы со списком документов клиентов ClientsDocFiles (2)
11. Сервисы для работы со списком инвестиционных счетов клиентов Contracts(1)
12. Сервисы для работы со списком банк. счетов клиентов ClientsBankAccounts (3)
13. Сервисы для работы со списком типов запросов клиентов ClientsRequestsTypes
14. Сервисы для работы со списком запросов клиентов ClientsRequests (4)
15. Сервисы для работы со списком параметров Options (1)
16. **Сервисные функции**

Группа сервисов serverjobs объединяет в себе сервисы, выполняющие разнообразные сервисные функции.

* 1. **Сервис для отправки e-mail сообщения**  
       
     http://10.0.4.7:8091/serverjobs/SendEMail/{'email': 'abc@def.com', 'body': 'Hello, ….’}  
       
     здесь вместо ‘[abc@def.com](mailto:abc@def.com)’ нужно указать e-mail адрес, на который должно быть послано сообщение,   
     а вместо 'Hello, ….’– указать текст самого сообщения

После отправки сообщения сервис возвращает значение 1.

Например:  
[http://10.0.4.7:8091/serverjobs/SendEMail/{'email':'aaa@bbb.gr','body](http://10.0.4.7:8091/serverjobs/SendEMail/%7b'email':'aaa@bbb.gr','body)': 'Your e-mail Verification Code is 123456'}

**1.2 Сервис для отправки стандартного e-mail сообщения**  
  
Для отправки сервером системы некоторого e-mail сообщения с определенным стандартным текстом предназначен сервис  
  
http://10.0.4.7:8091/serverjobs/SendEMail46/{'recipient\_email': 'abc@def.com', 'request\_action': '6', 'request\_id': '123', 'att': 'document.pdf', 'person\_name': ' Κ\*\*\*\*\*Σ Π\*\*\*\*\*\*\*\*\*\*Σ ', 'afm': '123456789', ‘delay’: ‘30’}  
  
где

recipient\_email – это e-mail адрес, на который должно быть послано сообщение, например, здесь email abc@def.com. Данный атрибут обязателен  
request\_action – номер одного из стандарных сообщений. Их список приведен ниже. Данный атрибут обязателен

request\_id- ID заявки, полученный после отправки заявки на сервер. Данный атрибут обязателен только при request\_action=1 или 2 или 3  
att – название файла, вложеннного в данное сообщение. Этот атрибут не обязателен  
person\_name – фамилия-имя человека. Может быть закодирована звездочками. Этот атрибут не обязателен  
afm - ΑΦΜ человека. Этот атрибут не обязателен  
delay – длительность паузы в сек. перед отправкой e-mail сообщения сразу после загрузки файлов  
Список стандартных сообщений:  
1. Ενημέρωση αιτήματος - сообщение об успешной отправке заявки клиента. Формат сообщения таков

{'recipient\_email': 'it@hellasfin.gr', 'request\_action': '1', 'request\_id': '3', 'person\_name': 'Βασιλειοσ Βενετικιδησ'}

где

recipient\_email – это e-mail адрес, на который должно быть послано сообщение, request\_action = 1

request\_id- ID заявки, полученный после отправки заявки на сервер

2. Ολοκλήρωση αιτήματος – сообщение об успешном исполнении заявки клиента  
3. Απόρριψη αιτήματος - сообщение об аннуляции заявки клиента  
4. Δημιουργία συνθηματικού – сообщение об успешном изменении password’а  
5. Επιτυχημένη εγγραφή - Αποδοχή όρων - сообщение клиенту об успешной регистрации  
6. Αποστολή πρόσκλησης στον master – сообщение master’у о приглашении к связи клиента с указаной фамилией-именем и АФМ   
7. Παραλαβή πρόσκλησης στον πελάτη– сообщение клиенту о приглашении к связи от лица с указаной фамилией-именем и АФМ  
8. Διαγραφή συνδέσεις στον πελάτη – сообщение клиенту об удалении связи с ним  
9. Διαγραφή συνδέσεις στον master– сообщение master’у об удалении связи с указанным клиентом  
10. Απόρριψη πρόσκλησης στον πελάτη - сообщение клиенту об аннуляции им приглашения к связи  
11. Απόρριψη πρόσκλησης στον master- сообщение master’у об аннуляции клиентом приглашения к связи  
12. Δημιουργία συνδέσεις στον πελάτη - сообщение клиенту об установлении связи с ним  
13. Δημιουργία συνδέσεις στον master - сообщение master’у об установлении связи с клиентом  
14. Λήξη πρόσκλησης στον master - сообщение master’у об истечении срока ожидания в 30 дней  
15. Λήξη πρόσκλησης στον πελάτη - сообщение клиенту об истечении срока ожидания в 30 дней  
16. Ακύρωση πρόσκλησης στον master - сообщение master’у об аннуляции связи с клиентом  
17. Ακύρωση πρόσκλησης στον πελάτη - сообщение клиенту об аннуляции связи с указанным лицом   
18. Ακύρωση αιτήματος στον χρήστη - сообщение юзеру, который создал заявку, о том, что он сам аннулировал эту заявку при том, что эта заявка отправлена, но пока не исполнена Алкисом  
19. Ακύρωση αιτήματος в HellasFin - сообщение о том, что юзер аннулировал заявку, которую он уже отправил, но она пока не исполнена

20. Έχετε ένα νέο αίτημα в HellasFin – сообщение о поступлении новой заявки клиента. Формат посылаемого сообщения таков

{'recipient\_email': 'it@hellasfin.gr', 'request\_action': '20', 'person\_name': 'Βασιλης Βενετικιδης' }

21. Εγγραφή νέου χρήστη στην εφαρμογή - сообщение в HellasFin об успешной регистрации клиента. Формат посылаемого сообщения таков  
 {'request\_action': '21', 'person\_name': 'Βασιλης Βενετικιδης', 'afm': '123456789' }

22. Αποτυχία ταυτοποίησης – сообщение «новому» клиенту о его удалении из списка клиентов. Формат посылаемого сообщения таков   
 {'request\_action': '22'}  
23. Υπενθύμιση ταυτοποίησης μέσω εγγράφων – напоминание клиенту о скором истечении срока проведения видеочата. Формат посылаемого сообщения таков  
 {'request\_action': '23'}  
24. Έληξε ο χρόνος αναμονής ταυτοποίησης – сообщение «старому» клиенту об истечении срока проведения видеочата и его блокировании в списке клиентов. Формат посылаемого сообщения таков  
 {'request\_action': '24'}  
25. Απορρίφθηκε η ταυτοποίηση μέσω βίντεο κλήσης– сообщение клиенту об аннуляции ранее проведенного видеочата и необходимости его повторения. Формат посылаемого сообщения таков  
 {'request\_action': '25', 'notes': 'text-cancellation-reason'}  
  
26. Απορρίφθηκε η ταυτοποίηση μέσω βίντεο κλήσης– сообщение в HellasFin об аннуляции ранее проведенного видеочата и необходимости его повторения. Формат посылаемого сообщения таков  
{'request\_action': '26', 'person\_name': 'DELLAS NIKOS', ‘afm’: ‘123456789’, 'notes': 'text-cancellation-reason' }  
  
27. One Time Password – сообщение клиенту об отправке ему одноразового кода  
{'request\_action': '27', 'otp': '123456' }

После отправки сообщения сервис возвращает значение 1.

* 1. **Сервис для отправки SMS сообщения**  
       
     http://10.0.4.7:8091/serverjobs/SendSMS/{'mobile': 'AAAAAA', 'c': 'BBBBB’}  
       
     здесь вместо AAAAAA нужно указать номер мобильного телефона, на который **должно быть послано сообщение,**а вместо BBBBB – указать текст самого сообщения

Например:  
http://10.0.4.7:8091/serverjobs/SendSMS/{'mobile': XXXXXXXXXX, 'message': 'Your mobile Verification Code is 123456'}

* 1. **Сервис для загрузки файла на сервер**  
       
     http://10.0.4.7:8091/serverjobs/UploadFile/{'file\_name': 'alfa.pdf', 'client\_id': '356’, 'doc\_type': '27', ‘status’ : ‘1’ }  
       
     этот сервис загружает на сервер файл alfa.pdf, который является документом типа 27 со статусом равным 1 в папку клиента с ID = 356

1. **Сервисы для работы со списком пользователей мобильного приложения WebUsers** 
   1. **Сервис для регистрации нового пользователя и его мобильного устройства в базе данных системы**

Вызов сервиса осуществляется методом POST с помощью команды

<http://10.0.4.7:8091/webusers/create>  
При вызове сервиса на сервер передаются данные о новом пользователе мобильного устройства в JSON-формате со следующей структурой:

int Client\_ID  
 string Surname  
 string Firstname  
 string AFM  
 string email   
 string mobile\_phone   
 string password   
 int WU\_ID  
 string Manufacturer   
 string Brand   
 string Model   
 string Board  
 string Hardware   
 string Unique\_ID  
 string ScreenResolution   
 string ScreenDensity   
 string Host   
 string Version   
 string API\_level   
 string Build\_ID   
 string Build\_Time   
 string Fingerprint   
 string PhoneType   
 string NetworkCountryISO   
 string NetworkOperatorName   
 string DeviceId   
 string DeviceSoftwareVersion   
 string SimCountryIso   
 string SimOperatorName   
 string SimSerialNumber   
 string Imei   
 string Meid   
 string MmsUAProfUrl   
 string MmsUserAgent   
 string SubscriberId   
 string TypeAllocationCode  
 string OS  
 string Video  
 int Status  
 datetime DateIns  
  
После отработки запроса сервис возвращает полный набор данных о web пользователя со структурой, показанной выше с новыми значениями Client\_ID и WU\_ID.

* 1. **Сервис для регистрации нового пользователя**Вызов сервиса осуществляется методом POST с помощью команды

<http://10.0.4.7:8091/webusers/addwebuser>  
При вызове сервиса на сервер передаются данные о новом пользователе мобильного устройства в JSON-формате со следующей структурой:

int Client\_ID  
 string Surname  
 string Firstname  
 string AFM  
 string email   
 string mobile\_phone   
 string password

* 1. **Сервис определения ID web пользователя по любому сочетанию параметеров email, mobile, afm, dob, client\_id.**

Вызов сервиса осуществляется методом GET с помощью команды

<http://10.0.4.7:8091/webusers/>getwebuser\_id/{'email': 'gundatsev@gmail.ru', 'mobile': 'XXXXXXXXXX', 'afm': 'XXXXXXXXX', 'dob': ‘YYYY-MM-DD TIME’, 'client\_id': '1234' }

Параметры email, mobile, afm, dob, client\_id могут указываться в любом порядке и в любом сочетании. Обязательно указание хотя бы одного параметра.   
Любой параметр может быть опущен  
  
После отработки запроса сервис возвращает ID найденного web пользователя.  
Если web пользователь с указанными параметрами не найден возвращается 0.  
Если найдено несколько web пользователей с указанными параметрами, то возвращается ID последнего пользователя, т.е. максимальное значение из найденных ID   
Использование только одного параметра dob технически возможно, но правильно его использовать в сочетании с другим параметром, например, afm.

Например,  
http://10.0.4.7:8091/webusers/GetWebUser\_ID/{'email': 'gundartsev@gmail.com'} – возврат ID web пользователя по email

http://10.0.4.7:8091/webusers/getwebuser\_id/{'mobile': 'XXXXXXXXXX', 'afm': 'XXXXXXXXX'}– возврат ID web пользователя по mobile и afm

* 1. **Сервис определения полного набора данных web пользователя по любому сочетанию параметеров email, mobile, afm, dob, client\_id.**

Вызов сервиса осуществляется методом GET с помощью команды

<http://10.0.4.7:8091/webusers/>getwebuser\_data/{'email': 'gundatsev@gmail.ru', 'mobile': 'XXXXXXXXXX', 'afm': 'XXXXXXXXX', 'dob': ‘YYYY-MM-DD TIME’, 'client\_id': '1234' }

Параметры email, mobile, afm, dob, client\_id могут указываться в любом порядке и в любом сочетании. Обязательно указание хотя бы одного параметра.   
Любой параметр может быть опущен  
  
После отработки запроса сервис возвращает полный набор данных о web пользователя со структурой, показанной ниже.  
Если web пользователь с указанными параметрами не найден возвращается пустой набор данных.  
Если найдено несколько web пользователей с указанными параметрами, то возвращается набор данных последнего пользователя, т.е. набор с максимальным значением ID

Структура возвращаемого набора данных  
id: 1  
\_Email: [aaa@bbb.gr](mailto:vitaly@otenet.gr)  
\_Mobile: "XXXXXXXXXX"  
\_Password: "2"  
\_AFM0: "123456789"  
surname: "XXXXXXXXXXXXXX"  
firstname: "XXXXXXXX"  
surnameFather: ""  
firstnameFather: ""  
surnameMother: ""  
firstnameMother: ""

doB: "1980-01-01T00:00:00"  
bornPlace: ""  
adt: "AE 176152"  
expireDate: ""  
police: ""  
doy: ""  
afm: "XXXXXXXXX"  
address: ""  
city: ""  
zip: ""  
country\_ID: 0  
tel: ""  
mobile: "XXXXXXXXXX"  
eMail: [aaa@bbb.gr](mailto:vitaly@otenet.gr)  
client\_ID: 11884  
currentStep: 4  
termsAgreement: "TermsAgreement\_202011275616.pdf"  
status: 1  
dateIns: "0001-01-01T00:00:00"

Например,   
http://10.0.4.7:8091/webusers/GetWebUser\_Data/{'mobile': 'XXXXXXXXXX'} - возврат полного набора данных о web пользователя пользователя по mobile

* 1. **Сервис изменения пароля webuser**

Вызов сервиса осуществляется методом POST с помощью команды

<http://10.0.4.7:8091/webusers/updatepassword>При вызове сервиса на сервер передаются   
 int wu\_id – ID webuser  
 string password - новый password

* 1. **Сервис изменения статуса webuser**

Вызов сервиса осуществляется методом POST с помощью команды

<http://10.0.4.7:8091/webusers/updatestatus>При вызове сервиса на сервер передаются   
 int id – ID webuser  
 int status - новый статус

По завершении работы сервис возвращает 1 – если изменение произошло, 0 – если не произошло, например, по причине отсутствия записи WebUsers с указанным ID

* 1. **Сервис связывания webuser с клиентом.**

Сервис устанавливает связь между webuser с WU\_ID и клиентом с Client\_ID.  
 Вызов сервиса осуществляется методом GET с помощью команды

<http://10.0.4.7:8091/>webusers/MatchWUC/{'WU\_ID': '1', 'Client\_ID': '234'}

По завершении работы сервис возвращает WU\_ID

* 1. **Сервис cохранения состояния регистрации**

Вызов сервиса осуществляется методом POST с помощью команды

<http://10.0.4.7:8091/webusersstates/savedata>При вызове сервиса на сервер передаются   
 int wu\_id – ID webuser  
 status - статус записи

Возвращаемое значение: 1 - success, 0 - fail

* 1. **Сервис определения статуса состояния регистрации**

Вызов сервиса осуществляется методом GET с помощью команды

<http://10.0.4.7:8091/webusersstates/getstatus/123> где 123 – ID webuser

После отработки запроса сервис возвращает значение status

**2.10 Сервис удаления webuser**

Сервис ВРЕМЕННЫЙ – доступен на период разработки

Сервис удаляет webuser по его ID.  
 Вызов сервиса осуществляется методом GET с помощью команды

<http://10.0.4.7:8091/webusers/delete?id=>123

По завершении работы сервис возвращает ID удаленной записи

1. **Сервисы для работы со списком устройств пользователей мобильного приложения WebUsersDevices**
   1. **Сервис для регистрации нового мобильного устройства в базе данных системы**  
      Вызов сервиса осуществляется методом POST с помощью команды <http://10.0.4.7:8091/webusersdevices/create>  
        
      При вызове сервиса на сервер передаются данные о новом мобильном устройстве в JSON-формате со следующей структурой:

WU\_ID int,  
 Manufacturer nvarchar(100),  
 Brand nvarchar(100),  
 Model nvarchar(100),  
 Board nvarchar(100),  
 Hardware nvarchar(100),  
 Unique\_ID nvarchar(100),  
 ScreenResolution nvarchar(100),  
 ScreenDensity nvarchar(100),  
 Host nvarchar(100),  
 Version nvarchar(100),  
 API\_level nvarchar(100),  
 Build\_ID nvarchar(100),  
 Build\_Time nvarchar(100),  
 Fingerprint nvarchar(100),  
 PhoneType nvarchar(100),  
 NetworkCountryISO nvarchar(100),  
 NetworkOperatorName nvarchar(100),  
 DeviceId nvarchar(100),  
 DeviceSoftwareVersion nvarchar(100),  
 SimCountryIso nvarchar(100),  
 SimOperatorName nvarchar(100),  
 SimSerialNumber nvarchar(100),  
 Imei nvarchar(100),  
 Meid nvarchar(100),  
 MmsUAProfUrl nvarchar(100),  
 MmsUserAgent nvarchar(100),  
 SubscriberId nvarchar(100),  
 TypeAllocationCode nvarchar(100),  
 OS nvarchar(100),  
 Video nvarchar(100),  
 Status smallint  
  
По завершении работы сервис возвращает ID нового мобильного устройства

* 1. **Сервис определения ID устройства клиента по любому сочетанию параметеров этого клиента ID, email, mobile, afm, dob.**

Вызов сервиса осуществляется методом GET с помощью команды

[http://10.0.4.7:8091/webusersdevices/getwebusersdevices\_id/{'id](http://10.0.4.7:8091/webclientsdevices/getwebusersdevices_id/%7b'id)': '123', 'email': 'gundatsev@gmail.ru', 'mobile': 'XXXXXXXXXX', 'afm': 'XXXXXXXXX', 'dob': ‘YYYY-MM-DD TIME’ }

Параметры ID, email, mobile, afm, dob могут указываться в любом порядке и в любом сочетании. Обязательно указание хотя бы одного параметра.   
Любой параметр может быть опущен  
  
После отработки запроса сервис возвращает ID найденного устройства.  
Если устройство с указанными параметрами клиента не найдено, то возвращается 0.  
Если найдено несколько устройств с указанными параметрами клиента, то возвращается ID последнего устройства, т.е. максимальное значение из найденных ID   
Использование только одного параметра dob технически возможно, но правильно его использовать в сочетании с другим параметром, например, afm.

Например,  
http://10.0.4.7:8091/webusersdevices/GetWebUsersDevices\_ID /{'id': 1234} – возврат ID устройства по ID клиента, которому это устройство принадлежит

http://10.0.4.7:8091/ webusersdevices/GetWebUsersDevices\_ID /{'mobile': 'XXXXXXXXXX', 'afm': 'XXXXXXXXX'}– возврат ID устройства по mobile и afm клиента, которому это устройство принадлежит

* 1. **Сервис определения полного набора данных устройств клиента по любому сочетанию параметеров этого клиента ID, email, mobile, afm, dob.**

Вызов сервиса осуществляется методом GET с помощью команды

[http://10.0.4.7:8091/webusersdevices/getwebusersdevices\_data/{'id](http://10.0.4.7:8091/webclientsdevices/getwebusersdevices_data/%7b'id)': '123', 'email': 'gundatsev@gmail.ru', 'mobile': 'XXXXXXXXXX', 'afm': 'XXXXXXXXX', 'dob': ‘YYYY-MM-DD TIME’ }

Параметры ID, email, mobile, afm, dob могут указываться в любом порядке и в любом сочетании. Обязательно указание хотя бы одного параметра.   
Любой параметр может быть опущен  
  
После отработки запроса сервис возвращает полный набор данных найденных устройств.  
Если устройство с указанными параметрами клиента не найдено, то возвращается 0.  
Если найдено несколько устройств с указанными параметрами клиента, то возвращается список устройства со всеми их характеристиками.   
Использование только одного параметра dob технически возможно, но правильно его использовать в сочетании с другим параметром, например, afm.

Например,  
http://10.0.4.7:8091/webusersdevices/GetWebUsersDevices\_Data /{'id': 1234} – возврат характеристик устройства по ID клиента, которому это устройство принадлежит

http://10.0.4.7:8091/ webusersdevices/GetWebUsersDevices\_Data /{'mobile': 'XXXXXXXXXX', 'afm': 'XXXXXXXXX'}– возврат характеристик устройств по mobile и afm клиента, которому это устройство принадлежит  
  
Пример

A picture containing text

Description automatically generated

* 1. **Сервис определения списка ID устройств клиента по любому сочетанию параметеров этого клиента ID, email, mobile, afm, dob.**

Вызов сервиса осуществляется методом GET с помощью команды

[http://10.0.4.7:8091/webusersdevices/getwebusersdeviceslist\_id/{'id](http://10.0.4.7:8091/webclientsdevices/getwebusersdeviceslist_id/%7b'id)': '123', 'email': 'gundatsev@gmail.ru', 'mobile': 'XXXXXXXXXX', 'afm': 'XXXXXXXXX', 'dob': ‘YYYY-MM-DD TIME’ }

Параметры ID, email, mobile, afm, dob могут указываться в любом порядке и в любом сочетании. Обязательно указание хотя бы одного параметра.   
Любой параметр может быть опущен  
  
После отработки запроса сервис возвращает список ID найденных устройств.  
Если устройств с указанными параметрами клиента не найдено, то возвращается пустой набор данных.  
Использование только одного параметра dob технически возможно, но правильно его использовать в сочетании с другим параметром, например, afm.

Например,  
http://10.0.4.7:8091/webusersdevices/GetWebUsersDevicesList\_ID /{'id': 1234} – возврат списка ID устройства по ID клиента, которому это устройство принадлежит

http://10.0.4.7:8091/ webusersdevices/GetWebUsersDevicesList\_ID /{'mobile': 'XXXXXXXXXX', 'afm': 'XXXXXXXXX'}– возврат списка ID устройства по mobile и afm клиента, которому это устройство принадлежит

* 1. **Сервис определения списка полных наборов данных устройств клиента по любому сочетанию параметеров этого клиента ID, email, mobile, afm, dob.**

Вызов сервиса осуществляется методом GET с помощью команды

[http://10.0.4.7:8091/webusersdevices/getwebusersdeviceslist\_data/{'id](http://10.0.4.7:8091/webclientsdevices/getwebusersdeviceslist_data/%7b'id)': '123', 'email': 'gundatsev@gmail.ru', 'mobile': 'XXXXXXXXXX', 'afm': 'XXXXXXXXX', 'dob': ‘YYYY-MM-DD TIME’ }

Параметры ID, email, mobile, afm, dob могут указываться в любом порядке и в любом сочетании. Обязательно указание хотя бы одного параметра.   
Любой параметр может быть опущен  
  
После отработки запроса сервис возвращает список полных наборов данных найденных устройств.  
Если устройств с указанными параметрами клиента не найдено, то возвращается пустой набор.  
Использование только одного параметра dob технически возможно, но правильно его использовать в сочетании с другим параметром, например, afm.

Например,  
http://10.0.4.7:8091/webusersdevices/GetWebUsersDevicesList\_Data /{'id': 1234} – возврат списка характеристик устройств по ID клиента, которому это устройство принадлежит

http://10.0.4.7:8091/ webusersdevices/GetWebUsersDevicesList\_Data /{'mobile': 'XXXXXXXXXX', 'afm': 'XXXXXXXXX'}– возврат списка характеристик устройств по mobile и afm клиента, которому это устройство принадлежит

* 1. **Сервис определения списка полных наборов данных устройств по их unique\_ID**

Вызов сервиса осуществляется методом GET с помощью команды

http://10.0.4.7:8091/webusersdevices/GetList\_Unique\_ID/{'unique\_ID': '4d691c8f74c87626af'}

После отработки запроса сервис возвращает список полных наборов данных найденных устройств.  
Если устройств с указанными параметрами клиента не найдено, то возвращается пустой набор.  
 **Graphical user interface, text, application, email

Description automatically generated**

* 1. **Сервис определения статуса устройства по его unique\_ID**

Вызов сервиса осуществляется методом GET с помощью команды

[http://10.0.4.7:8091/webusersdevices/GetDeviceStatus/{'unique\_ID': ‘9182nb0k2751qa8hl45'}](http://10.0.4.7:8091/webusersdevices/GetDeviceStatus/%7b'unique_ID':%20‘9182nb0k2751qa8hl45'%7d)

где unique\_ID – значение параметра unique\_ID.   
  
После отработки запроса сервис возвращает статус найденного устройства:   
1 – устройство активно,   
2 – устройство заблокировано,   
0 – определить статус устройства не удалось. Например, устройство с указанным unique\_ID не найдено или др.  
  
Поскольку параметр unique\_ID может быть не уникальным, то сервис определяет статус первого найденного устройства с указанным unique\_ID. Если ранее не было допущено ошибок все устройства с одним и тем же unique\_ID должны иметь один и тот же статус

* 1. **Сервис изменения любых данных устройств по их ID**

Вызов сервиса осуществляется методом POST с помощью команды

<http://10.0.4.7:8091/webusersdevices/update>

При вызове сервиса на сервер передается id устройства, т.е WebUsersDevices.ID, а также данные, которые дожны быть изменены, в JSON-формате со следующей структурой:  
 {‘id’:123 , ‘wu\_id’: 274, ‘Manufacturer’: ‘Apple Inc’, ‘Brand’ : ‘Apple Inc ‘, ….}

где .... значения любых данных об этом устройстве. Все возможные данные перечисленны в пункте 3.1

* 1. **Сервис изменения значения WebUser ID (WU\_ID).**

Вызов сервиса осуществляется методом POST с помощью команды

<http://10.0.4.7:8091/webusersdevices/updatewu_id>При вызове сервиса на сервер передаются   
 int id   
 int wu\_id,  
где id – ID устройства, статус которого должен быть изменен  
 wu\_id – новое значение WebUser ID

* 1. **Сервис изменения статуса мобильного устройства**

Данный сервис позволяет изменять статус мобильного устройства по значению его ID и по или по значению поля Unique\_ID. Указываться должно значение только одного из этих параметров. Если по ошибке при вызове сервера будут указаны значения обоих параметров, то значение поля Unique\_ID будет проигнорировано.  
Поскольку ID устройства уникально, то при указании значения ID исполнение данного сервиса может привести к изменению статуса только в одной записи таблицы WebUsersDevices.  
Поскольку значение поля Unique\_ID устройства не уникально, то при указании его значения исполнение данного сервиса может привести к изменению статуса во всех записях таблицы WebUsersDevices с таким Unique\_ID.

Вызов сервиса осуществляется методом POST с помощью команды

[http://10.0.4.7:8091/webusersdevices/updatestatus](http://10.0.4.7:8091/webclientsdevices/updatestatus)При вызове сервиса на сервер передаются   
 string id,

string unique\_id,  
 int status,  
где id – поле ID устройства

unique\_id– поле Unique\_ID,

status – 1 – устройство активно, 2 – устройство заблокировано

* 1. **Сервис удаления устройства.**

Вызов сервиса осуществляется методом POST с помощью команды

[http://10.0.4.7:8091/webusersdevices/delete](http://10.0.4.7:8091/webclientsdevices/delete)При вызове сервиса на сервер передаются   
 int id – ID записи, которая должна быть удалена

1. **Сервисы для работы со списком клиентов компании Clients**
   1. **Сервис определения ID клиента по любому сочетанию параметеров email, mobile, afm, dob.**

Вызов сервиса осуществляется методом GET с помощью команды

<http://10.0.4.7:8091/clients/>getclient\_id/{'email': 'gundatsev@gmail.ru', 'mobile': 'XXXXXXXXXX', 'afm': 'XXXXXXXXX', 'dob': ‘YYYY-MM-DD TIME’ }

Параметры email, mobile, afm, dob могут указываться в любом порядке и в любом сочетании. Обязательно указание хотя бы одного параметра.   
Любой параметр может быть опущен  
  
После отработки запроса сервис возвращает ID найденного клиента.  
Если клиент с указанными параметрами не найден возвращается 0.  
Если найдено несколько клиентов с указанными параметрами, то возвращается ID последнего клиента, т.е. максимальное значение из найденных ID   
Использование только одного параметра dob технически возможно, но правильно его использовать в сочетании с другим параметром, например, afm.

Например,  
http://10.0.4.7:8091/clients/GetClient\_ID/{'email': 'gundartsev@gmail.com'} – возврат ID клиента по email

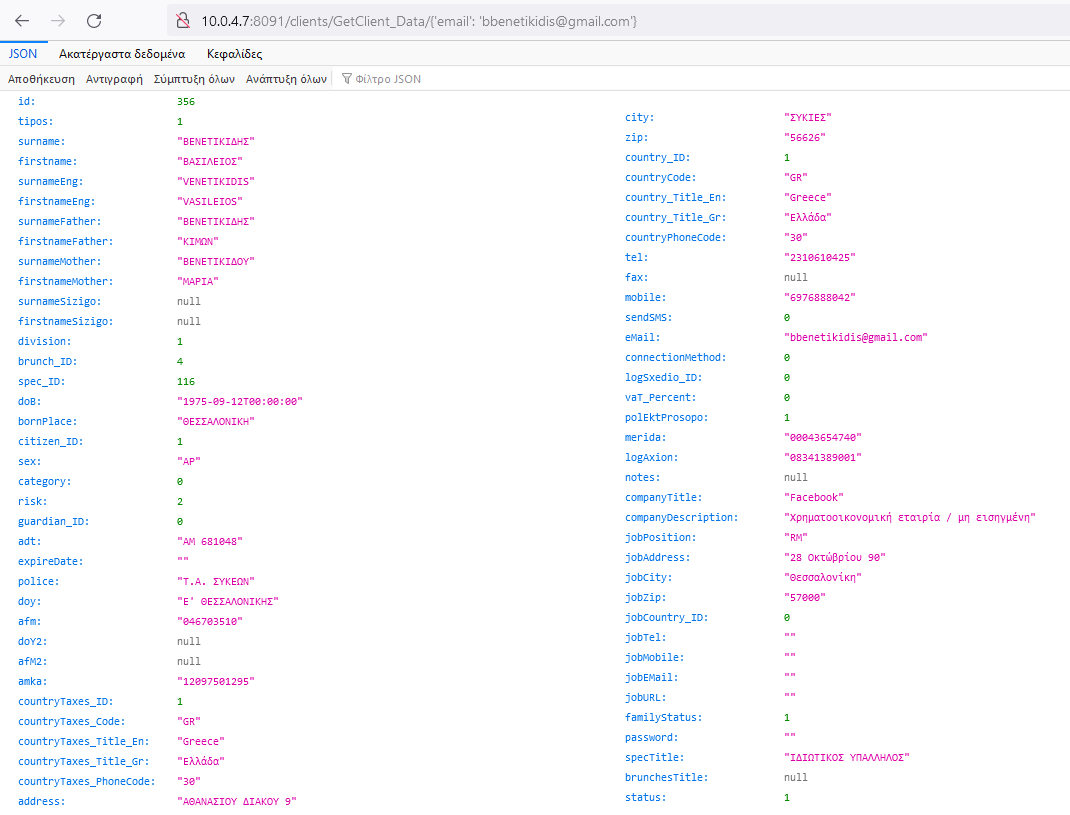
http://10.0.4.7:8091/client/getclient\_id/{'mobile': 'XXXXXXXXXX', 'afm': 'XXXXXXXXX'}– возврат ID клиента по mobile и afm

* 1. **Сервис определения полного набора данных клиента по любому сочетанию параметеров email, mobile, afm, dob.**

Вызов сервиса осуществляется методом GET с помощью команды

<http://10.0.4.7:8091/clients/>getclient\_data/{'email': 'gundatsev@gmail.ru', 'mobile': 'XXXXXXXXXX', 'afm': 'XXXXXXXXX', 'dob': ‘YYYY-MM-DD TIME’ }

Параметры email, mobile, afm, dob могут указываться в любом порядке и в любом сочетании. Обязательно указание хотя бы одного параметра.   
Любой параметр может быть опущен  
  
После отработки запроса сервис возвращает полный набор данных о клиенте со структурой, показанной ниже.  
Если клиент с указанными параметрами не найден возвращается пустой набор данных.  
Если найдено несколько клиентов с указанными параметрами, то возвращается набор данных последнего клиента, т.е. набор данных с максимальным значением ID

Структура возвращаемого набора данных  


Например,   
http://10.0.4.7:8091/webusers/GetWebUser\_Data/{'mobile': '6971234567'} - возврат полного набора данных о web пользователя по номеру мобильного устройства 6971234567 .

* 1. **Сервис определения полного набора данных из списка клиентов по любому сочетанию параметеров email, mobile, afm, dob.**

Вызов сервиса осуществляется методом GET с помощью команды

<http://10.0.4.7:8091/clients/>getclient\_list/{'email': 'gundatsev@gmail.ru', 'mobile': 'XXXXXXXXXX', 'afm': 'XXXXXXXXX', 'dob': ‘YYYY-MM-DD TIME’ }

Параметры email, mobile, afm, dob могут указываться в любом порядке и в любом сочетании. Обязательно указание хотя бы одного параметра.   
Любой параметр может быть опущен  
  
После отработки запроса сервис возвращает полный набор данных о списке клиентов со структурой, показанной ниже.  
Если клиенты с указанными параметрами не найдены возвращается пустой набор данных.  
Если найдено несколько клиентов с указанными параметрами, то возвращается список всех найденных клиентов с их данными

Структура возвращаемого набора данных состоит из нескольких групп данных с одной и той же структурой. Количество групп данных зависит от количества найденных клиентов.

Например,   
http://10.0.4.7:8091/clients/getclient\_list/{'email': 'ceo@hellasfin.gr'} - возврат полного набора данных из списка клиентов с указанным email

Результирующий набор данных состоит из 3 групп данных. Группа первая – данные клиента с ID = 277, вторая – данные клиента с ID = 3227, третья – данные клиента с ID = 9085

Тип клиента задается в поле tipos: 1 – физическое лицо, 2 – юридическое лицо

* 1. **Сервис для ввода связи между двумя клиентами.**

Вызов сервиса осуществляется методом POST с помощью команды

<http://10.0.4.7:8091/clients/AddClients_Clients>

При вызове сервиса на сервер передаются ID обоих клиентов в JSON-формате со следующей структурой:

Client\_ID int,  
 Client2\_ID int,  
 Status int (0 – не подтверждено, 1-подтверждено) ,  
 DateIns smalldatetime

где Client\_ID – это ID клиента, который делает приглашение,  
 Client2\_ID – это ID клиента, которому делают приглашение,  
 Status = 0,  
 DateIns – дата ввода данной записи в базу данных

* 1. **Сервис для изменения статуса связи между двумя клиентами.**

Вызов сервиса осуществляется методом POST с помощью команды

<http://10.0.4.7:8091/clients/UpdateStatusClients_Clients>

При вызове сервиса на сервер передаются   
 int id – ID записи  
 int Status - новый status

* 1. **Сервис определения списка всех связей для указанного клиента с ID**

Вызов сервиса осуществляется методом GET с помощью команды

<http://10.0.4.7:8091/clients/GetClients_Clients/id=269>  
  
где ID – это Client\_ID клиента, для которого запрашивается список связей.  
  
После отработки запроса сервис возвращает данные обо всех найденных связях с указанием Client\_ID, Фамилии-Имени клиентовЮ их АФМ и e-mail

* 1. **Сервис удаления связи между клиентами.**

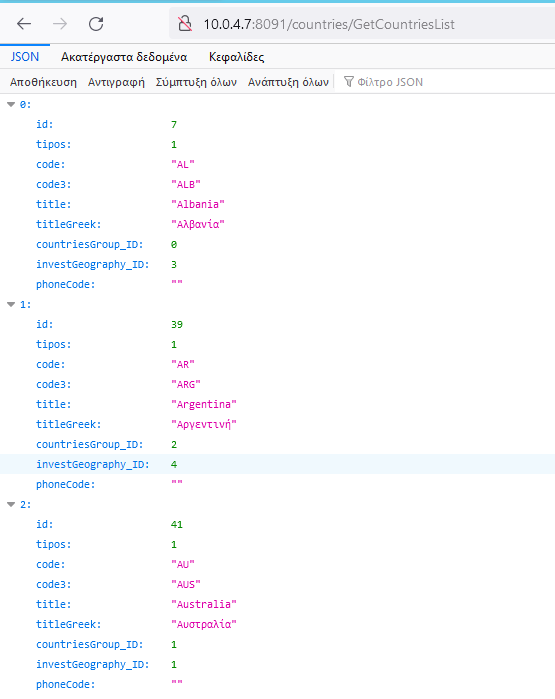
Вызов сервиса осуществляется методом POST с помощью команды

<http://10.0.4.7:8091/clients/DeleteClients_Clients>При вызове сервиса на сервер передаются   
 int id – ID связи, которая должна быть удалена

1. **Сервисы для работы со списком стран Countries**
   1. **Сервис определения списка стран**

Вызов сервиса осуществляется методом GET с помощью команды

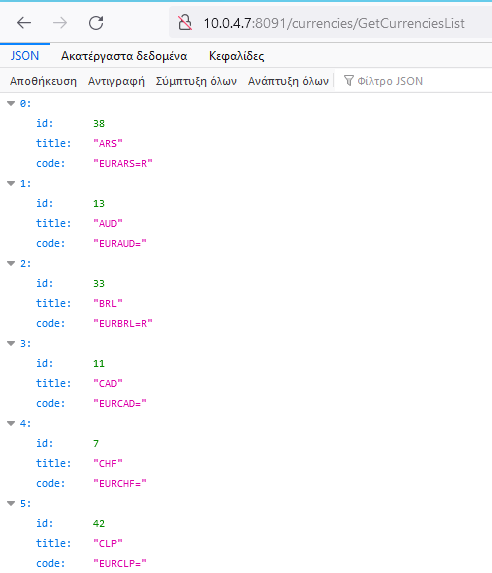
http://10.0.4.7:8091/countries/GetCountriesList

После отработки запроса сервис возвращает список всех стран:  
  


1. **Сервисы для работы со списком валют Currencies**
   1. **Сервис определения списка валют**

Вызов сервиса осуществляется методом GET с помощью команды

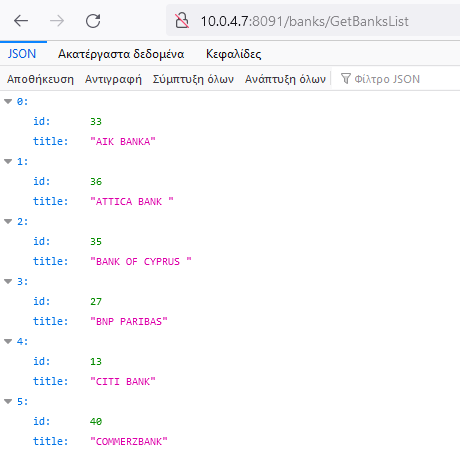
http://10.0.4.7:8091/currencies/GetCurrenciesList

После отработки запроса сервис возвращает список всех валют:  
  


1. **Сервисы для работы со списком банков Banks**
   1. **Сервис определения списка банков**

Вызов сервиса осуществляется методом GET с помощью команды

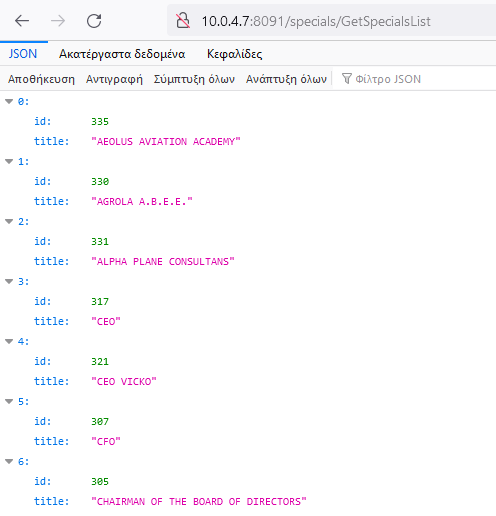
http://10.0.4.7:8091/banks/GetBanksList

После отработки запроса сервис возвращает список всех банков:  
 ****

1. **Сервисы для работы со списком специальностей Specials**
   1. **Сервис определения списка специальностей**

Вызов сервиса осуществляется методом GET с помощью команды

http://10.0.4.7:8091/specials/GetSpecialsList

После отработки запроса сервис возвращает список всех специальностей:  
  
****

1. **Сервисы для работы со списком отраслей деятельности клиентов Brunches**
   1. **Сервис определения списка отраслей**

Вызов сервиса осуществляется методом GET с помощью команды

<http://10.0.4.7:8091/brunches/GetBrunchesList>  
  
После отработки запроса сервис возвращает список всех отраслей:

**Graphical user interface, text, application

Description automatically generated**

**10. Сервисы для работы со списком документов клиентов ClientsDocFiles  
  
10.1 Сервис определения списка документов по любому сочетанию параметеров client\_id, contract\_id, precontract\_id, doctypes\_id.**

Вызов сервиса осуществляется методом GET с помощью команды  
  
http://10.0.4.7:8091/clientsdocfiles/GetClientsDocFilesList/{'client\_id': '356', 'contract\_id': '0', 'precontract\_id': '0', 'doctypes\_id': '3924' }

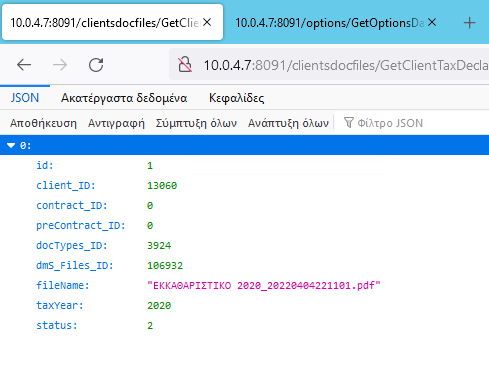
где doctypes\_id – код типа документа. Например, 3924 – это Εκκαθαριστικό  
  
Параметры client\_id, contract\_id, precontract\_id, doctypes\_id могут указываться в любом порядке и в любом сочетании. Обязательно указание хотя бы одного параметра.   
Любой параметр может быть опущен.  
  
После отработки запроса сервис возвращает список найденных документов со структурой, показанной ниже. **Graphical user interface, text, application

Description automatically generated**

**10.2 Сервис определения списка ΕΚΚΑΘΑΡΙΣΤΙΚΟ по client\_id, doctypes\_id**

Вызов сервиса осуществляется методом GET с помощью команды  
  
http://10.0.4.7:8091/clientsdocfiles/GetClientTaxDeclarationsList/{'client\_id': '13060', 'doctypes\_id': '3924', 'start\_year': '2016', 'last\_year': '2021' }  
  
где client\_id – ID клиента  
 doctypes\_id – код типа документа. Например, 3924 – это Εκκαθαριστικό  
 start\_year – начиная с года  
 last\_year - кончая годом

После отработки запроса сервис возвращает список найденных файлов со структурой, показанной ниже.



1. **Сервисы для работы со списком инвестиционных счетов клиентов Contracts**

**11.1 Сервис определения списка инвестииционных счетов клиента**Вызов сервиса осуществляется методом GET с помощью команды

<http://10.0.4.7:8091/webusers/>getwebuser\_id/{'email': 'gundatsev@gmail.ru', 'mobile': 'XXXXXXXXXX', 'afm': 'XXXXXXXXX', 'dob': ‘YYYY-MM-DD TIME’, 'client\_id': '1234' }

Параметры email, mobile, afm, dob, client\_id могут указываться в любом порядке и в любом сочетании. Обязательно указание хотя бы одного параметра.   
Любой параметр может быть опущен

1. **Сервисы для работы со списком банковских счетов клиентов ClientsBankAccounts**
   1. **Сервис определения списка банковских счетов клиента по его Client\_ID**

Вызов сервиса осуществляется методом GET с помощью команды

http://10.0.4.7:8091/ClientsBankAccounts/GetClientBankAccountsList\_Data/{'client\_id':'356}  
  
где client\_id – это ID клиента системы.

В ответ система возвращает список всех активных банковских счетов этого клиента с их основными данными:  
ID – ID счета в базе данных  
tipos – 0 – индивидуальный, 1- коллективный

Bank\_ID – ID банка

Bank\_Title – название банка

Client\_ID –

AccNumber – номер банковского счета

Currency – валюта банковского счета

**Graphical user interface, text, application, email

Description automatically generated**

* 1. **Сервис для ввода (добавления) нового банковского счета**Вызов сервиса осуществляется методом POST с помощью команды

[http://10.0.4.7:8091/ClientsBankAccounts /addrecord](http://10.0.4.7:8091/ClientsBankAccounts%20/addrecord)  
При вызове сервиса на сервер передаются данные о новом пользователе мобильного устройства в JSON-формате со следующей структурой:

int Client\_ID - ID клиента, которому принадлежит этот банковский счет

int Bank\_ID - ID банка, в котором открыт этот банковский счет

string Currency – валюта банковского счета

После отработки сервис возвращает ID нового банковского счета

* 1. **Сервис изменения статуса банковского счета**

Сервис изменяет статус банковский счет по его ID.  
 Вызов сервиса осуществляется методом POST с помощью команды

[http://10.0.4.7:8091/ ClientsBankAccounts /](http://10.0.4.7:8091/%20ClientsBankAccounts%20/)EditRecord  
  
При вызове сервиса на сервер передаются данные о новом пользователе мобильного устройства в JSON-формате со следующей структурой:

int ID - ID записи данного банковского счета

int Status - новый статус банковского счета.   
 Значение 0 означает, что счет аннулирован, 1 – счет активен

При успещном завершении работы сервис возвращает ID записи данного банковского счета. Если статус не изменен по какой-то причине, то возвращается 0

**13. Сервисы для работы со списком типов запросов клиентов ClientsRequestsTypes**Клиенты HF могут присылать различные типы запросов:

|  |  |  |  |
| --- | --- | --- | --- |
| Тип запроса | Наименование запроса | DocType1\_ID | DocType2\_ID |
| 1 | Αίτημα αλλαγής ταυτότητας | 2 | 0 |
| 2 | Αίτημα αλλαγής κινητού τηλεφώνου | 9418 | 9328 |
| 3 | Αίτημα αλλαγής σταθερού τηλεφώνου | 9418 | 9328 |
| 4 | Αίτημα αλλαγής email επικοινωνίας | 9431 | 0 |
| 5 | Αίτημα αλλαγής διεύθυνσης κατοικίας | 0 | 9328 |
| 6 | Αίτημα αλλαγής ΑΦΜ | 1 | 0 |
| 7 | Αίτημα προσθήκης εκκαθαριστικού | 3924 | 0 |
| 8 | Αίτημα αλλαγής επαγγέλματος | 4 | 0 |
| 9 | Αίτημα αλλαγής xώρας φορολόγησης | 1 | 0 |
| 10 | Αίτημα αλλαγής ειδικής κατηγορίας προσώπου | 0 | 0 |
| 11 | Αίτημα προσθήκης νέου τραπεζικού λογαριασμού | 5 | 0 |
| 12 | Αίτημα προσθήκης νέου συνδεδεμένου ανήλικου προσώπου | 9432 | 9432 |
| 13 | Αίτημα αλλαγής W8 ΒΕΝ | 58 | 0 |
| 14 | Αίτημα αλλαγής Διαβατηρίου | 9267 | 0 |
| 15 | Αίτημα αλλαγής Χρηματιστηριακής Μερίδας | 0 | 0 |
| 16 | Αίτημα αλλαγής Λογαριασμόυ Αξιών | 0 | 0 |
| 17 | Αίτημα αλλαγής ΑΜΚΑ | 9419 | 0 |
| 18 |  |  |  |
| 19 | Άνοιγμα Νέου Επενδυτικού Λογαριασμού | 0 | 0 |

1. **Сервисы для работы со списком запросов клиентов ClientsRequests**

**14.1 Сервис для ввода нового запроса от клиента**Вызов сервиса осуществляется методом POST с помощью команды

http://10.0.4.7:8091/clientsrequests/AddClientsRequests  
При вызове сервиса на сервер передаются данные о новом запросе в JSON-формате со следующей структурой:

**Client\_ID** int - client\_id  
**Group\_ID** string - идентификатор группы заявок. Все заявки из одной группы должны иметь одно и то же значение Group\_ID  
 если заявка индивидуальная, то Group\_ID = “”  
**Tipos**  smallint - тип заявки (список приведен выше)  
**Aktion**  smallint - действие (0-ввод новой заявки, 1-изменение заявки,2-удаление)  
**Source\_ID** int - кто ввел заявку: 1 - сам клиент, 2 - наш сотрудник   
**Description** nvarchar (1000) - содержание заявки. Оно для каждого типа заявки свое. Подробное описание приведено ниже  
**Warning**  nvarchar(500) - текст сообщения, которое в случае направильного ввода заявки пишет наш сотрудник, проверяющий заявку  
**DateIns**  smalldatetime - дата ввода заявки  
**DateWarning**  smalldatetime - дата отправки предупредительного e-mail  
**DateClose** smalldatetime - дата закрытия (исполнения) заявки  
**User\_ID** int - ID нашего сотрудника, который ввел заявку для случая Source\_ID = 2. Если заявку ввел сам клиент т.е. Source\_ID = 2, то User\_ID = 0  
**Status** smallint - статус заявки:   
1 - заявка находится на этапе подготовки (draft)   
2 - заявка отправлена на рассмотрение ответственным лицам (Compliance, BackOffice)  
3 – сделана первая проверка заявки   
4 - сделана вторая проверка заявки   
5 – сделана первая проверка результатов видеочата   
6 - сделана вторая проверка результатов видеочата  
7 - заявка проверена и отклонена, т.к. неправильно введена. Комментарий проверяющего хранится в поле Warning  
0 - заявка аннулирована, для клиента удалена  
**VideoChatStatus** smallint - статус видеочата по данной заявке:   
0 – видеочат не нужен  
1 - видеочат нужен, но он пока не проведен  
2 - видеочат проведен и сделана первая проверка  
3 - видеочат проведен, сделана первая проверка, а значит он успешно завершен  
4 - видеочат проведен, но завершен отменой запросов (аннулирован)  
**VideoChatFile** string **-** имя файла, в котором записан видеочат с клиентом  
  
Если запрос введен успешно, то после завершения сервис возвращает ID нового запроса, в противном случае возвращает 0.  
  
Описание поля **Description**.  
Поле Description предназначено для описания данных всех типов запросов.   
Это строковое поле, содержащее данные в JSON-формате. Строка состоит из нескольких элементов, в конце каждого из которых стоит символ ~.

В основном, MobiApp в поле Description будет указывать набор файлов-документов, которые загружает пользователь. Но есть запросы клиентов, в которых поле Description должно содержать и другие данные.   
Наборы файлов-документов могут содержать переменное количество файлов, разделеных знаком^ . Каждый файл описывается двумя аттрибутами: именем файлы и его типом.  
Например, {'file\_name' : 'pay\_1.pdf','file\_type' : 25}^{'file\_name' : 'bill\_2.pdf','file\_type' : 83}^ описывает два файла pay\_1.pdf и bill\_2.pdf. Первый из них имеет тип документа 25, а второй тип документа 83. Первый файл описывается так  
 {'file\_name' : 'pay\_1.pdf','file\_type' : 25}^  
а второй файл – так  
 {'file\_name' : 'bill\_2.pdf','file\_type' : 83}^

Кроме наборов файлов-документов, поле Description может содержать аттрибут action, обозначающий тип действия, например,   
 {' action ' : '1'}~  
где 0-ввод нового значения, 1-замена, 2-удаление,  
  
а также ответ на вопрос является ли пользователем хозяином мобильного или жилья  
 {'owner' : '0'}~  
0 – означает нет, 1 – означает да

Содержание поля Description для каждого конкретного типа запроса:  
  
- замена номера тавтотиты  
{'file\_name' : 'tavtotita\_17-12-2021\_15-36-47.png','file\_type' : 2}^{'file\_name' : 'tavtotita\_17-12-2021\_15-36-47.png','file\_type' : 2}^~

- замена номера мобильного телефона  
{'action' : '0'}~ {'owner' : '0'}~{'new\_number' : '6971234567'}~{'file\_name' : 'mobile\_1.pdf',’file\_type' : 9418}^{'file\_name' : 'mobile\_2.pdf',’file\_type' : 52}^~ {'file\_name' : 'file3.pdf',’file\_type' : 9418}^~

- замена номера стационарного телефона  
{'owner' : '0'}~ {'new\_number' : '2310234567'}~ {'file\_name' : 'tel\_1.pdf',’file\_type' : 9418}^{'file\_name' : 'tel\_2.pdf',’file\_type' : 31}^~ {'file\_name' : 'file3.pdf',’file\_type' : 9418}^~

- замена e-mail   
 {'new\_email' : 'newemail@yahoo.com'}~

-замена домашнего адреса  
 {'owner' : '0'}~{'file\_name' : 'add\_20211221141836.pdf','file\_type' : 4}^{'file\_name' : 'add\_20211221141924.pdf','file\_type' : 4}~{'file\_name' : 'yd\_20211221141958.png','file\_type' : 7}^{'file\_name' : 'yd\_20211221142041.png', 'file\_type' : 7}^~

- замена ΑΦΜ  
 {'file\_name' : 'afm\_17-12-2021\_15-36-47.png','file\_type' : 1}^{'file\_name' : 'afm\_17-12-2021\_15-36-47.png','file\_type' : 1}^~

- ввод ΕΚΚΑΘΑΡΙΣΤΙΚΟ  
 {'tax\_declaration' : '0'}~{ ΕΚΚΑΘΑΡΙΣΤΙΚΟ 2018.pdf^ΕΚΚΑΘΑΡΙΣΤΙΚΟ 2019.pdf^}~{ 'file\_name' : 'abc.pdf', 'file\_type' : '7'}^{'file\_name' : 'def.pdf', 'file\_type' : '8'}^~

- замена специальности  
 { 'new\_spec\_id' : '3'}~ {'file\_name' : 'spec\_17-12-2021\_15-36-47.png','file\_type' : 4}^{'file\_name' : 'spec\_17-12-2021\_15-36-47.png','file\_type' : 4}^~

- замена страны налогообложения  
 { 'new\_country\_id' : '25'}~ {'file\_name' : 'tax\_17-12-2021\_15-36-47.png','file\_type' : 1}^{'file\_name' : 'tax\_17-12-2021\_15-36-47.png','file\_type' : 1}^~

- замена специальной категории клиента  
{'cat' : '0','file\_name' : ''}~{'file\_name' : 'dilosi.pdf'}~{'cat' : '1','file\_name' : 'abc.pdf'}~{'cat' : '2','file\_name' : 'def.pdf'}~  
 где cat – номер выбранной категории  
 file\_name – имя соответствующего файла  
  
- замена номера паспорта   
 {'file\_name' : 'psp\_17-12-2021\_15-36-47.png','file\_type' : 9267}^{'file\_name' : 'psp\_17-12-2021\_15-36-47.png','file\_type' : 9267}^~

- замена Μερίδα  
 {'file\_name' : 'psp\_17-12-2021\_15-36-47.png','file\_type' : 0}^{'file\_name' : 'psp\_17-12-2021\_15-36-47.png','file\_type' : 0}^~

- замена Λογαριασμόυ Αξιών  
 {'file\_name' : 'psp\_17-12-2021\_15-36-47.png','file\_type' : 0}^{'file\_name' : 'psp\_17-12-2021\_15-36-47.png','file\_type' : 0}^~

- замена ΑΜΚΑ  
 {'file\_name' : 'psp\_17-12-2021\_15-36-47.png','file\_type' : 9419}^{'file\_name' : 'psp\_17-12-2021\_15-36-47.png','file\_type' : 9419}^~

- ввод нового банковского счета   
 {'file\_name' : 'ba\_17-12-2021\_15-36-47.png','file\_type' : 0}^{'file\_name' : 'ba\_17-12-2021\_15-36-47.png','file\_type' : 0}^~

- изменение статуса банковского счета   
{'action' : '2'}~{id' : '156', ‘status’ : ‘0’}~

- ввод (добавление) взрослого клиента HellasFin к списку ΣΥΝΔΕΔΕΜΕΝΑ ΠΡΟΣΩΠΑ  
 {'client\_id' : '356'}~ {afm' : 123456789}~{'file\_name' : 'ba\_17-12-2021\_15-36-47.png','file\_type' : 0}^{'file\_name' : 'ba\_17-12-2021\_15-36-47.png','file\_type' : 0}^~

- ввод (добавление) ребенка к списку ΣΥΝΔΕΔΕΜΕΝΑ ΠΡΟΣΩΠΑ предусматривает загрузку разных документов. В общем случае, формат записи будет таким:  
{'client\_id' : '123'}~{'surname' : 'DELLAS'}~{ 'firstname' : 'NIKOS '}~{ 'dob' : '15/04/2009' }~{'file\_name' : file1.pdf','file\_type' : 9432}^{'file\_name' : file2.pdf','file\_type' : 2}^~{'file\_name' : file11.pdf','file\_type' : 9432}^{'file\_name' : file22.pdf','file\_type' : 3968}^~{'file\_name' : file111.pdf','file\_type' : 3968}~~~ file1111.pdf','file\_type' : 3968}^{'file\_name' : file2222.pdf','file\_type' : 1}^~

Где 9432 – разрешение от отца, 9432 – разрешение от матери, 3968 – тавтотита, 55 – свидетельство о рождении, 1 – документ от АФМ

------------------Требуют уточнения---------------------  
13 Αίτημα αλλαγής W8 ΒΕΝ

-------------------------------------------------------------------

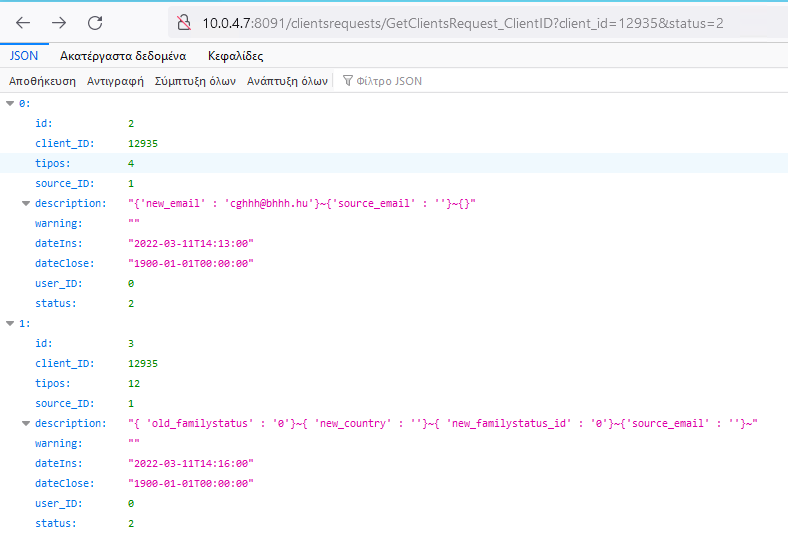
**14.2 Сервис определения (чтения) данных запроса от клиента по ID запроса**Вызов сервиса осуществляется методом GET с помощью команды

[http://10.0.4.7:8091/ clientsrequests /get](http://10.0.4.7:8091/%20clientsrequests%20/get)clientsrequest\_data?id=1234  
где id – ID записи запроса  
  
После отработки запроса сервис возвращает все данные о запросе с указанным ID



**14.3 Сервис определения (чтения) данных о запросах по Client\_ID, т.е. ID автора запросов**Вызов сервиса осуществляется методом GET с помощью команды

[http://10.0.4.7:8091/ clientsrequests /](http://10.0.4.7:8091/%20clientsrequests%20/)GetClientsRequest\_ClientID?client\_id=4469&status=2  
где client\_id – ID клиента – автора запроса. Данный параметр обязателен.  
 status – статус запроса. Значения данного парамера приведены в разделе14.1.  
 Данный параметр не обязателен.   
После отработки запроса сервис возвращает наборы данных о запросах клиента с указанным client\_ID. Если указан и статус, например, status=2, то в результирующий набор попадают только запросы с указанным статусом.



**14.4 Сервис изменения данных запроса от клиента**

Вызов сервиса осуществляется методом POST с помощью команды

http://10.0.4.7:8091/clientsrequests/UpdateClientsRequests  
  
При вызове сервиса на сервер передаются данные о новом запросе в JSON-формате со следующей структурой:

**ID** int - ID редактируемой заявки  
**Aktion**  smallint - действие (1-изменение заявки)  
**Description** nvarchar (1000) - содержание заявки. Оно для каждого типа заявки свое.   
**Warning**  nvarchar(500) - текст сообщения, которое в случае направильного ввода заявки пишет наш сотрудник, проверяющий заявку  
**Status** smallint - статус заявки:   
**VideoChatStatus** smallint - статус видеочата по данной заявке   
**VideoChatFile** string **-** имя файла, в котором записан видеочат с клиентом  
  
Обязательным является только ID редактируемой заявки и как минимум один из других параметров

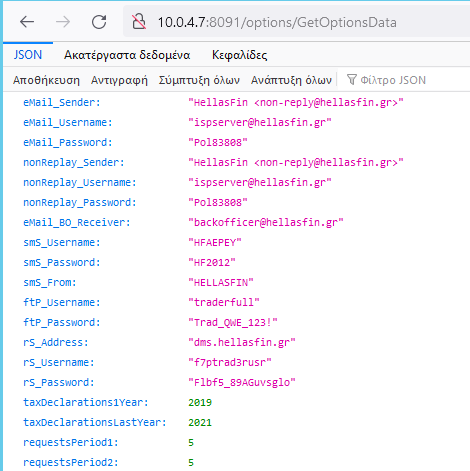
**14.5 Сервис изменения статуса запроса от клиента**Вызов сервиса осуществляется методом POST с помощью команды

[http://10.0.4.7:8091/ clientsrequests/updatestatus](http://10.0.4.7:8091/%20clientsrequests/updatestatus)При вызове сервиса на сервер передаются   
 int id   
 int status,  
где id – ID записи запроса, статус которой должен быть изменен  
 status – статус запроса. Возможные значения этого статуса описаны в пункте 14.1  
  
В MobiApp устанавливаются только статусы 2 и 0. Статус 2 устанавливается при вводе нового запроса, т.е. в сервисе AddClientsRequests, а статус 0 при аннуляции (удалении).

1. **Сервисы для работы со списком параметров Options**
   1. **Сервис определения списка параметров системы**

Вызов сервиса осуществляется методом GET с помощью команды  
 <http://10.0.4.7:8091/options/GetOptionsData>

После отработки запроса сервис возвращает список параметров, показанных ниже



Где eMail\_ … - параметры адреса для отправки сообщений  
 nonReplay\_\_ … - параметры адреса для отправки сообщений, не требующих ответа  
 eMail\_BO\_Receiver – адрес-получатель сообщений, отправленных на BackOffice  
 smS\_ … - параметры адреса для отправки SMS-сообщений  
 ftp\_ … - параметры адреса для FTP-доступа в данным из DMS  
 rs\_ … - параметры удаленного сервера  
 taxDeclarations1Year – начальный год для загрузки Εκκαθαριστικά  
 taxDeclarationsLastYear – последний год для загрузки Εκκαθαριστικά  
 requestsPeriod1 - кол-во дней ожидания до отправки предупреждающего сообщения  
 requestsPeriod2 - кол-во дней, через которое удаляется заявка и клиент