

Документация модуля skydns_website.api.application (API Агентов)

Доступные методы API

Модуль хранения представлений JSON-RPC API. Используется агентами.

Curl запросы для тестирования выглядят так:

```
curl -X POST http://localhost:8000/api/json/v2 -L -u user:password > -d '{"id":0, "jsonrpc":'
```

В простейшем случае связку кредитов можно использовать такую:

```
plantest-premium@skydns.ru: plantest
```

```
skydns_website.api.application.views.add_domain(request, profile, choice, domain, ip=None, comment=None)
```

JSONRPC Method: addDomain

Добавить домен.

Параметры:	• profile – профиль пользователя • choice – тип домена • domain – домен • ip – IP адрес (если передать IP, то будет добавлен не домен, а алиас) • comment – комментарий к домену
Результат:	id добавленного домена.
Тип результата:	int

```
skydns_website.api.application.views.add_profile(request, name)
```

JSONRPC Method: addProfile

Метод создает профиль и возвращает словарь с данными о профиле.

Параметры:	name – имя создаваемого профиля
Результат:	данные созданного профиля в формате: <pre>{ 'id': id профиля, 'default': является ли профиль профилем по умолчанию, 'token': токен, 'name': название профиля, 'white_list_only': включена ли фильтрация только по белому списку, 'is_schedule_enabled': включено ли расписание, 'safe_search_enabled': статус безопасного поиска, 'safe_youtube_enabled': статус фильтрации YouTube (для API v2), 'block_unknown_enabled': статус фильтрации неизвестных сайтов (для API v2) }</pre>
Тип результата:	dict

```
skydns_website.api.application.views.categories(request, language='en')
```

JSONRPC Method: categories

Получить список сгруппированных категорий.

Параметры:	language – язык выдачи
-------------------	-------------------------------

Результат:	список сгруппированных категорий в формате: <pre>[{ "title": "group1", "items": [{ "title": "item1", "id": 1 }] }]</pre>
Тип результата:	list

```
skydns_website.api.application.views.change_schedule(request, profile, segments)
```

JSONRPC Method: setSchedule

Установить расписание фильтрации.

Параметры:	• profile – профиль пользователя • segments – список сегментов вида: [[<offset1>, <true false>], [<offset2>, <true false>]]
Результат:	[True]
Тип результата:	list

```
skydns_website.api.application.views.clear_domains(request, profile, choice )
```

JSONRPC Method: clearDomains

Удалить все домены определенного типа у профиля.

Параметры:	• profile – профиль пользователя • choice – тип домена
Результат:	None

```
skydns_website.api.application.views.domains(request, profile, choice)
```

JSONRPC Method: domains

Получить список доменов определенного типа для профиля.

Параметры:	• profile – профиль пользователя • choice – тип домена ('black', 'white' или 'alias')
Результат:	для запроса чёрных или белых доменов - список словарей в формате: <pre>{ "domain": домен, "id": id домена }</pre> для запроса алиасов - список словарей в формате: <pre>[{ "domain": домен, "id": id домена, "ip": ip-адрес }]</pre>
Тип результата:	list

```
skydns_website.api.application.views.enable_schedule(request, profile, flag)
```

JSONRPC Method: setScheduleEnabled

Включить/выключить расписание для профиля.

Параметры:	• profile – профиль пользователя • flag – флаг включить/выключить
Результат:	flag
Тип результата:	bool

```
skydns_website.api.application.views.feedback(request, title, message)
```

JSONRPC Method: feedback

Отправить сообщение обратной связи пользователя.

Параметры:	• title – заголовок сообщения для отправки • message – сообщение для отправки
Результат:	None

```
skydns_website.api.application.views.get_advertising(request)
```

JSONRPC Method: getAdvertising

Получить информацию об акции.

Результат:	[<strAdvUrl>, <strImageUrl>]
Тип результата:	list

```
skydns_website.api.application.views.get_agent_profile(request, uid, hostname, version, os_info, address)
```

JSONRPC Method: getProfile

Возвращает необходимые настройки (токен, UID) для работы агентского приложения. Если передан UID - метод вернёт информацию об уже существующем профиле с этим UID. Если UID пуст, но переданы все остальные параметры - будет создан новый профиль.

Параметры:	• uid – uid • hostname – имя хоста • version – версия • os_info – информация • address – IP адрес
Результат:	список с данными профиля агента: [uid, token, profile_id]
Тип результата:	list

```
skydns_website.api.application.views.get_categories_daily_stat(request, profile, lang='en', categories_list=None)
```

JSONRPC Method: getCategoryDailyStats

Получить статистику по категориям за сутки для нужного профиля.

Параметры:	• profile – профиль пользователя • lang – язык выдачи • categories_list – список идентификаторов категорий. Если его нет, то будет получена статистика блокировок по всем категориям для профиля пользователя
Результат:	словарь формата: { <code><category_name>: <value> ...</code> }
Тип результата:	dict

```
skydns_website.api.application.views.get_filtering_status(request)
```

JSONRPC Method: getFilteringStatus

Получить состояние фильтрации пользователя. :return: <состояние фильтрации>
:rtype: bool

```
skydns_website.api.application.views.get_limited_agent_profile(request, uid, hostname, version, os_info, address)
```

JSONRPC Method: getLimitedProfile

Это модификация метода get_agent_profile (getProfile). Этот метод проверяет, сколько уже выдано токенов пользователю (мобильных или десктопных Linux), и если это число превышает максимум (ПФ max_mobile_agents и max_desktop_agents соответственно), отдаёт фейловые токены ('', 9999999999, <profile_id>) Если запрос не мобильного или десктопного Linux агента, токены генерируются без условий. Если передан UID - метод вернёт информацию об уже существующем профиле с этим UID. Если UID пуст, но переданы все остальные параметры - будет создан новый профиль.

Параметры:	• uid – uid • hostname – имя хоста • version – версия • os_info – информация • address – IP адрес
Результат:	список с данными профиля агента: [uid, token, profile_id]
Тип результата:	list

```
skydns_website.api.application.views.get_linux_supporting(request, tag=None, version=None)
```

JSONRPC Method: getLinuxSupporting

Получить версии Linux-агента: минимальную поддерживаемую (подлежащую принудительному обновлению) и текущую.

Параметры:	• tag – тег для получения версий (по умолчанию - самая новая без разбора версии ОС) • version – текущая версия установленного агента
-------------------	---

Результат:	словарь вида: <pre>{ 'minimalSupported': <минимально поддерживаемая версия (целое неотрицательное число)>, 'current': <текущая версия мобильного агента (целое неотрицательное число)>, 'url': <url до пакета>, 'checksum': <контрольная сумма пакета>, 'announces': анонсы версий, от version до последней, если version не указан, то все [{ 'version': <номер версии> 'announce': <анонс для этой версии> }] }</pre>
Тип результата:	dict

```
skydns_website.api.application.views.get_plans_list(request, mobile=False)
```

JSONRPC Method: getPlans

Получить список тарифных планов.

Параметры:	mobile – True - выдавать только мобильные планы, False - все
-------------------	---

Результат:	список тарифов в формате: <pre>[{ 'code': код тарифа, 'name': название тарифа, 'price': стоимость тарифа, 'trialPeriod': сколько дней триального периода на тарифе, 'period': на сколько месяцев покупается тариф, 'isMobile': является ли тариф мобильным, 'description': URL страницы сайта с описанием тарифа }]</pre>
Тип результата:	list

```
skydns_website.api.application.views.get_popular_report(request, start, end, profile_id=None)
```

JSONRPC Method: getPopularReport

Отчет о популярных запросах пользователя за определенный период времени.

Параметры:	• start – дата начала периода для получения статистики в формате YYYY-MM-DD • end – дата окончания периода для получения статистики в формате YYYY-MM-DD • profile_id – id профиля. Если не задан, будет создан отчет по всем профилям пользователя
-------------------	--

Результат:	словарь вида: <pre>{ "labels": ["example.com", "google.com", "asdfg.com"], "datasets": [{"label": "Requests", "data": [630, 474, 290]}, {"label": "NXdomain", "data": [0, 0, 290]}, {"label": "Blocks", "data": [630, 0, 0]}] }</pre>
-------------------	---

```
skydns_website.api.application.views.get_ssl_certificate(request, version=1)
```

JSONRPC Method: getSSLCertificate

Получить ссылку на корневой SSL-сертификат СкайДНС.

Результат:	<прямая ссылка до сертификата>
Тип результата:	str

```
skydns_website.api.application.views.get_ssl_certificate_data_v2(request, version=1)
```

Получить ссылки на скачивание SSL-сертификатов СкайДНС.

```
:return: {

  "certs": [
    "cert/url", "cert/url"
  ], "expires": "YYYY-MM-DDTHH:MM:SS"

} :rtype: dict curl test request: curl -location -request POST
'https://skynds.ru/api/json/v2' -header 'Content-Type: application/json' -data-raw '{
```

```

    "id": 0, "jsonrpc": "2.0", "method": "getSSLCertificatesDataV2", "params": ["3"]
  }'

  >{
    "id": 0, "jsonrpc": "2.0", "result": {

      "expires": "2028-01-19T00:00:00", "certs": [

        "https://www.skydns.ru/userfiles/certs/skydns_iss.crt",
        "https://www.skydns.ru/userfiles/certs/skydns_root.crt"
      ]

    }

  }

```

```
skydns_website.api.application.views.get_user_plan_info(request)
```

JSONRPC Method: getPlan

Получить информацию о тарифе пользователя.

Результат:	информация о тарифе в формате: { 'name': название плана, 'expired': количество дней до истечения оплаченного периода, 'isMobile': поддерживает ли тариф мобильные устройства (true false) }
Тип результата:	dict

```
skydns_website.api.application.views.get_version(request, tag=None)
```

JSONRPC Method: getAPCVersion

Получить версии мобильного приложения.

Параметры:	tag – тег для получения версий, по умолчанию 'default'
-------------------	---

Результат:	словарь вида: { ‘minimalSupported’: <минимально поддерживаемая версия (целое неотрицательное число)>, ‘current’: <текущая версия мобильного агента (целое неотрицательное число)> }
Тип результата:	dict

```
skydns_website.api.application.views.intentional_crash(request)
```

JSONRPC Method: intentionalCrash

Специальный метод, всегда выдающий JsonRpcRuntimeError.

Результат:	None
-------------------	------

```
skydns_website.api.application.views.multi_schedule(request, profile)
```

JSONRPC Method: multiSchedule

Получить расписание фильтрации.

Параметры:	profile – профиль пользователя
Результат:	расписание фильтрации в формате: <pre>[[<offset1>, <True False>], [<offset2>, <True False>], ...]</pre>
Тип результата:	list

```
skydns_website.api.application.views.my_ip(request)
```

JSONRPC Method: myip

Получение текущего IP адреса клиента.

Результат:	IP адрес клиента
-------------------	------------------

Тип результата:	str
-----------------	-----

```
skydns_website.api.application.views.profile_schedule_activity(request, profile)
```

JSONRPC Method: profileScheduleActivity

Получить количество минут до изменения состояния фильтрации профилей.

Параметры:	profile - профиль пользователя
Результат:	[<флаг: работает ли фильтрация по профилю>, <количество минут до смены состояния>] количество минут = -1, если расписание выключено или нет других состояний.

Пример:

```
first segment state last segment current offset
change | /

v v v v

|/////////| False |/////////X/| 0—260—1090—10k |
True |/////////| True |
```

Тип результата:	list
-----------------	------

```
skydns_website.api.application.views.profiles(request)
```

JSONRPC Method: profiles

Получение списка профилей.

Результат:	список профилей в формате: <pre>[{ 'id': id профиля, 'default': является ли профиль профилем по умолчанию, 'token': токен, 'name': название профиля, 'white_list_only': включена ли фильтрация только по белому списку, 'is_schedule_enabled': включено ли расписание, 'safe_search_enabled': статус безопасного поиска, 'safe_youtube_enabled': статус фильтрации YouTube (для API v2), 'block_unknown_enabled': статус фильтрации неизвестных сайтов (для API v2) }]</pre>
Тип результата:	list

```
skydns_website.api.application.views.register_user(request, username, password)
```

JSONRPC Method: register

Регистрация нового пользователя в облачном сервисе контент-фильтрации SkyDNS.

Параметры:	• username – имя пользователя • password – пароль создаваемого пользователя
Результат:	[True], если регистрация прошла успешно, [False, "error"] в противном случае
Тип результата:	list

```
skydns_website.api.application.views.remove_agent_profile(request, uid)
```

JSONRPC Method: removeAgentProfile

Метод удаляет AgentInfo отфильтрованный по заданному uid атрибуту.

Параметры:	uid – uid
Результат:	количество удалённых записей
Тип результата:	int

```
skydns_website.api.application.views.remove_domain(request, profile, choice, id)
```

JSONRPC Method: removeDomain

Удалить домен у заданного профиля.

TODO: параметр choice можно убрать и выбирать домен только по id.

Параметры:	• profile – профиль пользователя • choice – тип домена • id – id домена
Результат:	None

```
skydns_website.api.application.views.remove_profile(request, profile)
```

JSONRPC Method: removeProfile

Метод удаляет указанный профиль.

Параметры:	profile – числовой id удаляемого профиля
Результат:	None

```
skydns_website.api.application.views.rename_profile(request, profile_id, name)
```

JSONRPC Method: renameProfile

Метод устанавливает новое имя для профиля. Возвращает актуальное имя.

Параметры:	• profile_id – числовой id профиля, имя которого необходимо изменить • name – новое имя профиля
Результат:	новое имя профиля

Тип результата:	string
------------------------	--------

```
skydns_website.api.application.views.send_debug_info(request, uid, info)
```

Отправить логи для отладки.

Параметры:	info – информация лога формата [{"command_title": <log_name>, "command_info": <log_message>}, ...]
Результат:	количество созданных объектов

curl запрос для тестирования выглядит так:

```
$ curl -X POST http://localhost:8000/api/json/v2 -L -user plantest-premium@skydns.ru:plantest1 -d '{"id": 0, "jsonrpc": "2.0", "method": "sendDebugInfo", "params": [{"uid", [{"command_title": <log_name>, "command_info": <log_message>}]}'
```

```
skydns_website.api.application.views.send_log(request, profile_id=None, log_content='', os_info=None)
```

JSONRPC Method: sendLog

Фиксация в лог-файл лога пользователя с агентского решения.

Параметры:	• profile – профиль пользователя • log_content – лог пользователя • os_info – версия системы
Результат:	True
Тип результата:	boolean

```
skydns_website.api.application.views.send_package(request, package, file_name, system_tag, version, checksum, announce)
```

JSONRPC Method: sendPackage

Загрузка новых версий агентов на сервер.

Параметры:	• package – содержимое пакета в Base64 • file_name – имя файла • system_tag – тег версии ОС • version – версия агента • checksum – контрольная сумма • announce – анонс версии
Результат:	True
Тип результата:	boolean

```
skydns_website.api.application.views.set_agent_profile(request, uid, profile_id)
```

JSONRPC Method: setProfile

Установить профиль AgentInfo.

Параметры:	• uid – uid AgentInfo • profile_id – id профиля, который требуется установить
Результат:	None

```
skydns_website.api.application.views.set_block_unknown_enabled(request, profile, flag)
```

JSONRPC Method: setBlockUnknownEnabled

Включить/выключить блокировку неизвестных сайтов.

Параметры:	• profile – профиль пользователя • flag – флаг включить/выключить
Результат:	None

```
skydns_website.api.application.views.set_categories(request, profile_id, cat_ids_list)
```

JSONRPC Method: setFilterCats

Метод привязывает список категорий фильтрации к профилю пользователя и возвращает актуальный список категорий.

Параметры:	• profile_id – числовой id профиля • cat_ids_list – список id категорий
Результат:	актуальный список id категорий
Тип результата:	list

```
skydns_website.api.application.views.set_category(request, profile, cat, state)
```

JSONRPC Method: setFilterCat

Метод позволяет изменить состояние фильтра для категории.

Параметры:	• profile – профиль пользователя • cat – числовой id категории • state – булево значение состояния категории (true/false)
Результат:	None

```
skydns_website.api.application.views.set_filtering_status(request, enabled)
```

JSONRPC Method: setFilteringStatus

Установить состояние фильтрации пользователя. :param enabled: состояние фильтрации для установки

Результат:	<состояние фильтрации>
Тип результата:	bool

```
skydns_website.api.application.views.set_networks(request, profile, hostname)
```

JSONRPC Method: updateNic

Установить динамический IP адрес. IP адрес определяется автоматически из запроса.

Параметры:	• profile – профиль пользователя • hostname – имя хоста
Результат:	установленный IP адрес
Тип результата:	str

```
skydns_website.api.application.views.set_safe_search(request, profile, flag)
```

JSONRPC Method: setSafeSearchEnabled

Включить/выключить работу по безопасному поиску.

Параметры:	• profile – профиль пользователя • flag – флаг включить/выключить
Результат:	None

```
skydns_website.api.application.views.set_safe_youtube(request, profile, flag)
```

JSONRPC Method: setSafeYoutubeEnabled

Включить/выключить безопасный Youtube.

Параметры:	• profile – профиль пользователя • flag – флаг включить/выключить
Результат:	None

```
skydns_website.api.application.views.set_whitelist(request, profile, flag)
```

JSONRPC Method: setWhiteListOnly

Включить/выключить работу по белому списку.

Параметры:	• profile – профиль пользователя • flag – флаг включить/выключить
Результат:	None

```
skydns_website.api.application.views.system_info(request)
```

JSONRPC Method: systemInfo

Получение информации о настройках системы.

Результат:	системная информация в формате: <pre>{ 'public_dns': DNS, 'nxdomain': NX domain, 'blockpage': blockpage, 'blockpage_token': blockpage token, 'blockapi': block api }</pre>
Тип результата:	dict

```
skydns_website.api.application.views.test_auth(request, username, password)
```

JSONRPC Method: testAuth

Аутентификация пользователя.

Результат:	результат аутентификации (true или false)
Тип результата:	bool

```
skydns_website.api.application.views.user_filter(request, profile)
```

JSONRPC Method: userFilter

Получить список категорий профиля пользователя.

Результат:	список id категорий профиля
Тип результата:	list

```
skydns_website.api.application.views.user_info(request)
```

JSONRPC Method: userInfo

Получить информацию о пользователе.

Результат:	информация о пользователе в формате: <pre>{ 'username': юзернейм, 'plan': { 'name': название тарифного плана, 'code': код тарифного плана, 'date_end': дата конца оплаченного периода, 'features': { название_опции: значение, ... }, }, 'tz': смещение от UTC в часах, 'tz_minutes': смещение от UTC в минутах }</pre>
Тип результата:	dict