

# Почта для доменов

Справочник разработчика

9.12.2014

**Я**ндекс

Почта для доменов. Справочник разработчика. Версия

Дата сборки документа: 9.12.2014.

Этот документ является составной частью технической документации Яндекса.

Сайт справки к сервисам Яндекса: <http://help.yandex.ru>

© 2008—2014 ООО «ЯНДЕКС». Все права защищены.

## **Предупреждение об исключительных правах**

Яндексу (а также указанному им правообладателю) принадлежат исключительные права на все результаты интеллектуальной деятельности и приравненные к ним средства индивидуализации, используемые при разработке, поддержке и эксплуатации сервиса Почта для доменов. К таким результатам могут относиться, но не ограничиваясь указанными, программы для ЭВМ, базы данных, изображения, тексты, другие произведения, а также изобретения, полезные модели, товарные знаки, знаки обслуживания, коммерческие обозначения и фирменные наименования. Эти права охраняются в соответствии с Гражданским кодексом РФ и международным правом.

Вы можете использовать сервис Почта для доменов или его составные части только в рамках полномочий, предоставленных вам Пользовательским соглашением сервиса Почта для доменов или специального соглашения.

Нарушение требований по защите исключительных прав правообладателя влечет за собой дисциплинарную, гражданско-правовую, административную или уголовную ответственность в соответствии с российским законодательством.

## **Контактная информация**

ООО «ЯНДЕКС»

<http://www.yandex.ru>

Тел.: +7 495 739 7000

Email: [pr@yandex-team.ru](mailto:pr@yandex-team.ru)

Главный офис: 119021, Россия, г. Москва, ул. Льва Толстого, д. 16

# Содержание

Документация .....	5
<b>API Яндекс.Почты для доменов .....</b>	<b>6</b>
Получение токена .....	7
<b>Управление доменом .....</b>	<b>7</b>
reg_domain .....	8
reg_default_user .....	8
del_domain .....	9
add_logo .....	10
del_logo .....	11
get_domain_users .....	12
check_user .....	13
add_admin .....	14
del_admin .....	15
get_admins .....	15
<b>Аккаунты пользователей .....</b>	<b>17</b>
reg_user_token .....	17
reg_user_crypto .....	17
reg_user .....	18
get_mail_info .....	19
get_user_info .....	20
edit_user .....	21
set_forward .....	22
get_forward_list .....	22
delete_forward .....	23
delete_user .....	23
del_user .....	23
<b>Списки рассылки .....</b>	<b>25</b>
create_general_maillist .....	25
delete_general_maillist .....	26
<b>Авторизация пользователей .....</b>	<b>27</b>
user_oauth_token .....	27
passport .....	28
set_mail_callback .....	29
Применение методов авторизации .....	30
<b>Импорт почты .....</b>	<b>35</b>
set_domain .....	35
start_import .....	36
check_import .....	36
stop_import .....	36
reg_and_imp .....	37
import_imap .....	37
<b>Управление DKIM .....</b>	<b>39</b>
enable .....	39
status .....	40
disable .....	42
<b>API DNS .....</b>	<b>43</b>
get_token .....	43
add_a_record .....	44
add_aaaa_record .....	45
add_cname_record .....	45
add_mx_record .....	46
add_ns_record .....	47
add_srv_record .....	48
add_txt_record .....	49
get_domain_records .....	49

edit_a_record .....	50
edit_aaaa_record .....	51
edit_cname_record .....	52
edit_mx_record .....	53
edit_ns_record .....	54
edit_srv_record .....	55
edit_soa_record .....	56
edit_txt_record .....	57
delete_record .....	57
Спецификация запроса .....	58
<b>API для хостеров, регистраторов и парковщиков доменов .....</b>	<b>60</b>
Получение ID регистратора .....	61
Задание URL'ов обратного вызова .....	62
Запрос подключения к ПДД .....	63
Аутентификация домена .....	63
Подтверждение владения доменом .....	64
Подтверждение подключения домена .....	65
Импорт почты .....	66
Подтверждение импорта почты .....	67
Получение информации об отключении домена .....	68
Получение информации о текущем состоянии домена .....	68
Проверка состояния импорта почты .....	70

## Документация

В раздел включена техническая документация с описанием программного интерфейса (API), который предоставляет сервис Яндекс.Почта для доменов.

### API Яндекс.Почты для доменов

[Документ](#) содержит описание API-методов, предназначенных для управления доменом. Разработаны методы для управления доменом в целом и аккаунтами отдельных пользователей. Методы можно вызывать непосредственно в программном коде (например, коде вашего сайта). Для каждого метода приведено подробное описание: синтаксис, параметры, формат ответа.

Основные разделы документа:

- [Управление доменом](#) – методы, относящиеся ко всему домену: регистрация и удаление домена, операции с логотипами, управление списком администраторов.
- [Аккаунты пользователей](#) – управление почтовыми ящиками пользователей.
- [Списки рассылки](#) – управление общими для домена списками рассылки.
- [Авторизация пользователей](#) – управление авторизацией пользователей по короткоживущему OAuth токену.
- [Импорт почты](#) – управление переносом почтовых ящиков.

### API DNS

[Документ](#) является справочником API-методов Яндекс.Почты, предназначенных для управления DNS вашего домена.

Описаны методы для работы с записями форматов [A](#), [AAAA](#), [CNAME](#), [MX](#), [NS](#), [SRV](#), [TXT](#), [SOA](#).

### API для хостеров и регистраторов

Документ предназначен специально для хостинг-провайдеров, регистраторов и парковщиков доменов. Может быть использован как справочник API или как инструкция по подключению пользовательских доменов к Яндекс.Почте.

Приведенные в документе сведения помогут хостинг-провайдерам, регистраторам и парковщикам доменов подключать домены своих пользователей к Яндекс.Почте, импортировать их почту в Яндекс, получать информацию о состоянии подключения, импорта почты.

Подключение происходит автоматически (по одной кнопке) и реализовано с помощью API-методов, описанных в [данном документе](#).

## API Яндекс.Почты для доменов

Описаны методы API Яндекс.Почты для доменов.

API Яндекс.Почты для доменов позволяет производить административные действия над Вашими почтовыми ящиками из программного кода (например, из кода вашего сайта).

API представляет собой HTTP(S)-интерфейс вызовов. Каждый вызов выполняет какое-то действие. Параметры передаются в строке запроса или в теле запроса. Все параметры надо передавать в кодировке `urlencode`. Результат выдается в теле ответа.

Ниже в таблице приведен перечень методов и их назначение.

Метод	Назначение
<a href="#">get_token</a>	Получение авторизационного токена
<b>Управление доменом</b>	
<a href="#">reg_domain</a>	Подключение домена
<a href="#">reg_default_user</a>	Задание почтового ящика по умолчанию для домена
<a href="#">del_domain</a>	Отключение домена
<a href="#">add_logo</a>	Добавление логотипа домена
<a href="#">del_logo</a>	Удаление логотипа домена
<a href="#">get_domain_users</a>	Получение списка почтовых ящиков
<a href="#">check_user</a>	Проверка существования пользователя
<a href="#">add_admin</a>	Добавление дополнительного администратора для домена
<a href="#">del_admin</a>	Удаление дополнительного администратора для домена
<a href="#">get_admins</a>	Получение списка дополнительных администраторов домена
<b>Аккаунты пользователей</b>	
<a href="#">reg_user_token</a>	Регистрация пользователя
<a href="#">reg_user_crypto</a>	Создание пользователя с зашифрованным паролем
<a href="#">reg_user</a>	Создание почтового ящика в неосновном домене
<a href="#">get_mail_info</a>	Получение числа непрочитанных писем
<a href="#">get_user_info</a>	Получение данных пользователя
<a href="#">edit_user</a>	Изменение данных пользователя
<a href="#">set_forward</a>	Установка переадресации
<a href="#">get_forward_list</a>	Получение списка переадресаций и фильтров
<a href="#">delete_forward</a>	Удаление переадресации и фильтров
<a href="#">delete_user</a>	Удаление пользователя
<a href="#">del_user</a>	Удаление почтового ящика в неосновном домене
<b>Списки рассылки</b>	
<a href="#">create_general_maillist</a>	Создание общего списка рассылки
<a href="#">delete_general_maillist</a>	Удаление общего списка рассылки
<b>Авторизация пользователей</b>	
<a href="#">user_oauth_token</a>	Получение короткоживущего токена для авторизации
<a href="#">passport</a>	Авторизация по короткоживущему токenu
<a href="#">set_mail_callback</a>	Установка авторизационного URL-колбэка
<b>Импорт почты</b>	
<a href="#">set_domain</a>	Сохранение настроек импорта для домена
<a href="#">start_import</a>	Запуск импорта почтового ящика

Метод	Назначение
<a href="#">check_import</a>	Проверка состояния процесса импорта почты
<a href="#">stop_import</a>	Останов импорта почты
<a href="#">reg_and_imp</a>	Регистрация пользователя и запуск импорта почты
<a href="#">import_imap</a>	Запуск импорта папки по IMAP

## Получение токена

Метод `get_token` предназначен для получения авторизационного токена.

Авторизационный токен используется для активации API Яндекс.Почты для доменов. Получать токен нужно только один раз.

Чтобы получить токен, следует иметь подключенный домен, авторизоваться его администратором и перейти по указанному ниже адресу.

### Синтаксис запроса

```
https://pddimp.yandex.ru/get_token.xml ? domain_name=<имя домена>
```

domain\_name   Имя домена.

### Пример

Получение токена для домена `example.com`:

```
https://pddimp.yandex.ru/get_token.xml?domain_name=example.com
```

### Формат ответа

Если ошибок нет, метод возвращает `<ok token="..." />`, в противном случае — `<error reason='...' />`.

## Управление доменом

Описаны методы API Яндекс.Почты, предназначенные для управления доменом.

Метод	Назначение
<a href="#">reg_domain</a>	Подключение домена
<a href="#">reg_default_user</a>	Задание почтового ящика по умолчанию для домена
<a href="#">del_domain</a>	Отключение домена
<a href="#">add_logo</a>	Добавление логотипа домена
<a href="#">del_logo</a>	Удаление логотипа домена
<a href="#">get_domain_users</a>	Получение списка почтовых ящиков
<a href="#">check_user</a>	Проверка существования пользователя
<a href="#">add_admin</a>	Добавление дополнительного администратора для домена
<a href="#">del_admin</a>	Удаление дополнительного администратора для домена
<a href="#">get_admins</a>	Получение списка дополнительных администраторов домена

## reg\_domain

Метод позволяет подключить домен.

### Синтаксис запроса

```
https://pddimp.yandex.ru/api/reg_domain.xml ?
token=<токен>
& domain=<имя домена>
```

**token**            Токен, полученный в результате вызова метода `get_token`. Метод следует вызывать только после подключения хотя бы одного домена «вручную» и получения для него токена посредством метода `get_token`.

**domain**            Имя домена.

### Формат ответа

Метод возвращает xml-структуру вида:

```
<action>
  <status>
    <success/>
    <error>...</error>
  </status>
  <domains>
    <domain>
      <name>...</name>
      <secret_name>...</secret_name>
      <secret_value>...</secret_value>
    </domain>
  </domains>
</action>
```

**success**            Элемент присутствует в случае успешного вызова метода.

**error**              Сообщение об ошибках. Элемент включается в данную структуру, если в результате выполнения метода возникли ошибки.

**name\***                Имя домена.

**secret\_name\***        Секретное имя файла или поддомена. Используется для подтверждения домена через размещение файла или через CNAME, аналогично тому, как это делается при ручном подключении домена.

**secret\_value\***        Секретная строка (для секретного файла). Используется для подтверждения домена через размещение файла или через CNAME, аналогично тому, как это делается при ручном подключении домена.

Если домен уже подключен, то метод `reg_domain` не выполняет никаких действий, возвращая секретное имя и секретную строку.

## reg\_default\_user

Метод позволяет задать почтовый ящик по умолчанию для домена.

Ящик по умолчанию — это ящик, в который приходят все письма на домен, адресованные в несуществующие на этом домене ящики.



## Синтаксис запроса

```
https://pddimp.yandex.ru/api/reg_default_user.xml ?
token=<токен>
& domain=<домен>
& login=<имя ящика>
```

token	Токен, полученный в результате вызова метода <code>get_token</code> . Метод следует вызывать только после подключения хотя бы одного домена «вручную» и получения для него токена посредством метода <code>get_token</code> .
domain	Имя домена.
login	Имя почтового ящика. Ящик с именем <code>login</code> должен уже существовать.

## Формат ответа

Метод возвращает XML-структуру вида:

```
<action>
  <status>
    <success/>
    <error>...</error>
  </status>
  <domains>
    <domain>
      <name>...</name>
      <default-email>...</default-email>
    </domain>
  </domains>
</action>
```

success*	Элемент присутствует в случае успешного вызова метода.
error	Сообщение об ошибках. Элемент включается в данную структуру, если в результате выполнения метода возникли ошибки.
name*	Имя домена.
default-email*	Имя почтового ящика по умолчанию (логин пользователя).

## del\_domain

Метод позволяет отключить домен.

Отключенный домен перестает выводиться в списке доменов. После отключения домен можно подключить заново.

### Примечание:

Отключение домена не влечет за собой изменения MX-записей. MX-записи нужно устанавливать отдельно на DNS-серверах, куда делегирован домен.

## Синтаксис запроса

```
https://pddimp.yandex.ru/api/del_domain.xml ?
token=<токен>
& domain=<имя домена>
```

token	Токен, полученный в результате вызова метода <code>get_token</code> . Метод следует вызывать только после подключения хотя бы одного домена «вручную» и получения для него токена посредством метода <code>get_token</code> .
domain	Имя домена.

## Формат ответа

Метод возвращает XML-структуру вида:

```
<action>
  <status>
    <success/>
    <error>...</error>
  </status>
  <domains>
    <domain>
      <name>...</name>
    </domain>
  </domains>
</action>
```

success	Элемент присутствует в случае успешного вызова метода.
error	Сообщение об ошибках. Элемент включается в данную структуру, если в результате выполнения метода возникли ошибки.
name*	Имя домена.

## add\_logo

Метод позволяет добавить логотип домену.

Метод вызывается только как POST-запрос. Файл, содержащий логотип, и параметры передаются как `multipart/form-data`. Поддерживаются графические файлы форматов `jpg`, `gif`, `png` размером до 2 Мбайт. Имя файла и название параметра не важны.

## Синтаксис запроса

```
http://pddimp.yandex.ru/api/add_logo.xml
```

token	Токен, полученный в результате вызова метода <code>get_token</code> . Метод следует вызывать только после подключения хотя бы одного домена «вручную» и получения для него токена посредством метода <code>get_token</code> .
domain	Имя домена.

## Формат ответа

Метод возвращает XML-структуру вида:

```

<action>
  <status>
    <success/>
    <error>...</error>
  </status>
  <domains>
    <domain>
      <name>...</name>
    </domain>
    <logo>
      <url></url>
    </logo>
  </domains>
</action>

```

success	Элемент присутствует в случае успешного вызова метода.
error	Сообщение об ошибках. Элемент включается в данную структуру, если в результате выполнения метода возникли ошибки.
name*	Имя домена.
url*	URL, по которому расположен логотип домена.

### Пример

Ниже приведен пример cURL-запроса, который может использоваться, например, в интерпретаторе BASH.

```
curl -F "file=@logo.jpg" -F "domain=example.com" -F "token=a1b2c3" 'https://pddimp.yandex.ru/api/add_logo.xml'
```

Запрос позволяет добавить логотип, сохраненный в файле logo.jpg, для домена example.com. В запросе передается также авторизационный токен a1b2c3 данного домена.

## del\_logo

Метод позволяет удалить логотип домена.

### Синтаксис запроса

```
https://pddimp.yandex.ru/api/del_logo.xml ?
token=<токен>
& domain=<имя домена>
```

token	Токен, полученный в результате вызова метода get_token. Метод следует вызывать только после подключения хотя бы одного домена «вручную» и получения для него токена посредством метода get_token.
domain	Имя домена.

### Формат ответа

Метод возвращает XML-структуру вида:

```

<action>
  <status>
    <success/>
    <error>...</error>
  </status>
  <domains>
    <domain>
      <name>...</name>
    </domain>
  </domains>
</action>

```

success	Элемент присутствует в случае успешного вызова метода.
error	Сообщение об ошибках. Элемент включается в данную структуру, если в результате выполнения метода возникли ошибки.
name*	Имя домена.

## get\_domain\_users

Метод позволяет получить список почтовых ящиков.

Метод возвращает список ящиков в домене, привязанном к токену. Список выводится постранично, начиная со страницы с заданным порядковым номером. Нумерация страниц начинается с 1.

### Синтаксис запроса

```

https://pddimp.yandex.ru/get_domain_users.xml ?
token=<токен>
& on_page=<число записей на странице>
& page=<номер страницы>

```

token	Авторизационный токен.
on_page	Число записей о почтовых ящиках на одной странице. Значение параметра не может быть больше 100, по умолчанию число записей равно 100.
page	Номер страницы, с которой начинается вывод списка почтовых ящиков. По умолчанию значение равно 1.

### Формат ответа

Метод возвращает XML-структуру вида:

```

<page>
  <domains>
    <domain>
      <emails>
        <action-status/>
        <email><email-name>...</email-name></email>
        <email><email-name>...</email-name></email>
        ...
        <found>10</found>
        <total>100</total>
      </emails>
      <name>...</name>
      <status>added | mx-activate | domain-activate</status>
      <emails-max-count>100</emails-max-count>
    </domain>
  </domains>
</page>

```

<code>action-status*</code>	Сообщения об ошибках.
<code>name*</code>	Имя домена.
<code>email-name*</code>	Имя почтового ящика.
<code>found*</code>	Число ящиков, отображаемых на одной странице.
<code>status*</code>	Статус домена (added, mx-activate, domain-activate).
<code>emails-max-count*</code>	Максимальное число почтовых ящиков, разрешенное для этого домена.

## check\_user

Метод позволяет проверить существование пользователя.

### Синтаксис запроса

```

https://pddimp.yandex.ru/check_user.xml ?
token=<токен>
& login=<логин пользователя>

```

<code>token</code>	Авторизационный токен.
<code>login</code>	Логин пользователя для доступа к почтовому ящику.

### Формат ответа

Метод возвращает XML-ответ:

- `<result>exists</result>` — пользователь есть;
- `<result>no_user</result>` — пользователь не найден.

## add\_admin

Метод позволяет добавить дополнительного администратора домена.

Метод добавляет дополнительного администратора домена. Дополнительный администратор может авторизовываться на странице `pdd.yandex.ru`, видеть у себя в списке доменов все добавленные ему домены, и совершать все действия с ящиками на этом домене.

### Внимание!

Не ошибитесь в логине дополнительного администратора, иначе вы предоставите доступ постороннему человеку к своим почтовым ящикам. После вызова метода обязательно разлогиньтесь, авторизуйтесь под дополнительным администратором и проверьте, что у него на странице `http://pdd.yandex.ru` появился домен.

## Синтаксис запроса

```
https://pddimp.yandex.ru/api/multiadmin/add_admin.xml ?
token=<токен>
& domain=<имя домена>
& login=<логин администратора>
```

token	Токен, полученный в результате вызова метода <code>get_token</code> . Метод следует вызывать только после подключения хотя бы одного домена «вручную» и получения для него токена посредством метода <code>get_token</code> .
domain	Имя домена, для которого добавляется дополнительный администратор.
login	Яндекс-логин дополнительного администратора.

## Формат ответа

Метод возвращает XML-структуру вида:

```
<action>
  <domain>
    <status>
      <success/>
      <error>...</error>
    </status>
    <name>...</name>
    <new-admin> ... </new-admin>
  </domain>
</action>
```

success	Элемент присутствует в случае успешного вызова метода.
error	Сообщение об ошибках. Элемент включается в данную структуру, если в результате выполнения метода возникли ошибки.
name*	Имя домена.
new-admin*	Имя дополнительного администратора домена.

## del\_admin

Метод позволяет удалить дополнительного администратора для домена.

### Синтаксис запроса

```
https://pddimp.yandex.ru/api/multiadmin/del_admin.xml ?
token=<токен>
& domain=<имя домена>
& login=<имя почтового ящика>
```

token	Токен, полученный в результате вызова метода <code>get_token</code> . Метод следует вызывать только после подключения хотя бы одного домена «вручную» и получения для него токена посредством метода <code>get_token</code> .
domain	Имя домена, для которого удаляется дополнительный администратор.
login	Яндекс-логин дополнительного администратора.

### Формат ответа

Метод возвращает XML-структуру вида:

```
<action>
  <domain>
    <status>
      <success/>
      <error>...</error>
    </status>
    <name>...</name>
    <new-admin> ... </new-admin>
  </domain>
</action>
```

success	Элемент присутствует в случае успешного вызова метода.
error	Сообщение об ошибках. Элемент включается в данную структуру, если в результате выполнения метода возникли ошибки.
name*	Имя домена.
new-admin*	Имя дополнительного администратора домена.

## get\_admins

Метод позволяет получить список дополнительных администраторов домена.

### Синтаксис запроса

```
https://pddimp.yandex.ru/api/multiadmin/get_admins.xml ?
token=<токен>
& domain=<имя домена>
```

token	Токен, полученный в результате вызова метода <code>get_token</code> . Метод следует вызывать только после подключения хотя бы одного домена «вручную» и получения для него токена посредством метода <code>get_token</code> .
domain	Имя домена, для которого следует получить список дополнительных администраторов.

## Формат ответа

Метод возвращает XML-структуру вида:

```
<action>
  <domain>
    <status>
      <success/>
      <error>...</error>
    </status>
    <name>...</name>
    <other-admins>
      <login>...</login>
    </other-admins>
  </domain>
</action>
```

success	Элемент присутствует в случае успешного вызова метода.
error	Сообщение об ошибках. Элемент включается в данную структуру, если в результате выполнения метода возникли ошибки.
name*	Имя домена.
login*	Имя дополнительного администратора домена.



## Аккаунты пользователей

Описаны методы API Яндекс.Почты для доменов, позволяющие выполнять действия над аккаунтами пользователей.

Метод	Назначение
<a href="#">reg_user_token</a>	Регистрация пользователя
<a href="#">reg_user_crypto</a>	Создание пользователя с зашифрованным паролем
<a href="#">reg_user</a>	Создание почтового ящика в неосновном домене
<a href="#">get_mail_info</a>	Получение числа непрочитанных писем
<a href="#">get_user_info</a>	Получение данных пользователя
<a href="#">edit_user</a>	Изменение данных пользователя
<a href="#">set_forward</a>	Установка переадресации
<a href="#">get_forward_list</a>	Получение списка переадресаций и фильтров
<a href="#">delete_forward</a>	Удаление переадресации и фильтров
<a href="#">delete_user</a>	Удаление пользователя
<a href="#">del_user</a>	Удаление почтового ящика в неосновном домене

### reg\_user\_token

Метод предназначен для регистрации пользователя.

#### Синтаксис запроса

```
https://pddimp.yandex.ru/reg_user_token.xml ?
token=<токен>
& u_login=<логин пользователя>
& u_password=<пароль пользователя>
```

token            Авторизационный токен.

u\_login          Логин пользователя для доступа к почтовому ящику.

u\_password      Пароль пользователя для доступа к почтовому ящику.

#### Формат ответа

Если ошибок нет, метод возвращает `<ok uid="..."/>`, в противном случае — `<error reason='...' />`.

### reg\_user\_crypto

Метод позволяет создать пользователя с зашифрованным паролем.

#### Примечание:

Шифрование не влияет на ввод пароля к почтовому ящику. Зашифрованный пароль вводится точно так же, как и до его шифрования.

## Синтаксис запроса

```
https://pddimp.yandex.ru/reg_user_crypto.xml ?
token=<токен>
& login=<имя почтового ящика>
& password=<пароль>
```

token	Авторизационный токен.
login	Имя почтового ящика.
password	Пароль к ящику в зашифрованном виде, т.е. MD5-CRYPT хеш пароля в виде строки следующего формата: "\$1\$" + 8 символов [a-zA-Z0-9./] + "\$" + 22 символа [a-zA-Z0-9./]

## reg\_user

Метод позволяет создать почтовый ящик на неосновном домене.

Метод работает аналогично методу [reg\\_user\\_token](#), за исключением того, что пользователь создается не на текущем домене (т.е. на домене, к которому привязан токен), а на любом домене, подключенном к вашему аккаунту.

## Синтаксис запроса

```
https://pddimp.yandex.ru/api/reg_user.xml ?
token=<токен>
& domain=<имя домена>
& login=<имя почтового ящика>
& passwd=<пароль ящика>
& cryptopasswd=<зашифрованный пароль>
```

token	Токен, полученный в результате вызова метода <code>get_token</code> . Метод следует вызывать только после подключения хотя бы одного домена «вручную» и получения для него токена посредством метода <code>get_token</code> .
domain	Имя домена.
login	Имя почтового ящика.
passwd	Пароль пользователя.
cryptopasswd	Зашифрованный по MD5 (аналогично методу <a href="#">reg_user_crypto</a> ) пароль для доступа к почтовому ящику. Если задан этот параметр, то параметр <code>passwd</code> игнорируется.

## Формат ответа

Метод возвращает XML-структуру вида:

```

<action>
  <status>
    <success/>
    <error>...</error>
  </status>
  <domains>
    <domain>
      <name>...</name>

      <email>
        <name_email>...</name_email>
      </email>

    </domain>
  </domains>
</action>

```

success	Элемент присутствует в случае успешного вызова метода.
error	Сообщение об ошибках. Элемент включается в данную структуру, если в результате выполнения метода возникли ошибки.
name*	Имя домена.
name_email*	Имя почтового ящика (login).

## get\_mail\_info

Метод позволяет получить количество непрочитанных писем.

### Синтаксис запроса

```

https://pddimp.yandex.ru/get_mail_info.xml ?
token=<токен>
& login=<имя ящика>

```

token	Авторизационный токен.
login	Имя почтового ящика.

### Формат ответа

- Ответ на успешно обработанный запрос (85 — количество непрочитанных писем):

```
<ok new_messages="85"/>
```

- Логин, указанный в значении параметра login, не зарегистрирован:

```
<error reason="no_user"/>
```

- Токен, указанный в значении параметра token, невалиден:

```
<error reason="no_auth"/>
```

## get\_user\_info

Метод позволяет получить данные пользователя.

### Синтаксис запроса

```
https://pddimp.yandex.ru/get_user_info.xml ?
token=<токен>
& login=<логин пользователя>
```

token            Авторизационный токен.

login            Логин пользователя, для которого необходимо получить данные.

### Формат ответа

Метод возвращает XML-структуру вида:

```
<page>
  <domain>
    <name></name>
    <user>
      <login>ivan</login>
      <birth_date>1900-01-01</birth_date>
      <fname>Ivanov</fname>
      <iname>Ivan</iname>
      <hinta></hinta>
      <hintq>Любимое блюдо</hintq>
      <mail_format>...</mail_format>
      <charset></charset>
      <nickname>user</nickname>
      <sex>1</sex>
      <enabled>1</enabled>
      <signed_eula>1</signed_eula>
    </user>
  </domain>
</page>
```

login\*            Логин пользователя.

birth\_date\*      Дата рождения в формате YYYY-MM-DD.

fname\*            Фамилия пользователя.

iname\*            Имя пользователя.

hinta\*            Ответ на секретный вопрос.

hintq\*            Секретный вопрос.

mail\_format\*     Формат почты, предпочтительный при создании письма.

charset\*          Кодировка.

nickname\*        Псевдоним пользователя.

sex\*              Пол пользователя:

0 – не указан;

1 – мужской;

2 – женский.

enabled\*          Состояние почтового ящика:

1 – включен и почта принимается;

0 – заблокирован, почта не принимается.

signed\_eula\* Признак того, что пользователь принял условия публичной оферты.

1 – да;

0 – нет.

## edit\_user

Метод предназначен для редактирования данных пользователя.

Метод позволяет изменить персональные данные пользователя: имя, фамилию, пол, пароль, секретный вопрос и ответ на секретный вопрос.

### Синтаксис запроса

```
https://pddimp.yandex.ru/edit_user.xml ?
token=<токен>
& login=<логин пользователя>
& [password=<пароль пользователя>]
& [domain_name=<имя домена>]
& [iname=<имя пользователя>]
& [fname=<фамилия пользователя>]
& [sex=<пол пользователя>]
& [hintq=<секретный вопрос>]
& [hinta=<ответ на секретный вопрос>]
```

token	Авторизационный токен.
login	Логин пользователя для доступа к почтовому ящику.
password	Пароль пользователя.
domain_name	Почтовый домен пользователя.
iname	Имя пользователя.
fname	Фамилия пользователя.
sex	Пол пользователя: 0 – не указан; 1 – мужской; 2 – женский.
hintq	Секретный вопрос.
hinta	Ответ на секретный вопрос.

### Формат ответа

Если ошибок нет, метод возвращает `<ok uid="..." />`, в противном случае — `<error reason='...' />`.

## set\_forward

Метод позволяет установить переадресацию для заданного пользователя.

### Синтаксис запроса

```
https://pddimp.yandex.ru/set_forward.xml ?
token=<токен>
& login=<логин пользователя>
& address=<е-mail для пересылки>
& copy=<признак сохранения исходников>
```

token	Авторизационный токен.
login	Логин пользователя для доступа к почтовому ящику.
address	e-mail, на который будет пересылаться почта с ящика login.
copy	Параметр определяет, сохранять ли после пересылки копию письма в ящике login: «yes» — сохранять; «no» — удалять.

### Формат ответа

Если ошибок нет, метод возвращает <ok/>, в противном случае — <error reason='...' />.

## get\_forward\_list

Метод позволяет получить список переадресаций и фильтров.

### Синтаксис запроса

```
https://pddimp.yandex.ru/get_forward_list.xml ?
token=<токен>
& login=<логин пользователя>
```

token	Авторизационный токен.
login	Логин пользователя для доступа к почтовому ящику.

### Формат ответа

Если ошибок нет, метод возвращает <ok><filters><filter>...</filter>...</filters></ok>, в противном случае — <error reason='...' />.

## delete\_forward

Метод позволяет удалить переадресацию или фильтр.

### Синтаксис запроса

```
https://pddimp.yandex.ru/delete_forward.xml ?
token=<токен>
& login=<логин пользователя>
& filter_id=<id фильтра>
```

token	Авторизационный токен.
login	Логин пользователя для доступа к почтовому ящику.
filter_id	ID фильтра, который надо удалить.

## delete\_user

Метод предназначен для удаления пользователя.

### Синтаксис запроса

```
https://pddimp.yandex.ru/delete_user.xml ?
token=<токен>
& login=<логин пользователя>
```

token	Авторизационный токен.
login	Логин пользователя для доступа к почтовому ящику.

## del\_user

Метод позволяет удалить почтовый ящик в неосновном домене.

### Синтаксис запроса

```
https://pddimp.yandex.ru/api/del_user.xml ?
token=<токен>
& domain=<имя домена>
& login=<имя почтового ящика>
```

token	Токен, полученный в результате вызова метода <code>get_token</code> . Метод следует вызывать только после подключения хотя бы одного домена «вручную» и получения для него токена посредством метода <code>get_token</code> .
domain	Имя домена.
login	Имя почтового ящика.

### Формат ответа

Метод возвращает XML-структуру вида:

```
<action>
  <status>
    <success/>
    <error>...</error>
  </status>
  <domains>
    <domain>
      <name>...</name>

      <email>
        <name_email>...</name_email>
      </email>

    </domain>
  </domains>
</action>
```

success	Элемент присутствует в случае успешного вызова метода.
error	Сообщение об ошибках. Элемент включается в данную структуру, если в результате выполнения метода возникли ошибки.
name*	Имя домена
name_email*	Имя почтового ящика (login).



## Списки рассылки

Описаны методы API Яндекс.Почты для доменов, позволяющие создавать и удалять общие для домена списки рассылки.

Метод	Назначение
<a href="#">create_general_maillist</a>	Создание общего списка рассылки
<a href="#">delete_general_maillist</a>	Удаление общего списка рассылки

## create\_general\_maillist

Метод позволяет создать общий список рассылки домена, включающий всех пользователей домена.

### Синтаксис запроса

```
https://pddimp.yandex.ru/api/create_general_maillist.xml ?
token=<токен>
& domain=<имя домена>
& ml_name=<имя рассылки>
```

token	Токен, полученный в результате вызова метода <code>get_token</code> . Метод следует вызывать только после подключения хотя бы одного домена «вручную» и получения для него токена посредством метода <code>get_token</code> .
domain	Имя домена.
ml_name	Имя общего списка рассылки.

### Формат ответа

Метод возвращает XML-структуру вида:

```
<action>
  <domain>
    <status>
      <success/>
      <error>...</error>
    </status>
    <name>...</name>
    <users>
      <login>...</login>
    </users>
  </domain>
</action>
```

success	Элемент присутствует в случае успешного вызова метода.
error	Сообщение об ошибках. Элемент включается в данную структуру, если в результате выполнения метода возникли ошибки.
name*	Имя домена.
login*	Имя пользователя домена.

## delete\_general\_maillist

Метод позволяет удалить общий список рассылки домена, включающий всех пользователей домена.

### Синтаксис запроса

```
https://pddimp.yandex.ru/api/delete_general_maillist.xml ?
token=<токен>
& domain=<имя домена>
```

**token**            Токен, полученный в результате вызова метода `get_token`. Метод следует вызывать только после подключения хотя бы одного домена «вручную» и получения для него токена посредством метода `get_token`.

**domain**            Имя домена.

### Формат ответа

Метод возвращает XML-структуру вида:

```
<action>
  <domain>
    <status>
      <success/>
      <error>...</error>
    </status>
    <name>...</name>
    <users>
      <login>...</login>
    </users>
  </domain>
</action>
```

**success**            Элемент присутствует в случае успешного вызова метода.

**error**              Сообщение об ошибках. Элемент включается в данную структуру, если в результате выполнения метода возникли ошибки.

**name\***                Имя домена.

**login\***                Имя пользователя домена.

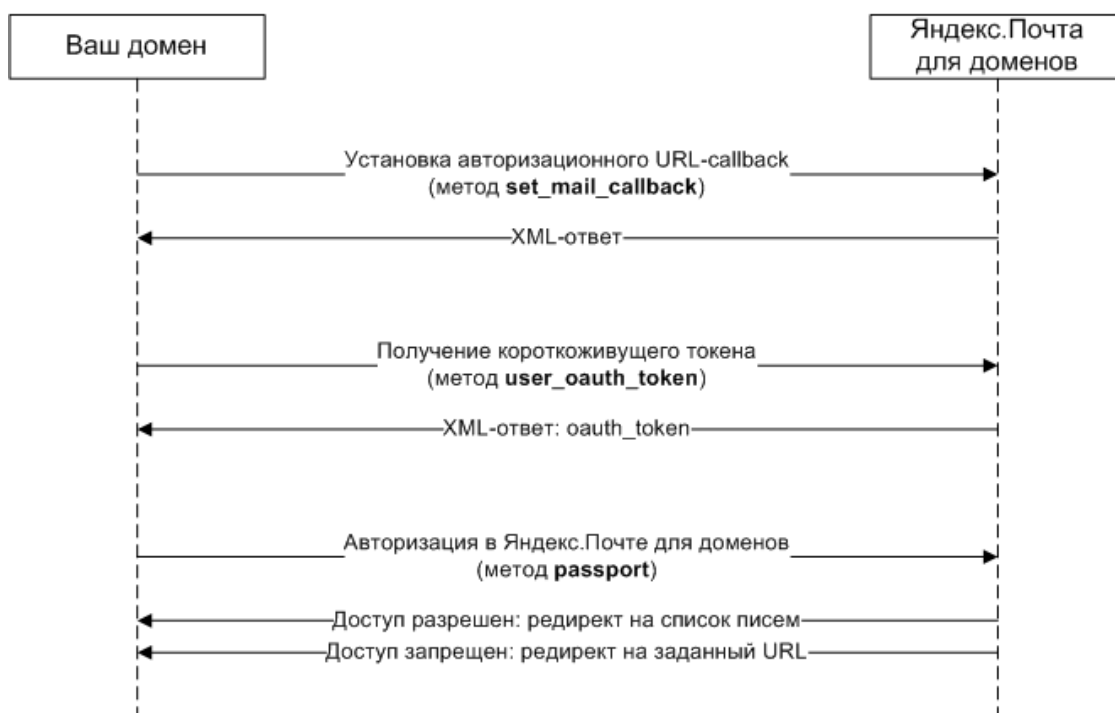
## Авторизация пользователей

Описаны методы API Яндекс.Почты для доменов, предназначенные для настройки авторизации по короткоживущему OAuth токену.

Метод	Назначение
<a href="#">user_oauth_token</a>	Получение короткоживущего токена для авторизации
<a href="#">passport</a>	Авторизация по короткоживущему токену
<a href="#">set_mail_callback</a>	Установка авторизационного URL-колбэка

Методы реализуют следующий сценарий авторизации в Яндекс.Почте для доменов (см. рисунок).

1. Установка авторизационного URL-callback. Выполняется методом [set\\_mail\\_callback](#). Метод устанавливает авторизационный URL-callback, на который выполняется редирект в случае попытки доступа к почте неавторизованного пользователя.
2. Получение короткоживущего токена. Собственно процедура авторизации выполняется по так называемому короткоживущему токену, валидному в течение 30 с. Короткоживущий токен возвращается в ответе на вызов метода [user\\_oauth\\_token](#).
3. Авторизация. Выполняется методом [passport](#). В случае подтверждения прав доступа Яндекс.Почта для доменов выполняет редирект браузера пользователя на список писем, иначе — на URL, определенный в параметрах данного метода.



Ниже подробно описано назначение и синтаксис каждого метода, а также их [применение](#).

### user\_oauth\_token

Метод позволяет получить короткоживущий токен для авторизации.

Метод возвращает короткоживущий токен, который позволяет авторизоваться в веб-интерфейсе Почты. Саму авторизацию по токену выполняет метод [passport](#). Время жизни короткоживущего токена – 30 секунд.

## Синтаксис запроса

```
https://pddimp.yandex.ru/api/user_oauth_token.xml ?
token=<токен>
& domain=<имя домена>
& login=<имя пользователя>
```

token	Токен, полученный в результате вызова метода <code>get_token</code> .
domain	Имя домена.
login	Логин пользователя для доступа к почтовому ящику.

## Формат ответа

Метод возвращает XML-структуру вида:

```
<action>
  <status>
    <success/>
    <error>...</error>
  </status>
  <domains>
    <domain>
      <name>...</name>

      <email>
        <email_name>...</email_name>
        <oauth-token>...</oauth-token>
      </email>

    </domain>
  </domains>
</action>
```

success	Элемент присутствует в случае успешного вызова метода.
error	Сообщение об ошибках. Элемент включается в данную структуру, если в результате выполнения метода возникли ошибки.
name*	Имя домена.
email_name*	Имя почтового ящика (включая доменную часть), для которого создан короткоживущий токен.
oauth-token*	Короткоживущий токен.

## passport

Метод позволяет авторизоваться по короткоживущему токenu.

Метод проверяет короткоживущий токен на корректность. Если токен корректный, метод авторизует пользователя в веб-интерфейсе Почты для доменов и выполняет редирект на список писем. В случае неудачной авторизации выполняется редирект на заданный URL.

## Синтаксис запроса

```
https://passport.yandex.ru/passport ? mode=oauth & type=trusted-pdd-partner
& error_retpath=<URL для редиректа>
& access_token=<токен>
https://passport.yandex.ru/passport ? mode=oauth & type=trusted-pdd-partner
& error_retpath=<URL для редиректа>
& access_token=<токен>
```

error\_retpath

URL, на который производится редирект при неудачной авторизации. URL должен быть корректно закодирован в кодировке urlencode.

access\_token Токен, полученный в результате вызова метода [user\\_oauth\\_token](#).

n

## set\_mail\_callback

Метод позволяет установить авторизационный URL-колбэк.

## Синтаксис запроса

```
https://pddimp.yandex.ru/api/set_mail_callback.xml ?
token=<токен>
& domain=<имя домена>
& callback=<URL-колбэк>
```

token Токен, полученный в результате вызова метода [get\\_token](#).

domain Домен пользователя, для которого устанавливается колбэк.

callback Авторизационный URL-колбэк. Должен быть правильно закодирован в кодировке urlencode.

## Формат ответа

Метод возвращает XML-структуру вида:

```
<action>
  <status>
    <success/>
    <error>...</error>
  </status>
  <domains>
    <domain>
      <name>...</name>

      <email>
        <email_name>...</email_name>
        <oauth-token>...</oauth-token>
      </email>

    </domain>
  </domains>
</action>
```

success Элемент присутствует в случае успешного вызова метода.

error Сообщение об ошибках. Элемент включается в данную структуру, если в результате выполнения метода возникли ошибки.

name\* Имя домена.

email\_name\* Имя почтового ящика (включая доменную часть), для которого создан короткоживущий токен.

`oauth-token*` Короткоживущий токен.

## Применение методов авторизации

В разделе описано применение методов авторизации на примере быстрого доступа к почте непосредственно с вашего сайта.

Вы можете разместить на вашем сайте ссылку (например, «Почта»), воспользовавшись которой авторизованный на сайте пользователь переходит в свой почтовый ящик сразу, без дополнительного ввода логина и пароля. При этом авторизация в Яндекс.Почте для доменов выполняется автоматически, с помощью логина и пароля, введенного при авторизации на сайте.

Для реализации быстрого доступа к почте по ссылке на вашем сайте необходимо:

1. [Создать почтовые ящики](#), если они не были созданы ранее.
2. [Получить авторизационный токен](#).
3. Разместить на вашем сайте [ссылку для быстрого доступа к почтовому ящику](#).
4. Разработать [страницу авторизации](#).
5. Разместить страницу авторизации на [авторизационном URL](#).
6. [Привязать авторизационный URL к вашему домену](#).
7. Разработать [модуль авторизации](#).
8. [Разместить модуль авторизации на сайте](#).
9. [Проверить работу ссылки на тестовых пользователях](#).

### Рекомендации по обеспечению безопасности данных

Вы полностью ответственны за безопасность авторизации на вашем сайте. Если злоумышленник сможет авторизоваться на вашем сайте, то он получит и доступ к почте пользователя по размещенной вами ссылке.

Вы полностью ответственны за то, что передаете в Яндекс.Почту для доменов логин авторизованного пользователя. Если вы передаете логин неавторизованного пользователя без каких-либо других авторизующих сущностей (например, пароль или id сессии в куке), то злоумышленнику будет очень просто получить доступ к почте этого пользователя.

Для сохранения персональных данных ваших пользователей настоятельно рекомендуем:

- использовать безопасные методы авторизации, а именно:
  - передавать логин, пароль и другие авторизационные сущности по HTTPS;
  - формировать длинные и трудно подбираемые авторизационные куки;
  - периодически (например, раз в несколько часов) обновлять авторизационную куку;
- не использовать методы так называемой «удобной» авторизации, которые облегчают доступ к почте не только пользователям, но и злоумышленникам:
  - «вечная» кука и «вечная» сессия. Пользователь, не завершивший авторизованную сессию на вашем сайте, остается авторизованным на компьютере постоянно, сколько бы ни заходил на ваш сайт.
  - автологинящие ссылки – ссылки, которые позволяют авторизоваться в почтовой системе автоматически.

Авторизованные сессии, длящиеся длительное время, и автологинящие ссылки увеличивают риск утечки персональных данных. Поэтому если на вашем сайте предусмотрены методы «удобной» авторизации, не рекомендуем предоставлять доступ пользователям с «вечной» кукой или воспользовавшимся авто-

логинящей ссылкой. Например, в случае использования автологинящей ссылки можно проверять авторизационную куку и, если она не установлена, предлагать пользователю ввести пароль.

Обращаем внимание, что в текущей реализации Яндекс.Почты для доменов завершение сеансов авторизованной работы на сайте и в почтовом ящике не синхронизируются, то есть происходят независимо. Это означает, что пользователь при выходе с вашего сайта (разлогинивании) может остаться авторизованным в почтовом ящике и наоборот.

## Создание почтовых ящиков

Почтовые ящики в Яндекс.Почте для доменов не создаются автоматически при переходе по ссылке, которую вы разместите на сайте. Поэтому первым шагом является проверка наличия и, при необходимости, создание почтовых ящиков для всех пользователей вашего сайта. Для регистрации почтовых ящиков используются API-методы, описанные в разделе [Аккаунты пользователей](#):

- [reg\\_user\\_token](#);
- [reg\\_user\\_crypto](#);
- [reg\\_user](#).

При регистрации почтовых ящиков рекомендуем указывать реальные пароли, так как пользователи могут переходить в почту не только по ссылке с вашего сайта, но и через почтовые клиенты (в том числе через мобильные). Пароли к существующим почтовым ящикам рекомендуем синхронизировать с текущими паролями пользователей вашего сайта посредством API.

В дальнейшем следует отслеживать регистрацию новых пользователей вашего сайта, создавая для них почтовые ящики в Яндекс.Почте для доменов. Соответственно, при удалении пользователем своей учетной записи на сайте необходимо удалить его почтовый ящик (методы [delete\\_user](#); [del\\_user](#)).

## Ссылка быстрого доступа к почтовому ящику

Нужно разместить на вашем сайте ссылку вида

```
<a href="http://mail.yandex.ru/for/domain.ru?loginfrompartner=user@domain.ru">Почта</a>
```

где `user@domain.ru` надо заменить на e-mail авторизованного на вашем сайте пользователя.

Клик на эту ссылку приведет к проверке авторизации пользователя, которая выполняется в Яндекс.Почте для доменов автоматически. В случае подтверждения прав доступа пользователь будет перенаправлен в почтовый ящик.

## Страница авторизации

Вам необходимо разработать страницу авторизации пользователя в Яндекс.Почте для доменов. На странице должны быть предусмотрены поля для ввода логина и пароля доступа к почтовому ящику. Кроме того, к странице нужно [привязать модуль авторизации](#) пользователя в Яндекс.Почте для доменов.

Страница авторизации размещается на вашем сайте, ее URL будем называть авторизационным URL для доступа ваших пользователей к почтовым ящикам.

## Привязка авторизационного URL к домену

Доступ к почте может пытаться получить как авторизованный, так и неавторизованный пользователь. Если пользователь не авторизован на сайте, при попытке войти в Яндекс.Почту для доменов по ссылке «Почта» он будет перенаправлен на страницу авторизации Яндекс.Почты для доменов. Чтобы этого не произошло, следует привязать авторизационный URL к домену с помощью метода [set\\_mail\\_callback](#). В этом случае при попытке входа в почту неавторизованного на сайте пользователя будет выполняться редирект на указанную вами [страницу авторизации](#). URL страницы передается в параметре `callback` метода [set\\_mail\\_callback](#).

## Модуль авторизации

Автоматическую авторизацию в Яндекс.Почте для доменов реализует специальный модуль. Модуль размещается на вашем ресурсе и выполняет все операции, необходимые для подтверждения прав доступа к почте.

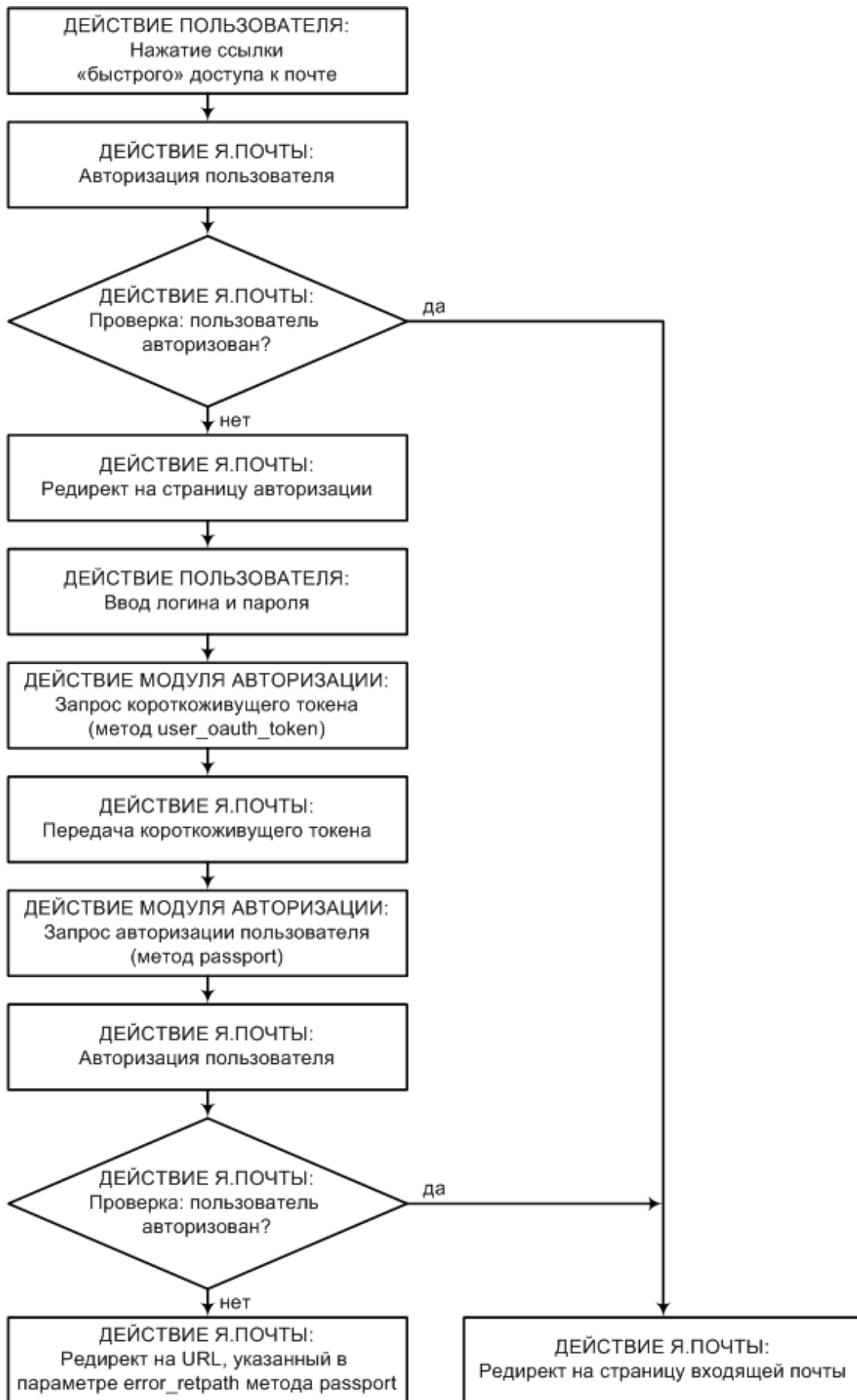
Модуль авторизации разрабатывается индивидуально для каждого сайта и может быть реализован по-разному (например, в виде скрипта). Его разработка требует навыков программирования на том языке, на котором реализован ваш сайт (например, PHP или Perl). Если вы не обладаете указанными навыками, рекомендуем обратиться к специалисту с соответствующим опытом (например, разработчику вашего сайта).

Модуль авторизации должен выполнять следующие действия:

- проверять факт авторизации пользователя на сайте (например, по куке);
- определять логин авторизованного пользователя;
- получать короткоживущий токен с помощью метода [user\\_oauth\\_token](#);
- авторизовать пользователя в Яндекс.Почте для доменов с помощью метода [passport](#).



Данный модуль реализует следующий сценарий авторизации в Яндекс.Почте для доменов (см. рисунок).



При нажатии пользователем ссылки быстрого доступа к почте выполняется попытка входа в почтовый ящик. Логин и имя домена передаются с вашего сайта в Яндекс.Почту для доменов, которая пытается авторизовать пользователя с данным логином. В случае подтверждения прав доступа браузер пользователя перенаправляется на страницу со списком писем.

Если авторизовать пользователя не удалось, Яндекс.Почта для доменов выполняет редирект браузера на [страницу авторизации](#). На данной странице пользователь указывает свой логин и пароль, после чего ваш ресурс запускает модуль авторизации.

Модуль авторизации вызывает метод `user_oauth_token`, передавая ему в качестве параметра логин, являющийся именем почтового ящика, и имя домена. В ответе на метод `user_oauth_token` возвращается короткоживущий токен, по которому будет выполнена авторизация. Длительность существования короткоживущего токена составляет 30 с, поэтому все дальнейшие операции должны быть выполнены до истечения этого времени.

---

#### Примечание:

Не путайте авторизационный токен с короткоживущим токеном. Время жизни первого токена не ограничено, и он используется для доступа ко всем методам API. Второй токен валиден в течение 30 секунд, и он необходим только для передачи Яндекс.Почте для доменов факта авторизации конкретного пользователя на вашем сайте.

---

Модуль авторизации вызывает метод `passport`, передавая ему в качестве параметра короткоживущий токен. Если токен валиден, Яндекс.Почта для доменов выполняет редирект браузера на страницу со списком писем. Иная ситуация означает, что произошла ошибка, и пользователь не может быть авторизован. В этом случае Яндекс.Почта выполняет редирект браузера на URL, указанный в параметре `error_retpath` метода `passport`.

## Размещение модуля авторизации на сайте

Разработанный [модуль авторизации](#) необходимо разместить на вашем сайте, привязав к авторизационному URL. Привязка должна обеспечить запуск модуля после редиректа браузера на [страницу авторизации](#) и ввода пользователем логина и пароля.

## Проверка работы автоматической авторизации

Настоятельно рекомендуем проверить работу ссылки быстрого доступа к почте на тестовых пользователях, чтобы в случае ошибок модуля авторизации не помешать нормальной работе вашего сайта.

Если вы уверены, что на вашей стороне все сделано правильно, но при этом авторизация не происходит или наши методы возвращают ошибки, сообщите об этом в службу поддержки через форму обратной связи. Для оперативного решения проблемы рекомендуем указывать в письме:

- ваш домен и имя ящика, для которого не работает авторизация;
- URL вашего сайта, на котором размещена ссылка быстрого доступа к почте;
- методы API, которые вы вызывали, или на которые выполняли редирект;
- результаты, которые вернули используемые вами методы;
- URL или (если это возможно) скриншот страницы, которую вы (или ваши пользователи) увидели после нажатия ссылки быстрого доступа к почте;
- исходный код модуля авторизации (если возможно).

## Импорт почты

Описаны методы API Яндекс.Почты для доменов, предназначенные для управления импортом почты.

Метод	Назначение
<a href="#">set_domain</a>	Сохранение настроек импорта для домена
<a href="#">start_import</a>	Запуск импорта почтового ящика
<a href="#">check_import</a>	Проверка состояния процесса импорта почты
<a href="#">stop_import</a>	Останов импорта почты
<a href="#">reg_and_imp</a>	Регистрация пользователя и запуск импорта почты
<a href="#">import_imap</a>	Запуск импорта папки по IMAP

### set\_domain

Метод позволяет сохранить настройки импорта для домена.

Перед запуском импорта следует сохранить настройки импорта для своего домена: адрес POP3 сервера и порт (если используется нестандартный).

### Синтаксис запроса

```
https://pddimp.yandex.ru/set_domain.xml ?
token=<токен>
& method=<протокол>
& ext_serv=<доменное имя сервера-источника>
& [ext_port=<порт сервера-источника>]
& [isssl=<поддержка SSL>]
& [callback=<запрос подтверждения импорта>]
```

token	Авторизационный токен.
method	Протокол: POP3 или IMAP.
ext_serv	Доменное имя POP3 или IMAP сервера-источника.
ext_port	Параметр указывается, если номер порта отличается от стандартного для данного протокола. Стандартные порты: 110 – POP3 без SSL; 995 – POP3 с SSL.
isssl	Параметр определяет, поддерживает ли сервер SSL. Если соединение осуществляется по SSL, параметр не указывается. В противном случае необходимо указать значение «по».
callback	Если параметр не пустой, то по окончании импорта пользовательского ящика выполняется HTTP-запрос по этому адресу с параметром <code>login="логин импортированного пользователя"</code> .  В случае корректного импорта запрос возвращает XML-структуру вида: <code>&lt;page&gt;&lt;status&gt;moved&lt;/status&gt;&lt;/page&gt;</code> .

### Формат ответа

Если ошибок нет, метод возвращает `<ok/>`, в противном случае — `<error reason='...' />`.

## start\_import

Метод предназначен для запуска импорта почтового ящика.

### Синтаксис запроса

```
https://pddimp.yandex.ru/start_import.xml ?
token=<токен>
& login=<имя почтового ящика>
& [ext_login=<логин пользователя на сервере-источнике>]
& password=<пароль пользователя>
```

token	Авторизационный токен.
login	Имя почтового ящика.
ext_login	Логин пользователя на сервере-источнике. Если логин на сервере-источнике совпадает с логином создаваемого ящика, то указывать параметр не надо.
password	Пароль пользователя на сервере-источнике.

## check\_import

Метод позволяет проверить состояние импорта почты.

### Синтаксис запроса

```
https://pddimp.yandex.ru/check_import.xml ?
token=<токен>
& login=<логин пользователя>
```

token	Авторизационный токен.
login	Логин нового пользователя в Яндексе; указывается часть адреса до символа «@».

## stop\_import

Метод предназначен для остановки импорта почтового ящика.

### Синтаксис запроса

```
https://pddimp.yandex.ru/stop_import.xml ?
token=<токен>
& login=<имя почтового ящика>
```

token	Авторизационный токен.
login	Имя почтового ящика.

## reg\_and\_imp

Метод позволяет зарегистрировать пользователя и запустить импорт его почты.

### Синтаксис запроса

```
https://pddimp.yandex.ru/reg_and_imp.xml ?
token=<токен>
& login=<логин пользователя>
& [ext_login=<логин пользователя на сервере-источнике>]
& inn_password=<пароль пользователя в Яндекс>
& ext_password=<пароль пользователя на сервере-источнике>
& [fwd_email=<e-mail для переадресации>]
& [fwd_copy=<признак сохранения копий писем>]
```

token	Авторизационный токен.
login	Логин пользователя для доступа к создаваемому почтовому ящику.
ext_login	Логин пользователя на сервере-источнике. Если логин на сервере-источнике совпадает с логином создаваемого ящика, то указывать параметр не надо.
inn_password	Пароль пользователя в Яндекс.
ext_password	Пароль пользователя на сервере-источнике.
fwd_email	Если параметр задан, то для ящика будет выполнена переадресация на e-mail, заданный в этом параметре, например, fwdaddress@somedomain.name
fwd_copy	Параметр имеет смысл только при заданном fwd_email. Если fwd_copy задан и равен 0, то после переадресации копии писем в текущем ящике не сохраняются. В противном случае копии писем сохраняются в текущем ящике.

### Формат ответа

Если ошибок нет, метод возвращает <ok/>, в противном случае — <error reason='...' />.

## import\_imap

Метод позволяет запустить импорт одной папки по IMAP.

Метод запускает импорт одной IMAP папки пользователя. Если пользователь не существует, то он создается, и для него задается пароль. Старые письма из папки не удаляются, то есть повторно запущенный импорт заново скопирует в папку все письма.

### Синтаксис запроса

```
https://pddimp.yandex.ru/import_imap.xml ?
token=<токен>
& login=<логин пользователя>
& [ext_login=<логин пользователя на сервере-источнике>]
& ext_password=<пароль пользователя на сервере-источнике>
& [int_password=<пароль пользователя на сервере-получателе>]
& [copy_one_folder=<имя копируемой папки>]
```

token	Авторизационный токен.
login	Имя почтового ящика.

---

<code>ext_login</code>	Логин пользователя на сервере-источнике. Если логин на сервере-источнике совпадает с логином создаваемого ящика, то указывать параметр не надо.
<code>ext_password</code>	Пароль пользователя на сервере-источнике.
<code>int_password</code>	Пароль пользователя в почте для доменов. Параметр указывается только в случае создания при импорте нового почтового ящика. Если почтовый ящик существует, параметр <code>int_password</code> игнорируется.
<code>copy_one_folder</code>	Имя папки, которую надо импортировать по IMAP. Если имя папки содержит символы кириллицы, то оно должно передаваться в UTF-8 (и закодировано в url encoded).

## Управление DKIM

Описаны методы API Яндекс.Почты для доменов, предназначенные для управления использованием DKIM.

DKIM ([DomainKeys Identified Mail](#)) — механизм подписи электронных писем, позволяющий идентифицировать домен отправителя. Такая подпись также уменьшает риск того, что письмо будет помечено как спам.

С помощью API Почты для доменов вы можете включать и отключать подпись писем, отправляемых с вашего домена.

Метод	Назначение
<a href="#">enable</a>	Включение DKIM.
<a href="#">status</a>	Получение статуса DKIM.
<a href="#">disable</a>	Отключение DKIM.

### enable

Метод позволяет включить использование DKIM для домена.

### Синтаксис запроса

```
https://pddimp.yandex.ru/dkim/enable.xml ?
token=<токен>
& domain=<имя домена>
```

token            Авторизационный токен.

domain            Имя домена, для которого следует включить DKIM.

### Формат ответа

XML-документ включает все элементы, которые могут войти в ответ, в том числе взаимоисключающие. Элементы приведены с примерами значений.

```
<domains>
  <action-status>
    <success/>
    <error>ERROR 403: Forbidden.</error>
  </action-status>
  <domain>
    <name>example.com</name>
    <dkim>
      <enabled>yes|no</enabled>
      <txtrecord>mail._domainkey IN TXT "v=DKIM1; k=rsa; t=s; p=MIGfMA0GCS//
EBtaCOteH4EBqJlKperJ
+5BPEGS7N3fFkdeK1lShrM73nm4xPdZmt2jNnmgWMeQySGYW5VUJ8PCePanwIXcW8YnqS7zw+grL/
PHhUt3ofSLmtVM3rSWmJ9qHFhxWmPFp1Pe5OsvpO+fphiMOrTnzzV/004S/jQIDAQAB" ;
DKIM key mail for example.com</txtrecord>
    </dkim>
  </domain>
</domains>
```

action-status

Статус обработки запроса. Вложенный элемент зависит от исхода обработки:

- `<success/>` — операция завершена успешно.
- `<error>` — операция завершена с ошибкой. Описание ошибки содержится в значении элемента. Возможные ошибки:

Описание ошибки	Комментарий
ERROR 400: Bad Request.	Отсутствует обязательный параметр, или указан несуществующий домен.
ERROR 403: Forbidden.	Переданный авторизационный токен невалиден.
domain	Домен, над которым производится операция.
name	Имя домена.
enabled	Статус опции DKIM. Возможные значения: <ul style="list-style-type: none"> <li>• «yes» — функция подписи писем включена.</li> <li>• «no» — функция отключена.</li> </ul>
txtrecord	ТХТ-запись с открытым ключом DKIM для самостоятельной настройки. Запись состоит из трех частей: <ul style="list-style-type: none"> <li>• Имя записи: <code>mail._domainkey</code></li> <li>• Параметры DKIM: <pre>v=DKIM1; k=rsa; t=s; p=MIGfMA0GCS//EBtaCoteH4EBqJlKperJ +5BPEGS7N3fFkdeKl1ShrM73nm4xPdZmt2jNnmgWMeQySGYW5VUJ8PCePanwIXcW8YngS7 zw+grL/PHhUtF3ofSLmtVM3rSWmJ9qHFhxWmPFp1Pe50svp0+fphiMOrTnzZV/004S/ jQIDAQAB</pre> </li> </ul> <p>Параметр <code>p</code> должен содержать открытый ключ DKIM.</p> <ul style="list-style-type: none"> <li>• Указание на домен ПДД: <code>DKIM key mail for example.com</code></li> </ul>

## status

Метод предназначен для проверки статуса DKIM для домена.

### Синтаксис запроса

```
https://pddimp.yandex.ru/dkim/status.xml ?
token=<токен>
& domain=<имя домена>
& [secretkey=<yes>]
```

token	Авторизационный токен.
domain	Имя домена ПДД.
secretkey	Параметр, с помощью которого можно запросить секретный ключ DKIM для вашего домена. Поддерживается только значение «yes».

### Формат ответа

XML-документ включает все элементы, которые могут войти в ответ, в том числе взаимоисключающие. Элементы приведены с примерами значений.



```

<domains>
  <action-status>
    <success/>
    <error>ERROR 400: Bad Request.</error>
  </action-status>
  <domain>
    <name>example.com</name>
    <dkim>
      <enabled>yes</enabled>
      <txtrecord>mail._domainkey IN TXT "v=DKIM1; k=rsa; t=s; p=MIGfMA0GCS//
EBtaCOteH4EBqJlKperJ
+5BPEGS7N3fFkdeK1lShrM73nm4xPdZmt2jNnmgWMeQySGYW5VUJ8PCePanwIXcW8YnqS7zw+grL/
PHhUtf3ofSLmtVM3rSWmJ9qHFhxWmPFplPe5OsvpO+fphiMOrTnzzV/004S/jQIDAQAB" ; DKIM key
mail for example.com</txtrecord>
      <nsready>yes</nsready>
      <mailready>yes</mailready>
      <secretkey>[секретный ключ RSA]</secretkey>
    </dkim>
  </domain>
</domains>

```

action-status

Статус обработки запроса. Вложенный элемент зависит от исхода обработки:

- <success/> — операция завершена удачно.
- <error> — операция завершена с ошибкой. Описание ошибки содержится в значении элемента. Возможные ошибки:

Описание ошибки	Комментарий
ERROR 400: Bad Request.	Отсутствует обязательный параметр, или указан несуществующий домен.
ERROR 403: Forbidden.	Переданный авторизационный токен невалиден.

domain Домен, над которым производится операция.

name Имя домена.

enabled Статус опции DKIM. Возможные значения:

- «yes» — функция подписи писем включена.
- «no» — функция отключена.

txtrecord TXT-запись с открытым ключом DKIM для самостоятельной настройки. Запись состоит из трех частей:

- Имя записи: mail.\_domainkey
- Параметры DKIM:

```

v=DKIM1;
k=rsa;
t=s;
p=MIGfMA0GCS//EBtaCOteH4EBqJlKperJ
+5BPEGS7N3fFkdeK1lShrM73nm4xPdZmt2jNnmgWMeQySGYW5VUJ8PCePanwIXcW8YnqS7
zw+grL/PHhUtf3ofSLmtVM3rSWmJ9qHFhxWmPFplPe5OsvpO+fphiMOrTnzzV/004S/
jQIDAQAB

```

Параметр p должен содержать открытый ключ DKIM.

- Указание на домен ПДД: DKIM key mail for example.com

nsready Приznak того, что TXT-запись с открытым ключом DKIM добавлена для домена. Возможные значения:

- «yes» — TXT-запись установлена.
- «no» — TXT-запись отсутствует

Для [делегированных доменов](#) необходимые DNS-записи добавляются автоматически.

mailready	<p>Признак готовности Почты для доменов подписывать письма согласно DKIM. Возможные значения:</p> <ul style="list-style-type: none"> <li>«yes» — ПДД подписывает письма, отправляемые с данного домена.</li> <li>«no» — ПДД не готова подписывать письма для данного домена.</li> </ul> <p>ПДД обновляет данные о DKIM с задержкой. Если признак готовности не стоит, хотя все условия для включения DKIM выполнены, просто подождите.</p>
secretkey	Секретный ключ DKIM, которым Почта для доменов подписывает исходящие письма.

## disable

Метод позволяет отключить использование DKIM для домена.

### Синтаксис запроса

```
https://pddimp.yandex.ru/dkim/disable.xml ?
token=<токен>
& domain=<логин пользователя>
```

token	Авторизационный токен.
domain	Логин нового пользователя в Яндексе; указывается часть адреса до символа «@».

### Формат ответа

XML-документ включает все элементы, которые могут войти в ответ, в том числе взаимоисключающие. Элементы приведены с примерами значений.

```
<domains>
  <action-status>
    <success/>
    <error>ERROR 400: Bad Request.</error>
  </action-status>
  <domain>
    <name>example.com</name>
  </domain>
</domains>
```

action-status	Статус обработки запроса. Вложенный элемент зависит от исхода обработки:
	<ul style="list-style-type: none"> <li>&lt;success/&gt; — операция завершена успешно.</li> <li>&lt;error&gt; — операция завершена с ошибкой. Описание ошибки содержится в значении элемента. Возможные ошибки:</li> </ul>

Описание ошибки	Комментарий
ERROR 400: Bad Request.	Отсутствует обязательный параметр, или указан несуществующий домен.
ERROR 403: Forbidden.	Переданный авторизационный токен невалиден.
domain	Домен, над которым производится операция.
name	Имя домена.

## API DNS

Описаны методы API DNS Яндекс.Почты.

API DNS Яндекс.Почты для доменов позволяет производить административные действия над DNS вашего домена, если вы [делегировали свой домен на Яндекс](#).

API представляет собой HTTP(S)-интерфейс. Параметры передаются в строке запроса или в теле запроса. Все параметры надо передавать в кодировке `urlencode`. Результат выдается в теле ответа.

Ниже в таблице приведен перечень методов и их назначение.

Метод	Назначение
<a href="#">get_token</a>	Получение доступа
<a href="#">add_a_record</a>	Создание А-записей
<a href="#">add_aaaa_record</a>	Создание AAAA-записей
<a href="#">add_cname_record</a>	Создание CNAME-записей
<a href="#">add_mx_record</a>	Создание MX-записей
<a href="#">add_ns_record</a>	Создание NS-записей
<a href="#">add_srv_record</a>	Создание SRV-записей
<a href="#">add_txt_record</a>	Создание TXT-записей
<a href="#">get_domain_records</a>	Чтение записей для зоны домена
<a href="#">edit_a_record</a>	Изменение А-записей
<a href="#">edit_aaaa_record</a>	Изменение AAAA-записей
<a href="#">edit_cname_record</a>	Изменение CNAME-записей
<a href="#">edit_mx_record</a>	Изменение MX-записей
<a href="#">edit_ns_record</a>	Изменение NS-записей
<a href="#">edit_srv_record</a>	Изменение SRV-записей
<a href="#">edit_soa_record</a>	Изменение SOA-записей
<a href="#">edit_txt_record</a>	Изменение TXT-записей
<a href="#">delete_record</a>	Удаление записей

### get\_token

Метод предназначен для получения авторизационного токена.

Авторизационный токен используется для авторизации в API DNS, которая требуется для обращения к остальным методам API. Получать токен нужно только один раз.

Чтобы получить токен, следует иметь подключенный домен, авторизоваться его администратором и перейти по указанному ниже адресу.

### Синтаксис запроса

```
https://pddimp.yandex.ru/get_token.xml ? domain_name=<имя домена>
```

`domain_name` Имя домена, для которого запрашивается авторизационный токен.

### Пример

Получение токена для домена `example.com`:

```
https://pddimp.yandex.ru/get_token.xml?domain_name=example.com
```

## Формат ответа

Если ошибок нет, метод возвращает следующий XML-ответ:

```
<page>
  <ok token="d604468e74ffce4cb31ceef915a8739a3a5cb9dec"/>
</page>
```

XML-ответ в случае ошибки:

```
<page>
  <error reason="no token found"/>
</page>
```

## add\_a\_record

Метод предназначен для создания А-записей.

## Синтаксис запроса

```
https://pddimp.yandex.ru/nsapi/add_a_record.xml ?
token=<токен пользователя>
& domain=<имя домена>
& [subdomain=<имя поддомена>]
& [ttl=<время жизни записи>]
& content=<содержимое записи>
```

token	Авторизационный токен пользователя API.
domain	Имя домена, для которого добавляется или меняется DNS-запись.
subdomain	Имя поддомена. Параметр необходим, если требуется создать или редактировать запись не для самого домена (например, example.com), а для его поддомена. Например, если subdomain=www, создается или редактируется запись для http://www.example.com. При subdomain=mail — для mail.example.com.
ttl	Время жизни записи в секундах (если не указывать, будет использоваться значение по умолчанию — «21600»).
content	IP-адрес для записи. Указывается в стандартном формате IPv4 (например, «194.84.46.241»).

## Формат ответа

Метод возвращает XML-ответ вида:

```
<page>
  <domains>
    <domain>
      <name>example.com</name>
    </domain>
    <error>[статус обработки]</error>
  </domains>
</page>
```

Если ошибок нет, элемент error содержит значение «ок», в противном случае — сообщение об ошибке.

## add\_aaaa\_record

Метод предназначен для создания AAAA-записей.

### Синтаксис запроса

```
https://pddimp.yandex.ru/nsapi/add_aaaa_record.xml ?
token=<токен пользователя>
& domain=<имя домена>
& [subdomain=<имя субдомена>]
& [ttl=<время жизни записи>]
& content=<содержимое записи>
```

token	Авторизационный токен пользователя API.
domain	Имя домена, для которого добавляется или меняется DNS-запись.
subdomain	Имя поддомена. Параметр необходим, если требуется создать или редактировать запись не для самого домена (например, example.com), а для его поддомена. Например, если subdomain=www, создается или редактируется запись для http://www.example.com. При subdomain=mail — для mail.example.com.
ttl	Время жизни записи в секундах (если не указывать, будет использоваться значение по умолчанию — «21600»).
content	IP-адрес для записи. Указывается в стандартном формате IPv6 (например, «2001:0db8:11a3:09d7:1f34:8a2e:07a0:765d»).

### Формат ответа

Метод возвращает XML-ответ вида:

```
<page>
  <domains>
    <domain>
      <name>example.com</name>
    </domain>
    <error>[статус обработки]</error>
  </domains>
</page>
```

Если ошибок нет, элемент error содержит значение «ок», в противном случае — сообщение об ошибке.

## add\_cname\_record

Метод предназначен для создания CNAME-записей.

### Синтаксис запроса

```
https://pddimp.yandex.ru/nsapi/add_cname_record.xml ?
token=<токен пользователя>
& domain=<имя домена>
& [subdomain=<имя субдомена>]
& [ttl=<время жизни записи>]
& content=<содержимое записи>
```

token	Авторизационный токен пользователя API.
domain	Имя домена, для которого добавляется или меняется DNS-запись.

subdomain	Имя поддомена. Параметр необходим, если требуется создать или редактировать запись не для самого домена (например, example.com), а для его поддомена. Например, если subdomain=www, создается или редактируется запись для http://www.example.com. При subdomain=mail — для mail.example.com.
ttl	Время жизни записи в секундах (если не указывать, будет использоваться значение по умолчанию — «21600»).
content	Полное доменное имя, например, example.com.

## Формат ответа

Метод возвращает XML-ответ вида:

```
<page>
  <domains>
    <domain>
      <name>example.com</name>
    </domain>
    <error>[статус обработки]</error>
  </domains>
</page>
```

Если ошибок нет, элемент error содержит значение «ok», в противном случае — сообщение об ошибке.

## add\_mx\_record

Метод предназначен для создания MX-записей.

## Синтаксис запроса

```
https://pddimp.yandex.ru/nsapi/add_mx_record.xml ?
token=<токен пользователя>
& domain=<имя домена>
& [subdomain=<имя поддомена>]
& [ttl=<время жизни записи>]
& content=<содержимое записи>
& [priority=<приоритет записи>]
```

token	Авторизационный токен пользователя API.
domain	Имя домена, для которого добавляется или меняется DNS-запись.
subdomain	Имя поддомена. Параметр необходим, если требуется создать или редактировать запись не для самого домена (например, example.com), а для его поддомена. Например, если subdomain=www, создается или редактируется запись для http://www.example.com. При subdomain=mail — для mail.example.com.
ttl	Время жизни записи в секундах (если не указывать, будет использоваться значение по умолчанию — «21600»).
content	Полное доменное имя, например, example.com.
priority	Приоритет записи. Чем меньше значение, тем выше приоритет. По умолчанию значение параметра равно 10.

## Формат ответа

Метод возвращает XML-ответ вида:

```
<page>
  <domains>
    <domain>
      <name>example.com</name>
    </domain>
    <error>[статус обработки]</error>
  </domains>
</page>
```

Если ошибок нет, элемент `error` содержит значение «ok», в противном случае — сообщение об ошибке.

## add\_ns\_record

Метод предназначен для создания NS-записей.

### Синтаксис запроса

```
https://pddimp.yandex.ru/nsapi/add_ns_record.xml ?
token=<токен пользователя>
& domain=<имя домена>
& [subdomain=<имя субдомена>]
& [ttl=<время жизни записи>]
& content=<содержимое записи>
```

token	Авторизационный токен пользователя API.
domain	Имя домена, для которого добавляется или меняется DNS-запись.
subdomain	Имя поддомена. Параметр необходим, если требуется создать или редактировать запись не для самого домена (например, <code>example.com</code> ), а для его поддомена. Например, если <code>subdomain=www</code> , создается или редактируется запись для <code>http://www.example.com</code> . При <code>subdomain=mail</code> — для <code>mail.example.com</code> .
ttl	Время жизни записи в секундах (если не указывать, будет использоваться значение по умолчанию — «21600»).
content	Полное доменное имя, например, <code>example.com</code> .

### Формат ответа

Метод возвращает XML-ответ вида:

```
<page>
  <domains>
    <domain>
      <name>example.com</name>
    </domain>
    <error>[статус обработки]</error>
  </domains>
</page>
```

Если ошибок нет, элемент `error` содержит значение «ok», в противном случае — сообщение об ошибке.

## add\_srv\_record

Метод предназначен для создания SRV-записей.

### Синтаксис запроса

SRV-запись составляется из значений переданных параметров. Например, запись «5 0 5269 xmpp.yandex.ru» состоит из следующих частей:

- 5 — приоритет записи (параметр `priority`);
- 0 — вес записи (параметр `weight`);
- 5269 — порт, на котором предоставляется сервис (параметр `port`);
- «xmpp.yandex.ru.» — имя домена, на котором предоставляется сервис (параметр `target`).

```
https://pddimp.yandex.ru/nsapi/add_srv_record.xml ?
token=<токен пользователя>
& domain=<имя домена>
& [subdomain=<имя субдомена>]
& [ttl=<время жизни записи>]
& [priority=<приоритет записи>]
& weight=<вес записи>
& port=<порт сервиса>
& target=<имя хоста>
```

<code>token</code>	Авторизационный токен пользователя API.
<code>domain</code>	Имя домена, для которого добавляется или меняется DNS-запись.
<code>subdomain</code>	Имя поддомена. Параметр необходим, если требуется создать или редактировать запись не для самого домена (например, <code>example.com</code> ), а для его поддомена. Например, если <code>subdomain=www</code> , создается или редактируется запись для <code>http://www.example.com</code> . При <code>subdomain=mail</code> — для <code>mail.example.com</code> .
<code>ttl</code>	Время жизни записи в секундах (если не указывать, будет использоваться значение по умолчанию — «21600»).
<code>priority</code>	Приоритет записи. Чем меньше значение, тем выше приоритет. По умолчанию значение параметра равно 10.
<code>weight</code>	Вес SRV-записи относительно других SRV-записей для того же домена, с тем же приоритетом.
<code>port</code>	TCP или UDP порт хоста, на котором размещен сервис.
<code>target</code>	Каноническое имя хоста, предоставляющего сервис.

### Формат ответа

Метод возвращает XML-ответ вида:

```
<page>
  <domains>
    <domain>
      <name>example.com</name>
    </domain>
    <error>[статус обработки]</error>
  </domains>
</page>
```

Если ошибок нет, элемент `error` содержит значение «ok», в противном случае — сообщение об ошибке.



## add\_txt\_record

Метод предназначен для создания TXT-записей.

### Синтаксис запроса

```
https://pddimp.yandex.ru/nsapi/add_txt_record.xml ?
token=<токен пользователя>
& domain=<имя домена>
& [subdomain=<имя субдомена>]
& [ttl=<время жизни записи>]
& content=<содержимое записи>
```

token	Авторизационный токен пользователя API.
domain	Имя домена, для которого добавляется или меняется DNS-запись.
subdomain	Имя поддомена. Параметр необходим, если требуется создать или редактировать запись не для самого домена (например, example.com), а для его поддомена. Например, если subdomain=www, создается или редактируется запись для http://www.example.com. При subdomain=mail — для mail.example.com.
ttl	Время жизни записи в секундах (если не указывать, будет использоваться значение по умолчанию — «21600»).
content	Текст для TXT-записи. Например: «v=spf1 redirect=_spf.yandex.ru».

### Формат ответа

Метод возвращает XML-ответ вида:

```
<page>
  <domains>
    <domain>
      <name>example.com</name>
    </domain>
    <error>[статус обработки]</error>
  </domains>
</page>
```

Если ошибок нет, элемент error содержит значение «ok», в противном случае — сообщение об ошибке.

## get\_domain\_records

Метод предназначен для чтения записей в зоне домена.

### Синтаксис запроса

```
https://pddimp.yandex.ru/nsapi/get_domain_records.xml ?
token=<токен пользователя>
& domain=<имя домена>
```

token	Авторизационный токен пользователя API.
domain	Имя домена, для которого выполняется чтение записей.

### Формат ответа

Метод возвращает XML-ответ вида:

```
<page>
  <domains>
    <domain>
      <name>example.com</name>
      <response>
        <record domain="yourdomain.ru" priority="" ttl="21600" subdomain="www"
type="A" id="342432432">127.0.0.1</record>
      </response>
      <nsdelegated/>
    </domain>
    <error>ok</error>
  </domains></page>
```

name	Имя домена, записи для которого были запрошены.
record	Адрес, указанный в DNS-записи. Характеристики записи указаны в атрибутах элемента.
priority	Приоритет DNS-записи.
ttl	Время жизни записи в секундах.
subdomain	Имя поддомена.
type	Тип DNS-записи.
id	Идентификатор DNS-записи.
nsdelegated	Признак домена, делегированного Яндексу. Если домен не делегирован, элемент в ответ не включается.
error	Если ошибок нет, элемент error содержит значение «ok», в противном случае — сообщение об ошибке.

## edit\_a\_record

Метод предназначен для изменения A-записей.

### Синтаксис запроса

```
https://pddimp.yandex.ru/nsapi/edit_a_record.xml ?
token=<токен пользователя>
& domain=<имя домена>
& [subdomain=<имя поддомена>]
& record_id=<id записи>
& [ttl=<время жизни записи>]
& content=<содержимое записи>
```

token	Авторизационный токен пользователя API.
domain	Имя домена, для которого добавляется или меняется DNS-запись.
subdomain	Имя поддомена. Параметр необходим, если требуется создать или редактировать запись не для самого домена (например, example.com), а для его поддомена. Например, если subdomain=www, создается или редактируется запись для http://www.example.com. При subdomain=mail — для mail.example.com.
record_id	Идентификатор записи. Идентификатор содержится в значении атрибута id элемента record, который возвращается методом <a href="#">get_domain_records</a> .
ttl	Время жизни записи в секундах (если не указывать, будет использоваться значение по умолчанию — «21600»).
content	IP-адрес для записи. Указывается в стандартном формате IPv4 (например, «194.84.46.241»).

### Формат ответа

Метод возвращает XML-ответ вида:

```
<page>
  <domains>
    <domain>
      <name>example.com</name>
    </domain>
    <error>[статус обработки]</error>
  </domains>
</page>
```

Если ошибок нет, элемент `error` содержит значение «ok», в противном случае — сообщение об ошибке.

## edit\_aaaa\_record

Метод предназначен для изменения AAAA-записей.

### Синтаксис запроса

```
https://pddimp.yandex.ru/nsapi/edit_aaaa_record.xml ?
token=<токен пользователя>
& domain=<имя домена>
& [subdomain=<имя субдомена>]
& record_id=<id записи>
& [ttl=<время жизни записи>]
& content=<содержимое записи>
```

token	Авторизационный токен пользователя API.
domain	Имя домена, для которого добавляется или меняется DNS-запись.
subdomain	Имя поддомена. Параметр необходим, если требуется создать или редактировать запись не для самого домена (например, <code>example.com</code> ), а для его поддомена. Например, если <code>subdomain=www</code> , создается или редактируется запись для <code>http://www.example.com</code> . При <code>subdomain=mail</code> — для <code>mail.example.com</code> .
record_id	Идентификатор записи. Идентификатор содержится в значении атрибута <code>id</code> элемента <code>record</code> , который возвращается методом <a href="#">get_domain_records</a> .
ttl	Время жизни записи в секундах (если не указывать, будет использоваться значение по умолчанию — «21600»).
content	IP-адрес для записи. Указывается в стандартном формате IPv6 (например, «2001:0db8:11a3:09d7:1f34:8a2e:07a0:765d»).

### Формат ответа

Метод возвращает XML-ответ вида:

```
<page>
  <domains>
    <domain>
      <name>example.com</name>
    </domain>
    <error>[статус обработки]</error>
  </domains>
</page>
```

Если ошибок нет, элемент `error` содержит значение «ok», в противном случае — сообщение об ошибке.

## edit\_cname\_record

Метод предназначен для изменения CNAME-записей.

### Синтаксис запроса

```
https://pddimp.yandex.ru/nsapi/edit_cname_record.xml ?
token=<токен пользователя>
& domain=<имя домена>
& [subdomain=<имя субдомена>]
& record_id=<id записи>
& [ttl=<время жизни записи>]
& content=<содержимое записи>
```

token	Авторизационный токен пользователя API.
domain	Имя домена, для которого добавляется или меняется DNS-запись.
subdomain	Имя поддомена. Параметр необходим, если требуется создать или редактировать запись не для самого домена (например, example.com), а для его поддомена. Например, если subdomain=www, создается или редактируется запись для http://www.example.com. При subdomain=mail — для mail.example.com.
record_id	Идентификатор записи. Идентификатор содержится в значении атрибута id элемента record, который возвращается методом <a href="#">get_domain_records</a> .
ttl	Время жизни записи в секундах (если не указывать, будет использоваться значение по умолчанию — «21600»).
content	Полное доменное имя, например, example.com.

### Формат ответа

Метод возвращает XML-ответ вида:

```
<page>
  <domains>
    <domain>
      <name>example.com</name>
    </domain>
    <error>[статус обработки]</error>
  </domains>
</page>
```

Если ошибок нет, элемент error содержит значение «ok», в противном случае — сообщение об ошибке.

## edit\_mx\_record

Метод предназначен для изменения MX-записей.

### Синтаксис запроса

```
https://pddimp.yandex.ru/nsapi/edit_mx_record.xml ?
token=<токен пользователя>
& domain=<имя домена>
& [subdomain=<имя субдомена>]
& record_id=<id записи>
& [ttl=<время жизни записи>]
& content=<содержимое записи>
& [priority=<приоритет записи>]
```

token	Авторизационный токен пользователя API.
domain	Имя домена, для которого добавляется или меняется DNS-запись.
subdomain	Имя поддомена. Параметр необходим, если требуется создать или редактировать запись не для самого домена (например, example.com), а для его поддомена. Например, если subdomain=www, создается или редактируется запись для http://www.example.com. При subdomain=mail — для mail.example.com.
record_id	Идентификатор записи. Идентификатор содержится в значении атрибута id элемента record, который возвращается методом <a href="#">get_domain_records</a> .
ttl	Время жизни записи в секундах (если не указывать, будет использоваться значение по умолчанию — «21600»).
content	IP-адрес, который нужно указать в DNS-записи.
priority	Приоритет записи. Чем меньше значение, тем выше приоритет. По умолчанию значение параметра равно 10.

### Формат ответа

Метод возвращает XML-ответ вида:

```
<page>
  <domains>
    <domain>
      <name>example.com</name>
    </domain>
    <error>[статус обработки]</error>
  </domains>
</page>
```

Если ошибок нет, элемент error содержит значение «ok», в противном случае — сообщение об ошибке.

## edit\_ns\_record

Метод предназначен для изменения NS-записей.

### Синтаксис запроса

```
https://pddimp.yandex.ru/nsapi/edit_ns_record.xml ?
token=<токен пользователя>
& domain=<имя домена>
& [subdomain=<имя субдомена>]
& record_id=<id записи>
& [ttl=<время жизни записи>]
& content=<содержимое записи>
```

token	Авторизационный токен пользователя API.
domain	Имя домена, для которого добавляется или меняется DNS-запись.
subdomain	Имя поддомена. Параметр необходим, если требуется создать или редактировать запись не для самого домена (например, example.com), а для его поддомена. Например, если subdomain=www, создается или редактируется запись для http://www.example.com. При subdomain=mail — для mail.example.com.
record_id	Идентификатор записи. Идентификатор содержится в значении атрибута id элемента record, который возвращается методом <a href="#">get_domain_records</a> .
ttl	Время жизни записи в секундах (если не указывать, будет использоваться значение по умолчанию — «21600»).
content	IP-адрес, который нужно указать в DNS-записи.

### Формат ответа

Метод возвращает XML-ответ вида:

```
<page>
  <domains>
    <domain>
      <name>example.com</name>
    </domain>
    <error>[статус обработки]</error>
  </domains>
</page>
```

Если ошибок нет, элемент error содержит значение «ok», в противном случае — сообщение об ошибке.

## edit\_srv\_record

Метод предназначен для изменения SRV-записей.

### Синтаксис запроса

```
https://pddimp.yandex.ru/nsapi/edit_srv_record.xml ?
token=<токен пользователя>
& domain=<имя домена>
& [subdomain=<имя субдомена>]
& record_id=<id записи>
& [ttl=<время жизни записи>]
& [priority=<приоритет записи>]
& weight=<вес записи>
& port=<порт сервиса>
& target=<имя хоста>
```

token	Авторизационный токен пользователя API.
domain	Имя домена, для которого добавляется или меняется DNS-запись.
subdomain	Имя поддомена. Параметр необходим, если требуется создать или редактировать запись не для самого домена (например, example.com), а для его поддомена. Например, если subdomain=www, создается или редактируется запись для http://www.example.com. При subdomain=mail — для mail.example.com.
record_id	Идентификатор записи. Идентификатор содержится в значении атрибута id элемента record, который возвращается методом <a href="#">get_domain_records</a> .
ttl	Время жизни записи в секундах (если не указывать, будет использоваться значение по умолчанию — «21600»).
priority	Приоритет записи. Чем меньше значение, тем выше приоритет. По умолчанию значение параметра равно 10.
weight	Вес SRV-записи относительно других SRV-записей для того же домена, с тем же приоритетом.
port	TCP или UDP порт хоста, на котором размещен сервис.
target	Каноническое имя хоста, предоставляющего сервис.

### Формат ответа

Метод возвращает XML-ответ вида:

```
<page>
  <domains>
    <domain>
      <name>example.com</name>
    </domain>
    <error>[статус обработки]</error>
  </domains>
</page>
```

Если ошибок нет, элемент error содержит значение «ok», в противном случае — сообщение об ошибке.

## edit\_soa\_record

Метод предназначен для изменения SOA-записи.

### Синтаксис запроса

Новое значение SOA-записи составляется из значений переданных параметров. Например, запись «nsl.yandex.ru. sysadmin.yourdomain.ru. 2012122006 600 300 2592000 900» состоит из следующих частей:

- «nsl.yandex.ru.» — DNS-сервер (API Почты для доменов определяет DNS-сервер автоматически).
- «sysadmin.yourdomain.ru.» — e-mail администратора домена (параметр `admin_mail`).
- 2012122006 — версия SOA-записи (API Почты для доменов определяет версию автоматически).
- 600 — период обновления второстепенных DNS-серверов (параметр `refresh`).
- 300 — период повтора попыток к первичному серверу (параметр `retry`).
- 2592000 — время жизни записей зоны (параметр `expire`).
- 900 — срок кэширования отрицательного ответа сервиса (параметр `neg_cache`).

```
https://pddimp.yandex.ru/nsapi/edit_soa_record.xml ?
token=<токен пользователя>
& domain=<имя домена>
& [ttl=<время жизни записи>]
& admin_mail=<е-mail администратора домена>
& refresh=<пауза между обновлениями>
& retry=<время между попытками получения записи>
& expire=<тайм-аут чтения записи>
& neg_cache=<длительность кэширования отрицательного ответа сервера>
```

<code>token</code>	Авторизационный токен пользователя API.
<code>domain</code>	Имя домена, для которого добавляется или меняется DNS-запись.
<code>ttl</code>	Время жизни записи в секундах (если не указывать, будет использоваться значение по умолчанию — «21600»).
<code>admin_mail</code>	E-mail адрес администратора домена. Отображается в SOA-записи.
<code>refresh</code>	Время между обновлениями slave DNS-серверов для записей зоны.
<code>retry</code>	Время между повторными попытками slave DNS-серверов получить записи зоны (в случае, если master сервер ничего не вернул).
<code>expire</code>	Время, по истечении которого slave DNS-сервера считают записи зоны несуществующими (в случае если master сервер продолжает ничего не возвращать).
<code>neg_cache</code>	Время, на которое будет кэшироваться отрицательный ответ от DNS-сервера (имеется в виду ответ <code>ERROR = NXDOMAIN</code> — нет такой записи).

### Формат ответа

Метод возвращает XML-ответ вида:

```
<page>
  <domains>
    <domain>
      <name>example.com</name>
    </domain>
    <error>[статус обработки]</error>
  </domains>
</page>
```

Если ошибок нет, элемент `error` содержит значение «ok», в противном случае — сообщение об ошибке.



## edit\_txt\_record

Метод предназначен для изменения TXT-записей.

### Синтаксис запроса

```
https://pddimp.yandex.ru/nsapi/edit_txt_record.xml ?
token=<токен пользователя>
& domain=<имя домена>
& [subdomain=<имя субдомена>]
& record_id=<id записи>
& [ttl=<время жизни записи>]
& content=<содержимое записи>
```

token	Авторизационный токен пользователя API.
domain	Имя домена, для которого добавляется или меняется DNS-запись.
subdomain	Имя поддомена. Параметр необходим, если требуется создать или редактировать запись не для самого домена (например, example.com), а для его поддомена. Например, если subdomain=www, создается или редактируется запись для http://www.example.com. При subdomain=mail — для mail.example.com.
record_id	Идентификатор записи. Идентификатор содержится в значении атрибута id элемента record, который возвращается методом <a href="#">get_domain_records</a> .
ttl	Время жизни записи в секундах (если не указывать, будет использоваться значение по умолчанию — «21600»).
content	Текст для TXT-записи. Например: «v=spf1 redirect=_spf.yandex.ru».

### Формат ответа

Метод возвращает XML-ответ вида:

```
<page>
  <domains>
    <domain>
      <name>example.com</name>
    </domain>
    <error>[статус обработки]</error>
  </domains>
</page>
```

Если ошибок нет, элемент error содержит значение «ок», в противном случае — сообщение об ошибке.

## delete\_record

Метод предназначен для удаления записей.

### Синтаксис запроса

```
https://pddimp.yandex.ru/nsapi/delete_record.xml ?
token=<токен пользователя>
& domain=<имя домена>
& record_id=<id записи>
```

token	Авторизационный токен пользователя API.
domain	Имя домена, для которого удаляется DNS-запись.

`record_id` Идентификатор записи. Идентификатор содержится в значении атрибута `id` элемента `record`, который возвращается методом `get_domain_records`.

## Формат ответа

Метод возвращает XML-ответ вида:

```
<page>
  <domains>
    <domain>
      <name>example.com</name>
    </domain>
    <error>[статус обработки]</error>
  </domains>
</page>
```

Если ошибок нет, элемент `error` содержит значение «ок», в противном случае — сообщение об ошибке.

## Спецификация запроса

В разделе описаны параметры вызовов API-методов и формат ответа.

### Параметры методов

<code>content*</code>	IP-адрес, который нужно указать в DNS-записи.
<code>admin_mail*</code>	E-mail адрес администратора домена. Отображается в SOA-записи.
<code>domain*</code>	Имя домена, для которого добавляется или меняется DNS-запись.
<code>domain_name*</code>	Имя домена, для которого запрашивается авторизационный токен.
<code>expire*</code>	Время, по истечении которого slave DNS-сервера считают записи зоны несуществующими (в случае если master сервер продолжает ничего не возвращать).
<code>neg_cache*</code>	Время, на которое будет кешироваться отрицательный ответ от DNS-сервера (имеется в виду ответ <code>ERROR = NXDOMAIN</code> — нет такой записи).
<code>port*</code>	TCP или UDP порт хоста, на котором размещен сервис.
<code>priority</code>	Приоритет записи. Чем меньше значение, тем выше приоритет. По умолчанию значение параметра равно 10.
<code>record_id</code>	Идентификатор записи. Идентификатор содержится в значении атрибута <code>id</code> элемента <code>record</code> , который возвращается методом <code>get_domain_records</code> .
<code>refresh</code>	Время между обновлениями slave DNS-серверов для записей зоны.
<code>retry</code>	Время между повторными попытками slave DNS-серверов получить записи зоны (в случае, если master сервер ничего не вернул).
<code>subdomain</code>	Имя поддомена. Параметр необходим, если требуется создать или редактировать запись не для самого домена (например, <code>example.com</code> ), а для его поддомена. Например, если <code>subdomain=www</code> , создается или редактируется запись для <code>http://www.example.com</code> . При <code>subdomain=mail</code> — для <code>mail.example.com</code> .
<code>target*</code>	Каноническое имя хоста, предоставляющего сервис.
<code>token*</code>	Авторизационный токен пользователя API.
<code>ttl</code>	Время жизни записи в секундах (если не указывать, будет использоваться значение по умолчанию — «21600»).
<code>weight*</code>	Вес SRV-записи относительно других SRV-записей для того же домена, с тем же приоритетом.

## Формат ответа

Метод возвращает XML-ответ вида:

```
<page>
  <domains>
    <domain>
      <name>example.com</name>
    </domain>
    <error>[статус обработки]</error>
  </domains>
</page>
```

Если ошибок нет, элемент `error` содержит значение «ok», в противном случае — сообщение об ошибке.

## API для хостеров, регистраторов и парковщиков доменов

Описан интерфейс взаимодействия с сервисом Яндекс.Почта для доменов, предназначенный для хостеров, регистраторов и парковщиков доменов.

Описываемый API позволяет хостинг-провайдерам, регистраторам и парковщикам доменов (далее – регистраторы) автоматически (по одной кнопке) подключать домены своих пользователей к сервису Яндекс.Почта для доменов (ПДД) и импортировать их почту в Яндекс.

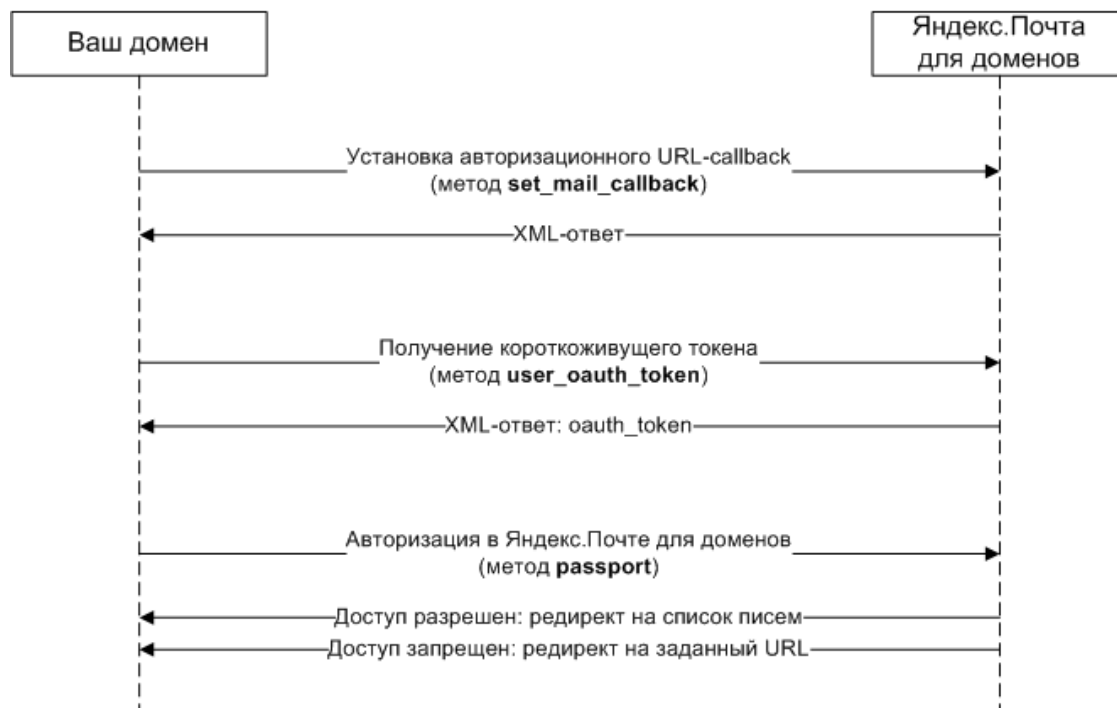
Для этого необходимо:

1. [Получить ID регистратора.](#)
2. [Задать URL'ы обратного вызова.](#)
3. [Выполнить запрос на подключение к ПДД.](#)
4. [Пройти аутентификацию.](#)
5. [Подтвердить владение доменом.](#)
6. [Получить подтверждение подключения домена.](#)
7. [Импортировать почту пользователя.](#)

Кроме того, методы данного API позволяют получать следующую информацию:

- [текущее состояние импорта почты;](#)
- [текущее состояние домена;](#)
- [данные об отключении домена.](#)

Схематично процесс взаимодействия регистратора с ПДД иллюстрирует нижеприведенный рисунок.



Взаимодействие осуществляется с помощью HTTP-методов, вызываемых на стороне ПДД или регистратора.

В таблице приведен перечень методов и их назначение.

Метод	Назначение
Методы, вызываемые на стороне регистратора	

Метод	Назначение
<a href="#">create_registrar</a>	Получение ID регистратора
<a href="#">save_registrar</a>	Задание URL'ов обратного вызова
<a href="#">registraradd/do</a>	Запрос подключения к ПДД
Импорт почты	Импорт почты
<a href="#">registrar_check_import</a>	Проверка состояния импорта почты
<a href="#">registrar_check_domain</a>	Получение информации о текущем состоянии домена
Методы, вызываемые на стороне ПДД	
<a href="#">check_payed_method</a>	Аутентификация домена
<a href="#">added_init_method</a>	Подтверждение владения доменом
<a href="#">added_method</a>	Подтверждение подключения домена
<a href="#">transfer_succeed</a>	Подтверждение импорта почты
<a href="#">delete_method, something_delete</a>	Получение информации об отключении домена

## Получение ID регистратора

Метод `create_registrar` позволяет получить ID регистратора.

Метод вызывается на стороне регистратора. Для получения ID следует авторизоваться под любым пользователем Яндекса и вызвать из того же окна браузера следующий URL:

```
https://pddimp.yandex.ru/create_registrar.xml ?
name=<название регистратора>
& password=<пароль>
```

`name`            Название регистратора.

`password`        Пароль для доступа к API. Тот же самый пароль используется при вызове других методов API.

## Формат ответа

В ответ на вызов метода возвращается XML-структура следующего вида:

```
<registrar>
  <error>ok|...</error>
  <id>...</id>
  <name>...</name>
</registrar>
```

`error*`            Результат выполнения запроса. При успешном вызове метода элемент содержит «ok», в противном случае – сообщение об ошибке.

`id*`                Уникальный ID регистратора.

`name*`             Имя регистратора.

## Задание URL'ов обратного вызова

Метод `save_registrar` позволяет задать URL'ы обратного вызова.

Сервер Яндекса вызывает URL'ы обратного вызова в случае различных событий:

- получение запроса на подключение к ПДД;
- получение ответа на аутентификационный запрос;
- подтверждение владения доменом;
- удаление домена;
- завершение импорта почты.

URL'ы задаются следующим образом:

```
https://pddimp.yandex.ru/save_registrar.xml ?
registrar_id=<id регистратора>
& password=<пароль>
& payed_url=<аутентификационный URL>
& added_init=<URL активации>
& added=<URL подтверждения>
& delete_url=<URL удаления домена>
& transfer_succeed=<URL завершения импорта>
```

<code>registrar_id</code>	ID регистратора, полученный с помощью метода <code>create_registrar</code> .
<code>password</code>	Пароль регистратора. Длина пароля должна быть не менее 8 символов.
<code>payed_url</code>	URL для проверки, действительно ли пользователь инициировал в Вашем веб-интерфейсе подключение домена к Яндекс.Почте.
<code>added_init</code>	URL, вызываемый на стороне ПДД, если домен успешно переведен в состояние «ожидает подтверждения» (см. ниже).
<code>added</code>	URL, вызываемый на стороне ПДД, если домен успешно добавлен, подтвержден и активирован. В момент вызова этого URL пользователь уже может создавать или мигрировать ящики на этом домене и полноценно пользоваться Почтой на этом домене.
<code>delete_url</code>	URL, вызываемый на стороне ПДД, если пользователь, пришедший с этим доменом от Вас, удаляет домен из ПДД.
<code>transfer_succeed</code>	URL, вызываемый на стороне ПДД, если перенос ящиков пользователя успешно завершен.

## Формат ответа

В ответ на вызов метода возвращается XML-структура следующего вида:

```
<registrar>
  <error>ok|...</error>
</registrar>
```

<code>error*</code>	Результат вызова метода. При успешном выполнении элемент содержит «ok», в противном случае – сообщение об ошибке.
---------------------	---

## Запрос подключения к ПДД

Метод `registraradd/do` позволяет выполнить запрос на подключение домена к ПДД.

Запрос выполняется при нажатии на кнопку «Подключить домен к Яндекс.Почте»:

```
https://pdd.yandex.ru/domains/domain/registraradd/do ?
domain=<имя домена>
& registrar_id=<id регистратора>
& service_id=<ID сервиса>
& payed_url=<аутентификационный URL>
```

<code>domain</code>	Имя домена.
<code>registrar_id</code>	ID регистратора, полученный с помощью метода <code>create_registrar</code> .
<code>service_id</code>	Уникальный номер, который регистратор генерирует на своей стороне. При каждом нажатии кнопки «Подключить домен к Яндекс.Почте» генерируется новый <code>service_id</code> . В качестве <code>service_id</code> следует использовать число от 0 до 4294967295. Не используйте в качестве <code>service_id</code> порядковые номера или другие легко подбираемые числа. Посредством этого <code>service_id</code> выполняется <a href="#">аутентификация домена</a> .
<code>payed_url</code>	URL для проверки, действительно ли пользователь инициировал в веб-интерфейсе подключение домена к ПДД.

На странице Яндекс пользователь введет свой логин и пароль к ПДД, после чего со стороны ПДД вызывается метод `check_payed_method`.

## Аутентификация домена

Метод `check_payed_method` позволяет выполнить аутентификационный запрос к домену.

Аутентификационный запрос вызывается на стороне ПДД, позволяя проверить, был ли оплачен домен и хочет ли пользователь подключить его.

Проверка осуществляется следующим образом. При вызове метода `registraradd/do` регистратор генерирует `service_id` и сохраняет на своей стороне пару `service_id` + `domain` (см. раздел [Запрос подключения к ПДД](#)). При выполнении аутентификационного запроса ПДД передает в методе `check_payed_method` `service_id` и имя домена. Регистратор принимает имя домена и `service_id`, сравнивает их с сохраненными и если они совпадают, отправляет подтверждение успешной аутентификации ПДД.

Если запрос `check_payed_method` вернул ошибочный ответ, не вернул ничего или завершился таймаутом, то домен не будет подключен.

Для вызова метода требуется указать в параметре `payed_url` метода `save_registrar` следующий URL: `https://registrar.domain.name/check_payed_method`.

### Внимание!

В случае успешной аутентификации домен не подключается сразу, а только добавляется в состоянии «ожидает подтверждения». Полноценное использование Почты на этом домене возможно только через некоторое время, когда со стороны регистратора поступит подтверждение владением домена и присвоение MX'ов (фактически посредством API выполняется вся работа, которая выполнялась бы пользователем вручную, если бы он подключал домен самостоятельно, а не через кнопку на вашем сайте). Подробнее об этом см. подраздел [Подтверждение владения доменом](#).

Возможен вызов метода `check_payed_method` по HTTP или по HTTPS.

```
http(s)://registrar.domain.name/check_payed_method ?
domain=<имя домена>
& service_id=<ID сервиса>
& (password=<пароль> | sign=<подпись>)
```

domain	Имя домена.
service_id	Уникальный номер, переданный при вызове <code>registraradd/do</code> .
password	Пароль для доступа к API.
sign	<code>sign = sha1 (service_id + domain + password)</code> . Передается только в HTTP версии метода.

### Внимание!

Метод `check_payed_method` выполняется с тайм-аутом 10 секунд, так как он вызывается в контексте формирования пользовательской страницы. Пользователи обычно уже после трехсекундной задержки начинают нервничать. Поэтому при выполнении `payed_url` не рекомендуется выполнять долгие блокирующие операции (если регистрация MX-записей требует таковых). Лучше выполнять их после ответа на HTTP-запрос `payed_url`.

## Формат ответа

В ответ на вызов метода возвращается XML-структура следующего вида:

```
<is_payed>
  <value>True|False</value>
  <error>...</error>
</is_payed>
```

value*	<p>Результат аутентификации:</p> <p>«True» – домен может быть подключен к ПДД. В этом случае регистратор должен прописать MX-записи у домена на <code>mx.yandex.ru</code> с приоритетом 10 и удалить все существующие MX-записи;</p> <p>«False» – отрицательный.</p> <p>В случае ошибки выполнения запроса элемент <code>value</code> не содержит данных.</p>
error*	Результат выполнения запроса. При успешном вызове метода элемент содержит «ок», в противном случае – сообщение об ошибке.

## Подтверждение владения доменом

Подтверждение владения доменом выполняется с помощью метода `added_init_method`.

Подтверждение владения доменом необходимо, чтобы избежать ошибок на Вашей стороне при передаче доменов. Теоретически, в Яндекс может быть передан запрос на подключение произвольного домена. Если этот домен будет подключен автоматически, то впоследствии настоящий владелец домена не сможет его подключить.

Метод `added_init_method` вызывается на стороне Яндекса, если получен положительный ответ на аутентификационный запрос `check_payed_method`.



Получив вызов `added_init_method`, регистратор должен подтвердить владение пользовательским доменом любым из способов:

- создать CNAME `yamail-secret_name` в зоне `domain` со значением `mail.yandex.ru`;
- положить в корень домена файл с именем `secret_name.html` (он должен быть доступен по HTTP), файл должен содержать строку `secret_value`.

Фактически данная процедура выполняется аналогично ручному подтверждению домена, описанному [здесь](#). Если при обработке `payed_url` регистратор не сформировал MX-запись, то ее можно создать, например, при обработке `added_init`, как это указано [здесь](#).

Для вызова данного метода требуется указать URL `http(s)://registrar.domain.name/added_init_method` в параметре `added_init` метода `save_registrar`.

Аналогично тайм-ауту на `check_payed_method`, есть тайм-аут и на `added_init`. Поэтому рекомендуется все долгие операции выносить за пределы этих методов, чтобы не блокировать веб-интерфейс пользователя.

Если запрос `added_init_method` вернул ошибочный ответ, не вернул ничего или завершился тайм-аутом, то домен не будет подключен.

Возможен вызов `added_init_method` по HTTP или HTTPS.

```
http(s)://registrar.domain.name/added_init_method ?
domain=<имя домена>
& secret_name=<секретная строка>
& secret_value=<секретная строка>
& (password=<пароль> | sign=<подпись>)
```

<code>domain</code>	Имя домена.
<code>secret_name</code>	Секретная строка, необходимая для подтверждения владения доменом.
<code>secret_value</code>	Секретная строка, необходимая для подтверждения владения доменом.
<code>password</code>	Пароль для доступа к API.
<code>sign</code>	<code>sign = sha1 хеш от (domain + secret_name + secret_value + password)</code> . Передается только в HTTP версии метода.

Если запрос вернул ошибочный ответ, не вернул ничего или завершился тайм-аутом, то домен не будет подключен.

## Формат ответа

В ответ на вызов метода возвращается XML-структура следующего вида:

```
<registrar>
  <error>ok|...</error>
</registrar>
```

<code>error*</code>	Результат вызова метода. При успешном выполнении элемент содержит «ok», в противном случае – сообщение об ошибке.
---------------------	---

## Подтверждение подключения домена

Метод `added_method` предназначен для информирования регистратора о подключении домена.

В случае успешного подключения домена на его NS-серверах появятся соответствующие CNAME и MX-записи. Яндекс сообщает об этом регистратору, вызывая метод `added_method`. После вызова `added_method` регистратору становится доступен импорт почты из ящиков владельца домена.

Результат вызова `added_method` игнорируется, то есть этот метод работает как уведомление.

Вызов может осуществляться по HTTP или HTTPS.

```
http(s)://registrar.domain.name/added_method ?
domain=<имя домена>
& (password=<пароль> | sign=<подпись>)
```

`domain`           Имя домена.

`password`        Пароль для доступа к API.

`sign`            `sign = sha1(domain + password)`. Передается только в HTTP версии метода.

## Формат ответа

В ответ на вызов метода возвращается XML-структура следующего вида:

```
<registrar>
  <error>ok|...</error>
</registrar>
```

`error*`           Результат вызова метода. При успешном выполнении элемент содержит «ok», в противном случае – сообщение об ошибке.

## Импорт почты

Метод `import_reg_domain` позволяет импортировать почту в Яндекс.

Автоматический импорт почты из ее предыдущего места хранения в Яндекс осуществляется с помощью POST-запроса.

```
https://pddimp.yandex.ru/import_reg_domain.xml
```

Запрос выполняется на стороне регистратора.

Параметры запроса:

- `domain` – имя домена;
- `password` – пароль регистратора, полученный в результате вызова метода `create_registrar`;
- `registrar_id` – ID регистратора, полученный с помощью метода `create_registrar`;
- `mail_proto` – тип почты (IMAP или POP3);
- `use_ssl` – признак использования SSL на IMAP/POP3 сервере:  
0 – не используется;  
1 – используется.
- `mail_server` – доменное имя IMAP/POP3 сервера, на котором хостится почта этого домена;
- `mail_port` – порт. Необязательный параметр. Если отсутствует, то используется порт по умолчанию для протокола;
- `e-mails` – XML-структура с именами импортируемых почтовых ящиков (без собаки и домена) и паролями этих ящиков. XML имеет следующий вид:

```
<emails>
  <email><name>name</name><password>password</password>
  ...
</emails>
```

## Формат ответа

В ответ на вызов метода возвращается XML-структура следующего вида:

```
<domains>
  <domain>
    <name>...</name> - название домена
    <emails>
      <email>
        <login>...</login>
        <error>ok|...</error>
      </email>
      ...
    </emails>
  </domain>
  <error_all>ok|...</error_all>
</domains>
```

name*	Имя домена.
login*	Имя почтового ящика без доменной части.
error*	Если задание на импорт почтового ящика успешно создано, элемент содержит «ок», в противном случае – сообщение об ошибке.
error_all*	Результат выполнения запроса. Если задания на импорт всех почтовых ящиков почты успешно созданы, элемент содержит «ок», в противном случае – сообщение об ошибке.

## Подтверждение импорта почты

Метод `transfer_succeed` предназначен для информирования регистратора об окончании импорта почты.

Метод `transfer_succeed` вызывается на стороне ПДД по окончании импорта почтовых ящиков.

Результат запроса `transfer_succeed` игнорируется, то есть этот метод работает как уведомление.

Вызов метода возможен по HTTPS или HTTP.

```
http(s)://registrar.domain.name/transfer_succeed ?
domain=<имя домена>
& (password=<пароль> | sign=<подпись>)
```

domain	Имя домена.
password	Пароль для доступа к API.
sign	sign = sha1(domain + password). Передается только в HTTP версии метода.

## Формат ответа

В ответ на вызов метода возвращается XML-структура следующего вида:

```
<registrar>
  <error>ok|...</error>
</registrar>
```

error*	Результат вызова метода. При успешном выполнении элемент содержит «ок», в противном случае – сообщение об ошибке.
--------	---

## Получение информации об отключении домена

Методы `delete_method` и `something_delete` предназначены для информирования об отключении домена.

В момент отключения пользователем домена от Яндекс.Почты со стороны ПДД вызывается метод `delete_method` или `something_delete`. Для их вызова необходима передача URL `delete_url` в методе `save_registrar`.

Результаты выполнения `delete_method` или `something_delete` игнорируются, то есть эти методы работают как уведомления.

Метод `delete_method` вызывается по HTTPS:

```
https://registrar.domain.name/delete_method ?
domain=<имя домена>
& password=<пароль>
```

`domain`       Имя домена.

`password`     Пароль для доступа к API.

Для вызова по HTTP используется метод `something_delete`:

```
http://registrar.domain.name/something_delete ?
domain=<имя домена>
& sign=<подпись>
```

`domain`       Имя домена.

`sign`         `sign = sha1(domain + password)`.

## Формат ответа

В ответ на вызов метода возвращается XML-структура следующего вида:

```
<registrar>
  <error>ok|...</error>
</registrar>
```

`error*`       Результат вызова метода. При успешном выполнении элемент содержит «ok», в противном случае – сообщение об ошибке.

## Получение информации о текущем состоянии домена

Метод `registrar_check_domain` позволяет получить информацию о текущем состоянии домена.

```
https://pddimp.yandex.ru/registrar_check_domain.xml ?
domain=<имя домена>
& registrar_id=<id регистратора>
& password=<пароль>
```

`domain`       Имя домена.

`registrar_id` ID регистратора, полученный с помощью метода `create_registrar`.

`password`     Пароль для доступа к API.

## Формат ответа

В ответ на вызов метода возвращается XML-структура следующего вида:

```
<domains>
  <registrar>
    <name>...</name>
  </registrar>
  <domain>
    <domain_name>...</domain_name>
  </domain>
  <stage>
    added|mx-check|owner-check
  </stage>
  <mx>
    <last-date></last-date>
    <next-date></next-date>
    <mx-found>no|...</mx-found>
  </mx>
  <owner>
    <last-date></last-date>
    <next-date></next-date>
    <sub-domain-name>...</sub-domain-name>
    <sub-domain-exists>ok|no</sub-domain-exists>
    <sub-domain-value>no|...</sub-domain-value>
    <file-url>...</file-url>
    <file-exists>ok|no</file-exists>
    <file-content>...</file-content>
    <file-content-match>ok|no</file-content-match>
  </owner>
  <error>ok|...</error>
</domains>
```

`domain_name*` Имя домена.

`stage*` Статус домена:  
 «added» – успешно активирован;  
 «mx-check» – ожидается проверка MX-записей;  
 «owner-check» – ожидается проверка владения доменом.

`mx` Сведения о проверке MX-записи. Элемент присутствует в случае, если в `stage` указан статус домена «mx-check».

`last-date*` Дата и время предыдущей проверки.

`next-date*` Дата и время следующей проверки.

`mx-found*` Содержимое MX-записи проверяемого домена. Если запись не найдена, элемент содержит «no».

`owner` Элемент присутствует в случае, если в `stage` указан статус домена «owner-check».

`sub-domain-name*`  
 Имя поддомена у которого проверяется CNAME-запись.

`sub-domain-value*`  
 Если поддомен существует, указано содержимое его CNAME-записи. Если CNAME-запись отсутствует, элемент содержит «no».

`file-url*` Имя проверяемого файла.

`file-exists*` Результат проверки, существует ли проверяемый файл:

«ok» – существует;  
 «no» – файл не найден.

<code>file-</code>	Строка поиска в проверяемом файле.
<code>content*</code>	
<code>file-content-match*</code>	Признак присутствия строки поиска в проверяемом файле. Возможные значения: <ul style="list-style-type: none"> <li>• «ок» — строка найдена;</li> <li>• «по» — строка не найдена.</li> </ul>
<code>error*</code>	Результат выполнения запроса. При успешном вызове метода элемент содержит «ок», в противном случае – сообщение об ошибке.

## Проверка состояния импорта почты

Метод `registrar_check_import` позволяет получить информацию о текущем состоянии импорта почты.

```
https://pddimp.yandex.ru/registrar_check_import.xml ?
domain=<имя домена>
& registrar_id=<id регистратора>
& password=<пароль>
```

<code>domain</code>	Имя домена.
<code>registrar_id</code>	ID регистратора, полученный с помощью метода <code>create_registrar</code> .
<code>password</code>	Пароль для доступа к API.

## Формат ответа

В ответ на вызов метода возвращается XML-структура следующего вида:

```
<domains>
  <registrar>
    <name>...</name>
  </registrar>
  <domain>
    <domain_name>...</domain_name>
    <emails>
      <email>
        <login>...</login>
        <imported>...</imported>
        <to-import>...</to-import>
        <error_import>no|ok|...</error_import>
      </email>
      ...
    </emails>
  </domain>
  <error>ok|...</error>
</domains>
```

<code>domain_name*</code>	Имя домена.
<code>login*</code>	Имя почтового ящика без доменной части.
<code>imported</code>	Число импортированных писем. Элемент присутствует, если <code>error</code> не содержит сообщение об ошибке.
<code>to-import</code>	Число писем, которые осталось импортировать. Элемент присутствует, если <code>error</code> не содержит сообщение об ошибке.
<code>error_import*</code>	Состояние процесса импорта почты:

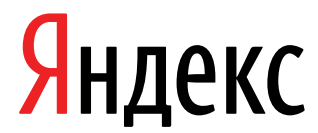
«ok» – импорт почтового ящика завершен успешно;

«no» – импорт почтового ящика не начал.

При некорректном выполнении импорта элемент содержит сообщение об ошибке.

error\*

Результат выполнения запроса. При успешном вызове метода элемент содержит «ok», в противном случае – сообщение об ошибке.



**Почта для доменов**  
Справочник разработчика

9.12.2014