

Документация, содержащая описание
функциональных характеристик
программного обеспечения

**«PROMT Neural Translation Server
Developer Edition»**

и информацию, необходимую для
установки и эксплуатации программного
обеспечения

PROMT Neural Translation Server

Developer Edition

Руководство разработчика

PROMT Neural Translation Server Developer Edition

Руководство разработчика

Никакая часть настоящего документа не может быть воспроизведена без письменного разрешения компании PROMT (ООО «ПРОМТ»).

© 2003–2023, ООО «ПРОМТ». Все права защищены.

Ревизия 1.2

Россия, 199155,

Санкт-Петербург, Уральская ул., д. 17, лит. Е, кор. 3.

E-mail: common@promt.ru

support@promt.ru

Internet: <http://www.promt.ru>

<http://www.translate.ru>

Телефон/ Факс: +7 812 655-0350

PROMT®, ПРОМТ® — зарегистрированные торговые марки ООО «ПРОМТ».

Все остальные торговые марки являются собственностью соответствующих владельцев.

Оглавление

Глава 1. О документации.....	5
1.2. Условные обозначения.....	6
Глава 2. Введение.....	7
2.1. Введение в PROMT Neural Translation Server Developer Edition SDK.....	7
Глава 3. Веб-сервис PNTS.....	8
3.1. Формат данных.....	8
3.2. Обработка ошибок.....	8
3.3. Аутентификация.....	8
3.4. Методы сервиса.....	10
3.4.1. Перевод текста.....	12
3.4.2. Перевод через очередь перевода.....	16
3.4.3. Определение языка.....	18
3.4.4. Поиск в словаре.....	19
3.4.5. Информация о PNTS.....	20
3.4.6. Другие методы.....	22
3.4.7. Сервис аутентификации.....	23
Глава 4. Примеры вызова методов API.....	24
4.1. Утилита ApiConsole.....	24
4.2. Использование утилиты.....	24

Глава 1. О документации

Документация PROMT Neural Translation Server Developer Edition (PNTS) содержит:

- руководство пользователя, в котором описаны функции перевода как через веб-интерфейс, так и в любом Windows приложении с помощью специального модуля;
- руководство администратора, в котором описаны все функции PNTS и представлены сведения об установке, активации, настройке PNTS, а также администрировании пользователей;
 - ✳️ Руководство администратора предназначено для администраторов PNTS — пользователей, входящих в группу *Администраторы*.
- данное руководство разработчика, которое содержит описание методов веб-сервиса PNTS, а также примеры вызова этих методов.

1.2. Условные обозначения

В документации используются следующие условные обозначения:

-  — советы и рекомендации;
-  — важные замечания;
-  — примечания или дополнительная информация;
-  — последовательность действий, выполняемых пользователем;
-  — результат выполнения пользовательской задачи;
-  — ссылка на подробные сведения в руководстве или справке.

Глава 2. Введение

2.1. Введение в PROMT Neural Translation Server Developer Edition SDK

PROMT Neural Translation Server Developer Edition (PNTS DE) — это клиент-серверная система машинного перевода, предоставляющая программный интерфейс (API) для использования функций перевода из пользовательских приложений.

PROMT Neural Translation Server Developer Edition Software Developer Kit (PNTS SDK) является частью PNTS DE.

В состав PNTS SDK входят:

- Документация с описанием методов PNTS API, а также утилиты APIConsole
- Скрипты (.sh для Linux и .bat для Windows), позволяющие вызвать все методы API через приложение APIConsole
- Архив с исходным кодом утилиты APIConsole

 В данном руководстве не рассматриваются вопросы установки, настройки и функциональности системы PROMT Neural Translation Server Developer Edition, которые подробно описаны в руководствах администратора и пользователя.

Глава 3. Веб-сервис PNTS

PNTS поддерживает REST API, позволяющее программным путем получить доступ к основным возможностям системы PROMT Neural Translation Server Developer Edition.

Точки входа веб-сервисов:

Название сервиса	Относительная ссылка
Сервис перевода	/services/v1/rest.svc
Сервис аутентификации	/services/auth/rest.svc

3.1. Формат данных

На входе большинство методов поддерживает JSON, но есть методы, например, `TranslateFormatAsync`, которые принимают на входе бинарный контент.

На выходе для старых методов API PNTS23 поддерживается XML или JSON (в зависимости от поля `Ассерпт` в запросе), а для новых методов, которые появились в PNTS24, поддерживается только JSON. Такие методы имеют значок "new" в названии.

3.2. Обработка ошибок

При генерации исключения на стороне сервера, клиенту возвращается:

- HTTP status code: 500
- HTTP status: Internal Server Error

Тело ответа содержит XML или JSON (в зависимости от поля `Ассерпт` в запросе) с описанием ошибки.

3.3. Аутентификация

В PNTS поддерживаются следующие способы проверки подлинности (аутентификации пользователей):

- *Проверка подлинности Windows*
- *Проверка подлинности с помощью форм*

Способ аутентификации задается на сервере. Только один способ аутентификации может быть активным в текущий момент времени. Изменить текущую схему аутентификации можно в конфигурационном файле.

Использование способа *Проверка подлинности Windows* является более безопасным, но в то же время менее производительным, т.к. на каждый вызов веб-метода приходится несколько дополнительных вызовов между клиентом и сервером (согласно спецификации NTLM).

При использовании способа *Проверка подлинности с помощью форм* возможны два варианта взаимодействия с сервисом PNTS:

1. Аутентификация по ключу API
2. Аутентификация с помощью cookie

Аутентификация по ключу API

Ключ API — это строка, которая позволяет идентифицировать пользователя при вызове методов веб-сервиса PNTS. Если для пользователя создан ключ API, то нет необходимости передавать сервису пару логин-пароль.

При использовании аутентификации по ключу, в заголовки HTTP запроса к веб-сервису нужно добавить заголовок "PTSAPIKEY", значением которого должна быть строка ключа, полученная из диалога редактирования свойств пользователя, поле API-ключ.

 Получение ключа API возможно, только если установлена *Проверка подлинности с помощью форм*. Для одного пользователя может существовать только один ключ API.

 Создать новый ключ API для пользователя, либо обновить или удалить уже существующий может администратор PNTS в диалоге свойств пользователя веб-интерфейса PNTS.

Аутентификация с помощью cookie

Во втором, более сложном, варианте необходимо сначала получить cookie, которые будут содержать данные, позволяющие идентифицировать пользователя. Для этого предназначен специальный сервис аутентификации и его метод `Login`. Время жизни cookie либо не задано (*isPersistent = false*), либо равняется 7 дням (*isPersistent = true*). Кроме этого, поддерживается т.н. скользящий срок действия (*sliding expiration*): если клиент выполнил в течение 7 дней запрос к серверу, то ему возвращаются новые cookie, продленные еще на 7 дней.

3.4. Методы сервиса

Полный перечень методов API:

Перевод		
TranslateText	GET POST	text, from, to, profile
TranslateTextArray ^{new}	POST	texts, from, to, profile, format
TranslateFormattedText	GET POST	text, from, to, profile, format
TranslateFormattedArray	POST	texts, from, to, profile, format
TranslateTextWithED	GET POST	text, from, to, profile, removeLinebreaks
TranslateFormat	POST	data, from, to, profile

Перевод через очередь перевода		
TranslateFormatAsync	POST	data, from, to, profile
RemoveTranslateFormatAsync	GET	doc
RemoveAllTranslateFormatAsync	GET	-
GetTranslateFormatAsyncState	GET	doc
GetTranslateFormatAsyncResult	GET	doc

Определение языка		
DetectTextLanguage	GET POST	text

Поиск в словаре		
GetWordEDInfo	GET POST	word, from, to

Другие методы		
OCR	POST	data, from, to

Информация о PNTS

GetPtsInfo	GET	filter
GetPtsInfo2	GET	filter

Сервис аутентификации (REST)

Login	POST	username, password, ispersistent
IsLoggedIn	GET	-
Logout	GET	-
IsEnabled	GET	-

3.4.1. Перевод текста

TranslateText

Метод `TranslateText` позволяет получить перевод текста. Возвращает строку, содержащую перевод исходного текста.

Параметры	
text	Исходный текст для перевода
from	Префикс входного языка
to	Префикс языка перевода
profile	Имя или GUID профиля перевода
Запрос ⇒	
POST	<pre>/Services/v1/rest.svc/TranslateText { "text": "case", "from": "en", "to": "ru", "profile": "General Domain" }</pre>
GET	<pre>/Services/v1/rest.svc/TranslateText?text=case&from=en&to=ru &profile=General Domain</pre>

TranslateTextArray ^{new}

Метод `TranslateTextArray` позволяет получить перевод массива текстов. Возвращает массив, содержащий переводы исходных текстов.

Параметры	
texts	Массив текстов для перевода
from	Префикс входного языка
to	Префикс языка перевода
profile	Имя или GUID профиля перевода
Запрос ⇒	
POST	<pre>/Services/v1/rest.svc/TranslateTextArray { "texts": ["First text", "Second text"], "from": "ru", "to": "en", "profile": "67200000-0001-0002-b4ba- de876de98ec9" }</pre>

TranslateFormattedText

Метод `TranslateFormattedText` позволяет получить перевод строки заданного формата. Возвращает строку, содержащую перевод с сохранением исходного форматирования.

Параметры	
text	Исходный текст для перевода
from	Префикс входного языка
to	Префикс языка перевода
profile	Профиль перевода
format	Формат переводимого текста: text/plain; text/html; text/xml; text/xliff
Запрос ⇒	
POST	<pre>/Services/v1/rest.svc/TranslateFormattedText { "text": "<title>Some title</title>", "from": "en", "to": "ru", "profile": "General Domain", "format": "text/html" }</pre>
GET	<pre>/Services/v1/rest.svc/TranslateFormattedText?text=<title>So me%20title</title>&from=en&to=ru&profile=General Domain&format=text/html</pre>

TranslateFormattedArray

Метод `TranslateFormattedArray` позволяет получить перевод массива строк заданного формата. Возвращает массив строк, содержащий переводы с сохранением исходного форматирования.

Параметры	
texts	Массив текстов для перевода
from	Префикс входного языка
to	Префикс языка перевода
profile	Профиль перевода
format	Формат переводимого текста: text/plain; text/html; text/xml; text/xliff. По умолчанию – text/html.
Запрос ⇒	
POST	<pre>/Services/v1/rest.svc/TranslateFormattedArray { "texts": ["<title>first text</title>", "<title>and second text!<title>"], "from": "ru", "to": "en", "profile": "business", "format": "text/h</pre>

```
tml " }
```

TranslateTextWithED

Метод `TranslateTextWithED` позволяет получить перевод текста или краткую словарную справку, если слово (словосочетание) нашлось в электронном словаре. Возвращает строку, содержащую перевод или справку о слове.

Параметры	
text	Исходный текст для перевода
from	Префикс входного языка
to	Префикс языка перевода
profile	Имя или GUID профиля перевода
removeLinebreaks	Принимает значение <code>true</code> или <code>false</code> в зависимости от того, требуется ли удалять разрывы между строками одного абзаца
Запрос ⇒	
POST	<pre>/Services/v1/rest.svc/TranslateTextWithED { "text": "mother", "from": "en", "to": "ru", "profile": "67200000-0001-0002-b4ba-de876de98ec9", "removeLinebreaks": "true" }</pre>
GET	<pre>/Services/v1/rest.svc/TranslateTextWithED?text=mother&from=en&to=ru&profile=67200000-0001-0002-b4ba-de876de98ec9&removeLinebreaks=true</pre>

TranslateFormat

Метод `TranslateFormat` позволяет переводить документы различных форматов с сохранением форматирования. Возвращает результат перевода.

Параметры	
data	Банарные данные исходного файла
from	Префикс входного языка
to	Префикс языка перевода
profile	Профиль перевода
format	Формат переводимого текста: информацию по доступным форматам можно получить из GetPtsInfo
Запрос ⇒	

POST

```
/Services/v1/rest.svc/TranslateFormat?from=en&to=ru&profile  
= 67200000-0001-0002-b4ba-de876de98ec9&format=text/html
```

бинарные данные исходного файла

3.4.2. Перевод через очередь перевода

TranslateFormatAsync

Метод `TranslateFormatAsync` предназначен для асинхронного перевода документов большого размера, время перевода которых превышает несколько секунд. В этом случае документ обрабатывается отдельным процессом (очередью перевода). Возвращает идентификатор, который можно использовать для получения статуса и результата перевода с помощью методов `GetTranslateFormatAsyncState` и `GetTranslateFormatAsyncResult`.

Параметры	
data	Бинарные данные исходного файла
name	Имя документа
from	Префикс входного языка
to	Префикс языка перевода
profile	Профиль перевода
format	Формат переводимого текста: информацию по доступным форматам можно получить из GetPtsInfo
Запрос ⇒	
POST	<pre>/Services/v1/rest.svc/TranslateFormatAsync?from=en&to=ru&profile=67200000-0001-0002-b4ba-de876de98ec9&format=text/html</pre> <p><i>бинарные данные исходного файла</i></p>

RemoveTranslateFormatAsync

Метод `RemoveTranslateFormatAsync` предназначен для удаления документа с заданным идентификатором из очереди перевода, полученным при успешном выполнении метода [TranslateFormatAsync](#)

Параметры	
doc	Идентификатор документа
Запрос ⇒	
GET	<pre>/Services/v1/rest.svc/RemoveTranslateFormatAsync?doc=ZQAzAGYAMQAzADIAZQB1AC0AZQAxAGQANAAtADQAMgBhADUALQA5ADMAOQA4AC0A0AA1AGUAMgA5ADUAMAA1ADAAOAA1ADMAXwAwADgAZABiAGMAMAAxADAALQA4AGEAYQA5AC0ANwA2ADYANQAtAGYANQAzADQALQBhADcAZQAwAGEAZAA5ADUA0AA0ADgAYwA=</pre>

RemoveAllTranslateFormatAsync

Метод `RemoveAllTranslateFormatAsync` предназначен для удаления всех документов из очереди перевода

Запрос ⇒

GET	<code>/Services/v1/rest.svc/RemoveAllTranslateFormatAsync</code>
-----	--

GetTranslateFormatAsyncState

Метод `GetTranslateFormatAsyncState` предназначен для получения статуса асинхронного перевода документа. Возвращает строку, содержащую текущий статус перевода заданного документа.

Возможные значения:

- *Wait* — документ находится в состоянии ожидания начала перевода;
- *Translate* — документ переводится;
- *Stopped* — перевод прерван пользователем;
- *Complete* — перевод завершен;
- *CompleteWithErrors* — перевод завершен, но часть абзацев переведена с ошибками.

Параметры

doc	Идентификатор документа (см. TranslateFormatAsync)
-----	---

Запрос ⇒

GET	<code>/Services/v1/rest.svc/GetTranslateFormatAsyncState?doc=ZQAzAGYAMQAzADIAZQBlAC0AZQAxAGQANAAtADQAMgBhADUALQA5ADMAOQA4AC0AOAA1AGUAMgA5ADUAMAA1ADAAOAA1ADMAXwAwADgAZABiAGMAMAAxADAALQA4AGEAYQA5AC0ANwA2ADYANQAtAGYANQAzADQALQBhADcAZQAwAGEAZAA5ADUAOAA0ADgAYwA=</code>
-----	--

GetTranslateFormatAsyncResult

Метод `GetTranslateFormatAsyncResult` предназначен для получения результата асинхронного перевода документа. Возвращает результат перевода.

Параметры

doc	Идентификатор документа (см. TranslateFormatAsync)
-----	---

Запрос ⇒

GET	<code>/Services/v1/rest.svc/GetTranslateFormatAsyncResult?doc=OQBjAGYAMQA5ADkAMwA3AC0ANAA3AGMAYwAtADQANgBjADEALQA4ADEAZgA2AC0ANgA4ADcAZABjAGYAYwBmADQANQAwAGEAXwAwADgAZABiAGMAMAAyADAALQBhAGIAYQA0AC0AMgAzADYAYQAtAGUAOQBjADIALQA0AGMAOQAzADMANAAxAGUANgBjAGYAOQA=</code>
-----	---

3.4.3. Определение языка

DetectTextLanguage

Метод DetectTextLanguage позволяет определить язык входного текста. Возвращает префикс языка.

Параметры	
text	Текст для определения языка
Запрос ⇒	
POST	<pre>/Services/v1/rest.svc/DetectTextLanguage {"text": "mother"}</pre>
GET	<pre>/Services/v1/rest.svc/DetectTextLanguage?text=mother</pre>

3.4.4. Поиск в словаре

GetWordEDInfo

Метод GetWordEDInfo позволяет получить информацию о заданном слове из электронного словаря в формате XML.

Параметры	
word	Слово для поиска в электронном словаре
from	Префикс входного языка
to	Префикс языка перевода
Запрос =>	
POST	<pre>/Services/v1/rest.svc/GetWordEDInfo { "word": "mother", "from": "en", "to": "ru" }</pre>
GET	<pre>/Services/v1/rest.svc/GetWordEDInfo?word=mother&from=en&to=ru</pre>

3.4.5. Информация о PNTS

GetPtsInfo

Метод `GetPtsInfo` позволяет получить информацию о конвертерах, словарях, языковых парах, базах ТМ, профиле пользователя, профилях перевода. Возвращает информацию, выбранную по заданному фильтру.

Возможные значения параметра `filter`:

- *converters* — информация о конвертерах;
- *dictionaries* — информация о словарях;
- *directions* — информация о языковых парах;
- *TM* — информация о базах ТМ;
- *profile* — информация о профиле пользователя;
- *topics* — информация о профилях перевода.

Значения можно задавать одной строкой через запятую. Если строка пустая, то возвращаются все значения.

Параметры	
filter	Фильтр поиска
Запрос ⇒	
GET	<code>/Services/v1/rest.svc/GetPtsInfo?filter=directions</code>

GetPtsInfo2

Метод `GetPtsInfo2` позволяет получить информацию о конвертерах, словарях, языковых парах, базах ТМ, профиле пользователя, профилях перевода. Возвращает информацию, выбранную по заданному фильтру. В отличие от метода `GetPtsInfo`, возвращает информацию в более общем виде (например, для фильтра *directions* все данные возвращаются в строковом формате).

Возможные значения параметра `filter`:

- *converters* — информация о конвертерах;
- *dictionaries* — информация о словарях;
- *directions* — информация о языковых парах;
- *TM* — информация о базах ТМ;
- *profile* — информация о профиле пользователя;
- *topics* — информация о профилях перевода.

Значения можно задавать одной строкой через запятую. Если строка пустая, процедура возвращает все значения.

Параметры	
filter	Фильтр поиска
Запрос ⇒	

GET

/Services/v1/rest.svc/GetPtsInfo2?filter=directions

3.4.6. Другие методы

OCR

Метод OCR позволяет распознать текст на картинке. Результатом метода является строка с распознанным текстом.

Параметры	
data	Бинарные данные картинки
from	Префикс входного языка
to	Префикс языка перевода
Запрос ⇒	
POST	<pre>/Services/v1/rest.svc/OCR {"from":"en", "to":"ru", "data":<i>бинарные данные</i>}</pre>

3.4.7. Сервис аутентификации

Login

Метод `Login` проверяет учетные данные пользователя и создает файл `cookie`, если учетные данные оказываются действительными.

Параметры	
<code>username</code>	Имя пользователя
<code>password</code>	Пароль
<code>ispersistent</code>	Определяет, остается ли действительным билет проверки подлинности в различных сеансах (<code>true/false</code>)
Запрос ⇒	
POST	<code>/Services/auth/rest.svc/Login</code> <pre>{"username": "...", "password": "...", "isPersistent": "true"}</pre>

IsLoggedIn

Метод `IsLoggedIn` позволяет определить, прошел ли текущий пользователь проверку подлинности. Возвращает значение `true` или `false`. С запросом необходимо передать файл `cookie`, полученный при вызове метода `Login`.

Запрос ⇒	
GET	<code>/Services/auth/rest.svc/IsLoggedIn</code>

Logout

Метод `Logout` очищает файл `cookie` в браузере.

Запрос ⇒	
GET	<code>/Services/auth/rest.svc/Logout</code>

IsEnabled

Метод `IsEnabled` возвращает значение `true` или `false` в зависимости от того, включена ли проверка подлинности с помощью форм.

Запрос ⇒	
GET	<code>/Services/auth/rest.svc/IsEnabled</code>

Глава 4. Примеры вызова методов API

4.1 Утилита ApiConsole

В состав PNTS24 включена консольная утилита ApiConsole, позволяющая вызвать все методы API через интерфейс командной строки.

Кроме бинарных модулей самой утилиты, в состав продукта включены командные файлы для Windows и скрипты для Linux - по одному файлу на каждый метод API.

В набор также входит архив с исходным кодом утилиты на с#. Для сборки утилиты разработчику понадобится Microsoft Visual Studio 2022.

Местоположение данных относительно корневой папки продукта:

\ApiConsole	Утилита ApiCosole
\Scripts\Samples	Примеры использования утилиты для Windows в виде bat-файлов и примеры использования утилиты для Linux в виде sh-скриптов
\Scripts\ApiConsole.zip	Архив исходного кода утилиты

4.2 Использование утилиты

В каждом скрипте прописаны параметры, которые необходимо настроить перед запуском утилиты.

Параметры, общие для всех скриптов:

- *--url* – строка подключения к PNTS, например <http://localhost/pnts>. Точку входа веб-сервиса указывать не надо.
- *--method* – имя метода API
- *--cookie_file* – (опционально) имя файла для хранения содержимого cookie между запросами. *cookie_file* требуется задать, если в PNTS включена forms-аутентификация, и запросы выполняются под аутентифицированным пользователем. Для аутентификации сначала вызывается метод Login.
- *--apikey* – (опционально) строка, содержащая ключ, позволяющий аутентифицировать пользователя без cookie и без вызова метода Login. Может использоваться только, если в PNTS включена forms-аутентификация.
- *--accept* – (опционально) формат выходных данных. Результат запросов от сервера по умолчанию возвращается в формате JSON. Для получения результата в формате XML надо указать значение "xml".

В тех случаях, когда метод API поддерживает как GET, так и POST, утилита использует POST.

Ниже приводится таблица с информацией о параметрах каждого скрипта.

<i>Метод API</i>	Параметры ApiConsole	Имя командного файла для Windows / скрипта для Linux
DetectTextLanguage	<pre>--url --method --cookie_file --apikey --text</pre>	<pre>detecttextlanguage.bat/ detecttextlanguage.sh</pre>
GetPtsInfo	<pre>--url --method --cookie_file --apikey --filter</pre>	<pre>getptsinfo.bat / getptsinfo.sh/</pre>
GetPtsInfo2	<pre>--url --method --cookie_file --apikey --filter</pre>	<pre>getptsinfo2.bat / getptsinfo2.sh</pre>
GetTranslateFormatAsyncResult	<pre>--url --method --cookie_file --apikey</pre>	<pre>gettranslateformatasyncresult.bat / gettranslateformatasyncresult.sh</pre>
GetTranslateFormatAsyncState	<pre>--url --method --cookie_file --apikey --doc</pre>	<pre>gettranslateformatasyncstate.bat / gettranslateformatasyncstate.sh</pre>
GetWordEDInfo	<pre>--url --method --cookie_file --apikey --from --to --word</pre>	<pre>getwordedinfo.bat / getwordedinfo.sh</pre>
IsEnabled	<pre>--url --method --cookie_file --apikey --url</pre>	<pre>isenabled.bat / isenabled.sh</pre>

IsLoggedIn	<i>--method</i> <i>--cookie_file</i> <i>--apikey</i>	isloggedin.bat / isloggedin.sh
Login	<i>--url</i> <i>--method</i> <i>--cookie_file</i> <i>--apikey</i> <i>--username</i> <i>--password</i>	login.bat / login.sh
Logout	<i>--url</i> <i>--method</i> <i>--cookie_file</i> <i>--apikey</i>	logout.bat / logout.sh
OCR	<i>--url</i> <i>--method</i> <i>--cookie_file</i> <i>--apikey</i> <i>--from</i> <i>--to</i> <i>--srcfile</i>	ocr.bat / ocr.sh
RemoveAllTranslateFormatAsync	<i>--url</i> <i>--method</i> <i>--cookie_file</i> <i>--apikey</i>	removealltranslateformatasync.bat / removealltranslateformatasync.sh
RemoveTranslateFormatAsync	<i>--url</i> <i>--method</i> <i>--cookie_file</i> <i>--apikey</i> <i>--doc</i>	removetranslateformatasync.bat / removetranslateformatasync.sh
TranslateFormat	<i>--url</i> <i>--method</i> <i>--cookie_file</i> <i>--apikey</i> <i>--from</i> <i>--to</i> <i>--profile</i> <i>--format</i> <i>--srcfile</i> <i>--resfile</i>	translateformat.bat / translateformat.sh
TranslateFormatAsync	<i>--url</i> <i>--method</i> <i>--cookie_file</i> <i>--apikey</i> <i>--from</i> <i>--to</i> <i>--profile</i> <i>--format</i> <i>--srcfile</i>	translateformatasync.bat / translateformatasync.sh

<u>TranslateFormattedArray</u>	<pre> --url --method --cookie_file --apikey --from --to --profile --format --texts </pre>	<pre> edarray.bat / translateformattedarray.sh </pre>
<u>TranslateFormattedText</u>	<pre> --url --method --cookie_file --apikey --from --to --profile --format --text </pre>	<pre> translateformattedtext.bat / translateformattedtext.sh </pre>
<u>TranslateText</u>	<pre> --url --method --cookie_file --apikey --from --to --profile --text </pre>	<pre> translatetext.bat / translatetext.sh </pre>
<u>TranslateTextArray</u>	<pre> --url --method --cookie_file --apikey --from --to --profile --texts </pre>	<pre> translatetextarray.bat / translatetextarray.sh </pre>
<u>TranslateTextWithED</u>	<pre> --url --method --cookie_file --apikey --from --to --profile --removeLineBreaks --text </pre>	<pre> translatetextwithed.bat / translatetextwithed.sh </pre>