PROMT Analyzer SDK

Документация, содержащая описание функциональных характеристик программного обеспечения «PROMT Analyzer SDK» и информацию, необходимую для установки и эксплуатации программного обеспечения

PROMT Analyzer SDK

Документация, содержащая описание функциональных характеристик программного обеспечения «PROMT Analyzer SDK» и информацию, необходимую для установки и эксплуатации программного обеспечения

Россия, 199155,

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

E-mail: common@promt.ru

support@promt.ru

Internet: https://www.promt.ru

https://www.translate.ru

Телефон: +7 812 655-0350

Факс: +7 812 655-0021

Оглавление

Введение

PROMT Analyzer SDK— это масштабируемая клиент-серверная система, предназначенная для анализа текстов на естественных языках с целью поиска, извлечения и обобщения информации о сущностях, фактах, событиях и их связях, путем лингвистического анализа соответствующих текстов с учетом синтаксиса и семантики.

Состав документации

Данный документ содержит описание PROMT Analyzer SDK.
В описание включены: функциональность, архитектура PROMT Analyzer SDK, системные требования, процесс установки, настройки, описание веб-сервиса и работы в нем.

Назначение и основные возможности

Функциональное назначение программы

PROMT Analyzer SDK разработан для использования в различных информационных и аналитических системах, предназначенных для сбора и систематизации разнородной информации, построения аналитических отчетов, в тех случаях, когда одним из объектов анализа являются текстовые документы. Использование PROMT Analyzer SDK предполагает обращение к его функциям через кросс-платформенный программный интерфейс, реализованный как веб-сервис.

В основе PROMT Analyzer SDK лежит глубокий синтактико-семантический разбор предложения, как единицы языка, текста, как единого комплекса составляющих его предложений, а также наличие словарной базы, содержащей слова и выражения с приписанными им синтаксическими и семантическими признаками.

Эксплуатационное назначение программы

PROMT Analyzer SDK – клиент серверный программный продукт, предназначенный для установки и использования в ОС Linux (список целевых операционных систем в разделе Системные требования), для решения следующих задач:

- анализ новостных текстов с целью поиска такой информации, как сущности, факты, события и их связи, определения тональности;
- анализ текстов договоров различных типов с целью поиска такой информации, как данные о сторонах договора, предмете договора, дате подписания, месте подписания, стоимости и пр;
- анализ текстов резюме с целью поиска такой информации, как данные о кандидате, дате его рождения, семейном положении, контактах, местах, где он работал и учился, и пр.

Уровень подготовки пользователя

Квалификация персонала должна обеспечивать эффективное функционирование технических и программных средств программы PROMT Analyzer SDK

Термины и сокращения

Термин	Описание
Языковая пара	Определяет, с какого языка и на какой язык будет переведен текст. В некоторых случаях (имена параметров и т.д.) может использоваться термин “направление перевода” или просто “направление”.
Профиль перевода	Набор лингвистических настроек, которые модуль перевода использует для повышения качества перевода в конкретной тематической области. В некоторых случаях может

использоваться термин “тематика”.

Архитектура PROMT Analyzer SDK

PROMT Analyzer SDK - набор программных модулей, позволяющих получить анализ текста на Unix-подобных операционных системах в среде интранет. PROMT Analyzer SDK предоставляет HTTP(S) веб-сервис с простым REST API. Кроме удаленного доступа по HTTP протоколу, клиенты могут использовать локальный API для создания утилит командной строки. PROMT Analyzer SDK поддерживает горизонтальное масштабирование за счет возможности распределения нагрузки на другие сервера перевода.

С точки зрения администратора, PROMT Analyzer SDK содержит следующие основные компоненты:

- Лингвистические данные
- Ядро перевода
- Балансировщик нагрузки
- Службы (демоны)
- Веб-сервис

Лингвистические данные

Лингвистические данные - это набор словарей, баз TM, слов, которые не требуется переводить, и других настроек. Словари хранятся в виде файлов, в проприетарном формате, совместимом с версией PTS для Windows. Базы TM используют формат открытой системы управления базой данных - SQLite. Остальные настройки хранятся в текстовом виде в конфигурационных файлах. Все эти данные должны располагаться на одной машине с ядром перевода.

Ядро перевода

Главный компонент в архитектуре PROMT Analyzer SDK - это ядро перевода. Оно предоставляет API для анализа текста и управления лингвистическими данными. Все модули ядра являются совместно используемыми библиотеками.
При загрузке, ядро перевода читает конфигурационный файл и определяет доступные лингвистические модули и данные. Само ядро работает как диспетчер для передачи запросов между модулями. Вся обработка делается модулями.
После получения запроса, ядро создает и инициализирует объект, который содержит информацию о запросе:

- тип запроса
- направление обработки запроса
- параметры запроса
- результат запроса

Ядро загружает лингвистические модули (и, таким образом, создает конвейер для обработки запросов) в порядке, определенном конфигурацией. Запрос проходит через конвейер в обоих направлениях, сначала вперед (от первого модуля до последнего), и затем назад. Каждый модуль получает объект запроса и анализирует его атрибуты.

Балансировщик нагрузки

Балансировщик нагрузки представляет собой промежуточный модуль между веб-сервером и ядром перевода. Он служит для распределения нагрузки между серверами перевода PROMT Analyzer SDK, разрешая таким образом проблему горизонтальной масштабируемости на уровне отдельных запросов веб-сервиса. Список серверов перевода задается в конфигурационном файле и считывается в момент запуска балансировщика. Балансировщик нагрузки используется как единая точка входа для веб-сервиса PROMT Analyzer SDK, в то время как серверов перевода может быть более одного.

Службы (демоны)

В состав PROMT Analyzer SDK входят следующие процессы, запускаемые в режиме демонов:

- **nginx (служба promtanalyzer21-nginx)** – автономная версия Nginx, которая устанавливается вместе с PROMT Analyzer SDK и не имеет системных зависимостей. Служба может конфликтовать с установленным системным веб-сервером (из-за использования одного TCP порта), поэтому рекомендуется отключать или удалять системную службу перед установкой PROMT Analyzer SDK.
- **transfcgid.run (служба promtanalyzer21-balancer, режим балансировки)** – серверный процесс, который взаимодействует с веб-сервером (nginx) по протоколу HTTP в качестве прокси сервера и перенаправляет запросы на другие сервера перевода. Балансировщик решает следующие задачи:
    - Балансирует нагрузку между серверами перевода
    - Реализует простую схему масштабируемости с возможностью наращивания мощностей перевода
    - Используется в качестве единой точки входа веб-сервиса (сервис доступен для вызова извне для клиентов PROMT Analyzer SDK)
- **transfcgid.run (служба promtanalyzer21-analyzer , режим анализа)** – серверный процесс, который взаимодействует с веб-сервером (nginx) по протоколу FastCGI. Сервер перевода решает следующие задачи:
    - Реализует методы веб-сервиса (клиенты PROMT Analyzer SDK не могут вызывать его напрямую, это может делать только балансировщик нагрузки)
    - Запускает дочерние процессы, которые выполняют анализ текста
    - Балансирует нагрузку между дочерними процессами
    - Реализует простую схему отказоустойчивости, когда дочерний процесс завершается с ошибкой или зависает

  Мастер-процесс запускается пользователем, дочерние процессы управляются мастер-процессом
- **dcs.run (служба promtanalyzer21-dcs)** - служба управления данными, обеспечивает доступ к Translation Memory. **dcs** использует базу данных SQLite для хранения сегментов Translation Memory. Запускается пользователем.
- **Promt.Host.exe (служба promtanalyzer21-managed) -** серверный мастер-процесс для решения некоторых вспомогательных задач. Процесс запускается с помощью mono. **Promt.Host.exe** управляет дочерними процессами - **Promt.Unit.exe**, таким образом реализуется отказоустойчивость. Мастер-процесс запускается пользователем, дочерние процессы управляются мастер-процессом.

Обработка запроса

Типичная схема обработки запроса на анализ текста:

1. Клиент отправляет запрос на анализ текста, используя для передачи HTTP-протокол.
2. Web сервер (Nginx) получает запрос и определяет, что запрос требуется передать для обработки процессу `transfcgid` по HTTP протоколу для дальнейшей балансировки между серверами перевода.
3. Процесс `transfcgid` находит наименее загруженный сервер перевода и пересылает запрос ему (этим сервером может быть в том числе и тот, на котором запущен балансировщик).
4. Web сервер (Nginx) получает запрос и определяет, что запрос требуется передать для обработки процессу `transfcgid` по FastCGI-протоколу для дальнейшего обработки ядром перевода.
5. Процесс `transfcgid` находит наименее загруженный дочерний процесс и пересылает запрос ему.
6. Дочерний процесс `transfcgid` осуществляет анализ текста.
7. `Promt.Host.exe` определяет дочерний процесс и вызывает процесс `Promt.Unit.exe`.

Установка и удаление

Системные требования

Требования к аппаратным средствам

Минимальные системные требования к набору:

1. Dual-core CPU

2. 4 GB RAM
3. 500 Мбайт свободного пространства (лингвистические данные пользователя могут занять дополнительное место)

Требования к программному обеспечению

PROMT Analyzer SDK устанавливается с помощью собственного инсталлятора и был протестирован со следующими ОС:

- Ubuntu 18.04.4
- Ubuntu 19.10
- Ubuntu 20.04
- Debian 10
- AstraLinux CE 2.12
- AstraLinux SE 1.6

Какой-либо жесткой привязки PROMT Analyzer SDK к определенному дистрибутиву ОС не существует, вместо этого используются маркеры совместимости. На текущий момент это:

- Наличие системной библиотеки GLIBC версии 2.17 или выше
- Наличие системных библиотек libgcc_s.so.1 и libstdc++.so.6
- Наличие менеджера системных служб systemctl

Программное обеспечение Nginx и mono, которое используется при работе PROMT Analyzer SDK, входит в состав дистрибутива и не требует дополнительной установки пакетов.

Установка PROMT Analyzer SDK

В общем случае для получения работающего сервера PROMT Analyzer SDK требуется только запустить инсталлятор и следовать инструкциям. Для работы PROMT Analyzer SDK требуется сгенерированный для данного компьютера файл лицензии. Для получения файла лицензии выполните следующее:

1. Получите уникальный идентификатор компьютера `HardwareId`.
2. Передайте полученный `HardwareId` в службу поддержки компании ПРОМТ для получения файла лицензии.
3. Установите набор, указав путь к полученному файлу лицензии.

Получение идентификатора компьютера

Для получения идентификатора компьютера до установки набора можно воспользоваться следующей командой (предполагается, что команда запускается из каталога с run-файлом):

Например, если установочный файл называется `promt-analyzer21.x.run`, где x - целое число, запустите run-файл с помощью команды:

```
chmod +x promt-analyzer21.x.run && ./promt-analyzer21.x.run -i
```

При этом в консоль будет выведено сообщение вида:

Current hardware id: Y27wF82sEHHo25v7DUMqUMttqRhmALqzXt99qke7bZk=

Текст после “Current hardware id:” (идентификатор, закодированный в base64-строку) необходимо скопировать и переслать службе поддержки компании ПРОМТ.

Установка основного набора

PROMT Analyzer SDK распространяется в виде файла с расширением .run, который представляет собой модифицированный 7z SFX архив с основным инсталлятором.

Например, если установочный файл называется `promt-analyzer21.x.run`, где x - целое число, запустите run-файл с помощью команды:

```
chmod +x promt-analyzer21.x.run && sudo ./promt-analyzer21.x.run
```

Инсталлятор автоматически найдет файл лицензии, если он находится в одной папке с инсталлятором. Файл лицензии можно также указать в процессе установки, либо передать с помощью ключа:

```
sudo ./promt-analyzer21.x.run -k file.lic
```

Работа PROMT Analyzer SDK без файла лицензии невозможна. Для установки лицензий в уже установленный набор используется скрипт `update-license.sh`. Пример запуска:

```
sudo /usr/local/promtanalyzer21/bin64/update-license.sh file.lic
```

В процессе установки инсталлятор выполняет следующие действия:

1. Проверка параметров системы. Система будет проверена на совместимость с продуктом.
2. Распаковка данных. Программные модули и лингвистические данные будут распакованы в корневую папку продукта. Настройки будут модифицированы под текущее окружение ОС.
3. Установка файла лицензии. Файл будет скопирован из указанного места в корневую папку продукта.
4. Настройка фаервола. Инсталлятор попытается найти фаервол и разрешить входящие соединения на порт 80. Если порт 80 занят, пользователь получает возможность указать другой номер порта.
5. Создание и запуск системных служб.

Во время установки инсталлятор может задать следующие вопросы:

- Подтверждение принятия лицензии (y/n)
- Подтверждение пути установки (ввод значения или “enter” для значения по умолчанию)
- Подтверждение пути файла лицензии (ввод значения или “enter” для принятия пути найденного лицензионного файла)
- Подтверждение создания нового пользователя (ввод значения или “enter” для значения по умолчанию)

На системах с активной защитой SELinux (например, семейство RHEL) необходимо убедиться, что указанная корневая папка продукта принадлежит директории /usr/. Рекомендуется использовать путь по умолчанию.

Ключи командной строки

Для установки в неинтерактивном режиме существует способ запуска инсталлятора с ключом “-y”, в этом случае будут использованы значения по умолчанию (или заданные с помощью ключей). Список других ключей инсталлятора:

- -h, --help: вывод справки
- -i, --id: вывод идентификатора оборудования (hardware id), который используется для генерации файла лицензии.
- -e, --extract [PATH]: распаковка содержимого архива в указанную директорию без запуска скрипта установки и без последующего удаления. Создает директорию, если ее не существует.

- -y, --yes: ответ на все запросы в автоматическом режиме. Для диалогов «y/n» - будет выбрано «y», для диалогов с вводом – значение по умолчанию, если это возможно.
- -k, --key [PATH]: путь файла лицензии PROMT Analyzer SDK.
- -p, --path [PATH]: путь установки PROMT Analyzer SDK.

Особенности установки

По-умолчанию PROMT Analyzer SDK устанавливается с доступом только по HTTP протоколу. Доступ по HTTPS протоколу требует дополнительной настройки сервера Nginx и не описан в данном документе.

Примечание: Лицензионный файл в %RootFolder% называется ptsu.lic.

Тестирование работоспособности

Для проверки работы веб-сервиса, вы можете выполнить следующую команду:

```
curl
'http://localhost/AnalyzerSDK/service/Analyze?text=PROMT&language=en&profile=News' && echo
```

Включение логирования

Поддержка логирования для сервиса promtanalyzer21-managed

В `Promt.Host.exe` и `Promt.Unit.exe` добавлена поддержка логирования. Для включения логирования необходимо изменить файл сервиса (`/etc/systemd/system/promtanalyzer21-managed.service`) и добавить ключ "-v" в строку:

```
ExecStart=/usr/local/promtanalyzer21/mono/mono
/usr/local/promtanalyzer21/bin64/Promt.Host.exe -v
```

Выполните команды:

```
sudo systemctl daemon-reload
sudo systemctl restart promtanalyzer21-managed
```

Лог файлы будут доступны в каталоге: `/usr/local/promtanalyzer21/log`

Для `Promt.Host.exe` - `managed-host.log`

Улучшение логирования для promtanalyzer21-analyzer/promtanalyzer21-balancer

В `transfcgid` увеличено количество логируемых мест и изменен режим записи в файл лога – теперь новые записи добавляются к существующему файлу, а не перезаписывают его.

Для активации режима логирования на уровне сервисов, необходимо добавить ключ "–v" в аргументы командной строки при запуске `transfcgid` в файлах сервисов `promtanalyzer21-analyzer` и `promtanalyzer21-balancer` (см. раздел "Поддержка логирования для сервиса promtanalyzer21-managed").

В каталоге `/usr/local/promtanalyzer21/log` будут созданы:

`analyzer.trace` – для `promtanalyzer21-analyzer`
`balancer.trace` – для `promtanalyzer21-balancer`

Удаление PROMT Analyzer SDK

Удаление продукта осуществляется посредством запуска скрипта “uninstall.sh” из папки “bin64” в корневой папке продукта. При запуске скрипт проверит, запущен ли он с правами суперпользователя, а также спросит явное разрешение на удаление продукта у пользователя. Скрипт полностью удаляет PROMT Analyzer SDK из системы, в том числе: системные сервисы и корневую папку продукта.

Управление службами

По умолчанию при установке служба PROMT Analyzer SDK автоматически запускается и настраивается на запуск при загрузке системы, однако в ряде случаев может возникнуть необходимость выполнения ручных операций (например, перезапуска службы). Служба PROMT Analyzer SDK реализована в виде нескольких сервисов (имена соответствуют именам файлов описания, расположенных в `/etc/systemd/system`):

- *promtanalyzer21-nginx.service* — запускает nginx, который входит в состав PROMT Analyzer SDK.
- *promtanalyzer21-balancer.service* — запускает transfcgid в режиме балансировщика.
- *promtanalyzer21-analyzer.service* — запускает transfcgid в режиме анализатора
- *promtanalyzer21-dcs.service* — запускает процесс DCS
- *promtanalyzer21-managed.service* — запускает процесс анализа

Для выполнения операций со службой и ее компонентами используется команда вида:

```
sudo systemctl <операция> <имя_сервиса>
```

где <*имя_сервиса*> — имя из списка выше (суффикс *.service* можно опускать), <*операция*> — название операции (start, stop, restart, status).

Балансировка нагрузки

Общее описание

По-умолчанию PROMT Analyzer SDK устанавливается в конфигурации, где балансировщик нагрузки располагается на том же компьютере, что и сервер перевода. Однако PROMT Analyzer SDK поддерживает горизонтальное масштабирование производительности за счет возможности распределения нагрузки на другие сервера. Для этого требуются модификации следующих параметров конфигурационного файла (/usr/local/promtanalyzer21/promtkernel.conf) PROMT Analyzer SDK:

В секции General:

- **Threads** – количество потоков, выделенное для анализа на текущем сервере. Значение по-умолчанию 0 означает количество потоков равное количеству ядер в процессоре.

В секции Balancer:

- **Threads** – количество потоков, выделенное для балансировки на текущем сервере. Значение по-умолчанию 0 означает количество потоков равное количеству ядер в процессоре.
- Список серверов перевода в формате “**TSX=127.0.0.1;N**”, где X – это порядковый номер сервера перевода (нумерация начинается с единицы, т.е. TS1 – это первый сервер), N – вес сервера (производительность в сравнении с остальными).

При задании этих параметров нужно исходить из общих параметров кластера серверов перевода. К примеру планируется создание кластера из 3 компьютеров, на одном из которых будет находиться балансировщик. Два из этих компьютеров имеет по 4 процессорных ядра, а третий – 16 ядер (для простоты будем считать, что производительность самих ядер и остальные параметры системы одинаковы). В таком случае рекомендуется следующая конфигурация балансировщика (некоторые параметры пропущены):

```
[General]
Threads=2

[Balancer]
Threads=50
TS1=127.0.0.1;2;
TS2=192.168.0.100;4;
TS3=192.168.0.101;16;
```

После изменения конфигурации требуется перезапустить сервисы PROMT Analyzer SDK на компьютере, где запущен балансировщик. На всех серверах перевода должны быть развернуты одинаковые экземпляры PROMT Analyzer SDK, лицензия должна быть установлена только на сервере с балансировщиком.

В такой конфигурации балансировщик будет работать на 50 одновременных входящих подключений (остальные будут ожидать в очереди обработки) и распределять нагрузку на 20 суммарных ядер перевода на 3 серверах (один из которых – он сам). На сервере с балансировщиком для анализа будут доступны только 2 потока из 4. Сами потоки балансировщика не вызывают значительной нагрузки на процессор, однако рекомендуется иметь хотя бы два свободных ядра, чтобы не возникло ситуации, когда потоки анализа будут конкурировать с потоками балансировщика и тем самым снижать суммарную производительность кластера. При дальнейшем увеличении количества серверов перевода имеет смысл вообще исключить компьютер с балансировщиком из списка серверов перевода.

Количество потоков балансировщика рекомендуется задавать большим, чем суммарное значение ядер перевода из-за возможной неравномерности запросов на анализ, что может приводить к ситуации, когда один сервер будет нагружен на 100%, а остальные будут простаивать при неполной нагрузке. Веса серверов перевода следует выставлять пропорционально производительности каждого сервера, при прочих равных условиях таким числом можно считать количество ядер процессора (в реальности будут также иметь значение тип и частота оперативной памяти, производительность самого процессора, загруженность сервера другими задачами и т.п.)

Важно: Все сервера, входящие в кластер, должны иметь одинаковый набор лингвистических данных.

Балансировка отдельных направлений

В PROMT Analyzer SDK существует возможность задания направлений перевода для каждого сервера в отдельности. Это может быть полезно для более тонкой настройки распределения нагрузки между серверами. Настройка осуществляется путем задания списка направлений перевода в конфигурационном файле:

```
[Balancer]
Threads=50
TS1=127.0.0.1;er,re;
TS2=192.168.0.100;fe,ge;
```

В данном примере TS1 будет переводить только ER и RE направления, TS2 – только FE и GE. В случае отсутствия направлений считается, что сервер поддерживает все доступные направления.

Анализ текста

Под анализом текста понимается алгоритм извлечения сущностей и определения тональности текста. На текущий момент поддержка реализована в виде отдельного API метода “Analyze”, который имеет следующие параметры:

Параметр	Описание
text	Входной текст для разбора

language	Префикс языка анализируемого текста
profile	Профиль настроек для разбора текста
localization	Язык локализации свойств сущностей

Список языков определяется лицензионным файлом, поддерживаются следующие префиксы: RU, EN, DE, ES, PT, IT, FR.
Список доступных профилей зависит от комплектации продукта, например:
ER:

- Информационные сообщения (News) — для анализа новостных текстов.

RE:

- Информационные сообщения (News) — для анализа новостных текстов;
- Документы (Delivery contract) — для анализа текстов договоров различных типов;
- Резюме (Curriculum Vitae) — для анализа текстов резюме.

В случае любого другого направления текст сначала переводится на английский язык, а потом используется ER направление.

Веб-сервис

В данном разделе описывается Application Programming Interface (API) PROMT Analyzer SDK. API реализован как веб-сервис.
В документе описаны методы веб-сервиса для извлечения сущностей из текста.

```
Формат запросов (POST):
Content-Type: application/x-www-form-urlencoded

Выходной формат: JSON
```

Если в процессе обработки запроса на сервере произошла ошибка, то код ответа от сервера будет равен 500, а тело ответа будет содержать описание ошибки.

Общие сведения о сущности

Анализатор получает на вход фрагмент текста, анализирует его и выделяет из текста следующие именованные сущности:

- персоны (имена собственные), т.е. упоминания конкретных людей с фамилиями, именами и отчествами, если они находятся в тексте. Кроме этого, для найденных персон система выделяет ряд атрибутов, если информация о них присутствует в тексте (должность, профессия, родственные связи, партийная принадлежность, национальность и т.п.)
- названия организаций – имена собственные. Для найденных организаций выделяется тип (ОАО, ООО и т.п.) и подтип организаций (производственная организация, банк и т.п.), если информация о них присутствует в тексте или в онтологиях системы
- географические названия с указанием типа (страна, город и т.п.) и подтипа (часть света, водоем и т.п.), если информация о типе и подтипе имеется в онтологиях системы
- геополитические сущности – географические названия в организационном контексте (Москва заявила, требование Анкары) с указанием типа и подтипа, если информация о типе и подтипе имеется в онтологиях системы.
- другие именованные сущности разных типов, найденные при разборе текста (события, устройства, документы и т.д.) с указанием типа и подтипа, если информация о типе и подтипе имеется в онтологиях системы.

Для каждой сущности система анализирует и выделяет факты - действия и события, в которых зафиксировано участие сущности. Для каждого выделенного факта строится фрейм, описывающий действие (кто, что, где, когда и т.п.) или событие с выводом других найденных сущностей, являющихся актантами в этом действии.
Пример:

Кто	пресс-секретарь Дмитрий Песков	персона, собственное
Действие	сказал	
Цитата	подобные угрозы не могут повлиять на последовательную политику России и президента Путина	высказывание с отрицательной тональностью
Кому	господину Франсуа Олланду	персона, собственное
Где	в Москве	география, собственное
Когда	Вчера	

Свойства сущности

Сущность (entity) описывается следующим набором свойств (в скобках даны идентификаторы свойств для JSON):

- **Идентификатор сущности** (id)

Используется для ссылки на сущность. Идентификаторы сущностей уникальны для сущностей одного разбора

```
{
	"id": 0
}
```

- **Имя сущности** (name)

```
{
	"name": "Дмитрий Песков"
}
```

- **Тип сущности** (type)

```
{
	"type": "персона"
}
```

- **Подтип сущности** (subtype)

Подтип зависит от типа сущности. В подтип заносится подтип 1 уровня и подтип 2 уровня, разделенные запятой.

```
{
	"subtype": "город, столица"
}
```

- **Атрибуты сущности** (attributes)

Массив элементов. Для найденных сущностей система выделяет ряд атрибутов. Атрибуты – это неименованные сущности, которые при разборе были связаны с описываемой именованной сущностью, а именно являются ее предикатом или приложением к ней. Значением типа атрибута является подтип неименованной сущности 1 уровня (см. таблицу выше), а для организаций будет написано, что тип “организация”

Примеры атрибутов:

```
{
	"type": "должность",
	"name": "пресс-секретарь"
}
```

- **Число ссылок** (count)

Количество упоминаний сущности во входном тексте.

```
{
      "count": 3
}
```

- **Список ссылок на позиции в тексте** (references)

Массив элементов. Список ссылок описывает позиции в тексте всех упоминаний сущности и имеет следующие параметры:

- позиция в тексте (position)
- длина в символах (length)

```
{
      "position": 0,
      "length": 23,
}
```

- **Части имени** (personnames)

Массив элементов. Имя персоны разбивается на части. Каждая часть имеет следующие поля:

- имя (name)
- тип (type).

Тип (type) может принимать следующие значения:

- Имя персоны (NamePart)
- Отчество (PatronimicPart)
- Фамилия (SurnamePart)

```
"personnames": [
      {
        "name": "Дмитрий",
        "type": "NamePart"
      },
      {
        "name": "Песков",
        "type": "SurnamePart"
      }
    ]
```

- **Факты** (facts)

  Массив элементов. Описание факта см. в разделе Свойства факта

Свойства факта

Для каждой сущности система анализирует и выделяет факты, в которых зафиксировано участие сущности. Факт имеет следующие свойства:

- **Имя факта** (name)

Имя факта в тексте

- **Тип факта** (type)

  Может принимать следующие значения:

- действие (action)
- событие (event)
- эмоция (emotion)
- высказывание (saying)
- упоминание (mentioning)
- **Тип актанта сущности, которой принадлежит факт** (atype)

Описание типа актанта см. в разделе Свойства актанта

- **Тональность факта** (tonality)

Для высказываний и упоминаний указывается тональность. Может принимать значения:

- высказывание с отрицательной тональностью
- высказывание с положительной тональностью
- тонально маркированное высказывание
- **Актанты** (actants)

Массив элементов. Описание актанта см. в разделе Свойства актанта

- **Семантика глагола** (semantic)
- **Подсемантика глагола** (subsemantic)
- **Позиции в тексте** (references)

Массив элементов. Позиции для факта — это позиция и длина его главного слова в тексте - обычно одна, но если факт состоит из 2-х слов, например, "хочу спать", то тогда 2 - для первого и второго слова. Каждый элемент массива имеет следующие параметры:

- позиция в тексте (position)
- длина в символах (length)

Свойства актанта

- **Сущность** (entity)

См. описание Свойства сущности

- **Тип актанта** (atype)

Это название члена предложения в синтаксическом разборе.

Методы веб-сервиса

Доступны следующие методы, описанные ниже более подробно:

- `Languages` - возвращает список языков, для которых поддерживается извлечение сущностей
- `Profiles` - возвращает список доступных профилей для каждого языка
- `Localizations` – возвращает список доступных локализаций
- `Analyze` - анализирует входной текст, извлекая сущности и определяя тональность текста

Метод Languages

Описание

Метод возвращает список языков, для которых может быть выполнено извлечение сущностей. В качестве идентификатора языка используется обозначение языка в соответствии с RFC 5646 (http://www.rfc-editor.org/rfc/rfc5646.txt).

```
GET http://servername/AnalyzerSDK/service/languages
```

Входные данные

Отсутствуют

Выходные данные

Список доступных языков. Пример:

```
[
   "ru",
   "en"
]
```

Метод Profiles

Описание

Метод возвращает список доступных профилей для каждого языка.

```
GET http://servername/AnalyzerSDK/service/profiles?language={PREFIX}
```

Входные данные

`Language` - язык разбора. Может принимать одно из значений, возвращаемых методом `Languages`

Выходные данные

Список профилей перевода. Пример:

```
[
      "News",
      "Curriculum Vitae",
      "Delivery contract",
…
]
```

Метод Localizations

Описание

Метод возвращает список языков, на которые могут быть локализованы предопределенные свойства сущностей. В качестве идентификатора языка используется обозначение языка в соответствии с RFC 5646 (http://www.rfc-editor.org/rfc/rfc5646.txt).

```
GET http://servername/AnalyzerSDK/service/localizations
```

Входные данные

Отсутствуют

Выходные данные

Список доступных языков. Пример:

```
[
   "ru",
   "en"
]
```

Метод Analyze

Описание

Метод анализирует входной текст, извлекая сущности и определяя тональность текста.

```
HTTP запрос
```

```
GET http://servername/AnalyzerSDK/service/analyze
--- или ---
POST http://servername/AnalyzerSDK/service/analyze
CONTENT-TYPE: application/x-www-form-urlencoded
```

Входные данные

`Text` – входной текст
`Language` - язык анализируемого текста. Может принимать одно из значений, возвращаемых методом `Languages`.
`Profile` - профиль настроек для разбора. Может принимать одно из значений, возвращаемых методом `Profiles`.
`Localization` (опциональный параметр) – язык локализации свойств сущностей. Может принимать одно из значений, возвращаемых методом `Localizations`. По умолчанию используется русская локализация (ru).

Выходные данные

На выходе получается массив извлеченных сущностей, а также тональность текста:

```
{
      "entities": [],
      "tonality": ""
}
```

Поле тональность текста может принимать следующие значения:

- отрицательная
- положительная
- смешанная

Пример разбора

В качестве примера для разбора использовался следующий текст:

Вчера пресс-секретарь Дмитрий Песков сказал в Москве господину Франсуа Олланду, что подобные угрозы не могут повлиять на последовательную политику России и президента Путина.

В процессе разбора были выделены следующие 5 сущностей (часть полей не показана, см. ниже):

```
[
  {
    "id": 0,
    "name": "Дмитрий Песков",
    "type": "персона"
  },
  {
    "id": 1,
    "name": "Франсуа Олланд",
    "type": "персона"
  },
  {
    "id": 2,
    "name": "Путин",
```

```
    "type": "персона"
  },
  {
    "id": 3,
    "name": "Москва",
    "subtype": "город, столица",
    "type": "география"
  },
  {
    "id": 4,
    "name": "Россия",
    "subtype": "страна",
    "type": "геополитика"
  }
]
```

Рассмотрим более подробно описание сущности “Дмитрий Песков”. У сущности есть атрибуты:

```
"attributes": [
      {
        "name": "пресс-секретарь",
        "type": "должность"
      }
    ],
```

Имя и фамилия:

```
"personnames": [
      {
        "name": "Дмитрий",
        "type": "имя"
      },
      {
        "name": "Песков",
        "type": "фамилия"
      }
    ]
```

Ссылки на позицию сущности в тексте:

```
"references": [
      {
        "length": 14,
        "position": 22
      }
    ]
```

Описание факта (часть полей не показана):

```
"facts": [
{
      "actants": [
            {
"atype": "Subject",
```

```
"entity": {
      "name": "пресс-секретарь Дмитрий Песков",
      "type": "персона"
}
            },
            {
"atype": "Predicate",
"entity": {
      "name": "сказал",
      "type": "Ж"
}
            },
            {
"atype": "Addressee",
"entity": {
      "name": "господину Франсуа Олланду",
      "type": "персона"
}
            },
            {
"atype": "AdverbialOfPlace",
"entity": {
      "name": "в Москве",
      "type": "география"
}
            },
            {
"atype": "AdverbialOfTime",
"entity": {
      "name": "Вчера",
      "type": ""
}
            },
            {
"atype": "Высказывание",
"entity": {
      "name": "подобные угрозы не могут повлиять на последовательную политику России и
президента Путина",
      "type": "высказывание с отрицательной тональностью"
}
            }
      ],
      "name": "сказал",
      "references": [
            {
"length": 6,
"position": 37
            }
      ],
      "tonality": "высказывание с отрицательной тональностью",
```

```
      "type": "Saying"
}
             ]
```

Описание ошибок

Ошибки, возникающие из-за неправильного запроса

Все ошибки, возвращающие код 400

Рекомендации: убедиться, что формат запроса соответствует указанному в документации.

Ошибки, возникающие из-за высокой нагрузки

500 Translation timeout

Рекомендации: снизить нагрузку или увеличить таймаут в конфигурационном файле. Значения таймаутов по умолчанию для вызываемых методов доступны в разделе [General] конфигурационного файла.

Ошибки, возникающие на конкретном тексте

500 Remote translator error: Empty response from child process - possibly crashed
500 Translation process crashed

Рекомендации: собрать логи, передать в Support. Пропустить «плохой» сегмент.

Критические ошибки, требующие перезагрузки сервисов

502 Bad Gateway
500 Remote translator error (connect)

Рекомендации: убедиться, что все сервисы PROMT Analyzer SDK запущены и функционируют.

Ошибки, свидетельствующие о проблемах с конфигурацией PROMT Analyzer SDK

500 Remote translator error: PromtUnit failed to start.
500 Forward request error (connect)
500 Remote translator error (connect)

Рекомендации: переустановить PROMT Analyzer SDK.

Лист регистрации изменений

Изм.	Номера листов (страниц)				Всего листов (страниц) в докум.	№ документа	Входящий № сопроводительного докум. и дата	Подп.	Дата
	измененных	замененных	новых	аннулированных					