

[Wiki »](#)

Установка и настройка

Установка

Установка выполняется через apt.

Репозиторий:

```
deb [arch=amd64] http://apt.ngn.eltex.loc/bionic/3.14 unstable main extras external
```

Необходимо установить пакет **ecss-clerk**.

Опционально можно установить kaldi-eltex-ru (собранная на пайпнайле гитлаба модель на основе телефонной книги офисного ssw, после установки нужно перезапустить systemctl сервис kaldi-ru).

Во время установки необходимо ответить на все задаваемые вопросы, особенно важно указать правильный url для импорта телефонной книги в формате xml и данные mysql пользователя. Во время установки будут применены миграции в бд (создание базы данных, таблицы и т.д.). Во время первого запуска начнут создаваться дефолтные алиасы в бд(около 6.5к).

После создания алиасов рекомендуется запустить **рекомпиляцию asr модели**

Управление через systemd

Убедитесь, что сервисы запущены:

```
sudo systemctl status ecss-clerk
```

```
sudo systemctl status kaldi-ru
```

В противном случае сервис необходимо включить.

Конфигурация

Конфигурация происходит при установке deb пакета ecss-clerk. Сам файл с конфигурацией расположен по пути /etc/ecss/ecss-clerk/config.yaml:

```
http:
  url: localhost # ip адрес ecss-clerk
  port: 9010     # порт ecss-clerk
asr:
  manager:
    ip: localhost # ip адрес http сервера kaldi (kaldi-ru)
    port: 9011    # порт http сервера kaldi (kaldi-ru)
    backend: tcp_kaldi # сервис, с помощью чего kaldi-ru запускает модель (сейчас не используется)
    frequency: 48000  # частота, на которой будет запущен asr
pcm:
  save_pcm: true # сохранять ли приходящие pcm
  pcm_dir: /var/lib/ecss/ecss-clerk/pcm_files # куда сохранять pcm, если нужно
phonebook:
  url_xml: http://system.restfs.ecss:9990/mysql?host=book&user_agent=cisco_search&translit=false&
log:
  level: debug # уровень лога
  console_level: info # уровень консольного лога (видно через journalctl)
  path: /var/log/ecss/ecss-clerk # куда писать лог
database:
  database: ecss_clerk_db_repo # название бд в mysql
  username: ecss-clerk
  password: password
  hostname: localhost
```

```
aliases_dir: /var/lib/ecss/ecss-clerk/default_aliases # директория, в которой хранятся файлы с /
```

Править конфиг ручками не рекомендуется, лучше делать это через dpkg-reconfigure

Проверка работоспособности

Для проверки работоспособности можно обратиться к сервису через curl:

```
curl <host>:9010/test.pcm -T test.pcm
```

Где host - адрес хоста, на котором запущен ecss-clerk, test.pcm - частота из config.yaml, кодек pcm_s16le, 1 канал (можно с wav заголовком).

Настройка на стороне ssw

Интеграция с ssw выполнена через IVR скрипт (сам скрипт будет находиться по пути /etc/ecss/ecss-clerk/simple_auto_attendant5.json и приложен к задаче).

В случае, если ecss-clerk будет стоять на отдельном хосте, нужно изменить адрес сервиса распознавания речи в блоке IVR с localhost на необходимый.

Дефолтный asr_service:

```
"asr_service": "localhost:9010"
```

После импорта скрипта необходимо настроить нового абонента и включить на нём дво "prpersonal ivr" и выбрать скрипт, проимпортированный ранее.

Также необходимо указать ключ от yandex speech-kit для работы синтеза речи.

Принцип работы дефолтного ivr скрипта

После звонка на ivr голос от абонента идет на asr_service (**БОТ ТАК БОТ**). Если до таймаута приходит ответ с номером, то происходит звонок на этого абонента. В противном случае мы оказываемся в другой ветке, из которой делается повторный запрос (блок goto, который выполняется 2 раза). Причем, если промежуточный ответ от автосекретаря содержал не пустое поле с распознанной строкой, то она будет проговорена.

Кроме этого есть два RPC блока, необходимые для вызова коллбеков на стороне ecss-clerk. Они тегируют запросы как удачные и не удачные. Соответственно, если было падение с таймаутом, то будет вызван негативный коллбек (negative_url). Если в ответе был номер, то будет вызван удачный коллбек (positive_url).

Настройка ecss-restfs

Ключ на синтез можно настроить прямо на restfs (dpkg-reconfigure ecss-restfs).

```
Проверить что настройки применились:  
curl http://ssw_ip:9990/tts_info
```

Отправить запрос на генерацию аудио:

```
curl "ssw1.eltex.loc:9990/?host=tts&voice=zahar&emotion=good&lang=ru-RU&text=алё+ну+как+там+с+ден  
ffmpeg -ar 8000 -ac 1 -f s16le -i test.pcm test.wav -y
```

И прослушать файл. В случае ошибки она будет содержаться в текстовом виде в test.pcm

После настройки можно позвонить на созданного абонента. Если всё настроено правильно вы услышите сообщение приветствие.

Логирование

Логи пишутся в /var/log/ecss/ecss-clerk/* и journalctl.

Формат логов

```
date time [level] [stage] message
```

- date - дата
- time - время
- level - уровень лога (info, debug, error)
- stage - этап, который в данный момент проходит принятый запрос. Вот некоторые из них:
 - Behaviour - поиск шаблона, в соответствии с которым будет обрабатываться пришедший запрос. Сейчас их два:
 - recognize and search - распознавание и поиск по абонентской книге
 - Collecting statistics - обработка колбеков для сбора статистики
 - Receive PCM - прием PCM
 - Saving PCM - сохранение PCM (сначала в оперативной, а потом и в постоянной памяти)
 - Recognize - распознавание PCM
 - Phonebook search - поиск по телефонной книге
 - Response JSON - возвращаемый JSON
- message - собственно, само сообщение

Прием PCM начинается с примерно такого сообщения:

```
2020-08-02 15:28:51.401 [info] PUT /2020-03-05_14-00-40-903204_asr_4813-4892.pcm
```

И заканчивается таким сообщением:

```
2020-08-02 15:28:52.418 [info] Sent 200 in 1016ms
```

Т.к. PCM приходит по частям(чанками), таких PUT запросов может приходиться несколько и у таких запросов будет одинаковый url.

simple_auto_attendant5.json Скрипт для импорта (8,13 КБ) Антон Черненко, 25.01.2021 16:57