

ё

G-MD2-1 «Платформа Атом.РИТА»
Модуль Атом.РИТА: Голос
Руководство пользователя

На 14 листах

Дата: 06.04.2022

Версия: 1.0.0

Лист изменений

| Дата | Версия | Краткое описание изменений | ФИО |
| --- | --- | --- | --- |
| 06.04.2022 | 1.0.0 | Документ создан | Поляков А.В. |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

Содержание

[Глоссарий 4](#_Toc100134261)

[1. Введение 5](#_Toc100134262)

[1.1. Назначение системы 5](#_Toc100134263)

[1.2. Документация, с которой необходимо ознакомиться пользователю 5](#_Toc100134264)

[2. Требования к пользователю 6](#_Toc100134265)

[3. Состав платформы 7](#_Toc100134266)

[4. Работа с модулем 8](#_Toc100134267)

[4.1. API модуля Атом.РИТА: Голос 8](#_Toc100134268)

[4.1.1. Синтез речи 8](#_Toc100134269)

[4.1.2. Анализ речи 9](#_Toc100134270)

[4.1.3. Диагностика сервиса 10](#_Toc100134271)

[4.2. Сценарий взаимодействия роботов с модулем Атом.РИТА: Голос 11](#_Toc100134272)

[5. Аварийные ситуации 14](#_Toc100134273)

Глоссарий

| **Термины/сокращения** | **Определение** |
| --- | --- |
| Платформа Атом.РИТА, Платформа | Роботизированный интеллектуальный технологичный ассистент |
| Программный робот (Робот) | Программное обеспечение для эмуляции действий человека, взаимодействующего с информационными системами, для выполнения типового бизнес-процесса |

# Введение

## Назначение системы

Платформа Атом.РИТА предназначена для разработки и управления программными роботами, позволяющими автоматизировать бизнес-процессы путем воспроизведения действий пользователей в установленных на рабочих станциях программах. Данный документ описывает модуль Атом.РИТА: Голос из состава Платформы Атом.РИТА. Состав Платформы подробно рассмотрен в разделе 3 настоящего документа.

## Документация, с которой необходимо ознакомиться пользователю

В зависимости от конфигурации Платформы, может потребоваться ознакомление с Руководством пользователя модуля Атом.РИТА: Разработчик.

# Требования к пользователю

Пользователь должен знать и соблюдать установленную технологию обработки информации, выполнять руководства, инструкции и положения по обеспечению защиты информации. Ответственным за ознакомление Пользователей c руководствами, инструкциями и положениями по обеспечению защиты информации является Администратор информационной безопасности (далее – Администратор ИБ)

Обо всех выявленных нарушениях, связанных с информационной безопасностью, Пользователь должен незамедлительно сообщать Администратору ИБ.

# Состав платформы

Платформа Атом.РИТА предназначена для разработки и управления программными роботами, позволяющими автоматизировать бизнес-процессы путем воспроизведения действий пользователей в программах, установленных на рабочих станциях.

 Платформа состоит из пяти модулей (см. рисунок 1):

* Атом.РИТА: Разработчик – модуль, предназначенный для создания, хранения, изменения и отладки программных роботов, реализованный как веб-приложение.
* Атом.РИТА: Администратор – модуль, предназначенный для автоматического запуска и управления роботами, созданными в модуле Атом.РИТА: Разработчик, а также мониторинга их состояния и аудита работы, реализованный как веб-приложение.
* Атом.РИТА: Сотрудник – модуль, предназначенный для выполнения программного робота, устанавливаемый на рабочую станцию.
* Атом.РИТА: Классификатор - модуль, предназначенный для автоматического определения классов (признаков, характеристик) текстовых объектов. Модуль возможно обучить для работы с различными текстами и определения выбранных пользователем классов (от 1 до 5).
* Атом.РИТА: Голос – модуль, предназначенный для распознавания речи и ее перевода в текст, а также синтеза речи на основании заданного текста.



Рисунок 1. Функциональная структура платформы Атом.РИТА

# Работа с модулем

## API модуля Атом.РИТА: Голос

* + 1. Синтез речи

**Отправка запроса на синтез речи**

**Endpoint:**<http://10.126.109.106:8888/api/v1/tts/synthesis>

**Метод:** Post

**Входные параметры** (передаются в теле запроса в виде JSON-документа):

|  |  |
| --- | --- |
| uid | string (Uid)Уникальный идентификатор запроса |
| text | string (Text)Текст для воспроизвдения |
| name | string (Name)Имя актера для воспроизведения (пол) |
| speed | number (Speed)Тембр воспроизведения речи (темп)от 0,75 до 1,25, 1 - по умолчанию |
| callback | string (Callback)URL для отправки ответа |

**Логика:**

При поступлении нового сообщения модуль Атом.РИТА: Голос:

* извлекает текст из файла (если поступил файл);
* синтезирует речь согласно полученному тексту и настройкам голоса (пол и темп). При выборе голоса и темпа используются данные, полученные в сообщении, при их отсутствии в сообщении - настройки по умолчанию (из конфигурации);
* записывает синтезированную речь в файл формата .wav;
* отправляет сформированный файл и идентификатор запроса на адрес сервиса, полученный в запросе (в callback).

**Ответ:**

Response Schema: application/json

|  |  |
| --- | --- |
| uidrequired | string (Uid)Уникальный идентификатор запроса |

Успех - код 200.

Ошибка валидации - код 422.

**Получение записи по uid**

**Метод:** Get

**Входные параметры:**

|  |  |
| --- | --- |
| uid | string (Uid)Уникальный идентификатор запроса |

**Логика:**

При поступлении запроса Голос:

* производит поиск в базе файлов (результатов синтеза речи) по полученному в запросе uid;
* в случае наличия файла - отправляет его и uid в ответ на запрос;
* при отсутствии файл с необходимым uid (запрос с данным uid не поступал ранее, файл еще не сформирован или уже удален) - отправляет ошибку 422.

**Ответ:**

Response Schema: application/json

|  |  |
| --- | --- |
| uidrequired | string (Uid)Уникальный идентификатор запроса |
| file |  |

Успех - код 200.

Ошибка валидации - код 422.

* + 1. Анализ речи

**Отправка запроса на синтез речи**

**Endpoint:** http://[10.126.109.106:8888](http://10.126.109.106:8888/api/v1/tts/synthesis)/asr/recognize

**Метод:** Post

**Входные параметры:**

|  |  |
| --- | --- |
| uid | string (Uid)Уникальный идентификатор запроса |
| callback | string (Callback)URL для отправки ответа |

**Логика:**

При поступлении нового сообщения Голос:

* получает звуковую дорожку и преобразовывает ее в текстовое сообщение с помощью технологии распознавания речи (ASR);
* отправляет сформированный текст и идентификатор запроса на адрес сервиса, полученный в запросе (в callback).

**Ответ:**

Response Schema: application/json

|  |  |
| --- | --- |
| uidrequired | string (Uid)Уникальный идентификатор запроса |

Успех - код 200.

Ошибка валидации - код 422.

**Получение записи по uid**

**Метод:** Get

**Входные параметры:**

|  |  |
| --- | --- |
| uid | string (Uid)Уникальный идентификатор запроса |

**Логика:**

При поступлении запроса Голос:

* производит поиск в базе текстов (результатов анализа речи) по полученному в запросе uid;
* в случае наличия записи - отправляет ее текст и uid в ответ на запрос;
* при отсутствии записи с необходимым uid (запрос с данным uid не поступал ранее, запись еще не распознана полностью или результат уже удален) - отправляет ошибку 422.

**Ответ:**

Response Schema: application/json

|  |  |
| --- | --- |
| uidrequired | string (Uid)Уникальный идентификатор запроса |
| text |  |

Успех - код 200.

Ошибка валидации - код 422.

* + 1. Диагностика сервиса

**Запрос ping**

**Endpoint:** [http://](http://10.126.109.106:8888/ping)[10.126.109.106:8888](http://10.126.109.106:8888/api/v1/tts/synthesis)/ping

**Метод:** GET

**Входные параметры:** отсутствуют

**Логика:**

При поступлении запроса Голос:

* прокси-сервер отправляет обратно ответ pong, что свидетельствует о работоспособности обратного прокси-сервера.

**Ответ:** pong

Успех - код 200.

**Запрос текущего состояния модулей (самодиагностика)**

**Endpoint:** [http://](http://10.126.109.106:8888/ping)[10.126.109.106:8888](http://10.126.109.106:8888/api/v1/tts/synthesis)/health

**Метод:** GET

**Входные параметры:** отсутствуют

**Логика:**

При поступлении запроса Голос:

* прокси-сервер отправляет запросы /ping всем зарегистрированным у него API-модулям, которые возвращают свой статус;
* В случае, если все зарегистрированные в обратном прокси-сервере API-модули возвращают ответ с кодом 200, то и ответ на этот запрос вернет код 200;
* В случае, если хотя бы один зарегистрированный в обратном прокси-сервере API-модули вернет ответ с кодом ошибки (400 и пр.), то и ответ на этот запрос вернет код 417.

**Ответ:**

Response Schema: application/json

|  |  |
| --- | --- |
| status | stringтекущий статус сервиса, например, 'ok' |
| services | intКоличество работающих экземпляров API-модулей  |

Система работает корректно - код 200.

Один из модулей системы работает некорректно - код 417.

## Сценарий взаимодействия роботов с модулем Атом.РИТА: Голос

Пример сценария настройки робота в модуле Атом.РИТА: Разработчик для взаимодействия с модулем Атом.РИТА: Голос в части синтеза речи:

1. Создать проект робота.
2. В проект робота добавить активность/набор активностей для получения текста, который будет синтезирован в голос: например, "Прочитать файл" (Read Text File) или "Открыть Excel" (Open Excel) + "Прочитать ячейку" (Read Cell).
3. Добавить активность "Выполнить python код" (Invoke Python Code) для формирования тела запроса в заданном виде:

import json
import uuid

json\_body = json.dumps({
  'uid': str(uuid.uuid4()),
  'text': text\_content,
  'name': 'rita'
})

где:

* uid - сгенерированный идентификатор записи;
* text - переменная с полученным текстом для синтеза речи;
* name - имя актера для воспроизведения.
1. Добавить активность "HTTP Запрос" (Http Request) для отправки текста в модуль Голос. Данная активность настраивается следующим образом:
* repeats\_num: 3;
* delay\_btw\_requests: 5000ms;
* endpoint: <http://10.126.109.106:8888/api/v1/tts/synthesis>;
* method: POST;
* auth\_type: NONE;
* body\_format: application/json;
* body: переменная с телом запроса (сформированная ранее);
* accept\_format: JSON;
* result\_body: переменная типа base.string для сохранения ответа от сервиса Голоса (для случаев короткого текста в ней придет сразу файл с записью речи и шаги 5-8 не требуются, переход к шагу 9);
* status\_code: переменная типа base.integer для сохранения кода состояния (ответа от сервиса Голоса).
1. Проверяется ответ от модуля Атом.РИТА: Голос с помощью активности "Условие TRUE/FALSE": если ответ =200, то выполняются следующие шаги. В случае другого ответа - направляться ошибка в лог/ответственному лицу.
2. После активности запроса установить задержку на 30 с, чтобы модуль Атом.РИТА: Голос успел обработать файл: добавить активность "Ожидание" (Delay) с длительностью (duration) 30000 мс (//приведено как пример, при настройке или отладке робота должно быть скорректировано с учетом задач робота).
3. Добавить активность "Выполнить python код" (Invoke Python Code) для формирования переменной, в которой будет адрес запроса с учетом ранее принятого/сгенерированного uid: <http://10.126.109.106:8888/api/v1/tts/synthesis>**?{uid}** .
4. Добавить активность "HTTP Запрос" (Http Request) для запроса у модуля Атом.РИТА: Голос результата синтеза речи (аудиофайла). Данная активность настраивается следующим образом:
* repeats\_num: 3;
* delay\_btw\_requests: 5000ms;
* endpoint: Созданная на шаге 7 переменная;
* method: GET;
* auth\_type: NONE;
* body\_format: application/json;
* accept\_format: JSON;
* result\_body: переменная типа base.string для сохранения ответа от сервиса Голоса;
* status\_code: переменная типа base.integer для сохранения кода состояния (ответа от сервиса Голоса).
1. Далее провести проверку переменной status\_code с помощью активности "Условие TRUE/FALSE": если ответ = 200, то выполняются следующие шаги. В случае другого ответа - направляется ошибка в лог/ответственному лицу.
2. Добавить активность "Десериализовать JSON" (Deserialize JSON) для преобразования ответа в виде строки (от модуля Голос) в объект JSON.
3. Для полученного объекта JSON с помощью активности "Присвоить значение" получить свойство "audio" (имя ключа, в котором находится синтезированный файл в формате base64).
4. Преобразовать полученный файл из формата base64 в файл с помощью активности "Base64 в файл" (Base64 to file) и сохранить его по заданному пути.
5. Далее, в зависимости от проекта, добавить активность/набор активностей для размещения файла в необходимой папке, отправке другой программе (например, осуществляющей обзвоны) или ответственному лицу по почте.

# Аварийные ситуации

При отказе или сбое в работе модуля необходимо обратиться к Администратору или в службу технической поддержки.