**ТЗ на разработку сервиса**

**хранения данных, полученных с сайта**

1. **Структура данных**

Request\_log

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **# п/п** | **Наименование поля** | **Тип** | **Обязательность** | **Значение по умолчанию** |
| 1 | id | bigint | + | Auto increment |
| 2 | createdon | datetime | + | Now() |
|  | external\_code | varchar | + |  |
| 3 | bot\_Id | int | + |  |
| 4 | source\_d | int | + |  |
| 5 | result\_code | int |  |  |
|  | requestdata | varchar |  |  |
|  | responsedata | varchar |  |  |
| 6 | started | datetime |  | Now() |
| 7 | finished | datetime |  |  |

Request\_Data

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **# п/п** | **Наименование поля** | **Тип** | **Обязательность** | **Значение по умолчанию** |
| 1 | id | bigint | + | Auto increment |
| 2 | createdon | datetime | + | Now() |
| 3 | request\_id | int | + |  |
| 4 | field\_name | varchar |  |  |
| 5 | field\_value | varchar |  |  |
| 6 | response\_code | int |  |  |
| 7 | response\_message | varchar |  |  |

Response\_data

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **# п/п** | **Наименование поля** | **Тип** | **Обязательность** | **Значение по умолчанию** |
| 1 | id | bigint | + | Auto increment |
| 2 | createdon | datetime | + | Now() |
| 3 | request\_id | int | + |  |
| 4 | field\_name | varchar |  |  |
| 5 | field\_value | varchar |  |  |
| 6 | response\_code | int |  |  |
| 7 | response\_message | varchar |  |  |

1. **Методы**

Во всех методах в заголовке должны быть представлены следующие поля, если не указано иное:

а. в поле Authorization должен быть авторизационный токен, который служит для идентификации вызывающей стороны. При неверном авторизационном токене ответ от сервера 401 Unauthorized

б. в поле Content-Type значение application/json

На все запросы сервер отвечает статусом 200 OK, любой другой статус считается ошибкой.

Если не указано обратное, то в ответе от сервера содержится json вида:

{

"result": 'ok', //строка, при успешном выполненном запросе содержит значение 'ok'. При неуспешном - тип ошибки

"result\_message": null, //строка, дополнительное описание ошибки. Если поле result содержит 'ok', то поле "result\_message" содержит null

.... //другие поля, специфические для каждого запроса

}

* 1. **Загрузка данных Request Data**

POST /request\_add

{

"external\_code": '765E9DE8-C25A-E911-A2BF-CE1FDA20D24A', //Guid, внешний идентификатор сущности (заявки), обязательный

"bot\_id": 'код модуля-бота', //целое число, обязательное поле

"source\_id": 'код внешнего ресурса-поставщика данных', //целое число, обязательное поле

"request\_data": 'json объект’// строка, данные, упакованные в объект, с которой будет работать бот

}

response: 200 OK

{

"result": 'ok' | 'access\_denied' | 'error',

"result\_message": null,

"request\_id": 'ad4eba40-69be-e911-82f2-1c1b0dafbcae', //Guid, идентификатор запроса. Обязательное поле}

В данном методе на сервисе проходит создание записей в таблице Request\_log (с генерацией ключа RequestId=Request\_log.Id) и request\_data (разбор json в поля FieldName и FieldValue.

* 1. **Загрузка извлеченных данных Response Data**

POST /response\_add

{

"external\_code": '765E9DE8-C25A-E911-A2BF-CE1FDA20D24A', //Guid, внешний идентификатор сущности (заявки), обязательный

"bot\_id": 'код модуля-бота', //целое число, обязательное поле

"source\_id": 'код внешнего ресурса-поставщика данных', //целое число, обязательное поле

"response\_data": 'json объект’// строка, параметры, которые получены ботом из внешнего сервиса', упакованные в объект

}

response: 200 OK

{

"result": 'ok' | 'access\_denied' | 'error',

"result\_message": null,

}

В данном методе на сервисе проходит

* Сохранение поля Finished в таблице Request\_log
* Создание записи в таблице request\_data (разбор json в поля FieldName и FieldValue).