

## Описание библиотеки

Данная библиотека позволяет подключаться в автоматическом режиме к центральному серверу и передавать данные свои. Передача “своих” данных состоит в том, что у программного агента есть свой REST API интерфейс

Строится на spring boot, создавая стартер, который сам конфигурируется

## Конфигурация

program-agent:

client:

host: "http://localhost:8027/api/"

myHost: "http://localhost:1016/api/"

token: "6b0ecc2f-3e32-422d-a152-1c035e28c918"

time-delay-send: 15000

Внутри спринг конфигурации указываются данные поля:

1. **host** - куда будет подключаться программный агент и передавать данные
2. **token** - секретный ключ генерируемый центральным сервером для подключения программного агента
3. **time-delay-send** - как часто программный агент будет сообщать своё состояние центральному серверу
4. **myHost** - куда будет обращаться центральный сервер за данными, сервер должен иметь к этому IP доступ, в удаленном режиме локальной проверки нет. Для этого будет реализована система в docker контейнерах

## Интерфейсы

Библиотека позволяет реализовывать интерфейсы для предоставления данных и регистрировать их в библиотеке. После чего сервер будет обращаться к программному агенту, вытягивая из интерфейсов данные.

На текущий момент есть 4 системы и 4 интерфейса (*ru.wdeath.programagent.lib* пакет):

1. `chart.ChartHandler` - представляет из себя представление временных графиков, формируя точки во времени с полезными данными. Это могут быть кол-во сообщений в одной минуте, или же нагрузка процессора за 5 минут
2. `form.actions.FormActionHandler` - представляет данные события, передавая “шаблон” или данные, и получая данные в виде строки на вход. Это может быть json конфигурации, или yaml настройки рассылки ботом
3. `progressbar.ProgressBarHandler` - данные о процессе, от 0 до 1
4. `tables.TableHandler` - данные в табличном виде, массив имен колонок и двухмерный массив данных

Соответственно каждый интерфейс имеет свой регистратор в движке, в каждом сервисе есть метод ***registration(obj)***

1. `ChartTimestampHandlerService`
2. `FormActionService`
3. `ProgressBarService`
4. `TableHandlerService`