



605aad03f0327629020004a76b249a1077ec



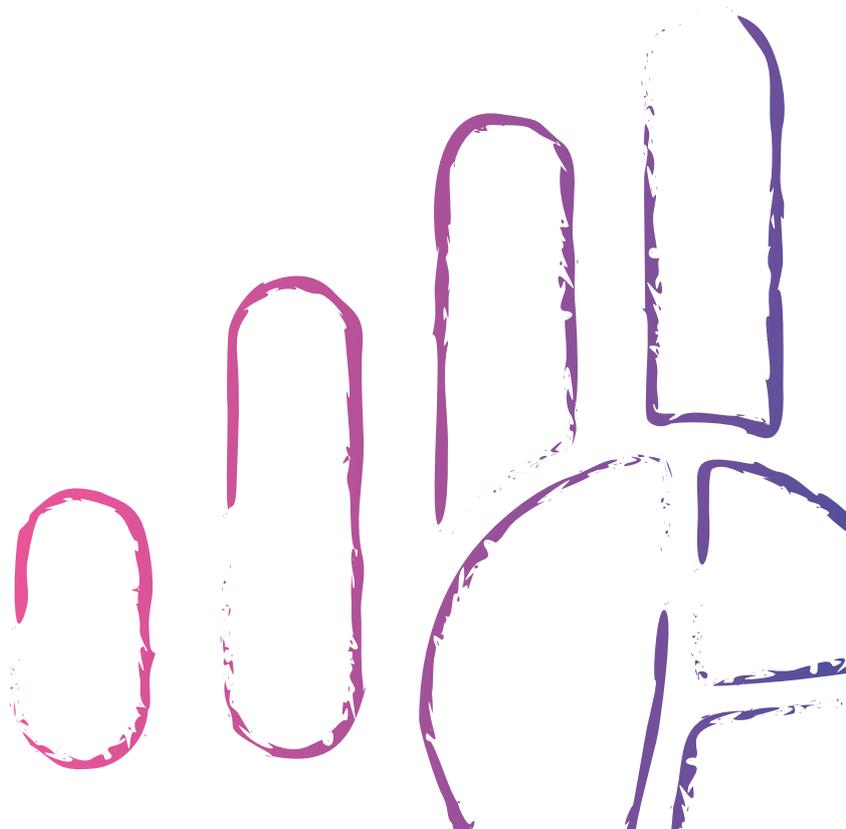
Функциональное описание

ПРИНЦИПЫ РАБОТЫ

ТЕРМИНОЛОГИЯ

СИСТЕМНЫЕ ТРЕБОВАНИЯ

2023-02-16





Оглавление

1	«Data Boring» Функциональное описание системы	1
1.1	Описание документа	1
1.2	Вводная информация	1
1.3	Терминология	2
1.4	Архитектура	3
1.4.1	Цель создания Data Boring	3
1.4.2	Структура	3
1.5	Техническая информация	6
1.6	Требования к аппаратной части	6
1.7	Минимальные технические требования	6
1.8	Стороннее ПО	7

1 «Data Boring» Функциональное описание системы

1.1 Описание документа

Документ описывает:

- Принципы и особенности программного продукта
- Необходимую техническую информацию и спецификации
- Архитектуру программного продукта «Data Boring»
- Описание системы
- Терминологию
- Описание области потокового программирования

Документ не подлежит копированию и/или распространению, а также использованию в целях, отличающихся от прямой цели ее предоставления, без согласия автора и правообладателя — ООО «ЯСП лабс».

1.2 Вводная информация

Data Boring — это инструмент программирования на основе потоков. Потоковое программирование, изобретенное Джоном Полом Моррисоном, представляет собой способ описания поведения приложения как сети черных ящиков или «узлов», связанных между собой определённым образом. Каждый узел имеет четко определенную цель — ему на вход даются определенные данные либо команды, он проводит заданные операции, а затем передает обработанные данные дальше по связям в другие узлы. При программировании на основе потоков основное внимание уделяется преобразованию данных с использованием набора готовых компонентов, которые по сути являются функциями обработки данных или сообщений.

Это модель, которая очень хорошо поддается визуальному представлению и понижает порог вхождения для использования системы. Это позволяет разбить задачу на отдельные шаги, без необходимости понимать отдельные строки кода в каждом узле.

Каждый узел принимает данные через абстрактный интерфейс, известный как порт, который работает так же, как сетевой порт в компьютерной сети. Данные отправляются на порт через буфер, который имеет ограниченный размер. Один порт может относиться к нескольким экземплярам одного компонента, что упрощает использование структуры в распределенной системе или для параллельной обработки.

Большая часть кода инкапсулирована, поэтому исходный код имеет высокий потенциал повторного использования. Это также означает, что обновление или масштабирование приложения, использующего потоковое программирование, может быть проще, чем при использовании более интегрированного приложения, поскольку систему обмена сообщениями, модули и систему портов можно изменять независимо, не затрагивая большую программу.

В качестве интерфейса пользователя в Data Boring используется проект с открытым исходным кодом «Node-RED». В браузере вы создаете свое приложение, перетаскивая узлы из своей палитры (подготовленного набора узлов) в рабочее пространство и соединяете их вместе. Одним щелчком мыши приложение развертывается на сервер в среду выполнения, где оно выполняется.

Упрощенная среда выполнения Node-RED разработана на базе Node.JS и благодаря этому максимально использует его событийно-ориентированную, неблокирующую модель. Это делает Node-RED идеальным приложением для запуска на сетевой периферии, а также в «облаке».

1.3 Терминология

Front-end (Клиент) – веб-приложение для пользователей, реализованное в виде HTML5/Javascript приложения для браузеров, для запуска открывается окно браузера и указывается URL адрес приложения.

Back-end (Сервер) – основанное на Node.JS приложения для выполнения, управления, изменения потоков и узлов

URL адрес - Uniform Resource Locator – унифицированный указатель ресурса.

Холст – часть Front-end, предназначенная для визуального создания и редактирования Потоков.

Администратор – именованный пользователь с доступом на чтение через пользовательский интерфейс, а также расширенным доступом на управление учётными записями и правами доступа

Браузер – программа для работы с Web ресурсами.

Поток (flow) – набор узлов, выполняющих определённую задачу последовательно согласно схеме расположения узлов в рабочей области

Узел (node) – независимый программный компонент, который решает определённую четко определённую задачу в рамках заданных пределов и с использованием заданных настроек

Палитра – набор доступных для использования в рабочей области узлов

Подпоток – совокупность узлов и настроек, вызываемых из текущего выполняемого потока, результат будет возвращен в родительский поток

Связь – соединение для передачи данных/сообщений между узлами

Источник данных – любое хранилище данных, в том числе файл Excel или CSV.

Группа – совокупность узлов, объединённых по назначению либо области применения

Отладка – узел который отображает данные из потока на боковой панели

Публикация (deploy) – команда которая разворачивает все потоки на backend

Пользователь – именованный пользователь с доступом на чтение через пользовательский интерфейс.

Входной порт - интерфейс узла для приема входных данных

Выходной порт – интерфейс узла для передачи обработанных данных в другие узлы

Триггер — это хранимая процедура особого типа, которую пользователь не вызывает непосредственно, а исполнение которой обусловлено действием заданных в условиях срабатывания

Боковая панель – панель рядом с рабочей областью, отображающая информацию о выделенном узле или о потоке

Импорт потока – загрузка в систему конфигурационного JSON файла который хранит описание узлов их расположение и контекстные данные

Экспорт потока – сохранение из системы JSON файла который описывает всю текущую конфигурацию потоков для дальнейшего использования.

1.4 Архитектура

1.4.1 Цель создания Data Boring

Программный продукт создан для решения задач ETL (выборки, преобразования и загрузки данных из различных источников) с помощью визуального интерфейса.

1.4.2 Структура

Условно программный продукт можно разделить на 3 блока:

1. Среда выполнения

Среда выполнения Data Boring — это сервер Node.JS, где с помощью команд вы можете развернуть и запустить backend-часть для работы приложения.

Существует обычный режим и безопасный режим запуска. При безопасном режиме откладывается выполнения потоков для возможности отладки узлов и потоков.

Также с помощью консольных команд можно посмотреть текущий статус работы системы.

Вы можете расширить палитру доступных модулей через специализированные консольные команды и модули из менеджера пакетов NPM или локальных узлов, описанных в соответствии с правилами и требованиями.

Data Boring расширяет среду выполнения за счёт дополнительных компонентов на Java, которые интегрированы с Node.JS через протокол RSocket. С помощью этой технологии реализованы дополнительные узлы для эффективной обработки данных и работы с различными источниками данных через JDBC.

2. Редактор

Front-end часть системы которая предоставляет доступ авторизованному пользователю к управлению конфигурацией потоков и узлов.

Условно редактор можно разбить на 3 области:

Палитра — список узлов, доступных для использования;

Холст — рабочее поле потока, куда требуется переместить необходимые узлы, задать им конфигурацию, выстроить связи между узлами. Для отображения длинных потоков вы можете использовать навигационные инструменты.

Боковая панель — содержит вкладки «информация», «справка», «отладочные сообщения», «узлы конфигурации». На каждой доступной вкладке есть набор инструментов и информации для использования и управления.

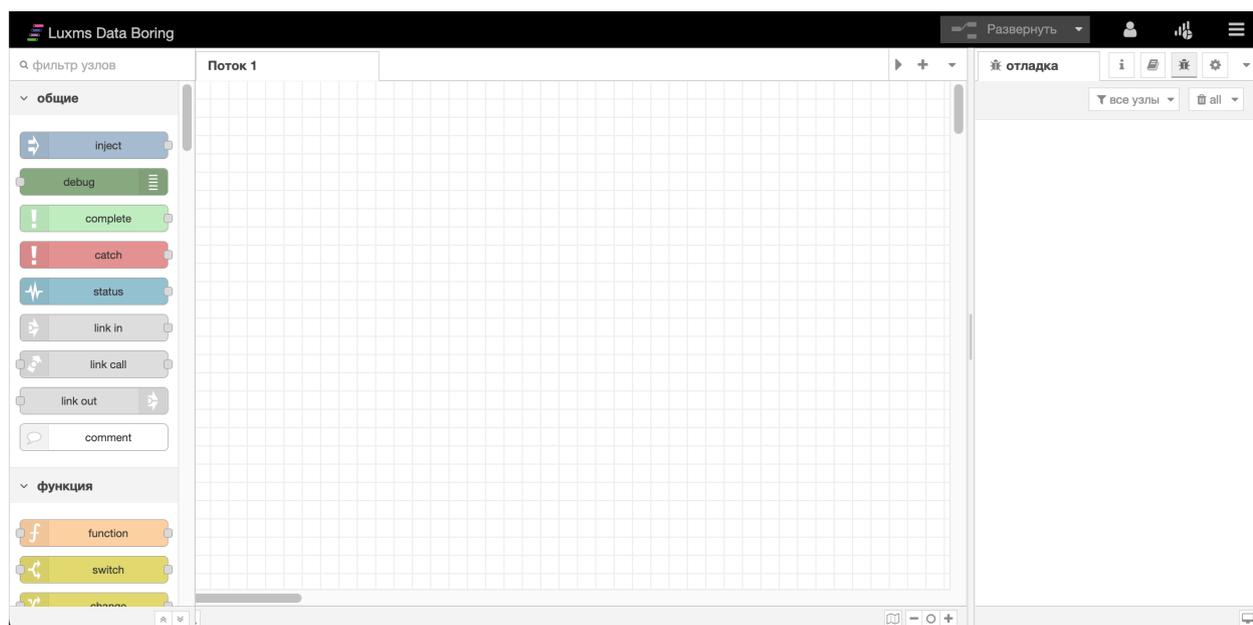


Рис. 1.1 Интерфейс пользователя Luxms Data Boring

В данном редакторе пользователь так же может изучить справочную информацию о системе, узлах и настройках, получить отладочную информацию и текущий статус узлов если это поддерживает сам узел и настройки отображения заданы. Все узлы доступные на панели управления сгруппированы по функциональному признаку. При наведении на любой узел из доступных вы получите справочную информацию о том, что данный узел выполняет и какие данные возвращает в следующий узел либо какие действия выполняет в случаи если узел не имеет выходных портов.

Основное меню позволит пользователю совершить ряд заданных действий, таких как импорт/экспорт потоков, упорядочивание узлов и связей, смену внешнего вида самого редактора, сгруппировать потоки по заданному признаку, установить дополнительные узлы. Так же в данном меню пользователь может пройти «краткий тур по системе», где будет показано как создать первый поток и решить задачу по преобразованию данных в БД.

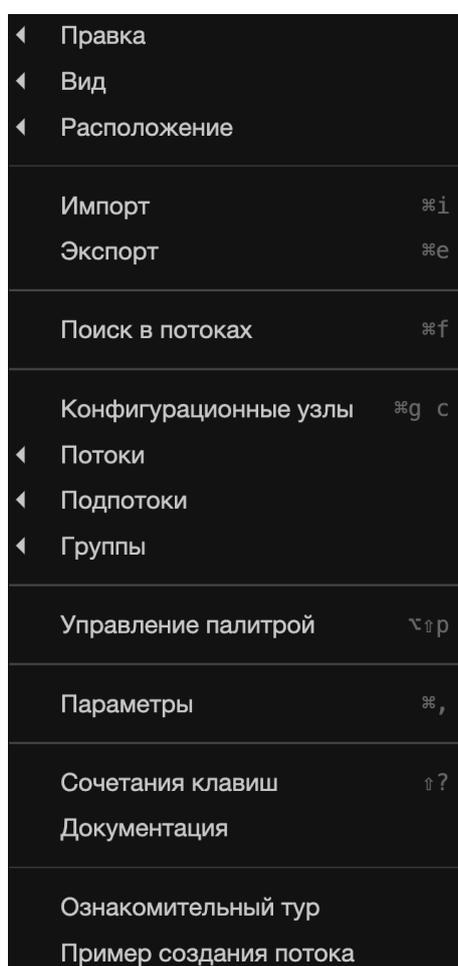


Рис. 1.2 Основное меню

3. Панель мониторинга

Панель мониторинга — конечный модуль системы, отвечающий за вывод результатов работы системы в графическом виде, в заданном виде и с указанными показателями. Панель мониторинга позволяет настроить под конкретные задачи формат отображения конкретных динамических данных в заданном диапазоне параметров и свойств.

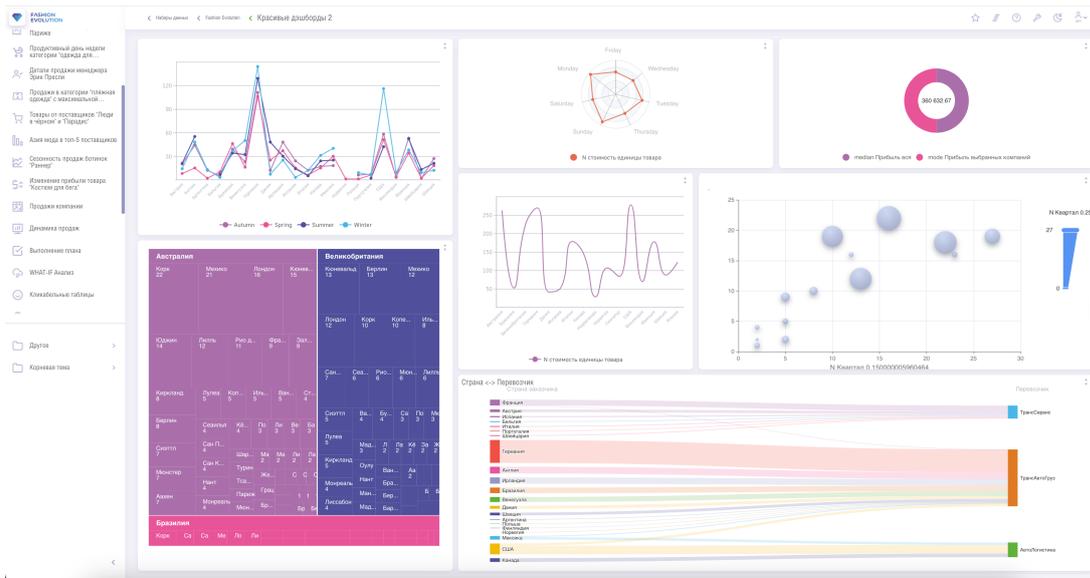


Рис. 1.3 Панель мониторинга

Для отображения динамических данных можно настроить период обновления и формат обновления, для экономии системных ресурсов обновляется не вся панель, а только заданная ее часть, тем самым, не загружая клиент избыточной отрисовкой.

1.5 Техническая информация

ПО Data Boring может быть изменено или доработано путем добавления новых групп узлов с заданной конечной и четко формируемой функцией, с входным/выходным портом. Логика работы узлов может быть реализована как на JavaScript так и на других языках, например на Java.

1.6 Требования к аппаратной части

Для успешного запуска и развертывания системы вам потребуется:

- Операционная система Linux CentOS Linux 7
- Административный доступ к системе
- Установленная платформа Node.JS не ниже версии 14
- JVM 11
- YUM (Yellowdog Updater, Modified) — открытый консольный менеджер пакетов

1.7 Минимальные технические требования

Для запуска программного продукта Data Boring сервер должен обладать следующим минимальным набором характеристик:

- Процессор с частотой от 1 ГГц
- Оперативная память 2 ГБ
- 1 ГБ свободного дискового пространства

1.8 Стороннее ПО

Стороннее ПО, которое используется в Data Boring удовлетворяет следующим условиям:

- Открытые исходные коды (open-source)
- Наличие публично доступного репозитория для скачивания исходных кодов
- Выход новых версий (или исправление ошибок) не реже двух раз в год
- Свободная лицензия: Apache 2.0, MIT, BSD, LGPL или совместимая

Название	Сайт	Лицензия
Node.js 14	https://nodejs.org/	MIT
jQuery 3	https://jquery.com/	MIT
Node-RED 2	https://nodered.org/	Apache 2.0



