ГОСУДАРСТВЕННАЯ КОРПОРАЦИЯ ПО АТОМНОЙ ЭНЕРГИИ «РОСАТОМ»

АКЦИОНЕРНОЕ ОБЩЕСТВО
«ОРДЕНА ЛЕНИНА НАУЧНО-ИССЛЕДОВАТЕЛЬСКИЙ И КОНСТРУКТОРСКИЙ ИНСТИТУТ ЭНЕРГОТЕХНИКИ ИМЕНИ Н.А. ДОЛЛЕЖАЛЯ»
(АО «НИКИЭТ»)

Функциональные характеристики программного обеспечения jCjS

2023

ПЕРЕЧЕНЬ СОКРАЩЕНИЙ И ОБОЗНАЧЕНИЙ

В настоящем документе применяют следующие сокращения и обозначения

БД – база данных

ПО – программное обеспечение

РБДРВ – распределенная база данных реального времени

СКУ – система контроля и управления оборудованием

СОДЕРЖАНИЕ

1. Общие сведения о продукте

jCjS – кроссплатформенное программное обеспечение, предназначенное для обмена данными и реализации клиентского взаимодействия с оборудованием различного типа через интернет браузер и позволяющее создавать системы контроля и управления оборудованием (СКУ), построенные на клиент-серверной архитектуре.

jCjS представляет собой среду исполнения и систему ввода-вывода для программ (скриптов), написанных на языке JavaScript (ECMA-262).

Основной задачей jCjS является опрос и обмен данными с устройствами через последовательные порты и отображение информации через WEB-интерфейс в браузере.

Функционал jCjS может быть расширен за счет плагинов, что позволяет реализовать управления любым нестандартным оборудованием, взаимодействие с различными базами данных, электронной почтой и т.д.

Помимо использования встроенных в jCjS плагинов функционал может быть расширен и адаптирован под решение иных задач путем подключения к jCjS дополнительных плагинов или реализацией необходимого функционала средствами JavaScript.

2. Функциональные характеристики

Функции jCjS основаны на событийно-ориентированной модели. Программные объекты, доступные в контексте среды JavaScript, генерируют сигналы событий, при наступлении которых выполняются функции, обслуживающие их.

Базовый функционал jCjS заключается в доступе к последовательным портам операционной системы и выводе данных на WEB-интерфейс.

Базовый функционал может быть расширен за счет подключения плагинов, формирующих плагин систему jCjS. Назначение плагинов может заключаться, например (рис. 1):

– в расширении средств ввода-вывода JavaScript,

– в обеспечении интерфейса к различному оборудованию,

– в использовании различных библиотек сторонних производителей,

– в обеспечении доступа к базам данных,

– в обеспечении доступа к системным функциям.

Рисунок 1 - Задачи, решаемые реализованными плагинами jCjS

В набор встроенных в программное обеспечение (ПО) jCjS плагинов входят:

– PluginActiveX – плагин для работы с объектами ActiveX в среде Windows,

– PluginBattery – плагин для контроля бесперебойного питания,

– PluginCamera – плагин вывода изображения с WEB-камеры в браузер,

– PluginExif - плагин для анализа основной информации из файлов JPEG,

– PluginExtUi - плагин управления менеджером оконных форм UI,

– PluginHID – плагин для доступа к устройствам HID,

– PluginLDAP - плагин для работы с LDAP,

– PluginMem - плагин контроля памяти, используемой приложением jCjS,

– PluginNetwork – плагин для работы с TCP/UDP сокетами,

– PluginPHP – плагин, позволяющий интегрировать интерпретатор PHP,

– PluginPlayer – плагин для проигрывания звуковых файлов (средствами Qt),

– PluginSound – плагин для проигрывания звуковых файлов wave (средствами ОС),

– PluginProcess – плагин для запуска приложений,

– PluginRBDRV – плагин для доступа к распределенной базе данных реального времени РБДРВ,

– jCjSPluginQrCode – плагин для генерации изображений кодов Qr,

– jCjSPluginQuaZip – плагин для упаковки файлов в архив zip,

– PluginSetting – плагин для работы с файлами формата ini,

– PluginSmtpClient – плагин для рассылки E-Mail уведомлений,

– PluginSQL – плагин для работы с различными БД,

– PluginWebkit – плагин для запуска оконного браузера.

3. Архитектура программного обеспечения

К основным составным частям ПО jCjS относятся HTTP-сервер и плагин система.

HTTP-сервер поддерживает минимально-достаточное подмножество протокола HTTP/1.1 (GET и POST запросы, базовую авторизацию, шифрованное соединение HTTPS и пр.) и содержит настроечные файлы, файлы скриптов и страниц HTML и т.д.

Сервер обрабатывает входящие запросы от клиентских браузеров при помощи постов.

Параметры запуска сервера, перечень и описание выполняемых постов содержатся в основном конфигурационном файле jCjS.

Плагин система обеспечивает доступ к расширению функционала ПО за счёт подключения плагинов, реализующих дополнительные по сравнению с базовыми возможности.

Перечень встроенных плагинов приведен в разделе 2. Плагин система непрерывно развивается и дополняется новыми плагинами, решающими возникающие перед пользователем задачи.

Пост – это программный объект, в котором запускается программа JavaScript, предназначенная для решения пользовательских задач.

Пост описывается следующими файлами:

1) Конфигурационный файл поста – специальный файл в формате xml, предназначенный для задания настроек поста. Данный файл создается при необходимости вынести настройки поста из основного конфигурационного файла jCjS в отдельный файл. При этом в основном конфигурационном файле jCjS остаются базовые параметры поста с указанием ссылки на отдельный конфигурационный файл поста.

2) Серверные скрипты – специальные скрипты, выполняемые в определенные моменты жизненного цикла поста на стороне сервера.

– скрипт инициализации (init.js) – выполняется при инициализации поста. Перечень подключаемых к данному посту плагинов осуществляется через этот файл.

– командный скрипт (cmd.js) – выполняется в процессе обработки запросов поста к серверу.

– финальный скрипт (end.js) – выполняется при завершении работы поста.

Все серверные скрипты указываются в конфигурационном файле (поста или основном конфигурационном файле). В случае отсутствия скриптов на соответствующих этапах жизненного цикла поста будут выполнены базовые скрипты.

3) Клиентские файлы – файлы, обеспечивающие взаимодействие с пользователем на стороне клиента (набор скриптов, стилей, шаблонов, WEB-страниц и т.д.).

Опционально пост может иметь графический интерфейс в виде WEB-страниц. Наличие графического интерфейса устанавливается специальным полем в конфигурации поста. При попытке перейти на WEB-интерфейс поста, для которой он отсутствует, будет выведено уведомление об отсутствии графического интерфейса для данного поста, после чего в автоматическом режиме произойдет переход на страницу по умолчанию, указанную в основном конфигурационном файле jCjS.

В общем виде схема функционирования ПО jCjS представлена на рис. 2.

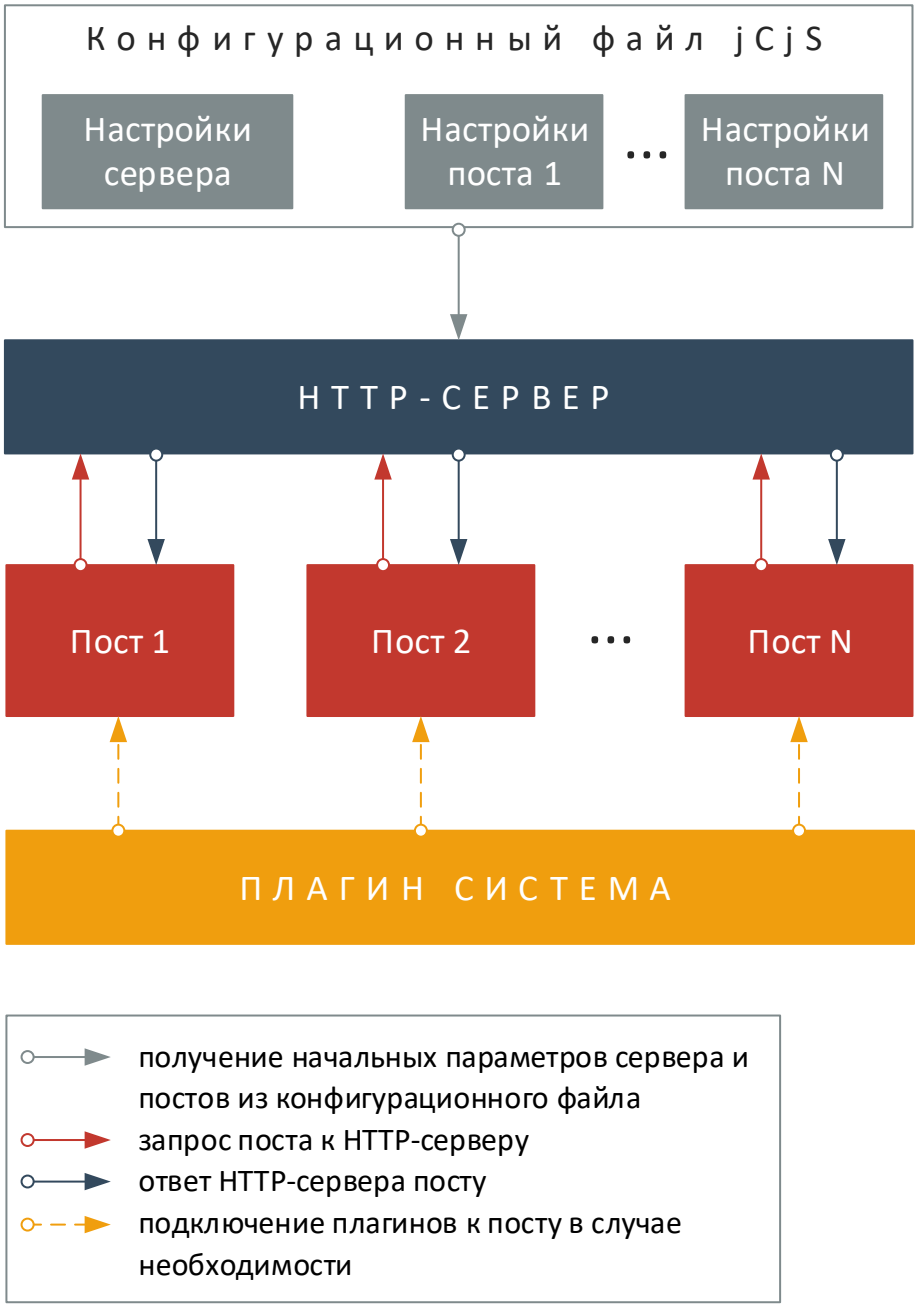

Рисунок 2 - Общий вид функционирования ПО jCjS

В момент начального запуска jCjS обращается к основному конфигурационному файлу, читает из него настройки сервера и описание постов, затем открывает HTTP порт и создает посты. Каждый пост создает интерпретатор JavaScript (JS-контекст), объявляет в своем JS-контексте глобальные объекты, а также специфические объекты, описанные в конфигурации поста, выполняет скрипт инициализации поста (init.js), что включает в себя подключение требуемых для функционирования плагинов, и отправляет серверу уведомление о статусе своей инициализации

(ошибка/успех). Если все посты инициализированы, сервер готов принимать входящие запросы от клиентских браузеров.

При обращении к серверу браузер инициализирует соединение по протоколу HTTP. Как только соединение становиться установленным, сервер создает объект http и передает созданный объект посту. Пост принимает объект соединения, внедряет его в свой JS-контекст и выполняет командный скрипт (cmd.js), задача которого обслужить запрос и ответить браузеру. Командный скрипт выполнят необходимые функции по анализу аргументов HTTP запроса и передает запрошенные данные браузеру.

Если jCjS получил сигнал к остановке, сервер закрывает все соединения и рассылает команды постам на завершение работы. Команды отсылаются в обратном инициализации порядке. Последним останавливается серверный пост. Пост, получив команду остановки, выполняет финальный скрипт (end.js), на котором лежат обязанности по приведению оборудования в нормальное состояние и отправке серверу сигнала об успешном завершении работы поста. Сервер ожидает от постов сигнала о финале работы. Если по каким-то причинам сигнал не был получен, сервер через 15 секунд принудительно завершит работу поста (это время можно изменить в JS-контексте).