

ЗАО «АЙТИ»

УТВЕРЖДАЮ



Генеральный директор

ЗАО «АЙТИ»

Бакиев О.Р.

2011 г.

Создание высокотехнологичного производства комплексных решений в области предметно-ориентированных облачных вычислений для нужд науки, промышленности, бизнеса и социальной сферы

**ПРОГРАММНЫЙ КОМПОНЕНТ СРЕДСТВ
МОНИТОРИНГА И НАСТРОЙКИ ОСНОВНЫХ
КОМПОНЕНТОВ ПЛАТФОРМЫ CLAVIRE/ADMINTOOL**

ОПИСАНИЕ ПРОГРАММЫ

RU.СНАБ.80066-06 13 64

Име. № подл.	Подпись и дата
Взам. инв. №	Име. № дубл.
Подпись и дата	Подпись и дата

УТВЕРЖДЕН
RU.СНАБ.80066-06 13 64-ЛУ

**МНОГОПРОФИЛЬНАЯ ИНСТРУМЕНТАЛЬНО-
ТЕХНОЛОГИЧЕСКАЯ ПЛАТФОРМА СОЗДАНИЯ
И УПРАВЛЕНИЯ РАСПРЕДЕЛЕННОЙ СРЕДОЙ
ОБЛАЧНЫХ ВЫЧИСЛЕНИЙ CLAVIRE**

**ПРОГРАММНЫЙ КОМПОНЕНТ СРЕДСТВ
МОНИТОРИНГА И НАСТРОЙКИ ОСНОВНЫХ
КОМПОНЕНТОВ ПЛАТФОРМЫ CLAVIRE/ADMINTOOL**

ОПИСАНИЕ ПРОГРАММЫ

RU.СНАБ.80066-06 13 64

ЛИСТОВ 21

Инв.№ подл.	
Подп. и дата	
Взам.инв.№	
Инв.№ дубл.	
Подп. и дата	

АННОТАЦИЯ

Документ содержит описание программного компонента CLAVIRE/AdminTool средств мониторинга и настройки основных компонентов платформы RU.СНАБ.80066-06 13 64, предназначенного для администрирования и контроля работы платформы. Компонент позволяет редактировать пользователей, вычислительные ресурсы, прикладные пакеты, тарифы на использование вычислительных ресурсов, а также позволяет задавать индивидуальные правила для каждого пользователя и анализировать состояние платформы. Программный компонент разработан в рамках проекта «Создание распределенной вычислительной среды на базе облачной архитектуры для построения и эксплуатации высокопроизводительных композитных приложений» в рамках реализации постановления Правительства РФ № 218 «О мерах государственной поддержки развития кооперации российских высших учебных заведений и организаций, реализующих комплексные проекты по созданию высокотехнологичного производства».

СОДЕРЖАНИЕ

1. ОБЩИЕ СВЕДЕНИЯ	4
2. ФУНКЦИОНАЛЬНОЕ НАЗНАЧЕНИЕ	4
2.1. Классы решаемых задач.....	4
2.2. Основные технологические функции.....	5
2.3. Ограничения на применение	5
3. ОПИСАНИЕ ЛОГИЧЕСКОЙ СТРУКТУРЫ	6
3.1. Общее описание структуры.....	6
3.2. Описание исполняемых модулей представления системы управления контентом.....	8
3.3. Реализация базы данных.....	11
3.4. Основные классы компонента.....	16
4. ИСПОЛЬЗУЕМЫЕ ТЕХНИЧЕСКИЕ СРЕДСТВА	17
5. ВЫЗОВ И ЗАГРУЗКА	17
6. ВХОДНЫЕ ДАННЫЕ	19
7. ВЫХОДНЫЕ ДАННЫЕ.....	19
ПЕРЕЧЕНЬ СОКРАЩЕНИЙ	20

1. ОБЩИЕ СВЕДЕНИЯ

Программный компонент CLAVIRE/AdminTool средств мониторинга и настройки основных компонентов платформы RU.СНАБ.80066-06 13 64 предназначен для администрирования и контроля работы основных компонентов платформы CLAVIRE.

Программный компонент CLAVIRE/AdminTool разработан на языке PHP с использованием технологии AJAX (асинхронный Javascript и XML). Для хранения данных была использована СУБД MySQL. Компонент использует программную библиотеку jQuery (версии 1.4 и выше), фокусирующуюся на взаимодействии JavaScript и HTML, которая помогает получать доступ к любому элементу DOM, обращаться к атрибутам и содержимому элементов DOM, манипулировать ими. В компоненте используется модуль FancyBox (версии 1.3.4 и выше), реализующий дополнительные визуальные возможности анимации интерфейса. Компонент может функционировать:

- 1) на вычислительной системе под управлением ОС Linux (с ядром 2.6.32 и выше) с установленным web-сервером Apache (версии 2.2.16 и выше) установленной средой PHP (версии 5.2 и выше) с поддержкой стандартного набора библиотек. Для корректного функционирования необходимо наличие модулей web-сервера CURL, DOM, FTP, GD, JSON, MYSQL, SOAP, XML, поставляющихся в составе стандартного набора библиотек web-сервера Apache;
- 2) на вычислительной системе под управлением ОС Windows (версии XP и выше) с установленным web-сервером Apache (версии 2.2.16 и выше) установленной средой PHP (версии 5.2 и выше) с поддержкой стандартного набора библиотек.

Программный компонент AdminTool используется программными компонентами PackageBase, ResourceBase, GateKeeper, Billing и по сути является интерфейсом для данных компонентов.

2. ФУНКЦИОНАЛЬНОЕ НАЗНАЧЕНИЕ

2.1. Классы решаемых задач

Программный компонент предназначен для решения следующих задач:

- 1) администрирование платформы, при помощи модулей конфигурирования;
- 2) анализ состояния платформы, просмотр отчетов и истории событий при помощи модулей просмотра журналов и статистики.

2.2. Основные технологические функции

Программный компонент представляет собой высокоуровневый интерфейс администрирования для нескольких компонентов системы: база пакетов CLAVIRE/PackageBase, база ресурсов CLAVIRE/ResourceBase, компонента обеспечения безопасности CLAVIRE/GateKeeper, компонента учета использованных ресурсов CLAVIRE/Billing. Схема взаимодействия представлена на рис. 2.1.

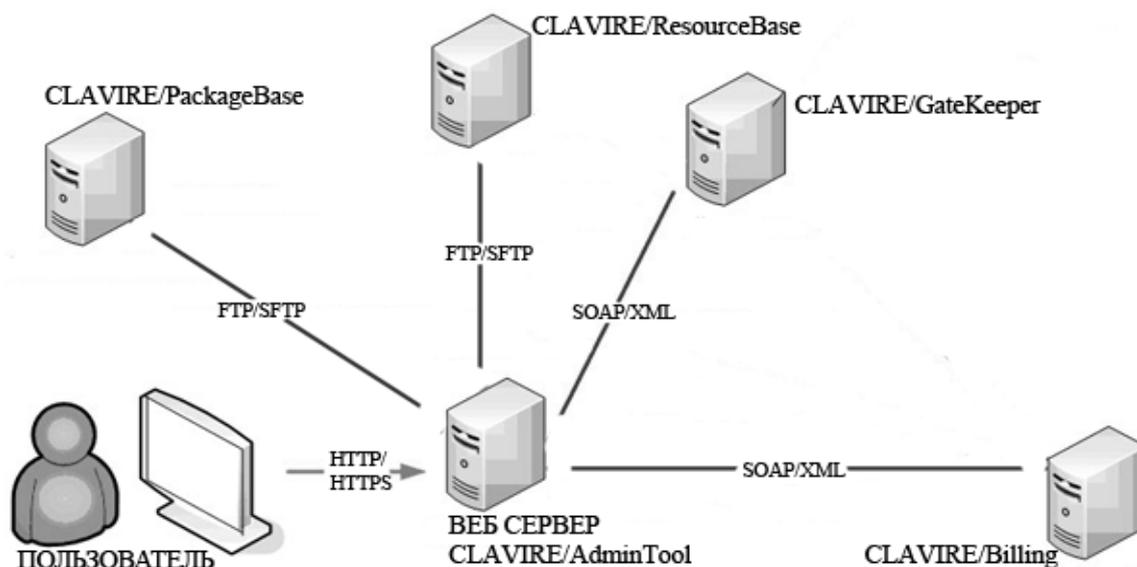


Рисунок 2.1 – Взаимодействие программного компонента AdminTool с другими компонентами системы

Основными технологическими функциями, реализуемыми программным компонентом AdminTool, являются:

- 1) просмотр журналов исполнения основных компонентов комплекса и статистики использования ресурсов платформы;
- 2) установка, настройка и корректировка прикладных пакетов в составе платформы;
- 3) добавление и настройка вычислительных ресурсов, используемых платформой;
- 4) управление списком пользователей платформы, а также настройка их прав в системе и задание ограничений на использование тех или иных ресурсов;
- 5) задание тарифов на использование системы для конкретных ресурсов и пакетов.

2.3. Ограничения на применение

Для корректной работы программного компонента необходимо обеспечить постоянный доступ со связанными компонентами комплекса, а также обеспечить работоспособность пользовательского интерфейса. В связи с этим основным условием

обеспечения бесперебойной работы компонента является обеспечение доступа всеми необходимыми компонентами системы и пользователями через сеть Интернет или локальную сеть (в случае расположения всех объектов и субъектов системы в рамках одной общей локальной сети).

3. ОПИСАНИЕ ЛОГИЧЕСКОЙ СТРУКТУРЫ

3.1. Общее описание структуры

Ядром компонента является система управления контентом (CMS), данная система разработана с использованием архитектуры MVC (Model–View–Controller, Модель–Представление–Контроллер).

Использование данного способа разработки позволяет:

- 1) Организовать повторное использование кода, создавая различные интерфейсы для доступа к функциональности приложений. Это важно, например, в случае работы с web-сервисами.
- 2) При необходимости реализовать дополнительные интерфейсы доступа к сторонним web-ресурсам при помощи фреймов HTML или создание интерфейсов доступа к web-сервисам – посредством обмена данными по протоколу SOAP к существующему поисковому сервису эта задача сводится к описанию нового формата представления, а также созданию объекта, отвечающего за вывод результатов в новом формате. При этом не требуется каких-либо изменений в существующей функциональности, а весь процесс отличается прозрачностью и простотой реализации.
- 3) Используя данную модель, можно изменить любую часть кода, не затрагивая при этом верстку, или изменить шаблоны без изменения программной части.

В состав архитектуры входят блоки: Модель, Представление, Контроллер.

Модель определяет бизнес-логику, содержит программный код, определяющий типы данных, их свойства и взаимосвязи, а также дополнительные модули, содержащие набор функций, которые позволяют сайту работать с базой данных посредством SQL-запросов, а также с web-сервисами при помощи SOAP-запросов.

Контроллер формирует логику ввода, интерпретирует данные, введенные пользователем, манипулирует сущностями Модели так, чтобы их можно было превратить в Представление.

Представление определяет логику интерфейса, генерирует HTML-страницы сайта из данных, полученных от Модели, интерпретированных Контроллером на основе запросов пользователя. К Представлению относятся шаблоны страниц, файлы стилей, Javascript-сценарии, приложения Flash и Silverlight.

Благодаря такой архитектуре логика ввода, бизнес-логика и логика интерфейса разделяются, но достаточно тесно взаимодействуют друг с другом. Это разделение позволяет работать со сложными структурами при создании сайта, так как обеспечивает одновременную реализацию только одного аспекта.

Схему типовой процедуры взаимодействия пользователя с интерфейсом можно описать следующим образом (рис. 3.1).

- Пользователь обращается к ресурсу через браузер, после чего браузер отправляет запрос с помощью метода GET или POST, обрабатываемый Контроллером, который, в свою очередь, обращается к Модели.
- «Модель» запрашивает данные у сервера базы данных с помощью SQL-запросов или у web-сервисов при помощи SOAP-запросов.
- Получив необходимые данные через Модель, Контроллер выводит нужную информацию пользователю через Представление, генерируя HTML-страницу.

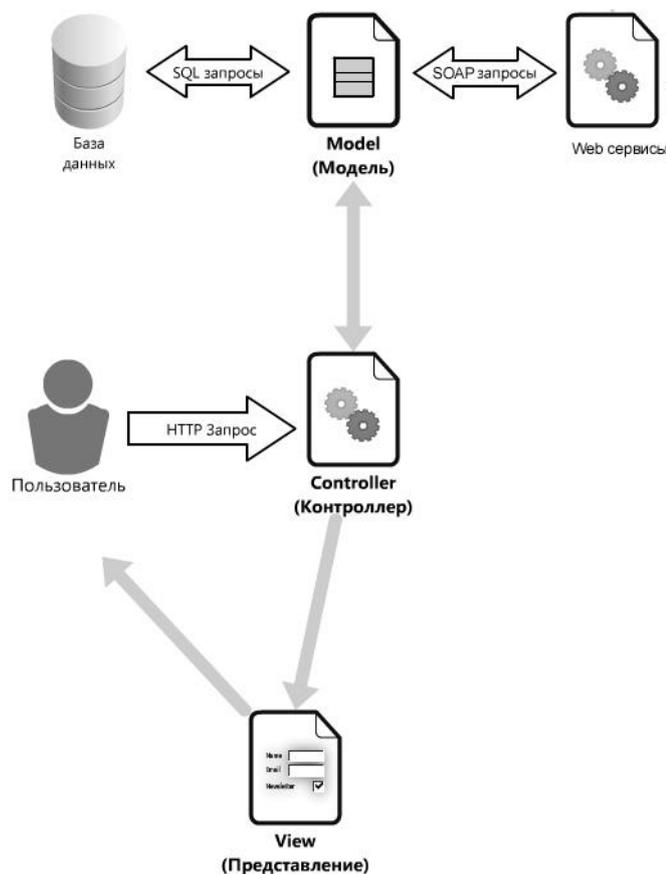


Рисунок 3.1 – Архитектура системы управления контентом

3.2. Описание исполняемых модулей представления системы управления контентом

В текущей реализации компонента CLAVIRE/AdminTool используется восемь модулей представления системы управления контентом. Данные модули обеспечивают исполнение технологического функционала: обмен, сбор и подготовку данных для дальнейшего предоставления пользователю и обеспечению функций администрирования.

Далее представлен перечень данных модулей с описанием принципа их работы:

- Модуль авторизации в системе.** Предоставляет функционал для авторизации пользователя на сайте. В качестве входных предоставляются параметры учетной записи пользователя (имя пользователя и пароль), после чего происходит запрос к базе данных о существовании учетной записи с такими параметрами и получение сведения о правах пользователя в системе. В случае совпадения пары логин/ пароль, а также наличия у пользователя прав администратора системы происходят успешная авторизация и предоставление доступа к интерфейсу администратора. Авторизованному пользователю данный модуль предоставляет возможность выйти из своей учетной записи.

- *Модуль настроек администратора.* Предоставляет функционал для базовых настроек параметров доступа по FTP к базе пакетов CLAVIRE/PackageBase и базе ресурсов CLAVIRE/ResourceBase. Данный модуль позволяет просматривать и хранить параметры доступа к данным компонентам в базе данных.
- *Модуль просмотра статистики.* Предоставляет функционал для просмотра статистики с удаленного web-сервера посредством отображения во всплывающем фрейме содержимого данного ресурса.
- *Модуль просмотра журналов.* Предоставляет функционал для просмотра журналов с удаленного web-сервера посредством отображения во всплывающем фрейме содержимого данного ресурса.
- *Модуль администрирования пакетов.* Предоставляет функционал для администрирования компонента-базы пакетов CLAVIRE/PackageBase. Доступ к данным осуществляется по протоколу FTP или SFTP. У пользователя есть возможность устанавливать, настраивать, корректировать и удалять прикладные пакеты в составе платформы CLAVIRE. Данный сервис управляет удаленными данными компонента, обеспечивая безопасное редактирование и удаление с возможностью контроля и восстановления версий. На рис. 3.2 продемонстрирован интерфейс модуля в действии.

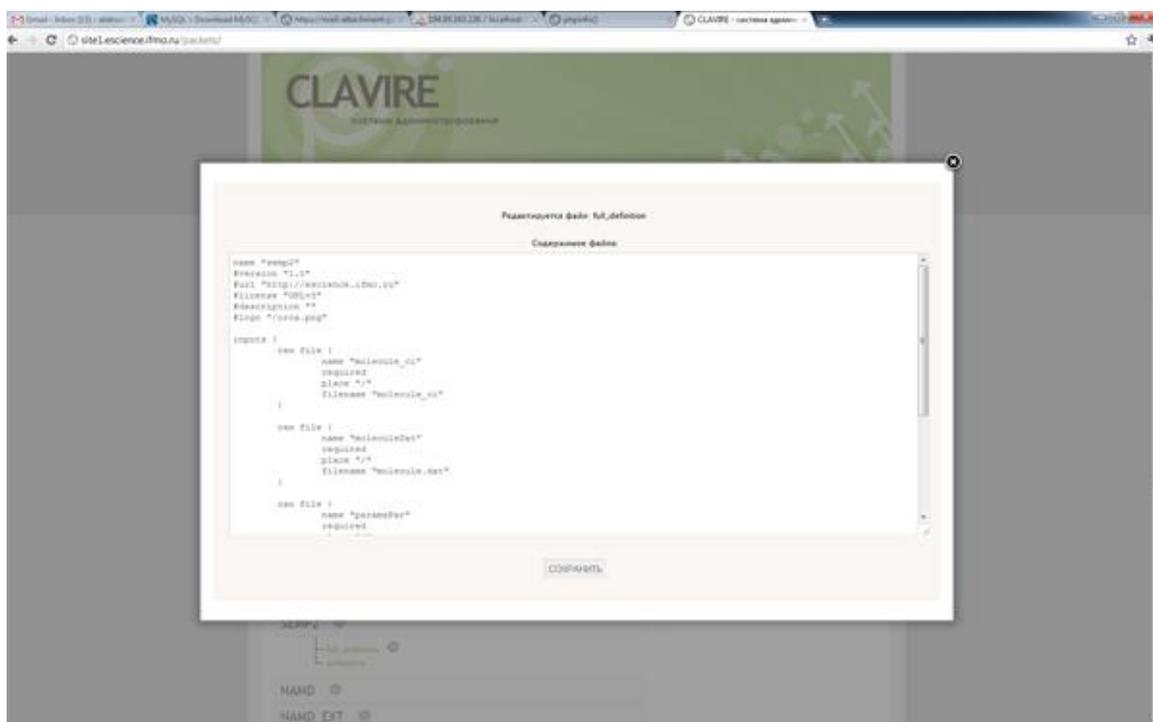


Рисунок 3.2 – Интерфейс модуля администрирования пакетов
в режиме редактирования

- *Модуль администрирования ресурсов.* Предоставляет функционал для администрирования компонента-базы ресурсов CLAVIRE/ResourceBase. Доступ к данным осуществляется по протоколу FTP или SFTP. У пользователя есть возможность добавлять, настраивать и удалять вычислительные ресурсы, используемые платформой CLAVIRE. Сервис управляет удаленными данными компонента, обеспечивая безопасное редактирование и удаление с возможностью контроля и восстановления версий.
- *Модуль администрирования пользователей.* Предоставляет функционал для администрирования компонента обеспечения безопасности CLAVIRE/GateKeeper. Доступ к данным осуществляется по протоколу HTTP или HTTPS с использованием технологии SOAP, которая использует для обмена данными формат XML. У пользователя есть возможность сменить пароль, назначить роль в системе, настроить прав пользователя и указать квоту использования ресурсов.
- *Модуль администрирования тарифов.* Предоставляет функционал для администрирования компонента учета использованных ресурсов CLAVIRE/Billing. Доступ к данным осуществляется по протоколу HTTP или HTTPS с использованием технологии SOAP, которая использует для обмена данными формат XML. У пользователя есть возможность задать и удалить тариф на использование системы для конкретных ресурсов и пакетов, с указанием точного времени действия. На рис. 3.3 продемонстрирован интерфейс модуля в действии.



Рисунок 3.3 – Интерфейс модуля администрирования тарифов
в режиме отображения данных

3.3. Реализация базы данных

Структура базы данных AdminTool, является довольно простой, так как компонент сам по себе является интерфейсом для нескольких других компонентов и в базе данных не содержится каких-либо промежуточных данных.

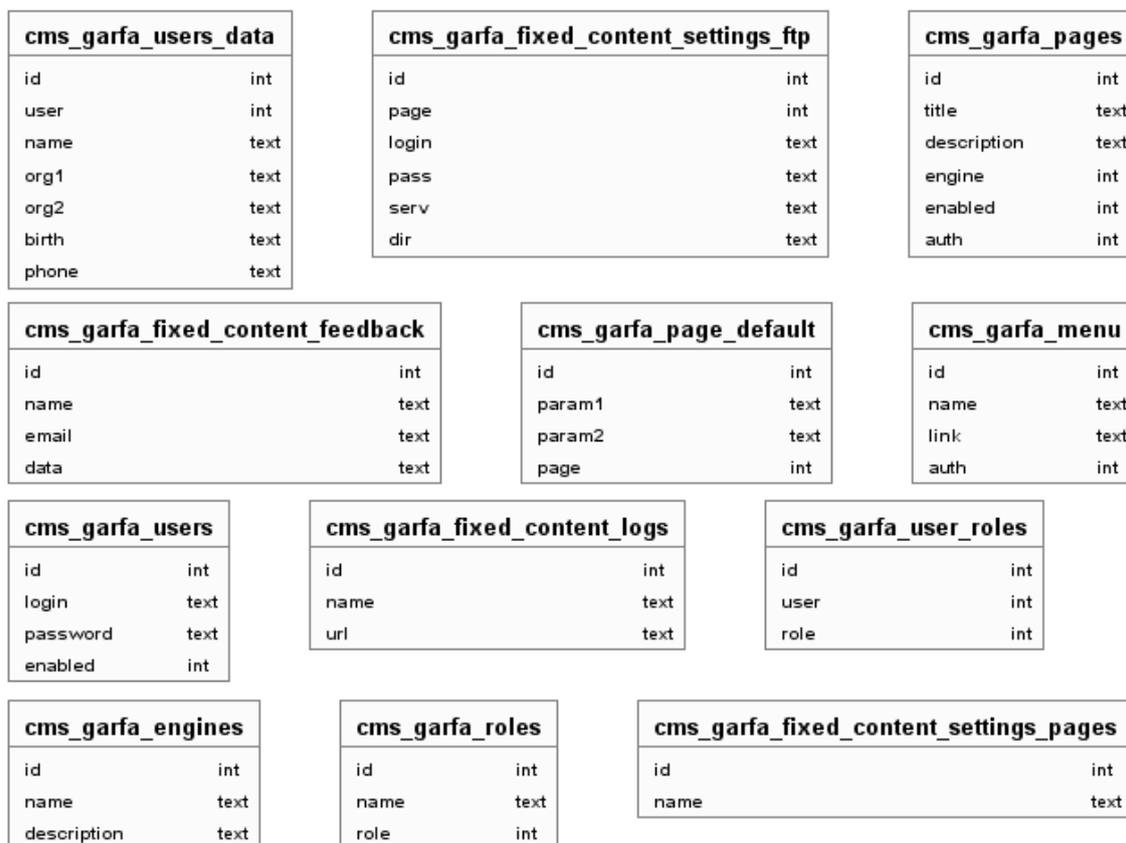


Рисунок 3.4 – Структура базы данных

Структуру базы данных можно разделить на таблицы, содержащие:

- технические данные системы управления контентом;
- пользовательские данные администратора системы.

К первой категории относятся таблицы с данными: о пользователях системы и их ролях, о меню, страницах и модулей представления.

- **cms_garfa_engines** – в данной таблице хранится перечень исполняемых модулей представления системы
 - a) id (тип: int) – идентификатор;
 - b) name (тип: text) – название модуля;
 - c) description (тип: text) – описание модуля.
- **cms_garfa_lang** – в данной таблице хранится перечень языковых локализаций системы
 - a) id (тип: int) – идентификатор;
 - b) name (тип: text) – название локализации;
 - c) bydefault (тип: int) – является ли вариантом по умолчанию;
 - d) data (тип: text) – пользовательское отображение в системе.

- `cms_garfa_local` – в данной таблице хранится перечень географических локализаций системы
 - a) `id` (тип: `int`) – идентификатор;
 - b) `name` (тип: `text`) – название локализации;
 - c) `bydefault` (тип: `int`) – является ли вариантом по умолчанию;
 - d) `data` (тип: `text`) – пользовательское отображение в системе.
- `cms_garfa_menu` – в данной таблице хранится перечень пунктов меню системы
 - a) `id` (тип: `int`) – идентификатор;
 - b) `name` (тип: `text`) – название пункта меню;
 - c) `link` (тип: `text`) – ссылка на исполняемый модуль;
 - d) `auth` (тип: `int`) – статус авторизация пользователя для отображения данного пункта.
- `cms_garfa_pages` – в данной таблице хранится перечень страниц перечня исполняемых модулей представления системы
 - a) `id` (тип: `int`) – идентификатор;
 - b) `title` (тип: `text`) – титул страницы;
 - c) `description` (тип: `text`) – описание страницы;
 - d) `engine` (тип: `int`) – идентификатор исполняемого модуля представления системы;
 - e) `enabled` (тип: `int`) – статус активации данной страницы;
 - f) `auth` (тип: `int`) – статус авторизации пользователя для отображения данной страницы.
- `cms_garfa_page_default` – в данной таблице хранится перечень условий запуска по умолчанию страниц перечня исполняемых модулей представления системы
 - a) `id` (тип: `int`) – идентификатор;
 - b) `param1` (тип: `text`) – параметр условия запуска;
 - c) `param2` (тип: `int`) – параметр значения запуска;
 - d) `page` (тип: `int`) – идентификатор страницы исполняемого модуля представления системы.
- `cms_garfa_page_role` – в данной таблице хранится перечень ролей пользователей, имеющих права запуска страниц перечня исполняемых модулей представления системы
 - a) `id` (тип: `int`) – идентификатор;

- b) page (тип: int) – идентификатор страницы исполняемого модуля представления системы;
- c) role (тип: int) – идентификатор роли пользователей.
- cms_garfa_roles – в данной таблице хранится перечень ролей пользователей системы
 - a) id (тип: int) – идентификатор;
 - b) name (тип: text) – наименование роли;
 - c) role (тип: int) – идентификатор роли пользователей.
- cms_garfa_users – в данной таблице хранится перечень пользователей системы
 - a) id (тип: int) – идентификатор;
 - b) login (тип: text) – имя пользователя;
 - c) password (тип: text) – пароль пользователя;
 - d) enabled (тип: int) – статус активации данного пользователя.
- cms_garfa_roles – в данной таблице хранится перечень привязок пользователей к ролям системы
 - a) id (тип: int) – идентификатор;
 - b) user (тип: int) – идентификатор пользователя;
 - c) role (тип: int) – идентификатор роли пользователей.
- cms_garfa_users_data – в данной таблице хранится дополнительная информация о пользователе системы
 - a) id (тип: int) – идентификатор;
 - b) user (тип: int) – идентификатор пользователя;
 - c) name(тип: text) – настоящее имя пользователя;
 - d) org1(тип: text) – данные о месте работы пользователя;
 - e) org2(тип: text) – данные об должности пользователя;
 - f) birth (тип: text) – дата рождения пользователя;
 - g) phone(тип: text) – контактный телефон пользователя.
- cms_garfa_users_register – в данной таблице хранится информация о регистрациях пользователей в системе
 - a) id (тип: int) – идентификатор;
 - b) user (тип: int) – идентификатор пользователя;
 - c) ip (тип: text) – ip-адрес регистрации пользователя;
 - d) date (тип: text) – дата регистрации пользователя ;
 - e) time(тип: text) – время регистрации пользователя.

Ко второй категории относятся таблицы с логами доступа к системе, настройками доступа к компонентам и модулям, а также другие пользовательские промежуточные данные, например, сообщения с формой обратной связи.

- cms_garfa_fixed_content_feedback – в данной таблице хранятся сообщения обратной связи пользователей системы
 - a) id (тип: int) – идентификатор;
 - b) name (тип: text) – имя пользователя;
 - c) email (тип: text) – e-mail пользователя ;
 - d) data (тип: text) – текст сообщения пользователя.
- cms_garfa_fixed_content_logs – в данной таблице хранится перечень журналов
 - a) id (тип: int) – идентификатор;
 - b) name (тип: text) – название журнала;
 - c) url (тип: text) – путь к расположению журнала.
- cms_garfa_fixed_content_logs – в данной таблице хранится перечень журналов
 - a) id (тип: int) – идентификатор;
 - b) name (тип: text) – название журнала;
 - c) url (тип: text) – путь к расположению журнала.
- cms_garfa_fixed_content_settings_ftp – в данной таблице хранится перечень настроек ftp-доступа к сервисам
 - a) id (тип: int) – идентификатор;
 - b) page (тип: int) – идентификатор страницы исполняемого модуля представления системы, на которой представлен интерфейс доступа к сервису;
 - c) login (тип: text) – имя пользователя по протоколу ftp;
 - d) pass (тип: text) – пароль пользователя по протоколу ftp;
 - e) serv (тип: text) – адрес сервиса по протоколу ftp;
 - f) serv (тип: text) – путь к расположению сервиса по протоколу ftp.
- cms_garfa_fixed_content_settings_pages – в данной таблице хранится перечень имен настроек ftp-доступа к сервисам
 - a) id (тип: int) – идентификатор;
 - b) name (тип: text) – имя настройки.

3.4. Основные классы компонента

Ниже приводятся сокращенные описания структуры и методов основных классов компонента CLAVIRE/AdminTool.

3.4.1. Класс *mod_admin_soap*

Реализация процесса обработки данных, переданных по протоколу SOAP со стороны других сервисов комплекса.

Открытые методы

- month_show – конвертация языковой локализации представления месяца
 - a) входной параметр month (тип:string) – месяц;
 - b) возвращает конвертированное значение (string).
- show_tru_time – конвертация времени в общий формат
 - a) входной параметр min (тип:int) – значение времени в минутах;
 - b) возвращает значение времени в общем формате (string).
- newsoap – создает новый класс клиента, подключенного по протоколу SOAP к удаленному web-сервису
 - a) входной параметр WSDL (тип: string) – полный адрес к wsdl-файлу в формате XML web-сервиса, работающего по протоколу SOAP;
 - b) входной параметр PROXY_HOST (тип: string) – адрес, по которому расположен web-сервис, указанный в параметре WSDL;
 - c) входной параметр PROXY_PORT (тип: string) – порт, по которому расположен web-сервис по адресу, указанному в параметре PROXY_HOST;
 - d) возвращает новый класс (class).

3.4.2. Класс *mod_content*

Реализация функций работы с контентом.

Открытые методы

- footer_login_auth – отображение интерфейса доступа к авторизации пользователя
 - a) возвращаемое значение отсутствует (void).
- check_auth – имеет ли авторизованный пользователь идентификатор.
 - a) возвращает идентификатор (id).
- check_auth_admin – имеет ли авторизованный пользователь доступ к контенту администратора.
 - a) возвращает значение запроса (boolean).

- `interpr_ws` – возвращает наименование исполняемого модуля представления системы.
 - a) входной параметр `ws` (тип: `string`) – идентификатор исполняемого модуля представления системы;
 - b) возвращает наименование исполняемого модуля представления системы (`string`).

4. ИСПОЛЬЗУЕМЫЕ ТЕХНИЧЕСКИЕ СРЕДСТВА

Компонент CLAVIRE/AdminTool работает на серверной ЭВМ на базе ОС Linux с установленным web-сервером Apache, установленной средой PHP (версии 5.2 и выше) с поддержкой стандартного набора библиотек и характеристиками, обеспечивающими стабильную работу ОС и web-сервера в процессе организации последним доступа к сервисам. Для запуска и работы основных модулей требуется x86-совместимый компьютер с тактовой частотой процессора не ниже 500 МГц и оперативной памятью не ниже 256 МБ, на котором должен быть установлен интернет-браузер Google Chrome, Mozilla Firefox, Opera или Safari (последних версий) с включенной поддержкой JavaScript. Для взаимодействия с другими компонентами системы требуется наличие выхода в Интернет или локальную сеть (если сервисы других подсистем доступны из локальной сети) с соответствующей поддержкой со стороны оборудования.

5. ВЫЗОВ И ЗАГРУЗКА

Установка и настройка работы компонента осуществляется на базе web-сервера Apache (версии 2.2.16 и выше) установленной средой PHP (версии 5.2 и выше) с поддержкой стандартного набора библиотек, установленных стандартным образом в соответствии с документацией на указанные продукты. Для корректной работы компонента необходимо наличие следующих файлов, входящих в дистрибутив компонента:

- конфигурационного файла `config.php`, содержащего базовые настройки запуска компонента;
- файла `index.php` – входная точка запуска компонента;
- директория `cms`, содержащая файлы и директории, описывающие модель системы управления контентом
 - a) `classes` – основные классы модели системы;
 - b) `mod` – реализация модели системы;

- директория content, содержащая файлы и директории, описывающие представление системы управления контентом
 - a) classes – основные классы представления системы;
 - b) files – хранилище пользовательских файлов;
 - c) mod – реализация представления системы;
 - d) pages – реализация исполняемых модулей представления системы;
 - e) popup – реализация всплывающих окон исполняемых модулей представления системы;
 - f) scripts – реализация модулей интерфейса и их параметров на языке JavaScript;
 - g) tpl – реализация вида системы: макеты шаблонов, стили и элементы дизайна интерфейса.

Запуск компонента осуществляется при помощи интернет-браузера, посредством ввода в строку адреса полного URL пути к файлу входной точки запуска index.php без указания имени самого файла. Для примера, на рис. 5.1 файл index.php расположен по указанному URL: <http://site1.escience.ifmo.ru>.

После успешного запуска компонента для доступа к функциям администрирования пользователю необходимо авторизоваться. Для авторизации ему следует иметь учетную запись с правами администратора, которая состоит из двух параметров: имя пользователя и пароль.

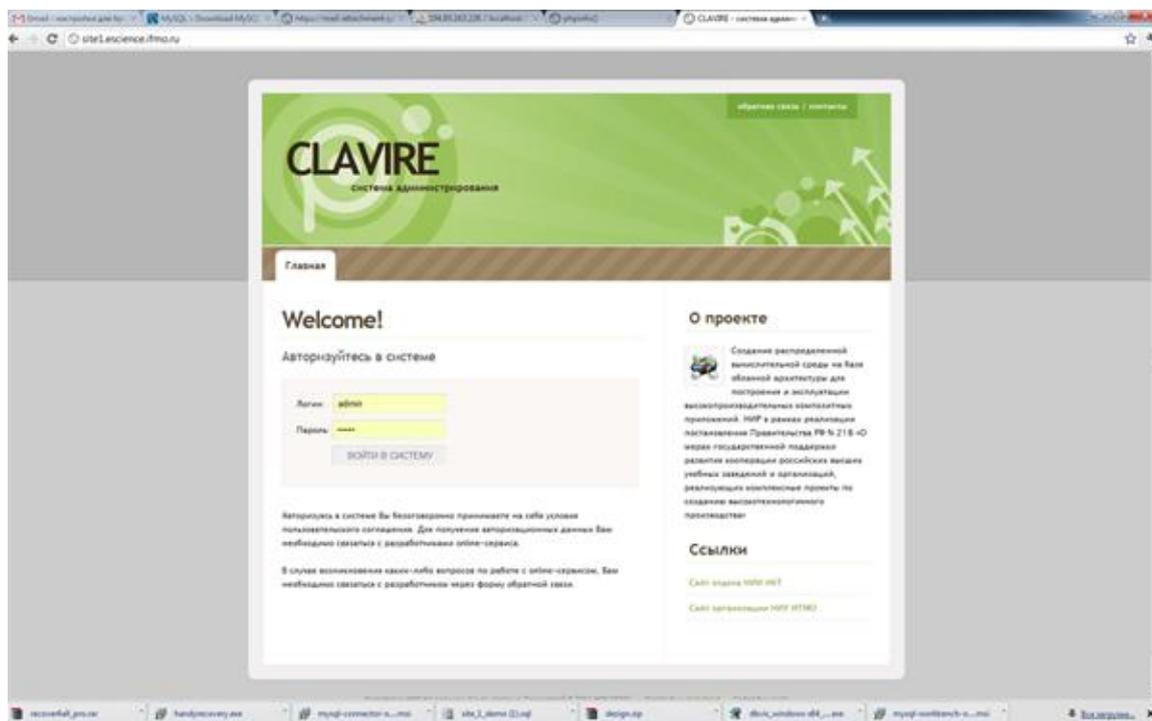


Рисунок 5.1 – Вызов компонента

6. ВХОДНЫЕ ДАННЫЕ

Для запуска и работы с компонентом не требуются специальные виды входных данных. Входные данные, получаемые от других компонентов системы при взаимодействии, задаются в процессе подключения к ним посредством обмена по протоколу обмена структурированными сообщениями в распределенной вычислительной среде SOAP, а также в виде констант и сохраненных ранее пользовательских и конфигурационных данных, хранящихся в СУБД.

7. ВЫХОДНЫЕ ДАННЫЕ

Выходными данными для компонента являются отчеты мониторинга и журналов, а также конфигурационные данные о состоянии и настройках системы, пользователей, ресурсов, пакетов и тарифов, полученные от других компонентов системы и отображаемые в пользовательском интерфейсе.

ПЕРЕЧЕНЬ СОКРАЩЕНИЙ

AJAX	Asynchronous Javascript and XML (асинхронный JavaScript и XML)
CMS	Content Management System (Система управления содержимым)
DOM	Document Object Model (Объектная модель документа)
GPL	General Public License (Открытое лицензионное соглашение GNU)
HTML	HyperText Markup Language (Язык разметки гипертекста)
LGPL	Lesser General Public License (Стандартная общественная лицензия ограниченного применения GNU)
MIME	Multipurpose Internet Mail Extension (Многоцелевое расширение почты)
MPL	Mozilla Public License (Лицензия на свободное ПО)
MVC	Model–View–Controller (Модель–представление–контроллер)
PHP	Personal Hypertext Preprocessor (Препроцессор гипертекста)
SOAP	Simple Object Access Protocol (Простой протокол доступа к объектам)
SQL	Structured Query Language (Язык структурированных запросов)
URL	Uniform Resource Locator (Единый указатель ресурсов)
БД	База данных
МИТП	Многопрофильная инструментально-технологическая платформа
ОС	Операционная система
ПО	Программное обеспечение
СУБД	Система управления базами данных
ЭВМ	Электронно-вычислительная машина

