РУКОВОДСТВО АДМИНИСТРАТОРА

«ЕСИА Wi-Fi»

Ростов-на-Дону

2018 год

Термины, определения, используемые сокращения

Используемые в настоящем документе термины и основные понятия области автоматизированных систем определены в ГОСТ 34.003-90.

АИС, ИС	Автоматизированная информационная система
АРМ (РМ)	Автоматизированное рабочее место - комплекс программных средств для решения определенной задачи
БД	База данных
ВИС	Ведомственная информационная система
ЕПГУ	Единый портал государственных и муниципальных услуг
ЕСМЭВ	Единая система межведомственного электронного взаимодействия
ИС	Информационная система
КЦОД	Коллективный центр обработки данных
МВ	Межведомственное взаимодействие
МО	Муниципальное образование
МФЦ	Многофункциональный центр
ОИВ	Орган исполнительной власти
ОМСУ	Орган местного самоуправления
ПМИ	Программа и методика испытаний
РИЭП	Региональная инфраструктура электронного правительства
РОИВ	Региональный орган исполнительной власти
РСМЭВ	Региональный сегмент единой СМЭВ
РСК	Региональный сервисный концентратор
СИР	Система исполнения регламентов
СМЭВ	Система межведомственного электронного взаимодействия
ТЗ	Техническое задание
ТКМВ	Технологическая карта межведомственного взаимодействия
ЭП	Электронная подпись
ЭП ОВ	Электронная подпись информационной системы органа власти
ЭП СП	Электронная подпись служебного пользования (электронная подпись государственного служащего)

Аннотация

Руководство рассчитано на системных администраторов со следующими компетенциями:

- навыки работы с операционной системой Альт Линукс СПТ 6.0**;**
- понимание XML и механизмов составления XML Schema;
- общее представление о функционировании веб-сервисов;
- навыки администрирования СУБД MySQL 5.0 и выше.

1 Общая информация

1.1 Предназначение системы

В рамках проведения работ по созданию Системы автоматизируется деятельность в следующих процессах:

- авторизация пользователей внешних информационных систем через ЕСИА;

1.2 Требования к системе

Система состоит из трех компонентов:

1. Клиент, в качестве которого выступает веб-браузер (рекомендуется Firefox версии 47 и выше, устанавливается клиентом).
2. Сервер базы данных PostgreSQL (версии 9 или выше, устанавливается администратором системы).
3. Сервер приложений (устанавливается администратором системы в соответствии с этой инструкцией).

Для работы клиента годится любая современная операционная система. Для комфортной работы рекомендуется:

* процессор – от 1 Ghz;
* оперативная память – от 1 Gb;
* свободное место на жестком диске – от 20 Gb.

Для сервера базы данных:

* процессор – от 2.5Ghz;
* оперативная память – от 4 Gb.

Для сервера приложений:

* процессор – от 2.5 Ghz, 64 бит;
* оперативная память – от 4 Gb.

Сервер базы данных и сервер приложений, могут устанавливаться на одном физическом сервере.

Установка и настройка

Дистрибутив состоит из 3-х файлов:

1. lod.txt – этот файл;

2. lod.zip – архив системы;

3. lod.pybundle – библиотеки.

2. Установка и настройка системы

2.1 Настройка рабочей среды

Для установки системы необходима ОС на ядре Linux.

Система написана на языке Ruby и для её работы необходим интерпретатор языка. Для этого рекомендуется использовать систему управлениями версиями *RVM.*

Установка *RVM:*

$ curl -sSL https://get.rvm.io | bash -s stable

Следует внимательно прочесть всю информацию, которая будет подаваться на стандартный вывод в процессе установки. В ней содержаться рекомендации по дополнительной настройке и оптимизации конкретной системы, на которую идет установка. После установки необходимо добавить строку инициализации системы управления версиями *RVM* в файл определения поведения командной оболочки(конкретные инструкции характерные для текущеё системы установщик отобразит в конце установки). На большинстве *linux*-систем это файл *~/.bashrc:*

$ cat 'source /etc/profile.d/rvm.sh' >> ~/.bashrc

Затем необходимо установить требуемую версию интерпретатора языка Ruby

$ rvm install 2.0

Начнется установка интерпретатора. Программа установки автоматически выберет стабильную подминорную версию, к примеру 2.0.0p247. Актуальный и стабильная подминорная версия интерпретатора гарантия отсутствия известных уязвимостей в системе.

После установки интерпретатора, необходимо установить сервер приложений, способный обрабатывать запросы на языке Ruby. Рекомендуется использовать сервер приложений *Passenger*.

Процесс установки сервера приложений *Passenger:*

$ gem install passenger

Данная команда скачает и установит в систему расширение языка Ruby, которое реализует функционал сервер приложений *Passenger*. Зачастую сервер приложений запускается на сервере локально, без доступа к нему из-вне, а роль веб-сервера выполняет *Apache*. Пакет установки сервера приложений passenger имеет в своем составе специальный модуль для *Apache*, интепретирующий *Ruby*-код. Для установки этого модуля следует запустить программу *passenger-install-apache2-module.* Так же как и в случае установки системы управления версиями *RVM,* следует внимательно читать выводимую информацию. Программа установки перечислит все необходимые для своей работоспособности компоненты системы и предложит команды установки, соответствующие установленой в операционной системе системы управления пакетами (*apt-get, emerge* и др.). Проверив наличие всех необходимых компонентов и скомпилировав модуль под соответствующую версию веб-сервера *Apache*, программа установки приведет пример конфигурации *Apache* для обработки *Ruby* запросов:

LoadModule passenger_module /usr/local/rvm/gems/ruby-2.0.0-p247/gems/passenger-4.0.14/buildout/apache2/mod_passenger.so

PassengerRoot /usr/local/rvm/gems/ruby-2.0.0-p247/gems/passenger-4.0.14

PassengerDefaultRuby /usr/local/rvm/wrappers/ruby-2.0.0-p247/ruby

Для конфигурации хоста в *Apache* появляются такие команды как *RailsEnv* и *PassengerMinInstances.* Первая отвечает за окружение запуска приложения *(development, production, test),* вторая за минимальное количество потоков *passenger* обрабатывающих запросы. Обязательным условием

настройки *Apache* на интерпретирование исходных кодов системы является указание в качестве *DocumentRoot* пути к папке *public* внутри кода системы, например */www/is/public*.

После настройки окружения и веб-сервера необходимо скопировать папку с исходными кодами системы в папку веб-сервера, указанную при конфигурировании. После копирования необходимо установить расширения языка Ruby, которые использует система. Это можно сделать используя систему управления зависимостями *bundler*. Команда установки зависимостей выглядит следующим образом:

$ bundle install

Программа автоматически определит необходимые версии расширений языка, скачает и установит их. В процессе также необходимо следить за советами по настройке и оптимизации системы. В случае если по прошествии некоторого времени некоторые версии расширений устареют, необходимо выполнить команду:

$ bundle update

которая произведет обновления всех расширений языка, от которых зависит система. Следует помнить, что с некоторыми версиями расширений система имеет сильную связность, вследствие чего возможность обновить их без изменения системы отсутствует.

После установки всех необходимых расширений языка следует произвести конфигурирование базы данных. В папке *config* следует создать файл *database.yml* (либо использовать существующий) со следующим содержимым

```
# SQLite version 3.x
#   gem install sqlite3
#
#   Ensure the SQLite 3 gem is defined in your Gemfile
#   gem 'sqlite3'
```

```
development: &development
 adapter: mysql2
 database: <dbname>
 username: <user>
 password: <pass>
 socket: /var/run/mysqld/mysqld.sock
 pool: 15
 timeout: 5000

# Warning: The database defined as "test" will be erased and
# re-generated from your development database when you run "rake".
# Do not set this db to the same as development or production.
test:
 adapter: sqlite3
 database: db/test.sqlite3
 pool: 5
 timeout: 5000
production:
 <<: *development
```

где

dbname - имя базы данных, которую должна использовать система;

user - имя пользователя для подключения к базе данных;

pass - пароль для данного пользователя;

Следует проверить наличие доступа указанного пользователя к указанной БД.

Система имеет в своем составе подсистему управления состоянием базы данных, основанной на методологии миграций. Для создания базы данных следует выполнить команду

$ rake db:create

После чего запустить процесс наложения миграций

$ rake db:migrate

Эти команды создадут базу данных и необходимую структуру таблиц для функционирования системы. Затем необходимо заполнить базу данных предопределенными значениями констант:

$ rake db:seed

После прохождения шагов система готова к работе.

При запуске системы в режиме окружения *production* может потребоваться прекомпилирование таблиц стилей, клиентских скриптов и элементов отображения. Такое требование существует для оптимизации клиент-серверного взаимодействия в процессе работы системы и ускорения обработки запросов. Выполняется этот шаг следующим образом:

$ rake assets:precompile RAILS_ENV=production

2.2 Настройка доступа пользователя к СУБД PostgreSQL

1. Редактируем /etc/postgresql/<версия>/main/pg_hba.conf дописываем

после строчки::

local all postgres peer

строку::

local all lod password

2. Перезапускаем postgres::

$ sudo /etc/init.d/postgresql restart

3. Регистрируем пользователя в СУБД и создаем рабочую базу::

$ sudo -u postgres psql

и в консоли БД::

create user homeuser with password '<пароль>' createdb;

create database lod_db owner homeuser;

create database homeuser owner homeuser;

grant all privileges on tablespace pg_default to homeuser;

4. Редактируем conf.py – указываем пароль для доступа к базе::

...

'NAME': 'lod_db', # Or path to database file if using sqlite3.

'USER': 'homeuser', # Not used with sqlite3.

'PASSWORD': '<пароль>', # Not used with sqlite3.

...

2.3. Настраиваем веб-сервер

1. Проверяем, что включен модуль mod_xsendfile::

$ ls /etc/apache2/mods-enabled/x*

/etc/apache2/mods-enabled/xsendfile.load

В противном случае выполняем::

$ sudo ln -s /etc/apache2/mods-available/xsendfile.load \

/etc/apache2/mods-enabled

2. httpd.conf – часть конфигурационного файла Apache (система запускается на виртуальном хосте), надо включить в главный конфигурационный файл директивой include, либо, для Ubuntu::

$ sudo ln -s /home/homeuser/lod/lod/httpd.conf \

/etc/apache2/sites-available/lod

$ sudo a2ensite lod

$ sudo /etc/init.d/apache2 reload

3. Веб-серверу нужны права на доступ к файлам системы, права на запись в каталоги ./media/attachments, ./media/collect_requests,

./media/xls_templates для сохранения файлов-приложений.

Самый простой способ::

$ chmod -R a+rw ../

2.4. Инициализация

Выполняем настройку и инициализацию БД:

$ ./init

Инициализируем сайт:

$ sudo apache2ctl graceful

2.5. Установка и настройка сервисов

1. установить необходимые пакеты

$ sudo apt-get install openjdk-7-jdk libapache2-mod-jk tomcat7

2. установить и настроить КриптоПро JCP с использованием актуальных ключей и носителей, используя jre из пакета openjdk-7-jdk. Например,

$ chmod a+x install.sh

$ sudo ./install.sh /usr/lib/jvm/java-7-openjdk-$(dpkg-architecture -qDEB_BUILD_ARCH)/jre

3. распаковать содержимое архива lod-service-bundle.zip

$ unzip lod-service-bundle.zip

4. отредактировать файл config.properties, заменив значения параметров на актуальные:

database_host - адрес сервера БД в формате хост[:порт]

database_name - имя БД

database_user - имя пользователя БД

database_password - пароль для доступа к БД

key_store - имя контейнера КриптоПро

key_alias - имя хранилища ключей в контейнере

key_password - пароль для доступа к хранилищу

key_authtoken - метка носителя (необязательный параметр)

lod_docs - имя пользователя ЛОД, ответственного за подачу документов

lod_manager - имя пользователя ЛОД, ответственного за ведение лицензионных дел

lod_mediaroot - каталог для хранения внешних файлов ЛОДа

5. запустить скрипт config.sh, передав ему в качестве параметра путь к файлу

httpd.conf развернутого стенда ЛОД

$ chmod a+x config.sh

$ sudo ./config.sh /home/user/lod/lod/httpd.conf

2.6. Настройка модуля ЕСИА

Для настройки модуля ЕСИА, необходимо прописать настройки в файле conf_esia.py. Путь к файлу конфигурации: home/<user_name>/lod/esia. Далее по тексту, значения, которые необходимо изменить выделены жирным.

1. Конфигурирование ключей для работы с ЕСИА:

1.1. В каталог home/<user_name>/lod/esia необходимо скопировать файл с приватным ключом и файл с сертификатом в формате pem.

1.2. Необходимо прописать названия файлов в конфигурационном файле.

Закрытая часть ключа происывается в секции: 'key_file'. Открытая часть – в секции: 'cert_file'. Пример:

'key_file': path.join(BASEDIR, '**myesia.key**'), # private part

'cert_file': path.join(BASEDIR, '**myesia.pem**'), # public part

2. Указать модулю ЕСИА файл с метаданными поставщика идентификации. По умолчанию в комплект поставки системы входит файл с метаданными тестового поставщика идентификации ЕСИА, указанного в методических рекомендациях по интеграции с ЕСИА. Что бы указать метаданные поставщика идентификации нужно сохранить его в каталоге модуля esia и прописать ссылку на него в конфигурационном файле в секции ‘metadata’ подсекция ‘local’

Пример:

'metadata': {

'local': [path.join(BASEDIR, '**shibboleth.xml**')],

},

3. Указываем имя поставщика услуг, которое выдается при регистрации системы в ЕСИА. Секция 'entityid'.

4. Настраиваем урл’ы поставщика услуг (ЛОД). В секции 'service' в подсекции 'sp' два узла отвечают за настройку урл’ов:

· 'assertion_consumer_service' – урл на который служба ЕСИА присылает ответ на запрос авторизации;

· 'single_logout_service' – урл на который есиа ответ на запрос разрыва соединения – logout.

Пример:

'sp' : {

'name': 'lod',

'endpoints': {

'assertion_consumer_service': [

```
            ('http://<сервер>/esia/acs/',
             saml2.BINDING_HTTP_POST),
            ],
        'single_logout_service': [
            ('http:// <сервер>/esia/ls/',
                saml2.BINDING_HTTP_REDIRECT),
            ],
    },
```

5. Настраиваем урл’ы поставщика идентификации. В секции 'service' в подсекции 'idp' два узла отвечают за настройку урл’ов:

· 'single_sign_on_service' – урл на который отправляется запрос на авторизацию. По-умолчанию указывает на URL тестовой ЕСИА, в соответствии с методическими рекомендациями по интеграции с ЕСИА.

· 'single_logout_service' – урл отправляется запрос на разрыв соединения. По-умолчанию указывает на URL тестовой ЕСИА, в соответствии с методическими рекомендациями по интеграции с ЕСИА.

6. Данные о компании и контактных лицах. Секция 'contact_person' – контактные лица, где:

- 'company' – имя комании
- 'email_address' – электронный адрес контактного лица.
- 'contact_type' – тип контактного лица

Секция 'organization' – данные о компании, где:

- 'name' – имя компании
- 'display_name' – имя компании
- 'url' – урл комании

2.7. Резервное копирование

Резервное копирование БД осуществляется средствами СУБД::

```
$ pg_dump -U homeuser -C lod_db > lod.dump
```

Получили дамп БД. Резервное копирование загруженных файлов (каталогmedia) и дампа надо проводить средствами системы в соответствии спринципами организации.

Для восстановления надо настроить систему (пропустив запуск ./init),

восстановить БД и загруженные файлы. Восстановление БД::

```
$ psql -U homeuser < lod.dump
```

.. local variables:

.. mode: rst

.. end:

3. Начало работы в системе

3.1 Вход в систему

Работа в системе осуществляется с помощью интернет-браузера.

При входе в систему (рисунок 1), пользователь вводит логин/пароль, при необходимости устанавливает флаг «Запомнить меня». Если флаг установлен, то при повторном входе в систему ввод логина/пароля не требуется.

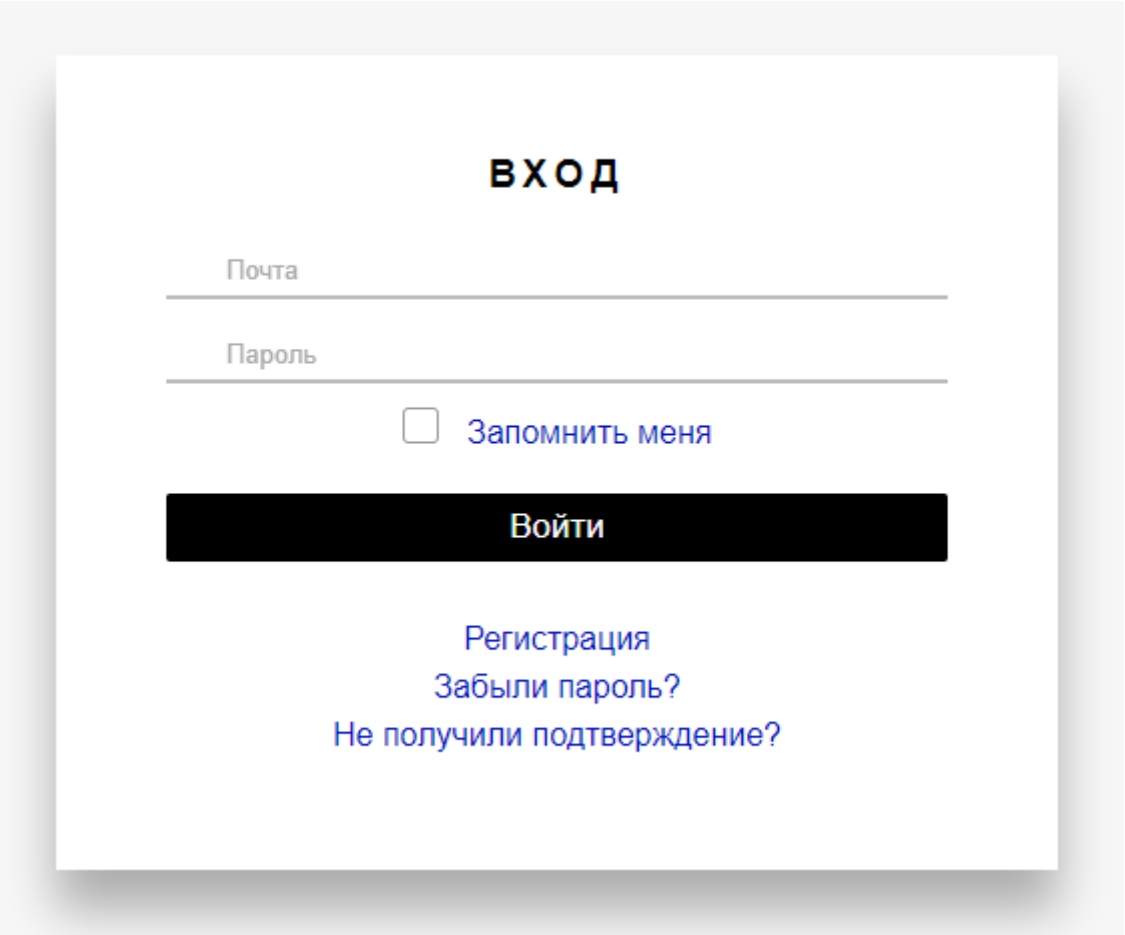

Рисунок 1 - Вход в систему

После входа в систему пользователь может начать работать с Системой.

Рабочая область системы состоит из следующих частей (рисунок 2):

1) Меню перехода между модулями системы.

2) Рабочее окно - отображается информация, в зависимости от выбранного Модуля/Раздела.

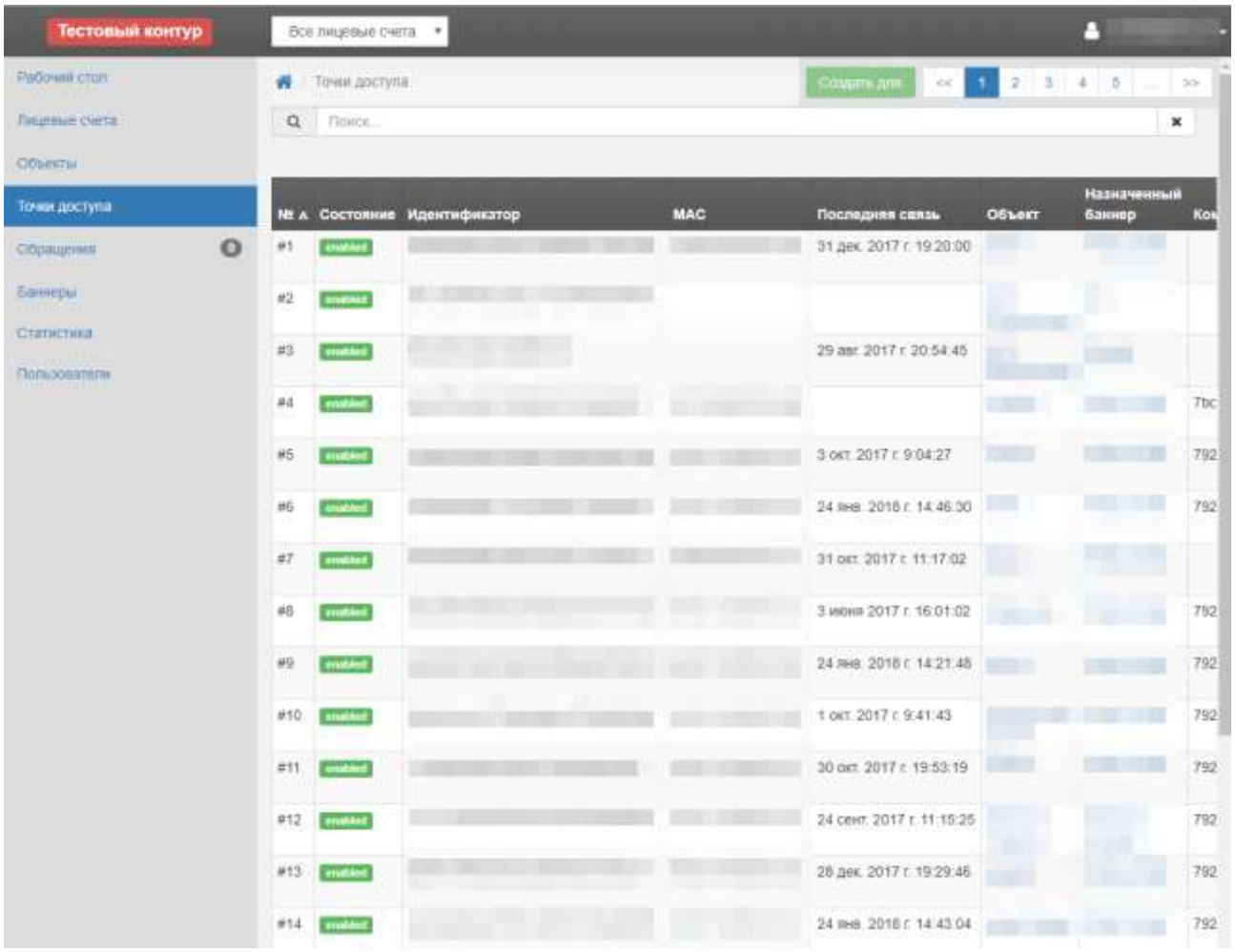

Рисунок 2 - Рабочая область системы