



**Система  
«Галактика ECM.CORP»**

**Инструкция по установке  
и настройке системы**

[galaktika.ru](http://galaktika.ru)  
[galaktika.ru/ecm/](http://galaktika.ru/ecm/)

**2019**

## Оглавление

Установка приложения и вход в систему .....	2
Состав дистрибутива.....	2
Установка БД PostgreSQL .....	2
Настройка удаленного доступа (опционально) .....	2
Создание пользователя и базы данных .....	3
Установка Java .....	4
Настройка приложения.....	4
Запуск приложения .....	4
Проверка работоспособности приложения .....	4
Настройка модели данных .....	5
Конфигурация приложения.....	8
Проверка рабочего стола пользователя .....	14

# Установка приложения и вход в систему

Для установки приложения необходимо:

- Установить СУБД “PostgreSQL 9”
- Установить Java
- Произвести конфигурацию и запуск приложения

## Состав дистрибутива

Дистрибутив приложения Galaktika ECM.CORP.zip состоит из следующих файлов:

- galaktika-ecm-corp-1.3.x.war - запускаемый дистрибутив
- / db - папка с SQL скриптами для создания БД

Необходимо создать на диске папку ecm и распаковать в нее содержимое дистрибутива.

Пример:

```
unzip Galaktika ECM.CORP.zip -d ecm/Galaktika ECM.CORP
```

## Установка БД PostgreSQL

Для операционной систем семейства Debian

```
sudo apt-get install postgresql
```

Для операционной систем семейства Windows

Скачать и установить дистрибутив с сайта <https://www.postgresql.org/>

## Настройка удаленного доступа (опционально)

Если сервер БД находится на отдельном сервере, необходимо разрешить удаленное подключение к нему.

В файле postgresql.conf найти строку

```
#listen_addresses = 'localhost' и заменить ее на listen_addresses = '*'
```

(вместо \* могут быть IP адреса)

В файле pg.hba.conf добавить в конце строчку

```
host all all 192.168.7.0/24 md5
```

для того, чтобы разрешить доступ к БД внешним хостам.

**Пример скрипта для автоматизации предоставления удаленного доступа**

```
sed -i '/listen_address/alisten_addresses = \x27*\x27'
/etc/postgresql/9.5/main/postgresql.conf
echo 'host all all 192.168.7.0/24 md5' >> /etc/postgresql/9.5/main/pg_hba.conf
service postgresql restart
```

После завершения настройки удаленного доступа необходимо перезапустить postgres.

## **Создание пользователя и базы данных**

Необходимо создать пользователя, под которым приложение будет подключаться к БД

Создание БД и запуск скриптов можно производить с использованием утилиты psql либо с помощью программы pgAdmin

Пример создания пользователя и БД утилитой psql

```
su - postgres
psql
CREATE ROLE g_ecm WITH LOGIN PASSWORD 'test123' CREATEDB
VALID UNTIL 'infinity';
CREATE DATABASE g_ecm WITH ENCODING='UTF8' CONNECTION
LIMIT=-1 OWNER g_ecm;
```

Необходимо создать базу данных, для этого нужно выполнить скрипты из папки /db дистрибутива.

**Важно!** Скрипты нужно запускать под пользователем БД `g_ecm`

Пример запуска скриптов утилитой `psql`

```
psql -U g_ecm -h localhost g_ecm -f /<path>/1__base_descriptions.sql
```

## Установка Java

Необходимо установить 'Oracle SDK 8' или 'OpenJDK 8'.

## Настройка приложения

Для настройки приложения следует указать параметры доступа к базе данных, созданной на предыдущем этапе. Для этого в файле `'application.groovy'` в разделе

`'environments.development.dataSource'` следует установить правильные значения для следующих ключей: `'url'` (IP-адрес или имя сервера и порт СУБД PostgreSQL, имя базы данных), `'username'`, `'password'`.

## Запуск приложения

Выполнить shell-скрипт `«run.sh»` или следующую команду:

```
```sh
java -jar galaktika-ecm-corp-1.3.x.war
```
```

Либо `«run.bat»` в зависимости от типа операционной системы.

## Проверка работоспособности приложения

Открыть в браузере ссылка <http://localhost:8081>

Ожидаемый результат: открылась страница входа в приложение

Войти в приложение под пользователем по умолчанию, логин и пароль - admin / test123

Ожидаемый результат: сообщение “Для данного интерфейса и пользователя отсутствует конфигурация папок. Перейдите в интерфейс Системный администратор для изменения конфигурации.”

## Настройка модели данных

1. Выбираем «Редактор классов» на левой панели. Нажимаем на ссылку. В браузере откроется отдельная закладка с редактором классов.
2. Встаем на «Базовый класс» и нажимаем кнопку «Добавить описание класса»
3. В структуре классов появится новый класс, справа необходимо указать его свойства как приведено ниже

|                       |                              |
|-----------------------|------------------------------|
| Идентификатор:        | 2                            |
| Является виртуальным: | <input type="checkbox"/> Нет |
| Системное имя:        | Document                     |
| Название:             | Документ                     |
| Тип:                  | DOCUMENT                     |
| Тип хранилища:        | CE                           |

4. Нажимаем кнопку «Сохранить». Подтверждаем сохранение.
5. Далее встаем на класс «Документ» и нажимаем кнопку «Добавить описание свойств».

6. Добавляем следующие свойства:

| Системное имя | Наименование | Тип Данных | Значение по умолчанию | Свойства                   |
|---------------|--------------|------------|-----------------------|----------------------------|
| Date          | Дата         | Date       |                       |                            |
| Description   | Описание     | STRING     |                       | Большая колонка            |
| Sum           | Сумма        | FLOAT      | 0                     | Обязательно для заполнения |

|                                            |                                   |
|--------------------------------------------|-----------------------------------|
| Идентификатор:                             | 3                                 |
| Является виртуальным:                      | <input type="checkbox"/> Нет      |
| Является системным:                        | <input type="checkbox"/> Нет      |
| Системное имя:                             | <input type="text" value="Date"/> |
| Название:                                  | <input type="text" value="Дата"/> |
| Тип данных:                                | <input type="text" value="DATE"/> |
| Значение по умолчанию:                     | <input type="text"/>              |
| Системное имя справочника:                 | <input type="text"/>              |
| Метод отображения:                         | <input type="text"/>              |
| Пользовательская конфигурация поля (JSON): | <pre>null</pre>                   |
| Только для чтения:                         | <input type="checkbox"/>          |
| Обязательно для заполнения:                | <input type="checkbox"/>          |
| Является скрытым:                          | <input type="checkbox"/>          |
| Является множественным:                    | <input type="checkbox"/>          |
| Сортируемое:                               | <input type="checkbox"/>          |

7. Выбрать в Справочники – Пользователи свойство AccessRule. Снять галочку «Обязательно для сохранения». Сохранить.




8. Закрываем редактор классов и возвращаемся на рабочий стол системного администратора.

## Конфигурация приложения

1. Создаем пользователя со следующими свойствами

|                     |                                                          |
|---------------------|----------------------------------------------------------|
| * Login             | <input type="text" value="user1"/>                       |
| * Пароль            | <input type="password" value="....."/>                   |
| * Имя               | <input type="text" value="user1"/>                       |
| * Роли пользователя | <input type="text" value="* Пользователь"/>              |
| * Рабочий стол      | <input type="text" value="* Рабочий стол пользователя"/> |

2. На панели переходим в раздел «Поиски». Нажимаем кнопку «Новый документ» ().

Создаем конфигурацию с названием «DefaultSearch»

Данные конфигурации:

```
{
  "criteria": {
    "id": "criteria_1",
    "binding": "AND",
    "conditions": [
      {"id": "IsCurrentVersion", "field": "IsCurrentVersion", "visible": false,
"condition": "EQUAL", "value": true},
      {"id": "Title", "field": "Title", "visible": true, "condition": "LIKE"}
    ]
  }
}
```

3. Переходим в раздел «карточки» и создаем конфигурацию «Document».

Данные конфигурации:

```

        {
    "panels": [
      { "legend": "System fields",
        "componentsRows": [
          { "components": [ { "cssClass": "span4", "propName": "Author" },
            { "cssClass": "span4", "propName": "DateCreated" } ] },
          { "components": [ { "cssClass": "span4", "propName":
"LastModifier" }, { "cssClass": "span4", "propName": "DateLastModified" }
          ] }
        ]
      },
      { "legend": "Основное",
        "componentsRows": [
          { "components": [ { "cssClass": "span10", "propName": "Title" } ] },
          { "components": [ { "cssClass": "span10", "propName": "Date" } ] },
          { "components": [ { "cssClass": "span10", "propName":
"Description" } ] },
          { "components": [ { "cssClass": "span10", "propName": "Sum" } ] }
        ]
      }
    ]
  }

```

4. Переходим в раздел «Папки» и создаем конфигурацию «Document».

Данные конфигурации:

```

{
  "label": "Документ",
  "caseName": "Document",
  "type": "CE",
  "inboxView": "Document",
  "roles": ["appUser"],
  "search": "DefaultSearch"
}

```

5. Переходим в раздел «Группы папок» и создаем конфигурацию для «Рабочий стол пользователя» (необходимо выбрать из выпадающего списка).

Данные конфигурации:

```
{
  "ecm.meta.generator.inboxGroup.defaultName" : {
    "class" : "com.galantis.ecm.model.inbox.InboxGroup",
    "label" : "Документы",
    "inboxes" : [ "Document" ]
  }
}
```

6. Переходим в раздел «Отображение папок» и создаем конфигурацию «Document»

Данные конфигурации:

```
{
  "class" : "com.galantis.ecm.model.inbox.InboxView",
  "fields" : [ {
    "fieldName" : "Author"
  }, {
    "fieldName" : "Title"
  }, {
    "fieldName" : "DateCreated"
  } ],
  "orderBy" : {
    "fieldName" : "DateCreated",
    "direction" : "DESC"
  }
}
```

```
}
```

7. В разделе «Пользовательская конфигурация» выбираем справа «tabs».

Откроется конфигурация «tabs». В нее нужно добавить:

```
comments: new Tab(  
    label: 'Комментарии',  
    controller: 'historyTab',  
    params: [view: 'index_comments', searchCriteria: new  
Criteria(Criteria.AND, [new Condition("OperationType",  
ConditionType.EQUAL, "comment.added")])],  
    classStyle: 'icon-comment-alt'  
)
```

После добавления необходимо сохранить.

8. Переходим в раздел «Скрипты» и создаем две конфигурации

а. Скрипт «tabsScript»

Данные конфигурации:

```
import com.galantis.ecm.meta.DefaultTabsCalculator  
import com.galantis.ecm.api.object.model.ValueObject  
import com.galantis.ecm.meta.context.CalculatorContextForTab  
import  
org.springframework.web.context.request.RequestContextHolder  
//import com.galantis.ecm.auth.type.Desktop  
import com.galantis.ecm.common.auth.type.Desktop
```

```
class CommonTabCalculator extends DefaultTabsCalculator {  
    @Override  
    Collection<String> getCaseTabs(CalculatorContextForTab context)  
    {  
        def session =  
RequestContextHolder.currentRequestAttributes().getSession()  
        def isNew = context.vo.id == null  
  
        def tabs = []  
        switch (context.vo.classType) {  
            case 'Document':
```

```

                tabs << 'details'
    if (!isNew) {
        tabs << 'details'
        tabs << 'content'
        tabs << 'history'
        tabs << 'comments'
    }
    break

    default:
        tabs << 'details'
        if (!isNew ) {
            tabs << 'content'
            tabs << 'history'

        }
    }
    tabs
}
}

```

b. Скрипт «toolsScript»

ДАННЫЕ КОНФИГУРАЦИИ

```

import com.galantis.ecm.meta.*
import com.galantis.ecm.BeanUtils
import com.galantis.ecm.api.auth.model.User
import com.galantis.ecm.api.content.model.Content
import com.galantis.ecm.api.meta.ToolsCalculator
import com.galantis.ecm.api.object.model.ValueObject
import com.galantis.ecm.common.auth.type.Desktop
import com.galantis.ecm.meta.context.CalculatorContext
import com.galantis.ecm.meta.context.CalculatorContextForTab
import
com.galantis.ecm.meta.context.ToolCalculatorContextForContent
import
com.galantis.ecm.meta.context.ToolCalculatorContextForInbox
import com.galantis.ecm.model.inbox.Inbox
import com.galantis.ecm.model.inbox.InboxViewType

class ToolsScript extends DefaultToolsCalculator {

```

```

Collection<String> getTools (ToolCalculatorContextForInbox
context) {
    def result = super.getTools(context)
    result << 'exportToFile'

    if ( context.inboxConfig.name == 'Document') {
        tools << 'addComment'
    }

    result
}

```

```

Collection<String> getTools(CalculatorContextForTab context) {
    def tools = super.getTools(context)
    def isNew = context.vo.id == null

    tools
}

```

```

Collection<String> getTools(ToolCalculatorContextForContent
context) {
    def tools = super.getTools(context)
    tools.remove('deleteContent')
    tools << 'downloadContent'
    tools << 'printContent'

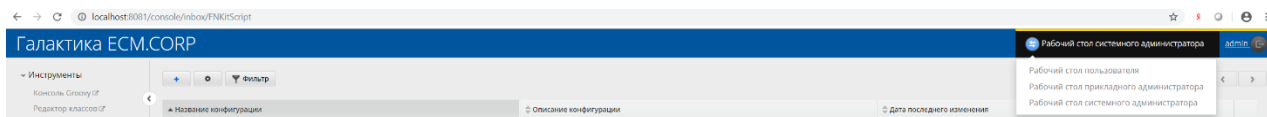
    tools
}
}

```

9. После настройки и сохранения всех конфигураций нажимаем на «Обновить настройки».

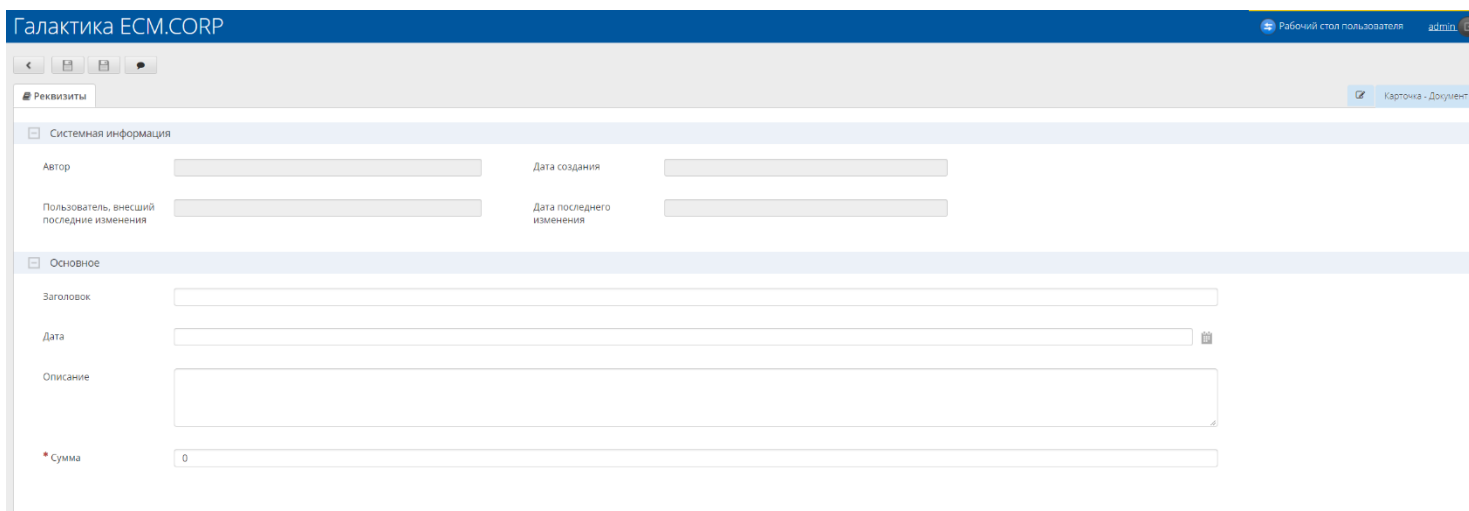
# Проверка рабочего стола пользователя

1. Переключаемся на рабочий стол пользователя.




2. Рабочий стол пользователя должен выглядеть следующим образом

3. Создаем новый документ по «плюсику». Поднимется карточка на заполнение, которая выгядит следующим образом



4. Заполняем поля. Сохраняем документ. После сохранения на карточке появились закладки «Вложение», «История», «Комментарии» и карточка должна выгядеть следующим образом:

5. Добавляем комментарий на карточку. Комментарий появился на закладке «Комментарий»
6. Нажимаем кнопку перехода «Назад» (). Переходим на основной стол пользователя.
7. Добавляем по аналогии еще несколько документов.
8. Далее нажимаем кнопку «Фильтр». Над табличным отображением появится фильтр по заголовку. Заполним фильтр подходящим значением и нажмем «Найти». Фильтр успешно отработал.
9. Перезайдем под пользователем, который был создан ранее. Пользователь имеет доступ только к «Рабочему столу пользователя». А также видит все созданные документы.