

# ГАЛАКТИКА EAM.WEB

## СИСТЕМА

## Галактика EAM.WEB

ИНСТРУКЦИЯ ПО УСТАНОВКЕ СИСТЕМЫ

# Аннотация

В данном документе описан порядок установки, настройки и запуска системы *Галактика EAM.WEB*, разработанной на базе системы *Галактика EAM*.

**Издание 10.2021**

© ООО «Галактика НТЦ», 2021  
Запрещается распространять без согласования  
с собственником ПО

## Содержание

<b>1</b>	<b>Установка, настройка СУБД PostgreSQL</b> .....	<b>4</b>
<b>2</b>	<b>Установка, настройка и запуск сервера приложений</b> .....	<b>6</b>
2.1	Установка сервера приложений и дополнительных компонент .....	6
2.2	Настройка сервера приложений .....	6
2.3	Запуск сервера приложений .....	7
<b>3</b>	<b>Установка, настройка WEB интерфейса и WEB сервера, запуск WEB сервера</b> .....	<b>8</b>
3.1	Установка apache .....	8
3.2	Установка веб-приложения galaktika.start .....	9
3.3	Установка и настройка виртуального хоста приложения galaktika.start .....	10
<b>4</b>	<b>Вход в систему</b> .....	<b>12</b>

# 1 Установка, настройка СУБД PostgreSQL

- 1.1. Установить PostgreSQL выполнив запрос: “apt-get install postgresql”.
- 1.2. Изменить файл “/etc/postgresql/9.6/main/pg\_hba.conf” в соответствии с рисунком 1.

```
# TYPE DATABASE USER ADDRESS METHOD
# "local" is for Unix domain socket connections only
local all all peer
# IPv4 local connections:
#host all all 127.0.0.1/32 md5
host all all 127.0.0.1/32 trust
host all all all trust
```

Рисунок 1. Изменение файла настройки

- 1.3. Установить пароль для учетной записи postgres.

Запустить psql под пользователем «postgres», выполнив команду «sudo -u postgres psql postgres».

Установить пароль: \password postgres

Выход: \q

- 1.4. Установить pgAdmin3 выполнив “apt-get install pgadmin3”.
- 1.5. Зарегистрировать сервер PostgreSQL (в pgAdmin).

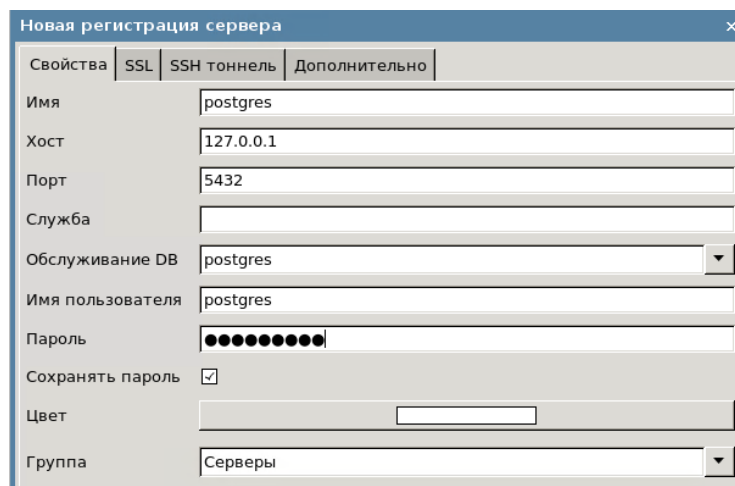
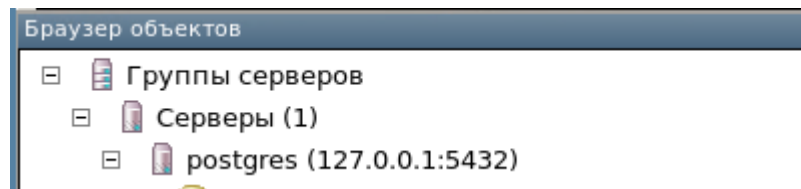


Рисунок 2. Регистрация сервера PostgreSQL

- 1.6. Проверить наличие сервера.



*Рисунок 3. Проверка наличия сервера*

## 2 Установка, настройка и запуск сервера приложений

---

Все настройки производить под root пользователем.

### 2.1 Установка сервера приложений и дополнительных компонент

Для установки сервера приложений на ОС Astra Linux необходимо извлечь содержимое из архива eam.WEB.server.astra.zip в папку сервера приложений (/root/Документы).

#### **Примечание:**

Для работы сервера приложений на ОС Astra Linux требуется наличие фреймворка .NET Core 3.1. При отсутствии .NET Core 3.1 требуется провести работу по его установке согласно официальной инструкции: <https://docs.microsoft.com/en-us/dotnet/core/install/linux-debian>

### 2.2 Настройка сервера приложений

В файле EAM.Web.Server.dll.config ввести данные в соответствии с перечисленными ниже примерами:

- указать адрес и порт, по которым будет доступно API сервера.  
`<add key="DefaultApiServerPort" value="5000" />`  
`<add key="ApiServerHost" value="http://localhost" />`
- указать имя пользователя, с правами которого запустится сервер:  
`<add key="UserName" value="WorkflowService" />`
- указать строку подключения к БД:  
`<add name="ConnectionString"`  
`connectionString="XpoProvider=EAMPostgres;Server=10.2.4.`  
`100;Port=5432;User ID=postgres;Database=Galaktika.EAM" />`

## 2.3 Запуск сервера приложений

Выполнить команду “dotnet EAM.Web.Server.dll -u”.

При запуске сервера с указанием порта через команду ”dotnet EAM.Web.Server.dll -p 5000 -u” настройка в конфигурационном файле игнорируется:

```
root@itsm-23:~/Документы/server# dotnet EAM.Web.Server.dll -p 5000 -u
info: Microsoft.AspNetCore.DataProtection.KeyManagement.XmlKeyManager[0]
      User profile is available. Using '/root/.aspnet/DataProtection-Keys' as key repository; keys will not be encrypted at rest.
info: Microsoft.Hosting.Lifetime[0]
      Now listening on: http://localhost:5000
info: Microsoft.Hosting.Lifetime[0]
      Application started. Press Ctrl+C to shut down.
info: Microsoft.Hosting.Lifetime[0]
      Hosting environment: Production
info: Microsoft.Hosting.Lifetime[0]
      Content root path: /root/Документы/server
```

*Рисунок 4*

При необходимости прервать работу сервера по комбинации клавиш Ctrl+C.

## 3 Установка, настройка WEB интерфейса и WEB сервера, запуск WEB сервера

---

### 3.1 Установка apache

- a) Установить apache выполнив команду “sudo apt-get install apache2”.
- b) В конец файла «/etc/apache2/apache2.conf» добавить строку:

ServerName localhost

```
# Include of directories ignores editors' and dpkg's backup files,
# see README.Debian for details.

# Include generic snippets of statements
IncludeOptional conf-enabled/*.conf

#Include the virtual host configurations:
IncludeOptional sites-enabled/*.conf

# vim: syntax=apache ts=4 sw=4 sts=4 sr noet
ServerName localhost
```

Рисунок 5. Вставка строки в файл «/etc/apache2/apache2.conf»

- c) Перегрузить сервер apache выполнив команду “sudo systemctl restart apache2”.
- d) Проверить состояние apache выполнив команду “sudo systemctl status apache2”.

```
rash@astra:~$ sudo systemctl status apache2
● apache2.service - The Apache HTTP Server
   Loaded: loaded (/lib/systemd/system/apache2.service; enabled; vendor preset: enabled)
   Active: active (running) since Wed 2021-04-07 06:36:03 MSK; 22s ago
     Docs: https://httpd.apache.org/docs/2.4/
   Process: 1822 ExecStop=/usr/sbin/apachectl stop (code=exited, status=0/SUCCESS)
   Process: 1830 ExecStart=/usr/sbin/apachectl start (code=exited, status=0/SUCCESS)
  Main PID: 1835 (apache2)
    Tasks: 55 (limit: 4915)
   CGroup: /system.slice/apache2.service
           └─1835 /usr/sbin/apache2 -k start
             └─1836 /usr/sbin/apache2 -k start
               └─1837 /usr/sbin/apache2 -k start

apr 07 06:36:03 astra systemd[1]: Starting The Apache HTTP Server...
apr 07 06:36:03 astra systemd[1]: Started The Apache HTTP Server.
```

Рисунок 6. Проверка состояния apache



- е) Проверить работоспособность арасче открыв веб-страницу с адресом localhost.

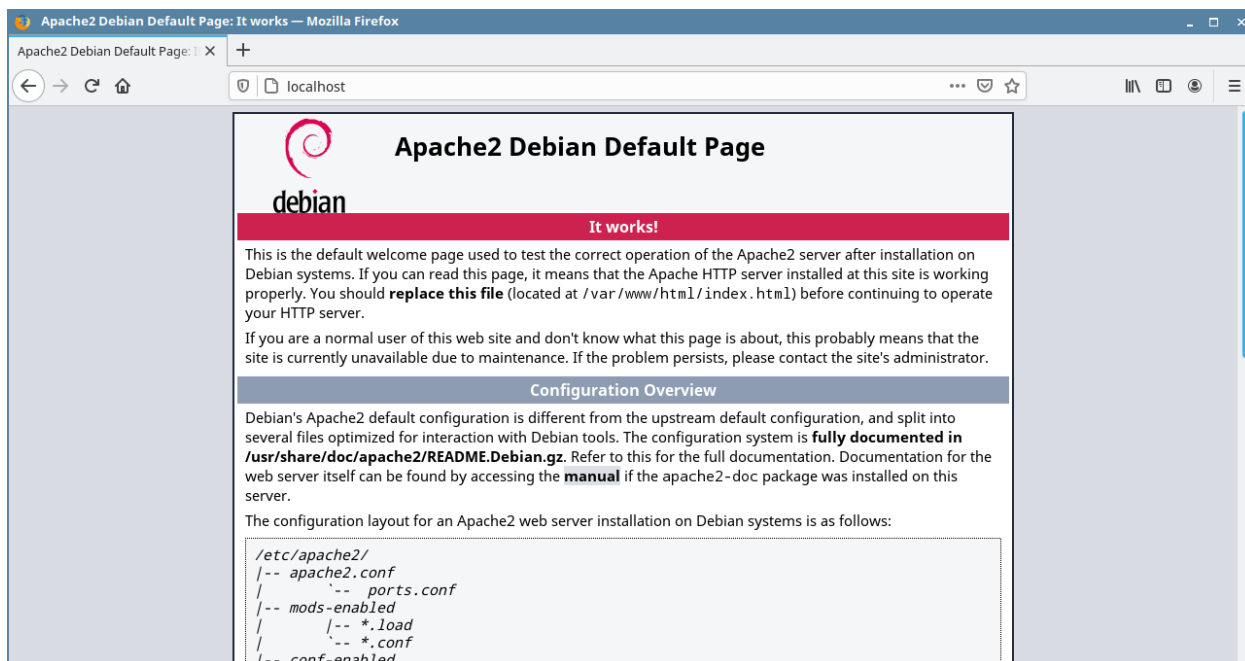


Рисунок 7. Проверка работоспособности

## 3.2 Установка веб-приложения galaktika.start

- а) Приложение galaktika.start использует порт 4400. Для прослушивания этого порта сервером Apache добавить в файл «/etc/apache2/ports.conf» под строкой «Listen 80» строку «Listen 4400». Сохранить файл.

```
# If you just change the port or add more ports h  
# have to change the VirtualHost statement in  
# /etc/apache2/sites-enabled/000-default.conf  
  
Listen 80|  
Listen 4400  
  
<IfModule ssl_module>  
  > Listen 443  
</IfModule>  
  
<IfModule mod_gnutls.c>  
  > Listen 443  
</IfModule>  
  
# vim: syntax=apache ts=4 sw=4 sts=4 sr noet
```

Рисунок 8. Добавление строки в файл «/etc/apache2/ports.conf»

- b) Перезагрузить apache выполнив команду “sudo systemctl restart apache2”.
- c) Проверить прослушиваемые порты выполнив команду “sudo netstat -anp | grep apache”.

```
rash@astra:~$ sudo systemctl restart apache2
rash@astra:~$ sudo netstat -anp | grep apache
tcp6      0      0  :::4400                :::*                LISTEN      2472/apache2
tcp6      0      0  :::80                  :::*                LISTEN      2472/apache2
rash@astra:~$
```

Рисунок 9. Проверка прослушиваемых портов

### 3.3 Установка и настройка виртуального хоста приложения galaktika.start

- a) Создать папку galaktika.start в рабочем каталоге.
- b) Скопировать содержимое архива eam.web.client.zip в созданную папку galaktika.start.
- c) Создать файл «/etc/apache2/sites-available/galaktika.start.conf».

Ввести блок строк в созданный файл:

```
<VirtualHost *:4400>
    ServerAdmin admin@galaktika.start
    ServerName localhost
    ServerAlias www.galaktika.start
    DocumentRoot /var/www/galaktika.start
    ErrorLog ${APACHE_LOG_DIR}/error.log
    CustomLog ${APACHE_LOG_DIR}/access.log combined
</VirtualHost>
```

Сохранить файл.

- d) Проверить конфигурацию выполнив команду “sudo apache2ctl configtest”.
- e) Активировать виртуальный хост выполнив команду “sudo a2ensite galaktika.start.conf”.

- f) Проверить результат. Ввести в браузере адрес localhost:4400.  
Загрузить начальную страницу.

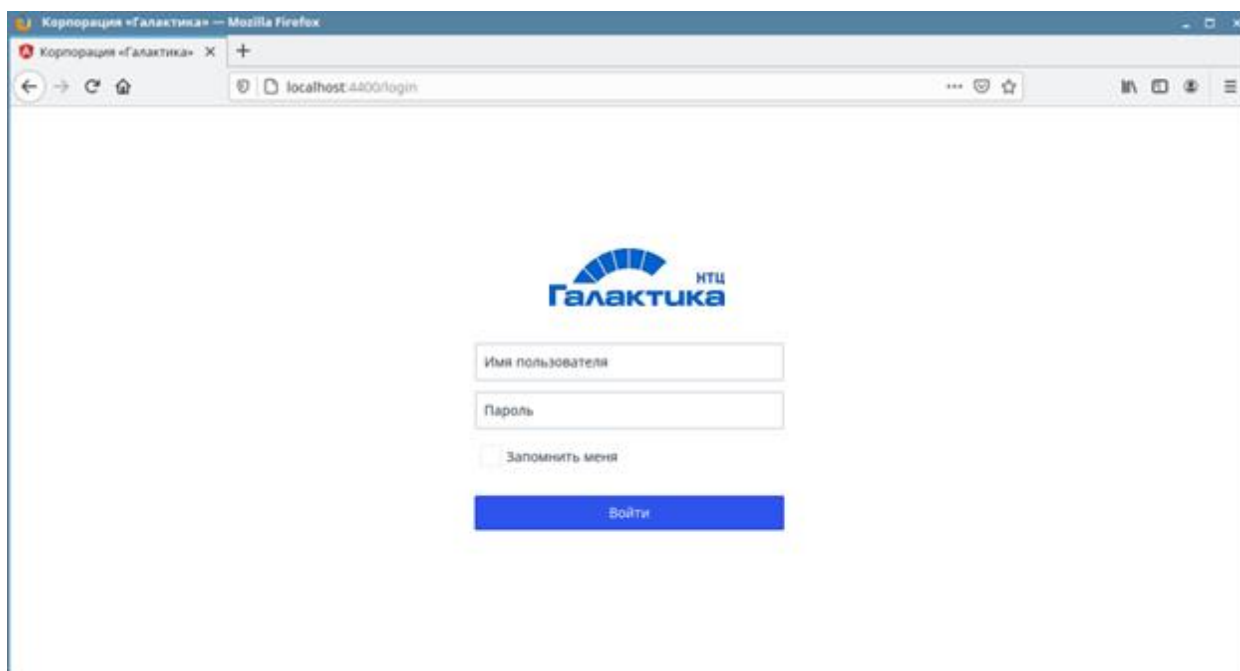


Рисунок 10. Начальная страница «Галактики EAM.WEB»

## 4 Вход в систему

---

Для входа в систему необходимо в браузере указать адрес и порт, соответствующий п.2.6 «Настройка WEB сервера», например, «http://10.2.5.88:4400». Откроется окно базового WEB интерфейса системы для входа в систему (см. Рисунок 21). Ввести имя пользователя и пароль.

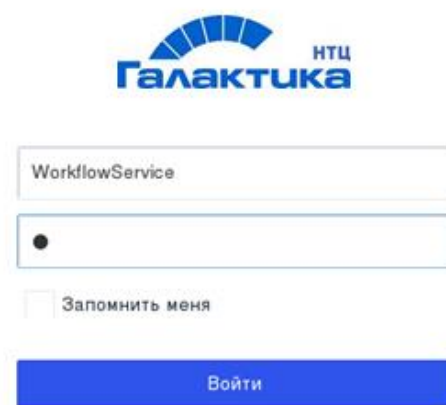


Рисунок 11. Страница входа (Начальная страница)

Обработка действий пользователя так же выводится в консоль запущенного сервера приложений.

```
mc [root@itm-21]~  
Info: Microsoft.AspNetCore.Hosting.Diagnostics[2]  
Request finished in 0.3576ms 204  
Info: Microsoft.AspNetCore.Hosting.Diagnostics[1]  
Request starting HTTP/1.1 POST http://10.2.5.88:5000/api/login application/json 39  
Info: Microsoft.AspNetCore.Cors.Infrastructure.CorsService[4]  
CORS policy execution successful.  
Info: Microsoft.AspNetCore.Routing.EndpointMiddleware[0]  
Executing endpoint 'Galaktika.Web.Server.Controllers.WebApiController.LoginAction (Galaktika.Web.Server)'  
Info: Microsoft.AspNetCore.Mvc.Infrastructure.ControllerActionInvoker[3]  
Route matched with {action = "LoginAction", controller = "WebApi"}. Executing controller action with signature System.Object LoginAction(Galaktika.Web.Server.WebModels.Incoming.LoginRequestModel) on controller Galaktika.Web.Server.Controllers.WebApiController (Galaktika.Web.Server).  
Info: Microsoft.AspNetCore.Mvc.Infrastructure.ControllerActionInvoker[1]  
Executing ObjectResult, writing value of type 'Galaktika.Web.Server.WebModels.Outgoing.LoginResponseModel'.  
Info: Microsoft.AspNetCore.Mvc.Infrastructure.ControllerActionInvoker[2]  
Executed action Galaktika.Web.Server.Controllers.WebApiController.LoginAction (Galaktika.Web.Server) in 60.2206ms  
Info: Microsoft.AspNetCore.Routing.EndpointMiddleware[1]  
Executed endpoint 'Galaktika.Web.Server.Controllers.WebApiController.LoginAction (Galaktika.Web.Server)'  
Info: Microsoft.AspNetCore.Hosting.Diagnostics[2]  
Request finished in 63.2625ms 200 application/json; charset=utf-8  
Info: Microsoft.AspNetCore.Hosting.Diagnostics[1]  
Request starting HTTP/1.1 OPTIONS http://10.2.5.88:5000/api  
Info: Microsoft.AspNetCore.Cors.Infrastructure.CorsService[4]  
CORS policy execution successful.  
Info: Microsoft.AspNetCore.Hosting.Diagnostics[2]  
Request finished in 1.1109ms 204  
Info: Microsoft.AspNetCore.Hosting.Diagnostics[1]  
Request starting HTTP/1.1 POST http://10.2.5.88:5000/api 121  
Info: Microsoft.AspNetCore.Cors.Infrastructure.CorsService[4]  
CORS policy execution successful.  
Info: Microsoft.AspNetCore.Authentication.JwtBearer.JwtBearerHandler[2]  
Successfully validated the token.  
Info: Microsoft.AspNetCore.Authorization.DefaultAuthorizationService[1]  
Authorization was successful.  
Info: Microsoft.AspNetCore.Routing.EndpointMiddleware[0]  
Executing endpoint 'Galaktika.Web.Server.Controllers.WebApiController.DataAction (Galaktika.Web.Server)'  
Info: Microsoft.AspNetCore.Mvc.Infrastructure.ControllerActionInvoker[3]  
Route matched with {action = "DataAction", controller = "WebApi"}. Executing controller action with signature System.Object DataAction(Galaktika.Web.Server.WebModels.Incoming.ActionRequestModel) on controller Galaktika.Web.Server.Controllers.WebApiController (Galaktika.Web.Server).  
Info: Microsoft.AspNetCore.Mvc.Infrastructure.ControllerActionInvoker[1]  
Executing ObjectResult, writing value of type 'Galaktika.Web.Server.WebModels.Outgoing.ActionResponse.Responses.ActionResponseOpenModel'.  
Info: Microsoft.AspNetCore.Mvc.Infrastructure.ControllerActionInvoker[2]  
Executed action Galaktika.Web.Server.Controllers.WebApiController.DataAction (Galaktika.Web.Server) in 84.8532ms  
Info: Microsoft.AspNetCore.Routing.EndpointMiddleware[1]  
Executed endpoint 'Galaktika.Web.Server.Controllers.WebApiController.DataAction (Galaktika.Web.Server)'  
Info: Microsoft.AspNetCore.Hosting.Diagnostics[2]  
Request finished in 92.915ms 200 application/json; charset=utf-8
```

Рисунок 12. Консоль запущенного сервера приложений