# Руководство администратора информационной системы «Работающий регион»

# Содержание

# 1. Общие положения

Руководство описывает требования и порядок работы информационной системы «Работающий регион».

## 1.1. Компоненты системы

Архитектура системы основана на принципах трехуровневой структуры, разделяющей данные, уровень их обработки, и уровень представления (тонкий клиент). База данных и бизнес-приложение должны работать на специально выделенном сервере (или раздельных серверах), тогда как клиентская часть доступна с различных рабочих станций пользователей через локальную сеть или глобальную сеть Интернет.

Система состоит из следующих компонентов:

- серверная часть;
- база данных;
- клиентская часть.

Серверная части системы может работать на одной из следующих операционных систем:

- CentOS 7 Server
- Fedora Linux 30 Desktop;
- Fedora Linux 30 Server;
- Fedora Linux 29 Desktop;
- Fedora Linux 29 Server;
- Ubuntu Linux 18.04 Bionic Beaver.

В качестве интерпретатора используется Java 11.0.7.

База данных работает под управлением СУБД PostgreSQL 11.

На клиентской части доступ к системе может обеспечиваться посредством следующих браузеров:

Internet Explorer;

Mozilla Firefox;

Яндекс Браузер;

Safari для обеспечения возможности работы пользователей на IPad.

Дополнительной установки программного обеспечения на клиентской части не требуется.

### 1.2. Технические характеристики

Система может быть развернута на одном или нескольких серверах.

Аппаратные характеристики серверов должны соответствовать следующим минимальным требованиям:

| Сервера | ОЗУ, Гб | Процессор, ядер | Место на жестком, Гб |
|---|---|---|---|
| Сервер приложения | 8 | 2 | Не менее 400 |
| Сервер баз данных | 8 | 2 | Не менее 250 |

## 2. Администрирование системы

Информационная система «Работающий регион» состоит из трех сервисов:

- сервис подачи заявок,
- сервис обработки заявок,
- сервис проверки заявок и сотрудников.

**Сервис подачи заявок.** Через данный сервис осуществляется подача заявок от организаций.

**Сервис обработки заявок.** Предназначен для принятия или отклонения заявок контролирующими подразделениями региона.

**Сервис проверки заявок и сотрудников.** В данном сервисе осуществляется проверка заявок на статус обработки, а также для проверки сотрудников организаций проверяющими органами региона.

### 2.1. Справочник подразделений

В системе предусмотрен справочник подразделений. Данный справочник используется при подаче заявки, а также для работы со справочником пользователей.

Администрирование справочника выполняется средствами СУБД. Записи по подразделениям хранятся в таблице cls_department. Для того, чтобы добавить подразделение в справочник необходимо открыть базу данных системы и выполнить скрипт по шаблону:

```
INSERT INTO cls_department (id, name, description) VALUES (_id, _name, _description)
```

, где _id – идентификатор в таблице,

_name – название подразделения,

_description – описание подразделения.

Пример:

INSERT INTO cls_department (id, name, description) VALUES (32, ‘Тестовое подразделение’, ‘Подразделение для тестирования’)

## 2.2. Справочник пользователей

Справочник пользователей используется для разделения доступа в сервисе обработки заявок. Для того, чтобы обрабатывать заявки, необходимо пройти аутентификацию. После прохождения аутентификации пользователю открывается доступ к заявкам только того подразделения, к которому он относится.

Администрирование справочника пользователей выполняется средствами СУБД. Записи по пользователям хранятся в таблице dep_user. Для того, чтобы добавить пользователя в справочник необходимо открыть базу данных системы и выполнить скрипт по шаблону:

INSERT INTO dep_user (id, id_department, lastname, firstname, patronymic, login, password) VALUES (_id, _id_department, _lastname, _firstname, _patronymic, _login, _password)

, где _id – идентификатор в таблице,

_id_department – идентификатор подразделения, к которому относится пользователь,

_lastname – фамилия,

_firstname – имя,

_patronymic – отчество,

_login и _password – логин и пароль пользователя для входа в сервис.

Пример:

INSERT INTO dep_user (id, id_department, lastname, firstname, patronymic, login, password) VALUES (52, 32, ‘Иванов’, ‘Иван’, ‘Иванович’, ‘ivanov’, ‘ivanov52’)

## 2.3. Справочник типов заявок

Данный справочник предназначен для разделения заявок по типу. Справочник используется в сервисе подачи заявок и сервисе обработки заявок.

Администрирование справочника выполняется средствами СУБД. Записи по типам заявок хранятся в таблице cls_type_request. Добавление нового типа заявки осуществляется через выполнение скрипта в базе данных по шаблону:

INSERT INTO cls_type_request (id, activity_kind, id_department, prescription, status_registration, status_visible, begin_registration, end_registration)
VALUES (_id, _activity_kind, _id_department, _prescription, _status_registration, _status_visible, _begin_registration, _end_registration)

, где _id – идентификатор в таблице,

_activity_kind – название вида деятельности,

_id_deparment – идентификатор подразделения (если не указать, то список подразделений будет доступен для выбора),

_prescription – текст предписания Роспотребнадзора в формате html,

_status_registration – статус регистрации (0 – закрыта, 1 – открыта),

_status_visible – статус видимости на при проверке (0 – отображать, 1 – не отображать),

_begin_registration – дата начала видимости сотрудников в сервисе проверки,

_end_registration – дата окончания видимости сотрудников в сервисе проверки.

Пример:

```
INSERT INTO cls_type_request (id, activity_kind, id_department, prescription,
status_registration, status_visible, begin_registration, end_registration)
VALUES (20, ‘СПА-салоны, бани, сауны’, 4,
‘<p><strong>Предписание</strong></p>’, 1, 1, '2020-04-06', '9999-12-31 23:59:59')
```