

# Руководство администратора ПО

«Система анализа любимых игр и рекомендации»

## 1. ВВЕДЕНИЕ

### 1.1. Область применения

Настоящий документ предназначен для администраторов, эксплуатирующих программное обеспечение «Система анализа любимых игр и рекомендации» (GameRanker) (далее — «Система»). Руководство содержит порядок установки, настройки, администрирования и взаимодействия с API Системы.

### 1.2. Перечень выполняемых функций администратора/оператора

В перечень выполняемых функций администратора Системы входят:

- Установка и настройка Системы в контейнерной среде
- Настройка переменных окружения и подключение к базе данных PostgreSQL
- Мониторинг работы Системы через health-check эндпоинты
- Анализ структурированных логов операций
- Восстановление работоспособности при сбоях
- Конфигурация параметров безопасности (ACCESS\_KEY)
- Управление пулом соединений с базой данных
- Развертывание и масштабирование в оркестраторах (Docker Swarm, Kubernetes)

### 1.3. Уровень подготовки администратора/оператора

Администратор/оператор (далее по тексту Администратор) Системы должен уметь пользоваться и настраивать среду функционирования контейнеров или систему оркестрации, используемую на предприятии.

Рекомендуемая численность персонала для эксплуатации Системы — 1 штатная единица.

Администраторы Системы должны пройти обязательную общую и специальную подготовку для работы с Системой.

Общая подготовка должна включать в себя получение знаний и навыков работы с Системой в качестве администратора.

Специальная подготовка должна включать в себя получение знаний и навыков в объеме, необходимом для выполнения своих должностных обязанностей.

### 1.4. Перечень документации

В состав документации, с которой необходимо ознакомиться администратору Системы входят:

- Функциональная спецификация системы GameRanker
- Техническое задание на разработку системы
- Настоящее руководство администратора

## 2. УСТАНОВКА СИСТЕМЫ

### 2.1. Системные требования

#### 2.1.1. Аппаратные требования

- Количество логических ядер процессора: 2
- Семейство процессоров: x86/x64
- Частота процессора: 2.0 ГГц
- Объем установленной памяти: 4 Гб
- Дисковое пространство: 10 Гб

#### 2.1.2. Программные требования

- Операционная система: Debian 11+ или аналогичная, поддерживающая Docker
- Docker: 20.10+ (open-source community edition)
- База данных: PostgreSQL 16+
- Дополнительно (опционально):
  - Prometheus для сбора метрик
  - Grafana для визуализации метрик
  - Docker Swarm или Kubernetes для оркестрации

#### 2.1.3. Язык программирования

Система разработана на GoLang 1.24+.

## 2.2. Порядок установки

1. Смонтируйте диск с дистрибутивом в папку /mnt
2. Скопируйте из дистрибутива исходники из папки /mnt в папку /opt/gameranker
3. Отредактируйте файл docker-compose.yml в соответствии с разделом 3.2 данного документа
4. Смените текущую папку на /opt/gameranker и выполните команду:  
docker compose up -d --build
5. Проверьте работоспособность системы.

## 3. НАСТРОЙКА СИСТЕМЫ

### 3.1. Общие сведения

В данном документе приводятся примеры настройки Системы с использованием среды Docker Compose. Настройка операционной системы, СУБД, а также возможная настройка использования систем оркестрации, находятся вне компетенции этого документа и не будут тут описаны.

### 3.2. Конфигурируемые параметры

**Для корректной работы API сервиса необходимо настроить следующие переменные окружения:**

- HOST — адрес и порт, который слушает сервис (по умолчанию :80)
- METRICS\_PORT — порт для health-check эндпоинта (по умолчанию :3000)
- ACCESS\_KEY — ключ доступа к приватным методам API (обязательно)
- DB\_HOST — хост базы данных (по умолчанию db)
- DB\_PORT — порт базы данных (по умолчанию 5432)
- DB\_USER — пользователь базы данных (обязательно)
- DB\_PASSWORD — пароль базы данных (обязательно)
- DB\_NAME — имя базы данных (по умолчанию portable\_db)
- DB\_MAX\_OPEN\_CONNECTIONS — максимальное количество открытых соединений (по умолчанию 30)
- DB\_MAX\_IDLE\_CONNECTIONS — максимальное количество неактивных соединений (по умолчанию 30)
- DB\_CONN\_MAX\_TIME — максимальное время жизни соединения (по умолчанию 3m)

## 4. ОПИСАНИЕ API

### 4.1. Общие сведения

Система предоставляет JSON-RPC API версии 2.0 по HTTP.

Все приватные методы доступны по адресу /

Публичный метод ping доступен по адресу /public

Доступ к приватным методам защищен с помощью ключа доступа, передаваемого в заголовке Authorization в формате Bearer token

Заголовок Content-Type должен быть application/json

## 4.2. Методы API

### 4.2.1. ping

Проверка доступности сервиса.

**Запрос:**

```
{  
  "jsonrpc": "2.0",  
  "method": "ping",  
  "id": 1  
}
```

**Ответ:**

```
{  
  "jsonrpc": "2.0",  
  "result": "pong",  
  "id": 1  
}
```

### 4.2.2. games.create

Создание новой игры в справочнике.

**Запрос:**

```
{  
  "jsonrpc": "2.0",  
  "method": "games.create",  
  "id": 2,  
  "params": {  
    "game_id": "game_001",  
    "title": "The Witcher 3",  
    "genre": "RPG"  
  }  
}
```

**Ответ:**

```
{  
  "jsonrpc": "2.0",  
  "result": "game_001",  
  "id": 2  
}
```

### 4.2.3. games.read

Получение информации об игре.

**Запрос:**

```
{
  "jsonrpc": "2.0",
  "method": "games.read",
  "id": 3,
  "params": "game_001"
}
```

**Ответ:**

```
{
  "jsonrpc": "2.0",
  "result": {
    "game_id": "game_001",
    "title": "The Witcher 3",
    "genre": "RPG",
    "created_at": "2024-01-15T10:30:00Z",
    "updated_at": "2024-01-15T10:30:00Z"
  },
  "id": 3
}
```

### 4.2.4. games.update

Обновление информации об игре.

**Запрос:**

```
{
  "jsonrpc": "2.0",
  "method": "games.update",
  "id": 4,
  "params": {
    "game_id": "game_001",
    "title": "The Witcher 3: Wild Hunt",
    "genre": "Action RPG"
  }
}
```

**Ответ:**

```
{ "jsonrpc": "2.0",
  "result": true,
  "id": 4
```

```
}
```

#### 4.2.5. games.delete

Удаление игры из справочника.

**Запрос:**

```
{
  "jsonrpc": "2.0",
  "method": "games.delete",
  "id": 5,
  "params": "game_001"
}
```

**Ответ:**

```
{
  "jsonrpc": "2.0",
  "result": true,
  "id": 5
}
```

#### 4.2.6. games.list

Получение списка всех игр.

**Запрос:**

```
{
  "jsonrpc": "2.0",
  "method": "games.list",
  "id": 6
}
```

**Ответ:**

```
{
  "jsonrpc": "2.0",
  "result": [
    {
      "game_id": "game_001",
      "title": "The Witcher 3: Wild Hunt",
      "genre": "Action RPG",
      "created_at": "2024-01-15T10:30:00Z",
      "updated_at": "2024-01-15T11:45:00Z"
    },
    {
      "game_id": "game_002",
      "title": "Cyberpunk 2077",
      "genre": "Action RPG",

```

```
    "created_at": "2024-01-15T12:00:00Z",
    "updated_at": "2024-01-15T12:00:00Z"
  }
],
  "id": 6
}
```

#### 4.2.7. events.create

Регистрация игровой сессии пользователя.

**Запрос:**

```
{
  "jsonrpc": "2.0",
  "method": "events.create",
  "id": 7,
  "params": {
    "user_id": "user_001",
    "game_id": "game_001",
    "duration": 7200,
    "ts": "2024-01-15T20:30:00Z"
  }
}
```

**Ответ:**

```
{
  "jsonrpc": "2.0",
  "result": 1,
  "id": 7
}
```

#### 4.2.8. events.list\_by\_user

Получение истории игровых сессий пользователя.

**Запрос:**

```
{
  "jsonrpc": "2.0",
  "method": "events.list_by_user",
  "id": 8,
  "params": "user_001"
}
```

**Ответ:**

```
{
  "jsonrpc": "2.0",
  "result": [
    {
      "id": 1,
      "user_id": "user_001",
      "game_id": "game_001",
      "duration": 7200,
      "ts": "2024-01-15T20:30:00Z"
    }
  ],
  "id": 8
}
```

#### 4.2.9. preferences.list\_by\_user

Получение всех предпочтений пользователя.

**Запрос:**

```
{
  "jsonrpc": "2.0",
  "method": "preferences.list_by_user",
  "id": 9,
  "params": "user_001"
}
```

**Ответ:**

```
{
  "jsonrpc": "2.0",
  "result": [
    {
      "user_id": "user_001",
```

```
    "game_id": "game_001",
    "sessions_count": 1,
    "total_duration": 7200,
    "last_session_ts": "2024-01-15T20:30:00Z",
    "preference_score": 2.22
  }
],
"id": 9
}
```

#### 4.2.10. preferences.list\_by\_user\_limited

Получение топ-N предпочтений пользователя.

**Запрос:**

```
{
  "jsonrpc": "2.0",
  "method": "preferences.list_by_user_limited",
  "id": 10,
  "params": {
    "user_id": "user_001",
    "limit": 5
  }
}
```

**Ответ:**

```
{
  "jsonrpc": "2.0",
  "result": [
    {
      "user_id": "user_001",
      "game_id": "game_001",
      "sessions_count": 1,
      "total_duration": 7200,
      "last_session_ts": "2024-01-15T20:30:00Z",
      "preference_score": 2.22
    }
  ],
  "id": 10
}
```

#### 4.2.11. preferences.user\_top\_genres

Получение топ-3 любимых жанров пользователя.

**Запрос:**

```
{
  "jsonrpc": "2.0",
  "method": "preferences.user_top_genres",
  "id": 11,
  "params": "user_001"
}
```

**Ответ:**

```
{
  "jsonrpc": "2.0",
  "result": [
    {
      "genre": "Action RPG",
      "cnt": 1
    }
  ],
  "id": 11
}
```

#### 4.2.12. preferences.recommended\_games

Получение рекомендованных игр для пользователя.

**Запрос:**

```
{
  "jsonrpc": "2.0",
  "method": "preferences.recommended_games",
  "id": 12,
  "params": {
    "user_id": "user_001",
    "genres": ["RPG", "Action"],
    "limit": 10
  }
}
```

**Ответ:**

```
{
  "jsonrpc": "2.0",
  "result": [
    {
      "game_id": "game_002",
```

```
"title": "Cyberpunk 2077",
```

```
"genre": "Action RPG"
```

```
}
```

```
,
```

```
"id": 12
```

```
}
```