

Система мониторинга терминала НРД: Метрики HealthCheck

Общее описание

Система HealthCheck в терминале НРД предназначена для мониторинга состояния всех критически важных компонентов системы и обеспечивает непрерывный контроль работоспособности инфраструктуры документооборота.

Архитектура системы мониторинга

Основные компоненты

- **Терминал НРД** - центральный узел системы мониторинга
- **Маршрутизатор (Gateway)** - промежуточный компонент для маршрутизации запросов
- **Адаптеры банков** - специализированные модули для интеграции с банками
- **База данных** - хранилище данных мониторинга и метрик

Базовые метрики терминала

1. Доступность базы данных (TerminalDatabase)

- **Описание:** Проверка доступности подключения к базе данных терминала
- **Статусы:**
 - **OK** - База данных доступна
 - **Error** - Нет доступа к базе данных
- **Критичность:** Критическая
- **Воздействие при сбое:** Полная остановка работы терминала

2. Доступность папок файлового шлюза (FGateFolders)

- **Описание:** Проверка доступности рабочих директорий файлового шлюза
- **Статусы:**
 - **OK** - Все папки доступны для чтения/записи
 - **Error** - Нет доступа к одной или нескольким папкам
- **Критичность:** Критическая
- **Воздействие при сбое:** Невозможность обработки файлов документооборота

3. Загрузка модулей Python (PythonModules)

- **Описание:** Проверка успешной загрузки всех необходимых Python-модулей
- **Статусы:**
 - **OK** - Все модули загружены успешно
 - **Error** - Один или несколько модулей не загружены
- **Критичность:** Критическая
- **Рекомендации при сбое:** Требуется перезагрузка терминала

Системные метрики

1. Использование процессора (CpuUsage)

- **Описание:** Процент загрузки процессора
- **Тип данных:** `int` (процент от 0 до 100)
- **Периодичность:** Обновляется каждый цикл мониторинга
- **Пороговые значения:** Настраиваются индивидуально

2. Использование оперативной памяти (RamUsage)

- **Описание:** Процент использования оперативной памяти
- **Тип данных:** `int` (размер в МБ)
- **Периодичность:** Обновляется каждый цикл мониторинга

3. Размер базы данных (DbSize)

- **Описание:** Размер базы данных в мегабайтах
- **Тип данных:** `int` (размер в МБ)
- **SQL запрос:**

```
SELECT SUM (db_size)
FROM (SELECT pg_database_size(pg_database.datname)/1024/1024 as db_size
FROM pg_database) as size;
```

Метрики очередей

1. Общий размер очереди (QueueSize)

- **Описание:** Общее количество сообщений в очереди обработки
- **Тип данных:** `int`

2. Очередь подписи (SignQueueSize)

- **Описание:** Количество документов в очереди на подпись
- **Тип данных:** `int`

3. Очередь отправки (SendQueueSize)

- **Описание:** Количество документов в очереди на отправку
- **Тип данных:** `int`

4. Очередь импорта (ImportQueueSize)

- **Описание:** Количество документов в очереди импорта
- **Тип данных:** `int`

Версия и совместимость

1. Версия сервиса (Version)

- **Описание:** Текущая версия терминала
- **Формат:** Семантическое версионирование (например, "24.5.1")

2. Проверка версии маршрутизатора (GatewayVersion)

- **Описание:** Проверка совместимости версии маршрутизатора с терминалом
- **Поля:**
 - **IsCompatible:** Булево значение совместимости
 - **CurrentVersion:** Текущая версия
 - **RequiredVersion:** Требуемая версия

Статусы адаптеров банков

Система поддерживает мониторинг состояния адаптеров различных банков:

Статусы адаптеров (AdapterStatus)

- **Unknown** (0) - Неизвестное состояние
- **Running** (1) - Адаптер работает нормально
- **Stopped** (2) - Адаптер остановлен
- **Error** (3) - Ошибка в работе адаптера

Поддерживаемые адаптеры

- **Райффайзенбанк API Middleware**
- **Сбербанк**
- **ПСБ API**
- И другие банковские адаптеры

Специальные метрики адаптеров

1. Статус токена (TokenStatus)

Применимо для адаптеров, использующих токенную аутентификацию:

Значения TokenStatus

- **Actual** - Токен действителен
- **Expires** - Токен скоро истекает
- **Expired** - Токен истек

Метрика HealthCheck

- **Ключ:** **TOKEN_STATUS**
- **Статус HealthCheck:**
 - **OK** - при **TokenStatus.Actual**
 - **Error** - при **TokenStatus.Expires** или **TokenStatus.Expired**

2. Специфические метрики банков

Каждый адаптер может предоставлять дополнительные метрики:

- Статистика API-запросов
- Время отклика сервисов банка
- Количество ошибок за период
- Средняя продолжительность операций

Статусы HealthCheck

Enum HealthCheckStatus (внутренние проверки)

- **OK** (0) - Система работает нормально
- **Error** (1) - Обнаружена ошибка

Enum HealthStatus (Microsoft.Extensions.Diagnostics.HealthChecks)

Представляет общий статус результата проверки здоровья системы. Значения упорядочены от наименее здорового к наиболее здоровому:

Unhealthy (0) - Неработоспособное состояние

- **Описание:** Указывает, что проверка работоспособности определила компонент как неработоспособный, или было выброшено необработанное исключение во время выполнения проверки
- **Критичность:** Критическая
- **Использование в системе:** В текущей реализации не используется напрямую, но является статусом по умолчанию для критических сбоев
- **Рекомендации:** Требуется немедленное вмешательство администратора

Degraded (1) - Деградированное состояние

- **Описание:** Указывает, что проверка работоспособности определила компонент как находящийся в деградированном состоянии
- **Критичность:** Предупреждение
- **Использование в системе:** Возвращается когда хотя бы одно из условий не выполнено:
 - Нет доступа к базе данных терминала
 - Нет доступа к папкам файлового шлюза
 - Не загружены модули Python
 - Хотя бы один адаптер не в статусе **Running**
 - Хотя бы одна метрика адаптера не в статусе **OK**
- **Воздействие:** Система частично работоспособна, но с ограничениями

Healthy (2) - Здоровое состояние

- **Описание:** Указывает, что проверка работоспособности определила компонент как здоровый
- **Критичность:** Норма
- **Использование в системе:** Возвращается только когда выполнены ВСЕ условия:

- ☒ `TerminalDatabaseAccess = true` - база данных доступна
- ☒ `FGateFoldersAccess = true` - папки файлового шлюза доступны
- ☒ `PythonModulesLoaded = true` - модули Python загружены
- ☒ Все адаптеры имеют статус `AdapterStatus.Running`
- ☒ Все метрики адаптеров имеют статус `HealthCheckStatus.OK`
- **Воздействие:** Система полностью работоспособна

Логика определения статуса

Система использует следующую логику для определения общего статуса:

```
if (healthCheck.TerminalDatabaseAccess
    && healthCheck.FGateFoldersAccess
    && healthCheck.PythonModulesLoaded
    && healthCheck.Adapters.All(_ => _.Value == AdapterStatus.Running)
    && metricsStatus is HealthCheckStatus.OK)
{
    return HealthCheckResult.Healthy(HealthCheckStatus.OK.ToString(), data:
    statuses);
}

return new HealthCheckResult(HealthStatus.Degraded, data: statuses);
```

Важно: В текущей реализации система НЕ возвращает статус `Unhealthy` - только `Healthy` или `Degraded`. Статус `Unhealthy` может использоваться внешними системами мониторинга или при критических системных сбоях.

Предупреждения системы мониторинга

MonitoringWarningContract

Структура предупреждений системы:

```
public class MonitoringWarningContract
{
    public string Warning { get; set; } // Текст предупреждения
    public string Severity { get; set; } // Уровень критичности
    public string Summary { get; set; } // Заголовок
}
```

Типы предупреждений

1. Нет доступа к базе данных терминала
2. Нет доступа к папкам файлового шлюза
3. Не загрузились модули Python, необходима перезагрузка терминала
4. Предупреждение об истечении токенов адаптеров

API endpoints

1. Получение статуса сервисов

```
GET /api/monitoring
Возвращает: HealthCheckContract
```

2. Получение текущих метрик

```
GET /api/monitoring/current
Возвращает: Dictionary<string, MonitoringInfoContract[]>
```

3. Загрузка архива мониторинга

```
GET /api/monitoring/download
Возвращает: ZIP-архив с данными мониторинга
```

4. Health Check endpoint

```
GET /health
Возвращает: JSON со статусом здоровья системы
```

Конфигурация мониторинга

Параметры конфигурации

- **MonitoringRate**: Периодичность сбора метрик (в секундах)
- **EnablePrometheusMetrics**: Включение метрик Prometheus
- **SendSystemInfoToNsd**: Отправка системной информации в НРД

Prometheus метрики

При включении поддержки Prometheus система экспортирует стандартные метрики:

- Использование CPU и RAM
- Размеры очередей
- Статусы адаптеров
- Время выполнения операций

Архивирование и экспорт данных

Структура архива

Система создает ZIP-архив со следующими файлами:

- `{Источник}_метрики_{Номер}.json` - данные мониторинга по источникам
- `Исключения_{Номер}.json` - файлы с исключениями и ошибками
- Метаданные о состоянии системы

Ограничения экспорта

- Максимальное количество записей на файл: настраивается
- Пакетная обработка для больших объемов данных
- Автоматическая очистка устаревших данных

Интеграция с внешними системами

Отправка в НРД

Система может автоматически отправлять информацию о состоянии в НРД:

- Архивы с метриками мониторинга
- Данные об исключениях
- Системная информация сервера

SignalR уведомления

Поддержка real-time уведомлений через SignalR:

- Мгновенные предупреждения о критических ошибках
- Уведомления об истечении токенов
- Статусные сообщения о работе адаптеров

Локализация

Система поддерживает локализацию сообщений:

- Все сообщения об ошибках на русском языке
- Локализованные названия источников мониторинга
- Адаптация интерфейса под региональные настройки

Примечание: Данная документация описывает метрики HealthCheck системы мониторинга терминала НРД. Для получения актуальной информации о конфигурации и специфических настройках обратитесь к документации администратора системы.