

Семинар №11

Spring Actuator. Настройка мониторинга с Prometheus и Grafana.

1. Инструментарий:

[Урок](#)

[Презентация](#)

2. Цели семинара №11:

- Основные принципы и понятия Spring Actuator.
- Как конфигурировать и активировать различные эндпоинты Spring Actuator.

По итогам семинара №11 слушатель должен **знать**:

- Какие метрики и информацию о состоянии приложения можно получить с помощью Spring Actuator.
- Основные принципы работы Prometheus как системы мониторинга и его интеграции с Spring Actuator.

По итогам семинара №11 слушатель должен **уметь**:

- Активировать и настраивать различные эндпоинты Spring Actuator в своем приложении.
 - Собирать и анализировать метрики и данные о состоянии приложения с использованием Spring Actuator.
-

3. План Содержание:

Этап урока	Тайминг, минуты	Формат
Введение, обзор темы	20	Модерирует преподаватель
Задание 1	40	Студенты выполняют, преподаватель помогает в решении проблем
Задание 2	40	Студенты выполняют, преподаватель

		помогает в решении проблем
Вопросы и обсуждение	20	Модерирует преподаватель
Длительность:	120	

4. Блок 1.

Тайминг:

Объяснение правил – 10 минут

Работа в команде – 30 минут

Задание:

Создайте микросервисную архитектуру с использованием Spring Cloud, включая сервисы "Пользователи" и "Задачи", а также API Gateway, и настройте мониторинг с помощью Spring Actuator, Prometheus и Grafana. Вам нужно выполнить следующие шаги:

1. Создание микросервисов:

a. Создайте микросервис "Пользователи", аналогичный примеру "Сервис 'Пользователи'" из описания. Этот сервис должен иметь конечную точку для регистрации пользователей.

b. Создайте микросервис "Задачи", аналогичный примеру "Сервис 'Задачи'" из описания. Этот сервис должен иметь конечную точку для получения списка задач.

c. Создайте API Gateway с использованием Spring Cloud Netflix Zuul, аналогичный примеру "API Gateway" из описания. API Gateway должен перенаправлять запросы к соответствующим микросервисам.

d. Включите Spring Actuator в каждом из микросервисов и API Gateway для сбора метрик и информации о состоянии.

2. Настройка Eureka:

a. Создайте Eureka-сервер, аналогично примеру "Eureka Server" из описания. Этот сервер должен использоваться для обнаружения и регистрации всех микросервисов.

b. Настройте микросервисы и API Gateway для регистрации в Eureka-сервере.

3. Интеграция с Prometheus:

a. Настройте каждый микросервис и API Gateway для экспорта метрик Spring Actuator в формате, понятном Prometheus.

b. Установите и настройте Prometheus для сбора метрик от ваших микросервисов.

4. Визуализация с Grafana:

a. Установите и настройте Grafana для визуализации данных мониторинга.

b. Создайте дашборды в Grafana для отображения метрик из Prometheus, связанных с работой ваших микросервисов и API Gateway.

5. Проверка и тестирование:

a. Проверьте, что микросервисы успешно регистрируются в Eureka.

b. Убедитесь, что метрики собираются и доступны в Prometheus.

c. Создайте дашборды в Grafana для визуализации метрик и проверьте их работу.

6. Документация и презентация:

a. Подготовьте документацию, описывающую вашу микросервисную архитектуру и настройку мониторинга.

b. Подготовьте презентацию, в которой объясните шаги, выполненные в рамках задания, и продемонстрируйте работу мониторинга с использованием Grafana.

Часто встречающиеся ошибки:

1. Ошибки конфигурации Eureka: убедитесь, что все микросервисы и API Gateway правильно настроены для регистрации в Eureka.
2. Ошибки в конфигурации API Gateway: проверьте, что маршрутизация запросов настроена правильно и применены необходимые фильтры.
3. Проблемы с безопасностью: обеспечьте защиту конечных точек API и правильную настройку авторизации и аутентификации.
4. Неадекватная обработка ошибок: убедитесь, что Hystrix настроен для обработки сбоев сервиса.
5. Проблемы с зависимостями: используйте совместимые версии библиотек и инструментов Spring Cloud.

5. Блок 2.

Тайминг:

Объяснение правил – 10 минут

Работа в команде – 30 минут

Задание на тему Spring Actuator. Настройка мониторинга с Prometheus и Grafana:

Задание:

Настройте мониторинг ваших микросервисов, используя Spring Actuator, Prometheus и Grafana. Ваша задача - создать конфигурацию, которая позволит собирать метрики о состоянии и производительности ваших сервисов, а также визуализировать эти метрики с помощью Grafana.

Пример решения:

1. Добавьте зависимости Spring Actuator, Prometheus и Grafana в файлы `pom.xml` вашего микросервиса.
2. Включите Spring Actuator в вашем микросервисе, добавив конфигурацию в файл `application.properties` (или `application.yml`):

```
management.endpoints.web.exposure.include=*  
management.endpoint.prometheus.enabled=true
```

3. Включите поддержку Prometheus, добавив конфигурацию в файл ``application.properties`` (или ``application.yml``):

```
spring.metrics.export.prometheus.enabled=true
```

4. Создайте файл конфигурации для Prometheus (например, ``prometheus.yml``) и определите в нем конфигурацию для сбора метрик из вашего микросервиса. Пример конфигурации может выглядеть следующим образом:

```
global:
  scrape_interval: 15s
scrape_configs:
  - job_name: 'spring-actuator'
    metrics_path: '/actuator/prometheus'
    static_configs:
      - targets: ['your-microservice-host:port']
```

5. Запустите Prometheus, используя ваш файл конфигурации.

6. Установите Grafana и добавьте источник данных Prometheus.

7. Создайте панели Grafana для визуализации метрик из вашего микросервиса.

8. Настройте алерты в Grafana, если необходимо, для мониторинга критических метрик.

Часто встречающиеся ошибки:

1. Ошибки конфигурации Prometheus: Проблемы с конфигурацией ``prometheus.yml`` могут привести к неправильному сбору метрик или их отсутствию.

2. Ошибки в настройке Spring Actuator: Неправильная настройка Spring Actuator может привести к тому, что метрики не будут доступны.

3. Проблемы с безопасностью: Необходимо обеспечить безопасность доступа к метрикам и API Actuator.

4. Ошибки в настройке Grafana: Неправильная настройка Grafana может вызвать проблемы с визуализацией данных или настройкой алертов.

Обратите внимание, что вы должны адаптировать этот общий подход к вашей конкретной архитектуре и требованиям.

6. Домашнее задание

Домашнее задание на тему Spring Actuator. Настройка мониторинга с Prometheus и Grafana:

Задание:

1. Создайте новый микросервис на базе Spring Boot и включите в него зависимости Spring Actuator, Prometheus и Grafana.

2. Настройте Spring Actuator в вашем микросервисе для сбора следующих метрик: информации о состоянии приложения (`/actuator/health``), использования памяти (`/actuator/metrics/jvm.memory.used``), количества HTTP запросов (`/actuator/metrics/http.server.requests``), и любых других метрик по вашему выбору.

3. Создайте файл конфигурации Prometheus (`prometheus.yml``), в котором определите конфигурацию для сбора метрик из вашего микросервиса. Настройте интервал сбора и выберите метрики, которые вы хотите собирать.

Пример решения:

1. Создайте новый проект Spring Boot и добавьте необходимые зависимости в файл `pom.xml``.

2. Настройте Spring Actuator, добавив конфигурацию в файл `application.properties``:

```
management.endpoints.web.exposure.include=*
management.endpoint.prometheus.enabled=true
```

3. Создайте контроллеры и сервисы для вашего микросервиса, чтобы генерировать данные для метрик. Например:

```
@RestController
public class MyController {

    private final Counter httpRequestsCounter =
Metrics.counter("http.requests.total");

    @GetMapping("/api/some-endpoint")
    public ResponseEntity<String> someEndpoint() {
        // Ваша логика
        httpRequestsCounter.increment();
        return ResponseEntity.ok("Success");
    }
}
```

4. Создайте файл конфигурации Prometheus (` prometheus.yml `) и определите в нем конфигурацию для сбора метрик:

```
global:
  scrape_interval: 15s
scrape_configs:
  - job_name: 'spring-actuator'
    metrics_path: '/actuator/prometheus'
    static_configs:
      - targets: ['your-microservice-host:port']
```

5. Запустите ваш микросервис и Prometheus.

6. Установите Grafana и добавьте источник данных Prometheus.

7. Создайте панели Grafana для визуализации метрик из вашего микросервиса.

Рекомендации для преподавателей по оценке задания:

1. Оцените правильность настройки Spring Actuator и Prometheus в микросервисе.
2. Проверьте, что метрики собираются и доступны для Prometheus.
3. Оцените наличие и правильность создания файлов конфигурации для Prometheus и Grafana.
4. Проверьте визуализацию метрик в Grafana и наличие алертов, если они были настроены.
5. Оцените общую четкость и оформление кода и конфигураций.