

Ошибки при регистрации сервисов как localhost на Eureka

Ервлева Регина Викторовна

Приамурский государственный университет имени Шолом-Алейхема

Студент

Ервлев Павел Андреевич

Приамурский государственный университет имени Шолом-Алейхема

Студент

Аннотация

В данной статье будут рассмотрены основные ошибки, возникающие при регистрации сервисов на localhost с использованием Eureka. Так же будут предложены решения исправления каждой ошибки.

Ключевые слова: Spring Cloud, Spring, Spring boot

Errors when registering services as localhost on Eureka

Eroleva Regina Viktorovna

Sholom-Aleichem Priamursky State University

student

Erolev Pavel Andreevich

Sholom-Aleichem Priamursky State University

Student

Abstract

This article will discuss the main errors that occur when registering services on localhost using Eureka. Also, solutions will be offered to fix each error.

Keywords: Spring Cloud, Spring, Spring boot

Eureka — это реестр служб, позволяющий микросервисам использовать его для регистрации. Это позволяет другим приложениям и службам искать адрес микросервиса в реестре. Преимущество этого в том, что больше не нужно полагаться на жестко запрограммированные имена хостов и порты, потому что они могут измениться.

Eureka часто используется с Feign или Ribbon, которые разрешают имя хоста и обеспечивают балансировку нагрузки на стороне клиента, если существует несколько экземпляров одних и тех же микросервисов.

В своей статье А.И. Егунова и др. рассматривают проблемы повышения эффективности образовательной деятельности вуза и использование научной интеллектуальной собственности в преподавательской деятельности. Так же предложили реализацию

информационного хранилища и поисковой системы в виде J2EE-приложения с использованием фреймворка Spring [1]. П.В. Прохоров и Н.В. Разговоров рассмотрели современные подходы и технологии в разработке серверных приложений на примере онлайн-магазина с использованием Spring [2]. В статье В.С. Жданова описала подход в реализации серверной части клиент-серверного мобильного приложения для просмотра расписания [3]. Так же А.Д. Нарижный и Н.Е. Губенко провели сравнительный анализ технологий, которые имеют схожую функциональность и которые предназначены для одних и тех же задач. Это технологии стеков Spring и JavaEE, которые предназначены для разработки Enterprise-приложений [4]. В.Л. Волушкова рассмотрела в своей работе технологии программирования на примере языка Java [5].

Одна из проблем с Eureka заключается в том, что службы иногда случайно регистрируются как localhost. Это вызывает проблемы, потому что приложения на других серверах теперь будут пытаться подключиться к микросервису на локальном хосте, а не в правильном месте.

При поиске решений в Интернете люди часто советуют использовать IP-адреса. Однако это скорее всего лишь обходной путь, а не решение проблемы.

Если посмотреть на код, то будет видно, что Spring полагается на класс, вызываемый «InetUtils» для разрешения имени хоста.

Этот класс работает следующим образом. Он получает список всех сетевых интерфейсов, после этого он будет перебирать все сетевые интерфейсы и находить первый сетевой адрес IPv4 этих интерфейсов «InetUtils.findFirstNonLoopbackAddress()». Как только он получит этот адрес, он зарегистрирует это приложение с именем хоста этого адреса «InetUtils.convertAddress()».

Прежде чем рассмотреть, почему это может привести к сбою, необходимо включить ведение журнала отладки для «InetUtils» класса (рис.1).

```
logging:  
  level:  
    org.springframework.cloud.commons.util.InetUtils: trace
```

Рисунок 1 – Включение ведения журнала

Одна из причин, по которой можно получить неправильное имя хоста, заключается в том, что он выбрал неправильный сетевой интерфейс или адрес. Чтобы узнать, какой сетевой интерфейс используется, нужно проверить журналы на наличие следующих сообщений (рис.2).

```
Testing interface: Software Loopback Interface 1
Testing interface: vmxnet3 Ethernet Adapter
Found non-loopback interface: vmxnet3 Ethernet Adapter
```

Рисунок 2 – Ошибка не правильного адреса

Был выбран адаптер «vmxnet3 Ethernet». Если это не тот сетевой интерфейс, то его можно проигнорировать, настроив «spring.cloud.inetutils.ignored-interfaces» свойство (рис.3).

```
spring:
  cloud:
    inetutils:
      ignored-interfaces:
        - vmxnet3*
```

Рисунок 3 – Игнорирование адаптера

Не следует игнорировать все сетевые адаптеры, так как это может вызвать другую ошибку.

При возникновении ошибки (An error occurred when looking up the network interfaces) «InetUtils» класс попытается получить адрес локального хоста путем вызова «InetAddress.getLocalHost()». Если это не удастся, он будет полагаться на вызываемое резервное свойство «spring.cloud.inetutils.default-hostname».

Чтобы узнать, происходит ли это, найдите либо сообщение «Невозможно получить первый адрес без обратной связи», за которым следует исключение, либо сообщение «Невозможно получить локальный хост».

Если не удастся найти сетевой интерфейс или адрес, произойдет то же самое, что и при возникновении предыдущей ошибки.

Чтобы проверить, так ли это, в журналах не должно быть сообщения, начинающегося с «Тестирование интерфейса», а также должно отсутствовать сообщение в журнале «Найден интерфейс без обратной связи».

Нужно помнить, что существуют определенные свойства, которые ограничивают допустимые сетевые интерфейсы или IP-адреса. Например, «spring.cloud.inetutils.ignored-interfaces» можно использовать для игнорирования определенных сетевых интерфейсов.

Кроме того, можно ограничить адреса, настроив предпочтительные диапазоны (рис.4).

```
spring:
  cloud:
    inetutils:
      preferred-networks:
        - 192.168
```

Рисунок 4 – Настройка диапазона

Наиболее частая проблема, которая вызывает localhost - это когда на разрешение имени хоста уходит слишком много времени. Чтобы проверить, так ли это, нужно найти в журналах следующее сообщение (рис.5).

```
Cannot determine local hostname
```

Рисунок 5 – Ошибка времени

В «InetUtils.convertAddress()», есть ограничение по времени. По умолчанию это 1 секунда. Это означает, что, если для разрешения имени хоста требуется более одной секунды, код вернется к работе localhost.

Но можно увеличить количество секунд, которое требуется до использования резервной копии (рис.6).

```
spring:
  cloud:
    inetutils:
      timeout-seconds: 10
```

Рисунок 6 – Увеличение времени

Как можно заметить, микросервисы не просто случайным образом регистрируют себя как localhost. Всегда есть ясная причина, и включение ведения журнала трассировки или отладки «InetUtils» класса поможет выяснить причину.

В данной статье были рассмотрены основные ошибки при регистрации микросервисов, а так же подобраны методы их решения.

Библиографический список

1. Егунова А.И., Аббакумов А.А., Воропаева М.А., Вечканова Ю.С. Система управления хранилищем электронных образовательных ресурсов // Образовательные технологии и общество. 2019. №3. С. 145-154.
2. Прохоров П.В., Разговоров Н.В. Современные подходы в backend разработке на примере онлайн-магазина // Прикладная математика и

- фундаментальная информатика. 2020. №2. С. 23-28.
3. Жданова В.С. Серверный модуль системы уведомления об изменениях в расписании занятий // Молодость. Интеллект. Инициатива. 2017. С. 21-22.
 4. Нарижный А.Д., Губенко Н.Е. Сравнительный анализ стеков технологий spring и javaee (jakartaee) для разработки enterprise приложений // Информатика, управляющие системы, математическое и компьютерное моделирование. 2020. №3. С. 549-462.
 5. Волушкова В.Л. Архитектурные решения java для доступа к данным // Теоретические основы программирования. Учебное пособие. 2019. 137с.