Требования к программному обеспечению

Для выполнения ПО в среде JRE, в дополнении к необходимым техническим средствам, требуется наличие:

- Операционной системы CentOS 7.2, Red Hat Enterprise Linux Server 7.2 или Ubuntu 18.04 LTS;
- Open JDK версия 11 доступная в пути среды операционной системы. Например, на операционной системе Ubuntu 18.04 LTS, установка Open JDK 11 производится посредством выполнения: sudo apt install default-jdk.

Описание поставки ПО

ПО включает следующие файлы:

- Компонент прокси http-proxy-${version}.jar предоставленный производителем ПО в виде библиотеки JAR, содержащий все необходимые библиотеки поддержки,
- [опционально] файл override.yaml с настройками ПО, которые отличаются от настроек поставляемых с компонентом прокси по умолчанию.

Инструкция по установке ПО

Для установки программного обеспечения необходимо скопировать файл ПО http-proxy-${version}.jar в необходимую директорию. В случае использования настроек, отличных от используемых по умолчанию, в требуемую директорию копируется файлом конфигурации override.yaml.

Запуск ПО

Запуск осуществляется при помощи интерфейса командной строки (или с помощью скрипта) посредством следующей команды:

java -jar http-proxy-${version}.jar --spring.config.additional-location=file:${path-to-config}/override.yaml

где,

${version} – версия библиотеки компонента прокси, в настоящее время 1.0.0

${path-to-config} – путь к файлу с настройками ПО

При успешном старте компонент прокси выведен на консоль следующее сообщение:

Started Application in X.Y seconds (JVM running for X.Y)

После этого компонент прокси готов к получению запросов на порты, открытые во время запуска. В процессе работы компонент будет выводить всю отладочную информацию на консоль.

Настройки ПО

ПО хранит все базовые настройки в файле application.yaml который находится в корневом директории библиотеки JAR компонента. Пользователи ПО могут запросить полную версию application.yaml с каталогом всех настроек и их допустимых значений у производителя ПО. Для изменения параметров ПО пользователи могут создать

дополнительный файл в формате YAML (например override.yaml) и поместить все желаемые настройки в данный файл, например:

```
anonymizer:

 routes:
  - name: 'SOAP'
    ingressPort: 2220
    resolver: 'Generic'
    services:
      - name: 'SOAP Service'
        baseUrl: 'https://some-service.com:443'
        path: '/custom'
```

Затем, пользователи могут указать на дополнительный файл настроек через опцию компонента --spring.config.additional-location как указано в разделе запуск ПО.

Для логирования ПО использует библиотеку SLF4J, и настройка уровней логирования осуществляется способом, который является стандартным для Spring Boot: https://www.baeldung.com/spring-boot-logging. Производитель предоставляет базовую конфигурацию компонента логирования по отдельному запросу.

Тестирование работоспособности ПО

Данный раздел приводит пример функционирования ПО с использованием встроенного тестового файла в формате DSL.

Для реализации тестирования ПО предоставляет следующие файлы:

- ПО (http-proxy-1.0.0.jar),
- Файл override.yaml с настройками для тестирования ПО

```
anonymizer:
 routes:
   - name: 'Test'
     ingressPort: 2220
     resolver: 'Generic'
     services:
       - name: 'Hello World'
         baseUrl: 'http://localhost:9000'
         path: '/'
```

- Файл в формате DSL (встроенный в ПО для упрощения тестирования),

```
ResponseTemplate:
 content: custom GoodBye
```

- Java класс GoodBye (встроенный в ПО), реализующий акцию для файла DSL,

```
public class GoodBye implements Action {

   GoodBye(String... args) { }

   @Override
   public Object execute(String value, Variables variables, SharedContextReference scr) {
      return value.replace("Hello", "Good Bye");
   }
}
```

- Библиотека hello-service.jar с простой службой для тестирования (https://github.com/spring-guides/gs-actuator-service).

Для выполнения примера необходимо осуществить следующие шаги:

1. Использовать операционную систему Ubuntu 18.04 LTS,
2. Убедиться, что Open JDK 11 установлен и доступен (см. раздел Установка, выполнение и доступ в ПО в среде JRE),
3. Убедиться, что порты 2220, 8080 и 9000 операционной системы не используются другими приложениями,
4. Поместить все необходимые файлы (http-proxy-1.0.0.jar, hello-service.jar, override.yaml) в папку, в которой будет производиться тестирование,
5. Открыть три терминальных окна и в каждом из них перейти в папку, в которой будет производиться тестирование.
6. Запустить тестовую службу в первом окна:

   ```
   java -jar hello-service.jar
   ```

7. Запустить ПО во втором терминальном окне:

   ```
   java -jar http-proxy-1.0.0.jar --spring.config.additional-location=file:$(pwd)/override.yaml
   ```

8. Из третьего терминального окна вызвать тестовую службу напрямую:

   ```
   curl http://localhost:9000/hello-world?name=Proxy

   {"id":2,"content":"Hello, Proxy!"}
   ```

9. В продолжение из третьего окна вызвать тестовую службу через ПО:

   ```
   curl http://localhost:2220/hello-world?name=Proxy

   {
    "id" : 1,
    "content" : "Good Bye, Proxy!"
   }
   ```

Как видно из шага №9, ПО находит стратегию по обработке запроса и ответа, вызывает её. Стратегия модифицирует ответ на основании логики, указанной в файле DSL.