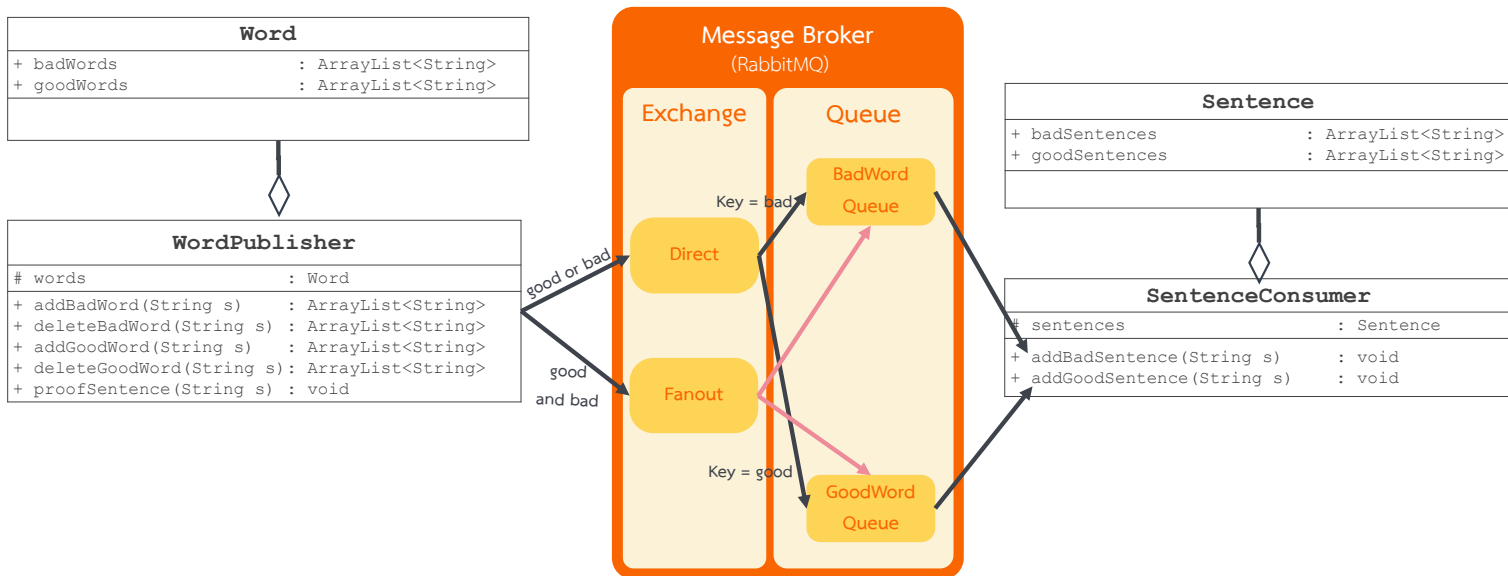


แบบฝึกปฏิบัติ ครั้งที่ 5

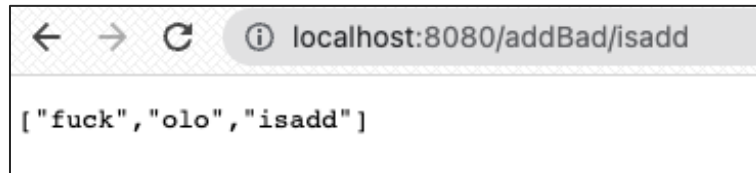
- เรื่อง** การทำงานร่วมกับ Queues และ Routing ด้วย RabbitMQ
- วัตถุประสงค์**
1. เพื่อฝึกฝนการสร้างและใช้งาน Queue และ Exchange
 2. เพื่อฝึกฝนการสื่อสารระหว่าง Service ด้วย RabbitMQ

ข้อที่ 1 ให้นักศึกษาพัฒนา Service เพื่อแยกชนิดข้อความ

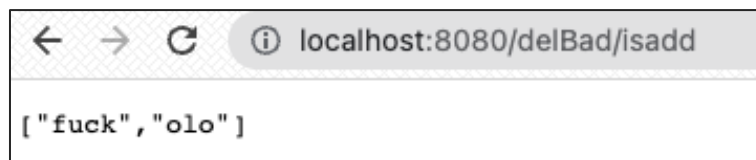


1. ให้นักศึกษาร่างคิวใน RabbitMQ จำนวน 2 คิว ได้แก่ BadWordQueue และ GoodWordQueue ดังรูป
2. ให้นักศึกษาร่าง Exchange ใน RabbitMQ จำนวน 2 อัน โดยมีทั้งชื่อและชนิดดังนี้ Direct และ Fanout ดังรูป
3. ให้นักศึกษาเชื่อมโยง Exchange เข้ากับคิว โดยมีรายละเอียดดังนี้
 - 3.1. Direct กำหนดให้ route ตัว Message ไปให้ BadWordQueue ก็ต่อเมื่อ routing key มีค่าเท่ากับ bad ขณะที่ route ไปให้ GoodWordQueue ก็ต่อเมื่อ routing key มีค่าเท่ากับ good
 - 3.2. Fanout กำหนดให้ route ตัว Message ไปให้ทั้ง BadWordQueue และ GoodWordQueue
4. ให้นักศึกษาร่างคลาส Word ที่ประกอบไปด้วย 2 แอททริบิวต์ ได้แก่ badWords และ goodWords ที่มีชนิดข้อมูลแบบ ArrayList และมีการทำ Generic เป็น String
5. ให้นักศึกษาร่างคลาส Sentence ที่ประกอบไปด้วย 2 แอททริบิวต์ ได้แก่ badSentences และ goodSentences ที่มีชนิดข้อมูลแบบ ArrayList และมีการทำ Generic เป็น String
6. ให้นักศึกษาร่างคลาส WordPublisher ที่ประกอบด้วยแอททริบิวต์ words ที่สร้างจากคลาส Word และเมธอดจำนวน 5 เมธอด ได้แก่

- 6.1. เมธอด `addBadWord(String s)` จะนำค่าพารามิเตอร์ `s` ไปเพิ่มสมาชิกใหม่ให้ `badWords` ในแอททริบิวต์ `words` นอกจากนี้ เมธอด `addBadWord(String s)` จะคืนค่าข้อมูลทั้งหมดใน `badWords` ในแอททริบิวต์ `words` ให้กับผู้เรียกใช้งาน และกำหนดให้ URL ในการเรียกใช้งานเป็น `http://localhost/addBad/{word}` ด้วยตัวดำเนินการ GET ผลลัพธ์ดังแสดงในภาพ



- 6.2. เมธอด `deleteBadWord(String s)` จะนำค่าพารามิเตอร์ `s` ไปลบสมาชิกออก `badWords` ในแอททริบิวต์ `words` นอกจากนี้ เมธอด `deleteBadWord(String s)` จะคืนค่าข้อมูลทั้งหมดใน `badWords` ในแอททริบิวต์ `words` ให้กับผู้เรียกใช้งาน และกำหนดให้ URL ในการเรียกใช้งานเป็น `http://localhost/delBad/{word}` ด้วยตัวดำเนินการ GET ผลลัพธ์ดังแสดงในภาพ



- 6.3. เมธอด `addGoodWord(String s)` จะนำค่าพารามิเตอร์ `s` ไปเพิ่มสมาชิกใหม่ให้ `goodWords` ในแอททริบิวต์ `words` นอกจากนี้ เมธอด `addGoodWord(String s)` จะคืนค่าข้อมูลทั้งหมดใน `goodWords` ในแอททริบิวต์ `words` ให้กับผู้เรียกใช้งาน และกำหนดให้ URL ในการเรียกใช้งานเป็น `http://localhost/addGood/{word}` ด้วยตัวดำเนินการ GET ผลลัพธ์ดังแสดงในภาพ



- 6.4. เมธอด `deleteGoodWord(String s)` จะนำค่าพารามิเตอร์ `s` ไปลบสมาชิกออก `goodWords` ในแอททริบิวต์ `words` นอกจากนี้ เมธอด `deleteGoodWord(String s)` จะคืนค่าข้อมูลทั้งหมดใน `goodWords` ในแอททริบิวต์ `words` ให้กับผู้เรียกใช้งาน และกำหนดให้ URL ในการเรียกใช้งานเป็น `http://localhost/delGood/{word}` ด้วยตัวดำเนินการ GET ผลลัพธ์ดังแสดงในภาพ



6.5. เมธอด `proofSentence(String s)` จะนำค่าพารามิเตอร์ `s` ที่เก็บประโยคไปเปรียบเทียบกับคำใน `goodWords` และ `badWords` ซึ่งกำหนดให้ URL ในการเรียกใช้งานเป็น `http://localhost/proof/{sentence}` ด้วยตัวดำเนินการ GET โดยที่

- ถ้าพบว่าประโยคในพารามิเตอร์ `s` มีคำที่อยู่ใน `goodWords` เท่านั้น จะส่งค่าพารามิเตอร์ `s` ไปหา `GoodWordQueue` แบบ Direct Exchange ด้วย RabbitMQ
- ถ้าพบว่าประโยคในพารามิเตอร์ `s` มีคำที่อยู่ใน `badWords` เท่านั้น จะส่งค่าพารามิเตอร์ `s` ไปหา `BadWordQueue` แบบ Direct Exchange ด้วย RabbitMQ
- ถ้าพบว่าประโยคในพารามิเตอร์ `s` มีคำอยู่ทั้งใน `badWords` และ `goodWords` จะส่งค่าพารามิเตอร์ `s` ไปหา `BadWordQueue` และ `GoodWordQueue` แบบ Fanout ด้วย RabbitMQ

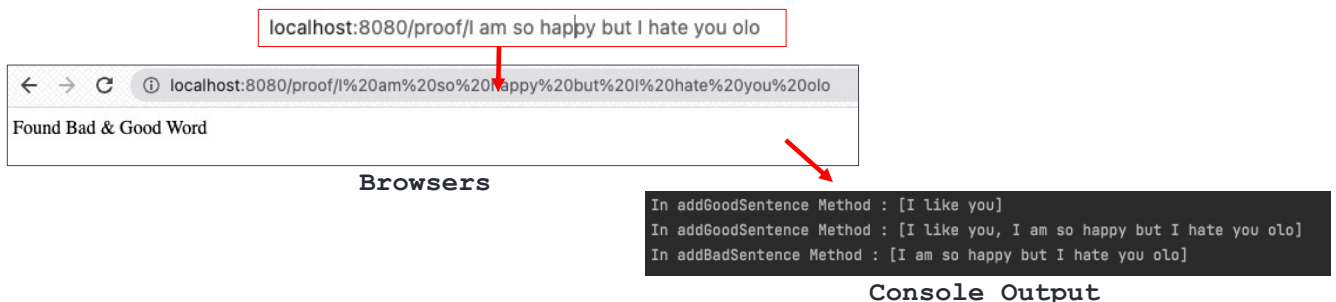
ตัวอย่าง

<https://javarevisited.blogspot.com/2016/10/how-to-check-if-string-contains-another-substring-in-java-indexof-example.html#axzz79yriS7bG>

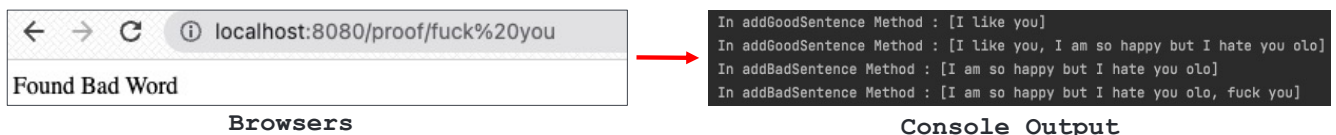
ตัวอย่างการเรียกใช้งานพร้อมแสดงผลครั้งที่ 1



ตัวอย่างการเรียกใช้งานพร้อมแสดงผลครั้งที่ 2



ตัวอย่างการเรียกใช้งานพร้อมแสดงผลครั้งที่ 3



6.6. กำหนดให้สมาชิกใน `goodWords` ในแอททริบิวต์ `words` มีค่าเริ่มต้น ได้แก่ “happy”, “enjoy” และ “life” ขณะที่ สมาชิกใน `badWords` ในแอททริบิวต์ `words` มีค่าเริ่มต้น ได้แก่ “fuck” และ “olo”

7. ให้นักศึกษาร่างคลาส SentenceConsumer ที่ประกอบด้วยแอททริบิวต์ sentences ที่สร้างจากคลาส Sentence และเมธอดจำนวน 2 เมธอด ได้แก่
 - 7.1. เมธอด addBadSentence(String s) จะนำค่าพารามิเตอร์ s ไปเพิ่มใน badSentences ในแอททริบิวต์ sentence และจะพิมพ์ทุกสมาชิกใน badSentences ผ่าน Console ซึ่งกำหนดให้ เมธอด addBadSentence(String s) ค่อยฟัง Message จาก RabbitMQ ของ BadWordQueue เท่านั้น ดังภาพ
 - 7.2. เมธอด addGoodSentence(String s) จะนำค่าพารามิเตอร์ s ไปเพิ่มใน goodSentences ในแอททริบิวต์ sentence และจะพิมพ์ทุกสมาชิกใน goodSentences ผ่าน Console ซึ่งกำหนดให้ เมธอด addGoodSentence (String s) ค่อยฟัง Message จาก RabbitMQ ของ GoodWordQueue เท่านั้น ดังภาพ

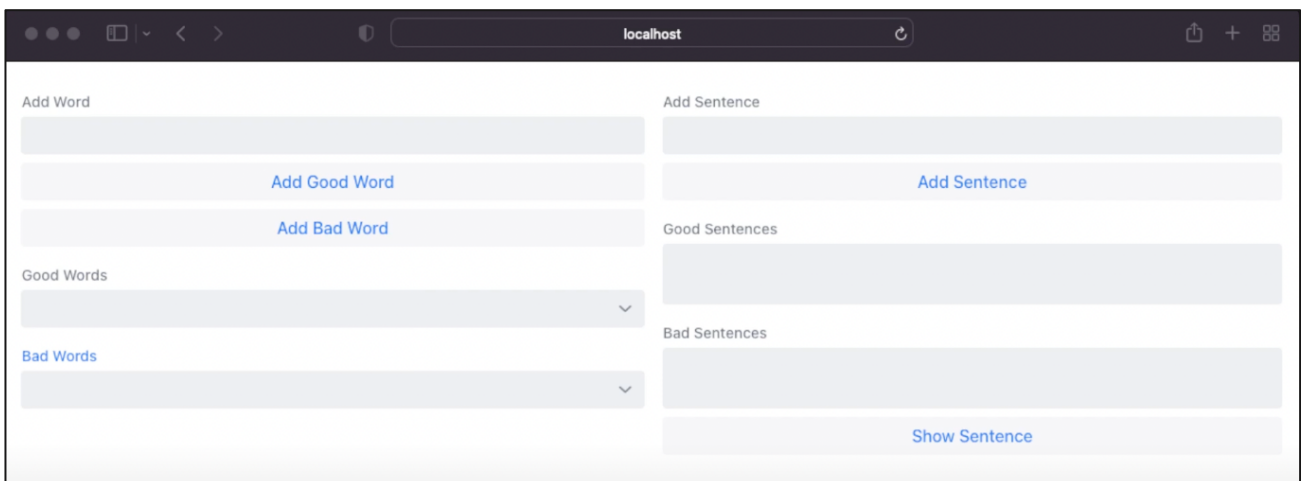
ข้อที่ 2 ให้นักศึกษาพัฒนาโปรแกรมแยกชนิดข้อความ

วิดีโอตัวอย่างการทำงานของโปรแกรม

<https://youtu.be/HvV2oyvwtEw>

รายละเอียดในการพัฒนา

1. ให้นักศึกษาคัดลอกไฟล์ Sentence, SentenceConsumer, Word, และ WordPublisher จากข้อ 1 มาใช้งาน
2. ให้นักศึกษาแก้ไขไฟล์ Sentence ดังนี้
 - 2.1. กำหนดให้ implements Serializable ให้กับคลาส Sentence
3. ให้นักศึกษาแก้ไขไฟล์ SentenceConsumer ดังนี้
 - 3.1. ให้สร้างเมธอดชื่อ getSencens() ที่ไม่มีการรับค่าและให้คืนค่ามาเป็นแอททริบิว Sentence ของคลาส SentenceConsumer โดยกำหนดให้เมธอดดังกล่าวต้องเชื่อมต่อกับคิวชื่อ GetQueue ใน RabbitMQ ที่มีการสื่อสารแบบ Direct Exchange
4. ให้นักศึกษาแก้ไขไฟล์ WordPublisher ดังนี้
 - 4.1. ให้นักศึกษาสร้าง Service ชื่อ getSentence ที่ไม่มีการรับค่า จากนั้น ให้คืนค่าผลลัพธ์เป็น Sentence ที่ได้รับจากเมธอด **getSencens() ของคลาส SentenceConsumer** ด้วย RabbitMQ โดยกำหนดรูปแบบการสื่อสารแบบ GET และมี URL ในการเรียกใช้เป็น `http://localhost:8080/getSentence`
5. ให้นักศึกษาสร้างส่วนติดต่อผู้ใช้งาน โดยกำหนดชื่อคลาสเป็น MyView2
 - 5.1. ให้นักศึกษากำหนด URL ในการเรียกใช้เป็น `http://localhost:8080/index2`
 - 5.2. ให้นักศึกษาสร้างส่วนติดต่อผู้ใช้งานดังภาพด้วย Vaadin



5.3. ให้นักศึกษาเขียนโปรแกรมเพื่อจัดการเหตุการณ์เมื่อผู้ใช้กดปุ่มต่อไปนี้

- 5.3.1. เมื่อผู้ใช้งานกดปุ่ม Add Good Word โปรแกรมจะเรียกใช้เมธอด `addGoodWord()` ของคลาส `WordPublisher` ผ่าน `WebClient` แบบ POST พร้อมกับส่งค่าในช่อง Add Word ไป จากนั้น จะนำผลลัพธ์ที่ Service ตอบกลับมาแสดงผลใน Good Words
- 5.3.2. เมื่อผู้ใช้งานกดปุ่ม Add Bad Word โปรแกรมจะเรียกใช้เมธอด `addBadWord()` ของคลาส `WordPublisher` ผ่าน `WebClient` แบบ POST พร้อมกับส่งค่าในช่อง Add Word ไป จากนั้น จะนำผลลัพธ์ที่ Service ตอบกลับมาแสดงผลใน Bad Words
- 5.3.3. เมื่อผู้ใช้งานกดปุ่ม Add Sentence โปรแกรมจะเรียกใช้เมธอด `proofSentence()` ของคลาส `WordPublisher` ผ่าน `WebClient` แบบ POST พร้อมกับส่งค่าในช่อง Add Sentence ไป จากนั้น จะนำผลลัพธ์ที่ Service ตอบกลับมาแสดงผลในรูปแบบ Notification
- 5.3.4. เมื่อผู้ใช้งานกดปุ่ม Show Sentence โปรแกรมจะเรียกใช้เมธอด `getSentence()` ของคลาส `WordPublisher` ผ่าน `WebClient` แบบ GET จากนั้น จะนำผลลัพธ์ที่ Service ตอบกลับมาแสดงผลในช่อง Good Sentences และ Bad Sentenc ดังแสดงในวิดีโอตัวอย่าง