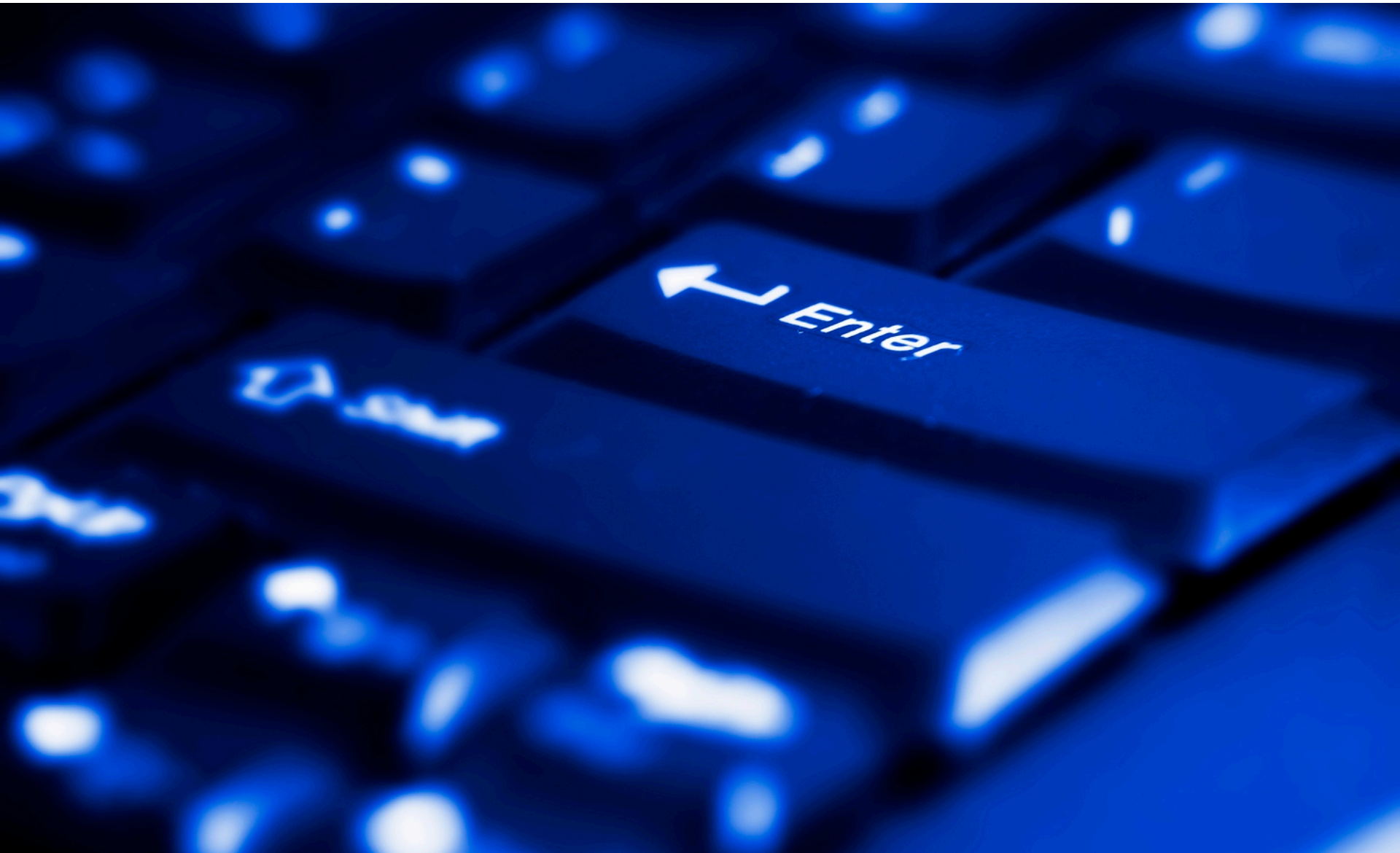




Messaging EasyNetQ and RabbitMQ

Modern Application Architectures



RabbitMQ



**spring
boot**



Install RabbitMQ

```
<dependencies>
  <dependency>
    <groupId>org.springframework.boot</groupId>
    <artifactId>spring-boot-starter-amqp</artifactId>
  </dependency>
  <dependency>
    <groupId>org.springframework.boot</groupId>
    <artifactId>spring-boot-starter-web</artifactId>
  </dependency>
</dependencies>
```

@SpringBootApplication

```
public class RabbitPublisherApplication {
```

@Bean

```
public RabbitTemplate rabbitTemplate(ConnectionFactory factory) {
    RabbitTemplate rabbitTemplate = new RabbitTemplate(factory);
    rabbitTemplate.setMessageConverter(new Jackson2JsonMessageConverter());
    return rabbitTemplate;
}
```

```
class Greeting {
    public String message;
    public int number;
```

@RequestMapping("/")

```
public String sendMessage() {
    this.template.convertAndSend( routingKey: "spring-boot",
    new Greeting( message: "hi through rabbit", number: 1));
    return "hello";
}
```



```
@EnableRabbit
@SpringBootApplication
public class RabbitSampleApplication {

    final static String queueName = "spring-boot";

    @Bean
    Queue queue() {
        return new Queue(queueName, durable: false);
    }

    @Bean
    TopicExchange exchange() {
        return new TopicExchange(name: "spring-boot-exchange");
    }

    @Bean
    Binding binding(Queue queue, TopicExchange exchange) {
        return BindingBuilder.bind(queue).to(exchange).with(queueName);
    }

    @Bean
    public SimpleRabbitListenerContainerFactory jsaFactory(
        ConnectionFactory connectionFactory,
        SimpleRabbitListenerContainerFactoryConfigurer configurer) {
        SimpleRabbitListenerContainerFactory factory = new SimpleRabbitListenerContainerFactory(
            configurer.configure(factory, connectionFactory);
        factory.setMessageConverter(new Jackson2JsonMessageConverter());
        return factory;
    }
}
```

```
@Component
class Receiver {

    @RabbitListener(queues = "spring-boot", containerFactory = "jsaFactory")
    public void receiveMessage(Greeting greet) {
        System.out.println("Received <" + greet.message + "-" + greet.number + ">");
    }

}

class Greeting {
    public String message;
    public int number;
}
```