



<http://www.lycato.com>



Spring Boot

Building Useful Services

Nguyễn Nghiệm





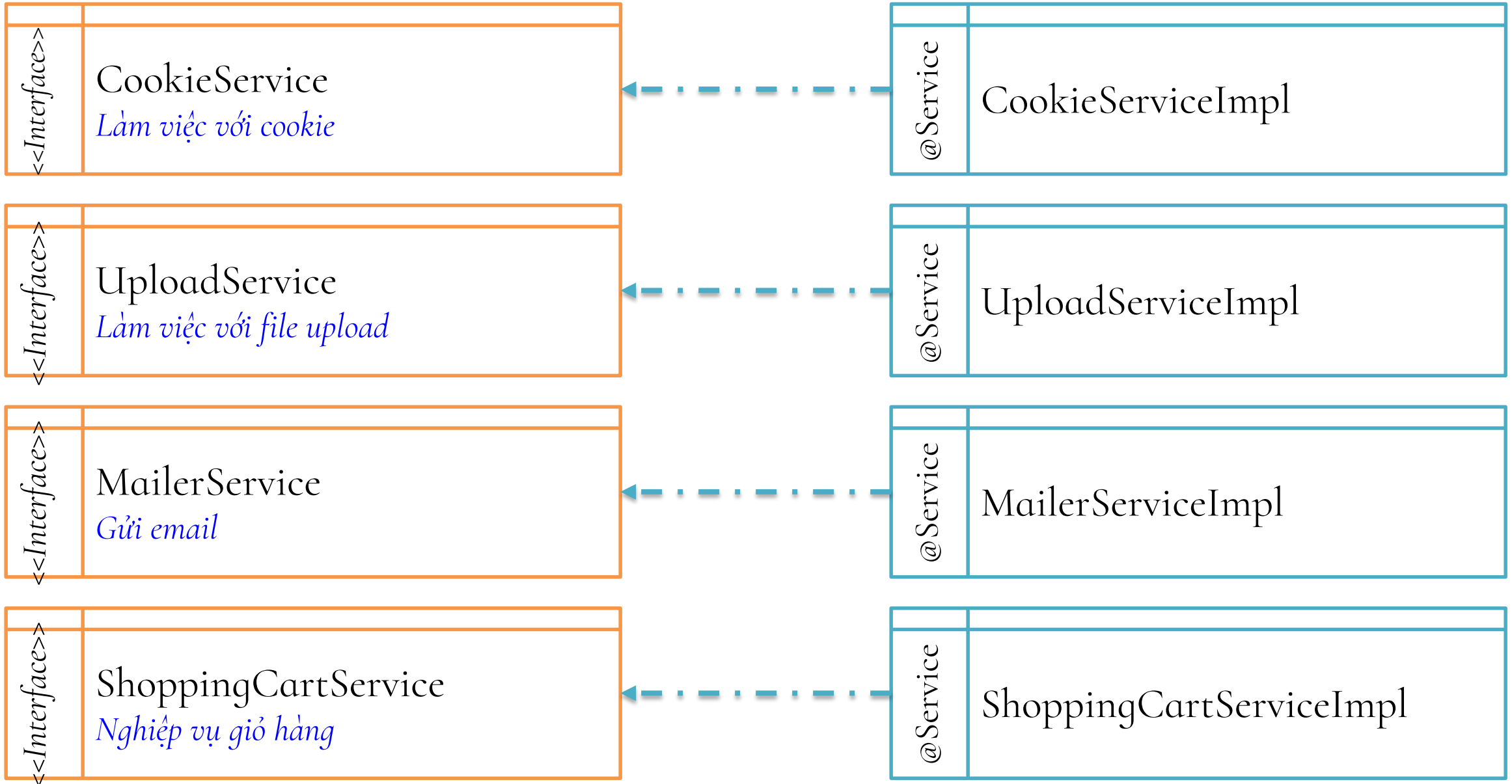
AGENDA

- ☐ COOKIESERVICE
- ☐ SESSIONSERVICE
- ☐ UPLOADSERVICE
- ☐ MAILSERVICE
- ☐ SHOPPINGCARTSERVICE





USEFUL BEANS



```
public interface CookieService {  
    Cookie get(String name);  
    String getValue(String name);  
    Cookie create(String name, String value, int hours);  
    void delete(String name);  
}
```

```
@Service  
public class CookieServiceImpl  
    implements CookieService{  
    ...?  
}
```



```
public interface MailerService {  
    void send(Mail mail);  
}
```

```
@Service  
public class MailerServiceImpl  
    implements MailerService {  
    ...?  
}
```



```
public interface UploadService {  
    File save(MultipartFile file, String folder);  
    byte[] read(String filename, String folder);  
    void delete(String filename, String folder);  
}
```

```
@Service  
public class UploadServiceImpl  
    implements UploadService{  
    ...?  
}
```



```
public interface ShoppingCartService {  
    CartItem add(CartItem item);  
    void remove(Integer id);  
    void update(Integer id, int qty);  
    void clear();  
    Collection<CartItem> getItems();  
    int getCount();  
    double getAmount();  
}
```

```
@SessionScope  
@Service  
public class ShoppingCartServiceImpl  
    implements ShoppingCartService {  
    ...  
}
```

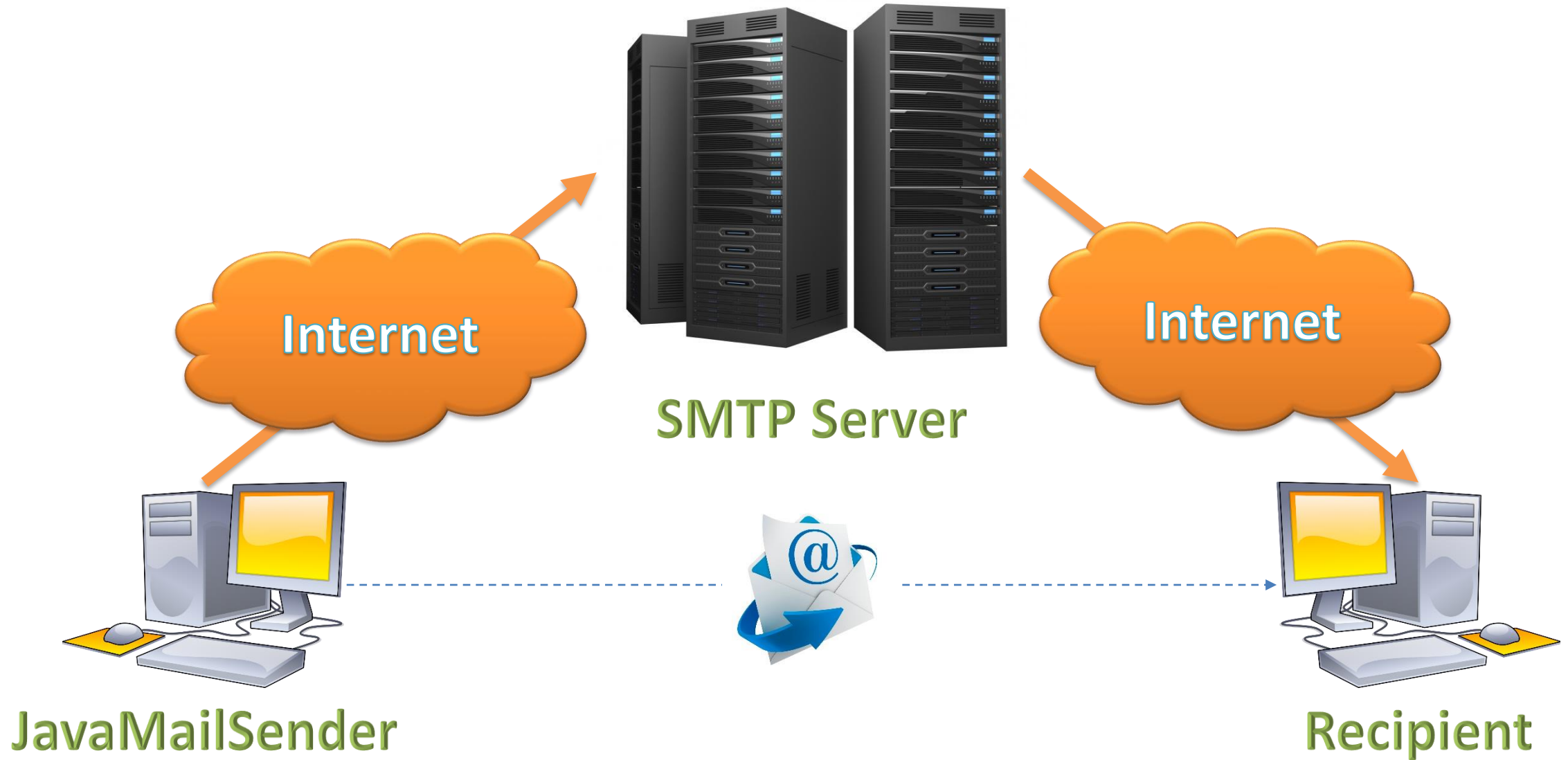
```
@Data  
public class CartItem{  
    Integer id;  
    String name;  
    double price;  
    int quantity;  
    double discount;  
    public double getAmount(){  
        return quantity*price*(1-discount);  
    }  
}
```



JavaMailSender API



DEPLOYMENT DIAGRAM





☐ Smtip Server

- ❖ Host: smtp.gmail.com:587
- ❖ Gmail Account (less secure apps)
- ❖ applications.properties

☐ Java Mail API (pom.xml)

- ❖ JavaMailSender
- ❖ MimeMessage
- ❖ MimeMessageHelper

☐ Building MailService

- ❖ MailService
- ❖ MailServiceImpl



JAVAMAILSENDER CONFIGURATION

```
<dependency>  
  <groupId>org.springframework.boot</groupId>  
  <artifactId>spring-boot-starter-mail</artifactId>  
</dependency>
```

```
spring.mail.host=smtp.gmail.com  
spring.mail.port=587  
spring.mail.username=username  
spring.mail.password=password  
spring.mail.properties.mail.smtp.auth=true  
spring.mail.properties.mail.smtp.starttls.enable=true
```

*Tài khoản gmail cần phải được
kích hoạt mới được phép sử dụng
gửi email qua ứng dụng*

<https://myaccount.google.com/lesssecureapps>



ACTIVATING GMAIL ACCOUNT

The screenshot shows a web browser window with the address bar containing `https://myaccount.google.com/lesssecureapps`. The page title is "Less secure apps". The main content area explains that some apps use less secure sign-in technology, which makes the account more vulnerable. It offers the option to turn off access for these apps or turn on access if the user wants to use them despite the risks. A link "Learn more" is provided. At the bottom, there is a toggle switch for "Allow less secure apps: ON". The toggle switch is currently turned on, and it is circled in red. The footer contains links for "Google", "Terms & Privacy", and "Help".

Less secure apps

<https://myaccount.google.com/lesssecureapps>

Google

Less secure apps

Some apps and devices use less secure sign-in technology, which makes your account more vulnerable. You can **turn off** access for these apps, which we recommend, or **turn on** access if you want to use them despite the risks. [Learn more](#)

Allow less secure apps: ON

Google Terms & Privacy Help



SENDING EMAIL

```
@Service
public class MailerServiceImpl implements MailerService {
    @Autowired
    JavaMailSender sender;
    public void send(Mail mail) {
        MimeMessage msg = sender.createMimeMessage();
        MimeMessageHelper helper = new MimeMessageHelper(msg, true, "utf-8");
        helper.setFrom(mail.getFrom()); // người gửi
        helper.setTo(mail.getTo()); // người nhận
        helper.setCc(mail.getCc()); // những người đồng nhận
        helper.setSubject(mail.getSubject()); // tiêu đề email
        helper.setText(mail.getText(), true); // nội dung email
        sender.send(msg);
    }
}
```

```
@Data
public class Mail{
    String from, to, cc,
    bcc, subject, text, files;
}
```



- ❑ `setFrom(String from);`
 - ❖ Người gửi. Ví dụ: "user@gmail.com" hoặc "Full Name <user@gmail.com>"
- ❑ `setReplyTo(String email);`
 - ❖ Người nhận phản hồi. Ví dụ: tương tự from
- ❑ `setTo(String to);`
 - ❖ Người nhận: "user@gmail.com"
- ❑ `setCc(String cc);`
 - ❖ Những người đồng nhận công khai: "user1@gmail.com, user2@gmail.com,..."
- ❑ `setBcc(String bcc);`
 - ❖ Những người đồng nhận bí mật: Ví dụ: tương tự CC
- ❑ `setSubject(String subject);`
 - ❖ Tiêu đề email. Ví dụ: "Your order purchased"
- ❑ `setText(String text, boolean html);`
 - ❖ Nội dung email: Ví dụ: "This is you order"
- ❑ `addAttachment(String filename, File file);`
 - ❖ File đính kèm: Ví dụ: "user.png", new File("c:/photos/user.png")



SUMMARY

- ✓ COOKIESERVICE
- ✓ SESSIONSERVICE
- ✓ UPLOADSERVICE
- ✓ MAILERSERVICE
- ✓ SHOPPINGCARTSERVICE





Thank
you