

# SOLID Documentation

Devapriya L

January 2024

## 1 Single Responsibility Principle

Interface:

```
1 package com.ilp.interfaces;
2
3 public interface NotificationService {
4     void sendNotification(String message);
5 }
```

Different services :

```
1 package com.ilp.service;
2
3 import com.ilp.interfaces.NotificationService;
4
5 public class EmailNotificationService implements
6     NotificationService{
7
8     @Override
9     public void sendNotification(String message) {
10         // TODO Auto-generated method stub
11         System.out.println("Email Notification: " +
12             message);
13     }
14 }
```

```

1 package com.ilp.service;
2
3 import com.ilp.interfaces.NotificationService;
4
5 public class SMSNotificationService implements
    NotificationService {
6
7     @Override
8     public void sendNotification(String message) {
9         // TODO Auto-generated method stub
10        System.out.println("SMS Notification: " +
11            message);
12    }
13 }

```

```

1 package com.ilp.service;
2
3 import com.ilp.interfaces.NotificationService;
4
5 public class WhatsAppNotificationService implements
    NotificationService{
6
7     @Override
8     public void sendNotification(String message) {
9         // TODO Auto-generated method stub
10
11    }
12
13 }

```

## 2 Open Close Principle

```
1 package com.ilp.services;
2
3 import com.ilp.interfaces.SendNotification;
4
5 public class NotificationServices {
6     public static SendNotification
7         createNotification(String type) {
8         if ("SMS".equals(type)) {
9             return new SMSNotification();
10        } else if ("Email".equals(type)) {
11            return new EmailNotification();
12        } else {
13            throw new
14                IllegalArgumentException("Invalid_
15                    notification_type");
16        }
17    }
18 }
```

```
1 package com.ilp.utility;
2
3 import com.ilp.interfaces.SendNotification;
4 //import com.ilp.services.EmailNotification;
5 import com.ilp.services.NotificationServices;
6 //import com.ilp.services.SMSNotification;
7
8 public class NotificationSystem {
9
10    public static void main(String[] args) {
11        // TODO Auto-generated method stub
12
13        SendNotification smsNotification =
14            NotificationServices.createNotification("SMS");
15        smsNotification.sendNotifications("Your_
16            booking_is_successful.");
17
18        SendNotification emailNotification =
19            NotificationServices.createNotification("Email");
20        emailNotification.sendNotifications("Your_
21            booking_is_successful.");
22    }
23 }
```

### 3 Liskov Substitution Principle

Base class:

```
1 package com.ilp.services;
2
3 import com.ilp.interfaces.SendNotification;
4
5 public class BaseNotification implements
6     SendNotification{
7
8     @Override
9     public void sendNotifications(String message) {
10         // TODO Auto-generated method stub
11         System.out.println("BaseNotification: " +
12             message);
13     }
14 }
```

Child classes:

```
1 package com.ilp.services;
2
3 public class EmailNotification extends
4     BaseNotification{
5
6     @Override
7     public void sendNotifications(String message) {
8         // TODO Auto-generated method stub
9         System.out.println("EmailNotification: " +
10             message);
11     }
12 }
```

```
1 package com.ilp.services;
2
3 public class SMSNotification extends BaseNotification{
4
5     @Override
6     public void sendNotifications(String message) {
7         System.out.println("SMSNotification: " + message);
8     }
9
10 }
```

## 4 Interface Segregation Principle

Interfaces:

```
1 package com.ilp.interfaces;
2
3 public interface EmailNotification {
4     void sendEmail(String message);
5 }
```

```
1 package com.ilp.interfaces;
2
3 public interface PushNotification {
4     void sendPushNotification(String message);
5 }
```

Classes:

```
1 package com.ilp.services;
2
3 import com.ilp.interfaces.EmailNotification;
4
5 public class EmailNotificationService implements
6     EmailNotification{
7
8     @Override
9     public void sendEmail(String message) {
10         // TODO Auto-generated method stub
11         System.out.println("Email_Notification:" +
12             message);
13     }
14 }
```

```
1 package com.ilp.services;
2
3 import com.ilp.interfaces.PushNotification;
4
5 public class PushNotificationService implements
6     PushNotification{
7
8     @Override
9     public void sendPushNotification(String message) {
10         // TODO Auto-generated method stub
11         System.out.println("Push_Notification:" + message);
12     }
13 }
```

## 5 Dependency Inversion Principle

```
1      package com.ilp.utility;
2
3      import com.ilp.interfaces.SendNotification;
4      import com.ilp.services.EmailNotification;
5      import com.ilp.services.SMSNotification;
6
7      public class NotificationSystem {
8
9          private final SendNotification sendNotification;
10
11         public NotificationSystem(SendNotification
12             sendNotification) {
13             this.sendNotification = sendNotification;
14         }
15
16         public void sendNotifications(String message) {
17             sendNotification.sendNotifications(message);
18         }
19
20         public static void main(String[] args) {
21             SendNotification smsNotification = new
22                 SMSNotification();
23             SendNotification emailNotification = new
24                 EmailNotification();
25
26             NotificationSystem smsSystem = new
27                 NotificationSystem(smsNotification);
28             smsSystem.sendNotifications("Your booking is
29                 successful.");
30
31             NotificationSystem emailSystem = new
32                 NotificationSystem(emailNotification);
33             emailSystem.sendNotifications("Your booking
34                 is successful.");
35         }
36     }
```