CSCI 5308 Assignment-2

Manan Amin (B00897712)
mn959427@dal.ca
CSID:mamin

S.O.L.I.D. Principles of Object-Oriented Programming

Project Overview:

The project includes an example service that has covered all the SOLID principles. Service fetch the latest image from Reddit and send it to Endpoint. It is simple REST APIs that expose endpoints for different platform integration. For example, the same service can work with web apps, slack, etc. For simplicity, I've avoided third-party lib/API usage. So Project has all static responses. WebEndpoint(/Reddit) is working as described. or Discord, I need to add DiscordAPI code to send messages in Discord; instead, I've only returned the URL link and message type.

Testing:
The project has a running pipeline, and artifacts get uploaded to GitLab. All good code(java classes) has been covered in Unit testing. Generally, We write unit tests for critical interfaces. Thus, I've not written that.

Repository: https://git.cs.dal.ca/courses/2021-fall/csci-5308/assignments/manan_a

REST Endponits:

/bad/reddit
/bad/discord

/good/reddit
/good/discord

Single Responsibility Principle:

SRP states that every class should only have one reason to change.

Bad Code:
In the lousy code example, RedditController class has Controller and service code, which violates this rule.

Good Code:
In Good Code Example, I've created a new file named RedditService to separate the service logic from Controller code.

Dependency Inversion Principle:

DIP states that a high-level module should not depend upon a low level, and abstractions should not rely upon details, but details should depend upon abstractions.

Spring boot makes this very simple as the All Controller class has a service class injected into it using spring beans which removed the direct dependency of the course.

Bad Code:

In the wrong Code, RedditController does not have a separate service file, and also DiscordController directly loads DiscordResponseService Class.

Good Code:
In good code, the Direct dependency of the service class has been removed. Instead, it uses RedditService Interface, which does not have Implementation details.

ISP: Interface Segregation Principle

The user does not need to implement unnecessary methods to use the interface. In other words, There should be multiple client-specific interfaces instead of one thick interface.

Code:

DiscordService in a good package uses two separate interfaces instead of one. 1 RedditService 2 DiscordResponseService.

OCP: Open-Closed Principle

OCP states that the class should open for extension but closed for any change.

Code:
After Applying Refactoring, Now code is more flexible. One can new type of integration like Slack with Reddit with interfering existing code. Also, In DiscordResponse, I've implemented a generic Abstract class. So now, Anyone can add any New type of Response by extending it.

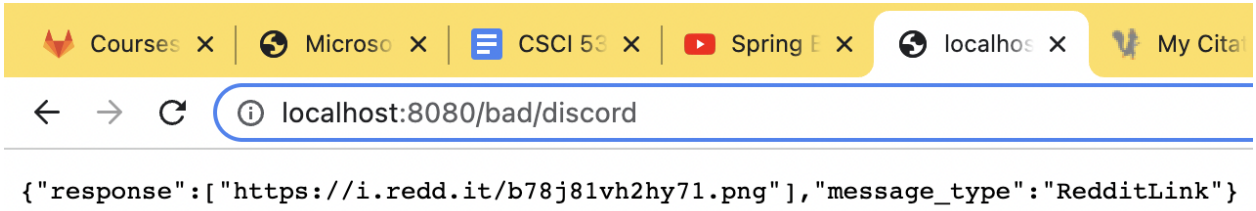
LSP: Liskov Substitution Principle

LSP states that Objects in Code should be replaceable by their subtype without altering the correctness.

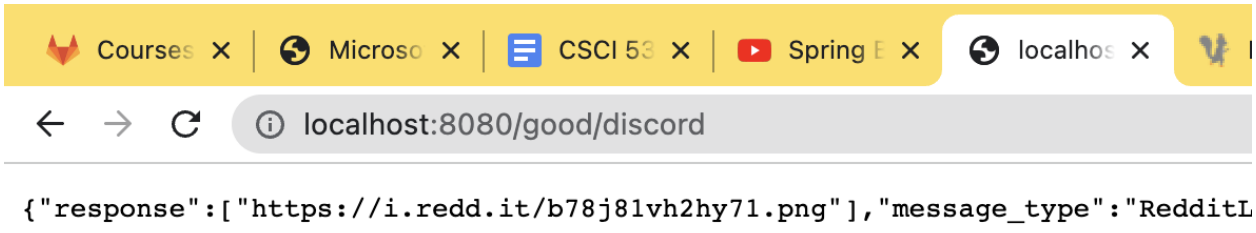
Bad:
In DiscordResponse have two sub-classes in which two methods are implemented for each. But DiscordErrorResponse sub-class only return null for getResponse(), which violets LSP. because the subclass doesn't need that method, but it must implement it because of the interface defined in the parent.

Good:
In the Good package, DiscordResponse class only have message type as attributes. So, DiscordErrorResponse does not need to implement it.

Screenshots:
Bad Endpoint:



Good Endpoint:



CI-CD pipeline results:
https://git.cs.dal.ca/courses/2021-fall/csci-5308/assignments/manan_a/-/pipelines/103752

References

E. Paraschiv, “Jackson annotation examples,” Baeldung, 30-Jan-2021. [Online]. Available: https://www.baeldung.com/jackson-annotations. [Accessed: 09-Nov-2021].

“Files · master · gitlab-examples / spring-gitlab-CF-deploy-demo,” GitLab. [Online]. Available: https://gitlab.com/gitlab-examples/spring-gitlab-cf-deploy-demo/-/tree/master. [Accessed: 09-Nov-2021].

M. Silverman, “Solid principles in action: From slack to Twilio,” Twilio Blog, 15-Aug-2018. [Online]. Available: https://www.twilio.com/blog/2017/11/solid-principles-slack-twilio.html?fbclid=IwAR0-zKfxQcvG5creD7SbVDX3vFYip9eiY0f2MFV-PWVOJ1Yk1_-fiJGJm6E. [Accessed: 08-Nov-2021].

S. Millington, “A solid guide to solid principles,” Baeldung, 18-May-2021. [Online]. Available: https://www.baeldung.com/solid-principles. [Accessed: 08-Nov-2021].

“Testing the web layer,” Spring. [Online]. Available: https://spring.io/guides/gs/testing-web/. [Accessed: 09-Nov-2021].