포팅 매뉴얼

1. Gitlab 소스 클론 이후 빌드 및 배포

사용 기술

```
1. JVM
  - Zulu 8
2. IDE
  - Intellij IDEA 2022.1 (Ultimate Edition)
  - VS Code
```

배포 과정

방화벽 설정

```
sudo ufw allow 22
sudo ufw allow 80
sudo ufw allow 443
sudo ufw allow 3000
sudo ufw allow 3001
sudo ufw allow 8080
sudo ufw allow 8081
```

Docker 설치

```
sudo apt-get update
sudo apt-get install \
    ca-certificates \
    curl \
    gnupg \
    lsb-release
sudo mkdir -p /etc/apt/keyrings
curl -fsSL https://download.docker.com/linux/ubuntu/gpg | sudo gpg --dearmor -o /etc/a
pt/keyrings/docker.gpg
echo \
  "deb [arch=$(dpkg --print-architecture) signed-by=/etc/apt/keyrings/docker.gpg] http
s://download.docker.com/linux/ubuntu \
  $(lsb_release -cs) stable" | sudo tee /etc/apt/sources.list.d/docker.list > /dev/nul
l
sudo apt-get update
sudo apt-get install docker-ce docker-ce-cli containerd.io docker-compose-plugin
```

Jenkins 설치

docker-compose.yml

```
version: '3'
services:
        jenkins:
                image: jenkins/jenkins:lts
                container_name: jenkins
                volumes:
                        - /var/run/docker.sock:/var/run/docker.sock
                        - /home/ubuntu/jenkins_compose/jenkins_home:/var/jenkins_home
                ports:
                        - "9090:8080"
                privileged: true
                user: root
```

Jenkins 내부에 Docker 설치

```
apt-get update
apt-get install \
    ca-certificates \
    curl \
    gnupg \
    lsb-release
mkdir -p /etc/apt/keyrings
curl -fsSL https://download.docker.com/linux/debian/gpg | gpg --dearmor -o /etc/apt/ke
yrings/docker.gpg
echo \
  "deb [arch=$(dpkg --print-architecture) signed-by=/etc/apt/keyrings/docker.gpg] http
s://download.docker.com/linux/debian \
  $(lsb_release -cs) stable" | tee /etc/apt/sources.list.d/docker.list > /dev/null
apt-get update
apt-get install docker-ce docker-ce-cli containerd.io docker-compose-plugin
```

Jenkins Send build artifacts over SSH Exec command

```
if (sudo docker ps | grep "front"); then sudo docker stop front; sudo docker rm front;
FRONT_IMAGE_ID=$(sudo docker image list | grep "errorshift-front" | awk '{print $3}');
sudo docker image rm $FRONT_IMAGE_ID; fi
if (sudo docker ps | grep "nginx"); then sudo docker stop nginx; sudo docker rm nginx;
NGINX_IMAGE_ID=$(sudo docker image list | grep "nginx" | awk '${print $3}'); sudo dock
er image rm $NGINX_IMAGE_ID; fi
if (sudo docker ps | grep "file_server"); then sudo docker stop file_server; sudo dock
er rm file_server; FILE_SERVER_IMAGE_ID=$(sudo docker image list | grep "file_server"
 | awk '{print $3}'); sudo docker image rm $FILE_SERVER_IMAGE_ID; fi

cd /home/ubuntu/jenkins_compose/jenkins_home/workspace/errorshift/
sudo docker compose up -d
```

Docker image

docker-compose.yml

```
version: "3.7"
services:
  nginx:
    image: nginx:1.23.2-alpine
    container_name: nginx
    ports:
      - "80:80"
      - "443:443"
    volumes:
      - /etc/letsencrypt:/etc/letsencrypt
      - /home/ubuntu/jenkins_compose/jenkins_home/workspace/errorshift/config/nginx.co
nf:/etc/nginx/nginx.conf
    depends_on:
      - front
      - file_server
  front:
    build: /home/ubuntu/jenkins_compose/jenkins_home/workspace/errorshift/frontend
    container_name: front
    expose:
      - "3000"
    environment:
      - TZ=Asia/Seoul
  file_server:
    build: /home/ubuntu/jenkins_compose/jenkins_home/workspace/errorshift/fileServer
    container_name: file_server
    expose:
      - "3001"
    environment:
      - TZ=Asia/Seoul
```

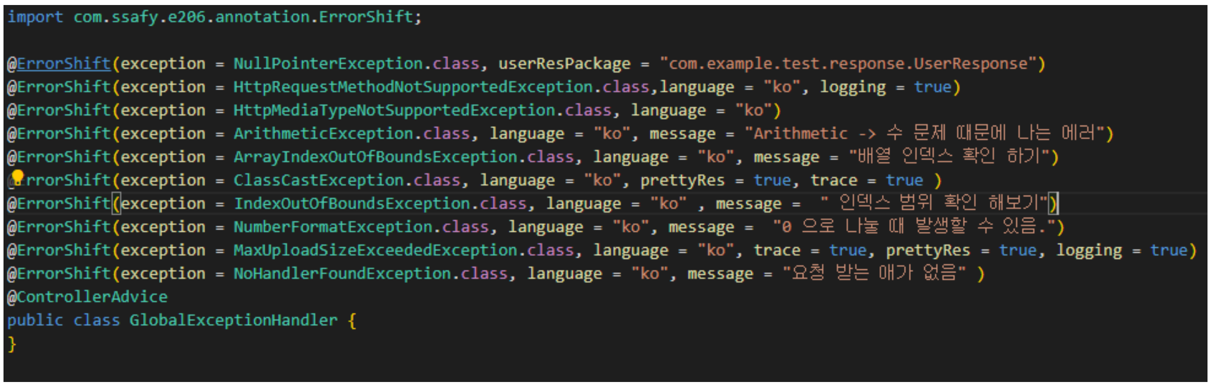
nginx.conf

```
user  nginx;
worker_processes  auto;
error_log  /var/log/nginx/error.log warn;
pid        /var/run/nginx.pid;
events {
    worker_connections  1024;
}
http {
    include       /etc/nginx/mime.types;
    default_type  application/octet-stream;

    upstream front {
        server front:3000;
    }
    upstream file_server {
        server file_server:3001;
    }
```

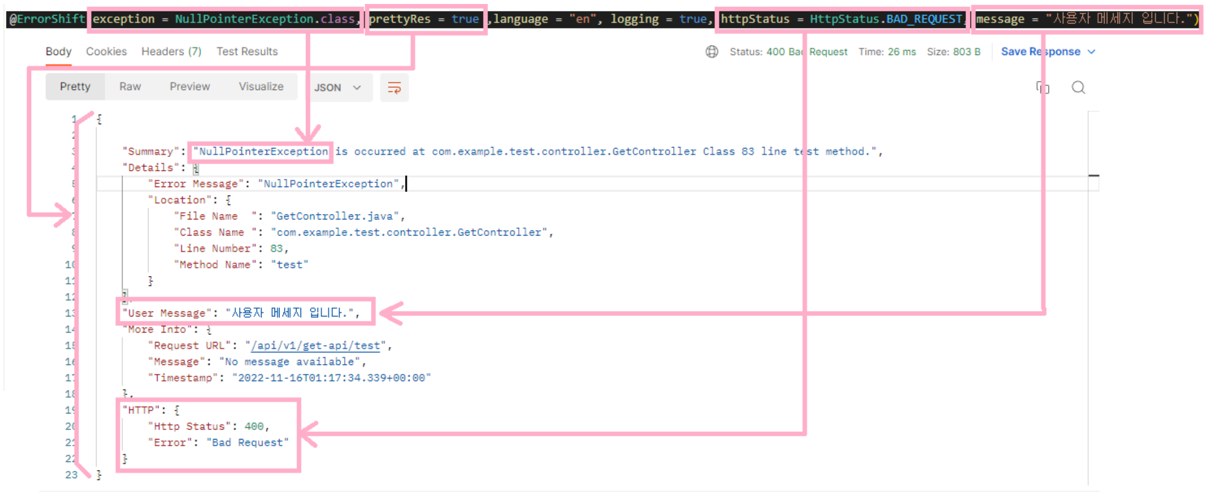
```
server {
    if ($host = www.errorshift.com) {
        return 301 http://errorshift.com$request_uri;
    }
    if ($host = k7e206.p.ssafy.io) {
        return 301 http://errorshift.com$request_uri;
    }
    if ($host = 13.124.236.47) {
        return 301 http://errorshift.com$request_uri;
    }

    listen 80;
    listen [::]:80;
}

server {
    listen 80;
    listen [::]:80;
    server_name errorshift.com;
    server_tokens off;

    location / {
        return 301 https://errorshift.com$request_uri;
    }
}

server {
    if ($host = www.errorshift.com) {
        return 301 https://errorshift.com$request_uri;
    }
    if ($host = k7e206.p.ssafy.io) {
        return 301 https://errorshift.com$request_uri;
    }
    if ($host = 13.124.236.47) {
        return 301 https://errorshift.com$request_uri;
    }

    listen 443 ssl;
    listen [::]:443 ssl;

    ssl_certificate /etc/letsencrypt/live/errorshift.com/fullchain.pem;
    ssl_certificate_key /etc/letsencrypt/live/errorshift.com/privkey.pem;
}

server {
    listen 443 ssl;
    listen [::]:443 ssl;
    server_name errorshift.com;
    server_tokens off;

    ssl_certificate /etc/letsencrypt/live/errorshift.com/fullchain.pem;
    ssl_certificate_key /etc/letsencrypt/live/errorshift.com/privkey.pem;

    root /usr/share/nginx/html;
    index index.html;

    location / {
```

```
                proxy_pass http://front;
            }

            location /file {
                proxy_pass http://file_server;
            }
        }

        log_format  main  '$remote_addr - $remote_user [$time_local] "$request" '
                          '$status $body_bytes_sent "$http_referer" '
                          '"$http_user_agent" "$http_x_forwarded_for"';
        access_log  /var/log/nginx/access.log  main;

        sendfile        on;
        keepalive_timeout  65;
        include /etc/nginx/conf.d/*.conf;
    }
```

2. 시연 시나리오

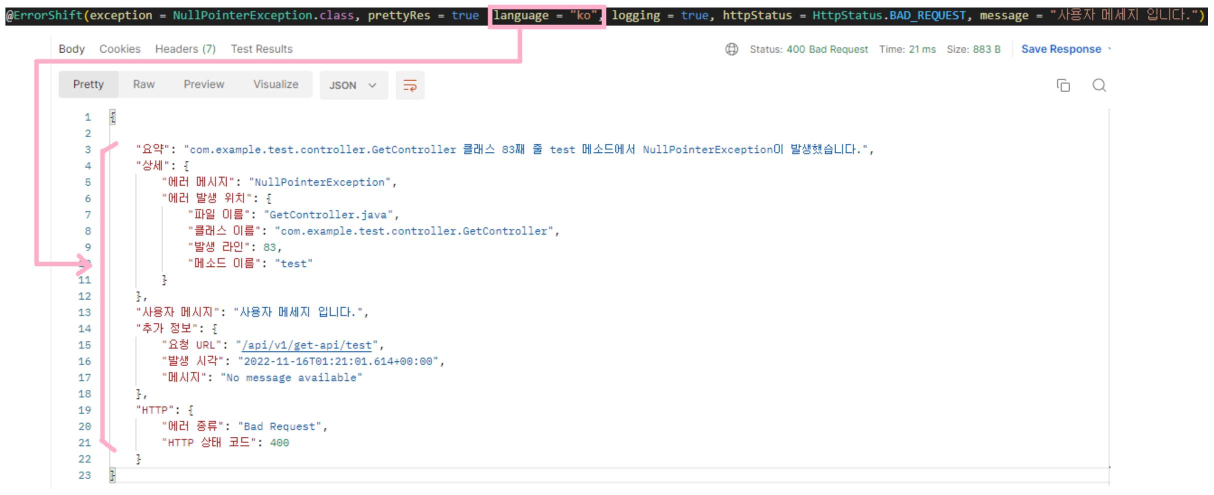


```
import com.ssafy.e206.annotation.ErrorShift;

@ErrorShift(exception = NullPointerException.class, userResPackage = "com.example.test.response.UserResponse")
@ErrorShift(exception = HttpRequestMethodNotSupportedException.class,language = "ko", logging = true)
@ErrorShift(exception = HttpMediaTypeNotSupportedException.class, language = "ko")
@ErrorShift(exception = ArithmeticException.class, language = "ko", message = "Arithmetic -> 수 문제 때문에 나는 에러")
@ErrorShift(exception = ArrayIndexOutOfBoundsException.class, language = "ko", message = "배열 인덱스 확인 하기")
@ErrorShift(exception = ClassCastException.class, language = "ko", prettyRes = true, trace = true )
@ErrorShift(exception = IndexOutOfBoundsException.class, language = "ko" , message =  " 인덱스 범위 확인 해보기")
@ErrorShift(exception = NumberFormatException.class, language = "ko", message =  "0 으로 나눌 때 발생할 수 있음.")
@ErrorShift(exception = MaxUploadSizeExceededException.class, language = "ko", trace = true, prettyRes = true, logging = true)
@ErrorShift(exception = NoHandlerFoundException.class, language = "ko", message = "요청 받는 애가 없음" )
@ControllerAdvice
public class GlobalExceptionHandler {
}
```



```
@ErrorShift(exception = NullPointerException.class, prettyRes = true ,language = "en", logging = true, httpStatus = HttpStatus.BAD_REQUEST, message = "사용자 메세지 입니다.")
```

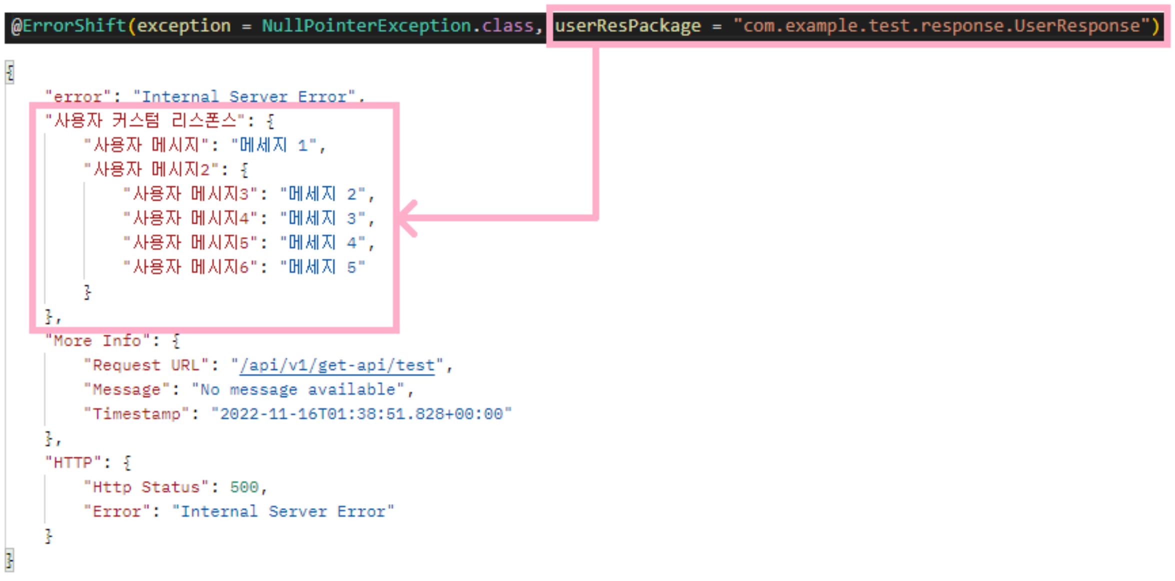
```
{

    "Summary": "NullPointerException is occurred at com.example.test.controller.GetController Class 83 line test method.",
    "Details": {
        "Error Message": "NullPointerException",
        "Location": {
            "File Name  ": "GetController.java",
            "Class Name ": "com.example.test.controller.GetController",
            "Line Number": 83,
            "Method Name": "test"
        }
    },
    "User Message": "사용자 메세지 입니다.",
    "More Info": {
        "Request URL": "/api/v1/get-api/test",
        "Message": "No message available",
        "Timestamp": "2022-11-16T01:17:34.339+00:00"
    },
    "HTTP": {
        "Http Status": 400,
        "Error": "Bad Request"
    }
}
```



```
@ErrorShift(exception = NullPointerException.class, prettyRes = true language = "ko", logging = true, httpStatus = HttpStatus.BAD_REQUEST, message = "사용자 메세지 입니다.")
```

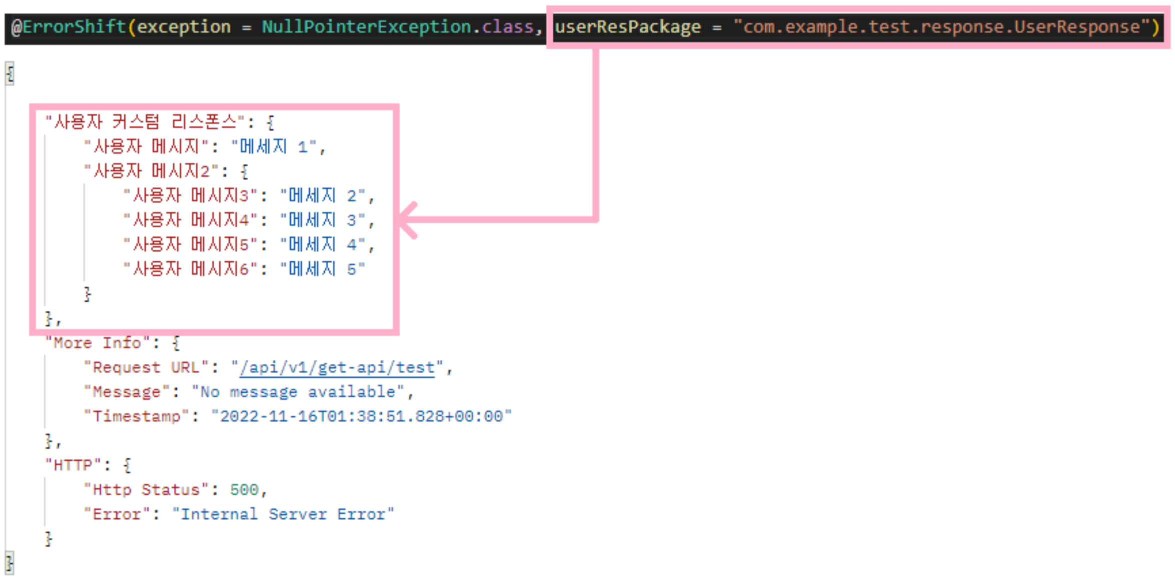
Status: 400 Bad Request Time: 21 ms Size: 883 B

```
{

    "요약": "com.example.test.controller.GetController 클래스 83째 줄 test 메소드에서 NullPointerException이 발생했습니다.",
    "상세": {
        "에러 메시지": "NullPointerException",
        "에러 발생 위치": {
            "파일 이름": "GetController.java",
            "클래스 이름": "com.example.test.controller.GetController",
            "발생 라인": 83,
            "메소드 이름": "test"
        }
    },
    "사용자 메시지": "사용자 메세지 입니다.",
    "추가 정보": {
        "요청 URL": "/api/v1/get-api/test",
        "발생 시각": "2022-11-16T01:21:01.614+00:00",
        "메시지": "No message available"
    },
    "HTTP": {
        "에러 종류": "Bad Request",
        "HTTP 상태 코드": 400
    }
}
```



```
@ErrorShift(exception = NullPointerException.class, userResPackage = "com.example.test.response.UserResponse")
```

```
{
    "error": "Internal Server Error",
    "사용자 커스텀 리스폰스": {
        "사용자 메시지": "메세지 1",
        "사용자 메시지2": {
            "사용자 메시지3": "메세지 2",
            "사용자 메시지4": "메세지 3",
            "사용자 메시지5": "메세지 4",
            "사용자 메시지6": "메세지 5"
        }
    },
    "More Info": {
        "Request URL": "/api/v1/get-api/test",
        "Message": "No message available",
        "Timestamp": "2022-11-16T01:38:51.828+00:00"
    },
    "HTTP": {
        "Http Status": 500,
        "Error": "Internal Server Error"
    }
}
```



```
@ErrorShift(exception = NullPointerException.class, userResPackage = "com.example.test.response.UserResponse")
```

```
{

    "사용자 커스텀 리스폰스": {
        "사용자 메시지": "메세지 1",
        "사용자 메시지2": {
            "사용자 메시지3": "메세지 2",
            "사용자 메시지4": "메세지 3",
            "사용자 메시지5": "메세지 4",
            "사용자 메시지6": "메세지 5"
        }
    },
    "More Info": {
        "Request URL": "/api/v1/get-api/test",
        "Message": "No message available",
        "Timestamp": "2022-11-16T01:38:51.828+00:00"
    },
    "HTTP": {
        "Http Status": 500,
        "Error": "Internal Server Error"
    }
}
```