

## 게임 가드 검증 서버 Docker, Docker-Compose 파일

```
FROM node:17-slim AS builder

WORKDIR /app

COPY ./desperado-server/package*.json ./
COPY ./desperado-server/tsconfig*.json ./

RUN apt-get update || : && apt-get install python3 -y

RUN apt-get install build-essential -y

RUN npm install npm

RUN npm install

COPY ../desperado-server .
```

```
#docker compose -f docker-compose-gameguard.yml up -d
```

```
version: "3.8"
services:
  game-guard-server:
    container_name: dsp-game-guard-server
    build:
      context: ./
      dockerfile: ./desperado-server/gameguard_local.Dockerfile

    platform: linux/amd64
    ports:
      - "8190:8080"
      - "10100:9229"
    volumes:
      - ./desperado-server:/app
      - exclude:/app/node_modules

    command: npm run start:debug:watch gameguard

    restart: always
volumes:
  exclude:
```

Service 부분 - native 라이브러리 호출을 위한 Init 과정, User Instance 선언 및 User Instance 관리를 위한 Object 선언

## Desperado Game-Guard Server 1.0 OAS3

The Game-Guard API description

### gameguard

POST	/gameguard/init	게임가드 init	✓	🔒
POST	/gameguard/get-packet	게임가드 인증 데이터 전달	✓	🔒
POST	/gameguard/check	게임가드 인증	✓	🔒
GET	/gameguard	게임가드 인스턴스 메모리 테스트	✓	🔒

```
export class UserInstance {
    5+ usages   👤 조길호
    userInstance: any;
    5+ usages   👤 조길호
    ttl: Date;
}

const gameGuardInstance: { [key: number]: UserInstance } = {};

4 usages   👤 조길호 *
@Injectable()
export class GameGuardService {
    👤 조길호
    private csa: any;
    👤 조길호
    private ffi: any;
    👤 조길호
    private ref: any;
    👤 조길호
    private readonly intPtr: any;
    👤 조길호
    private readonly bytePtr: any;
    👤 조길호
    private initResult: number;
```

```
constructor() {
  this.ffi = require(id: 'ffi-napi');
  this.ref = require(id: 'ref-napi');
  this.intPtr = this.ref.refType('int');
  this.bytePtr = this.ref.refType('uchar');
  const myLibraryPath = '/app/GameGuard/libggsrv30_x64';
  this.csa = this.ffi.Library(myLibraryPath, {
    // Linux //var csa = ffi.Library('./ggsrv30', { // Windows
    InitCSAuth3: ['int', ['string']],
    CSAuth3_CreateInstance: [this.bytePtr, []],
    CSAuth3_Release: ['void', [this.bytePtr]],
    CSAuth3_Init: ['int', [this.bytePtr, 'int']],
    CSAuth3_Close: ['int', [this.bytePtr]],
    CSAuth3_Get: ['int', [this.bytePtr, this.intPtr]],
    CSAuth3_Check: ['int', [this.bytePtr, this.bytePtr, 'int']],
    CSAuth3_GetPacket: [this.bytePtr, [this.bytePtr]],
    CloseCSAuth3: ['int', []],
  });
  this.initResult = this.csa.InitCSAuth3('/app/GameGuard/');
}
```

일정 시간 동안 활동이 없는 User Instance 릴리즈를 위한 Cron-Job

```
@Cron(CronExpression.EVERY_30_SECONDS)
releaseUserGameGuardInstanceOverExpireTime() {
  this.logger.log('call release user gameguard instance');
  Object.keys(gameGuardInstance).forEach((key) => {
    const ttl = gameGuardInstance[key].ttl;
    let expireTime = new Date(ttl);
    expireTime = TimeUtil.addSeconds(expireTime, addSecond: 300);

    if (expireTime < TimeUtil.now()) {
      const result = this.csa.CSAAuth3.Close(
        gameGuardInstance[key].userInstance,
      );

      // result = 0 : 유저 게임가드 인스턴스 릴리즈 성공
      this.logger.log(`user ${key} instance is terminated, result ${result}`);

      delete gameGuardInstance[key];
    }
  });
}
```

유저 게임가드 검증 인스턴스 생성 부분. 중복 생성 방지 처리 하였습니다.

```
async init(userId: number) {
    const gameGuardInitOutDto = new GameGuardInitOutDto();
    gameGuardInitOutDto.isSuccess = false;
    gameGuardInitOutDto.userInstanceInitResult = -1;
    gameGuardInitOutDto.moduleInitResult = this.initResult;
    // 모듈 초기화 실패 시 바로 리턴
    if (this.initResult !== 0) {
        return gameGuardInitOutDto;
    }
    // 게임가드 인스턴스 생성
    if (!gameGuardInstance.hasOwnProperty(userId)) {
        const ins = this.csa.CSAuth3_CreateInstance();
        const userInstance = new UserInstance();
        userInstance.userInstance = ins;
        userInstance.ttl = TimeUtil.now();
        gameGuardInstance[userId] = userInstance;
    }
    // 인스턴스 초기화
    const userInstanceInitResult = this.csa.CSAuth3_Init(
        gameGuardInstance[userId].userInstance,
        1,
    );
    gameGuardInitOutDto.userInstanceInitResult = userInstanceInitResult;

    if (userInstanceInitResult === 0) {
        gameGuardInitOutDto.isSuccess = true;
    }

    return gameGuardInitOutDto;
}
```