#######FrontEnd Setting######

ApiNode/config.hocon file --> gamefront : url

--- 클라이언트 도메인은 .com 형태로 되야함

#######백엔드 도메인 세팅######

-- 게임백엔드 도메인

ApiNode/config.hocon

gameBackend {

gs = "uat-wbslot-fd.jilidemo1.com" // 웹소켓

be = "uat-wbwebapi.jilidemo2.com" // api 도메인

}

--게임백엔드 웹소켓 도메인

FrontNode/config.hocon

websocket {

url = <https://uat-wbslot-fd.jilidemo1.com/lifeservice/ws2/> //웹소켓 엔드포인트

acceptlimit = 5000

}

--Spin database

SpinType 🡺 0 : normal spin

SpinType 🡺 1 : free spin

SpinType 🡺 2 : buy free spin

-- 요청처리방식

FrontNode/HttpService/JiliWebApiController 에서 프런트 웹요청들을 처리.  
 웹소켓은 단지 heartbeat 로만 사용되지만 반드시 필수입니다.

지금 프로젝트에 포함되있는 클라이언트들의 요청 형식 – protobuf 직렬화방식

FrontNode/ProtoGenerated/ 🡪 프로토파일들과 그에 의해 생성된 .cs 파일들이 저장되여있음.

application/x-protobuf http 요청타입.

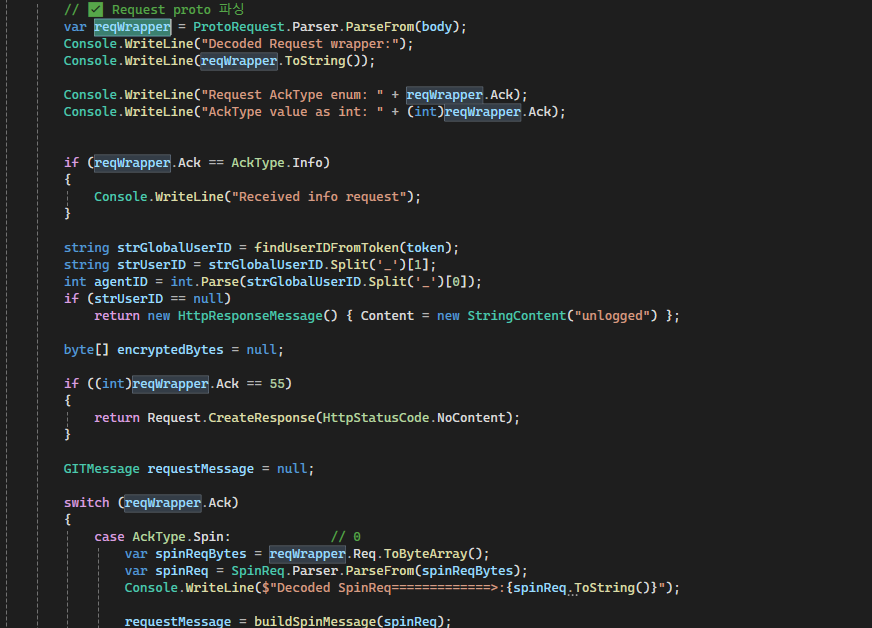
* 디코드순서

GaiaResponse -> RequestProto -> SpinAck(이 프로토는 매 게임들마다 각이합니다.)

* 인코드순서

SpinAck -> RequestProto -> GaiaResponse

reqWrapper 의 ack 마당을 식별하여 모든 요청을 아래와 같이 switch 문으로 처리.



(갱신된 정품사 클라이언트들은 webassem 을 사용하여 전체 요청을 감싸던 gaiareponse 의 wrapper 로 새로운 객체타입을 도입.

게임클라이언트들에서 사용되는 Webassem 파일자체를 akka .net 프로젝트에서 임포트하여 그 wrapper부분을 프로젝트에서 사용하고 업데이트된 게임들을 파싱할수 있는 가능성있음. )