언리얼 페스트 2024 서울

프로젝트로 알아보는 에픽온라인 서비스 (Epic online Service)

김정욱
Senior Game Security Engineer
Epic Games

Epic Online Service

Online Service 개요.
Online Service 에서 제공되는 기능 설명

개발자 포털
Online service를 사용하기 위한 필요한 개발자 포털 설명

Online Subsystem(EOS) Plugin
EOS Plug-in를 UE프로젝트에 적용 예시

Q/A

Online Service 개요

Online Service 에서 제공되는 기능

Your experience, your choice:

게임 개발을 위해 모듈식으로 설계
모든 플랫폼에 배포
모든 친구와 소통

Game Services

한번의 코드 작성을 통하여 모든 플랫폼과 계정 시스템을 연동하여 글로벌 규모로 배포하세요.

Account Services

Epic Games 계정 서비스를 활용하여 자체 게임을 개발하고, 이를 통해 대규모로 연결된 플레이어 기반을 구축하는 데 드는 노력을 줄일 수 있습니다.

EPIC GAMES ONLINE SERVICES

게임 서비스

- **Multiplayer**
  - Matchmaking
  - Voice Chat
  - Lobbies
  - Peer-to-peer
- **Progression**
  - Stats
  - Player Data Storage
  - Achievements
  - Leaderboards
- **Moderation**
  - Anti-cheat
  - Player Reports
  - Player Sanctions
  - Kids Web Services
- **Operation**
  - Title Storage
  - Game Analytics
  - Player ticketing
  - Kids Web Services

계정 서비스

- **Account**
  - Account Management
  - Account Linking
  - Login
- **Friends**
  - Presence
  - Friends list
- **Social**
  - Game Invites
- **Overlay**
  - Settings
  - Profile
  - UI

데브 포털

Online service를 사용하기 위한 필요한 개발자 포털 설명

개요

데브 포털

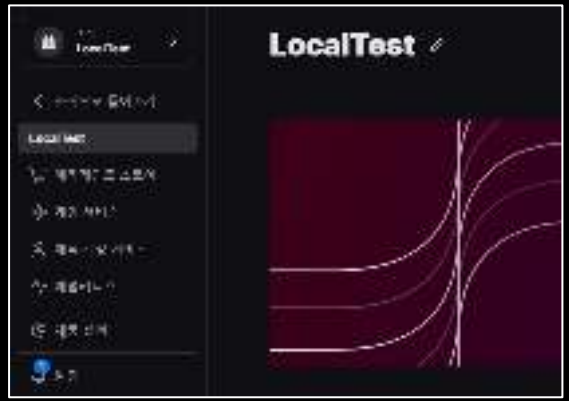
개요

데브 포털

조직

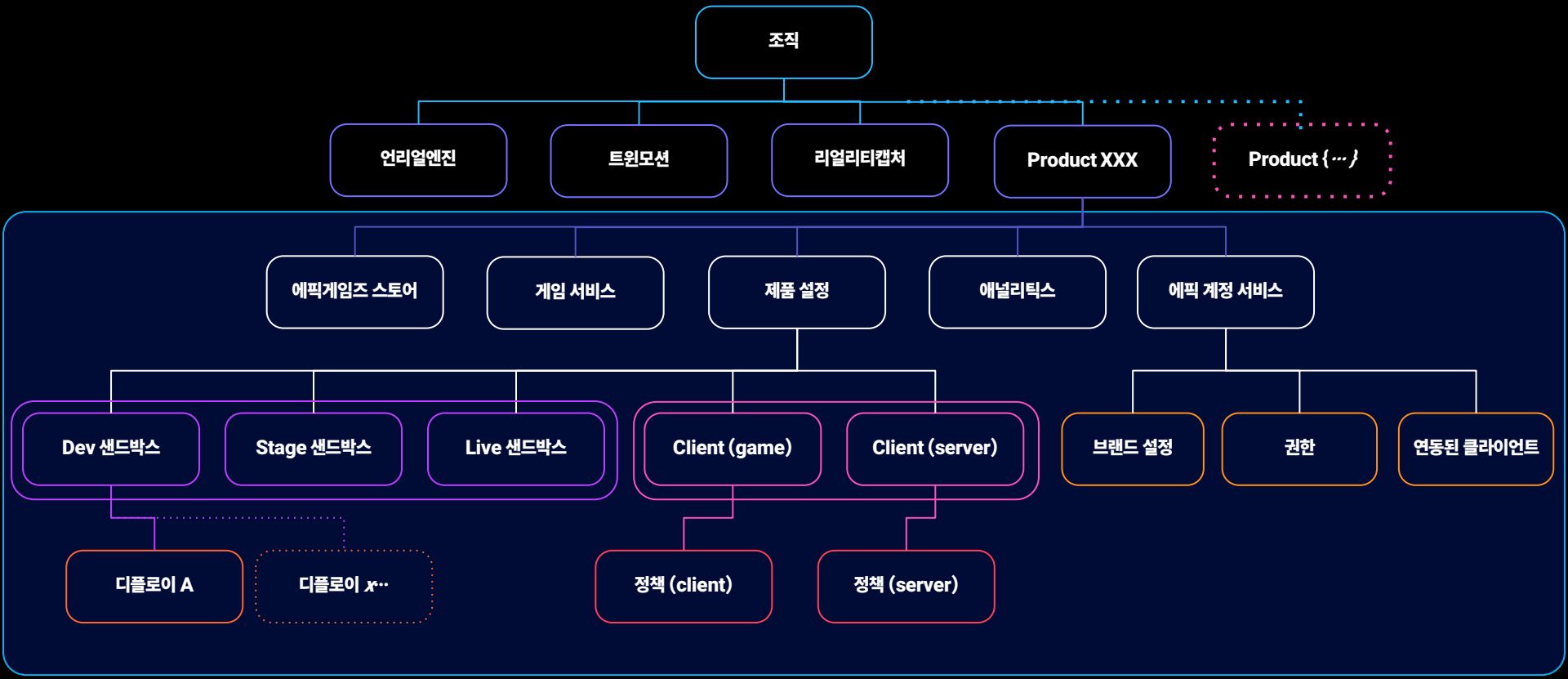
Product XXX

게임 서비스
계정 서비스



개요

데브 포털


조직
언리얼엔진
트윈모션
리얼리티캡처
Product XXX
Product {…}
에픽게임즈 스토어
게임 서비스
제품 설정
애널리틱스
에픽 계정 서비스
Dev 샌드박스
Stage 샌드박스
Live 샌드박스
Client (game)
Client (server)
브랜드 설정
권한
연동된 클라이언트
디플로이 A
디플로이 x…
정책 (client)
정책 (server)

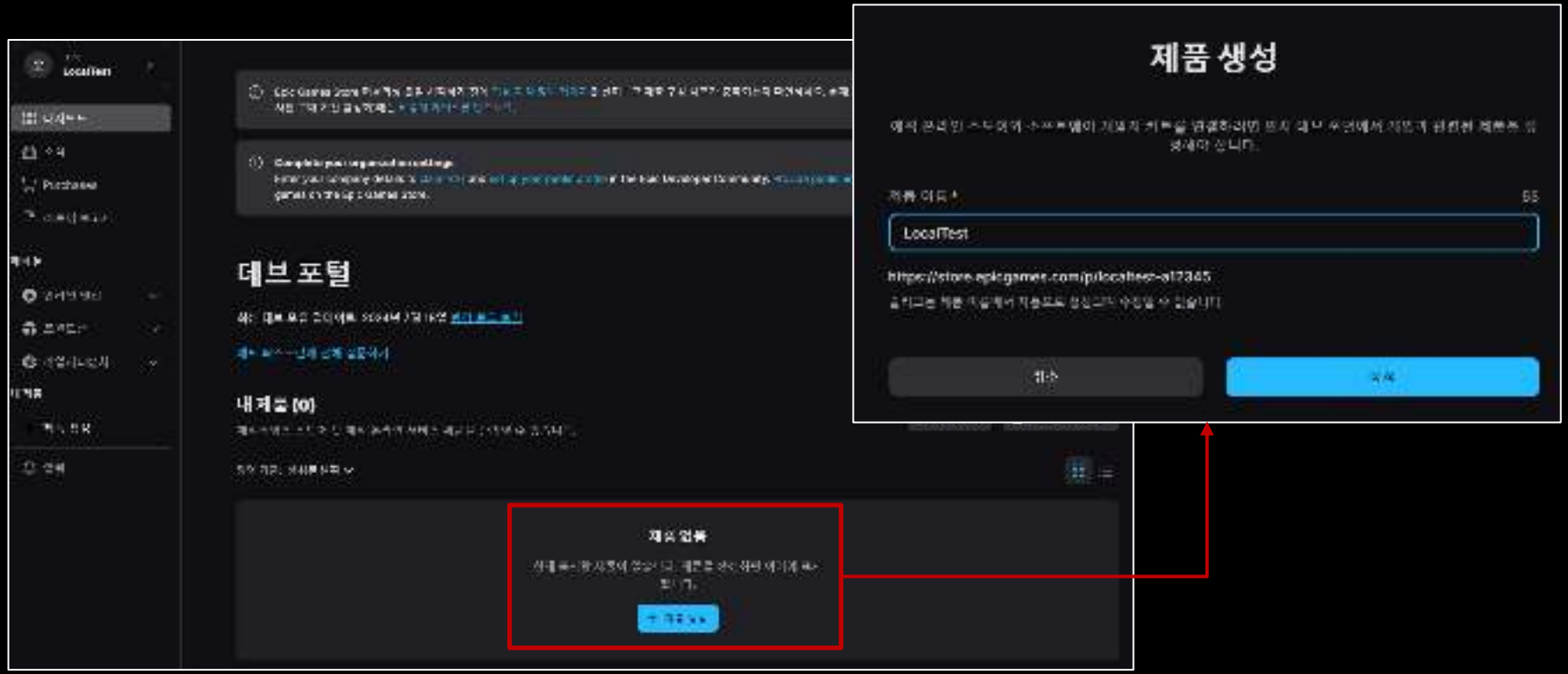
조직생성

데브 포털


조직 생성
먼저, 조직 이름을 입력하고 사업자 등록 국가 또는 거주 국가(해당하는 경우)를 선택하세요.
참고: 조직을 만들면 오너가 된다는 점에 유의하세요. 오너는 멤버를 추가하고, 시트를 관리하고, 지급을 처리할 수 있습니다.
조직 이름 *
0/128
필수입니다.
국가 *
REPUBLIC OF KOREA

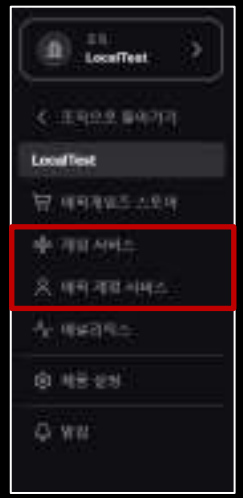
제품 생성

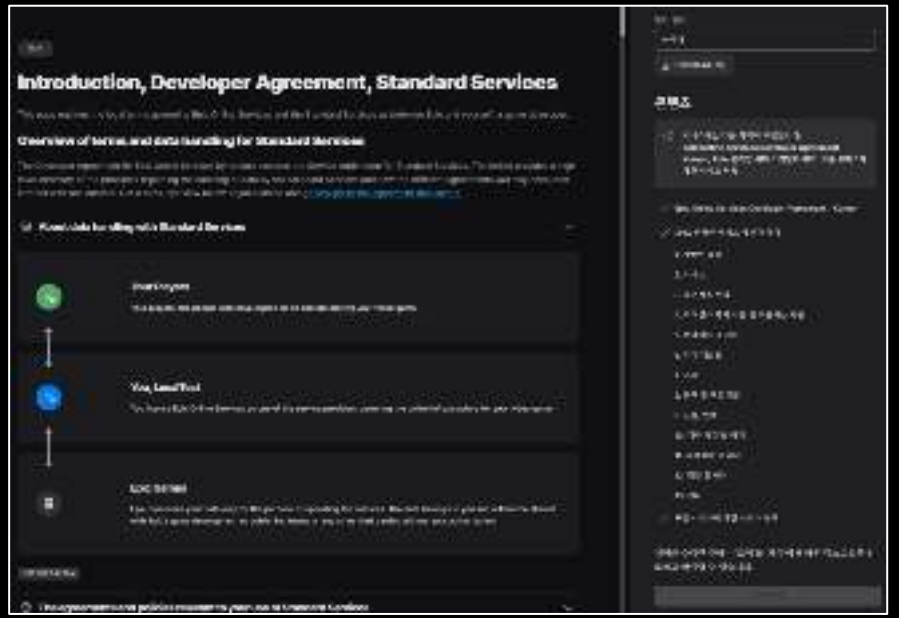
데브 포털


제품 생성
LocalTest
https://store.epicgamer.com/p/localtest-al2345
데브 포털
내게임 (0)
제품 업록

게임서비스, 계정서비스 추가 계약

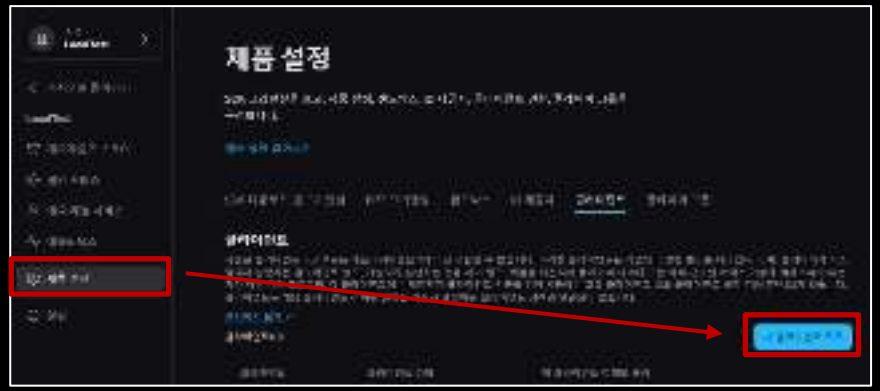
데브 포털





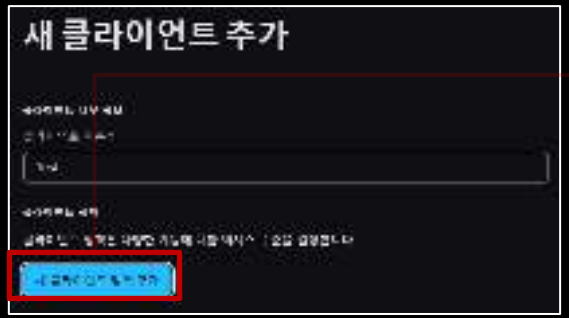
클라이언트 추가

제품 설정



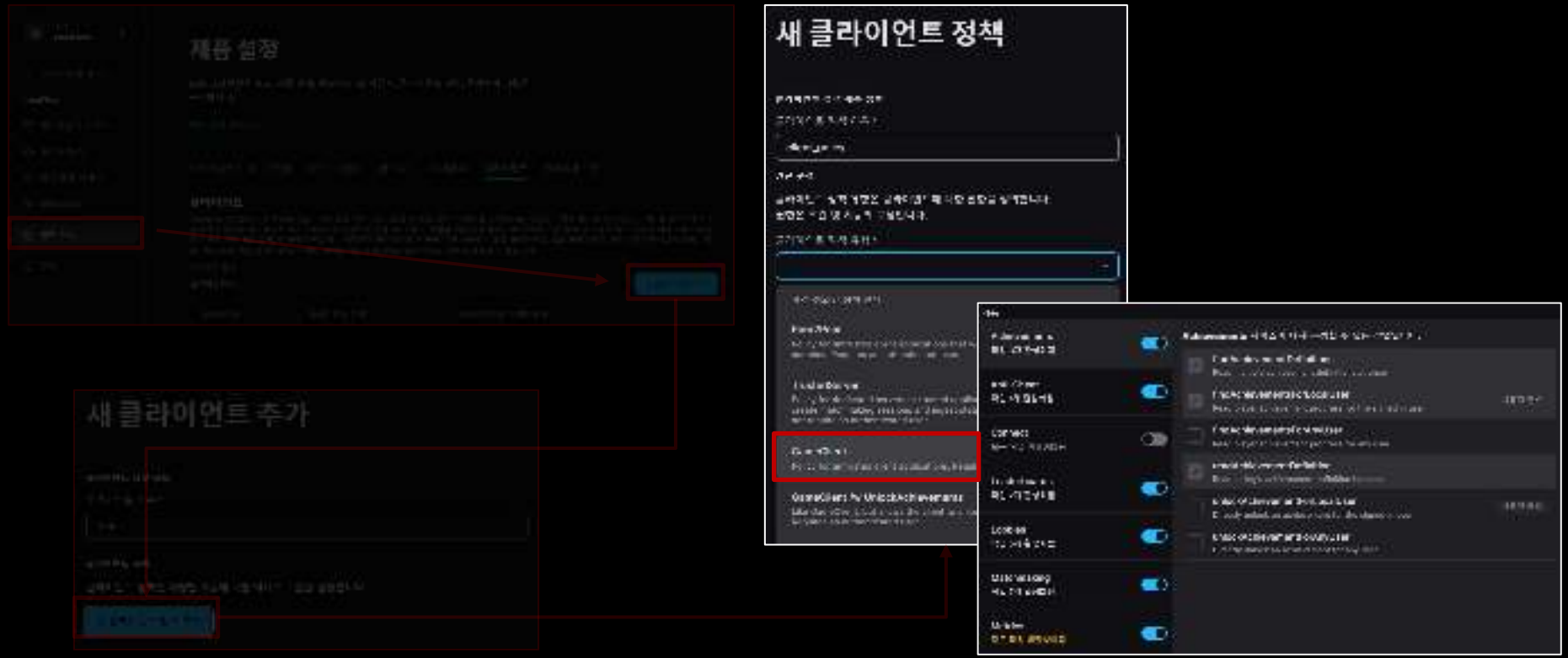
클라이언트 추가

제품 설정



클라이언트 추가

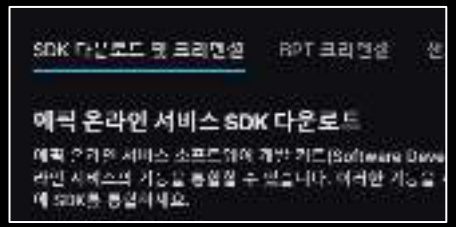
제품 설정



클라이언트 추가

제품 설정

제품 설정

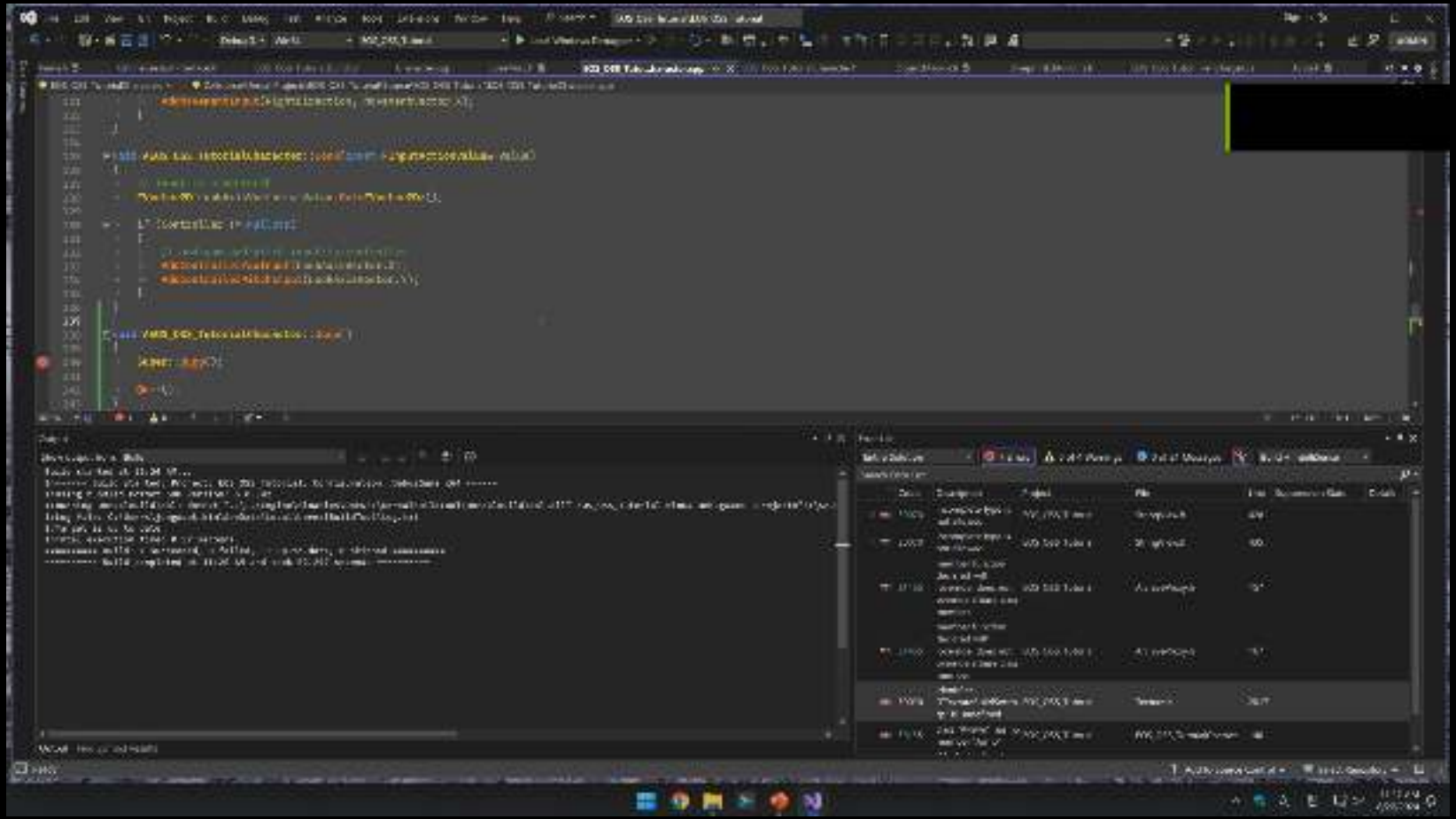


EAS를 통한 계정 설정

EAS를 통한 계정 설정

Online Subsystem(EOS) Plugin

Unreal 에서 제공되는 EOS Plug-in를 UE프로젝트에 적용.



OSS를 사용하기 위한 기본 환경설정

Online Subsystem(EOS) Plugin

강의 자료 URL & QR Code

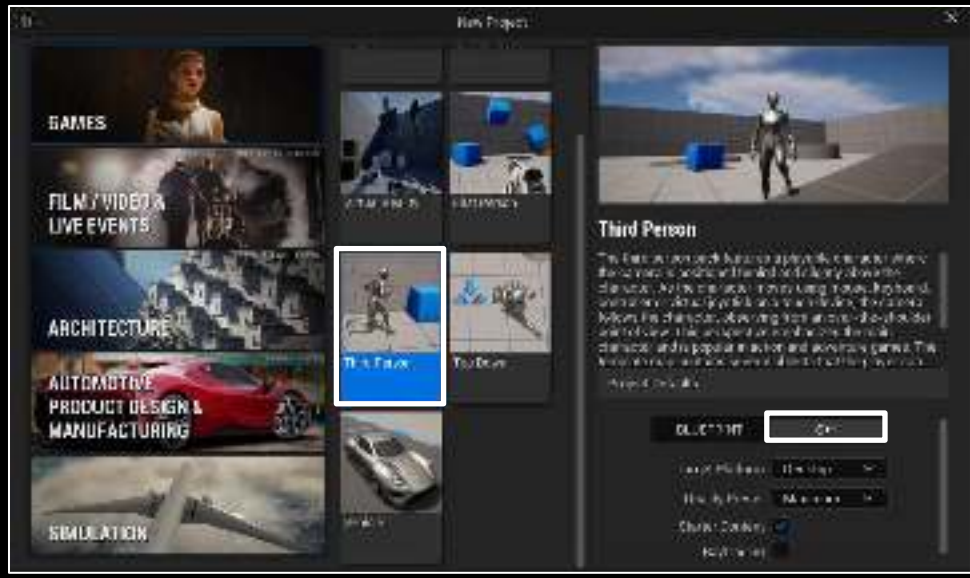
- https://dev.epicgames.com/community/learning/courses/1px/unreal-engine-online-services-the-eos-online-subsystem-oss-plugin/Lnjn/unreal-engine-online-services-introduction

OSS를 사용하기 위한 기본 환경설정

Online Subsystem(EOS) Plugin

Third Person UE 프로젝트 생성

- Games → Third Person
- Project Details → C++
- EOS_OSS_Tutorial 으로 프로젝트 이름을 설정



OSS를 사용하기 위한 기본 환경설정

Online Subsystem(EOS) Plugin

UE 프로젝트에서 Plugin 추가

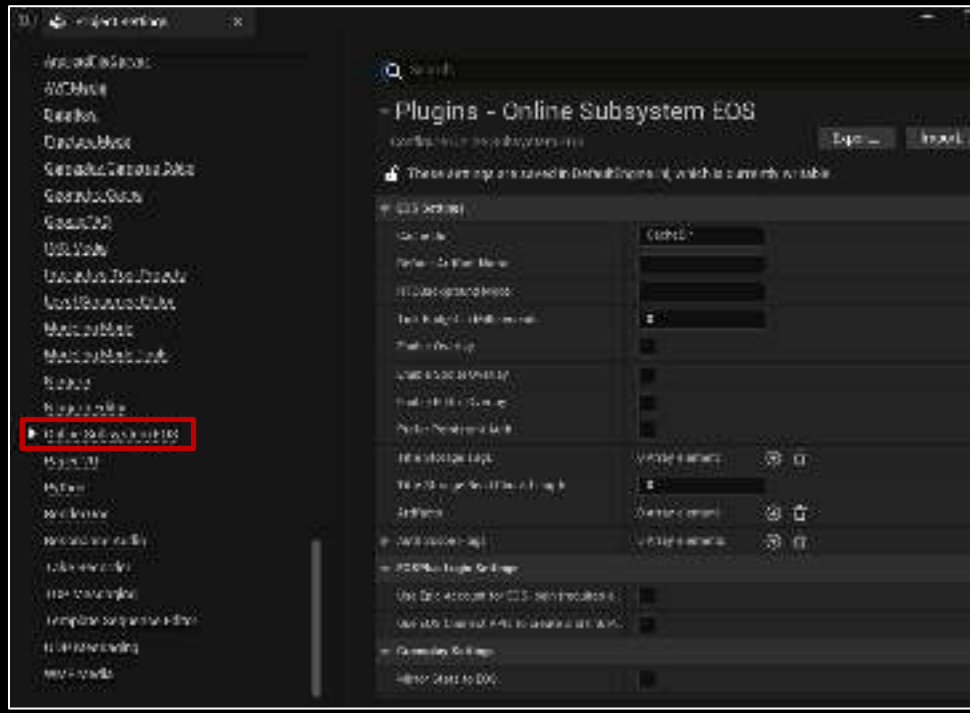
- Edit → Plugins
- EOS 으로 검색
- Online Subsystem EOS 설치

OSS를 사용하기 위한 기본 환경설정

Online Subsystem(EOS) Plugin

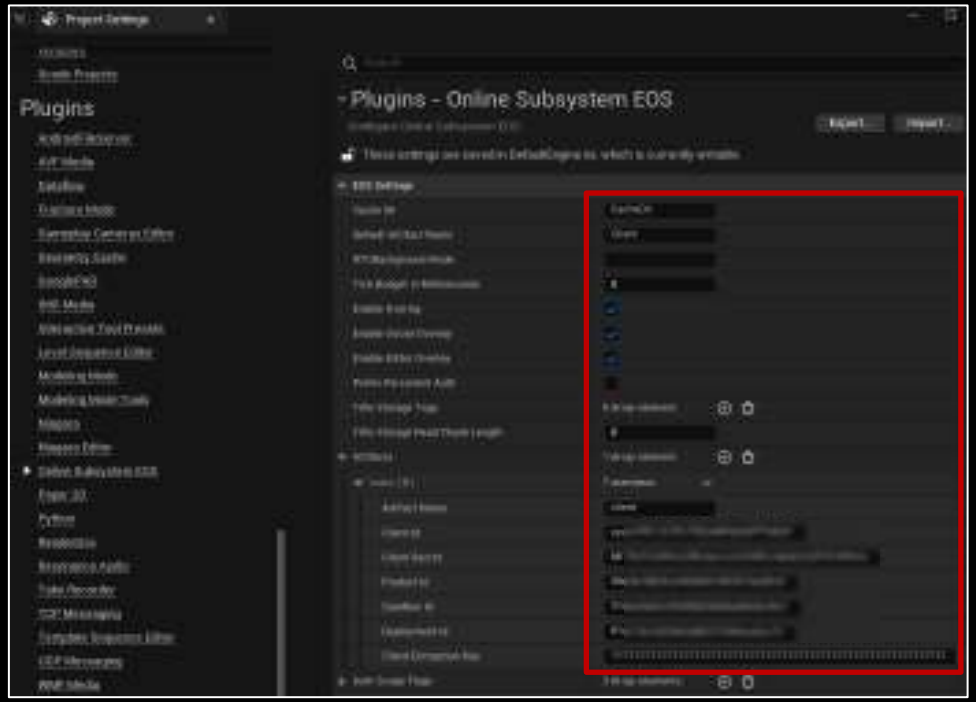
Onlie Subsystem EOS Plugin 설정

- Edit → Project Settings → 하단 Online Subsystme EOS
- Plugin 설치 확인



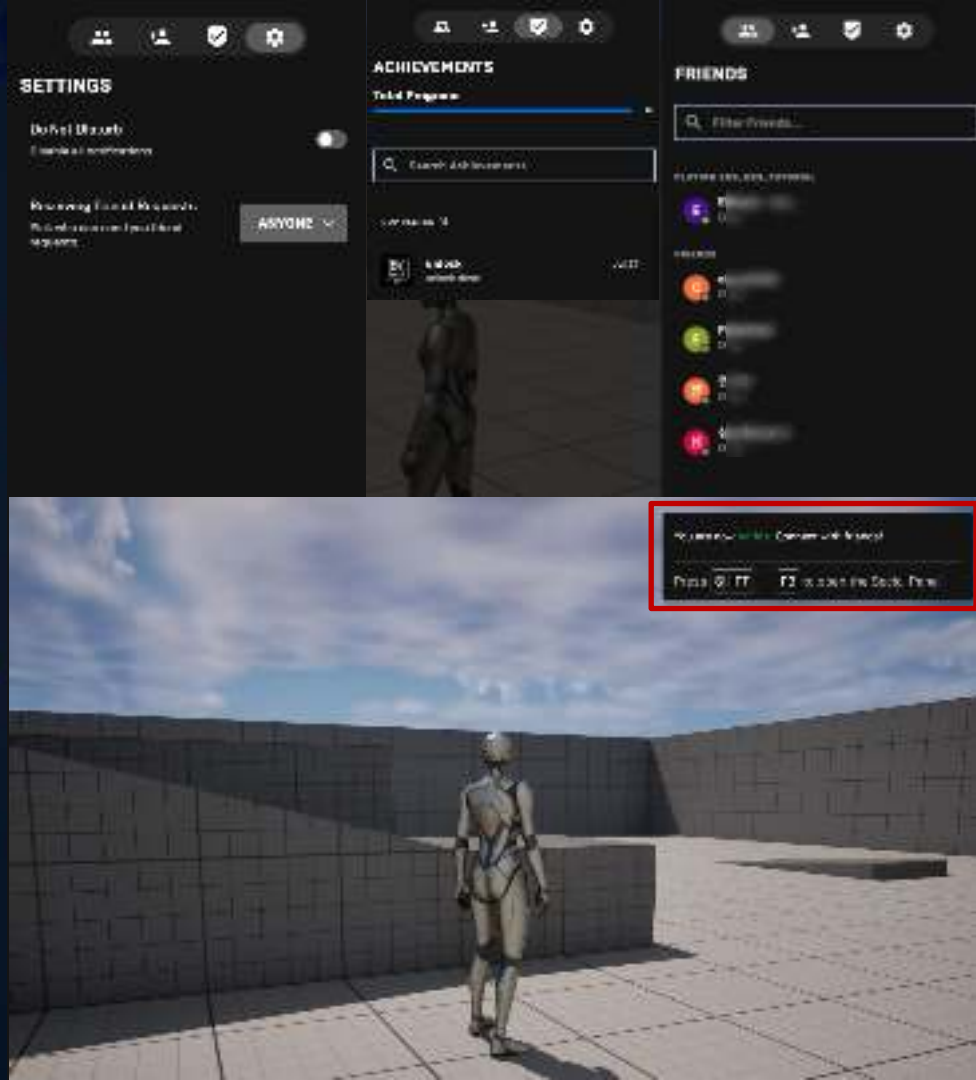
환경설정

Online Subsystem(EOS) Plugin

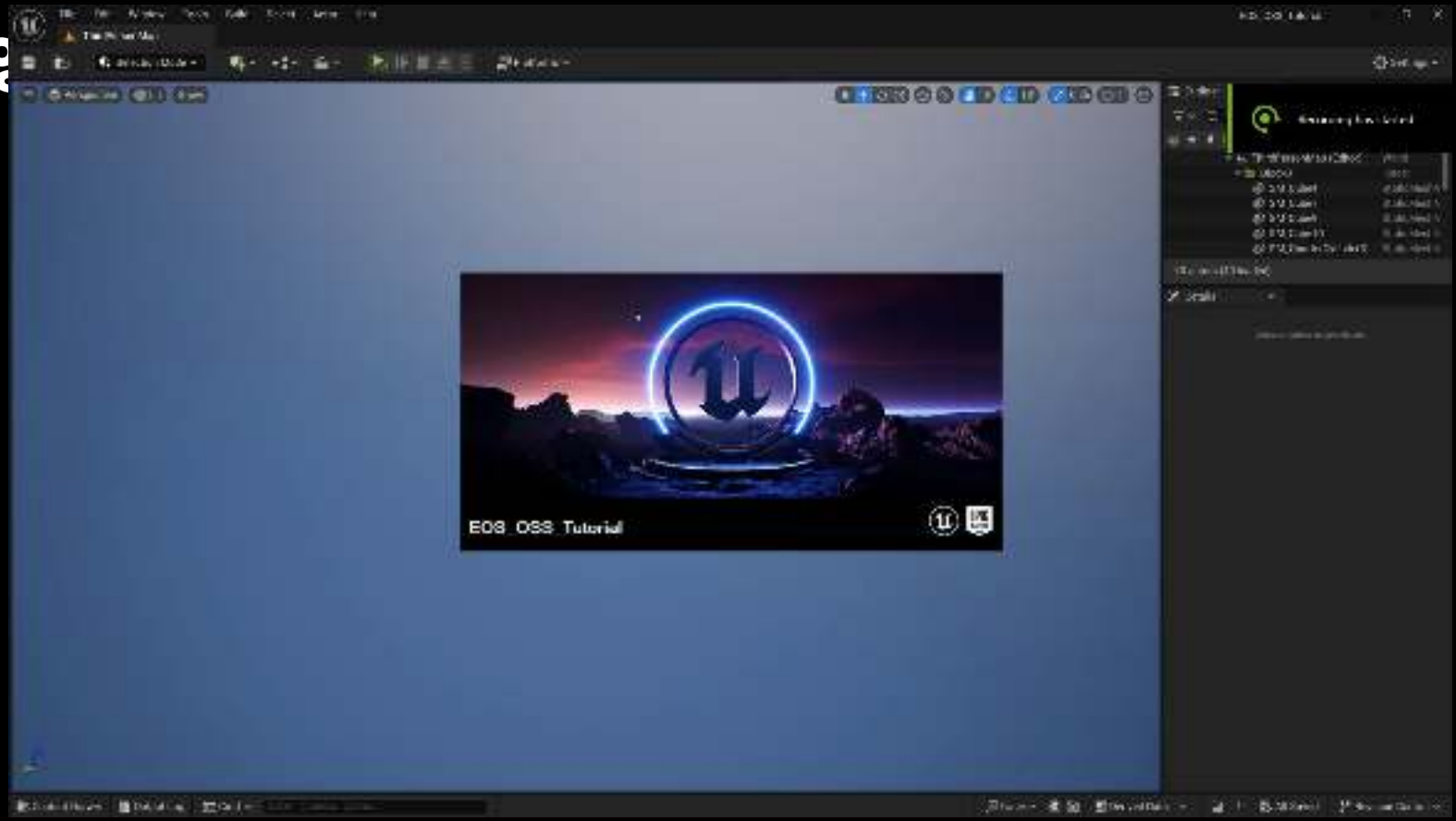


에픽온라인 서비스

OSS 로그인



동영



에픽온라인 서비스 OSS 로그인

{Project}. Build.cs

프로젝트에 필요한 OSS 모듈추가

```
using UnrealBuildTool;

public class EOS_OSS_Tutorial : ModuleRules
{
    public EOS_OSS_Tutorial(ReadOnlyTargetRules Target) :
base(Target)
    {
        PCHUsage = PCHUsageMode.UseExplicitOrSharedPCHs;

        PublicDependencyModuleNames.AddRange(new string[]
            {"Core", "CoreUObject", "Engine", InputCore",
            "HeadMountedDisplay", "EnhancedInput",
            "OnlineSubsystem",
            "OnlineSubsystemUtils",
            "OnlineSubsystemEOS"
        });
    }
}
```

에픽온라인 서비스 OSS 로그인

.CPP

OSS의 각기능을 사용하기 위해 필요한 Header 파일 추가.

```
#include "OnlineSubsystem.h"
#include "OnlineSubsystemUtils.h"
#include "OnlineSubsystemTypes.h"
#include "Interfaces/OnlineIdentityInterface.h"
```

에픽온라인 서비스 OSS 로그인

- 플레이어 로그인에 필요한 코드를 위한 PlayerController class 를 생성.
- IOnlineSubsystem, IOnlineIdentityPtr 인터페이스 이용하여 로그인 구현

에픽온라인 서비스 OSS 로그인

EOSPlayerController.h

로그인에 필요한 기능 선언

```
UCLASS()
class EOS_OSS_TUTORIAL_API AEOSPlayerController :
public APlayerController
{
    GENERATED_BODY()
public:
    AEOSPlayerController();

protected:
    virtual void BeginPlay();
    void Login();
    void HandleLoginCompleted(int32 LocalUserNum,
        bool bWasSuccessful,
        const FUniqueNetId& UserId,
        const FString& Error);
    FDelegateHandle LoginDelegateHandle;
};
```

에픽온라인 서비스 OSS 로그인

EOSPlayerController.cpp

로그인에 필요한 기능 구현

```
void AEOSPlayerController::Login()
{
    IOnlineSubsystem* Subsystem = Online::GetSubsystem(GetWorld());
    IOnlineIdentityPtr Identity = Subsystem->GetIdentityInterface();

    const auto delegate =
        FOnLoginCompleteDelegate::CreateUObject(this,
        &ThisClass::HandleLoginCompleted);
    this->LoginDelegateHandle =
    Identity->AddOnLoginCompleteDelegate_Handle(0, delegate);

    if (!Identity->Login(0, Credentials))
    {
        Identity->ClearOnLoginCompleteDelegate_Handle(
            0, LoginDelegateHandle);
        LoginDelegateHandle.Reset();
    }
}
```

에픽온라인 서비스 OSS 로그인

EOSPlayerController.cpp

로그인에 필요한 기능 구현

```cpp
void AEOSPlayerController::Login()
{
    IOnlineSubsystem* Subsystem = Online::GetSubsystem(GetWorld());
    IOnlineIdentityPtr Identity = Subsystem->GetIdentityInterface();

    const auto delegate =
        FOnLoginCompleteDelegate::CreateUObject(this,
        &ThisClass::HandleLoginCompleted);
    this->LoginDelegateHandle =
    Identity->AddOnLoginCompleteDelegate_Handle(0, delegate);
    ...
    if (!Identity->Login(0, Credentials))
    {
        Identity->ClearOnLoginCompleteDelegate_Handle(
            0, LoginDelegateHandle);
        LoginDelegateHandle.Reset();
    }
}
```

에픽온라인 서비스 OSS 로그인

EOSPlayerController.cpp

로그인에 필요한 기능 구현

```cpp
void AEOSPlayerController::Login()
{
    IOnlineSubsystem* Subsystem = Online::GetSubsystem(GetWorld());
    IOnlineIdentityPtr Identity = Subsystem->GetIdentityInterface();

    const auto delegate =
        FOnLoginCompleteDelegate::CreateUObject(this,
        &ThisClass::HandleLoginCompleted);
    this->LoginDelegateHandle =
    Identity->AddOnLoginCompleteDelegate_Handle(0, delegate);

    if (!Identity->Login(0, Credentials))
    {
        Identity->ClearOnLoginCompleteDelegate_Handle(
            0, LoginDelegateHandle);
        LoginDelegateHandle.Reset();
    }
}
```

에픽온라인 서비스 OSS 로그인

EOSPlayerController.cpp

로그인에 필요한 기능 구현

```cpp
void AEOSPlayerController::HandleLoginCompleted(
    int32 LocalUserNum, bool bWasSuccessful,
    const FUniqueNetId& UserId, const FString& Error)
{
    IOnlineSubsystem* Subsystem = Online::GetSubsystem(GetWorld());
    IOnlineIdentityPtr Identity = Subsystem->GetIdentityInterface();

    if (bWasSuccessful) {
        UE_LOG(LogTemp, Log, TEXT("Login callback completed!"));
    }
    else {
        UE_LOG(LogTemp, Warning, TEXT("EOS login failed."));
    }

    Identity->ClearOnLoginCompleteDelegate_Handle(
        LocalUserNum, LoginDelegateHandle);
    LoginDelegateHandle.Reset();
}
```

에픽온라인 서비스 OSS 로그인

EOSPlayerController.cpp

로그인에 필요한 기능 구현

```cpp
void AEOSPlayerController::HandleLoginCompleted(
    int32 LocalUserNum, bool bWasSuccessful,
    const FUniqueNetId& UserId, const FString& Error)
{
    IOnlineSubsystem* Subsystem = Online::GetSubsystem(GetWorld());
    IOnlineIdentityPtr Identity = Subsystem->GetIdentityInterface();

    if (bWasSuccessful) {
        UE_LOG(LogTemp, Log, TEXT("Login callback completed!"));
    }
    else {
        UE_LOG(LogTemp, Warning, TEXT("EOS login failed."));
    }

    Identity->ClearOnLoginCompleteDelegate_Handle(
        LocalUserNum, LoginDelegateHandle);
    LoginDelegateHandle.Reset();
}
```

에픽온라인 서비스 OSS 로그인

EOSPlayerController.cpp

로그인에 필요한 기능 구현

```cpp
void AEOSPlayerController::HandleLoginCompleted(
    int32 LocalUserNum, bool bWasSuccessful,
    const FUniqueNetId& UserId, const FString& Error)
{
    IOnlineSubsystem* Subsystem = Online::GetSubsystem(GetWorld());
    IOnlineIdentityPtr Identity = Subsystem->GetIdentityInterface();

    if (bWasSuccessful) {
        UE_LOG(LogTemp, Log, TEXT("Login callback completed!"));
    }
    else {
        UE_LOG(LogTemp, Warning, TEXT("EOS login failed."));
    }

    Identity->ClearOnLoginCompleteDelegate_Handle(
        LocalUserNum, LoginDelegateHandle);
    LoginDelegateHandle.Reset();
}
```

에픽온라인 서비스 OSS 로그인

EOSPlayerController.cpp

로그인에 필요한 기능 구현

```
AEOSPlayerController::AEOSPlayerController()
{
}

void AEOSPlayerController::BeginPlay()
{
    Super::BeginPlay();
    Login();
}
```

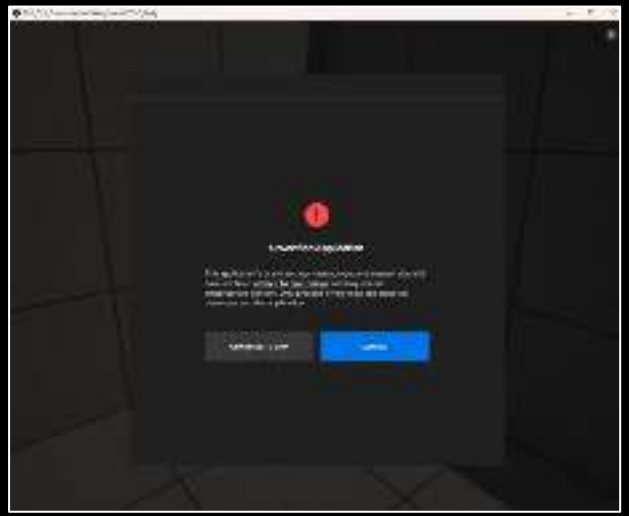
에픽온라인 서비스 OSS 로그인

EOS_OSS_TutorialGameMode.cpp

생성자에 컨트롤러관련 코드 추가

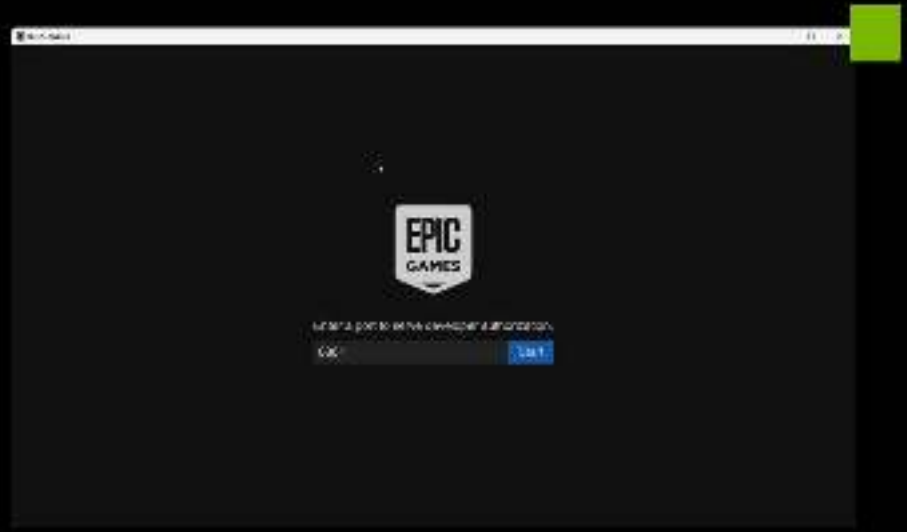
```
AEOS_OSS_TutorialGameMode::AEOS_OSS_TutorialGameMode
()
{
    ...
    PlayerControllerClass =
        AEOSPlayerController::StaticClass();
}
```

에픽온라인 서비스 OSS Login

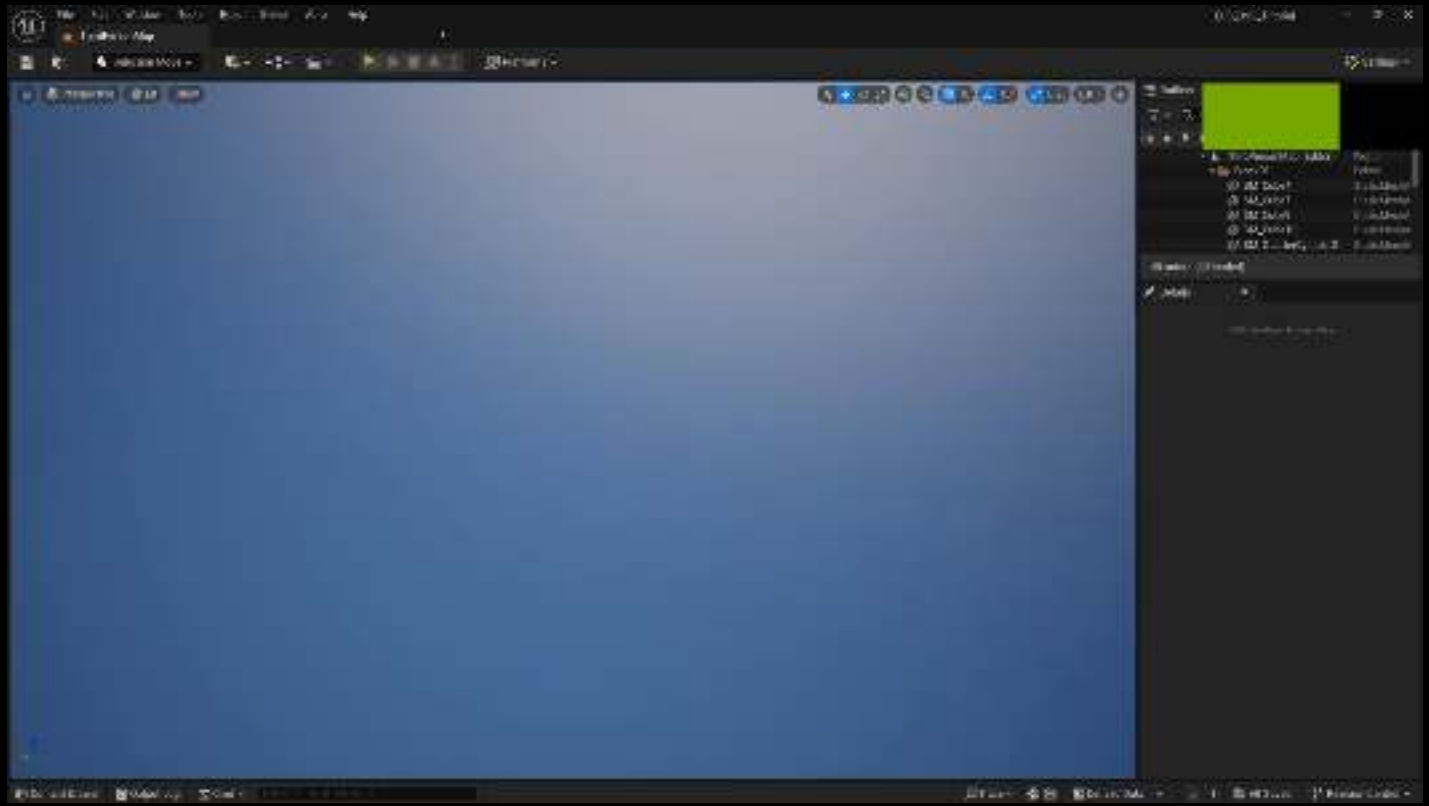


DevAuthTool

- 테스트에 사용한 동일한 계정으로 로그인
- 실행시 CLI에 AUTH_TYPE=developer
  AUTH_LOGIN=localhost:8081
  AUTH_PASSWORD=JW 인자를 추가하여 실행
- AEOSPlayerController::Login() 함수에서 Identity->Login 을 Identity->AutoLogin 으로 변경



DevAuthTool을 통한 로그인



에픽온라인 서비스

플레이어 데이터 저장소

플레이어 데이터 저장소

EOSPlayerController

SaveGame

```
void AEOSPlayerController::SaveGame()
{
    FVector FinalPlayerLocation =
        GetPawn()->GetActorLocation();

    TArray<uint8> SaveData;
    FMemoryWriter MemoryWriter(SaveData);
    MemoryWriter << FinalPlayerLocation;

    WritePlayerDataStorage(
        "CharacterPawnLocation", SaveData);
}
```

플레이어 데이터 저장소

EOSPlayerController

WritePlayerDataStorage

```
void AEOSPlayerController::WritePlayerDataStorage(
    FString FileName, TArray<uint8> FileData)
{
  // Get Subsystem, Identity (생략).
  IOnlineUserCloudPtr UserCloud =
    Subsystem->GetUserCloudInterface();

  WritePlayerDataStorageDelegateHandle =
     UserCloud->AddOnWriteUserFileCompleteDelegate_Handle(
       FOnWriteUserFileCompleteDelegate::CreateUObject(
         this,
         &ThisClass::HandleWritePlayerDataStorageCompleted));

  FUniqueNetIdPtr NetId = Identity->GetUniquePlayerId(0);
  if (!NetId || Identity->GetLoginStatus(*NetId)
          != ELoginStatus::LoggedIn)
  {
      UE_LOG(LogTemp, Log ,
    TEXT("Game won't be saved as player isn't logged in."));
      return;
  }

  if (!UserCloud->WriteUserFile(*NetId, FileName,FileData))
  {
    UE_LOG(LogTemp, Error,
      TEXT("Error writing file with name: %s to player
            data storage"), *FileName);
    UserCloud->
      ClearOnWriteUserFileCompleteDelegate_Handle(
          WritePlayerDataStorageDelegateHandle);
    WritePlayerDataStorageDelegateHandle.Reset();
  }
}
```

플레이어 데이터 저장소

EOSPlayerController

WritePlayerDataStorage

```
void AEOSPlayerController::WritePlayerDataStorage(
    FString FileName, TArray<uint8> FileData)
{
  // Get Subsystem, Identity (생략).
  IOnlineUserCloudPtr UserCloud =
    Subsystem->GetUserCloudInterface();

  WritePlayerDataStorageDelegateHandle =
     UserCloud->AddOnWriteUserFileCompleteDelegate_Handle(
       FOnWriteUserFileCompleteDelegate::CreateUObject(
         this,
         &ThisClass::HandleWritePlayerDataStorageCompleted));

  FUniqueNetIdPtr NetId = Identity->GetUniquePlayerId(0);
  if (!NetId || Identity->GetLoginStatus(*NetId)
          != ELoginStatus::LoggedIn)
  {
      UE_LOG(LogTemp, Log ,
    TEXT("Game won't be saved as player isn't logged in."));
      return;
  }

  if (!UserCloud->WriteUserFile(*NetId, FileName,FileData))
  {
    UE_LOG(LogTemp, Error,
      TEXT("Error writing file with name: %s to player
            data storage"), *FileName);
    UserCloud->
      ClearOnWriteUserFileCompleteDelegate_Handle(
          WritePlayerDataStorageDelegateHandle);
    WritePlayerDataStorageDelegateHandle.Reset();
  }
}
```

플레이어 데이터 저장소

EOSPlayerController

WritePlayerDataStorage

```
void AEOSPlayerController::WritePlayerDataStorage(
    FString FileName, TArray<uint8> FileData)
{
  // Get Subsystem, Identity (생략).
  IOnlineUserCloudPtr UserCloud =
    Subsystem->GetUserCloudInterface();

  WritePlayerDataStorageDelegateHandle =
     UserCloud->AddOnWriteUserFileCompleteDelegate_Handle(
       FOnWriteUserFileCompleteDelegate::CreateUObject(
         this,
         &ThisClass::HandleWritePlayerDataStorageCompleted));

  FUniqueNetIdPtr NetId = Identity->GetUniquePlayerId(0);
  if (!NetId || Identity->GetLoginStatus(*NetId)
          != ELoginStatus::LoggedIn)
  {
      UE_LOG(LogTemp, Log ,
    TEXT("Game won't be saved as player isn't logged in."));
      return;
  }

  if (!UserCloud->WriteUserFile(*NetId, FileName,FileData))
  {
    UE_LOG(LogTemp, Error,
      TEXT("Error writing file with name: %s to player
            data storage"), *FileName);
    UserCloud->
      ClearOnWriteUserFileCompleteDelegate_Handle(
          WritePlayerDataStorageDelegateHandle);
    WritePlayerDataStorageDelegateHandle.Reset();
  }
}
```

플레이어 데이터 저장소

EOSPlayerController

HandleWritePlayerDataStorageCompleted

```
void
AEOSPlayerController::HandleWritePlayerDataStorageCompleted(
    bool bWasSuccessful, const FUniqueNetId& UserId,
    const FString& FileName)
{
    // Get Subsystem, UserCloud (생략).
    if (bWasSuccessful)
    {
        // LOG: SUCCESS.
    }
    else
    {
    // LOG: Error writing file with name: %s to player data storage.
    }

    UserCloud->
    ClearOnWriteUserFileCompleteDelegate_Handle(
        WritePlayerDataStorageDelegateHandle);
    WritePlayerDataStorageDelegateHandle.Reset();
}
```

플레이어 데이터 저장소

EOSPlayerController

LoadPlayerData

```cpp
void AEOSPlayerController::LoadPlayerData()
{
    // Get Subsystem, Identity (생략).
    IOnlineUserCloudPtr PlayerData =
        Subsystem->GetUserCloudInterface();

    EnumPlayerFilesDelegateHandle =
      PlayerData->
        AddOnEnumerateUserFilesCompleteDelegate_Handle(
          FOnEnumerateUserFilesCompleteDelegate::CreateUObject
          (this, &ThisClass::HandleEnumPlayerFilesCompleted));

    FUniqueNetIdPtr NetId = Identity->GetUniquePlayerId(0);
    if (!NetId || Identity->GetLoginStatus(*NetId)
            != ELoginStatus::LoggedIn)
    {
        return;
    }

    PlayerData->EnumerateUserFiles(*NetId);
}
```

플레이어 데이터 저장소

EOSPlayerController

HandleEnumPlayerFilesCompleted

```
void AEOSPlayerController::HandleEnumPlayerFilesCompleted(
    bool bWasSuccessfull, const FUniqueNetId& NetId)
{
    // Get Subsystem, PlayerData (생략).
    if (bWasSuccessfull)
    {
        TArray<FCloudFileHeader> PlayerFiles;
        PlayerData->GetUserFileList(NetId, PlayerFiles);
        if (PlayerFiles.Num() == 0)
        {
            // LOG:No player files in player data storage.
            return;
        }

        TArray<FString> PlayerFileNames;
        for (const auto& File : PlayerFiles)
        {
            PlayerFileNames.Add(File.FileName);
        }
        ...
```

플레이어 데이터 저장소

EOSPlayerController

HandleEnumPlayerFilesCompleted

```
        ...

        ReadPlayerDataFileDelegateHandle =
        PlayerData->AddOnReadUserFileCompleteDelegate_Handle(
            FOnReadUserFileCompleteDelegate::CreateUObject(
            this, &ThisClass::HandleReadPlayerFileCompleted));
        if (!PlayerData->ReadUserFile(
                NetId, PlayerFileNames[0]))
        {
            // LOG: Error reading player data storage file,
            // filename(PlayerFileNames)
        }
    }
    else{
        // LOG: Error enumerating player data storage files.
    }

    PlayerData->
    ClearOnEnumerateUserFilesCompleteDelegate_Handle(
        EnumPlayerFilesDelegateHandle);
    EnumPlayerFilesDelegateHandle.Reset();
}
```

플레이어 데이터 저장소

EOSPlayerController

HandleReadPlayerFileCompleted

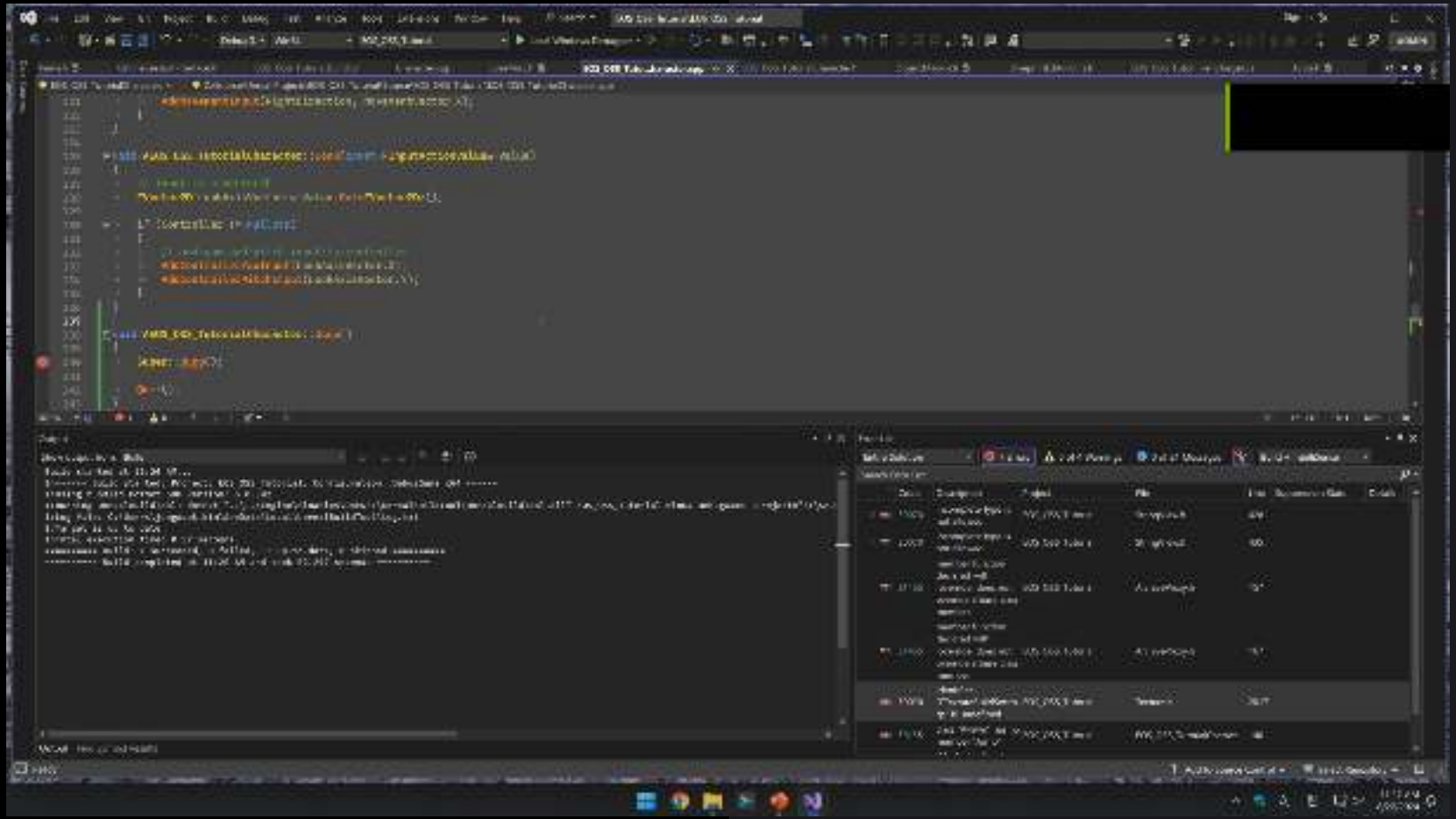
```
void AEOSPlayerController::HandleReadPlayerFileCompleted(
    bool bWasSuccessfull, const FUniqueNetId& UserId,
    const FString& FileName)
{
    // Get Subsystem, PlayerData (생략).
    if (bWasSuccessfull)
    {
        TArray<uint8> FileContents;
        if (PlayerData->GetFileContents(
                UserId, FileName, FileContents))
        {
            // Should use a USaveGame
            LoadGame(FileContents);
        }
        else
        {
            // LOG: Failed to get file contents for file named
            // (FileName)
        }
    }
    else
    {
        // LOG: Error reading player data storage file
        // (FileName)
    }
    PlayerData->
    ClearOnReadUserFileCompleteDelegate_Handle(
        ReadPlayerDataFileDelegateHandle);
    ReadPlayerDataFileDelegateHandle.Reset();
}
```

플레이어 데이터 저장소

EOSPlayerController

HandleLoginCompleted

```
void AEOSPlayerController::HandleLoginCompleted(...)
{
    ...
    if (bWasSuccessful)
    {
        // LOG: Login callback completed!
        LoadPlayerData();
    }
    ...
}
```



에픽온라인 서비스 OSS

EOSPlayerController

Login

```
void AEOSPlayerController::Login()
{
    IOnlineSubsystem* Subsystem = Online::GetSubsystem(GetWorld());
    IOnlineIdentityPtr Identity = Subsystem->GetIdentityInterface();

    const auto delegate =
        FOnLoginCompleteDelegate::CreateUObject(this,
        &ThisClass::HandleLoginCompleted);
    this->LoginDelegateHandle =
    Identity->AddOnLoginCompleteDelegate_Handle(0, delegate);

    if (!Identity->Login(0, Credentials))
    {
        Identity->ClearOnLoginCompleteDelegate_Handle(
            0, LoginDelegateHandle);
        LoginDelegateHandle.Reset();
    }
}
```

에픽온라인 서비스 OSS

EOSPlayerController

Login

```
void AEOSPlayerController::Login()
{
    IOnlineSubsystem* Subsystem = Online::GetSubsystem(GetWorld());
    IOnlineIdentityPtr Identity = Subsystem->GetIdentityInterface();

    const auto delegate =
        FOnLoginCompleteDelegate::CreateUObject(this,
        &ThisClass::HandleLoginCompleted);
    this->LoginDelegateHandle =
    Identity->AddOnLoginCompleteDelegate_Handle(0, delegate);

    if (!Identity->Login(0, Credentials))
    {
        Identity->ClearOnLoginCompleteDelegate_Handle(
            0, LoginDelegateHandle);
        LoginDelegateHandle.Reset();
    }
}
```

에픽온라인 서비스 OSS

EOSPlayerController

Login

```
void AEOSPlayerController::Login()
{
    IOnlineSubsystem* Subsystem = Online::GetSubsystem(GetWorld());
    IOnlineIdentityPtr Identity = Subsystem->GetIdentityInterface();

    const auto delegate =
        FOnLoginCompleteDelegate::CreateUObject(this,
        &ThisClass::HandleLoginCompleted);
    this->LoginDelegateHandle =
    Identity->AddOnLoginCompleteDelegate_Handle(0, delegate);

    if (!Identity->Login(0, Credentials))
    {
        Identity->ClearOnLoginCompleteDelegate_Handle(
            0, LoginDelegateHandle);
        LoginDelegateHandle.Reset();
    }
}
```

감사합니다.

강의 자료 QR Code