

# Unity Backend

“차트” 기능을 이용한 유저 레벨 관리

Created in 2023-06-13

Last Updated 2023-06-13

Unity Version 2022.2.2f1

## *Index*

- ◆ 차트 생성 및 등록
- ◆ 차트 데이터 불러오기
- ◆ 차트 데이터 사용하기

# 차트 생성 및 등록

- 개요
- 차트 생성
- 차트 등록



# 차트 생성 및 등록

## ■ 개요

### ■ 뒤끝에서 "차트"의 역할

- 게임 실행 도중에 변하지 않는 게임 내 데이터 관리  
(장비/아이템, 적, 플레이어의 각종 스탯 정보 등)
  - 이러한 데이터는 서버에 등록해두고 게임을 실행할 때 불러와 사용
  - 서버에 등록해두면 앱 업데이트 없이 데이터 수정 가능
- 우편, 쿠폰 기능과 연동해 차트에 있는 아이템을 유저에게 지급 가능



# 차트 생성 및 등록

## ■ 차트 생성

- 게임에 사용할 데이터를 엑셀 파일을 이용해 생성

LevelChart.xlsx - Excel

	A	B	C	D	E	F
1	level	maxExperience	rewardGold			
2	1	100	1111			
3	2	200	2222			
4	3	400	3333			
5	4	800	4444			
6	5	1600	5555			
7	6	3200	6666			
8	7	6400	7777			
9	8	12800	8888			
10	9	25600	9999			
11	10	51200	100000			
12						
13						

**LevelChart.xlsx**  
Lobby 씬에 출력되는 플레이어의 레벨(level)과 레벨업에 필요한 경험치(maxExperience), 레벨업 시 보상으로 받는 재화(rewardGold) 정보를 담고 있다.



# 차트 생성 및 등록

## ■ 차트 등록

■ Backend Console의 "차트 관리" 탭에 차트 생성

□ 차트 관리 - "차트 생성"

Backend Console

ProjectA

차트 관리 | 폴더 생성 | **차트 생성** | 폴더에 추가 | 삭제

SDK 문서 | 콘솔 가이드

차트 0/200 | 파일 0/200

번호	차트명	적용된 차트	차트 설명	우편 기능
차트가 존재하지 않습니다.				

차트 관리

- 즐거찾기
- 대시보드
- 서버 설정
- 뒤끝베이스
- 유저 관리
- 유저 접근 관리
- 게임 정보 관리
- 랭킹 관리
- 우편 관리
- 푸시 관리
- 쿠폰 관리
- 차트 관리**
- 확률 관리
- 로그 관리
- 영수증 검증
- 길드 관리



# 차트 생성 및 등록

- 사용할 차트에 대한 설명 입력 후 "확인" 버튼을 눌러 차트 생성

Backnd Console ProjectA

차트 관리 | 폴더 생성 | 차트 생성 | 폴더에 추가 | 삭제

차트 관리

번호 | 차트명 | 적용된 차트 | 차트 설명 | 우편 가능

차트가 존재하지 않습니다.

차트 생성

차트명\* | 레벨차트

차트 설명 | 레벨에 필요한 경험치와 레벨업 보상에 관련한 차트입니다.

우편 사용 여부 | 사용 | 사용안함

확인 | 취소

차트에 등록된 데이터(아이템)만 우편, 쿠폰으로 발송 가능하기 때문에 차트를 생성할 때 우편 사용 여부를 체크할 수 있다.  
레벨차트의 경우 순수하게 게임 내 데이터로만 사용하기 때문에 우편 사용 여부는 "사용안함"



# 차트 생성 및 등록

- 생성이 완료된 차트로 들어가 차트 파일 업로드
  - 차트 파일을 업로드할 차트 선택 : "레벨차트" (차트명)

The screenshot shows the Backnd Console interface for 'ProjectA'. The left sidebar contains a navigation menu with items like '즐거찾기', '대시보드', '서버 설정', '뒤끝베이스', '유저 관리', '유저 접근 관리', '게임 정보 관리', '랭킹 관리', '우편 관리', '푸시 관리', '쿠폰 관리', '차트 관리', '확률 관리', '로그 관리', '영수증 검증', '길드 관리', '캐시 관리', and '버전 관리'. The main content area is titled '차트 관리' (Chart Management) and shows a table of charts. The table has columns for '번호' (Number), '차트명' (Chart Name), '적용된 차트' (Applied Chart), '차트 설명' (Chart Description), and '우편 기능' (Mail Function). A single chart is listed with the name '레벨차트' (Level Chart), which is highlighted with a red box. The description for this chart is '레벨에 필요한 경험치와 레벨업 보상에 관련한 차트입니다.' (Chart related to experience points and level-up rewards required for leveling). The page also includes a pagination control showing '1' and a '10개씩 보기' (View 10 at a time) dropdown.

번호	차트명	적용된 차트	차트 설명	우편 기능
1	레벨차트	적용된 차트 없음	레벨에 필요한 경험치와 레벨업 보상에 관련한 차트입니다.	사용안함





# 차트 생성 및 등록

## □ “차트 파일 업로드” 선택

Backnd Console << ProjectA ⚙️ 🔔 📄 ? 4 관리자 👤

ProjectA 개발 모드

★ 차트 관리 **차트 파일 업로드** ⌚ 차트 파일 적용 📄 CSV 다운로드 🗑️ 삭제 SDK 문서 📄 콘솔 가이드 📄

차트 관리 / 레벨차트

레벨차트

현재 적용 차트 파일 적용된 차트가 없습니다.  
\*id 열(column)은 예약된 키이므로 차트에 존재하지 않아야 합니다.

<input type="checkbox"/> 번호	파일명	파일 ID	등록일
차트 파일 내역이 없습니다.			

회사소개 이용약관 서비스수준협약 개인정보처리방침 © AFI, Inc. All rights reserved.



# 차트 생성 및 등록

- 미리 생성해둔 엑셀 파일 등록 (LevelChart.xlsx)

Backnd Console

ProjectA

★ 차트 관리 | 차트 파일 업로드 | 차트 파일 적용 | CSV 다운로드 | 삭제

SDK 문서 | 콘솔 가이드

차트 관리 / 레벨차트

레벨차트

현재 적용 차트 파일

번호

차트 파일 내역이 없습니다.

적용된 차트가 없습니다.  
\*id 열(column)은 예약된 키이므로 차트에 존재하지 않아야 합니다.

열기

#1000 Backend > Unity Backend [비행 슈팅 게임][Resources]

이름	수정한 날짜	유형	크기
Images	2023-02-01 오전 10:26	파일 폴더	
LevelChart.xlsx	2023-06-13 오후 2:11	Microsoft Excel 워크...	9KB

파일 이름(N): LevelChart.xlsx

사용자 지정 파일 (\*.xlsx;\*.xls;\*.c

열기(O) | 취소



# 차트 생성 및 등록

- 등록된 차트를 클릭하면 온라인에서 데이터 확인 및 편집 가능

Backnd Console << ProjectA [설정]

ProjectA 개발 모드

★ 차트 관리 | 차트 파일 업로드 | 차트 파일 적용 | CSV 다운로드 | 삭제

SDK 문서 | 콘솔 가이드

차트 관리 / 레벨차트

레벨차트

현재 적용 차트 파일

적용된 차트가 없습니다.  
\*id 열(column)은 예약된 키이므로 차트에 존재하지 않아야 합니다.

<input type="checkbox"/>	번호	파일명	파일 ID	등록일
<input type="checkbox"/>	1	<span style="border: 2px solid red; padding: 2px;">LevelChart.xlsx</span>	82616	2023.06.13 14:45

< 1 >

10개씩 보기

회사소개 | 이용약관 | 서비스수준협약 | 개인정보처리방침

© AFI, Inc. All rights reserved.



# 차트 생성 및 등록

- 등록된 차트를 클릭하면 온라인에서 데이터 확인 및 편집 가능 (계속)

LevelChart.xlsx

편집 모드

on

off

삭제

<input type="checkbox"/>	level	maxExperience	rewardGold
<input type="checkbox"/>	1	100	1111
<input type="checkbox"/>	2	200	2222
<input type="checkbox"/>	3	400	3333
<input type="checkbox"/>	4	800	4444
<input type="checkbox"/>	5	1600	5555
<input type="checkbox"/>	6	3200	6666
<input type="checkbox"/>	7	6400	7777
<input type="checkbox"/>	8	12800	8888
<input type="checkbox"/>	9	25600	9999
<input type="checkbox"/>	10	51200	100000

< 1 >

10개씩 보기

확인



# 차트 생성 및 등록

- 차트 파일을 선택하고, "차트 파일 적용" 버튼을 눌러 차트 파일 적용

Backnd Console << ProjectA ⚙️

ProjectA 개발 모드

★ 차트 관리 📄 차트 파일 업로드 **🔄 차트 파일 적용** 📄 CSV 다운로드 🗑️ 삭제

SDK 문서 📄 콘솔 가이드 📄

차트 관리 / 레벨차트

레벨차트

현재 적용 차트 파일

적용된 차트가 없습니다.  
\*id 열(column)은 예약된 키이므로 차트에 존재하지 않아야 합니다.

<input checked="" type="checkbox"/>	번호	파일명	파일 ID	등록일
<input checked="" type="checkbox"/>	1	<a href="#">LevelChart.xlsx</a>	82616 📄	2023.06.13 14:45

< 1 >

10개씩 보기 ▾

회사소개 이용약관 서비스수준협약 개인정보처리방침 © AFI, Inc. All rights reserved.



# 차트 생성 및 등록

- 차트 파일을 선택하고, "차트 파일 적용" 버튼을 눌러 차트 파일 적용 (계속)

The screenshot shows the Backnd Console interface for ProjectA. The left sidebar contains a navigation menu with items like '즐거찾기', '대시보드', '서버 설정', '데이터베이스', '유저 관리', '유저 접근 관리', '게임 정보 관리', '랭킹 관리', '우편 관리', '푸시 관리', '쿠폰 관리', '차트 관리', '확률 관리', '로그 관리', '영수증 검증', '길드 관리', '캐시 관리', and '버전 관리'. The main content area is titled '차트 관리' and shows a table of chart files. A modal dialog is open in the center, asking for confirmation to apply the selected chart file.

Backnd Console << ProjectA

ProjectA 개발 모드

★ 차트 관리 | 차트 파일 업로드 | 차트 파일 적용 | CSV 다운로드 | 삭제

SDK 문서 | 콘솔 가이드

차트 관리 / 레벨차트

레벨차트

현재 적용 차트 파일

적용된 차트가 없습니다.  
\*id 열(column)은 예약된 키이므로 차트에 존재하지 않아야 합니다.

<input checked="" type="checkbox"/>	번호	파일명	파일 ID	등록일
<input checked="" type="checkbox"/>	1	<a href="#">LevelChart.xlsx</a>	88616	2023.06.13 14:45

10개씩 보기

안내

해당 차트 파일을 적용하시겠습니까?

회사소개 | 이용약관 | 서비스수준협약 | 개인정보처리방침

© AFI, Inc. All rights reserved.



# 차트 생성 및 등록

- "레벨차트"에 LevelChart.xlsx 파일이 적용되어 있는 것을 볼 수 있다.

Backnd Console << ProjectA

★ 차트 관리 | 폴더 생성 | 차트 생성 | 폴더에 추가 | 삭제

SDK 문서 | 콘솔 가이드

차트 관리

차트 1/200 | 파일 1/200

<input type="checkbox"/> 번호	차트명	적용된 차트	차트 설명	우편 기능
<input type="checkbox"/> 1	레벨차트	LevelChart.xlsx	레벨에 필요한 경험치와 레벨업 보상에 관련한 차트입니다.	사용안함

< 1 >

10개씩 보기

회사소개 | 이용약관 | 서비스수준협약 | 개인정보처리방침

© AFI, Inc. All rights reserved.

# 차트 데이터 불러오기

- 차트 파일 ID
- 차트 데이터 불러오기





# 차트 데이터 불러오기

## ■ 차트 파일 ID

- 사용할 차트로 이동해 아이디 정보 얻어오기

Backnd Console << ProjectA [Settings] [5] 관리자

ProjectA [개발 모드]

★ 차트 관리 | 폴더 생성 | 차트 생성 | 폴더에 추가 | 삭제

SDK 문서 | 콘솔 가이드

차트 관리

차트 1/200 | 파일 1/200

<input type="checkbox"/>	번호	차트명	적용된 차트	차트 설명	우편 기능
<input type="checkbox"/>	1	<b>레벨차트</b>	LevelChart.xlsx	레벨에 필요한 경험치와 레벨업 보상에 관련한 차트입니다.	사용안함

< 1 > 10개씩 보기

차트 관리 (highlighted in red)



# 차트 데이터 불러오기

- 사용할 차트로 이동해 아이디 정보 얻어오기 (계속)

Backnd Console << ProjectA [Settings]

ProjectA 개발 모드

★ 차트 관리 | 차트 파일 업로드 | 차트 파일 적용 | CSV 다운로드 | 삭제

SDK 문서 | 콘솔 가이드

차트 관리 / 레벨차트

레벨차트

현재 적용 차트 파일: LevelChart.xlsx  
\*id 열(column)은 예약된 키이므로 차트에 존재하지 않아야 합니다.

<input type="checkbox"/> 번호	파일명	파일 ID	등록일
<input type="checkbox"/> 1	<a href="#">LevelChart.xlsx</a> ✓	82616	2023.06.13 14:45

< 1 > 10개씩 보기

회사소개 | 이용약관 | 서비스수준협약 | 개인정보처리방침 © AFI, Inc. All rights reserved.



# 차트 데이터 불러오기

- 레벨 차트 ID를 저장하는 LEVEL\_CHART 변수 선언
  - Constants Script 수정

```
1 public static class Constants
2 {
3     public static readonly string USER_DATA_TABLE = "USER_DATA";
4     public static readonly string DAILY_RANK_UUID = "0fa14490-085a-11ee-b506-7d5eaba96eff";
5     public static readonly int MAX_RANK_LIST = 20;
6
7     public static readonly string LEVEL_CHART = "82616";
8 }
```



# 차트 데이터 불러오기

## ■ 차트 데이터 불러오기

- 콘솔에 등록된 모든 차트 데이터를 불러오는 스크립트 생성 및 작성
  - C# Script 생성 후 스크립트의 이름을 "BackendChartData"로 변경

```
1  using System.Collections.Generic;
2      using UnityEngine;
3      using BackEnd;
4
5  public static class BackendChartData...
66
67  [System.Serializable]
68  public class LevelChartData
69  {
70      public int    level;
71      public int    maxExperience;
72      public int    rewardGold;
73  }
```

뒷장



# 차트 데이터 불러오기

- 콘솔에 등록된 모든 차트 데이터를 불러오는 스크립트 생성 및 작성 (계속)

```
5 public static class BackendChartData
6 {
7     // 레벨별 레벨업 필요 경험치와 보상 정보
8     public static List<LevelChartData> levelChart;
9
10    static BackendChartData()
11    {
12        levelChart = new List<LevelChartData>();
13    }
14
15    public static void LoadAllChart()
16    {
17        LoadLevelChart();
18    }
19 }
```



# 차트 데이터 불러오기

- 콘솔에 등록된 모든 차트 데이터를 불러오는 스크립트 생성 및 작성 (계속)

```
20 public static void LoadLevelChart()
21 {
22     Backend.Chart.GetChartContents(Constants.LEVEL_CHART, callback =>
23     {
24         if ( callback.IsSuccess() )
25         {
26             // JSON 데이터 파싱 성공
27             try
28             {
29                 LitJson.JsonData jsonData = callback.FlattenRows();
30
31                 // 받은 데이터의 개수가 0이면 데이터가 없는 것
32                 if ( jsonData.Count <= 0 )
33                 {
34                     Debug.LogWarning("데이터가 존재하지 않습니다.");
35                 }
36             }
37         }
38     }
39 }
```

== backend method ==

```
Backend.Chart.GetChartContents(string selectedChartFileId);
```

selectedChartFileId는 차트 파일의 uuid 혹은 id로 해당 id 값을 가지는 차트 정보를 불러온다.



# 차트 데이터 불러오기

- 콘솔에 등록된 모든 차트 데이터를 불러오는 스크립트 생성 및 작성 (계속)

```
36     else
37     {
38         for ( int i = 0; i < jsonData.Count; ++ i )
39         {
40             LevelChartData newChart = new LevelChartData();
41             newChart.level         = int.Parse(jsonData[i]["level"].ToString());
42             newChart.maxExperience = int.Parse(jsonData[i]["maxExperience"].ToString());
43             newChart.rewardGold    = int.Parse(jsonData[i]["rewardGold"].ToString());
44
45             levelChart.Add(newChart);
46
47             Debug.Log($"Level : {newChart.level}, Max Exp : {newChart.maxExperience}, " +
48                 $"Reward Gold : {newChart.rewardGold}");
49         }
50     }
51 }
52 // JSON 데이터 파싱 실패
53 catch ( System.Exception e )
54 {
55     // try-catch 에러 출력
56     Debug.LogError(e);
57 }
58 }
59 else
60 {
61     Debug.LogError($"{{Constants.LEVEL_CHART}}의 차트 불러오기에 에러 발생 : {{callback}}");
62 }
63 });
64 }
65 }
```



# 차트 데이터 불러오기

## ■ 로그인할 때 차트 데이터 불러오기

### □ Login Script 수정

```
44  /// <summary> 로그인 시도 후 서버로부터 전달받은 message를 기반으로 로직 처리
47  private void ResponseToLogin(string ID, string PW)
48  {
49      // 서버에 로그인 요청
50      Backend.BMember.CustomLogin(ID, PW, callback =>
51      {
52          StopCoroutine(nameof(LoginProcess));
53
54          // 로그인 성공
55          if ( callback.IsSuccess() )
56          {
57              SetMessage($"{inputFieldID.text}님 환영합니다.");
58
59              // 모든 차트 데이터 불러오기
60              BackendChartData.LoadAllChart();
61
62              // Lobby 씬으로 이동
63              Utils.LoadScene(SceneNames.Lobby);
64          }
65          // 로그인 실패
66          else ...
99      });
100 }
```

Tip. 차트 불러오기는 클라이언트 로그인 이후에 사용할 수 있고, 게임 실행 도중에 데이터가 바뀔 일이 없기 때문에 1회만 불러오도록 로그인에 성공했을 때 1회 차트 데이터를 불러온다.





# 차트 데이터 불러오기

## ■ 계정 생성/로그인할 때 차트 데이터 불러오기

### □ RegisterAccount Script 수정

```
64      /// <summary> 계정 생성 시도 후 서버로부터 전달받은 message를 기반으로 로직 처리
67      private void CustomSignUp()
68      {
69          Backend.BMember.CustomSignUp(inputFieldID.text, inputFieldPW.text, callback =>
70          {
71              // "계정 생성" 버튼 상호작용 활성화
72              btnRegisterAccount.interactable = true;
73
74              // 계정 생성 성공
75              if ( callback.IsSuccess() )
76              {
77                  // E-mail 정보 업데이트
78                  Backend.BMember.UpdateCustomEmail(inputFieldEmail.text, callback =>
79                  {
80                      if ( callback.IsSuccess() )
81                      {
82                          SetMessage($"계정 생성 성공. {inputFieldID.text}님 환영합니다.");
83
84                          // 계정 생성에 성공했을 때 해당 계정의 게임 정보 생성
85                          BackendGameData.Instance.GameDataInsert();
86
87                          // 모든 차트 데이터 불러오기
88                          BackendChartData.LoadAllChart();
89
90                          // Lobby 씬으로 이동
91                          Utils.LoadScene(SceneNames.Lobby);
92                      }
93                  });
94              }
95              // 계정 생성 실패
96              else...
124          });
125      }
```



# 차트 데이터 불러오기

## ■ 결과 화면

Console

Clear Collapse Error Pause Editor

[14:25:56] TheBackend dll v5.9.2  
UnityEngine.Debug:Log (object)

[14:25:56] Server setting success  
UnityEngine.Debug:Log (object)

[14:25:56] 초기화 성공 : statusCode : 204  
message : Success

[14:26:03] Level : 1, Max Exp : 100, Reward Gold : 1111  
UnityEngine.Debug:Log (object)

[14:26:03] Level : 2, Max Exp : 200, Reward Gold : 2222  
UnityEngine.Debug:Log (object)

[14:26:03] Level : 3, Max Exp : 400, Reward Gold : 3333  
UnityEngine.Debug:Log (object)

[14:26:03] Level : 4, Max Exp : 800, Reward Gold : 4444  
UnityEngine.Debug:Log (object)

[14:26:03] Level : 5, Max Exp : 1600, Reward Gold : 5555  
UnityEngine.Debug:Log (object)

[14:26:03] Level : 6, Max Exp : 3200, Reward Gold : 6666  
UnityEngine.Debug:Log (object)

[14:26:03] Level : 7, Max Exp : 6400, Reward Gold : 7777  
UnityEngine.Debug:Log (object)

[14:26:03] Level : 8, Max Exp : 12800, Reward Gold : 8888  
UnityEngine.Debug:Log (object)

[14:26:03] Level : 9, Max Exp : 25600, Reward Gold : 9999  
UnityEngine.Debug:Log (object)

[14:26:03] Level : 10, Max Exp : 51200, Reward Gold : 100000  
UnityEngine.Debug:Log (object)

[14:26:03] 게임 정보 데이터 불러오기에 성공했습니다. : statusCode : 200  
message : Success

# 차트 데이터 사용하기

- 차트 데이터를 이용한 레벨 시스템
- 차트 데이터를 이용한 경험치 출력



# 차트 데이터 사용하기

## ■ 차트 데이터를 이용한 레벨 시스템

### ■ 경험치/레벨을 제어하는 스크립트 생성 및 작성

#### □ C# Script 생성 후 스크립트의 이름을 "LevelSystem"으로 변경

```
1 using UnityEngine;
2
3 public class LevelSystem : MonoBehaviour
4 {
5     private readonly int increaseExperience = 25; // 게임 플레이 당 획득 경험치
6
7     public void Process()
8     {
9         int currentLevel = BackendGameData.Instance.UserGameData.level;
10
11         // 게임을 한번 플레이할 때마다 경험치 획득
12         BackendGameData.Instance.UserGameData.experience += increaseExperience;
13
14         // 현재 경험치가 최대 경험치보다 크거나 같고, 현재 레벨이 최대 레벨보다 작을 때
15         if ( BackendGameData.Instance.UserGameData.experience >= BackendChartData.levelChart[currentLevel-1].maxExperience &&
16             BackendChartData.levelChart.Count > currentLevel )
17         {
18             // 레벨업 보상 지급
19             BackendGameData.Instance.UserGameData.gold += BackendChartData.levelChart[currentLevel-1].rewardGold;
20             // 경험치를 0으로 초기화
21             BackendGameData.Instance.UserGameData.experience = 0;
22             // 레벨 1 증가
23             BackendGameData.Instance.UserGameData.level ++;
24         }
25
26         // 게임 정보 업데이트
27         BackendGameData.Instance.GameDataUpdate();
28
29         Debug.Log($"현재 레벨 : {BackendGameData.Instance.UserGameData.level}," +
30             $"경험치 : {BackendGameData.Instance.UserGameData.experience}/" +
31             $"{BackendChartData.levelChart[currentLevel-1].maxExperience}");
32     }
33 }
```



# 차트 데이터 사용하기

## ■ 임시로 경험치/레벨을 관리하던 코드 삭제

### □ GameController Script 수정

```
20 public void GameOver()  
21 {  
22     // 중복 처리 되지 않도록 bool 변수로 제어  
23     if ( IsGameOver == true ) return;  
24  
25     IsGameOver = true;  
26  
27     // 게임오버 되었을 때 호출할 메소드들을 실행  
28     onGameOver.Invoke();  
29  
30     // 현재 점수 정보를 바탕으로 랭킹 데이터 갱신  
31     dailyRank.Process(score);  
32  
33     /*// 경험치 증가 및 레벨업 여부 검사  
34     // (현재 레벨 시스템에 대한 설정이 없기 때문에 경험치의 최대치를 100으로 가정)  
35     // (게임을 한번 플레이할 때마다 경험치는 25씩 증가)  
36     BackendGameData.Instance.UserGameData.experience += 25;  
37     if ( BackendGameData.Instance.UserGameData.experience >= 100 )  
38     {  
39         BackendGameData.Instance.UserGameData.experience = 0;  
40         BackendGameData.Instance.UserGameData.level ++;  
41     }  
42  
43     // 게임 정보 업데이트  
44     BackendGameData.Instance.GameDataUpdate();*/  
45 }  
46 }
```



# 차트 데이터 사용하기

- GameController 오브젝트에 컴포넌트 추가 및 설정

The screenshot displays the Unity Hierarchy and Inspector panels. In the Hierarchy panel, the 'GameController' object is selected and highlighted with a red box. A red arrow points from this box to the Inspector panel. The Inspector panel shows the 'GameController (Script)' component selected. The 'On Game Over ()' event list is visible, with the following entries:

- Runtime Only | GameObject.SetActive | Meteorite
- Runtime Only | GameObject.SetActive | EnemySpawner
- Runtime Only | GameObject.SetActive | Player
- Runtime Only | GameController.OnGameOver | GameController
- Runtime Only | LevelSystem.Process | GameController

The 'LevelSystem.Process' entry is highlighted with a red dashed box. Below the event list, there are '+' and '-' buttons. At the bottom of the Inspector, the 'Daily Rank' component is also visible, with the 'GameController (Daily Rank Register)' script assigned to it. The 'Level System (Script)' component is also visible at the bottom of the Inspector panel.



# 차트 데이터 사용하기

## ■ 차트 데이터를 이용한 경험치 출력

### ■ 경험치 Slider의 최대 경험치 수치 설정

#### □ TopPanelViewer Script 수정

```
20 private void Awake()...
24
25 public void UpdateNickname()...
31
32 public void UpdateGameData()
33 {
34     int currentLevel = BackendGameData.Instance.UserGameData.level;
35
36     textLevel.text = currentLevel.ToString();
37     sliderExperience.value = BackendGameData.Instance.UserGameData.experience /
38     BackendChartData.levelChart[currentLevel-1].maxExperience;
39     textHeart.text = $"{BackendGameData.Instance.UserGameData.heart} / 30";
40     textJewel.text = $"{BackendGameData.Instance.UserGameData.jewel}";
41     textGold.text = $"{BackendGameData.Instance.UserGameData.gold}";
42 }
43 }
```



# 차트 데이터 사용하기

## ■ 결과 화면

Console

Clear Collapse Error Pause Editor

UnityEngine.Debug:Log (object)

[15:23:37] 게임 정보 데이터 불러오기에 성공했습니다. : statusCode : 200  
message : Success

[15:23:43] 2023-06-11T13:12:57.994Z의 게임 정보 데이터 수정을 요청합니다.  
UnityEngine.Debug:Log (object)

[15:23:43] 현재 레벨 : 3, 경험치 : 25/400  
UnityEngine.Debug:Log (object)

[15:23:43] 게임 정보 데이터 수정에 성공했습니다. : statusCode : 204  
message : Success

[15:23:43] 새로운 랭킹 데이터 생성 및 등록 : statusCode : 404  
errorCode : NotFoundException

현재 레벨 : 3, 경험치 : 25/400  
UnityEngine.Debug:Log (object)  
LevelSystem:Process () (at [Assets/Scripts/#100Backend/LevelSystem.cs:28](#))  
UnityEngine.Events.UnityEvent:Invoke ()  
GameController:GameOver () (at [Assets/Scripts/#04Game/GameController.cs:28](#))  
Enemy:OnTriggerEnter2D (UnityEngine.Collider2D) (at [Assets/Scripts/#04Game/Enemy.cs:31](#))

**게임오버 되었을 때  
경험치 증가 및 레벨업 여부 검사, 레벨업 시 재화 제공**





# 차트 데이터 사용하기

## ■ 결과 화면

Backend Console | ProjectA | 게임 정보 관리 | 테이블 데이터

aedaa3e0-0859-11ee-8b11-73576fc879f3 수정

컬럼 명	데이터 타입	값
experience	number	25

컬럼 명	데이터 타입	값
level	number	3

컬럼 명	데이터 타입	값
jewel	number	0

컬럼 명	데이터 타입	값
heart	number	30

client\_date | dailyB... | expe... | gold | heart

2023-06-11T13:13:16.095Z	0	25	0	30
2023-06-11T13:12:58.024Z	200	25	0	30

회사소개 | 이용약관 | 서비스수준협약 | 개인정보처리방침 | © AFI, Inc. All rights reserved.



# 차트 데이터 사용하기

## ■ 결과 화면

