

Unity Backend

디버그

Created in 2023-10-22
Last Updated 2023-10-22
Unity Version 2022.2.2f1

Index

- ◆ 유저 정보 생성 전에 불러오는 문제
- ◆ 랭킹 등록/불러오기 오류

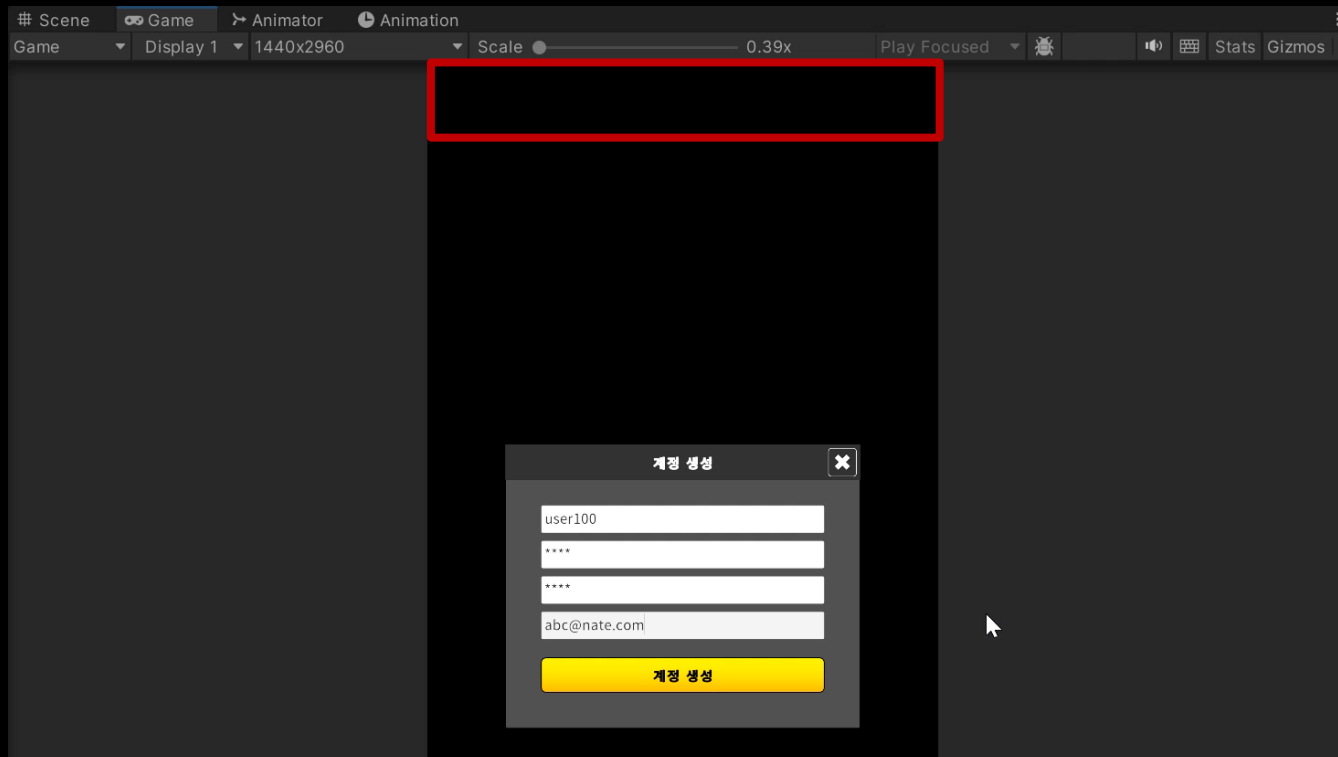
유저 정보 생성 전에 불러오는 문제

- 유저 정보 생성(Insert) 전에 불러오는(Load) 문제



유저 정보 생성 전에 불러오는 문제

■ 유저 정보 생성(Insert) 전에 불러오는(Load) 문제



현재는 Login 씬에서 계정을 생성하면 유저 정보를 생성함(비동기)과 동시에 Lobby 씬으로 이동하고, Lobby 씬에서 유저 정보를 불러와 화면에 출력함
이 과정에서 서버 통신 문제로 유저 정보 생성이 완료되지 않았다면 오류 발생



유저 정보 생성 전에 불러오는 문제

- 계정을 생성할 때 유저 정보를 생성하지 않도록 삭제
 - RegisterAccount Script 수정

```
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     });  
96 }
```



유저 정보 생성 전에 불러오는 문제

- 유저 정보 데이터를 불러올 때 데이터가 없으면 데이터 생성

- BackendGameData Script 수정

```
33 public void GameDataInsert()  
34 {  
35     // 유저 정보를 초기값으로 설정  
36     userGameData.Reset();  
37  
38     // 테이블에 추가할 데이터로 가공  
39     Param param = new Param(...);  
48  
49     // 첫 번째 매개변수는 뒤끝 콘솔의 "게임 정보 관리" 탭에 생성한 테이블 이름  
50     Backend.GameData.Insert("USER_DATA", param, callback =>  
51     {  
52         // 게임 정보 추가에 성공했을 때  
53         if ( callback.IsSuccess() )  
54         {  
55             // 게임 정보의 고유값  
56             gameDataRowInDate = callback.GetInDate();  
57  
58             Debug.Log($"게임 정보 데이터 삽입에 성공했습니다. : {callback}");  
59  
60             onGameDataLoadEvent?.Invoke();  
61         }  
62         // 실패했을 때  
63         else...  
67     });  
68 }  
69
```

GameDataInsert()를 Lobby 씬에서 호출하기 때문에 새로운 데이터를 생성했을 때 Lobby 씬 UI에 정보를 출력하는 이벤트 메소드 호출



유저 정보 생성 전에 불러오는 문제

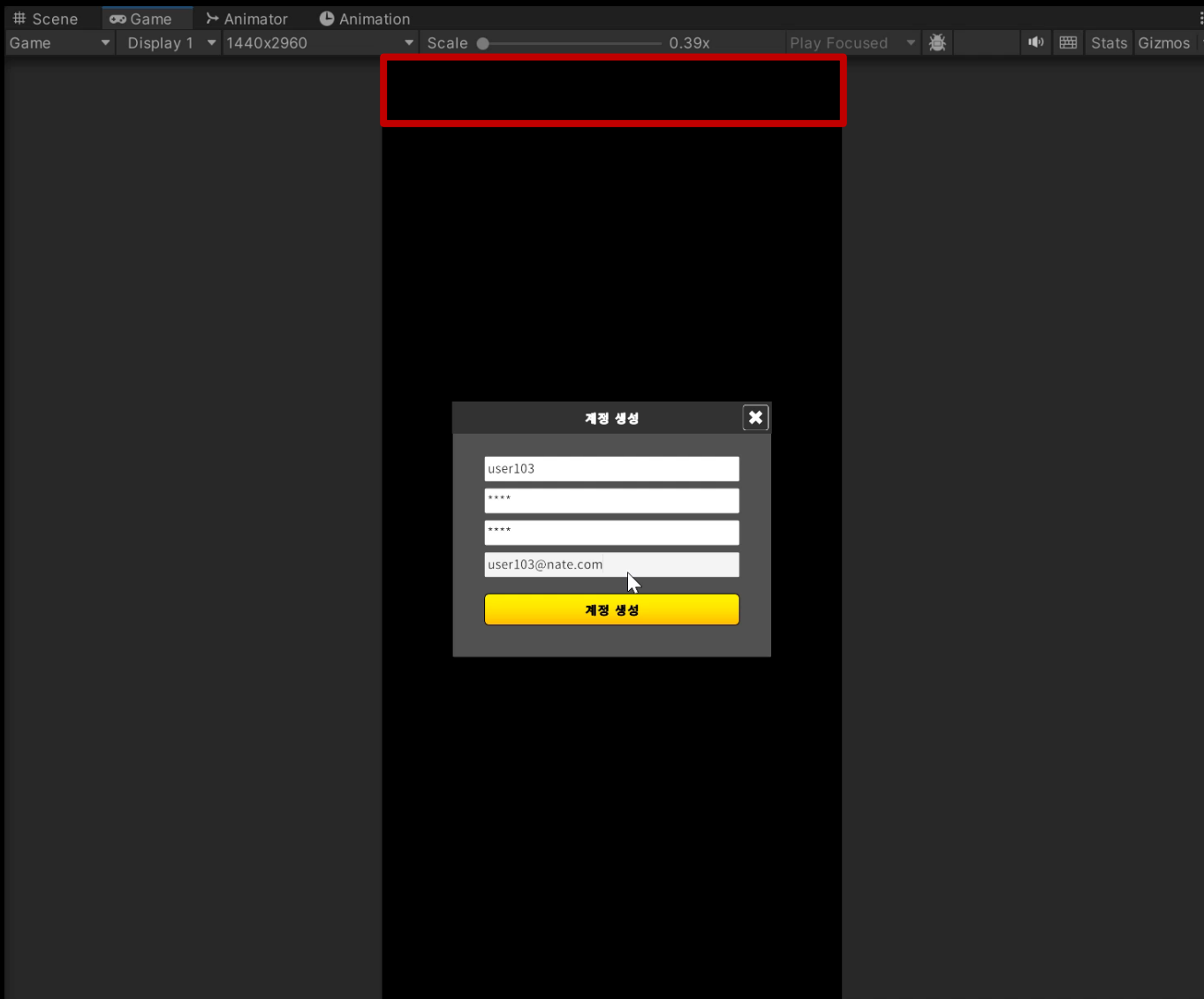
BackendGameData Script 수정 (계속)

```
73 public void GameDataLoad()
74 {
75     Backend.GameData.GetMyData("USER_DATA", new Where(), callback =>
76     {
77         // 게임 정보 불러오기에 성공했을 때
78         if ( callback.IsSuccess() )
79         {
80             Debug.Log($"게임 정보 데이터 불러오기에 성공했습니다. : {callback}");
81
82             // JSON 데이터 파싱 성공
83             try
84             {
85                 LitJson.JsonData gameDataJson = callback.FlattenRows();
86
87                 // 받아온 데이터의 개수가 0이면 데이터가 없는 것
88                 if ( gameDataJson.Count <= 0 )
89                 {
90                     Debug.LogWarning("데이터가 존재하지 않습니다.");
91
92                     // 유저 정보가 없으면 유저 정보 생성
93                     GameDataInsert();
94                 }
95                 else ...
109             }
110             // JSON 데이터 파싱 실패
111             catch ( System.Exception e )
```



유저 정보 생성 전에 불러오는 문제

■ 결과 화면



랭킹 등록/불러오기 오류

- 랭킹 데이터 저장을 위한 새 테이블 생성
- 랭킹 데이터 테이블 컬럼 추가
- 일일 최고 점수 랭킹



랭킹 등록/불러오기 오류

- 랭킹 데이터 저장을 위한 새 테이블 생성
 - Backend Console의 "게임 정보" 탭에 새 테이블 생성

Backend Console interface showing the '게임 정보' (Game Info) tab. The '테이블 생성' (Create Table) button is highlighted in red. The table list shows the following details:

테이블명	설명	분류	스키마 정의	상태
<input type="checkbox"/> USER_DATA		public	스키마 미정의	활성

Navigation: < 1 > 10개씩 보기



랭킹 등록/불러오기 오류

- 최고 점수를 저장하는 USER_RANK_DATA 테이블 생성

Backnd ProjectA

테이블 생성

테이블 생성

테이블명

USER_DATA

*테이블은 최대 100개까지 생성할 수 있습니다.

테이블 생성

분류 Private Public

테이블명

테이블 설명

스키마 정의 사용 미사용

상태 활성 비활성

확인 취소

Tip. 랭킹에 사용하는 테이블은 "Private"로 생성해야 함





랭킹 등록/불러오기 오류

- 랭킹 테이블 이름 설정
 - Constants Script 수정

```
1 public static class Constants
2 {
3     public static readonly string USER_DATA_TABLE = "USER_DATA";
4     public static readonly string USER_RANK_DATA_TABLE = "USER_RANK_DATA";
5     public static readonly string DAILY_RANK_UUID = "0+a14490-085a-11ee-b506-7d5eaba96eff";
6     public static readonly int MAX_RANK_LIST = 20;
7
8     public static readonly string LEVEL_CHART = "82683";
9
10    public static readonly string GOODS_CHART_NAME = "재화차트";
11
12    public static readonly int EXPIRATION_DAYS = 3;
13
14    public static readonly string RANDOM_GUILD_UUID = "6a675fc0-3084-11ee-99ec-b1646f9340c5";
15 }
```



랭킹 등록/불러오기 오류

■ 랭킹 데이터 테이블 컬럼 추가

- “USER_RANK_DATA” 테이블의 컬럼 데이터가 없으면 추가
 - BackendGameData Script 수정

```
33 public void GameDataInsert()
34 {
35     // 유저 정보를 초기값으로 설정
36     userGameData.Reset();
37
38     // 테이블에 추가할 데이터로 가공
39     Param param = new Param()
40     {
41         { "level",      userGameData.level },
42         { "experience", userGameData.experience },
43         { "gold",       userGameData.gold },
44         { "jewel",      userGameData.jewel },
45         { "heart",      userGameData.heart },
46         //{ "dailyBestScore", userGameData.dailyBestScore }
47     };
48
```

랭킹 점수 관리를 위한 별도의 테이블이 있기 때문에
“USER_DATA” 테이블의 dailyBestScore 컬럼 코드 삭제



랭킹 등록/불러오기 오류

BackendGameData Script 수정 (계속)

```
49  /**/ 첫 번째 매개변수는 뒤끝 콘솔의 "게임 정보 관리" 탭에 생성한 테이블 이름  
50  Backend.GameData.Insert("USER_DATA", param, callback =>  
51  {  
52      // 게임 정보 추가에 성공했을 때  
53      if ( callback.IsSuccess() )  
54      {  
55          // 게임 정보의 고유값  
56          gameDataRowInDate = callback.GetInDate();  
57  
58          Debug "USER_DATA" 테이블과 "USER_RANK_DATA" 테이블에 별도로 {callback}");  
59          Insert()를 하기 위해 GameDataInsert(string, Param) 메소드를 정의함  
60          onGameDataLoadEvent?.Invoke();  
61      }  
62      // 실패했을 때  
63      else  
64      {  
65          Debug.LogError($"게임 정보 데이터 삽입에 실패했습니다. : {callback}");  
66      }  
67  });*/  
68  
69  GameDataInsert(Constants.USER_DATA_TABLE, param);  
70  
71
```



랭킹 등록/불러오기 오류

BackendGameData Script 수정 (계속)

```
72 private void GameRankDataInsert()  
73 {  
74     userGameData.dailyBestScore = 0;  
75  
76     Param rankParam = new Param()  
77     {  
78         { "dailyBestScore", userGameData.dailyBestScore }  
79     };  
80  
81     GameDataInsert(Constants.USER_RANK_DATA_TABLE, rankParam);  
82 }  
83  
84 private void GameDataInsert(string tableName, Param param)  
85 {  
86     Backend.GameData.Insert(tableName, param, callback =>  
87     {  
88         // 게임 정보 추가에 성공했을 때  
89         if ( callback.IsSuccess() )  
90         {  
91             // 게임 정보의 고유값  
92             gameDataRowInDate = callback.GetInDate();  
93  
94             Debug.Log($"게임 정보 데이터 삽입에 성공했습니다. : {callback}");  
95  
96             onGameDataLoadEvent?.Invoke();  
97         }  
98         // 실패했을 때  
99         else  
100        {  
101            Debug.LogError($"게임 정보 데이터 삽입에 실패했습니다. : {callback}");  
102        }  
103    });  
104 }  
105
```




랭킹 등록/불러오기 오류

BackendGameData Script 수정 (계속)

앞에서 제작할 때 "USER_DATA"와 같이 문자열을 그대로 작성해서 수정

```
109 public void GameDataLoad()
110 {
111     Backend.GameData.GetMyData(Constants.USER_DATA_TABLE, new Where(), callback =>
112     {
113         // 게임 정보 불러오기에 성공했을 때
114         if ( callback.IsSuccess() )
115         {
116             Debug.Log($"게임 정보 데이터 불러오기에 성공했습니다. : {callback}");
117
118             // JSON 데이터 파싱 성공
119             try
120             {
121                 LitJson.JsonData gameDataJson = callback.FlattenRows();
122
123                 // 받아온 데이터의 개수가 0이면 데이터가 없는 것
124                 if ( gameDataJson.Count <= 0 )...
125             }
126             else
127             {
128                 // 불러온 게임 정보의 고유값
129                 gameDataRowInDate = gameDataJson[0]["inDate"].ToString();
130                 // 불러온 게임 정보를 userData 변수에 저장
131                 userGameData.level = int.Parse(gameDataJson[0]["level"].ToString());
132                 userGameData.experience = float.Parse(gameDataJson[0]["experience"].ToString());
133                 userGameData.gold = int.Parse(gameDataJson[0]["gold"].ToString());
134                 userGameData.jewel = int.Parse(gameDataJson[0]["jewel"].ToString());
135                 userGameData.heart = int.Parse(gameDataJson[0]["heart"].ToString());
136                 //userGameData.dailyBestScore = int.Parse(gameDataJson[0]["dailyBestScore"].ToString());
137
138                 onGameDataLoadEvent?.Invoke();
139             }
140         }
141         // JSON 데이터 파싱 실패
142         catch ( System.Exception e )
143         {
144             // 유저 정보를 초기값으로 설정
145             userGameData.Reset();
146             // try-catch 에러 출력
147             Debug.LogError(e);
148         }
149     }
150     // 실패했을 때
151     else...
152 });
153
154
155
156
157
158
159
160
161
```

랭킹 점수 관리를 위한 별도의 테이블이 있기 때문에 "USER_DATA" 테이블의 dailyBestScore 컬럼 코드 삭제



랭킹 등록/불러오기 오류

BackendGameData Script 수정 (계속)

```
162 // "USER_RANK_DATA" 테이블에 랭킹 데이터(dailyBestScore) 컬럼이 없는 유저의 경우 새로 생성
163 Backend.GameData.GetMyData(Constants.USER_RANK_DATA_TABLE, new Where(), callback =>
164 {
165     // 게임 정보 불러오기에 성공했을 때
166     if ( callback.IsSuccess() )
167     {
168         Debug.Log($"유저 랭킹 데이터 불러오기에 성공했습니다. : {callback}");
169
170         // JSON 데이터 파싱 성공
171         try
172         {
173             LitJson.JsonData gameDataJson = callback.FlattenRows();
174
175             if ( gameDataJson.Count <= 0 )
176             {
177                 Debug.LogWarning("랭킹 데이터가 존재하지 않습니다.");
178
179                 GameRankDataInsert();
180             }
181         }
182         // JSON 데이터 파싱 실패
183         catch ( System.Exception e )
184         {
185             // try-catch 에러 출력
186             Debug.LogError(e);
187         }
188     }
189     // 실패했을 때
190     else
191     {
192         Debug.LogError($"유저 랭킹 데이터 불러오기에 실패했습니다. : {callback}");
193     }
194 });
195 }
196
```

"USER_RANK_DATA" 테이블에 dailyBestScore 컬럼이 있는지 검사하고 없으면 생성



랭킹 등록/불러오기 오류

BackendGameData Script 수정 (계속)

```
200 public void GameDataUpdate(UnityAction action=null)
201 {
202     if ( userGameData == null )...
208
209     Param param = new Param()
210     {
211         { "level",          userGameData.level },
212         { "experience",     userGameData.experience },
213         { "gold",          userGameData.gold },
214         { "jewel",         userGameData.jewel },
215         { "heart",         userGameData.heart },
216         //{ "dailyBestScore": userGameData.dailyBestScore }
217     };
218
219     // 게임 정보의 고유값(gameDataRowInDate)이 없으면 에러 메시지 출력
220     if ( string.IsNullOrEmpty(gameDataRowInDate) )...
224     // 게임 정보의 고유값이 있으면 테이블에 저장되어 있는 값 중 inDate 컬럼의 값과
225     // 소유하는 유저의 owner_inDate가 일치하는 row를 검색하여 수정하는 UpdateV2() 호출
226     else
227     {
228         Debug.Log($"{gameDataRowInDate}의 게임 정보 데이터 수정을 요청합니다.");
229
230         Backend.GameData.UpdateV2( Constants.USER_DATA_TABLE, gameDataRowInDate, Backend.UserInDate, param, callback =>...);
231     }
232 }
244
245
246 }
```

랭킹 점수 관리를 위한 별도의 테이블이 있기 때문에
"USER_DATA" 테이블의 dailyBestScore 컬럼 코드 삭제

앞에서 제작할 때 "USER_DATA"와 같이
문자열을 그대로 작성해서 수정



랭킹 등록/불러오기 오류

■ 일일 최고 점수 랭킹

- “USER_RANK_DATA” 테이블에 컬럼이 1개 이상 존재하도록 설정

Backnd ProjectA

ProjectA 개발 모드

즐겨찾기 홈 프로젝트 설정 개발

유저 **게임 정보** 차트 확률 평션 랜덤 조회 뒤끝챗 구버전 뒤끝매치 버전 관리 운영

유저 접근 길드

★ 게임 정보 + 행 생성 삭제 테이블 초기화

테이블 데이터

테이블 선택 USER_RANK_DATA

상세 검색

검색 검색 조건 초기화

	gamer_id	inDate	owner_inDate	updatedAt	client_date	dailyBestScore
<input type="checkbox"/>	c0dd08c0-7169-11ee-be77-85d82f555f22	2023-10-23T06:02:32.469Z	2023-10-23T06:02:31.628Z	2023-10-23T06:02:32.469Z	2023-10-23T06:02:31.572Z	0

< 1 > 10개씩 보기

기존 계정으로 로그인하거나 새로운 계정을 생성해
“USER_RANK_DATA” 테이블에 1개 이상의 컬럼이 존재하도록 설정



랭킹 등록/불러오기 오류

■ 일일 최고 점수 랭킹 재생성

Backnd ProjectA

ProjectA 개발 모드

즐거찾기

홈

프로젝트 설정

개발

운영

유저 접근

길드

랭킹

우편

푸시 알림

공지

이벤트

쿠폰

1:1 문의

유저정보 찾기

약관 및 정책

결제

랭킹

랭킹 생성 삭제

신규 랭킹(SDK 5.4.0 이상) 전체 랭킹명으로 검색

랭킹 보상

<input checked="" type="checkbox"/>	유형	랭킹명	초기화 기간	랭킹 보상	UUID
<input checked="" type="checkbox"/>	유저	DAILY_RANK	1일	일일최고점수	UUID

< 1 >

10개씩 보기

1. 시리즈를 순서대로 따라했다면 "USER_DATA" 테이블을
일일 랭킹에 사용하고 있기 때문에 기존의 "DAILY_RANK"를 삭제
2. "랭킹 생성" 버튼을 눌러 새로운 랭킹 생성

?



랭킹 등록/불러오기 오류

■ 일일 최고 점수 랭킹 재생성 (계속)

Backnd ProjectA

랭킹 보상

랭킹 생성

*새로운 랭킹을 사용하기 위해서는 SDK 5.4.0 이상이 필요합니다.

유형 유저 랭킹 길드 랭킹

랭킹명* DAILY_RANK

초기화 기간 일간 주간 월간 누적랭킹 일회성 랭킹

랭킹 종료 시 컬럼 초기화* 적용 미적용

랭킹 항목*
테이블 USER_RANK_DATA
컬럼 dailyBestScore

추가 항목 없음

정렬 기준 오름차순 내림차순

랭킹 보상 일일최고점수
보상 우편 제목 일일 최고 점수 랭킹 보상입니다

확인 취소

코드 수정을 최소화하기 위해 랭킹명은 기존과 동일하게 "DAILY_RANK"로 사용



랭킹 등록/불러오기 오류

■ 일일 최고 점수 랭킹 재생성 (계속)

Backnd ProjectA

신규 랭킹(SDK 5.4.0 이상) 전체 랭킹명으로 검색

랭킹 보상

유형	랭킹명	초기화 기간	랭킹 보상	UUID
<input type="checkbox"/>	유저	DAILY_RANK	1일	일일최고점수

UUID

랭킹을 새로 생성해서 변경된 UUID를 다시 등록

10개씩 보기



랭킹 등록/불러오기 오류

- DAILY_RANK_UUID 파라미터 설정
 - Constants Script 수정

```
1 public static class Constants
2 {
3     public static readonly string USER_DATA_TABLE = "USER_DATA";
4     public static readonly string USER_RANK_DATA_TABLE = "USER_RANK_DATA";
5     public static readonly string DAILY_RANK_UUID = "e9111660-7169-11ee-94db-fd87268fe99e";
6     public static readonly int MAX_RANK_LIST = 20;
7
8     public static readonly string LEVEL_CHART = "82683";
9
10    public static readonly string GOODS_CHART_NAME = "재화차트";
11
12    public static readonly int EXPIRATION_DAYS = 3;
13
14    public static readonly string RANDOM_GUILD_UUID = "6a675fc0-3084-11ee-99ec-b1646f9340c5";
15 }
```




랭킹 등록/불러오기 오류

- 랭킹 등록에 사용하는 테이블 변경
 - DailyRankRegister Script 수정

```
1  +using ...
3
4  public class DailyRankRegister : MonoBehaviour
5  {
6  +  public void Process(int newScore)...
10
11 -  private void UpdateMyRankData(int newScore)
12  {
13      string rowInDate = string.Empty;
14
15      // 랭킹 데이터를 업데이트하려면 게임 데이터에서 사용하는 데이터의 inDate 값이 필요
16      Backend.GameData.GetMyData(Constants.USER_RANK_DATA_TABLE, new Where(), callback =>
17  {
18  +  if ( !callback.IsSuccess() )...
23
24      Debug.Log($"데이터 조회에 성공했습니다 : {callback}");
25
26  +  if ( callback.FlattenRows().Count > 0 )...
30  +  else...
35
36  +  Param param = new Param()...;
40
41  +  Backend.URank.User.UpdateUserScore(Constants.DAILY_RANK_UUID, Constants.USER_RANK_DATA_TABLE, rowInDate, param, callback =>...);
52  });
53  }
54
55  +  private void UpdateMyBestRankData(int newScore)...
108 }
```




랭킹 등록/불러오기 오류

■ “USER_DATA” 컬럼 데이터 불일치 오류

```
Console
Clear Collapse Error Pause Editor
message : Success
[15:53:48] 2023-10-23T06:53:46.626Z의 게임 정보 데이터 수정을 요청합니다.
UnityEngine.Debug:Log (object)
[15:53:48] 현재 레벨 : 1,경험치 : 25/100
UnityEngine.Debug:Log (object)
[15:53:48] 게임 정보 데이터 수정에 실패했습니다. : statusCode : 404
errorCode : NotFoundException
[15:53:48] 새로운 랭킹 데이터 생성 및 등록 : statusCode : 404
errorCode : NotFoundException
[15:53:48] 데이터 조회에 성공했습니다 : statusCode : 200
message : Success
게임 정보 데이터 수정에 실패했습니다. : statusCode : 404
errorCode : NotFoundException
message : gameInfo not found, gameInfo을(를) 찾을 수 없습니다
UnityEngine.Debug:LogError (object)
BackendGameData/<>c__DisplayClass13_0:<GameDataUpdate>b_0 (BackEnd.BackendReturnObject) (at Assets/Scripts/#100Backend/BackendGameData.cs:241)
```

기존 유저의 경우 “USER_DATA” 테이블에 dailyBestScore 컬럼 정보가 남아있기 때문에 코드 수정 후 첫 게임 한판의 경우 BackendGameData.GameDataUpdate() 메소드를 호출했을 때 Param 정보가 일치하지 않아 오류가 발생하고 무효 처리가 된다.

방법 1. “USER_DATA” 테이블의 모든 컬럼을 삭제해 초기화한다.

방법 2. 코드 수정 후 1회에 한해서 게임을 실행했을 때 GameDataUpdate()를 따로 호출한다.



랭킹 등록/불러오기 오류

■ 결과 화면

