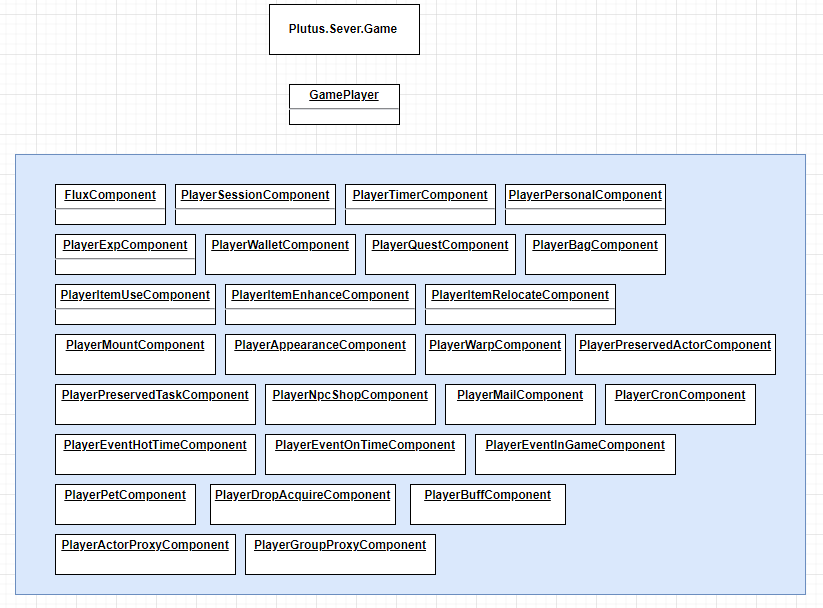
# 

Entity

1. Player Entity
2. Zone Entity
3. PlayerAcotr Entity
4. Session Entity

# 

# Plutus.Server.Game 관련



## .\src\Plutus.Server.Game\Worlds\Players\GamePlayer.cs

* 각각의 컴퍼넌트들을 생성한다.

| public class GamePlayer : XComponent { ........  public static GamePlayer Of(XEntity entity, PlayerRow row, TAppearance appearance = null)  {  ........................  var player = entity.AddComponent<GamePlayer>();  player.Flux = entity.AddComponent<FluxComponent>();  player.Session = entity.AddComponent<PlayerSessionComponent>();  player.Timer = entity.AddComponent<PlayerTimerComponent>();  player.Psn = entity.AddComponent<PlayerPersonalComponent>();  player.Exp = entity.AddComponent<PlayerExpComponent>();  player.Wallet = entity.AddComponent<PlayerWalletComponent>();  player.Quest = entity.AddComponent<PlayerQuestComponent>();  player.Bag = entity.AddComponent<PlayerBagComponent>();   player.ItemUse = entity.AddComponent<PlayerItemUseComponent>();  player.ItemEnhance = entity.AddComponent<PlayerItemEnhanceComponent>();  player.ItemRelocate = entity.AddComponent<PlayerItemRelocateComponent>();   player.Mount = entity.AddComponent<PlayerMountComponent>();  player.Appearance = entity.AddComponent<PlayerAppearanceComponent>();  player.Warp = entity.AddComponent<PlayerWarpComponent>();  player.PreservedActor = entity.AddComponent<PlayerPreservedActorComponent>();  player.PreservedTask = entity.AddComponent<PlayerPreservedTaskComponent>();  player.NpcShop = entity.AddComponent<PlayerNpcShopComponent>();  player.Mail = entity.AddComponent<PlayerMailComponent>();  player.Cron = entity.AddComponent<PlayerCronComponent>();   player.HotTime = entity.AddComponent<PlayerEventHotTimeComponent>();  player.OnTime = entity.AddComponent<PlayerEventOnTimeComponent>();  player.EventInGame = entity.AddComponent<PlayerEventInGameComponent>();   // player.Last = entity.AddComponent<PlayerLastComponent>();  player.Pet = entity.AddComponent<PlayerPetComponent>();  player.DropAcquire = entity.AddComponent<PlayerDropAcquireComponent>();  player.Buff = entity.AddComponent<PlayerBuffComponent>();   // 제일 나중에 들어가야 할 것  player.ActorProxy = entity.AddComponent<PlayerActorProxyComponent>();  player.GroupProxy = entity.AddComponent<PlayerGroupProxyComponent>();  ........................   } } |
| --- |

# UnitTests 관련 프로젝트 - Plutus.Server.All.Tests

## UnitTestWorld.cs

.\test\Plutus.Server.All.Tests\UnitTestWorld.cs

| public class UnitTestWorld : XWorld {  private static readonly ILoggerEx Logger = LoggerFactoryEx.CreateLogger<UnitTestWorld>();    public UnitTestWorld() : base(nameof(UnitTestWorld))  {  // 기본 서비스 생성  Res = AddSystem<ResourceSystem>();  Executor = AddSystem<ExecutorSystem>();  Timer = AddSystem<WheelTimerSystem>();  Maintenance = AddSystem<MaintenanceSystem>();  Db = AddSystem<DatabaseSystem>();  Dashboard = AddSystem<ServerDashboardSystem>();  Redis = AddSystem<RedisSystem>();  Unique = AddSystem<ServerUniqueSystem>();  Word = AddSystem<WordSystem>();  Alive = AddSystem<ServerAliveSystem>();  Relay = AddSystem<WorldRelaySystem>();   // 웹 관련 서비스들  Shard = AddSystem<ShardSystem>();  WebLobby = AddSystem<WebLobbySystem>();  WebPlayer = AddSystem<WebPlayerSystem>();  WebCoupon = AddSystem<WebCouponSystem>();  ServerQueue = AddSystem<ServerQueueSystem>();   // 게임 관련 서비스들  Account = AddSystem<AccountSystem>();  GamePlayer = AddSystem<GamePlayerSystem>();   Azit = AddSystem<AzitSystem>();  Lobby = AddSystem<GameLobbySystem>();  PromissoryNote = AddSystem<PromissoryNoteSystem>();  GameMail = AddSystem<GameMailSystem>();  GameActor = AddSystem<GameActorSystem>();  GameZone = AddSystem<GameZoneSystem>();  Boss = AddSystem<GameBossSystem>();  DoveGroupProxy = AddSystem<DoveGroupProxySystem>();   Cron = AddSystem<GameCronSystem>();  HotTime = AddSystem<EventHotTimeSystem>();  OnTime = AddSystem<EventOnTimeSystem>();  EventInGame = AddSystem<EventInGameSystem>();  Recast = AddSystem<GameRecastSystem>();   // 커뮤니티 관련 서비스들  DovePlayer = AddSystem<DovePlayerSystem>();  DoveSessions = AddSystem<DoveSessionSystem>();  DoveZone = AddSystem<DoveZoneSystem>();   DoveGroupMember = AddSystem<DoveGroupMemberSystem>();  DoveGroup = AddSystem<DoveGroupSystem>();  Channel = AddSystem<DoveChannelSystem>();   ZoneFront = AddSystem<ZoneFrontSystem>();   //signalr hub 관련  HubChatChannel = AddSystem<HubChatChannelSystem>();   // 유닛 테스트 서비스들  DummyResource = AddSystem<DummyResourceSystem>();  DummyWorld = AddSystem<DummyWorldSystem>();  DummyGamePlayer = AddSystem<DummyGamePlayerSystem>();  DummyActor = AddSystem<DummyActorSystem>();  DummyDovePlayer = AddSystem<DummyDovePlayerSystem>();  DummyTopic = AddSystem<DummyTopicSystem>();  }  } |
| --- |

### DummyWorldSystem.cs

* + .\test\Plutus.Server.All.Tests\Systems\DummyWorldSystem.cs

| public class DummyWorldSystem : XSystem {  public MockWorld NewEmptyWorld()  {  var world = new MockWorld();  world.Initialize();  return world;  } } |
| --- |

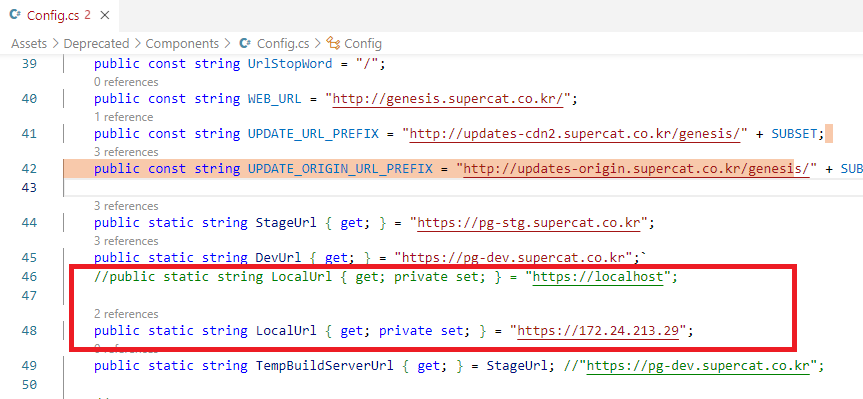
# 기획데이터 서버 리소스에 복사 - Plutus.PublishToClient

* 아래 배치 파일 실행하면 진행
  + .\projectg\_server\tools\Plutus.PublishToClient\publish\_to\_client.bat

| CD /d %~dp0  dotnet run --project ./ --argument D:/00-git/projectd/projectd/Client2/Assets/Plugins/Plutus.Unity |
| --- |

# 서버 로그인 서버 리스트 변경

* 클라이언트 소스 수정
  + .\Client2\Assets\Deprecated\Components\Config.cs





## 사전 필요 사항

* + 172.24.21.29 서버에 5080 포트 Proxy Service 필요

# 서버 리소스 오류

* 서버 리소스 오류시, client를 최신으로 받고 publish\_to\_client.bat 실행

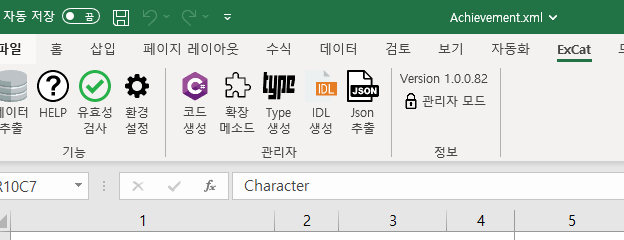
# 캐릭터 초기화 처리

| *// 준비*  public PlayerActor Prepare()  {  if (null != \_actor)  {  *// \_actor 가 null 이 아니면 재접속*  Logger.Info(  $"exist actor maybe reconnected - accountUid({\_player.Psn.AccountId}) playerUid({\_player.Psn.Uid})");  return \_actor;  *// // TODO : 이제 이 곳으로 들어오지 않는 것 같지만*  *// // Logging 해 놓고 체크.*  *// Logger.Error($"already exist actor - accountId:{\_player.Psn.AccountId}, playerId:{\_player.Psn.Uid}");*  *// return \_actor;*  }  int job = \_player.Psn.Job;  int lv = \_player.Exp.GetLevel();  var rSys = GetSystem<ResourceSystem>();  var actorSys = GetSystem<GameActorSystem>();  var option = new PlayerActorOption();  var actorDataId = ActorRules.JobTypeToActorDataId(job);  option.ResActor = rSys.Actor.GetById(actorDataId);  option.ResCharLevel = rSys.Level.GetLevelData(job, lv);  option.Position = \_player.PreservedActor.Position;  *// option.Status = player.Preserve.Status;*  *// TODO : 테스트를 위해*  option.Status = ActorStatus.**NORMAL**;  option.Hp = \_player.PreservedActor.Hp;  option.Mp = \_player.PreservedActor.Mp;  *// 확장값*  option.Player = \_player;  var tempActor = actorSys.CreateActor(option) as PlayerActor;  \_actor = tempActor;  *// Player Equip Item Value 적용*  \_player.Mount.Prepare();  *// 버프 적용*  \_player.Buff.Prepare();  var args = new FxPlayerActorBound()  {  Instant = DateTime.UtcNow,  };  \_player.Flux.Emit(args);  return tempActor;  } |
| --- |

| *public class RActorSet*  *{*  *private ImmutableDictionary<int, RActor> \_actors;*  *private ImmutableDictionary<int, RActorPortal> \_portals;*  *private ImmutableDictionary<int, RActorPlayer> \_players;*  *private ImmutableDictionary<int, RActorMonster> \_monsters;*  *private ImmutableDictionary<int, RActorNpc> \_npcs;*  *private ImmutableDictionary<int, RActorPet> \_pets;*  *// 플레이어 액터 생성/병합*  foreach (var element in loader.LoadPlayerData())  {  var rPlayer = RActorPlayer.OfPlayer(element.Value);  if (null == rPlayer)  {  throw new DataException($"not found player actor in base actors - id({element.Value.Id})");  }  tempPlayers.Add(element.Value.Id, rPlayer);  tempActors.Add(element.Value.Id, rPlayer);  }  \_actors = ImmutableDictionary.CreateRange(tempActors);  \_portals = ImmutableDictionary.CreateRange(tempPortals);  \_players = ImmutableDictionary.CreateRange(tempPlayers);  \_monsters = ImmutableDictionary.CreateRange(tempMonsters);  \_npcs = ImmutableDictionary.CreateRange(tempNpcs);  \_pets = ImmutableDictionary.CreateRange(tempPets); |
| --- |

# Excel 컬럼 추가

* 엑셀을 실행하여 ExCat에 메뉴





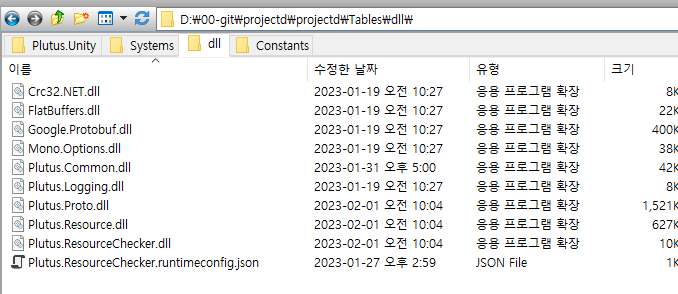
# 기획 리소스 체커

1. 소스 작업을 진행

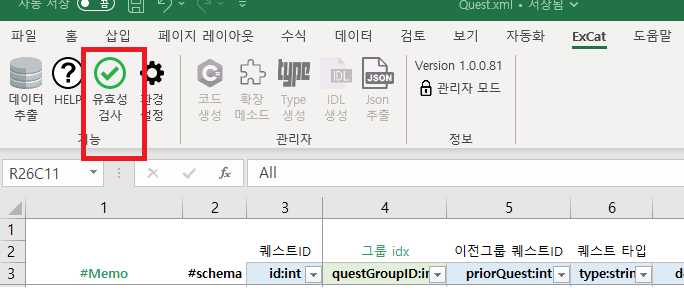
| public class QuestDataChecker : DataChecker  {  private ResourceContext \_res; |
| --- |

| *// quest objective 타입 유효성 검사*  private string ValidateQuestDatas()  {  foreach (var quest in \_quests)  {  var questData = quest.Value; |
| --- |

1. publish\_to\_client.bat 실행
2. 아래 클라이언트 폴더에 dll 파일이 맞게 변경되었는지 확인



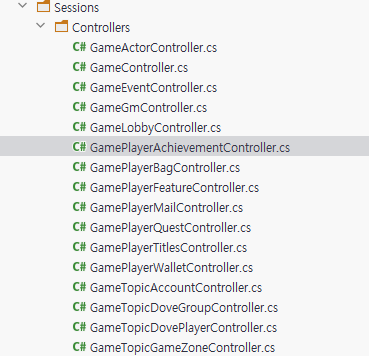
4.excel 을 열어서 체커실행



# 월드 broadcast

| var playerSys = GetSystem<GamePlayerSystem>();  var notify = new EventHotTimeUpdateNotify();  notify.AddedEventHotTimeSchedules.AddRange(modified.AddedSchedules.Select(x => ToTHotTime(x)));  notify.RemovedHotTimeIds.AddRange(modified.RemovedSchedules.Select(x => x.id));  playerSys.Broadcast(notify); |
| --- |

# 클라이언트 와 패킷 관련 처리



# Flux 관련

| var args = new FxPlayerActorBound()  {  Instant = Instants.Now(),  };  \_player.Flux.Emit(args); |
| --- |

PlayerEventHotTimeComponent.cs

| private void OnFxPlayerActorBound(FxPlayerActorBound args)  {  var hotTimeSys = GetSystem<EventHotTimeSystem>();  AddHotTimeBuffs(hotTimeSys.GetActivatedBuffIds());  } |
| --- |

# PlayerActor 에서 Play 가져오는 방법

| foreach (var value in AttrType.Values)  {  \_sources[value.Idx] = values[value.Idx];  } |
| --- |

| *// Actor*  var playerActor = \_actor as PlayerActor;  if (playerActor != null)  {  var player = playerActor.Bridge.Player;  var args = new FxAttrStats  {  Value = \_sources.ToArray()  };  player.Flux.Emit(args);  } |
| --- |

# 

# ItemCategory (ItemMainType과 ItemSubType)

* ItemMainType과 ItemSubType 간의 관계 정리

| private static ImmutableArray<ItemCategory> Values { get; } = ImmutableArray.Create(  None, Weapon, Armor, Accessory, Potion, Consume, ItemBox, Avatar, Currency  );  private static ItemCategory Consume { get; } = new ItemCategory(  ItemMainType.Consume,  ImmutableArray.Create(  ItemSubType.Material,  ItemSubType.Recipe,  ItemSubType.Quest,  ItemSubType.EnhanceWeapon,  ItemSubType.EnhanceArmor,  ItemSubType.EnhanceEssential,  ItemSubType.EnhanceSelectPreserve,  ItemSubType.ItemCraftSupport  )  ); |
| --- |