# 초기화 부분

| public class BaseTests  {  [OneTimeSetUp]  public void OnOneTimeSetUp()  {  if (null != TestWorld)  return;  ProtoDescriptor.Registry();  GameZonedDateTime.Initialize();  Env.Shared.Initialize("serilog.json", "server.properties");  Logger.Info($"tear up!!");  var tempWorld = new UnitTestWorld();  tempWorld.Initialize();  tempWorld.Start();  TestWorld = tempWorld;  }  [OneTimeTearDown]  public void OnOneTimeTearDown()  {  TestWorld.Stop();  Logger.Info($"tear down!!");  }  } |
| --- |

* 각 System들 초기화

| 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>();  Coupon = AddSystem<CouponSystem>();  ServerQueue = AddSystem<ServerQueueSystem>();  *// 게임 관련 서비스들*  Account = AddSystem<AccountSystem>();  GamePlayer = AddSystem<GamePlayerSystem>();  Azit = AddSystem<AzitSystem>();  Lobby = AddSystem<GameLobbySystem>();  PromissoryNote = AddSystem<PromissoryNoteSystem>();  ServerMail = AddSystem<ServerMailSystem>();  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*  DummyRes = AddSystem<DummyResourceSystem>();  DummyWorld = AddSystem<DummyWorldSystem>();  DummyGamePlayer = AddSystem<DummyGamePlayerSystem>();  DummyActor = AddSystem<DummyActorSystem>();  DummyDovePlayer = AddSystem<DummyDovePlayerSystem>();  DummyTopic = AddSystem<DummyTopicSystem>();  PromissoryNoteSystem = AddSystem<PromissoryNoteSystem>();  } |
| --- |

# 작업

* DummyGamePlayer 생성

| using var playerEntity = TestWorld.DummyGamePlayer.Create();  var player = playerEntity.GetComponent<GamePlayer>();  player.ActorProxy.Prepare();  player.Exp.Gain(ItemOriginFrom.Cheat, long.MaxValue);  var playerActor = player.ActorProxy.AsActor(); |
| --- |

* DummyActor 생성

| *// monster Resource Load*  *var rMonsters = TestWorld.Res.Actor.AsMonsterActors()*  *.Where(x => !string.IsNullOrEmpty(x.Monster.DropCurrency))*  *.Where(x => !string.IsNullOrEmpty(x.Monster.DropTableName));*  *var rMonster = rMonsters.ElementAt(Random.Shared.Next(0, rMonsters.Count()));*  *// Dummy 객체 생성*  using var entity = TestWorld.DummyActor.Create(*rMonster*);  var defenderActor = entity.GetComponent<DummyActor>();  var defenderHealth = entity.GetComponent<ActorHealthComponent>();  defenderHealth.Initialize(10000, 10000); |
| --- |

* DummyZone 생성

| *// zone 생성*  var rTowns = TestWorld.DummyRes.ToTowns();  var rMap = rTowns.ElementAt(Random.Shared.Next(rTowns.Count));  var zone = TestWorld.GameZone.Create(rMap.Spec.id);  // zone 에 입장  var playerActor = player.ActorProxy.AsActor();  zone.Gate.Enter(playerActor, GVector3.One); |
| --- |

# Assert 종류

| Assert.NotNull(assureProp);  Assert.AreEqual(StatusCodeEx.**Success**, status);  Assert.That(status, Is.EqualTo(StatusCodeEx.**Success**),  $"failed to complete - playerUid({player.Psn.Uid}) questId({achievementId})"); |
| --- |