Creating an Encounter for Encounter Journal

To create an encounter, follow these steps:

Step 1 – Create the encounter:

```
local boss_name = EJ_Data:CreateEncounter();
```

Step 2 – Setup the Attributes:

The attribute arguments are:

1. Name
2. Description
3. Instance ID
4. Encounter ID (Must be unique)
5. Index (Order in instance list)
6. DifficultyID

```
boss_name:setAttributes(name, desc, instanceID, encounterID, orderIndex, difficultyID);
```

Step 3 – Add the Loot List:

The loot argument is a list of all the item ids the boss drops – I usually try not to add consumables and quest items. If they really want that info they can download AtlasLoot.

```
boss_name:setLoot({});
```

Step 4 – Add encounter to the instance:

```
instance:addEncounter(boss_name);
```

Creating an Encounter for Encounter Journal

Step 5 – Creating Sections:

The section arguments are:

1. Header Name
2. Description
3. SpellID
4. IconMask
5. DifficultyMask

```
local section = EJ_Data:CreateSection("Phase 1", "", 0, 0, -1);
```

SpellID will change the section Header Name, add a spell icon, show its tooltip when hovered and append its tooltip to the section description.

IconMask can use up to 4 icons. To use them you will add the flags together:

```
EJ_Data.IconFlags.Tank + EJ_Data.IconFlags.Heroic
```

This will make the section header have the Tank Icon and the Heroic Icon
DifficultyMask can have however many flags you want. It also has a utility function that goes along with it:

```
local diffMask = DifficultyUtil.Mask;
local diffID = DifficultyUtil.ID;

diffMask[diffID.DungeonNormal] + diffMask[diffID.DungeonHeroic]
```

This will make the section only show up when viewing a dungeon and the difficulty is set to Normal or Heroic.
The function below allows for you to set a minimum difficulty

```
DifficultyUtil.OrHigher(diffID.DungeonHeroic);
```

This will make the section show up on Dungeon Heroic and Dungeon Mythic.

Creating an Encounter for Encounter Journal

Step 6 – Adding Sections to Encounter:

```
boss_name:addSection(section, nil);
```

The arguments for addSection are:

1. Section
2. ParentSection

Setting parent to nil will add it to the top level of the encounter.

To add a subsection, you will use the parent section as the second argument.

```
local sub_section = EJ_Data:CreateSection("Subsection", "", 0, 0, -1);
boss_name:addSection(sub_section, section);
```

Alternatively, you can all create the section in the argument if you don't need the section as a parent section.
Example:

```
boss_name:addSection(EJ_Data:CreateSection("Child", "", 0, 0, -1), sub_section);
```

This will add a child section to the sub_section made earlier.
By doing this however you will lose the ability to add additional children sections to this newly created section.

If you need any assistance with clarification or help setting up encounter sections message me on discord:
@merstrax

Creating an Encounter for Encounter Journal

Example Screenshot: (Not the real encounter)

```
----------------------
-------Lucifron-------
----------------------
local diffID = DifficultyUtil.ID;
local diffMask = DifficultyUtil.DifficultyMask;
local iconFlags = EJ_Data.IconFlags;

local lucifron = EJ_Data:CreateEncounter();
lucifron:setAttributes("Lucifron", "", 741, 663, 1, 8);
lucifron:setLoot({18872,19145,19146,18875,18870,18861,17109,18879,19147,18878,17077,2522362});

--Phase 1 Section
local lucifron_phase_1 = EJ_Data:CreateSection("Phase 1", "", 0, 0, -1);
lucifron:addSection(lucifron_phase_1, nil)
lucifron:addSection(EJ_Data:CreateSection("Test 1", "Test 1", 0, 0, -1), lucifron_phase_1);
lucifron:addSection(EJ_Data:CreateSection(GetSpellLink(72133), "Spell Link Test", 72133, iconFlags.Magic, -1), lucifron_phase_1);
lucifron:addSection(EJ_Data:CreateSection("Heroic", "This should only show up when Heroic is selected", 0, iconFlags.Heroic, diffMask[diffID.RaidHeroic]), lucifron_phase_1);
lucifron:addSection(EJ_Data:CreateSection("Mythic or Higher", "This only shows up when Mythic or higher is selected", 0, iconFlags.Mythic, DifficultyUtil.OrHigher(diffID.RaidMythic)), lucifron_phase_1);
lucifron:addSection(EJ_Data:CreateSection("Ascended", "This should only show up when Ascended is selected", 0, iconFlags.Ascended, diffMask[diffID.RaidAscended]), lucifron_phase_1);

--Phase 2 Section
local lucifron_phase_2 = EJ_Data:CreateSection("Phase 2", "Test description with subsections", 0, 0, -1);
lucifron:addSection(lucifron_phase_2, nil)
--Tank Section
local lucifron_phase_2_tank = EJ_Data:CreateSection("Tank", "", 0, iconFlags.Tank, -1);
lucifron:addSection(lucifron_phase_2_tank, lucifron_phase_2);
lucifron:addSection(EJ_Data:CreateSection("Fatal", "This is a fatal subsection", 0, iconFlags.Fatal, -1), lucifron_phase_2_tank);
lucifron:addSection(EJ_Data:CreateSection("Bleed", "This is a bleed subsection", 0, iconFlags.Bleed, -1), lucifron_phase_2_tank);
lucifron:addSection(EJ_Data:CreateSection("Adds", "This is an adds subsection", 0, iconFlags.Adds, -1), lucifron_phase_2_tank);
--DPS Section
local lucifron_phase_2_dps = EJ_Data:CreateSection("DPS", "", 0, iconFlags.DPS, -1);
lucifron:addSection(lucifron_phase_2_dps, lucifron_phase_2);
lucifron:addSection(EJ_Data:CreateSection("Important", "This is an important subsection", 0, iconFlags.Important, -1), lucifron_phase_2_dps);
lucifron:addSection(EJ_Data:CreateSection("Interruptable", "This is an interruptable subsection", 0, iconFlags.Interruptable, -1), lucifron_phase_2_dps);
--Healer Section
local lucifron_phase_2_heals = EJ_Data:CreateSection("Healer", "", 0, iconFlags.Healer, -1);
lucifron:addSection(lucifron_phase_2_heals, lucifron_phase_2);
lucifron:addSection(EJ_Data:CreateSection("Curse", "This is a curse subsection", 0, iconFlags.Curse, -1), lucifron_phase_2_heals);
lucifron:addSection(EJ_Data:CreateSection("Poison", "This is a poison subsection", 0, iconFlags.Poison, -1), lucifron_phase_2_heals);
lucifron:addSection(EJ_Data:CreateSection("Disease", "This is a disease subsection", 0, iconFlags.Disease, -1), lucifron_phase_2_heals);

instance:addEncounter(lucifron);
```