

# Live Ops Engine

## Background

Create a website that lists and allows editing of offers to be shown in a game. An offer is a purchasable item, that when bought gives you one or more in-game items (that the player can use). It may provide a discounted purchase price (like 50% off), have customized content (items are not fixed), targeted to a certain set of players (based on rules/conditions), and with some scheduling settings (like weekend only).

An offer is represented by the following data:

- **Id** – A unique identifier for the offer.
- **Display attributes** – Fields like name of the offer, its description, sort order, and a background image.
- **Content** – List of items packaged within it (each item is represented by a unique id and quantity).
- **Scheduling parameters** – The offer can be scheduled for specific days of the week, specific dates of a month, specific months of a year or any combination of them.
- **Targeting rules** – A set of conditions which a player should match for him/her to be eligible to see the offer and make a purchase (like age > 30 and installed\_days < 5).
- **Pricing options** – The offer can have multiple pricing like it can be bought for 1000 coins or 20 gems (where coins/gems are some currencies within the game).

## Solution Requirements

Build the offers website to demonstrate the following functionality:

1. Provide basic authentication or any other login mechanism (with username/password at a minimum and no unauthenticated access allowed).
2. Create a new offer object and save it.
3. Edit all attributes of an existing offer with validation.
4. Upload images to the server to be used by the offer display.
5. Delete an obsolete offer.
6. Search/filter offers over text fields.
7. Use drag and drop to change the sorting order of the offers.
8. Create a UI control to take as input the player target condition(s) using field names, operators, and matching values.

## Object Schema

Here is documentation for important objects required for the solution in JSON format.

### Player:-

```
{
  "player_id": "<GUID>",
  "age": 35,
  "country": "IN",
  "installed_days": 10,
  "coins": 10000,
  "gems": 2,
  "game_level": 10,
  "purchaser": false
}
```

### Offers:-

```
{
  "offer_id": "OFF-1000",
  "offer_title": "Diwali Offer",
  "offer_description": "Only for next 10 days!",
  "offer_image": "http:///offers/diwali_celerbration.png",
  "offer_sort_order": 100,
  "content": [{ "item_id": "ITEM-1", "quantity": 10}, {"item_id":
  "ITEM-2", "quantity": 1}],
  "schedule": { "days_of_week": [1, 2, 3], "dates_of_month": [5,
  6, 7, 8, 9, 10, 11, 12, 13, 14], "months_of_year": [11]},
  "target": "age > 30 and installed_days < 5",
  "pricing": [{ "currency": "coins", "cost": 1000 }, {"currency":
  "gems", "cost": 20} ]
}
```

**Condition** The conditions language is a combination of ANDed or ORed conditions on the fields of the player object and stored in a simple string format. age > 30 and installed\_days < 5

## Available API

Assume that the following HTTP REST APIs exist for you to consume and manipulate the offers' data.

1. **GET Offers/?page=1&records=100&attribute=offer\_title&query=Diwali:** Returns paginated list of offers matching the search criteria over an offer attribute and with a string value to look for.

Query parameters (optional):

page = Page number  
records = Number of records per page  
attribute = Offer field to search  
query = Actual value to search

Response (200 OK) with payload

```
{
  "page": 1, "has_more": false,
  "offer": [{"offer_id": "OFF-1000", "offer_title": "Diwali
Offer",. . . . .}, {"offer_id": "OFF-2000",
"offer_title": "New Year Offer",. . . . .}],. . . . .]
}
```

2. **PUT Offers/<offer-id>**

Creates/updates an offer.

Request

```
{
  "offer_id": "OFF-1000",
  "offer_title": "2020 Diwali Offer",
  "offer_description": "Only for next FEW days!",
  "offer_image": "http:///offers/diwali_celerbration.png",
  "offer_sort_order": 100,
  "content": [{"item_id": "ITEM-1", "quantity": 100},
{"item_id": "ITEM-2", "quantity": 10}],
  "schedule": {"days_of_week": [1, 2, 3], "dates_of_month":
[5, 6, 7, 8, 9, 10, 11, 12, 13, 14],
"months_of_year": [11]},
  "target": "age > 30 and installed_days < 5",
  "pricing": [{"currency": "coins", "cost": 1000},
{"currency": "gems", "cost": 20}]
}
```

Response (200 OK)

### 3. **DELETE Offers/**

Deletes a particular offer permanently. Response (200 OK)