

# OSF Gamification Project

---

## Professeur:

**Prof. Olivier Liechti**

## Etudiants:

**Diana AFFI**

**Joseph EL MALLAH**

**Ahmed Hachmi**

7 Octobre 2013

## Table of Contents

<b>Introduction .....</b>	<b>3</b>
<b>Gamification object diagram.....</b>	<b>4</b>
<b>Player .....</b>	<b>4</b>
<b>Awards .....</b>	<b>5</b>
<b>Event .....</b>	<b>5</b>
<b>Action .....</b>	<b>5</b>
<b>Restful URL patterns.....</b>	<b>7</b>
<b>Player .....</b>	<b>7</b>
<b>Person .....</b>	<b>7</b>
<b>Group.....</b>	<b>7</b>

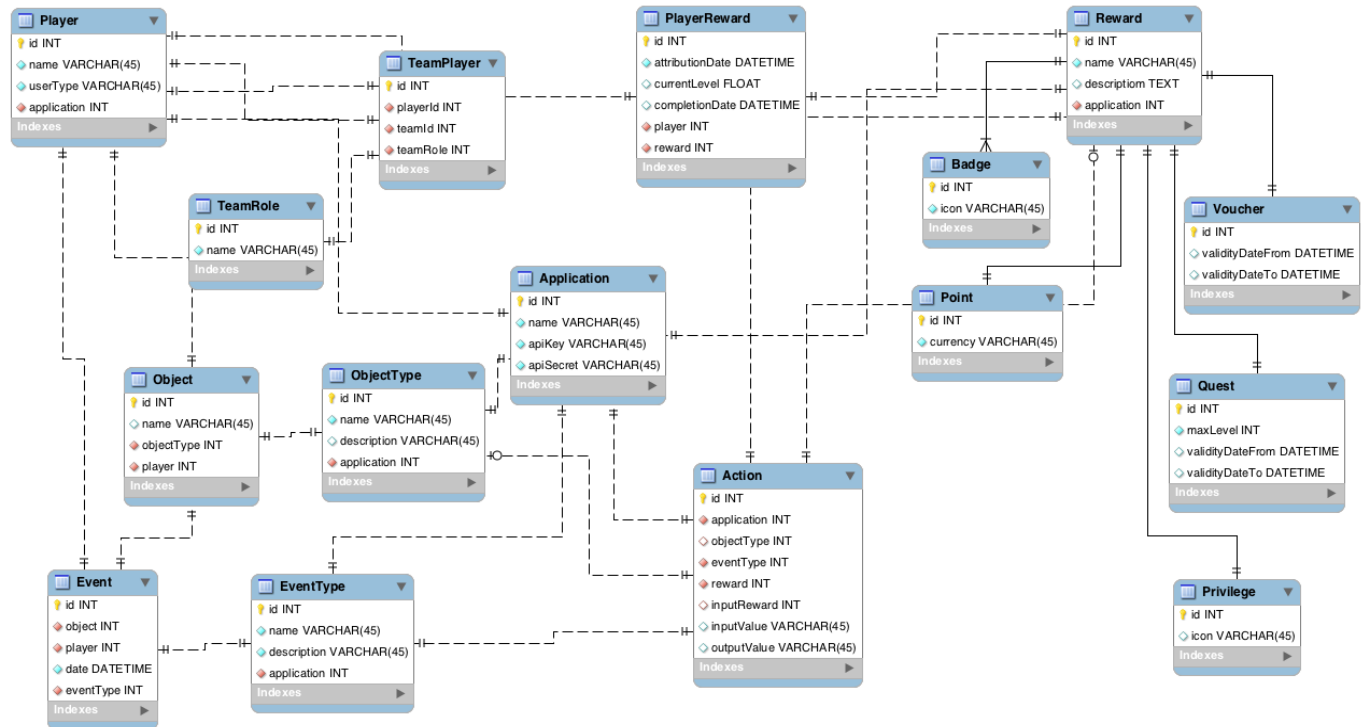
## Introduction

The project developed is a Gamification API that is designed to work as a service for other social applications. Our API manages all the users including their actions and progress through a specific application.

The access to the API is done through web services developed to control all the entities and aspects of the application. The client application will be able to manipulate the offered resources by calling different types of web services (GET, PUT, POST, DELETE).

In the following sections, the data model used to obtain all the functionalities and the different web services available will be detailed.

## Gamification object diagram



## Application

Each application using GamificationAPI is saved in the application table. On the creation of a new application, the user receives an apiKey and an apiSecret. These two parameters have to be saved and kept by the client. For now, the API do not use these parameters, but in the future, all the web service requests concerning a specific application, will require sending the credentials to assure the data security.

## Player

The main object in this diagram is the “Player”. The player can be a single person or a team. This is specified in the playerType attribute of the Player.

The members of a team are specified in the TeamPlayerObject. Each member of a team has specific roles in that team.

## Awards

Our API offers different types of rewards. A player can have multiple rewards which acquisition is saved in the PlayerReward object.

The reward types are the following:

- **Badge:** The badge is a simple reward that holds an icon.
- **Privilege:** The privilege is a level earned by the player that may guarantee him a bigger diversity of activity.
- **Point:** The points are a type of reward measured in a certain currency and which value may increase.
- **Voucher:** The voucher is a one time use reward that may expire in time
- **Quest:** The quest is a reward earned by passing through different levels to finally reach the highest level

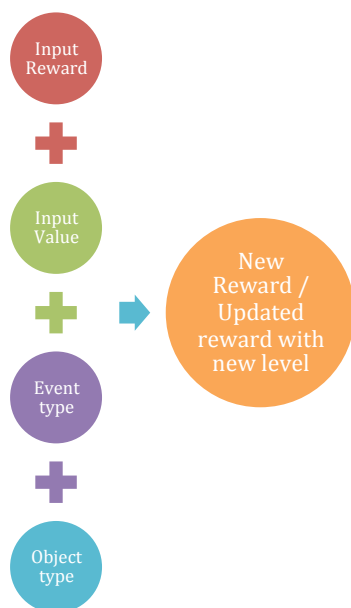
## Event

An event is invoked by a player on an object possessed by either another player or the same player. Each event has a type, and each object has a type too.

## Action

An action can be considered as a set of rules that enables a player getting a reward. In fact the action takes in input a player, an eventType and an objectType, and produces in output a reward that will be assigned to the player who invoked the event. The result may also contain an output value in the case of a point reward or a quest reward.

The rules are automatically checked when an event happens. In case of a match, the input rewards are consumed (voucher or points) or just used, and the output rewards are added to the player. If the reward already exists, the output value is edited for that reward, i.e. the level of the quest is updated and points are added.



## Restful URL patterns

This is a brief description of the web services patterns offered by our application. The other objects in the data model can be accessed in the same way. An interactive documentation can be found on <http://docs.gamificationapidianajosephahmed.apiary.io>.

### Player

- <http://www.my-gamification.com/webresources/players/>
  - GET to retrieve the list of all players including group and persons
- [http://www.my-gamification.com/webresources/players?search=\[string\]](http://www.my-gamification.com/webresources/players?search=[string])
  - GET to retrieve the list of all players matching the criterion [string]
- [http://www.my-gamification.com/webresources/players/\[id\]/rewards/](http://www.my-gamification.com/webresources/players/[id]/rewards/)
  - GET to retrieve all rewards of the player with id [id]
  - POST to add a reward to player with id [id]
    - Returns 201 Added
- [http://www.my-gamification.com/webresources/players/\[id\]/rewards/\[id1\]](http://www.my-gamification.com/webresources/players/[id]/rewards/[id1])
  - GET to retrieve the reward with id [id1] of the player with id [id]
- [http://www.my-gamification.com/webresources/players/\[id\]](http://www.my-gamification.com/webresources/players/[id])
  - redirects to [http://www.my-gamification.com/webresources/persons/\[id\]](http://www.my-gamification.com/webresources/persons/[id]) if the player of id [id] is a person
  - redirects to [http://www.my-gamification.com/webresources/groups/\[id\]](http://www.my-gamification.com/webresources/groups/[id]) if the player of id [id] is a group

### Person

- <http://www.my-gamification.com/webresources/persons/>
  - GET to retrieve the list of all persons
  - POST to create a new person
    - Returns 201 Created
- [http://www.my-gamification.com/webresources/persons/\[id\]](http://www.my-gamification.com/webresources/persons/[id])
  - GET to retrieve
  - PUT to update
  - DELETE to delete

### Group

- <http://www.my-gamification.com/webresources/groups/>
  - GET to retrieve the list of all groups
  - POST to create a new group
    - Returns 201 Created
- [http://www.my-gamification.com/webresources/groups/\[id\]](http://www.my-gamification.com/webresources/groups/[id])
  - GET to retrieve

- PUT to update
- DELETE to delete

*N.B: All the GET webservice should receive in parameter the application id in the key "appId"*