Name: Email:
Phone:

# Conway’s Game of Life

The universe of the Game of Life is an infinite two-dimensional orthogonal grid of square cells, each of which is in one of two possible states, alive or dead, or “populated” or “unpopulated”. Every cell interacts with its eight neighbours, which are the cells that are horizontally, vertically, or diagonally adjacent. At each step in time, the following transitions occur:

- Any live cell with fewer than two live neighbours dies, as if caused by underpopulation.
- Any live cell with two or three live neighbours lives on to the next generation.
- Any live cell with more than three live neighbours dies, as if by overpopulation.
- Any dead cell with exactly three live neighbours becomes a live cell, as if by reproduction.

## REST User Store

## `POST /grids`

### Request:

```
{
    "x" : <size_of_x_axis>,
    "y" : <size_of_y_axis>,
    "data": "<grid_representation>"
}
```

### Response:

**Status:** **`<Use Appropriate Status Code>`**

```
{
    "id": <id>,
    "x" : <size_of_x_axis>,
    "y" : <size_of_y_axis>,
    "data": <grid_representation>
}
```

# PATCH /grids/<id>

### Request:

```
{
    "x" : <size_of_x_axis>,
    "y" : <size_of_y_axis>,
    "data": <grid_representation>
}
```

### Response:

**Status:** **`<Use Appropriate Status Code>`**

```
{}
```

# GET /grids/{id}

### Response:

**Status:** **`<Use Appropriate Status Code>`**

```
{
    "id: <id>
    "x" : <size_of_x_axis>,
    "y" : <size_of_y_axis>,
    "data": <grid_representation>
}
```

# GET /grids/{id}?after=<age_1>,<age_2>,<age_3>

### Response:

**Status:** **<Use Appropriate Status Code>**

```
{
    "id: <id>
    "x" : <size_of_x_axis>,
    "y" : <size_of_y_axis>,
    "data": [
        {
            "age": <age_1>
            "grid": <grid_representation_after_age_1_secs>
        },{
            "age": <age_2>
            "grid": <grid_representation_after_age_2_secs>
        },{
            "age": <age_3>
            "grid": <grid_representation_after_age_3_secs>
        }
    ]
}
```

## Constraints:

- Done is better than Perfect.
- Has to be fast.
- **Must Dockerize the entire application.**
- **Entire codebase must be tracked in a Git Repository.**
- Field Values can be of arbitrary length.
- **Must be fault tolerant.**
- **Use appropiate status codes for all responses.**