

### **Getting Started:**

1. Download the code from  
<https://www.dropbox.com/s/mt5yxmhux49dznj/payjo-nodejs-test.zip?dl=0>
2. Run `npm run prestart` (to install the required dependencies)
3. To run the project use `npm start`
4. Complete the below tasks
5. Run this command to run the test cases on your code  
`mocha test/tests.js`

### **Data Structure**

```
{
  event: {
    id: INT
    type: STRING
    actor: {
      id: INT
      login: STRING
      avatar_url: STRING
    },
    repo: {
      id: INT
      name: STRING
      url: STRING
    },
    created_at: TIMESTAMP
  }
}
```

**# Use any database to created above data structure (Note: Use only a relational database)**

**# Tasks - Create the below mentioned api's**

- **Add Event**

Description: Create event, actor, repo

Path: /events

Method: POST

Request Payload:

```
{
  "id":4055191679,
  "type":"PushEvent",
  "actor":{
    "id":2790311,
```

```
"login":"daniel33",
"avatar_url":"https://avatars.com/2790311"
},
"repo":{
  "id":352806,
  "name":"johnbolton/exercitationem",
  "url":"https://github.com/johnbolton/exercitationem"
},
"created_at":"2015-10-03 06:13:31"
}
```

Reponse:

Status Code: 201

- **Get All events**

Description: Get all events, in ascending order by Event ID

Path: /events

Method: Get

Request Payload: {}

Response

Status Code: 200

Sample Response Body:

```
[
  {
    "id":4055191679,
    "type":"PushEvent",
    "actor":{
      "id":2790311,
      "login":"daniel33",
      "avatar_url":"https://avatars.com/2790311"
    },
    "repo":{
      "id":352806,
      "name":"johnbolton/exercitationem",
      "url":"https://github.com/johnbolton/exercitationem"
    },
    "created_at":"2015-10-03 06:13:31"
  }
]
```

- **Get Event by actor**

Description: Get event by actorID, in ascending order by Event ID

Path: /events/actors/:actorID

Method: GET

Response

Status Code: 200

Sample Response Body:

```
[
  {
    "id":4055191679,
    "type":"PushEvent",
    "actor":{
      "id":2790311,
      "login":"daniel33",
      "avatar_url":"https://avatars.com/2790311"
    },
    "repo":{
      "id":352806,
      "name":"johnbolton/exercitationem",
      "url":"https://github.com/johnbolton/exercitationem"
    },
    "created_at":"2015-10-03 06:13:31"
  }
]
```

- **Erase events**

Description: Clear all data events, actors & repo data

Path: /erase

Method: DELETE

Request payload: {}

Reponse

Status Code: 200

- **Get All actors**

Description: Get all actors, in descending order by number of events actor is associated with

Path: /actors

Method: GET

Response

Status Code: 200

Sample response Body

```
[
  {
    "id":3967160,
    "login":"crystal47",
    "avatar_url":"https://avatars.com/3967160"
  }
]
```

- **Update actor**

Description: Update actor details by actor Id

Path: /actors

Method: PushEvent

Request Body

```
{
  "id":3967160,
  "login":"crystal47",
  "avatar_url":"https://avatars.com/3967160"
}
```

Response

Status code: 200

- **Get Actos streak**

Description:

Get actors by their streak.

(streak is calculated by total number of consecutive events in which actor is associated).

Return response in descending order of actor streak.

If two or more actors have same streak, then sort them by total number of events they are associated with in descending order

If two or more actors have same streak and same events count, then sort them by created\_at in descending order

Path: '/actors/streak'

Method: GET

Request Payload: {}

Response

Status Code: 200

Sample response Body

```
[
  {
    "id":3967160,
    "login":"crystal47",
    "avatar_url":"https://avatars.com/3967160"
  }
]
```

**Note:**

- We have basic framework setup. You have to create routes and write the code
- You can install any dependencies you need.
- After completing the tasks, send your code as a zip file or in github repo
- Also include any setup docs (if required)
- **Evaluation will be based on coding style and test cases passed.**