**REST API Task**

Implement a REST API to manage PRODUCTS and ORDERS, using public VAT API to get country tax rates.

**Requirements**

* Comply with REST standard - <https://hackernoon.com/restful-api-designing-guidelines-the-best-practices-60e1d954e7c9>
* Answer with the relevant status codes, based on the result
* Create an endpoint for PRODUCTS
  + Each product should have name, category, price
  + The API should support getting all products, adding a product, editing a product, deleting a product
  + The response format for all products is added bellow
* Create an endpoint for ORDERS
  + Each order should have a date, products list (product ids) and status (Pending, Processing, Delivered, Cancelled)
  + The API should support getting all orders, adding an order (date should be automatically populated), change order status
  + The response format for all orders is added bellow

**Technologies**

Technologies by choice. Optionally database can be in-memory for simplicity. Make sure to prepopulate with data however (categories, several products and orders)

* Implement authentication
  + A request should be send with username and password
  + Retrieving all PRODUCTS should be allowed for anonymous users
  + Relevant status code should be send if token is not provided or is invalid/expiredс

**Products**

[

{

"id": 1,

"name": "Apple",

"category": "Fruit",

"price": "1"

},

{

"id": 2,

"name": "Milk",

"category": "Diary",

"price": "2.50"

}

]

**Orders**

[

{

"id": 1,

"date": "2018-05-29",

"products": [1, 2],

"status": "Delivered"

},

{

"id": 2,

"date": "2018-05-30",

"products": [1],

"status": "Pending"

}

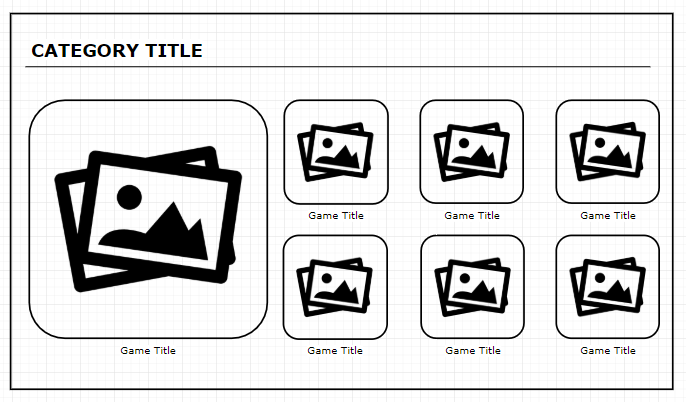
]

**FrontEnd - Game Grid**

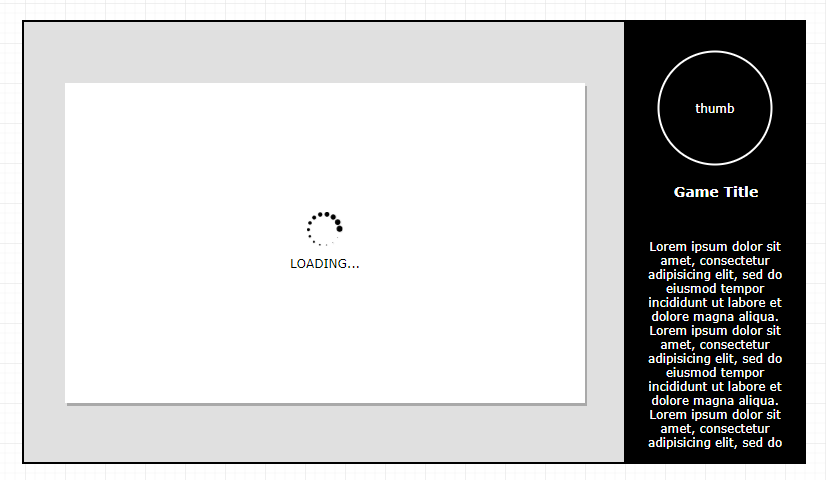
Implement a ReactJS casino grid app

**Requirements**

* Use React with JS or TS
* The app should have two views
  + **Grid View** – shows a list of games with title and thumbnail in a grid layout. First game in the collection should be shown as big one. Upon clicking on any of the items, the “Active Item View” is shown.



* + **Active Item View** – shows thumbnail, title and description of the game. A “playbox” is shown in the center, which will only include a loading animation (game loading simulation)



* Design – design and color is by choice. Colors in the screenshot are just an example
* Data – please use the following as mock data to fill the grid - <https://drive.google.com/file/d/1DUDNuGoBX1soI0jHg2HCUnyeN0JKsaRs>
* CSS libraries are allowed