| **Name:** | Lin Mei An |
| --- | --- |
| **Email:** | lin\_mei\_an@mpa.gov.sg |

**Question 1**

In your own words explain the 4 sub constraints in REST’s Uniform Interface. Give an everyday example to illustrate each of the constraint.

1. Resource can be uniquely identified
   1. resources should have unique names/primary keys for identification
   2. e.g. cust1 and cust2
2. Resource can be manipulated through representations
   1. different formats of the same data can be generated (e.g. text/html, pdf or json) and the changes made to the data are done through the different representations at different stages
   2. e.g.data is converted to html for display and edit on browser and then changed again when sending back to database for update
3. Self describing messages:
   1. all messages passed from client to server must be self contained and contains all information needed to understand and decode the message
   2. e.g. message contains request type, request item and query string
4. HATEOAS:
   1. information is provided dynamically through hypermedia/hyperlinks
   2. e.g. details page of an item that can be accessed through clicking on 1 item in a list

**Question 2**

What is the difference between the following HTTP methods?

1. POST, PUT and PATCH

POST creates a new record every call, PUT inserts the record or replaces the existing copy, PATCH updates only selected parts of a record.

(new, update full, update partial)

1. GET and HEAD

GET returns the payload along with the header while HEAD only returns the header.

**Question 3**

You have a monolithic web application for managing warehouses. The application exposes the following end points

* /warehouses – list of all warehouses
* /warehouse/<warehouse\_id> – returns the warehouse’s details
* /warehouse/<warehouse\_id>/inventories – inventory list for the warehouse
* /inventories – list of all the inventories
* /inventory/<inventory\_id> – inventory detail
* /inventory/<inventory\_id>/report – generate a report

Describe how you can scale this application

1. By duplication

increase the number of application servers

1. By functional decomposition

break up the applications into individual modules according to their functions and the portions of the database that they work with. Can be partitioned according to below:

* /warehouses
* /warehouse/<warehouse\_id>
* /warehouse/<warehouse\_id>/inventories
  + API call from warehouse database to inventories database
* /inventories
* /inventory/<inventory\_id>
* /inventory/<inventory\_id>/report

(separate database by domain name e.g. warehouse.domain.com vs inventory.domain.com → handled by the application or 1 database with 2 applications → each handling requests for a different domain name)



1. By data partitioning

logical partition of database (e.g. region)

* Data from warehouses of the same region can be stored in 1 database, multiple databases used to store all information of all warehouses
* Can have many or 1 application server retrieving data from all databases

**Question 4**

Study the top headlines REST API from newsapi.org. Answer the following questions

1. List the different ways you can present the API key when performing an invocation
2. Via the apiKey querystring parameter.
3. Via the X-Api-Key HTTP header.
4. Via the Authorization HTTP header. Including Bearer is optional, and be sure not to base 64 encode it like you may have seen in other authentication tutorials.

(always much better for it to be in the header > in query string because it can be bookmarked and revealed → security reasons)

1. Construct a URL to get 30 technologies headlines from Japan

https://newsapi.org/v2/everything?country=jp&category=technology&sortBy=popularity&pageSize=30&apiKey=API\_KEY

1. What is the status code if an incorrect API key is used?

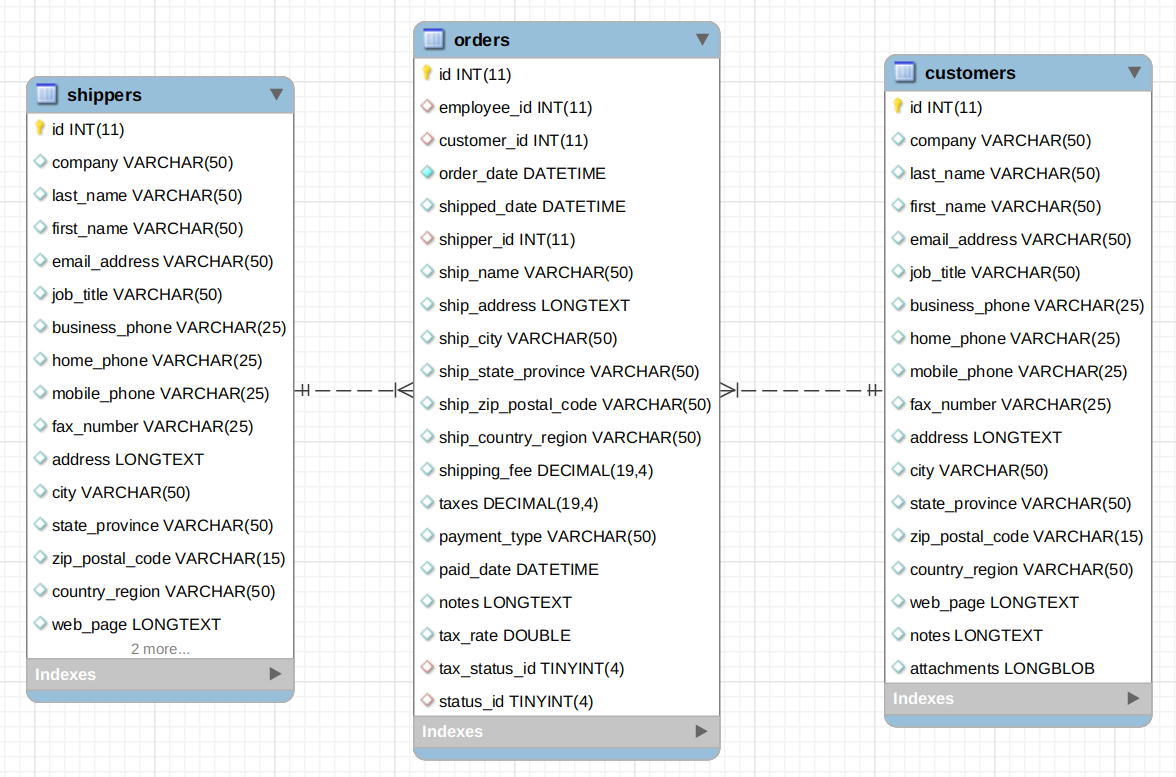
401 - Unauthorized

1. How long will the result be cached?

3 minutes

**Question 5**

Study the following entity-relationship diagram



Both customer and shippers has a one to many relationships with orders.

Answer the following questions.

1. Design one or more API endpoints to return a list of customers and a single customer

* list:

request.get(

url: ‘/customers, (req, resp) => {

cust[] = //retrieve list

resp.status(200).type(‘text/plain’)

resp.send(cust)})

* single:

request.get

url: ‘/customer/<id>, (req, resp) => {

cust = //get customer fromid

resp.status(200).type(‘text/plain’)

resp.send(cust)

})

1. What are some criteria and how might you might include in your endpoint (wrt Q5a)?
2. implement paging/cursor
3. chunked encoding
4. compression
5. Show a sample output of a customer’s list as a result of performing a GET on the resource. (wrt Q5a)
6. How do you provide flow control or pagination support (wrt Q5a)?

* use “accept-ranges”?????

**Question 6**

You have deployed a service to encode video viz. convert AVI to mp4, etc. Subscribers of your service uploads their video to the service; after conversion the converted video is returned to the subscriber (assume that the conversion time is short).

You charge the subscribers based on the 2 criteria.

1. Subscription rates based on the cumulative video sizes: 500GB, 1TB, 1.5TB, etc. A subscriber who subscribe to the 500GB package can upload a maximum amount of 500GB videos.
2. Charge the subscribe based on their ingress and egress traffic viz. the upload and downloads of the videos.

Design an API for this encoding service to give your subscribers control over their encoding process.

You can ignore authentication.

→ rate limiting

→ warnings → partial response to stop upload before it goes over the maximum amount

→ compression

**Submission**

Copy this Word document to your repository and commit it.

git add .

git commit -m ‘worksheet01’

git push origin master