```
from random import random
from flask_restful import Resource
from utils.cookieChecker import token_required
from utils.dbQuery import selectQuery
from utils.apiFetch import apiData
import random
from utils.testData import retArray


class Personal(Resource):
    @token_required
    def get(email, self):
        topicsArr = ["sport", "tech", "world", "finance", "politics", "business",
                     "economics", "entertainment", "beauty", "travel", "music", "food",
"science", "cricket"]
        fav = selectQuery("SELECT favourites from user where email=?",
(email,))[
            'FAVOURITES']
        fav = fav.split(',')
        for x in fav:
            topicsArr.remove(x)
        retArr = []
        favList = []
        # favList = retArray
        nonFavList = []
        for x in fav:
            try:
                data = apiData(x)
            except:
                data=None
            if (data is not None):
                for y in data:
                    favList.append(y)
        try:
            random.shuffle(favList)
            favList = sorted(favList, key=lambda k: k['date'], reverse=True)
        except:
            favList=[]
        for x in topicsArr:
            try:
                data = apiData(x)
            except:
                data=None
            if(data is not None):
                for y in data:
                    nonFavList.append(y)
        try:
            random.shuffle(nonFavList)
            nonFavList = sorted(nonFavList, key=lambda k: k['date'], reverse=True)
        except:
            nonFavList=[]
        retArr = favList+nonFavList
        return {"data": retArr}, 200

class News(Resource):
    @token_required
    def get(email,self,topic):
        topicsArr = ["headline","sport", "tech", "world", "finance", "politics",
"business",
                     "economics", "entertainment", "beauty", "travel", "music", "food",
"science", "cricket"]
        if(topic not in topicsArr):
            return {"status":"Not a valid topic"},404
        try:
            retArr=apiData(topic)
            random.shuffle(retArr)
            retArr=sorted(retArr,key=lambda k:k['date'],reverse=True)
        except:
            retArr=[]
        return {"data":retArr},200
```