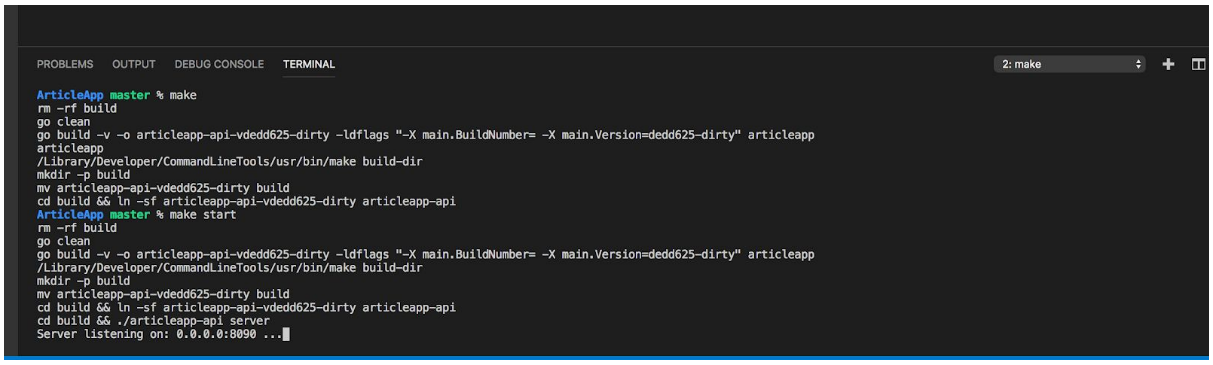
Assumptions:
In memory implementation is considered for this application. However, a more scalable cache such as redis can be easily incorporated. I have included a file that intends to implement a redis based cache.

Given time,
I would have implemented a cache/data store using redis as well.

Instructions for running the app:
Outside container:

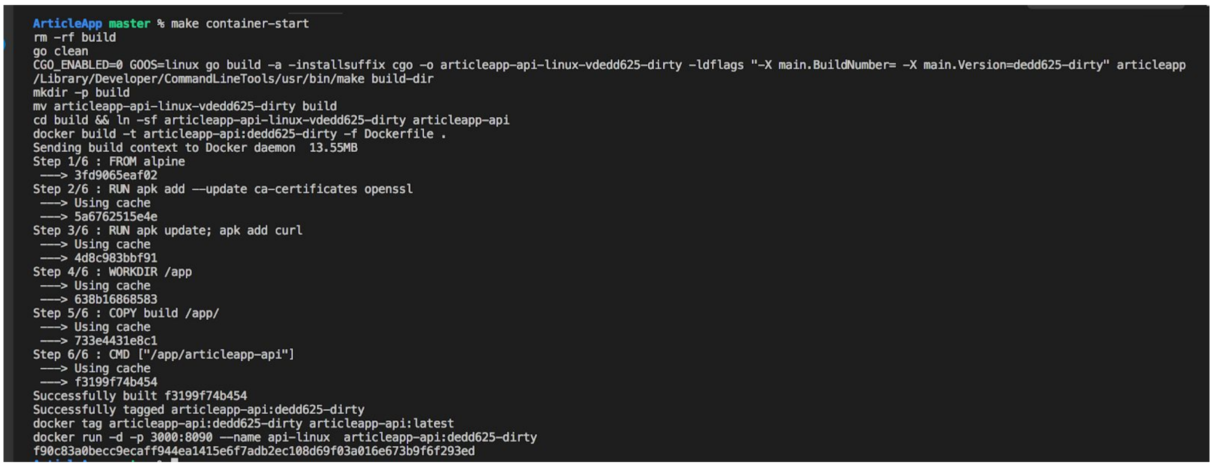
1. **make build**
2. **make start**



```
ArticleApp master % make
rm -rf build
go clean
go build -v -o articleapp-api-vdedd625-dirty -ldflags "-X main.BuildNumber= -X main.Version=dedd625-dirty" articleapp
articleapp
/Library/Developer/CommandLineTools/usr/bin/make build-dir
mkdir -p build
mv articleapp-api-vdedd625-dirty build
cd build && ln -sf articleapp-api-vdedd625-dirty articleapp-api
ArticleApp master % make start
rm -rf build
go clean
go build -v -o articleapp-api-vdedd625-dirty -ldflags "-X main.BuildNumber= -X main.Version=dedd625-dirty" articleapp
/Library/Developer/CommandLineTools/usr/bin/make build-dir
mkdir -p build
mv articleapp-api-vdedd625-dirty build
cd build && ln -sf articleapp-api-vdedd625-dirty articleapp-api
cd build && ./articleapp-api server
Server listening on: 0.0.0.0:8090 ...
```

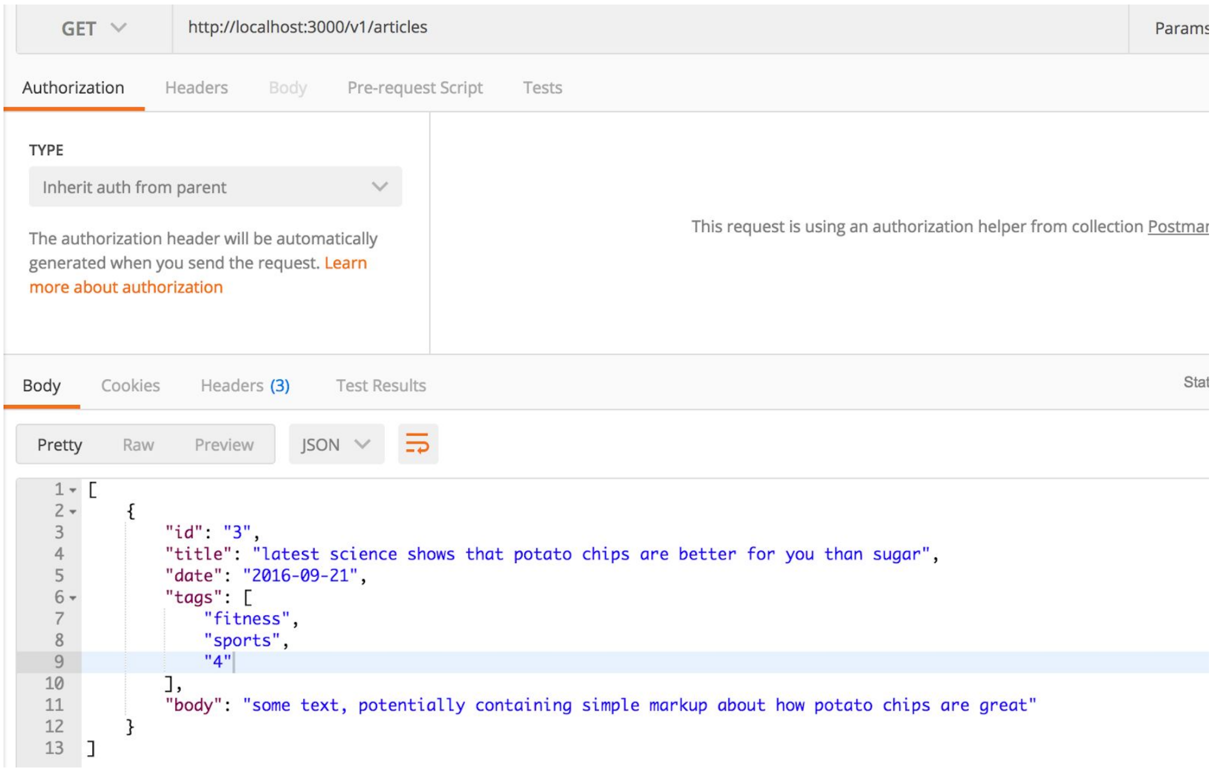
Run container:
make container-start # this will build and start a container

If container exists already, please run
make container-stop



```
ArticleApp master % make container-start
rm -rf build
go clean
CGO_ENABLED=0 GOOS=linux go build -a -installsuffix cgo -o articleapp-api-linux-vdedd625-dirty -ldflags "-X main.BuildNumber= -X main.Version=dedd625-dirty" articleapp
/Library/Developer/CommandLineTools/usr/bin/make build-dir
mkdir -p build
mv articleapp-api-linux-vdedd625-dirty build
cd build && ln -sf articleapp-api-linux-vdedd625-dirty articleapp-api
docker build -t articleapp-api:dedd625-dirty -f Dockerfile .
Sending build context to Docker daemon  13.55MB
Step 1/6 : FROM alpine
 ---> 3fd9065eaf02
Step 2/6 : RUN apk add --update ca-certificates openssl
 ---> Using cache
 ---> 5a6762515e4e
Step 3/6 : RUN apk update; apk add curl
 ---> Using cache
 ---> 4d8c983bbf91
Step 4/6 : WORKDIR /app
 ---> Using cache
 ---> 638b16868583
Step 5/6 : COPY build /app/
 ---> Using cache
 ---> 733e4431e8c1
Step 6/6 : CMD ["/app/articleapp-api"]
 ---> Using cache
 ---> f3199f74b454
Successfully built f3199f74b454
Successfully tagged articleapp-api:dedd625-dirty
docker tag articleapp-api:dedd625-dirty articleapp-api:latest
docker run -d -p 3000:8090 --name api-linux  articleapp-api:dedd625-dirty
f90c83a0becc9ecaff944ea1415e6f7adb2ec108d69f03a016e673b9f6f293ed
```

Container output (port 3000 mapped to 8090)


GET http://localhost:3000/v1/articles

Authorization Headers Body Pre-request Script Tests

TYPE

Inherit auth from parent

The authorization header will be automatically generated when you send the request. Learn more about authorization

This request is using an authorization helper from collection Postmar

Body Cookies Headers (3) Test Results

Pretty Raw Preview JSON

```
[
    {
        "id": "3",
        "title": "latest science shows that potato chips are better for you than sugar",
        "date": "2016-09-21",
        "tags": [
            "fitness",
            "sports",
            "4"
        ],
        "body": "some text, potentially containing simple markup about how potato chips are great"
    }
]
```

Inclusions apart from code files:

1. Dockerfile
2. Make file