

## Assignment: Dockerizing your microservices

In this assignment you will create docker images for the Catalog, Inventory and Trading microservices to get them ready to run anywhere.

You can use the table below as a reference for the versions, ports and tags you should be using for each microservice docker image.

Microservice	Version	Port	Tag
Catalog	1.0.2	5000	play.catalog:1.0.2
Inventory	1.0.2	5004	play.inventory:1.0.2
Trading	1.0.0	5006	play.trading:1.0.0

### Create a Dockerfile

For each microservice:

1. Generate and customize a Docker file
  - ❖ Remember to move the Dockerfile to the root of the repo
  - ❖ Make sure you use the correct http port in ASPNETCORE\_URLS
  - ❖ Remember that Trading has no Contracts project

### Build the Docker image

For each microservice:

2. Update the README file with instructions on how to build the docker image.
3. Build the docker image

### Run the Docker image

For each microservice:

4. Update the README file with instructions on how to run the docker image.
  - ❖ You can skip the IdentitySettings\_\_AdminUserPassword env var, which is only needed by the Identity microservice
5. Run the docker container and verify the application starts
  - ❖ Ensure your MongoDB and RabbitMQ containers are up and running.
  - ❖ Getting an "Application started" message in the VS Code console output is good enough for now.
  - ❖ If you try to query the microservice REST API from Postman you will notice that it won't work because, even if you provide the correct access token, your microservice is not ready to reach the Identity microservice to verify the access token. You will address this in a future module.
6. Commit and push your changes.