

Tickets Service

Route	Method	Body	Goal
/api/tickets	GET	-	Retrieve all tickets
/api/tickets/:id	GET	-	Retrieve ticket with specific ID
/api/tickets	POST	{ title: string, price: string }	Create a ticket
/api/tickets/:id	PUT	{ title: string, price: string }	Update a ticket

Tickets Service

Express App

MongoDB for Tickets

Tickets Collection

{ title, price, userId }

{ title, price, userId }

{ title, price, userId }

```
graph TD; A[Create package.json, install deps] --> B[Write Dockerfile]; B --> C[Create index.ts to run project]; C --> D[Build image, push to docker hub]; D --> E[Write k8s file for deployment, service]; E --> F[Update skaffold.yaml to do file sync for tickets]; F --> G[Write k8s file for Mongodb deployment, service];
```

Create package.json, install deps

Write Dockerfile

Create index.ts to run project

Build image, push to docker hub

Write k8s file for deployment, service

Update skaffold.yaml to do file sync for tickets

Write k8s file for Mongodb deployment, service

```
graph TD; A[Create package.json, install deps] --> B[Write Dockerfile]; B --> C[Create index.ts to run project]; C --> D[Build image, push to docker hub]; D --> E[Write k8s file for deployment, service]; E --> F[Update skaffold.yaml to do file sync for tickets]; F --> G[Write k8s file for MongoDB deployment, service];
```

Create package.json, install deps

Write Dockerfile

Create index.ts to run project

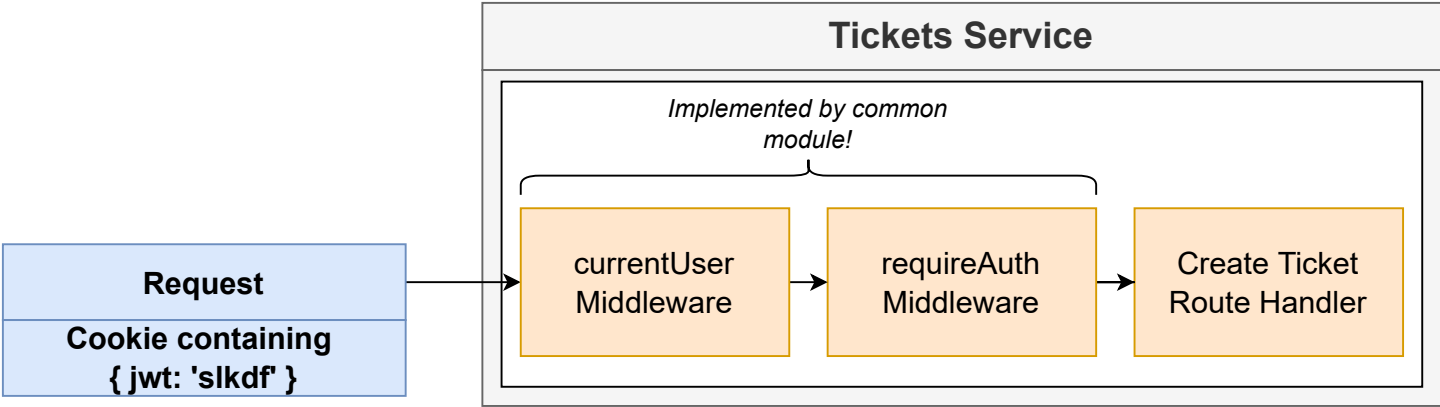
**Copy from
auth service to
save time!**

Build image, push to docker hub

Write k8s file for deployment, service

Update skaffold.yaml to do file sync for tickets

Write k8s file for MongoDB deployment, service



TicketAttrs

title	string
price	number
userId	string

*Properties that are
required to build a
new Ticket*

TicketDoc

title	string
price	number
userId	string

*Properties that a
Ticket has*

TicketModel

build	(attrs) => Doc
-------	----------------

*Properties tied to
the Model*