

A solution for Account Balance based on a Event-Driven architecture with DDD and CQRS. The full solution contains a Account Balance Web API which receives Commands and Queries and produces events and returns JSON. There is a Consoumer App that reads the Event Stream and do a projection on a MongoDB database. The Producer API is behind a security layer with Bearer Authentication and is necessary to run the Auth API to get the JWT.

Requirements

- [Visual Studio 2017 + Update 3](#)
- [.NET SDK 2.0](#)
- [Docker](#)

Environment setup

- Run the `./up-kafka-mongodb.sh` to run Kafka and MongoDB as Docker Containers. Please wait until the ~800mb download be complete.

```
$ ./up-kafka-mongodb.sh
Pulling mongodb (mongo:latest)...
latest: Pulling from library/mongo
Digest: sha256:2c55bcc870c269771aeade05fc3dd3657800540e0a48755876a1dc70db1e76d9
Status: Downloaded newer image for mongo:latest
Pulling kafka (spotify/kafka:latest)...
latest: Pulling from spotify/kafka
Digest: sha256:cf8f8f760b48a07fb99df24fab8201ec8b647634751e842b67103a25a388981b
Status: Downloaded newer image for spotify/kafka:latest
Creating setup_mongodb_1 ...
Creating setup_kafka_1 ...
Creating setup_mongodb_1
Creating setup_mongodb_1 ... done
```

- Check if the data layer is done with the command:

```
$ docker images
```

REPOSITORY	TAG	IMAGE ID	CREATED	SIZE
mongo	latest	d22888af0ce0	17 hours ago	361MB
spotify/kafka	latest	a9e0a5b8b15e	11 months ago	443MB

Running with dotnet commands

How to run the Bearer Authencation API

1. Run the command: `dotnet run` At `source\BearerAuthAPI` folder.

```
$ dotnet run
Using launch settings from
D:\git\myaccountbalanceapi\source\BearerAuthAPI\BearerAuthAPI.Infrastructure\Properties\launchSettings.json...
Hosting environment: Development
Content root path: D:\git\myaccountbalanceapi\source\BearerAuthAPI\BearerAuthAPI.Infrastructure
Now listening on: http://localhost:15878
Application started. Press Ctrl+C to shut down.
```

2. Navigate to the Kestrel URL and navigate to swagger (eg. <http://localhost:15878/swagger>).
3. Post the following credentials:

```
{
  "username": "ivanpaulovich",
  "password": "mysecret"
}
```

4. Store the Bearer Token.

```
{
  "token":
  "eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJqdGkiOiJhYzA4MmE3OS1lMWY3LTQ4MTktYmU1Mi1hOTQwMTBkM2VjZTciLCJzdzIiOiJzdHJpbmciLCJleHAiOjE1MTI0IiwiaXNja3kiOiJ1b3RlciJ9.eyJ0eXAiOiJKV1QiLCJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJqdGkiOiJhYzA4MmE3OS1lMWY3LTQ4MTktYmU1Mi1hOTQwMTBkM2VjZTciLCJzdzIiOiJzdHJpbmciLCJleHAiOjE1MTI0IiwiaXNja3kiOiJ1b3RlciJ9",
  "expiration": "2017-12-05T13:03:08Z"
}
```

How to run the Consumer API

1. At `source\MyAccountBalanceAPI\MyAccountAPI.Consumer.Infrastructure` folder, update the `appsettings.json` with the appropriate connections strings or leave with the default values:

```
{
  "MongoDB": {
    "ConnectionString": "mongodb://10.0.75.1:27017",
    "Database": "MyAccountAPIv05"
  },

  "ServiceBus": {
    "ConnectionString": "10.0.75.1:9092",
    "Topic": "MyAccountAPIv05"
  }
}
```

2. Run the command `dotnet run` at `source\MyAccountBalanceAPI\MyAccountAPI.Consumer.Infrastructure` folder

```
$ dotnet run
11/5/2017 11:17:20 AM Waiting for events..
11/5/2017 11:18:20 AM Waiting for events..
11/5/2017 11:19:20 AM Waiting for events..
11/5/2017 11:20:20 AM Waiting for events..
11/5/2017 11:21:20 AM Waiting for events..
11/5/2017 11:22:20 AM Waiting for events..
```

How to run the Producer API

The screenshot shows the Swagger UI for 'My Account API'. The browser address bar is 'localhost:14398/swagger/'. The Swagger logo is on the left, and the API URL 'http://localhost:14398/swagger/v1/swagger.json' is in the center. An 'Authorize' button is on the right. A blue arrow points to this button with the text '1) Click the authorize button'.

Below the header, the API title 'My Account API' and subtitle 'The My Account Service HTTP API' are shown. The left sidebar lists API endpoints under 'Accounts' and 'Customers'. The main area shows a list of operations for 'Accounts', including 'Deposit from an account', 'Withdraw from an account', 'Close an account', 'List all accounts', 'Get an account balance', and 'List all customers'.

An 'Available authorizations' dialog box is open in the center. It has a title 'Available authorizations' and a subtitle 'Api key authorization'. The text inside says 'Please insert JWT with Bearer into field name: Authorization in: header'. A blue arrow points to the input field containing 'bearer eyJhbGciOiJIUzI1Ni...' with the text '2) Type the text 'bearer' followed by the JWT Token.' Below the input field are 'Authorize' and 'Cancel' buttons. To the right of the dialog, the text 'For example: bearer eyJhbGciOiJIUzI1Ni...' is shown.

1. At `source\MyAccountBalanceAPI\MyAccountAPI.Producer.Infrastructure` folder, update the `appsettings.json` with the appropriate connections strings or leave with the default values:

```
{
  "MongoDB": {
    "ConnectionString": "mongodb://10.0.75.1:27017",
    "Database": "MyAccountAPIv05"
  },

  "ServiceBus": {
    "ConnectionString": "10.0.75.1:9092",
    "Topic": "MyAccountAPIv05"
  }
}
```

2. Run the command `dotnet run` at the `source\MyAccountBalanceAPI\MyAccountAPI.Producer.Infrastructure` folder.

```
$ dotnet run
Using launch settings from
D:\git\myaccountbalanceapi\source\MyAccountBalanceAPI\MyAccountAPI.Producer.Infrastructure\Properties\launchSettings.json...
Hosting environment: Development
Content root path: D:\git\myaccountbalanceapi\source\MyAccountBalanceAPI\MyAccountAPI.Producer.Infrastructure
Now listening on: http://localhost:14398
Application started. Press Ctrl+C to shut down.
```

2. Navigate to the Kestrel URL and navigate to swagger (eg. <http://localhost:14398/swagger>). Follow a few samples requests:

POST api/Customers

```
{
  "pin": "08724050601",
  "name": "Ivan Paulovich",
  "initialAmount": 1600
}
```

returns

```
{
  "customerId": "f5ea8e65-d9e1-4b33-aad5-b5ca022bc183",
  "ssn": "08724050601",
  "name": "Ivan Paulovich",
  "accountId": "f78c4764-5df2-4ad9-a6c8-210871e03313",
  "currentBalance": {
    "value": 1600
  }
}
```

GET api/Customers will returns

```
[
  {
    "_id": "f5ea8e65-d9e1-4b33-aad5-b5ca022bc183",
    "_version": 1,
    "name": {
      "Text": "Ivan Paulovich"
    },
    "pin": {
      "Text": "08724050601"
    }
  }
]
```

GET api/Accounts will returns

```
[
  {
    "_id": "f78c4764-5df2-4ad9-a6c8-210871e03313",
    "_version": 1,
    "currentBalance": {
      "Value": 1600
    },
  },
]
```

```
    "transactions": null,  
    "customerId": "f5ea8e65-d9e1-4b33-aad5-b5ca022bc183"  
  }  
]
```

PATCH /api/Accounts/Deposit

```
{  
  "customerId": "f5ea8e65-d9e1-4b33-aad5-b5ca022bc183",  
  "accountId": "f78c4764-5df2-4ad9-a6c8-210871e03313",  
  "amount": 350  
}
```

PATCH /api/Accounts/Withdraw

```
{  
  "customerId": "f5ea8e65-d9e1-4b33-aad5-b5ca022bc183",  
  "accountId": "f78c4764-5df2-4ad9-a6c8-210871e03313",  
  "amount": 670  
}
```

Running with Visual Studio 2017

Run the projects `BearerAuthAPI.Infrastructure` , `MyAccountAPI.Consumer.Infrastructure` and `MyAccountAPI.Producer.Infrastructure` .