|  |  |  |  |
| --- | --- | --- | --- |
| **Resource** | **Path** | **Action** | **Authorisation** |
| Client | api/clients | Get, post | Admin |
| Client | api/Clients/{id} | Get, put, delete | Admin |
| Currency | api/Currencies | Get, post | All |
| Currency | api/Currencies/{id} | Get, put, delete | All |
| Wallet | api/ Clients/{id}/Wallets |  | Admin, Client (customer, vendor) |
| Wallet | api/ Clients/{id}/Wallets/{id} |  | Admin, Client (customer, vendor) |
| Transaction | api/Clients/{id}/Transactions |  | Admin, Client (customer, vendor) |
| Transaction | api/Clients/{id}/Transactions/ |  | Admin, Client (customer, vendor) |
| Transaction | api/Clients/{id}/Transaction/{id}/Bargains |  |  |
| Transaction | api/Clients/{id}/Transaction/{id}/Bargain/{id} |  |  |
| Offer | api/Clients/{id}/Offers |  | Admin, Client (vendor) |
| Offer | api/Clients/{id}/Offers/{id} |  | Admin, Client (vendor) |
| Bargain | api/Clients/{id}/Bargains |  | Admin, Client (customer, vendor) |
| Bargain | api/Clients/{id}/Bargain/{id} |  | Admin, Client (customer, vendor) |
| FiatAccount | api/Clients/{id}/FiatAccounts |  | Admin, Client (customer | vendor) |
| FiatAccount | api/Clients/{id}/FiatAccount/{id} |  | Admin, Client (customer | vendor) |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

**Resources**

Currency

Client

Wallet

Transaction

Offer

Bargain

FiatAccount

Statup.cs

Add the json formatter contract resolver to ensure returned json property names are camel case. Useful if the endpoint consumer is java script.

Each entity has its own controller

Each entity has its own route configuration in startup.cs

A model factory is useful, better than auto mapper in passing URI’s etc.

Each entity has its own ‘Model’

Identifiers

Add a url property to all the ‘Models’. In the model factory add the string url

Create a constructor of the model factory class. Add a url helper

The ctor consumes an HttpRequestMessage as parameter, i.e. needs to know the request Url…

In each ‘EnityModel’

Use String.Format (http: …/{}, entity.id}

Url is useful when versioning comes in to play.

Create an Identity service class

For entities that use/need identity to access resources, inject it

Then get the current user (\_identityServices.CurrentUser

HrttpResponsemEssage

This should be the response type for Api requests,

It wraps the results and you can pass an Http status code (ok, not found)

Gives proper not found response when a requested resource is nul