**JeseryRestSecurity –**

Jersey based server;

Uses eacore database;

Start it by: RUN AS –Run on Server

If a Web page pops up – IGNORE

**RestClientSecurity - [DEPRECATED]**

This will TEST JeseryRestSecurity.

This is started as a Java Application with Main routine…

It does a “findAll” [read]; findOne [read (1)];

Creates a product & saves it. Then does another findall().

If you want see security authorization – change

edu.mum.rest.RestHttpHeader change username/password to guest instead of admin

**MemberMVC –**

The “standalone” MVC Web Application – BUT just presentation layer.

EXPECTS MemberRest to be running for service/persistence layers.

Alternatively can use JerseyRestSecurity for Products [see README.TXT]

Start it by: RUN AS –Run on Server

Will bring up Welcome web page.

**MemberRest –**

The Rest Server containing the Service & Persistence Tiers.

It uses entiremember database

Start it by: RUN AS –Run on Server

NOTE: if you want to use JeseryRestSecurity with MemberMVC for product – you need to also start this server[MemberRest] to run MemberMVC successfully. You also need to set the property productRest.url in

MemberMVC /application.properties to reference JerseyRestSecurity.

**MemberRestJSON –**

Like MemberRest ONLY has the @Json annotations placed to test variations.

The Rest Server containing the Service & Persistence Tiers.

It uses entiremember database

Start it by: RUN AS –Run on Server

**MemberRestTest [DEPRECATED]**

Test harness for MemberRest OR MemberRestJSON –

Initially set up to test product/category many to many bidirectional

With MemberRestJSON.

NOTE: Bidirectional compatability- Jason annotations need to be aligned in the domain models on the server & the client