**EDI:---**

#1> Complete customer modules for all operations and Itegrations :--

#2>:-- Generate Passwords, Token using “CodeUtil” Utility class:--

#3>Encode details and Store in DB by using “CoDecUtil” Utility class:-

#4>Actual Details Send to Email ((un/email), pwd, token) using “CommonUtil”utility class:--

#5>Configure FC-2 (SpringServlet) also called as RestFC in “web.xml”:--

=>“SpringServlet” = ServletContainer + Spring in web.xml using directory match URL Pattern

#6>Define Authentication manager and Service Provider class with Spring and Rest Annotations in “ItemServiceProvider” class in com.app.provider:--

#7>Define “JsonUtil” class, which converts JSON --->Object and Object --->JSON using JACSON API:--

#8>For Module Implementation logic, must be written in IL (Itegration Layer) in “ItemController” class:--

**Order of Code :--**

1>Customer Module :--

a>Registration

b>Password & Token Generation

c>Encoding and Decoding

d>Sending Email

2>Item Module:--

a>Registration

b>Integration with Customer

c>Export charts & Bar

3>JsonUtil :--

a>Object to JSON convertion

b>JSON to Object

4>CustomerValidation :--

a>ValidateUser

b>Validate password & Token

c>Validate Type

5>Rest Integration:--

a>Enable SpringServlet

b>Make Spring configuration global

6>ItemController:--

a>Item save Process

b>Basic Validation

c>User Validation

d>Password Token validation

e>Type (Seller, Customer)

f>Json to Object creation (Item)

g>save Item

h>return final Response

Rest Integration :--

1>configure one FC named as “SpringServlet” with directory match pattern in “web.xml”.

2>After DispatcherServlet configuration, define one provider class which must be configured using Rest Annotations & Spring Annotations.

3>FrontController is designed for spring Integration with RestService as :--

ServletContainer + SpringContainer = SpringServlet

4>SpringServlet will accept request only made to Integration Layer where as DispatcherServlet will accept request made to Presentation Layer.

5>In web.xml, Both FC’s must be configured using directory match url pattern, But In this case, Spring container fails to identify spring configured filed, Because Two servlets are configured.

For syntax:-- [<servlet-name>]-servlet.xml

=>Here solution is do not follow the Naming convension, follow context-param methodology which load programmer given file.

**Execution Steps:--**

#1>start Application and Goto customer module Registration page.

#2>Register customer, provider valid email Id, app will send cust details like username, password, token to email.

#3>open “PostMan” (chrome://app).

#4>Enter Details like bellow (postman Screen).

POST <http://localhost:2018/VendorApp/rest/item/save> Send

Header

Username : [udaykumar0023@gmail.com](mailto:udaykumar0023@gmail.com)

Password : 909256

Token : agdb

Body:--

{ “itemId”:55, “itemName”:”NOKAI”, “itemCost”: 675.78}