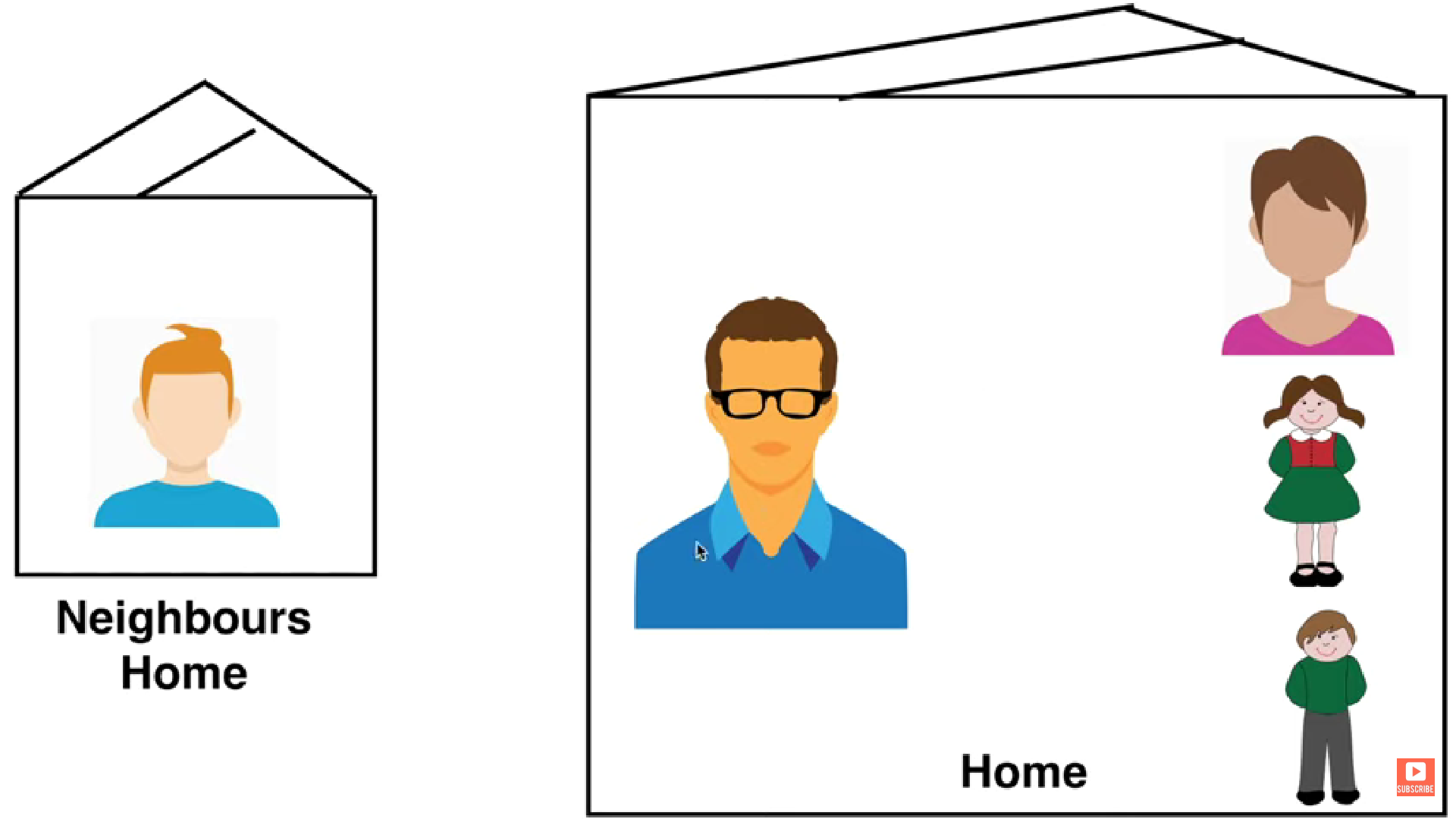
As a part of **Initial Setup**, We have already created a blank **dynamic web project**, added all spring .jar files and configure Tomcat9 server.

In this tutorial, we are going to discuss, run and execute our first project and also we will have a fair idea about the theme of spring mvc. Let’s start discussion,

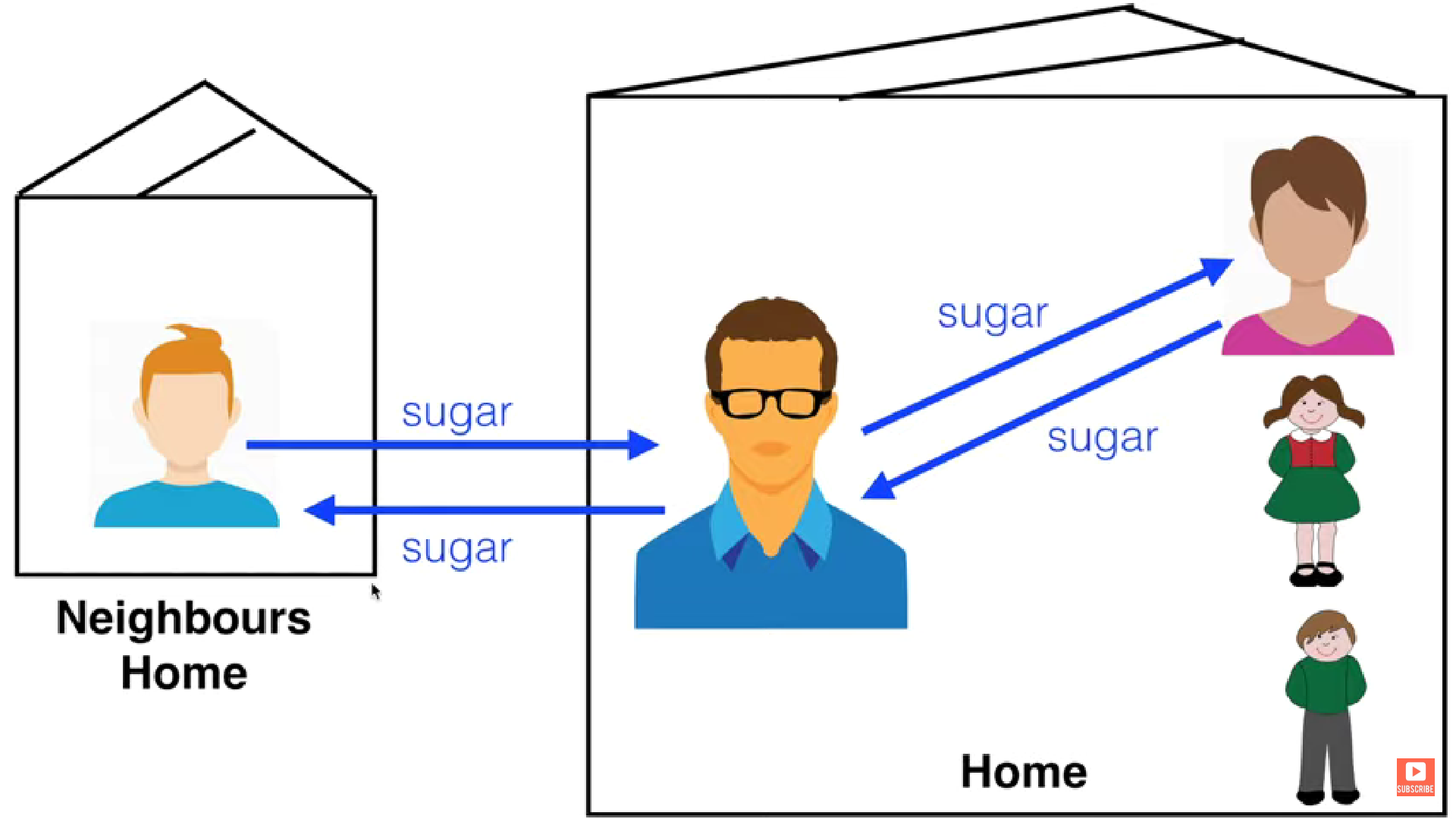
Have a look the below image :



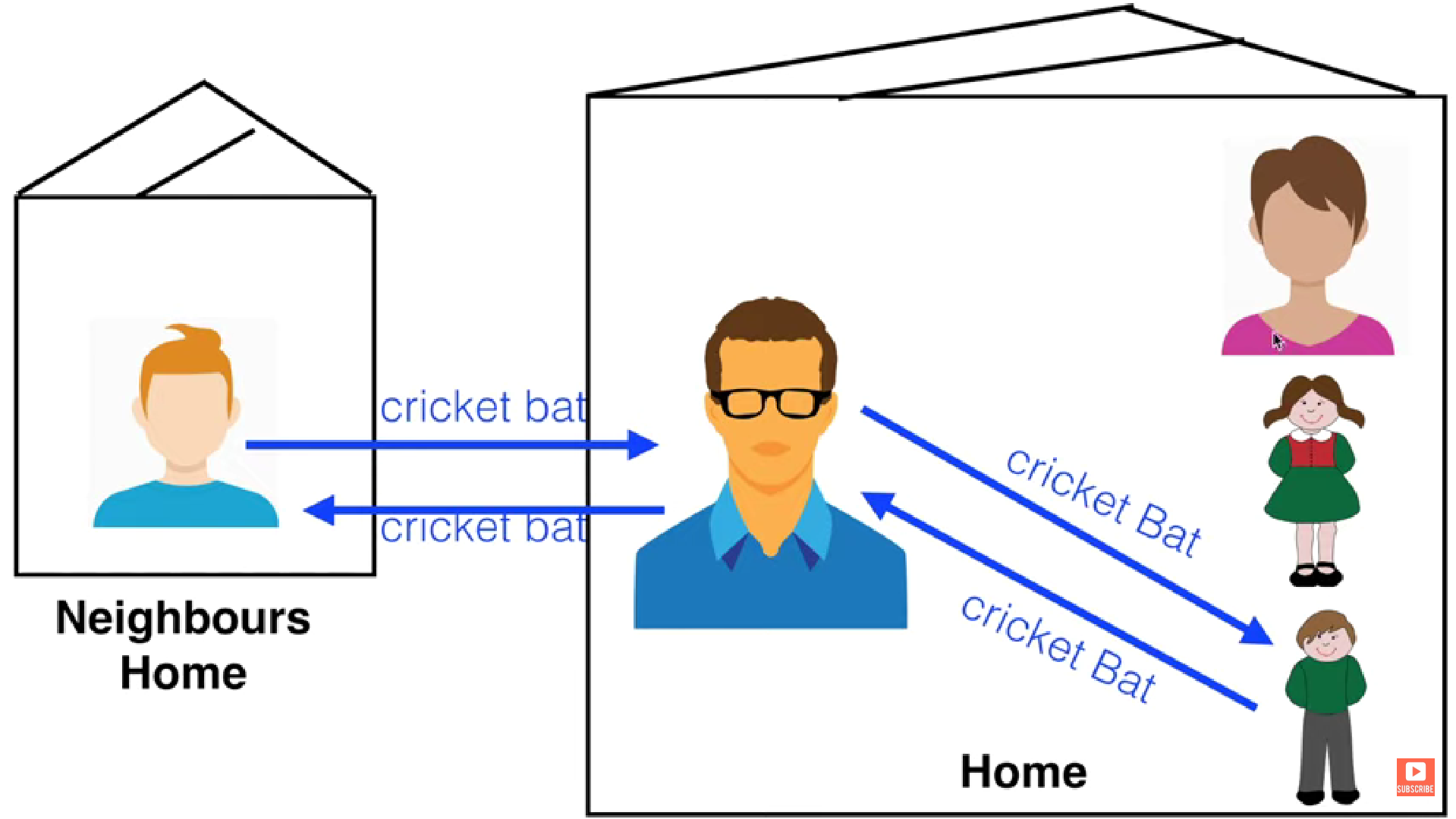
Here, there is a family ( Father, Mother, Son, Daughter ) live in **Home**.

On the other side, this family has a Neighbor.

As expected, the father is the leader of family. Now let’s suppose neighbor needs of sugar. So he will ask for sugar to the leader of family i.e. father. But father doesn’t know where the sugar is, but father knows that Mother know about sugar. So father will ask the mother, get the sugar and give back to the neighbor.

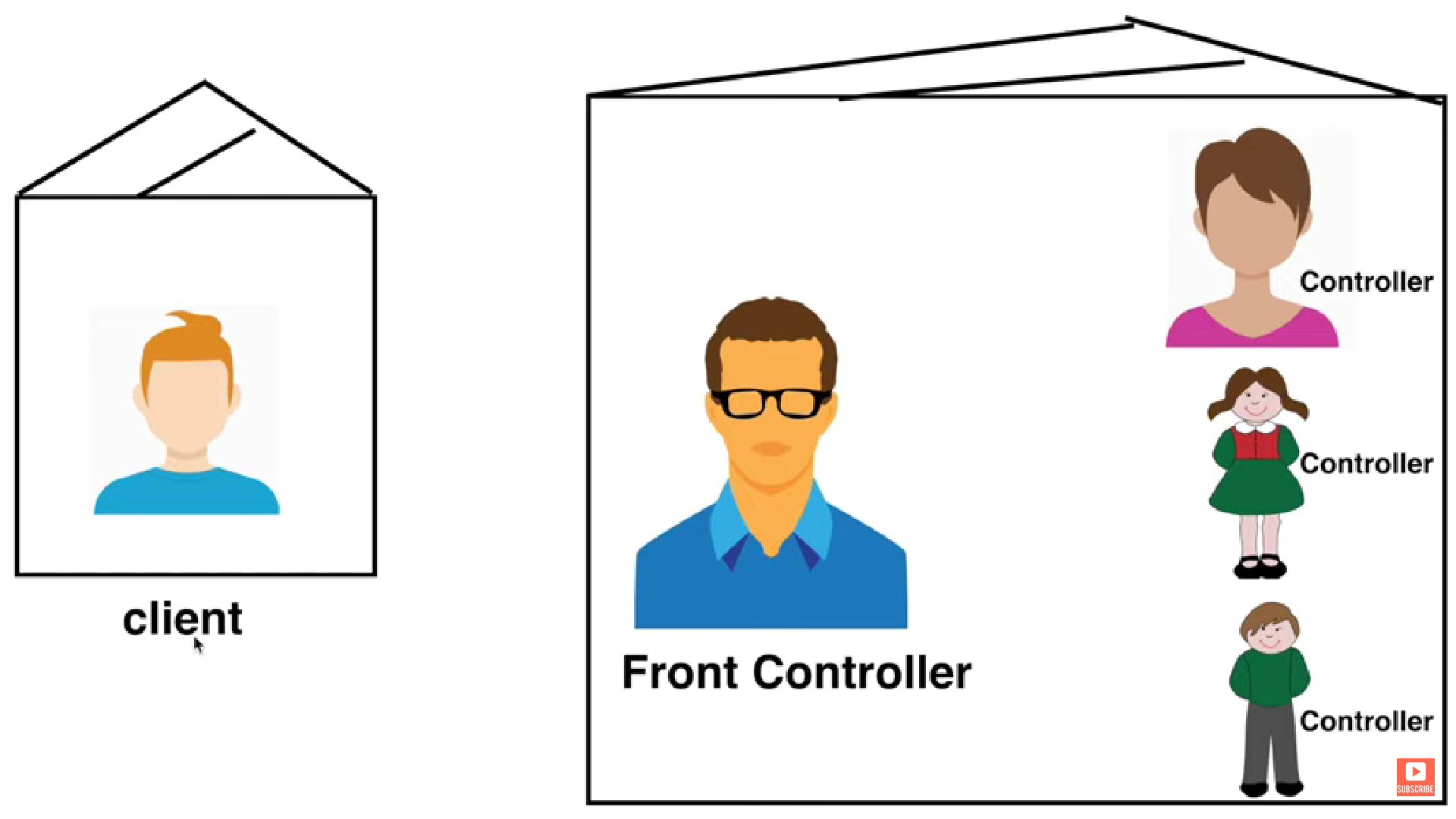


Now again neighbor needs of Cricket Bat. But again father don’t know about cricket bat but father know that my son has cricket bat so father will ask his son, get the cricket bat and give back to neighbor.

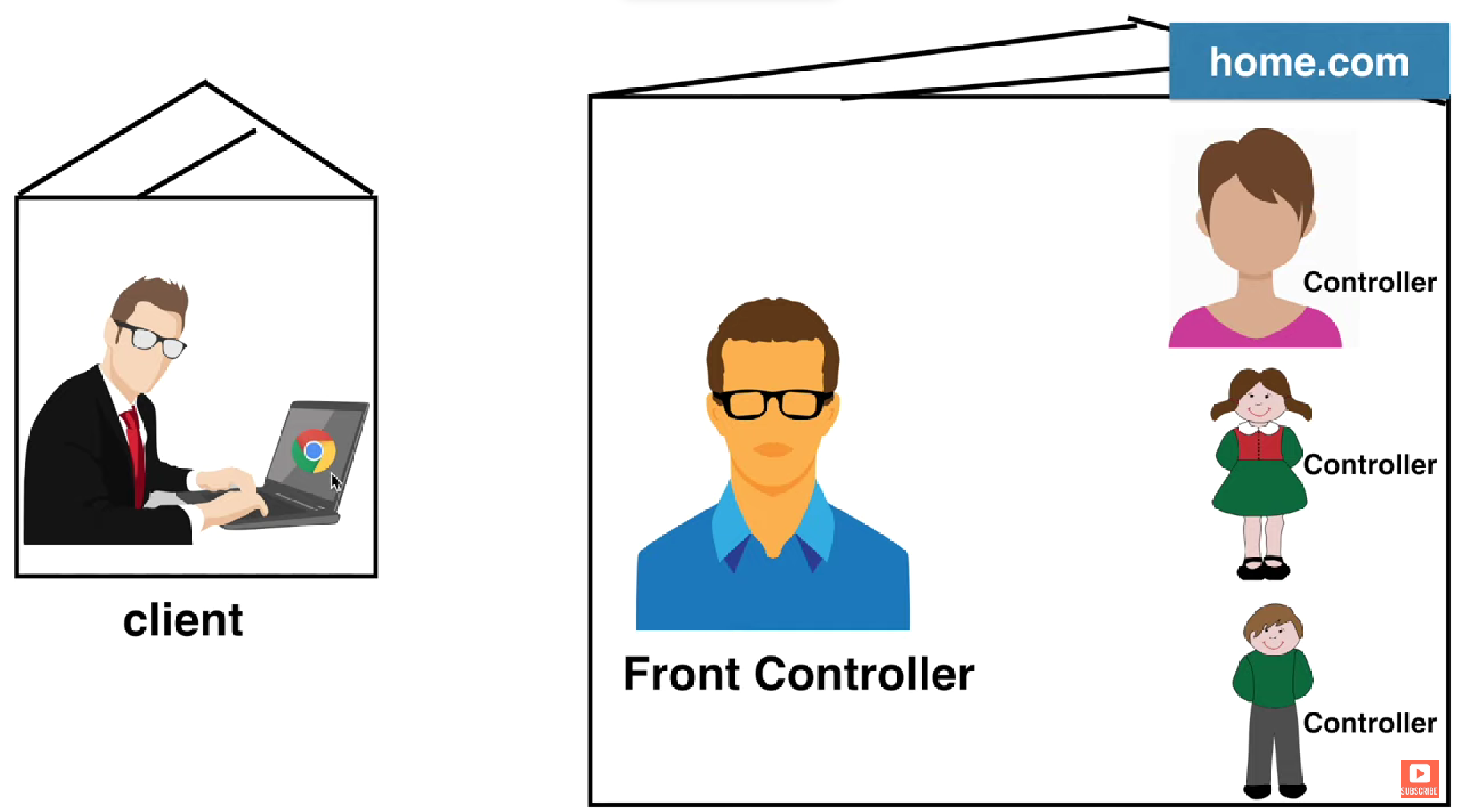


In context of Spring mvc,

* **Neighbour** is **Client**
* **Father** is **Front Controller**
* **Mother, Daughter, Son** is **Controller**

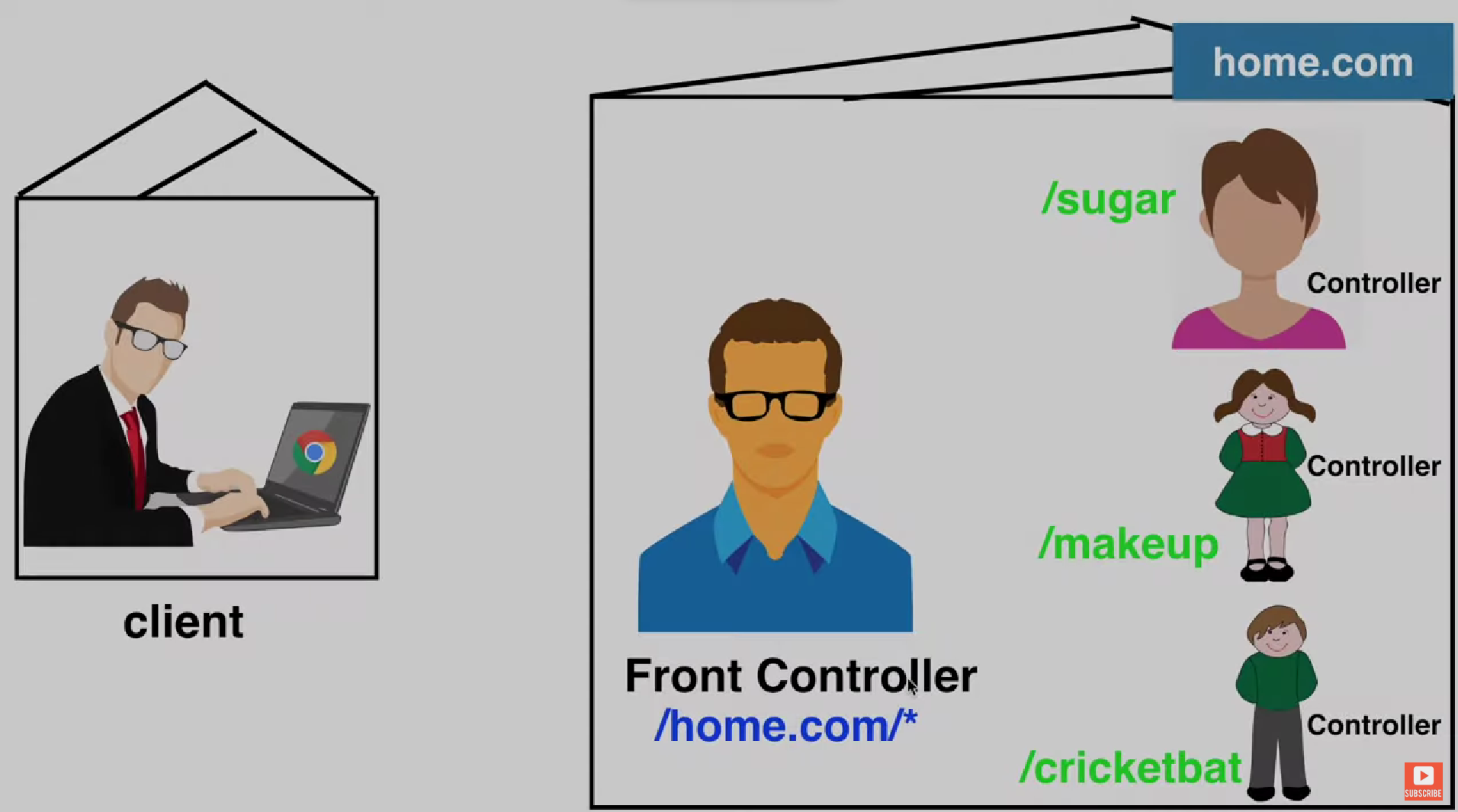


Now, suppose our application name is **home.com** and there is client sitting somewhere want to access our website.



Our controllers are mapped with following urls:

* **Front Controller :** /home.com/\*
* **Mother Controller :** /sugar
* **Daughter Controller :** /makeup
* **Son Controller :** /cricketbat

****

Any request coming to website will be handle first by Front Controller. Suppose client ask for sugar then url will be something like below:

[**/home.com/sugar**](http://www.myweb.com/home.com/sugar)

so as expected the request will handle by Front Controller first because of url containing **/home.com/**and then request will forward to Mother controller because of url also containing **/sugar**

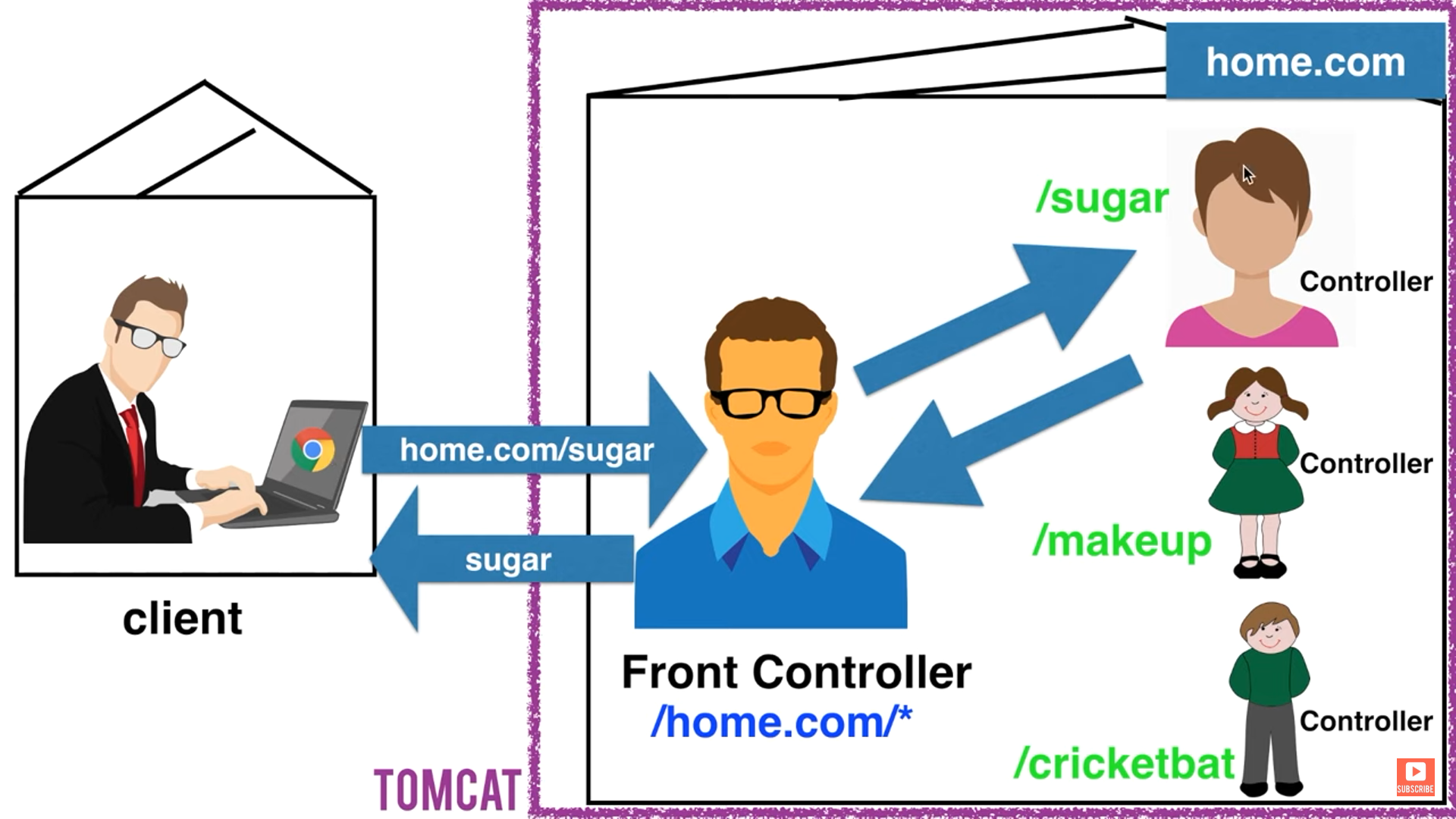
Whatever request start with **/home.com/** will be handle by Our Front Controller.

[**/home.com/sugar**](http://www.myweb.com/home.com/sugar)

[**/home.com/makeup**](http://www.myweb.com/home.com/sugar)

[**/home.com/cricketbat**](http://www.myweb.com/home.com/sugar)

So as we can notice, front controller is not actually doing anything. It is just routing the url and dispatching the request to the appropriate controller.



Now the good thing is that we don’t need to create the **Front Controller.** Spring has already created it for us. Its name is **org.springframework.web.servlet.DispatcherServlet**. We just have to configure it in our **web.xml** file.

<?**xml** version=*"1.0"* encoding=*"UTF-8"*?>

<**web-app** xmlns:xsi=*"http://www.w3.org/2001/XMLSchema-instance"*

xmlns=*"http://xmlns.jcp.org/xml/ns/javaee"*

xsi:schemaLocation=*"http://xmlns.jcp.org/xml/ns/javaee http://xmlns.jcp.org/xml/ns/javaee/web-app\_4\_0.xsd"*

id=*"WebApp\_ID"* version=*"4.0"*>

<**display-name**>home.com</**display-name**>

<**welcome-file-list**>

<**welcome-file**>index.html</**welcome-file**>

<**welcome-file**>index.jsp</**welcome-file**>

<**welcome-file**>index.htm</**welcome-file**>

<**welcome-file**>default.html</**welcome-file**>

<**welcome-file**>default.jsp</**welcome-file**>

<**welcome-file**>default.htm</**welcome-file**>

</**welcome-file-list**>

<**absolute-ordering** />

<**servlet**>

<**servlet-name**>frontcontroller</**servlet-name**>

<**servlet-class**>org.springframework.web.servlet.DispatcherServlet</**servlet-class**>

<**load-on-startup**>1</**load-on-startup**>

</**servlet**>

<**servlet-mapping**>

<**servlet-name**>frontcontroller</**servlet-name**>

<**url-pattern**>/home.com/\*</**url-pattern**>

</**servlet-mapping**>

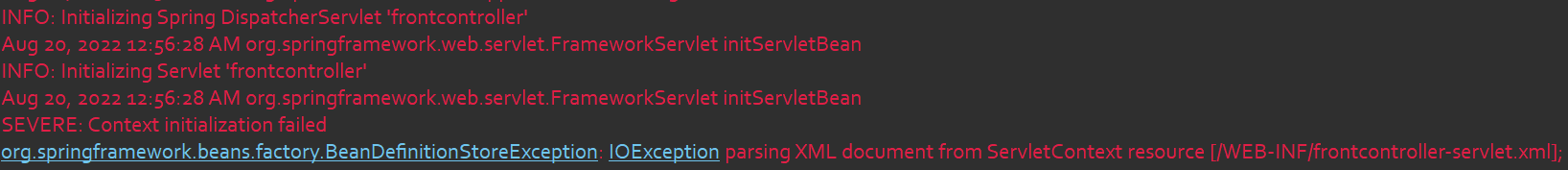
</**web-app>**

So it is the basic structure of our **web.xml** file with mapping of front-controller.

* <**absolute-ordering** />
  + We will talk later on this tag for now just add it.
* <**servlet**>
  + This tag simply use to tell which class will act as Front Controller and alias the our Front controller class for further use.
* <**servlet-name**>
  + In this tag, we alias the front controller.
* <**servlet-mapping**>
  + In this tag, we tell which class act as Front Controller.
* <**load-on-startup**>
  + This tag is telling spring that load the Front Controller as application start. If this tag is absent then by default spring load the Front Controller when the first request arrived to application.
* <**servlet-mapping**>
  + Here we define the url for our front controller.
* <**servlet-name**>
  + Here we tell the aliased name of our front controller.
* <**url-pattern**>
  + Inside this tag, we set the url for our front controller on which it is going to accept client requests.

Once if you have configured front controller with the above steps then next task is to run your project.

Now if your Front Controller configured successfully then you will get the below message :



Yes, we will get an exception showing in the image. The reason of exception is that spring trying to find one more .xml file with the name **frontcontroller-servlet.xml** in the **WEB-INF** directory**.**

To get rid of this exception we just have to add this file inside our **WEB-INF** directory**.** The file name should be same as Front Controller name ended up with **-servlet.xml**

**frontcontroller-servlet.xml :**

<**beans** xmlns=*"http://www.springframework.org/schema/beans"*

xmlns:xsi=*"http://www.w3.org/2001/XMLSchema-instance"*

xmlns:context=*"http://www.springframework.org/schema/context"*

xmlns:mvc=*"http://www.springframework.org/schema/mvc"*

xsi:schemaLocation=*"http://www.springframework.org/schema/beans*

*http://www.springframework.org/schema/beans/spring-beans.xsd*

*http://www.springframework.org/schema/mvc*

*http://www.springframework.org/schema/mvc/spring-mvc.xsd*

*http://www.springframework.org/schema/context*

*https://www.springframework.org/schema/context/spring-context.xsd "*>

</**beans**>

Did you remember something. Yes this file is as similar as the **config.xml** file we have created lots of time while studying **spring core.**

To know what is the use of this file in here. Just wait for later tutorial you will automatically get the idea.

After adding this file your project should run and the front controller should initialize successfully with no error.

