

What Spring MVC does?

Spring MVC will help you to develop java J2EE (web) applications easily.

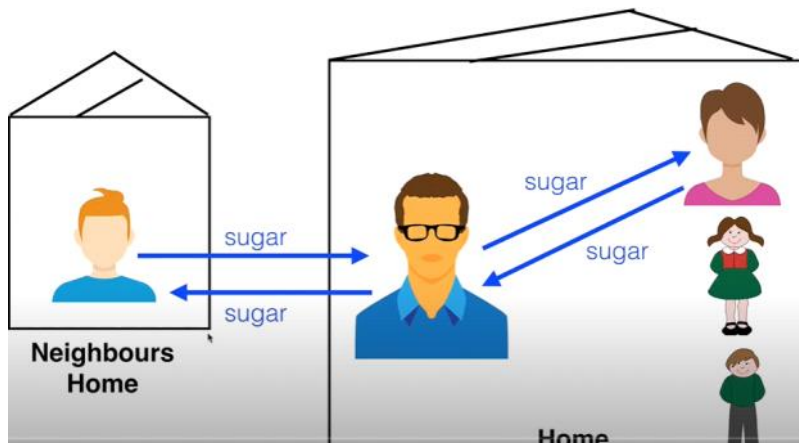
But the **problem** is setting up your project (or working environment) manually by doing a lot of configurations by using **XML** etc.

What Spring boot does?

Spring boot helps us to wrap all spring components in a convenient way where no external XML configuration is needed.

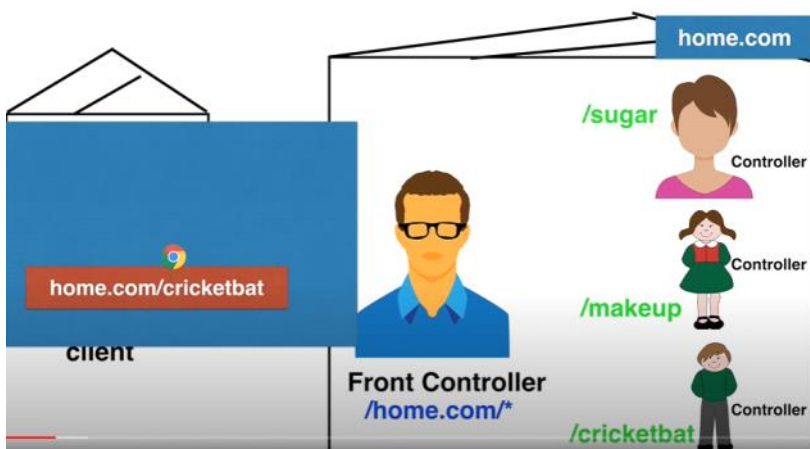
so the spring boot wraps spring MVC itself.

Front CONTROLLER



Any request come outside the home dad knows the available resource so where to dispatch that incoming request

Every request come it is handle by the front controller. He will accept all the request which has url starting with home.com
Routing to other controller



```
<servlet>
  <servlet-name>dad-frontcontroller-dispatcher</servlet-name>
  <servlet-class>org.springframework.web.servlet.DispatcherServlet</servlet-class>
</servlet>
<servlet-mapping>
  <servlet-name>dad-frontcontroller-dispatcher</servlet-name>
  <url-pattern>/home.com/*</url-pattern>
</servlet-mapping>
```

Dispatcher front controller server first point of hitting

Spring will automatically initialize the class having a **@controller** annotation and register that class with the spring container.

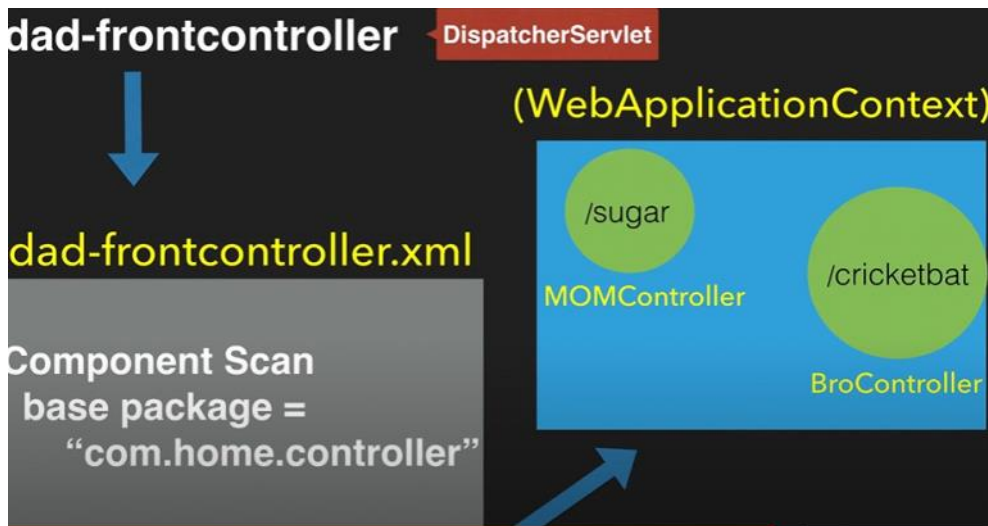
```
@Controller
public class MomController {

  @ResponseBody
  @RequestMapping("/sugar")
  public String giveSugar()
  {
    return "Ok..Here is your sugar".
  }
}
```

@Response body map the value to HTTP

To print on webpage we need to use @Response Body

After initialization framework will look for servlet name dad-controller with appended with .xml



```
@Controller
public class BroController {

    @ResponseBody
    @RequestMapping("/cricketbat")
    public String giveCricketBat()
    {
        return "MRFCricketbat";
    }
}
```

If we write `ResponseBody` annotation then that will write on the web page that string

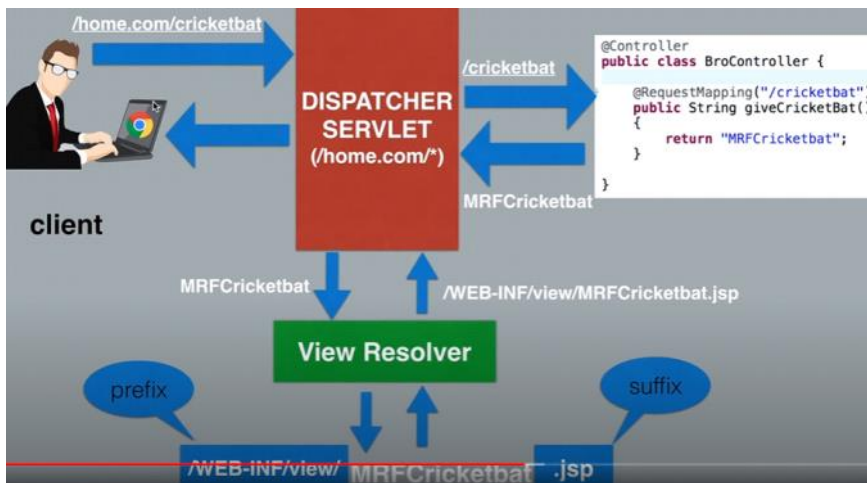
But if we remove the `@ResponseBody` then controller will expect a proper web page of name called `MRFCricketBat` name web page. It is called view name.

/WEB-INF/ view / MRFCricketbat.jsp

View resolver--> webinf will change

.jsp will also change

But page name will not change



```

class InternalResourceViewResolver
{
    private String prefix = "";
    private String suffix = "";

    public void setPrefix(String prefix)
    {
        this.prefix = prefix;
    }
    public void setSuffix(String suffix)
    {
        this.suffix = suffix;
    }
}

InternalResourceViewResolver viewResolver = new InternalResourceViewResolver();
viewResolver.setPrefix("/WEB-INF/view/");
viewResolver.setSuffix(".jsp");

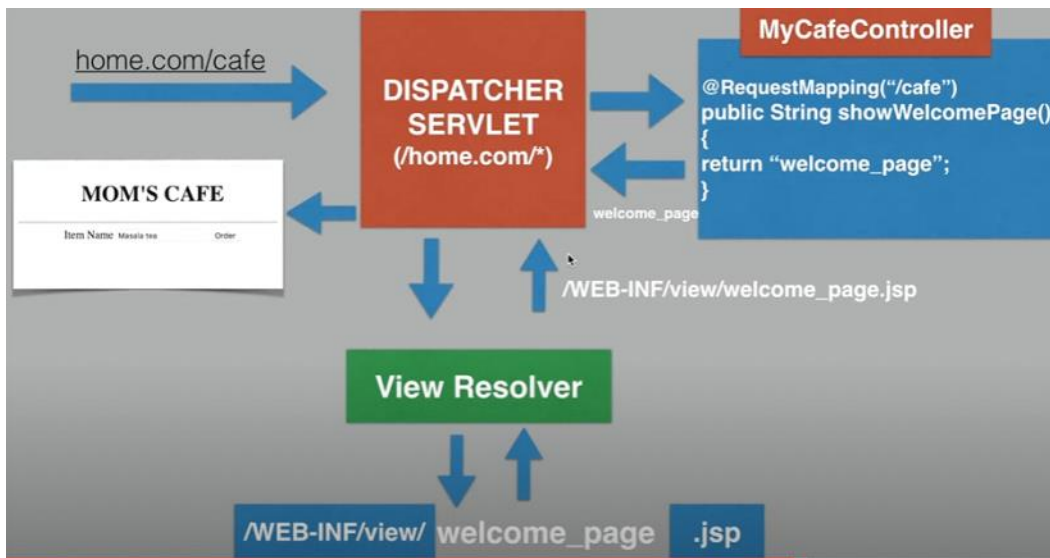
```

If we want to send data or string while returning to web page

```

7 @Controller
8 public class MyCafeControllers {
9
10     @RequestMapping("/cafe")
11     public String showWelcomePage(Model model)
12     {
13         //sending data to view(jsp page)
14         String myName = "Abhilash";
15
16         model.addAttribute("myNameValue", myName);
17
18         return "welcom-page";
19     }
20 }

```



```
@RequestMapping("/processOrder")
public String processOrder(HttpServletRequest request, Model model)
{
    //handle the data received from the user
    String userEnteredValue = request.getParameter("foodType");

    //adding the captured value to the model
    model.addAttribute("userInput", userEnteredValue);

    //set the user data with the model object and send it to view
    return "process-order";
}
```