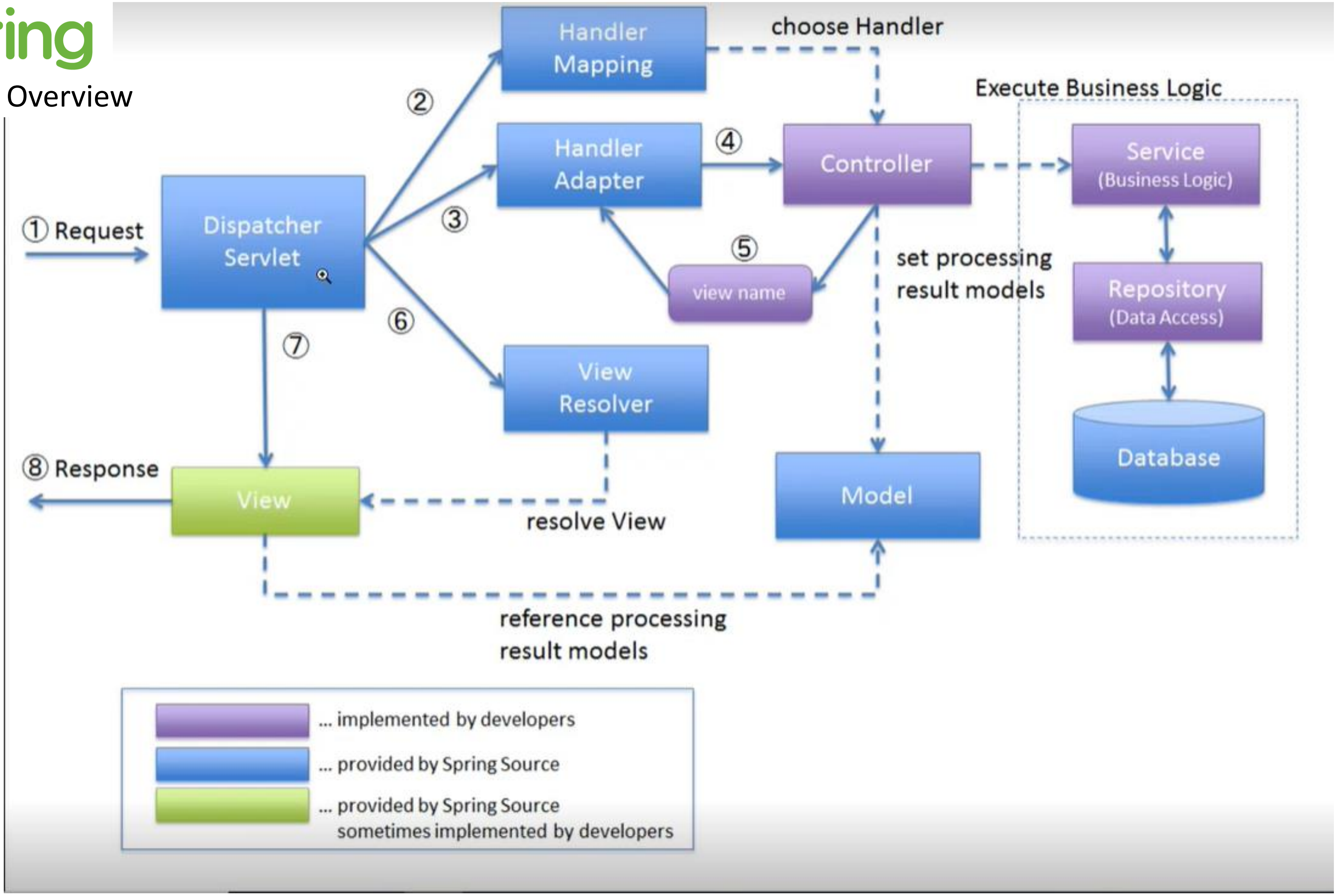
Hooking Up Spring Backend

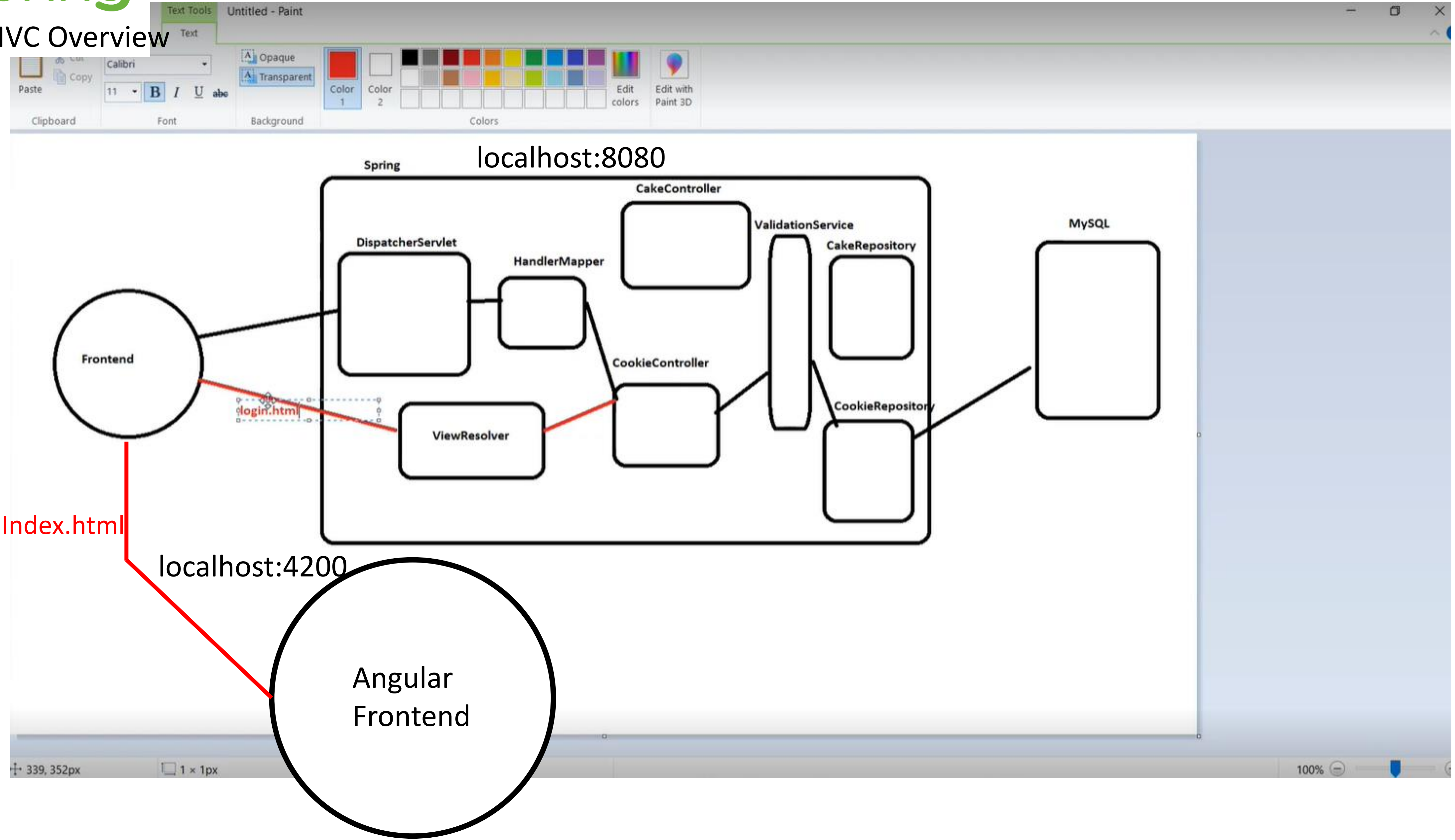
To Angular Frontend

Spring MVC Overview



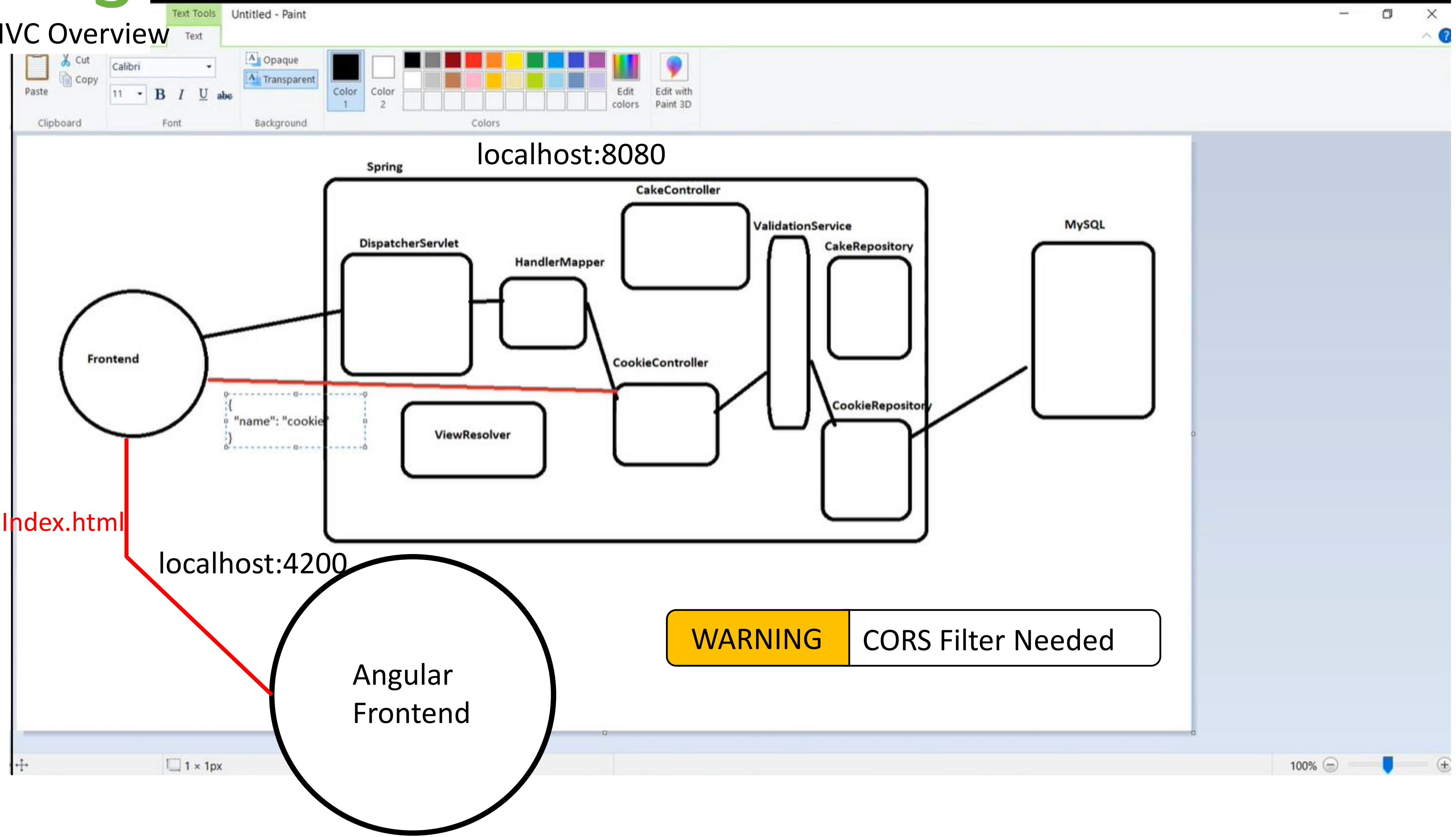
Spring MVC Overview

NOTE: We don't need the ViewHandler bc Angular is supplying index.html



Spring MVC Overview

NOTE: We will be getting data from the backend controller directly



WARNING CORS Filter Needed

Controller CORS Filter Implementation

DogController.java

```
@GetMapping("/dogs")
@CrossOrigin(origins =
             "http://localhost:4200")
@ResponseBody
public List<Dog> findAllDogs() {
          .  .  .
}
```

spring

CORS Filter Global Configuration (alternative)

SpringMvcConfig.java

```
                    . . .
@Bean
public WebMvcConfigurer corsConfigurer() {
    return new WebMvcConfigurer() {
        @Override
        public void addCorsMappings(CorsRegistry registry) {
            registry.addMapping("/dogs").allowedOrigins("http://localhost:4200");
        }
    };
}
                    . . .
```

NOTE: Use global corsConfiguration or controller CrossOrigin NOT BOTH

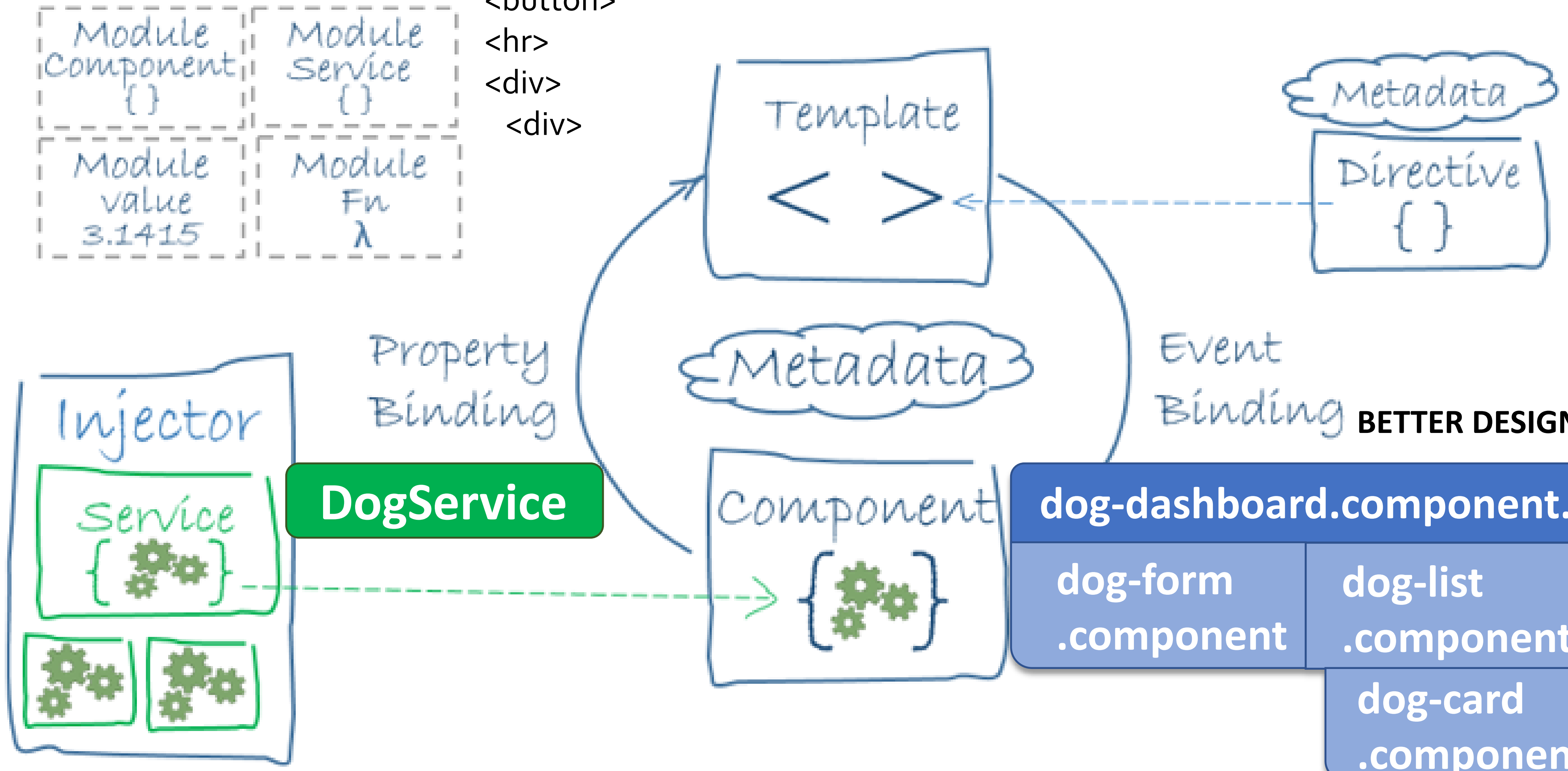
Angular Overview

SIMPLE DESIGN IDEA:

app-component.html

<app-dog-dashboard></app...>

<label>
<input>
<button>
<hr>
<div>
  <div>



DogService

BETTER DESIGN IDEA:

dog-dashboard.component.ts

dog-form .component

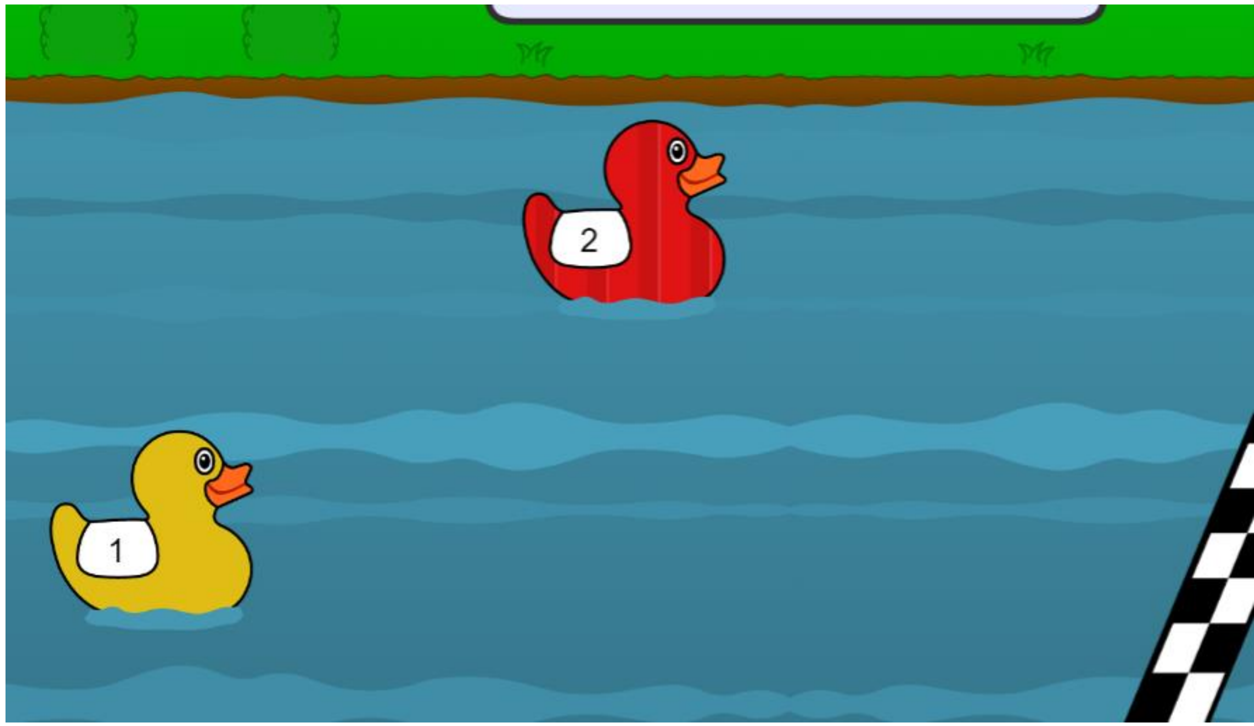
dog-list .component

dog-card .component

Steps

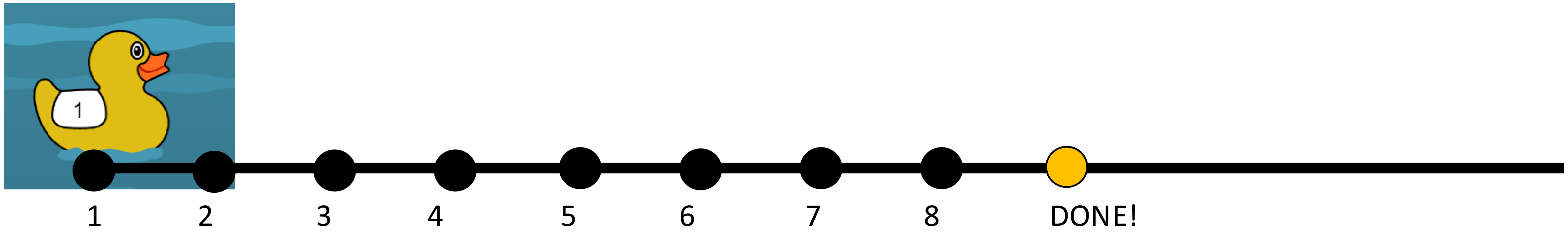
Angular Overview

1. CORs Filter
2. New Angular Project
3. New Angular Component
4. New Angular Service
5. Import HttpClientModule
6. Write getDogs() in dog.service.ts
7. Write getDogs() in component.ts
8. Write HTML with button (click) and list *ngFor
9. Edit getDogs() to take a parameter – update service, component.ts
10. Edit component.html to have <input #breed> <button (call)="getDogs(breed.value)" >



CODE DEMO

Follow Along ☺


1
1
2
3
4
5
6
7
8
DONE!