**SPRINGBOOT TUTORIALS :-**

Singleton scope and prototype scope in springboot - By default spring framework will provide an object without or with creating. even if we need or no need of object.

we have an embedded TOMCAT SERVER in spring frameword by default.

**ANNOTATIONS in Spring boot :-**

@**Component** - as soon as we gave this annotation to the class , the spring container will be able to give the object for that particular class  
**Definition**: @Component is a generic stereotype for any Spring-managed component. It indicates that an annotated class is a Spring bean.

**@AutoWired -** access By using this we will be able to connect different classes and only the object will be instantiated.

**Definition**: @Autowired is used for automatic dependency injection. It allows Spring to resolve and inject collaborating beans into your bean.

@Autowired

private Laptop laptop; //Laptop is a class name , laptop is the object we created using autowired.

**@Scope(value= “prototype”)** - By using this we wont get anobject by default( spring framework will not create an object by default) , so only if we create object by our own we will be able to access it.

**@Service :-**

**@Repository :-**

**@Controller :-**

### **@RestController :-**

**@Configuration :-**

**@SpringBootApplication :-**

**Definition**: @SpringBootApplication is a convenience annotation that combines @Configuration, @EnableAutoConfiguration, and @ComponentScan. It is used to mark a configuration class that declares one or more @Bean methods and triggers auto-configuration and component scanning.

**@Bean :-**

**@GetMapping :-**

**@PostMapping:-**

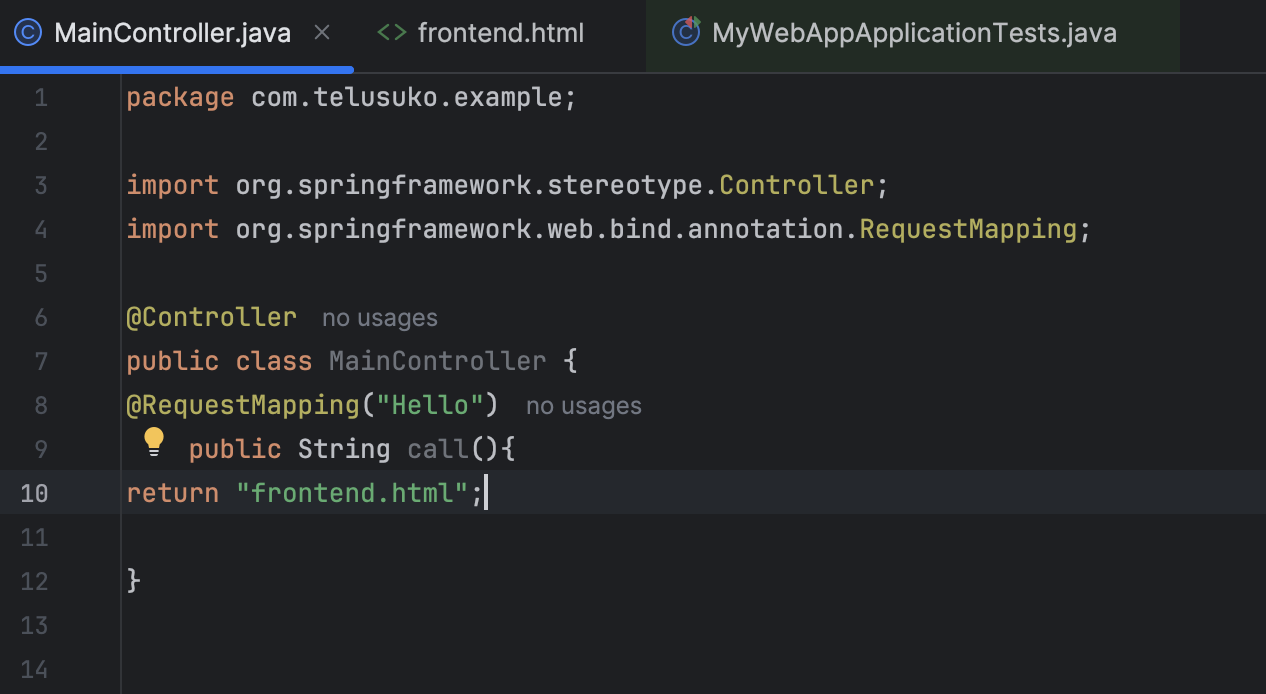
**@RequestMapping:-**

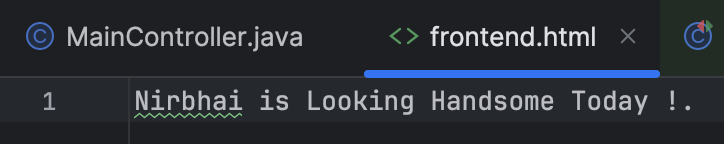
**@ResponseBody:-**

Spring container

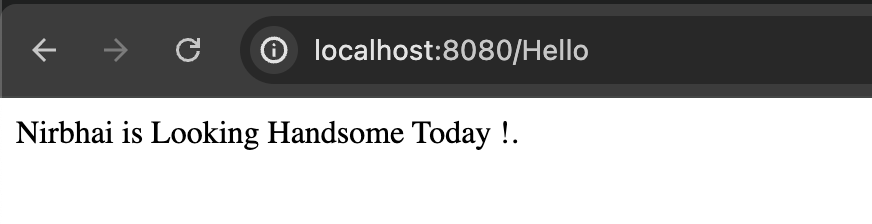
Spring bean -

Application.properties - Configuration prefix and sufix

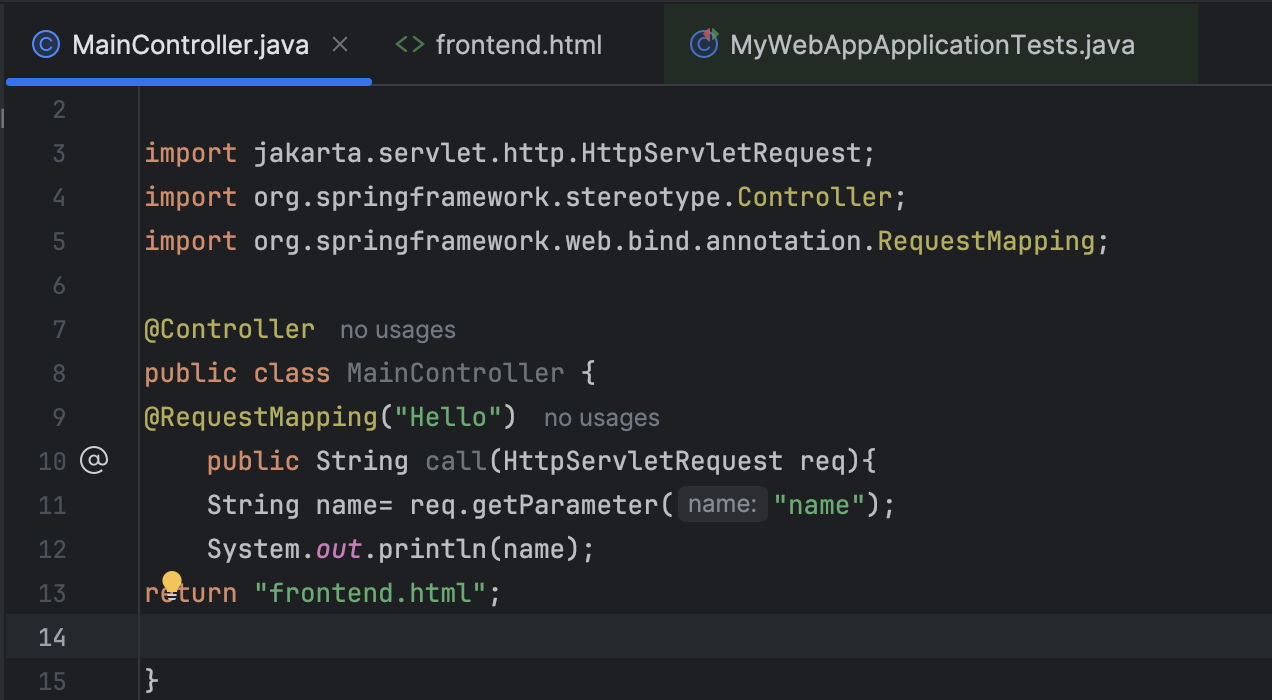




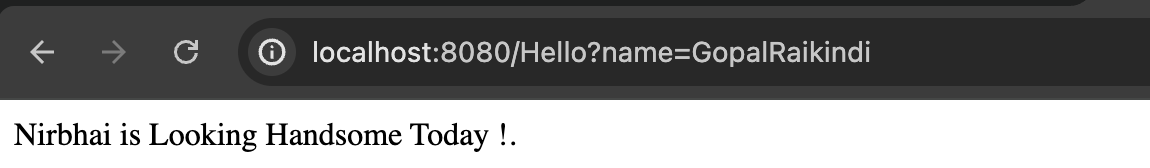
OUTPUT :-



SENDING DATA THROUGH REQUEST PARAMETER :-



INPUT :-



OUTPUT :-

