## Spring Boot

Features:

1. Create Stand-Alone Spring Applications (Microservices)
2. Embed Tomcat, Jetty, Undertow directly (no need for WAR file)
3. Opinionated 'Starter' dependencies
4. Auto configure third-party libraries
5. Actuator - Production Ready features eg Metrics, Health Checks, Externalized Configuration
6. No XML configuration required

@ComponentScan

@Component<br>
@Service<br>
@Configuration<br>
@Repository<br>
@Controller<br>
@RestController

Dependency Injection

```
@Configuration
@Bean
public UserService buildUserService() {
        UserService us = new UserService();
        us.setSomeData("Some Data");

        retrun us;
}
```

**RESTful APIs**

(ResourceFul API)

HTTP Verbs:

1. GET: idempotent + safe - readResource
2. POST: non-idempotent + non-safe - createResource
3. PUT: idempotent + non-safe - createOrUpdateResource
4. DELETE: idempotent + non-safe - deleteResource
5. PATCH: idempotent + non-safe - partialUpdate

User: id, name, gender
Post:   1, Ankita, F

Put: 1, Ankita -> Ankita Kumari
    1, Ankita Kumari, null

Patch: 1, Ankita -> Ankita Kumari
    1, Ankita Kumari, F

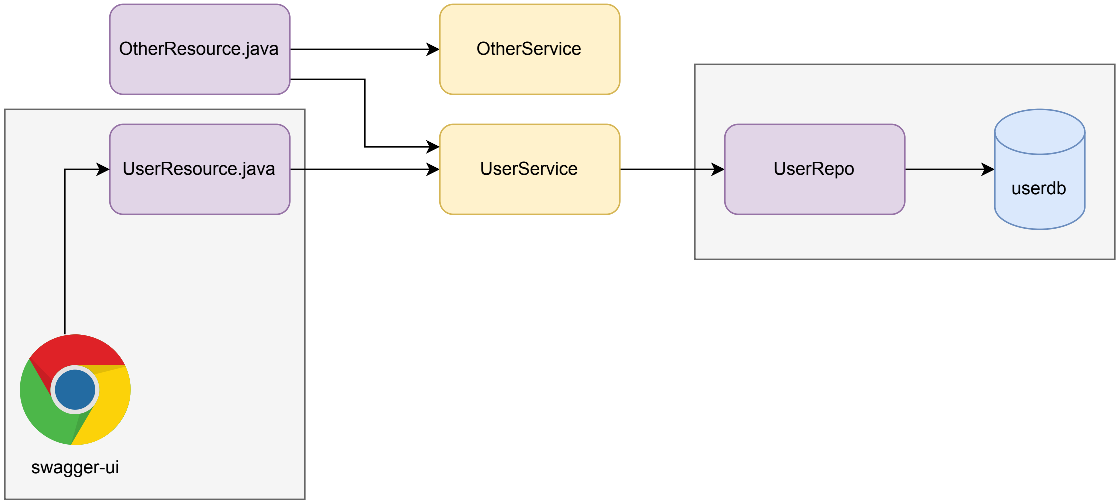
safe operation:

**Resource:**
model.User.java
model.Order.java

**Resource: User**

```
@GETMapping("/users")
public List<User> getAllUsers();

@POSTMapping("/users")
public User createUser(User user);

@PUTMapping("/users/{userId}")
public User updateOrCreateUser(Long userId);

@DELETEMapping("/users/{userId}")
public User deleteUser(Long userid);

@PATCHMapping("/users/{userId}")
public User updateUser(Long userId, User user);
```

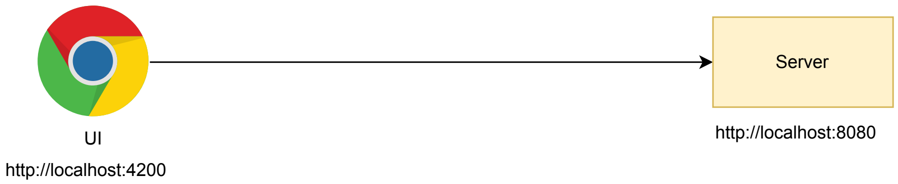
**Integration Test in Spring Boot**



**CORS**

(Cross Origin Resource Sharing)

http://abc.com:8081/users
http://localhost:8082/users



UI

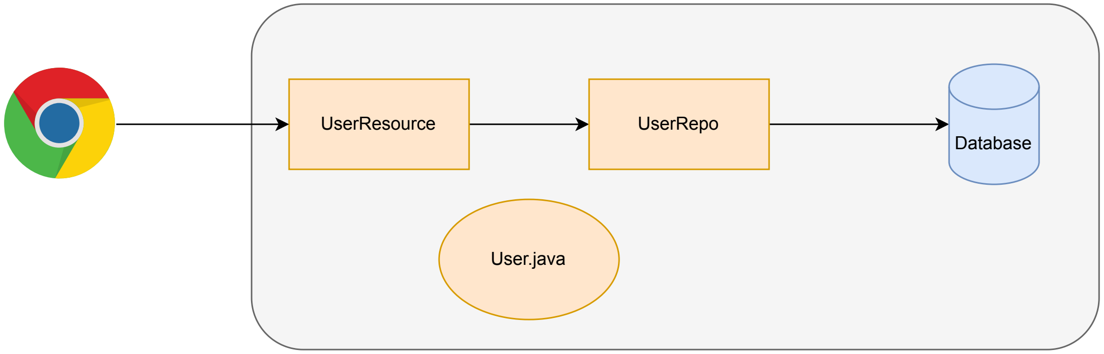
http://localhost:4200

Server

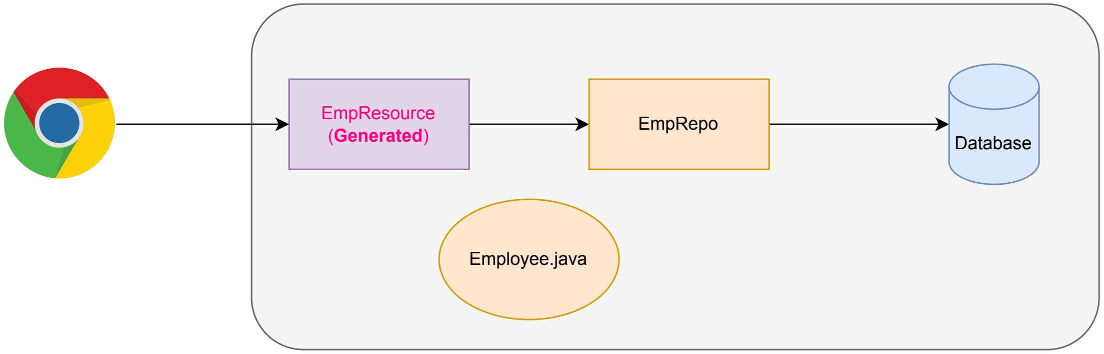
http://localhost:8080

**Spring Data Rest**

firstms with web(`spring-boot-starter-web`)

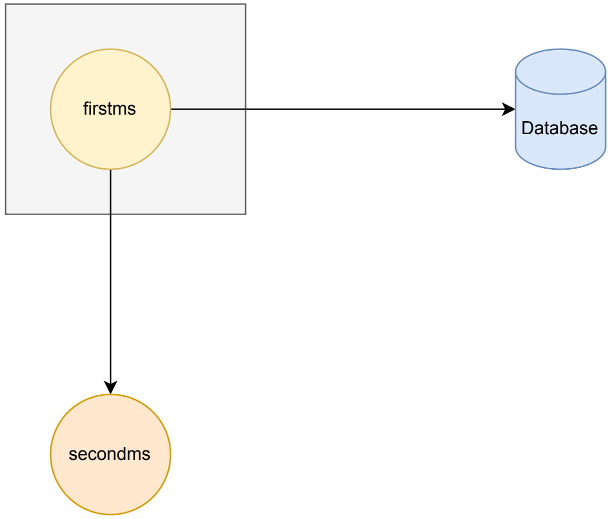


firstms with web(`spring-boot-starter-data-rest`)



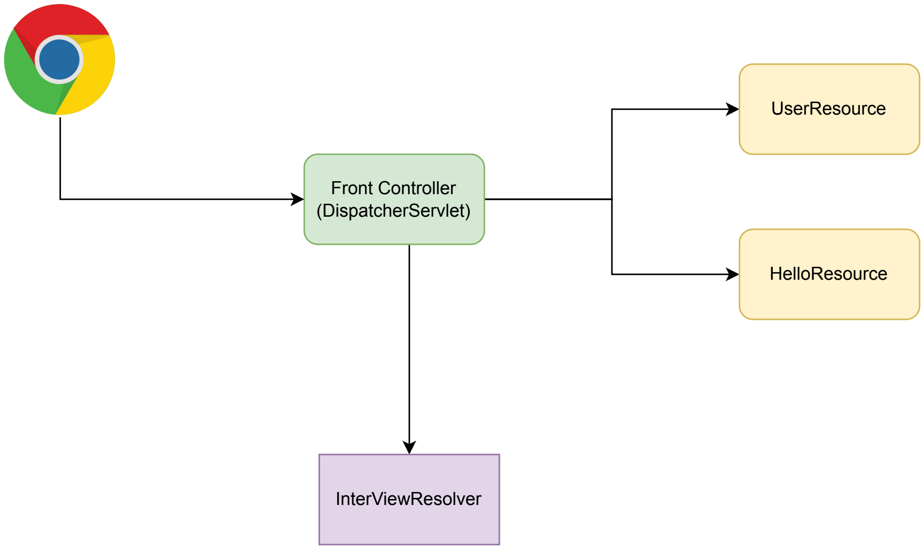
**Actuator**

CPU: usage, Memory: usage, disk, underlying OS, processors-number, processors usage, ping, thread-dump, heap-dump
JVM details, Garbage Collection, Tomcat, Tomcat Thread pool,  Database Thread pool,



**Dispatcher Servlet**

(Spring WebMVC)


UserResource
Front Controller
(DispatcherServlet)
HelloResource
InterViewResolver