```java
@RestController
@RequestMapping("/servers")
public class ServerController {

  @Autowired
  private ServerRepository serverRepository;

  @GetMapping
  public List<Server> getServers(@RequestParam(required = false) String id) {
   if (id != null) {
     Optional<Server> server = serverRepository.findById(id);
     if (server.isPresent()) {
       return Collections.singletonList(server.get());
     } else {
       throw new ResponseStatusException(HttpStatus.NOT_FOUND, "Server not found");
     }
   } else {
     return serverRepository.findAll();
   }
 }
}
@PostMapping("/servers")
Public void createServer(@RequestBody Server server) {
 serverRepository.save(server);
}
@DeleteMapping("/servers/{id}")
Public void deleteServer(@PathVariable String id) {
 Optional<Server> server = serverRepository.findById(id);
 If (server.isPresent()) {
```

```java
      serverRepository.delete(server.get());

  } else {

    Throw new ResponseStatusException(HttpStatus.NOT_FOUND, "Server not found");

  }

}

@GetMapping("/servers/search")

public List<Server> searchServersByName(@RequestParam String name) {

  List<Server> servers = serverRepository.findByNameContainingIgnoreCase(name);

  if (servers.isEmpty()) {

    throw new ResponseStatusException(HttpStatus.NOT_FOUND, "Server not found");

  } else {

    return servers;

  }

}

@Document(collection = "servers")

Public class Server {


  @Id

  Private String id;

  Private String name;

  Private String language;

  Private String framework;


  // constructors, getters and setters

}


Public interface ServerRepository extends MongoRepository<Server, String> {

  List<Server> findByNameContainingIgnoreCase(String name);

}
```

Spring.data.mongodb.host=localhost

Spring.data.mongodb.port=27017

Spring.data.mongodb.database=servers

```
{
  "name": "my centos",
  "id": "123",
  "language": "java",
  "framework": "django"
}
```