1. Zoom out
   1. Valuing architecture and then the code.
   2. Once you know architecture we can code by stages.
      1. Example.. service layer
      2. Dao layer independently
      3. We know how to use command line runner interface.
   3. Intent is to use things in a project than talk about it.
   4. Annotations
      1. What is benefit you get and when who gives you the benefit
   5. Reflection
      1. X wants to know about Y
      2. X cannot see Y directly
      3. So X shows Y to the mirror and gets asks questions and then mirror will tell about Y. based on X will take some decision
2. Spring framework
   1. Loose coupling
      1. Dependency injection
         1. Gives you an object locally in context of a function.. we can increase the scope if need is there..
      2. Inversion of control
   2. Application of template pattern
      1. X+ spring = x becomes easier.
         1. Jdbc Template
   3. Aspect oriented programming
   4. Spring boot
      1. We can reduce time spent in configuring.
3. Rest-based web services
   1. Knowing why web services.
      1. Distributed.
      2. You can let anybody call the service layer…
      3. Two standards at work
         1. http and its cousin
         2. JSON…
            1. Standard for representing variables..
      4. Knowing web basics would help
         1. Web for presentation
            1. http is used for makeup.. html, css..
         2. Web for web services
            1. http is used as a cover on service layer..
      5. we can check by using postman.. in real life it is some other application
         1. in context of spring . we have classes like RestTemplate, RestClient..WebClient..
4. Unit testing
   1. Culture plays very important role than technology..
   2. It can act like a documentation for the code, It should be fast..
   3. Importance of mocking whichever is slow.. and we should know
5. Cloud part of the story
   1. Availability of resources on demand…
      1. It is in this context
         1. Containers.. docker
         2. Microservices…