* topics:
    - zookeeper & curator
    - protobuf & RPC
    - HDFS & Spark & Yarn
    - spring boot & front-end
    - InMEM DB, Write-Ahead Logging, Sharding & Tablet, Replica

* bugs:
    - class loading -> dependency tree
    - circular dependency -> individual module
    - volatile -> NPE
    - get from ZK -> keep in MEM
    - open file -> try with resource
    - Spark date -> timestamp

* todo:
    - attract external developers
    - avro & protobuffer
    - data from DB & MQ
    - cache(not necessary), redis

* tradeoff:
    - popular technology || traditional & farmiliar technology
    - native lib || framework
    - normalizaton || denormalization DB
    - simplest API || customized API(configuration)
    - new features || technical support
    - external developers || internal support
    - spring-boot || jboss + severlet
    - simplest || cluster + distributed
    - nginx || ZK
    - monitor & auto config || manual config

* priority:
    - core functions -> user experience & performance
    - min modification -> completable solution, not block QA

* research:
    - data scientist & delivery team -> workshop
    - kaggle & data scientist -> dataset & competation
    - logistic regression & random forest classifier
    - pandas & sklearn

* contribute:
    - UT -> CICD
    - QPS -> Async Call
    - local -> distributed(HDFS+YARN)
    - native ZK -> curator
    - refactor -> common lib
    - lombok -> provided(compile + test)

* FEDB:
    - simply workflow -> reduce delibery time & build AI app
    - fe & DB
    - batch & real-time
    - standalone & cluster & docker
    - alone -> 4 developers
    - 1 -> 6~7 delivery projects

- bank -> bank & mobile-phone manufacturer & energe & government
    - innovation project in team -> core project in 4pd
    - patent

* position:
    - 2 completable producte -> fe & db
    - confused for my position & direction
    - bottom layer abilities -> expose
    - server layer development
    - resource management & task scheduling
    - distributed & high avalaible

* PM:
    - WIKI
    - estimate workload
    - todo list -> monthly plan & weekly plan
    - record progress & milestones
    - meeting -> review output & fix direction & reorganize plan
    - outout documents
    - function test
    - performance test
    - bug fix -> regression test

* develop:
    - collect requirements, join workshop
    - list priority
    - develop
    - function test
    - bug fix + UT & regression test
    - performance test -> highest QPS
    - release
    - group chat -> collect feedback -> new features

* performance:
    - ZK -> in MEM
    - sync + thread num -> async
    - list -> map
    - lock range: function -> code block

* difficult:
    - data input -> file & DB & MQ
    - real customers' requirements -> onsite to customer's office
    - schema infer -> input by customer

* HTTP:
    - DNS, url -> ip, based on location
    - TCP/UDP connection
    - HTTP request
    - server process
    - HTTP response
    - close TCP/UDP connection
    - browser parse html & render to user