Tiger AppStore

--Web in Java

Qun Wu

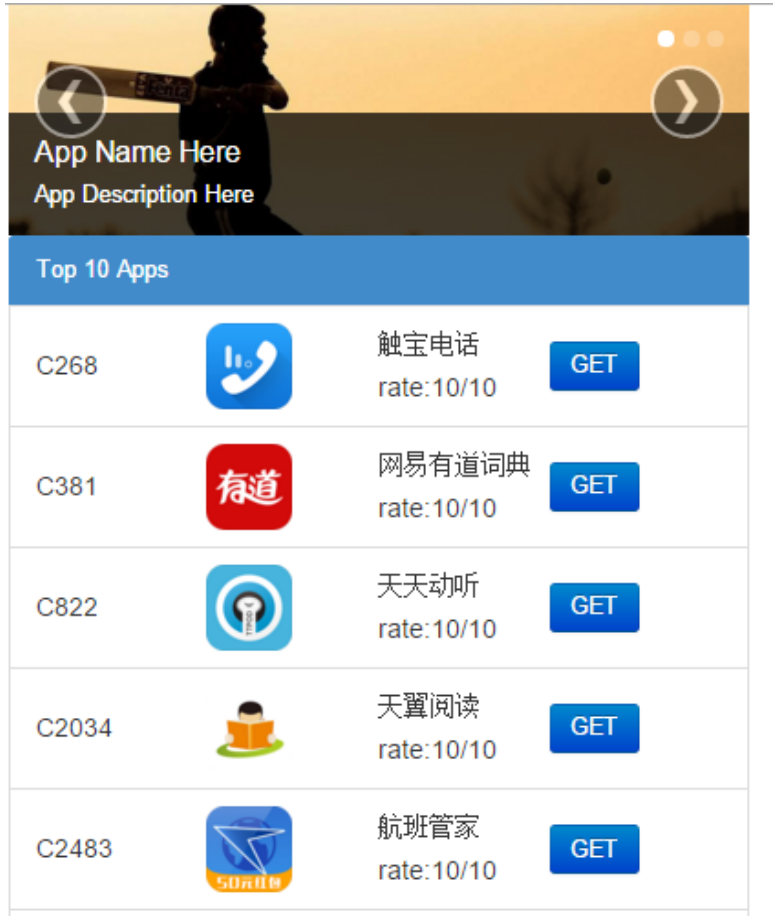
Content

- Project scope and goal
- Techniques used
- Web Architecture
- Data Source
- Problems
- Further work
- Reference
- Web Components—Coding Practice

Project scope and goal

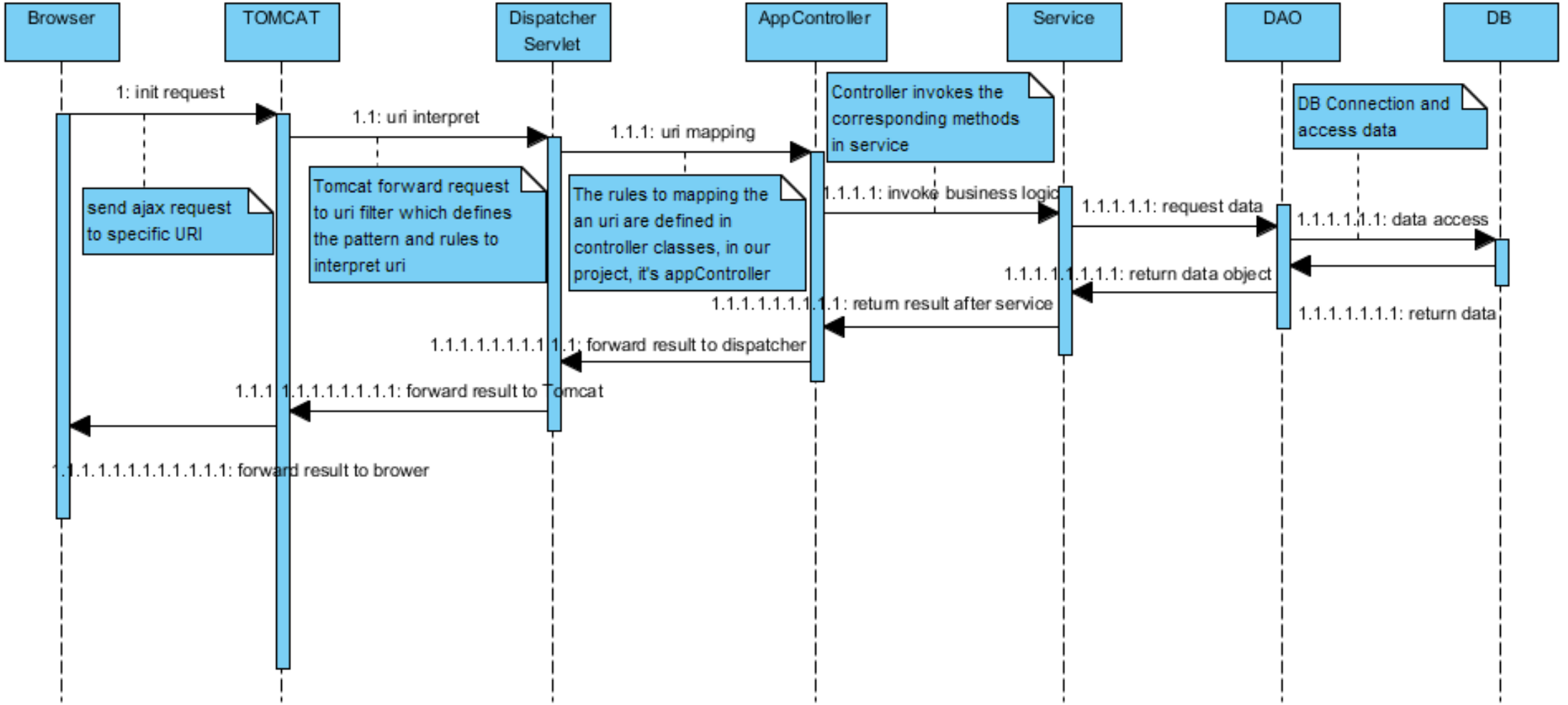
- Problem Domain: develop a web system to show apps and relative recommendation.
- Problem scope: 1. server/running environment setup,
  2. database, web services, front web pages
  → full stack project
- Use cases: 1. initially, show top 10 popular apps
  2. click “Get” button to obtain the app detail

Project - view of project



Project – Web Behavior

- Web Project General Behavior



Technique used- Front-end

- Front-end pages:

HTML, CSS(bootstrap), JS(jQuery, AngularJS)

1. Import JQuery before bootstrap lib
2. AngularJS is used to manipulates DOM tree like what JQuery can do, but it does not depend on JQuery. AngularJS has its own implementation to changes DOM tree, called JQLite
3. AngularJS has ‘http’ service to send Ajax request

Technique used-Back-end

- Back-end(JAVA web, Restful API):

Control layer: Spring-mvc

Business layer: JAVA

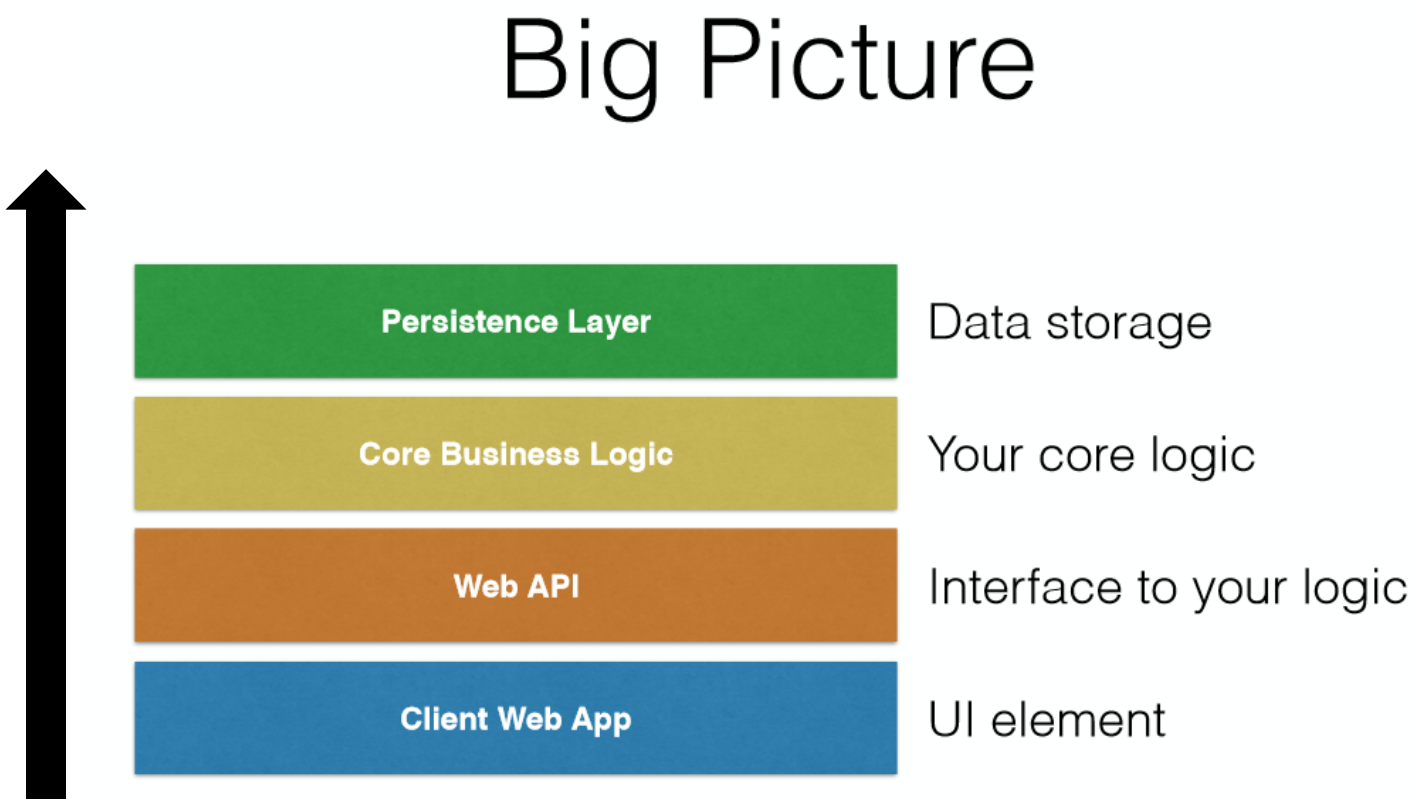
DB Connection & Operation: Hibernate

Spring framework throughout the whole back-end(be introduced later)

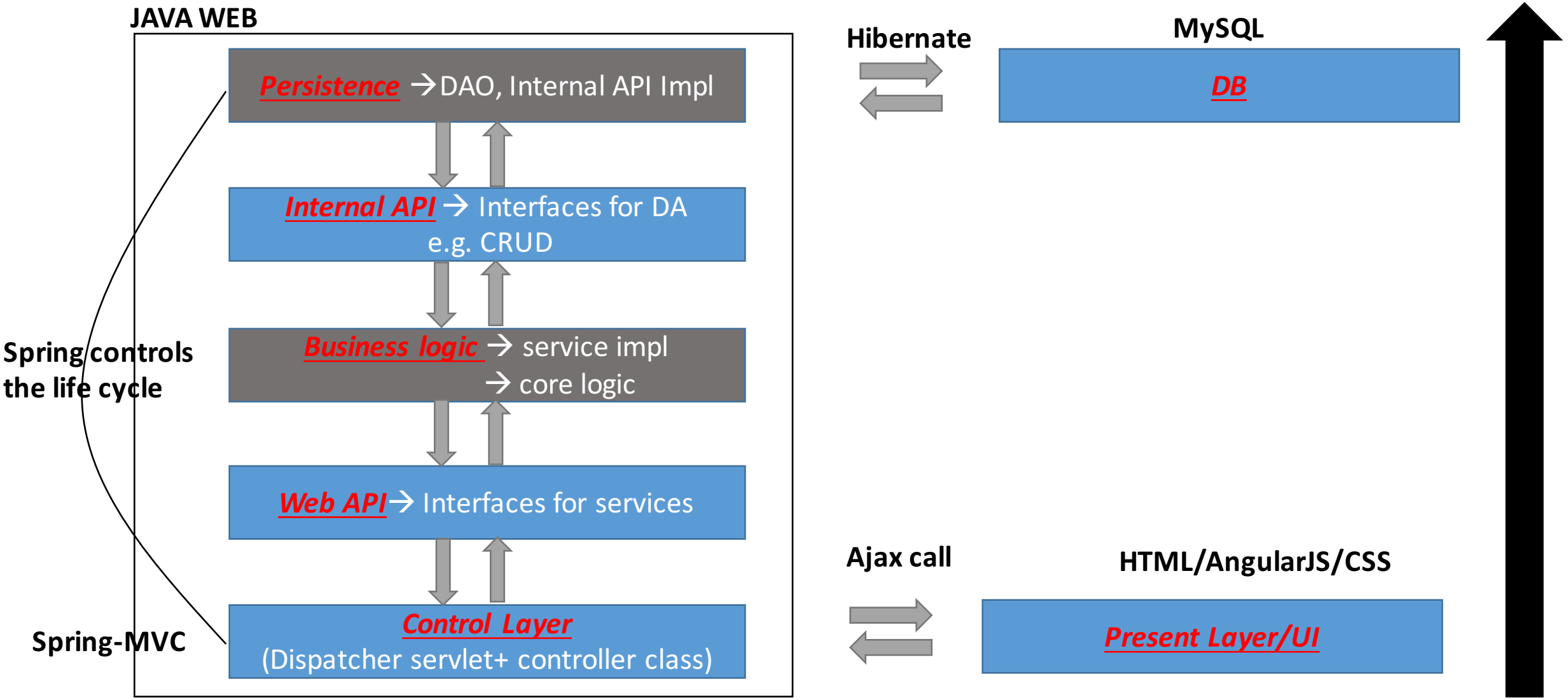
- Database: MySQL, not MongoDB

Web Architecture

- Service-oriented-architecture (SOA)



Web Architecture-Real architecture:



Web Component – Spring

- Java bean generation/ Instantiation of object(IOC)

Recommend: **setter** Injection (Jackson also use it). Also has constructor injection.

- Design pattern e.g. singleton, prototype.

Singleton vs Prototype:

e.g. List<User> users ;→every time after login, add user to list.

If “users” is

Singleton: one list “users”, keeps all users

Prototype: users will be initiated newly each time

- Java web frameworks integration depends on spring. Why not hibernate, struts? → java bean support

Web Component – Spring - Injection

- Which parts in project shall be initiated by Spring?

Java beans → service and dao

e.g. AppService appServ = new AppServImpl();

Spring→ Object of 'AppServImpl' to a JAVA BEAN with id 'appServBean'

→ 'appServBean' will be injected to variable "appServ" by Spring

- How about the entities.

e.g. When user login → encapsulate username and pwd into object

Jackson → encapsulate data from json into object

Hibernate → encapsulate data from MySQL into object

Data Source

- Storage Platform: MySQL 5.5

Download: http://dev.mysql.com/downloads/mysql/

(→ page partial update when select download options)

Installation:
http://jingyan.baidu.com/article/ed2a5d1f4968c909f6be179f.html

- Data Source:

Import sql file into MySQL directly

```
mysql> source file_name
```

http://dev.mysql.com/doc/refman/5.0/en/mysqlimport.html

Problems

- GET/delete method do **not** accept **JSON** DATA in request
- Spring-mvc Jackson: 500 internal error/400 bad request

1. How Jackson gets values from json data or object fields

→Getter and setter

2. What happen when null value appears in object or json data

→400 bad request when send json to server

→500 internal error when server send json to front-end

3. MySQL/mongodb Chinese language confict

→Code page 936 to fix the problem in mongodb

→In mysql it cannot be fixed. But it does not have impact on project. Pages work well

Further work

- Do not use SQL, intead, use mongodb directly

Sol-1: hibernate OGM (important: still support HQL, also support native query in JPA)

Sol-2: do not use hibernate either. Instead, use java code

- Enlarge the scope of project.

1. User management
2. Add category for app
3. Add comments

- Make it be a real single page website with AngularJS

Sol: make HTMLs be independent components of one page, dynamically load these components into single page

Reference

- IOC:

https://zh.wikipedia.org/wiki/%E6%8E%A7%E5%88%B6%E5%8F%8D%E8%BD%AC

- AOP: http://baike.baidu.com/subview/73626/13548606.htm
- MongoDB with JAVA, demo : http://www.cnblogs.com/hoojo/archive/2011/06/02/2068665.html
- MongoDB with JAVA, doc :

https://docs.mongodb.org/getting-started/java/

- Hibernate Doc:

http://docs.jboss.org/hibernate/orm/4.3/manual/en-US/html_single/

- Hibernate OGM HQL&JP-QL support

http://docs.jboss.org/hibernate/ogm/4.2/reference/en-US/html/ch07.html

http://docs.jboss.org/hibernate/ogm/4.2/reference/en-US/html/ch07.html

- Running environment setup

- Start to install MySQL (5.5), Tomcat (8.0)
- import data
- Connect Tomcat with your IDE (MyEclipse)

about 15 mins

Web Component – create a web project

- Create a web project (add maven support → optional)
- Correct web.xml

1. <welcome-file-list>
2. <servlet-name>+<servlet-class>→url filter, depends on control layer
3. <servlet-name>+<url-pattern>/</url-pattern>→url pattern to match

- Import jar files for spring-mvc, spring framework, hibernate

Web Component – create a web project

- Entity Definition
- Code Web service, including DAO layer which uses hibernate support

1. Web API + service implementation
2. Internal API + DAO(crud first, then extend others functions)

- Configure the spring-config.xml (dynamic change, read by web.xml)

1. Generate java beans, control their left cycle
2. Integrate Hibernate

Web Component - front-end

- Front-end pages:
- Import jQuery lib
- Import angularJS lib
- Import bootstrap lib and necessary style file
- Quickly generate a web layout

http://www.runoob.com/try/bootstrap/layoutit/