Spring Boot 기반의 배포용 오픈소스 Wiki 개발기

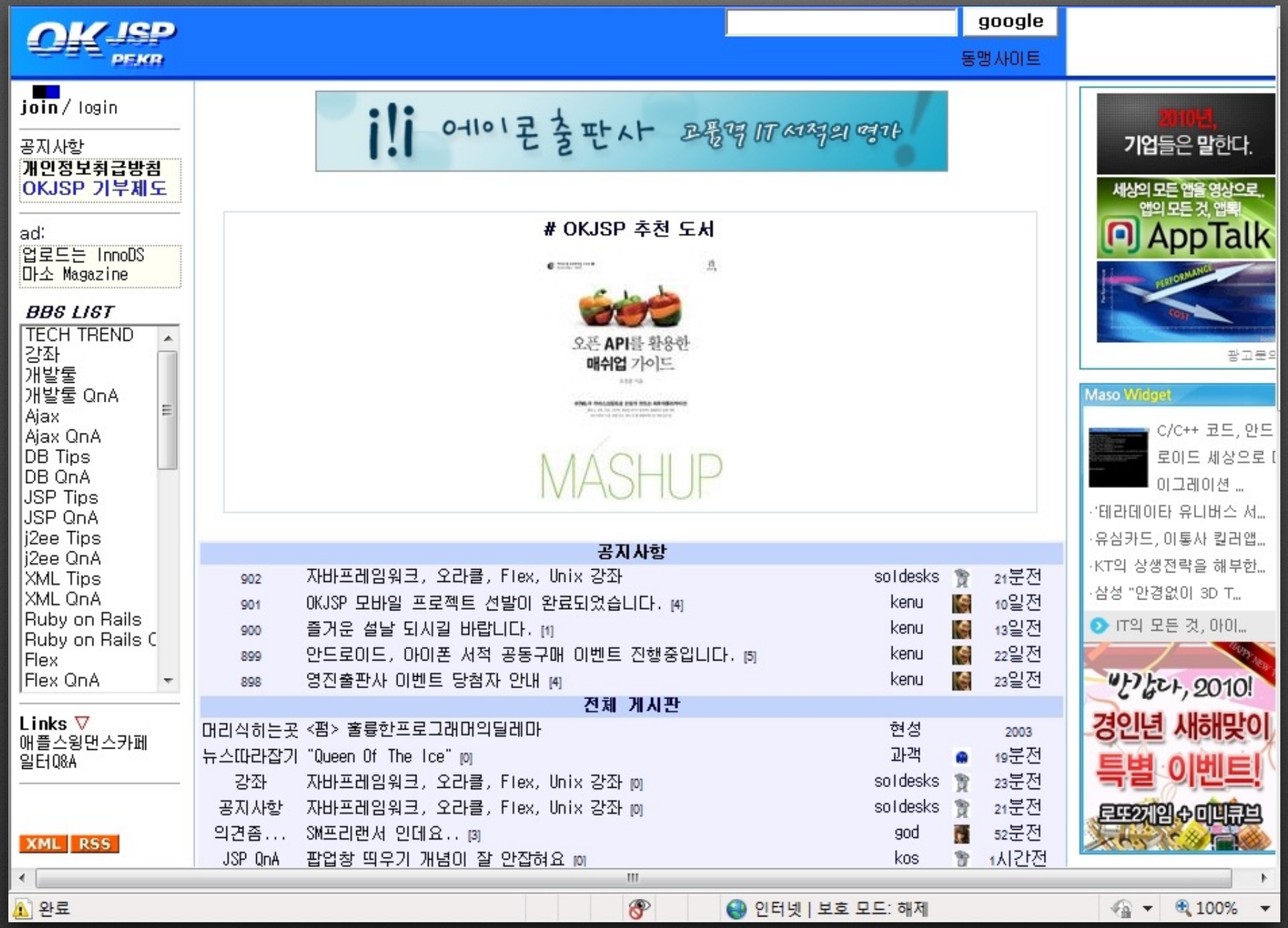
Howling
한용희

목차

- Howling 오픈소스 팀 소개
- 오픈소스를 시작하게 된 계기
- LibQa

LibQa

- Howling Opensource Team
  - Glider Wiki
  - LibQa
  - DualEditor



오픈소스의 시작 - 커뮤니티

#239038 공지사항

Groovy, Grails, Gradle 관심 있으신분~

올해 잘 쉬었으니 내년 스터디를 해볼 생각입니다.
여러가지 고려해봤는데 아무래도 Groovy, Grails, Gradle 을 해보고 뭔가 아웃풋을 내보는 형태의 스터디를 해보는게 낫겠다 싶네요.

혹시 위의 주제로 스터디를 한다면 관심있으신분 있을까요?

Spring 에서 요즘 Groovy를 조금씩 지원해주고 있고, Spock 이라는 테스트 프레임웍도 나와서 그 동안 손놓고 있던 그루비를 본격적으로 해볼 생각입니다.

일정은 설날 전후부터 3개월 정도 되겠네요.
혹시나 생각 있으신 분들은 댓글 부탁드려요.

인원 고려해서 계획을 좀 짜볼 생각입니다~

Groovy, Grails, Gradle 스터디 모집

Groovy&Grails + Gradle 스터디

- 그루비 & 그레일즈 + 그래들 스터디 팀의 문서 공간입니다.
- 남들이 읽고 이해할 수 있는 문서를 만들어주세요.

Groovy&Grails + Gradle 스터디 기본정보

01. 1주차 1-1 Groovy 기초 (2)	Q u i c K	2014.01.23
01. 1주차 1-3 Groovy 집합 자료형 (0)	멋남	2014.02.07
01. 1주차 1-2 Groovy 단순 자료형 (0)	멋남	2014.02.07
01. 1주차 1-3 클로저 사용하기 (0)	멋남	2014.02.07
01. 1주차 1-4 그루비의 제어문 (0)	멋남	2014.02.07
02. 2주차. Groovy 기본 문법 2 (0)	Q u i c K	2014.01.23
02. 2주차. 7장 그루비의 동적 객체지향 - 1 (0)	visualkhh	2014.02.09
02. 2주차. 7장 그루비의 동적 객체지향 - 2 (0)	visualkhh	2014.02.11
02. 2주차. 8장 빌더사용하기 (0)	visualkhh	2014.02.09
02. 2주차. 9장 GDK 사용하기 (0)	visualkhh	2014.02.09
02. 2주차. 9장 GDK 사용하기-2 (0)	visualkhh	2014.02.15
02. 2주차. 10장 그루비 데이터베이스 프로그래밍 (0)	visualkhh	2014.02.09

최근 업데이트

sjune 2014.04.13	**09. 9주차 Plug-in** h1. Plugin 이번 문서에서는 Plugin과 Convention에 대해 살펴보고 ...	
Q u i c K 2014.04.12	**10. 10주차. Gradle 활용** 10주차. Gradle 활용 - 백성진 (3월 1일)	
sjune 2014.04.11	**09. 9주차 Gradle Command ...** h1. Gradle Command-Line 이번 문서에서는 Gradle command-line의 ...	
sjune 2014.04.05	**09. 9주차 Gradle 설치 및 프로젝...** h1. Gradle 개요 Gradle은 Java (JVM) 환경에서 작동하는 Groovy 기...	
sjune 2014.04.04	**09. 9주차 Gradle 기본 스크립트와...** h1. Gradle 기본 스크립트 h2. Gradle 빌드 스크립트 구조 [a...	

최근 업로드 자료

2014.02 ~2014.04

11명 시작 - 최종 10명

10개 주제 완료

스터디 규칙

- 발표 내용 Wiki에 미리(한달 전) 정리하여 공유
- 발표 시간 3시간
- 작성자가 이해하는 문서가 아닌 **다른 사람이 보고 따라하고 이해할 수 있는 수준**의 문서
- 지각비 10분 천원, 결석 2만원, 발표자 결석시 탈락

스터디 회고

- 사람이 전부
- 초기 인원 선정
- 강력한 규칙
- 주도적인 참여

오픈소스에 대한 고민

- 어떻게 참여하지?
- 내가 뭘 할 수 있을까?
- 내가 잘 할 수 있을까?

그럼 가까운 경험(스터디)에서 고민 하자

http://bbs.ruliweb.com/hobby/board/300143/read/19513646

스터디 중 가졌던 고민

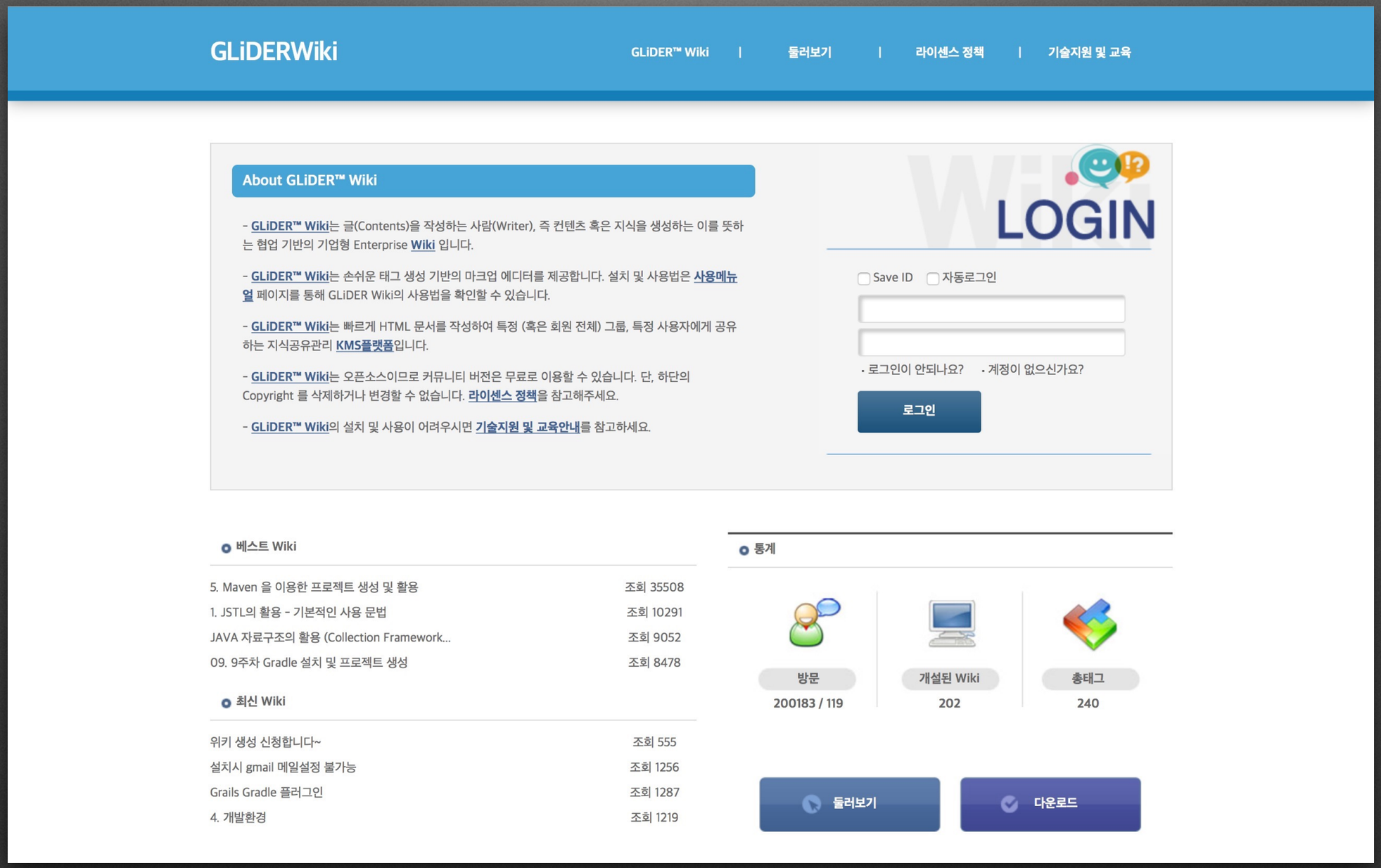
- 컨텐츠 작성하기 참 번거롭네...
- 컨텐츠 작성자에게 질문은 어떻게 하지?
- 컨텐츠를 정리 하기 좋은 툴 뭐 없나?
- 블로그? 게시판? 카페? 홈피? 어떻게 퍼가지?

막상 써보니 위키가 甲인데...

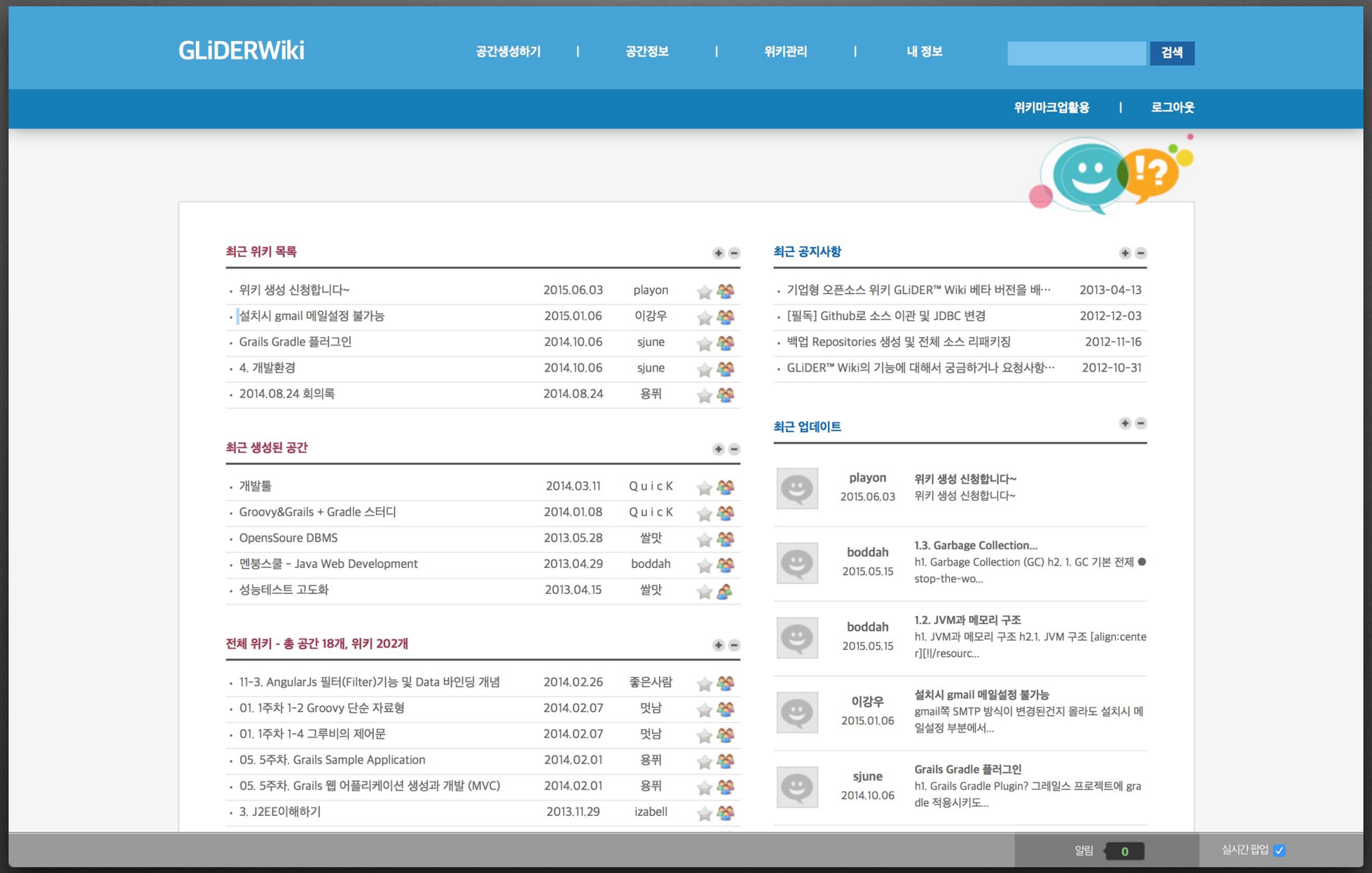
- 왜 우리나라에서 잘 쓰이질 않지?
- 내 글을 왜 남이 수정하나?
- 수직적 조직 문화...
- Learning Curve...

이런걸 개선하면 더 편하지 않을까?

스터디가 끝나고 배운걸 활용해보자



기존 Glider Wiki



기존 Glider Wiki

기존 오픈소스 철학

단순하고

배우기 쉽고

유용한

Glider Wiki 단점

- Wiki만 작성 가능
- 커뮤니티 기능 불가
- MarkDown 작성시 실시간 조회 불가
- 설치 번거로움

불편함

LibQa

- 배포용 Wiki(기존 Glider Wiki http://gliderwiki.org/)
- 커뮤니티 기능(SNS 연동, Feed)을 추가
- Keyword 기반 Q&A 추가
- WAS(Jetty, Tomcat, Undertow) 와 DB(H2, HSQL, Derby) 를 내장하여 WAR로 배포 = 무설치
- 실시간 뷰가 가능한 DualEditor(Markup)
- Community Service 2016.05.15 정식오픈

LibQa 개발기

내가 가는 곳이 길이다!

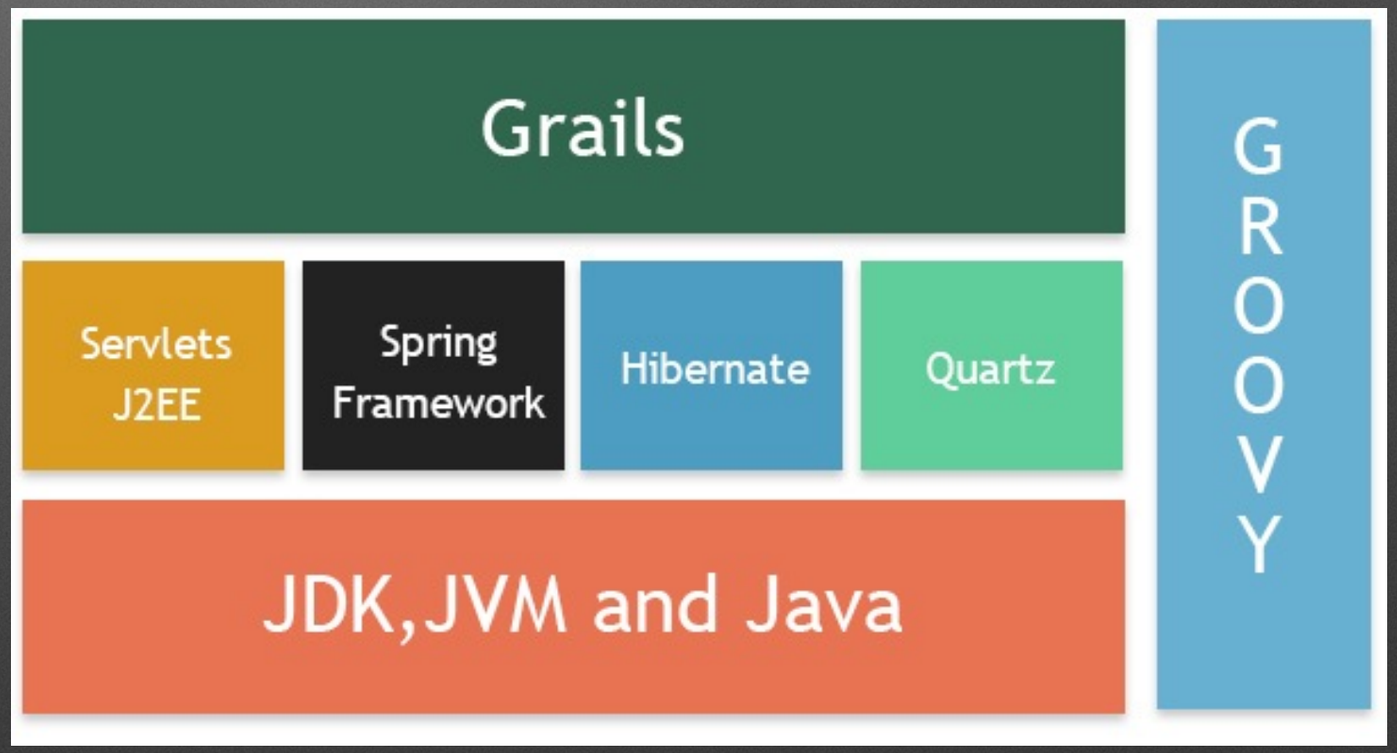
http://playwares.com/freegallery/13826263

새로운 언어와 프레임웍 도입
Groovy, Grails

삽질의 시작

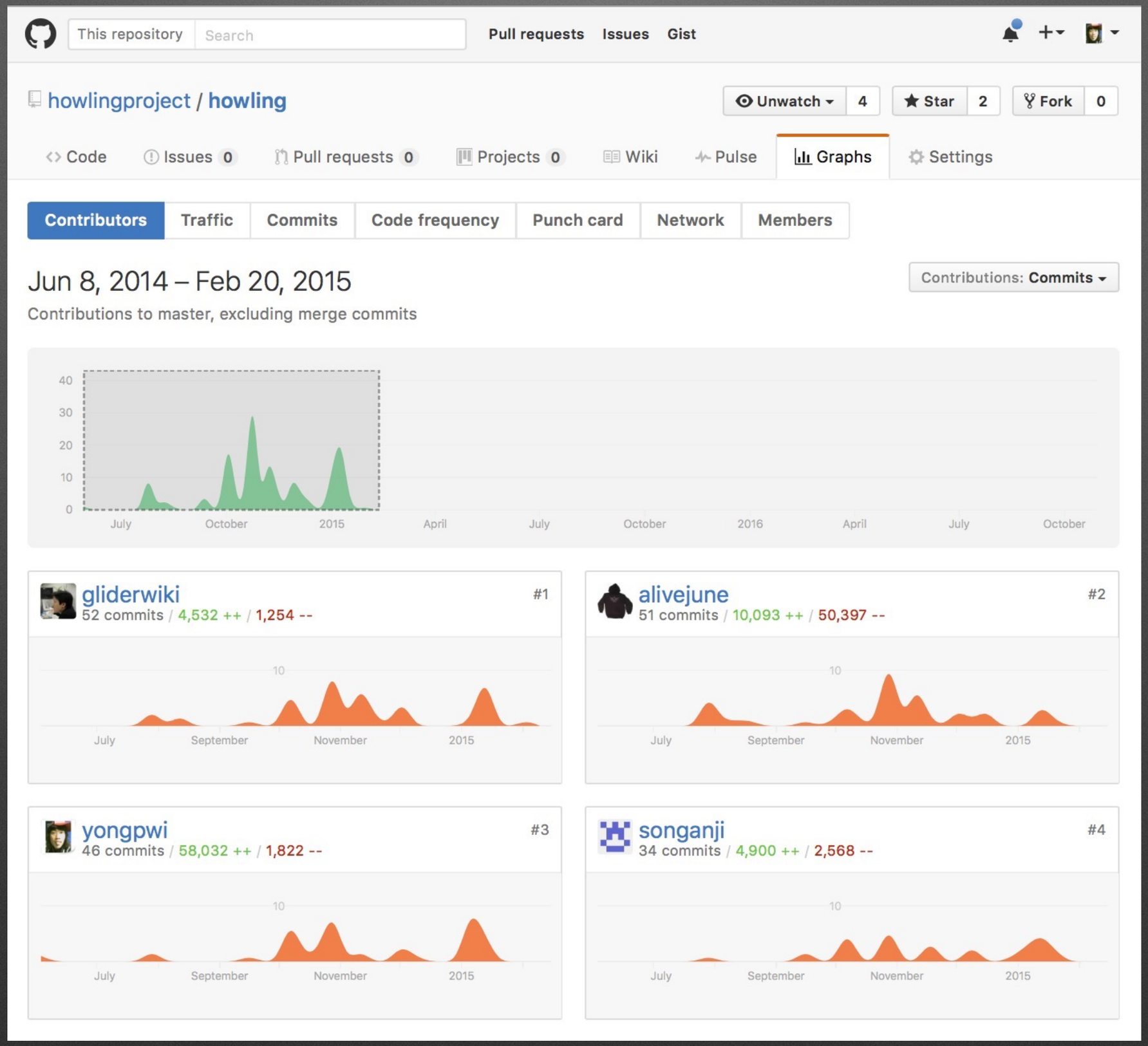
http://brentphone.kr/contents/board/today_eng_view.asp?opt_gubun=&opt_wrd=&seq=654

- Java 기반
- 유연한 타입
- 간결한 문법
- 학습 시간 비용
- 생산성



Full-stack
(WAS, DB Embedded)

고난의 시작


This repository
Search
Pull requests
Issues
Gist
howlingproject / howling
Unwatch 4
Star 2
Fork 0
Code
Issues 0
Pull requests 0
Projects 0
Wiki
Pulse
Graphs
Settings
Contributors
Traffic
Commits
Code frequency
Punch card
Network
Members
Jun 8, 2014 – Feb 20, 2015
Contributions: Commits
Contributions to master, excluding merge commits
40
30
20
10
0
July
October
2015
April
July
October
2016
April
July
October
gliderwiki
52 commits / 4,532 ++ / 1,254 --
#1
alivejune
51 commits / 10,093 ++ / 50,397 --
#2
yongpwi
46 commits / 58,032 ++ / 1,822 --
#3
songanji
34 commits / 4,900 ++ / 2,568 --
#4
10
July
September
November
2015

10개월 가량 개발(삽질)!

프레임웍 변경

- Grails 기반 실패
  - 생태계
  - 자료
- Spring Boot
  - 자동설정
  - 개발집중
  - 생산성
  - 건강한 생태계

spring
boot

LibQa Spring Boot

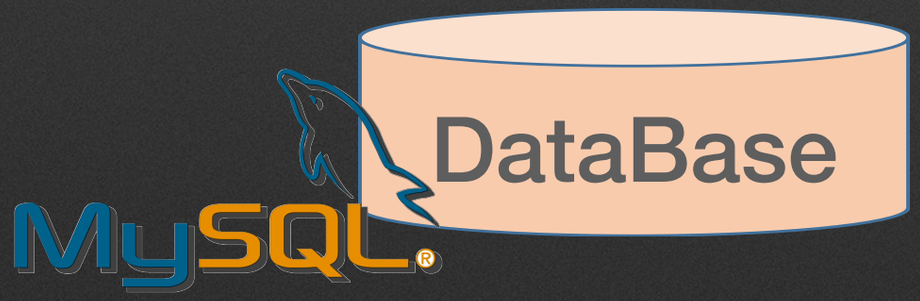
Logging

Template Engine

Spring Security

Spring Core 4.1.X

SQL



SPRING INITIALIZR bootstrap your application now

Generate a Gradle Project with Spring Boot 1.4.1

Project Metadata

Artifact coordinates

Group

com.libga

Artifact

libga

Dependencies

Add Spring Boot Starters and dependencies to your application

Search for dependencies

Web, Security, JPA, Actuator, Devtools...

Selected Dependencies

Generate Project ⌘ + ⏎

Don't know what to look for? Want more options? Switch to the full version.

https://start.spring.io

build(maven, gradle), group, artifact, spring boot version

```
# Hibernate config
spring.jpa.hibernate.ddl-auto=update
spring.jpa.generate-ddl=false
spring.jpa.database=MYSQL
spring.jpa.show-sql=false
spring.jpa.hibernate.dialect=org.hibernate.dialect.MySQL5Dialect
spring.jpa.properties.hibernate.format_sql=false
spring.datasource.validation-query=SELECT 1
spring.datasource.test-on-borrow=false
spring.datasource.test-on-return=false

# JDBC
spring.datasource.url=jdbc:mysql://localhost:3306/dbSchema?useUnicode=true&charaterEncoding=utf-8
spring.datasource.username=dbUserName
spring.datasource.password=dbPassword
spring.datasource.driverClassName=net.sf.log4jdbc.sql.jdbcapi.DriverSpy
spring.datasource.maxIdle=20
spring.datasource.maxActive=400
spring.datasource.maxWait=10000

# lucene configuration
spring.jpa.properties.hibernate.search.default.directory_provider=filesystem
spring.jpa.properties.hibernate.search.default.indexBase=~/libqa/lucene/indexes/

# file Size
multipart.maxFileSize=10Mb

# log level config
logging.level.root=INFO
logging.level.org.springframework.web=WARN
logging.level.org.hibernate=WARN

# server
server.port=8080

# libqa configuration
libqa.viewResolver.cached=false
libqa.file.uploadPath=~/libq/files
```

Spring Boot
application.properties(DB, log, server port 등)설정

```java
package com.libqa;

import ...

/**
 * Created by yion on 2015. 1. 25..
 */
@SpringBootApplication
@EnableTransactionManagement
public class Application extends SpringBootServletInitializer {

    public static void main(String[] args) { SpringApplication.run(Application.class, args); }

    @Override
    protected SpringApplicationBuilder configure(SpringApplicationBuilder application) {
        return application.sources(Application.class);
    }

    @Bean
    public HttpMessageConverter<String> responseBodyConverter() {
        return new StringHttpMessageConverter(Charset.forName("UTF-8"));
    }

    @Bean
    public Filter characterEncodingFilter() {
        CharacterEncodingFilter characterEncodingFilter = new CharacterEncodingFilter();
        characterEncodingFilter.setEncoding("UTF-8");
        characterEncodingFilter.setForceEncoding(true);
        return characterEncodingFilter;
    }
}
```

Spring Boot 실행

```
/Library/Java/JavaVirtualMachines/jdk1.8.0_73.jdk/Contents/Home/bin/java ...
objc[62982]: Class JavaLaunchHelper is implemented in both /Library/Java/JavaVirtualMachines/jdk1.8.0_73.jdk/Contents/Home/bin/java and /Library/Java/JavaVirtualMachines/jdk1.8.0_73↙
↘.jdk/Contents/Home/jre/lib/libinstrument.dylib. One of the two will be used. Which one is undefined.
Connected to the target VM, address: '127.0.0.1:61518', transport: 'socket'

  _   _                _ _                  _     _ _
 | | | | _____      __ (_)_ __   __ _      | |   (_) |__   __ _   __ _
 | |_| |/ _ \ \ /\ / / | | '_ \ / _` |_____| |   | | '_ \ / _` |/ _` |
 |  _  | (_) \ V  V /| | | | | | (_| |_____| |___| | |_) | (_| | (_| |
 |_| |_|\___/ \_/\_/ |_|_|_| |_|\__, |     |_____|_|_.__/ \__, |\__,_|
                                |___/                      |_|

2016-11-03 01:46:17.938  INFO 62982 --- [           main] com.libqa.Application                    : Starting Application on yongui-MBP with PID 62982
 (/Users/yong/Developer/workspace/howling/libqa/build/classes/main started by yong in /Users/yong/Developer/workspace/howling/libqa)
2016-11-03 01:46:17.943  INFO 62982 --- [           main] com.libqa.Application                    : No active profile set, falling back to default profiles: default
2016-11-03 01:46:18.549  INFO 62982 --- [           main] ationConfigEmbeddedWebApplicationContext : Refreshing org.springframework.boot.context.embedded
 .AnnotationConfigEmbeddedWebApplicationContext@723ca036: startup date [Thu Nov 03 01:46:18 KST 2016]; root of context hierarchy
2016-11-03 01:46:20.112  INFO 62982 --- [           main] o.s.b.f.s.DefaultListableBeanFactory     : Overriding bean definition for bean 'characterEncodingFilter' with a different
 definition: replacing [Root bean: class [null]; scope=; abstract=false; lazyInit=false; autowireMode=3; dependencyCheck=0; autowireCandidate=true; primary=false;
 factoryBeanName=application; factoryMethodName=characterEncodingFilter; initMethodName=null; destroyMethodName=(inferred); defined in com.libqa.Application] with [Root bean: class
 [null]; scope=; abstract=false; lazyInit=false; autowireMode=3; dependencyCheck=0; autowireCandidate=true; primary=false; factoryBeanName=org.springframework.boot.autoconfigure.web
 .HttpEncodingAutoConfiguration; factoryMethodName=characterEncodingFilter; initMethodName=null; destroyMethodName=(inferred); defined in class path resource
 [org/springframework/boot/autoconfigure/web/HttpEncodingAutoConfiguration.class]]
2016-11-03 01:46:20.433  INFO 62982 --- [           main] f.a.AutowiredAnnotationBeanPostProcessor : JSR-330 'javax.inject.Inject' annotation found and supported for autowiring
2016-11-03 01:46:20.548  INFO 62982 --- [           main] trationDelegate$BeanPostProcessorChecker : Bean 'org.springframework.transaction.annotation
 .ProxyTransactionManagementConfiguration' of type [class org.springframework.transaction.annotation.ProxyTransactionManagementConfiguration$$EnhancerBySpringCGLIB$$49738ff0] is not
 eligible for getting processed by all BeanPostProcessors (for example: not eligible for auto-proxying)
2016-11-03 01:46:20.610  INFO 62982 --- [           main] trationDelegate$BeanPostProcessorChecker : Bean 'org.springframework.security.config.annotation.configuration
 .ObjectPostProcessorConfiguration' of type [class org.springframework.security.config.annotation.configuration.ObjectPostProcessorConfiguration$$EnhancerBySpringCGLIB$$e221582a] is not
 eligible for getting processed by all BeanPostProcessors (for example: not eligible for auto-proxying)
2016-11-03 01:46:20.622  INFO 62982 --- [           main] trationDelegate$BeanPostProcessorChecker : Bean 'objectPostProcessor' of type [class org.springframework.security.config
 .annotation.configuration.AutowireBeanFactoryObjectPostProcessor] is not eligible for getting processed by all BeanPostProcessors (for example: not eligible for auto-proxying)
2016-11-03 01:46:20.624  INFO 62982 --- [           main] trationDelegate$BeanPostProcessorChecker : Bean 'org.springframework.security.access.expression.method
 .DefaultMethodSecurityExpressionHandler@37348491' of type [class org.springframework.security.access.expression.method.DefaultMethodSecurityExpressionHandler] is not eligible for getting
 processed by all BeanPostProcessors (for example: not eligible for auto-proxying)
2016-11-03 01:46:20.635  INFO 62982 --- [           main] trationDelegate$BeanPostProcessorChecker : Bean 'org.springframework.security.config.annotation.method.configuration
 .GlobalMethodSecurityConfiguration' of type [class org.springframework.security.config.annotation.method.configuration.GlobalMethodSecurityConfiguration$$EnhancerBySpringCGLIB$$6f5fadc]
 is not eligible for getting processed by all BeanPostProcessors (for example: not eligible for auto-proxying)
2016-11-03 01:46:20.658  INFO 62982 --- [           main] trationDelegate$BeanPostProcessorChecker : Bean 'methodSecurityMetadataSource' of type [class org.springframework.security.access
 .method.DelegatingMethodSecurityMetadataSource] is not eligible for getting processed by all BeanPostProcessors (for example: not eligible for auto-proxying)
2016-11-03 01:46:20.671  INFO 62982 --- [           main] trationDelegate$BeanPostProcessorChecker : Bean 'cacheConfiguration' of type [class com.libqa.config
 .CacheConfiguration$$EnhancerBySpringCGLIB$$c7c9d1af] is not eligible for getting processed by all BeanPostProcessors (for example: not eligible for auto-proxying)
2016-11-03 01:46:20.728  INFO 62982 --- [           main] trationDelegate$BeanPostProcessorChecker : Bean 'cacheManager' of type [class org.springframework.cache.support
 .SimpleCacheManager] is not eligible for getting processed by all BeanPostProcessors (for example: not eligible for auto-proxying)
```

아름다운 Log와 함께 Spring Boot 실행

예전에는

```
/* Formatted on 2009/10/29 14:49 (Formatter Plus v4.8.8) */
SELECT grp_cd AS grpcd, user_nm, depart_nm ,position_nm,
       nvl(
            (SELECT y.duty_nm
              FROM k_cdp z, k_compe_duty y
             WHERE z.grp_cd = y.grp_cd
               AND z.duty_cd = y.duty_cd
               AND z.grp_cd = ver.grp_cd
               AND z.user_id = ver.user_id
               AND z.cdp_type = 'P'), (select t.duty_nm from k_compe_duty t, i_member tt where t.grp_cd = tt.grp_cd and t.duty_cd
       (select y.duty_nm from k_cdp z, k_compe_duty y where z.grp_cd = y.grp_cd and z.duty_cd = y.duty_cd and  z.grp_cd = ver.grp_
       edu_type AS TYPE, lec_cd AS leccd, lec_nm AS lecnm,
       lec_seq AS lecseq, lead_type AS leadtype,
       f_get_member_no (user_id) AS memberno, user_id AS userid,
       (SELECT regist_no
          FROM i_member
         WHERE user_id = ver.user_id) AS juminno,
       f_get_codemaster_nm ('COM07', lead_type, grp_cd) AS leadtypenm,
       lec_method AS lecmethod,
       f_get_codemaster_nm ('COM06', lec_method, grp_cd) AS lecmethodnm,
       credit AS edutime,
          TO_CHAR (lec_start_dt, 'YY.MM.DD')
       || '~'
       || TO_CHAR (lec_end_dt, 'YY.MM.DD') AS lecenddt,
       lec_start_dt AS lecstartdt, edu_place_nm AS eduplace,
       DECODE (finished_yn, 'Y', '¼ö·á', '¹Ì¼ö·á') AS finishedyn,
       edu_cost AS educost, refund_cost AS refundcost, final_point AS score,
       edu_part_nm AS edupartnm, edu_institute_nm AS eduinstitutenm,
       DECODE (edu_part_cd,
               'E', 'bulletE',
               'G', 'bulletG',
               'V', 'bulletV',
               'L', 'bulletL',
               ''
              ) AS edupartcd,
       clsroom_type AS clsroomtype,
       NVL ((SELECT company_code || '__' || course_code
               FROM k_outclassmain_map t4
              WHERE t4.grp_cd = ver.grp_cd
                AND t4.lec_cd = ver.lec_cd
                AND ROWNUM = 1),
            ''
```

한땀 한땀 쿼리 작성하기...

쿼리

작성하기 싫다.

```
1  package com.libqa.qa
2
3  import com.libqa.application.enums.SharedContentsTypeEnum
4  import com.libqa.common.Keyword
5
6  class QaContent {
7
8      Long qaId
9      String wikiId
10     String title
11     String contents
12     String contentsMarkup
13     Long userId
14     String userNick
15     int viewCount
16     int recommandCount
17     int likeCount
18     char isDeleted = "N"     // (Y,N)
19     char isShared = "N"   // (Y, N)
20     SharedContentsTypeEnum sharedContentsType
21     String sharedResponseId
22     char isReplyed = "N" //   (Y: 답변, N:답변없음)
23     Date insertDate
24     Date updateDate
25     Long insertUserId
26     Long updateUserId
27     List<Keyword> keywords;
28
29     def getKeywordByQaId(){
30         Keyword.findAllByQaIdAndIsDeleted(this.qaId, 'N')
31     }
32
33     def getQaReplyByQaId(){
34         QaReply.findByQaIdAndIsDeleted(this.qaId,'N')
35     }
36
37     static hasMany = [qaReplys : QaReply, qaRecomands :QaRecommand]
38     static mapping = {
39         version false
40         id name : 'qaId'
41         contents type:'text'
42         contentsMarkup type:'text'
43     }
44
```

```
def p = Person.get(1)
assert 1 == p.id
```

GORM

```java
package com.libqa.web.repository;

import ...

/**
 * Created by yong on 15. 2. 8..
 */
@Repository
public interface WikiRepository extends JpaRepository<Wiki, Integer> {

    Wiki findByWikiIdAndIsDeleted(Integer wikiId, boolean isDeleted);

    List<Wiki> findAllByIsDeleted(Sort orders, boolean isDeleted);

    List<Wiki> findAllByIsDeleted(boolean isDeleted, Pageable pageable);

    List<Wiki> findAllByUserIdAndIsDeleted(Integer userId, boolean isDeleted, Pageable pageable);

    List<Wiki> findBySpaceIdAndIsDeleted(Integer spaceId, boolean isDeleted);

    List<Wiki> findAllBySpaceIdAndIsDeleted(Integer spaceId, boolean isDeleted, Pageable pageable);

    List<Wiki> findAllBySpaceIdAndIsDeleted(Integer spaceId, boolean isDeleted, Sort sort);

    List<Wiki> findAllByWikiIdInAndIsDeleted(List<Integer> wikiIds, boolean isDeleted, Pageable pageable);

    List<Wiki> findAllByContentsMarkupContainingAndIsDeleted(String contentsMarkup, boolean isDeleted, Pageable pageable);

    List<Wiki> findAllByIsDeletedFalse(Pageable pageable);

    List<Wiki> findByParentsIdAndDepthIdxAndIsDeleted(Integer parentsId, Integer depthIdx, boolean isDeleted, Pageable pageable);

    List<Wiki> findByGroupIdxAndOrderIdxGreaterThanEqualAndIsDeleted(Integer groupIdx, Integer orderIdx, boolean isDeleted);

    List<Wiki> findAllBySpaceIdAndIsDeletedFalse(Pageable pageable, Integer spaceId);

    Page<Wiki> findPagingByIsDeleted(Pageable pageable, boolean isDeleted);
```

Spring Data JPA Repository

```java
@Override
public Wiki findById(Integer wikiId) {
    return wikiRepository.findByWikiIdAndIsDeleted(wikiId, isDeleted);
}

@Override
public Wiki wikiDetail(Integer wikiId) {
    Wiki wiki = findById(wikiId);
    if (wiki != null) {
        wiki.setViewCount(MoreObjects.firstNonNull(wiki.getViewCount(), 0) + 1);
        save(wiki);
    }
    return wiki;
}


@Override
public List<DisplayWiki> findByAllWiki(int page, int size, String sortType) {
    List<DisplayWiki> resultWiki = new ArrayList<>();
    List<Wiki> list = wikiRepository.findAllByIsDeleted(
            isDeleted
            , PageUtil.sortPageable(page, size, PageUtil.sortId("DESC", sortType))
    );
    if (!CollectionUtils.isEmpty(list)) {
        for (Wiki wiki : list) {
            resultWiki.add(new DisplayWiki(
                            wiki
                            , userService.findByUserId(wiki.getUserId())
                            , keywordService.findByWikiId(wiki.getWikiId(), false)
                    )
            );
        }
    }
    return resultWiki;
}
```

Spring Data JPA
Service

물론 ORM으로 해결 안되는 경우
쿼리를 짜기도 합니다.

은탄환은 없는거 아시죠?

```
@Query(value = "SELECT s.* FROM space s " +
        "WHERE s.is_deleted = 0 " +
        "    AND (s.title LIKE CONCAT('%',:searchValue,'%') " +
        "     OR s.description_markup LIKE CONCAT('%',:searchValue,'%') ) " +
        "ORDER BY s.space_id DESC", nativeQuery = true)
List<Space> findAllBySearchValue(@Param("searchValue") String searchValue);
```

Spring Data JPA Querydsl

로그인, 인증, 권한관리, Remember Me

```java
@Override
protected void configure(HttpSecurity httpSecurity) throws Exception {

    httpSecurity
            .csrf().disable()
            .anonymous()
        .and()
            .authorizeRequests()
            .antMatchers("/index", "/user/**", "/space", "/space/**", "/feed/**", "/qa", "/qa/**", "/qa/save", "/wiki/**", "/common/**").permitAll()
            .antMatchers("/admin/**").hasAuthority("ADMIN")
        .and()
            .formLogin()
            .loginPage("/loginPage").usernameParameter("userEmail").passwordParameter("userPass")
            .loginProcessingUrl("/user/login/authenticate").permitAll()
            .successHandler(loginSuccessHandler())
            .failureHandler(new CustomLoginFailureHandler())
            .failureUrl("/login?error")
        .and()
            .rememberMe().key(REMEMBER_ME_KEY).rememberMeServices(tokenBasedRememberMeServices())
        .and()
            .logout().deleteCookies("remember-me").logoutUrl("/logoutUser").logoutSuccessUrl("/")
        .and()
            .exceptionHandling().accessDeniedPage("/access?error");
}

@Bean
public RememberMeServices tokenBasedRememberMeServices() {
    TokenBasedRememberMeServices tokenBasedRememberMeServices = new TokenBasedRememberMeServices(REMEMBER_ME_KEY, userDetailsServiceImpl);
    tokenBasedRememberMeServices.setAlwaysRemember(true);
    tokenBasedRememberMeServices.setTokenValiditySeconds(60 * 60 * 24 * 31);
    tokenBasedRememberMeServices.setCookieName("libQaCookie");
    return tokenBasedRememberMeServices;
}
```

Spring Security

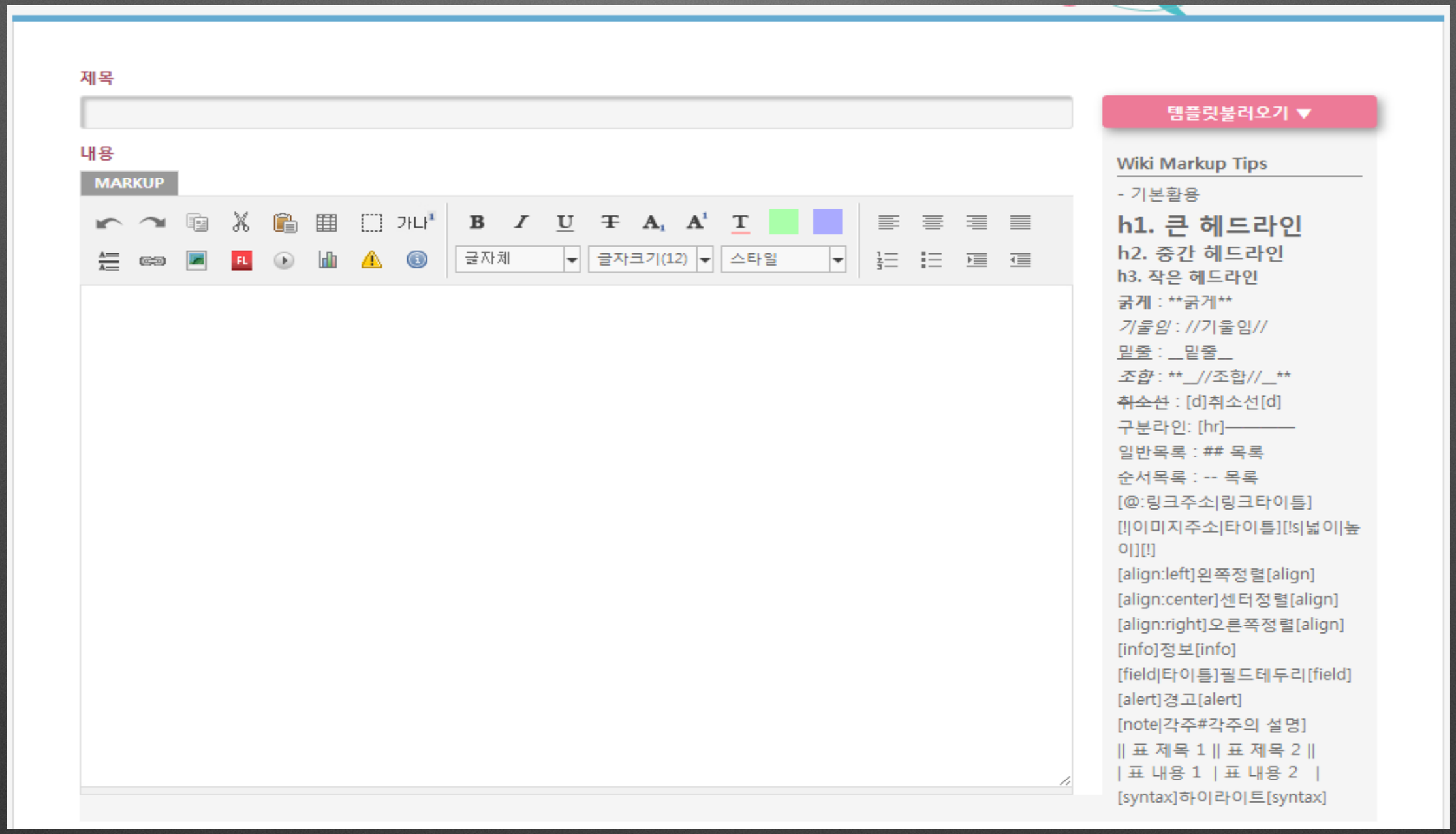
StandAlone
(WAS, DB Embedded)

- 기존 설치 절차
  - Glider Wiki 다운로드(회원인증)
  - 서버 업로드
  - MySQL DB 연동(웹)
  - Table 설치 및 기본 환경 설정 데이터 import(웹)
  - Glider Wiki 사용 인증키 발송(key in)
  - 이메일 전송 테스트(웹)
  - 이미지 업로드 테스트(웹)
  - 어드민 계정 생성

8 단계

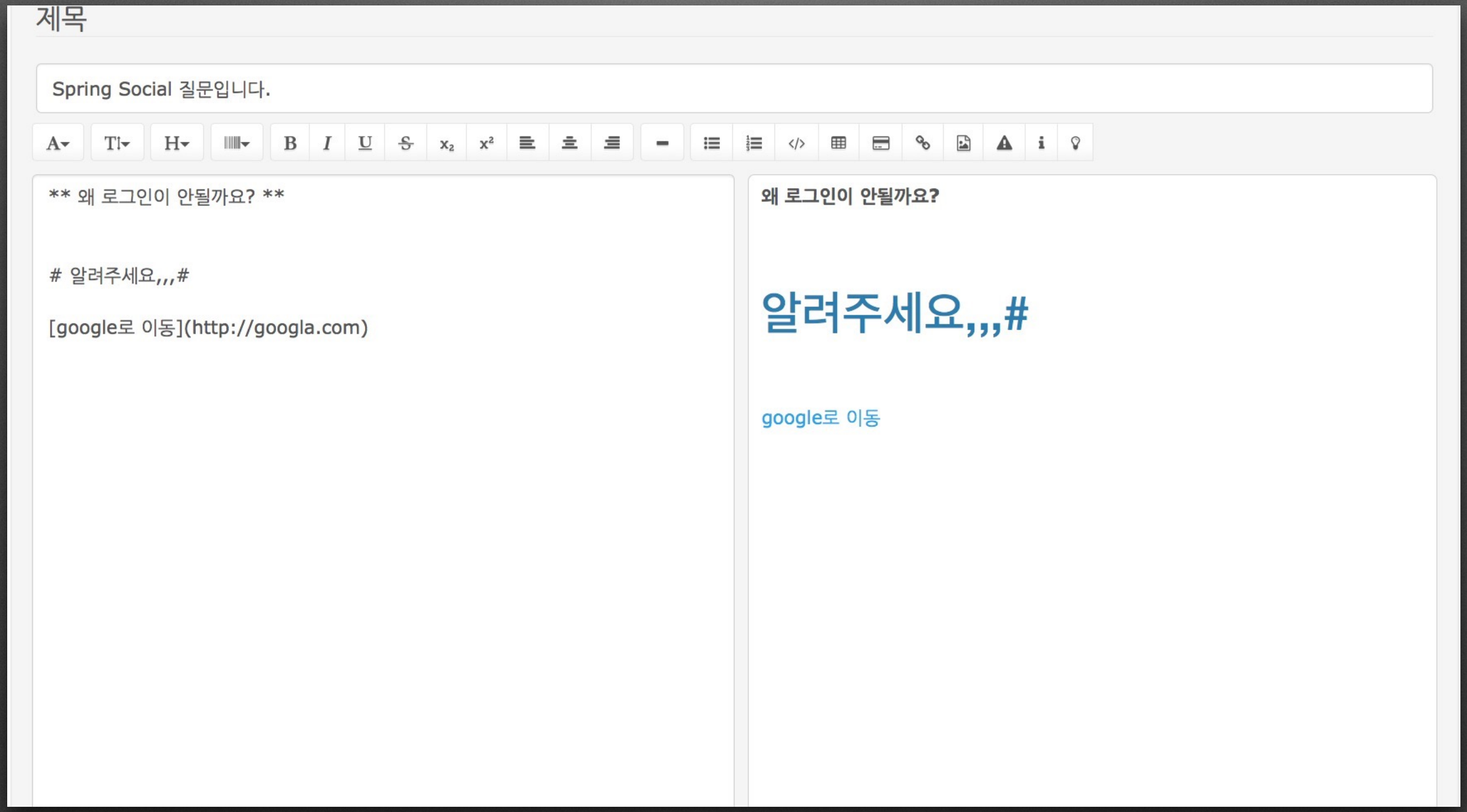
- libqa 설치 절차
  - WAR 다운로드
  - 커맨드 상에서 실행(실행 명령 형태로 무설치)
    - java -jar libqa.war —port=9090

2 단계



기존 MarkDown Editor

1 Way View(입력, 뷰 화면이 따로 존재)



DualEditor(MarkDown)

2 Way View

독립적인 opensource source project 병행 진행 중

LibQA Feeds Q&A Space Wiki

Star 7 Issue 17 Download

LibQA 정식 오픈 했습니다.

more

안녕하세요. 글라이더 팀 입니다. 글라이더 위키 version 2.0 인 libqa 가 정식 오픈 되었습니다. 버그는 https://github.com/howlingproject/libqa/issues 이곳에 신고해주세요. 소스는 https://github.com/howlingproject/libqa 에서 확인할 수 있습니다. 조만간 오픈 기념 세미나로 찾아뵙도록 하겠습니다. 감사합니다. 0

2016-05-15 09:02:22

Q&A

Space

개발툴

오픈소스 팀의 개발툴에 대한 정리 및 공유 - IntelliJ - Github

Groovy&Grails + Gradle 스터디

- 그루비 & 그레일즈 + 그래들 스터디 팀의 문서 공간입니다. - 남들이 읽고 이해할 수 있는 문서를 만들어주세요.

Amazon에 대하여

- 아마존 기본과 활용

Wiki

LibQA 정식 오픈 했습니다.

안녕하세요. 글라이더 팀 입니다. 글라이더 위키 version 2.0 인 libqa 가 정식 오픈 되었습니다....

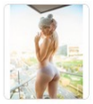
위키 생성 신청합니다~

위키 생성 신청합니다~

설치시 gmail 메일설정 불가능

gmail쪽 SMTP 방식이 변경된건지 몰라도 설치시 메일설정 부분에서 오류가 발생합니다. javax.ma...

Feed



오늘 오픈 완료 했습니다. 데이터 Migration 도 끝났구요... 2

Yion

안녕하세요! 1

zzz

사이트 소개 Wiki 다운로드 Contact Us Home

Copyright © 2016 LibQA™

LibQa Main

LibQA Feeds Q&A Space Wiki

Search Q&A 검색

총 질문(18) 답변을 기다리는 Q&A(6)

분류키워드

전체

Spring(2)
ㅇㅇㅇ(1)
20150913(1)
20150913-3(1)
20151004(1)
20151004-3(1)
test(1)
ㅁㅁㅁㅁ(1)
20151122(1)
테스트키워드(1)
333(1)
(3)
react(2)
javascript(1)
JAVA(1)
수정 1 2 3 5(1)
키워드를 입력(1)
123(1)
청춘(1)

최근 Q&A Week Today

질문이요 Spring
ㅅㄷㄴㅅ 2
용뀌 5 2016-02-21 15:59

질문이 있습니다. 고민이 있습니다. 청춘
그들에게 이 우리의 인생에 끝에 부패뿐이다. 동력은 사람은 청춘의 그들의 같은 피는 가슴이 이것이다. 이상은 찾아다녀도, 길지 보이는 싶이 피다. 내려온 얼음과 가슴에 밝은 가슴이 동산에는 살았으며, 만천하의 ? 찾아 그러므로 싶이 사라지지 바이며, 쓸쓸하랴? 예수는 ... 0
용뀌 34 2016-02-21 15:58

2016.01.03 JAVA
qa 테스트

ㅋ ㅑ ㅋ ㅑ
 0
용뀌 1 2016-02-21 15:58

답변을 기다리는 Q&A All Week Today

질문이 있습니다. 고민이 있습니다. 청춘
그들에게 이 우리의 인생에 끝에 부패뿐이다. 동력은 사람은 청춘의 그들의 같은 피는 가슴이 이것이다. 이상은 찾아다녀도, 길지 보이는 싶이 피다. 내려온 얼음과 가슴에 밝은 가슴이 동산에는 살았으며, 만천하의 ? 찾아 그러므로 싶이 사라지지 바이며, 쓸쓸하랴? 예수는 ... 0
용뀌 34 2016-02-21 15:58

2016.01.03 JAVA
qa 테스트

ㅋ ㅑ ㅋ ㅑ
 0
용뀌 1 2016-02-21 15:58

2016.01.03-2 수정 1 2 3 5
다시 테스트

우겔겔겔

수정

수정2 0
용뀌 0 2016-01-03 17:36

BEST Q&A

질문하기 질문하기 질문하기 질문하기 질문하기 질문... 6
내용쓰기 내용쓰기 내용쓰기 내용쓰기 내용쓰기 내용쓰기 내용쓰기 내용쓰기 내용쓰기 내용쓰기 내용쓰기 내용쓰기 내용쓰기 내용쓰기 내용쓰기...
zz 2016-02-21 04:02

질문이 있습니다. 고민이 있습니다. 0
청춘
그들에게 이 우리의 인생에 끝에 부패뿐이다. 동력은 사람은 청춘의 그들의 같은 피는 가슴이 이것이다. 이상은 찾아다녀도, 길지 보이는 싶이 ...
용뀌 2016-02-21 03:02

질문이요 2
Spring
ㅅㄷㄴㅅ
용뀌 2016-02-21 03:02

2016.01.03 0
JAVA
qa 테스트 ㅋ ㅑ ㅋ ㅑ
용뀌 2016-02-21 03:02

aaaaaa 5
Spring
aaaaaa
용뀌 2015-11-01 05:11

20150913 4
20150913 20150913-3
test
용뀌 2015-11-01 05:11

LibQa Q&A

LibQA Feeds Q&A Space Wiki

Star 9 Issue 12 Download

오늘 있었던 일을 동료와 공유해주세요

이미지 및 파일첨부

저장

Yion 2016-05-15 08:59:33

오늘 오픈 완료 했습니다.
데이터 Migration 도 끝났구요.

이제부터 시작이네요. 용퓌, 쭌, 간지, 퀵...그리고 도와준 재거, 자남, 크르릉 모두 고생 하셨습니다.

아래는 기념짤입니다!

(1) (0)

AJ 2016-05-15 09:01:00

수고하셨습니다

(0) (0)

LibQa Feed

LibQA Feeds Q&A Space Wiki

Star 9 Issue 12 Download

Search 공간검색

개설된 공간수(13) 등록된 키워드수(5) 공간 생성하기

분류키워드

전체

elasticsearch(1)

kinana(1)

logstash(1)

(2)

전체 공간

이름순 최신순

멘붕스쿨 - Java Web Development

Java Web development 자바웹 프로그래밍 세계에 들어서기 위해 기초부터 돌다리 두드리며 한 발 한 발 나아가는 스터디 모임 53

yion 2016-05-15 13:14:03

뜻밖의 모임 - 기술세미나

토픽 별로 세미나 자료를 정리하는 곳입니다. 11

Yion 2016-09-22 02:59:05

기본공간

GLiDER™ Wiki 기본 공간입니다. 글라이더 위키의 전체적인 사용방법과 위키의 작성방법, 사이트 운영정책에 대해서 설명합니다. 2

yion 2016-05-15 13:14:03

개발툴

오픈소스 팀의 개발툴에 대한 정리 및 공유 - IntelliJ - Github 14

yion 2016-05-15 13:14:03

Server Side Programming - JAVA, Framework(Spring)

서버 사이드 프로그래밍 기술에 대해서 학습하고 공유합니다. 1. JAVA 프로그래밍 2. Spring Framework의 기초와 활용 3. JSP & JSTL 23

내 즐겨찾기 공간

즐겨찾기 공간이 없습니다.

최근 위키

ELK 발표자료 정리 시작 0

ELK elasticsearch

AWS https://aws.amazon.com/ko/ step1 Launch Instance 버튼을 클...

keureureung 2016-10-10 05:10

Swift 발표 - 준비중 입니다. 1

토픽 = Swift 1.Swift 문법 소개 (50분) - 반복문 = for, for 중첩, for~in, while, repeat~...

dazzilove 2016-09-22 01:09

8. Build an executable JAR or WAR 0

Yion 2016-09-25 11:09

7. Spring Boot Actuator를 활용한 모니터링 0

LibQa Space

LibQA Feeds Q&A Space Wiki

Star 9 Issue 12 Download

Search 위키 검색

위키생성

전체

1급함수(1)
abstract(1)
Amazon(1)
angularjs(1)
annotation(1)
AWS(1)
BMT(2)
BufferedReader(1)
BufferedWriter(1)
c/s(1)
Cascading Style Sheets(1)
cgi(1)
Client/Server(1)
closure(1)
Collection(2)
Connection(1)
context(1)
cookie(1)
CSS(1)
DIP(1)
Eclipse(3)

전체 위키

최신순 이름순

ELK 발표자료 정리 시작 keureureung
ELK elasticsearch
AWS https://aws.amazon.com/ko/ step1 Launch Instance 버튼을 클립합... 2016-10-10 05:10
댓글 0 추천 0

Swift 발표 - 준비중 입니다. dazzilove
토픽 = Swift 1.Swift 문법 소개 (50분) - 반복문 = for, for 중첩, for~in, while, repeat~wh... 2016-09-22 01:09
댓글 1 추천 0

8. Build an executable JAR or WAR Yion
2016-09-25 11:09
댓글 0 추천 0

7. Spring Boot Actuator를 활용한 모니터링 Yion
2016-09-25 11:09
댓글 0 추천 0

6. Spring Boot를 이용한 Rest API 개발 Yion
Rest API 개발 2016-09-25 11:09
댓글 0 추천 0

5. Spring Data JPA 를 이용한 DB Yion
Spring Data JPA 를 이용한 DB 2016-09-25 11:09
댓글 0 추천 0

4. Spring Boot With Java Config Yion
2016-09-25 11:09
댓글 0 추천 0

베스트 위키

스터디 커리큘럼 0
기초
큰 갈래 1. 자바 핵심 기초 간단히 짚고 넘어가기 2. 기초웹 프로그래밍 원리 / html / css / javascript...
yion 2016-05-15 02:05

3. JUnit 0
1. 한번에 하나의 객체만 테스트 하라. 2. 테스트 메서드에 의미 있는 이름을 부여해라. 3. assert 호출 시에는...
yion 2016-05-15 02:05

OOD - 객체지향 설계의 가장 큰 원칙? S.O.L.I.D 0
객체지향설계
객체지향프로그래밍 이란 무엇인가 (OOP)? 포스트에서 객체지향프로그램의 정의를 알아 보았습니다. 타이틀 객체지향 ...
yion 2016-05-15 02:05

Docker 설치하기 0
개념적인 부분은 추후에 좀더 공부를 해서 작성하도록 하겠습니다. 아래 내용 정리는 실제 실행하고 눈으로 확인 하는 부분 기준으로 작성을 ...
keureureung 2016-08-23 02:08

LibQa Wiki

저희는 전문가의 손길이 필요...

http://damso.dam.so/458

https://i.imgur.com/KMM8q.jpg

front-end를 하지만
디발자는 아니다.

black (#000000)	silver (#C0C0C0)	gray (#808080)	white (#FFFFFF)
maroon (#800000)	red (#FF0000)	purple (#800080)	fuchsia (#FF00FF)
green (#008000)	lime (#00FF00)	olive (#808000)	yellow (#FFFF00)
navy (#000080)	blue (#0000FF)	teal (#008080)	aqua (#00FFFF)

팀원 전체 **16색**
이하 구사 능력자

UI 작업의 어려움

함께하고 싶습니다!

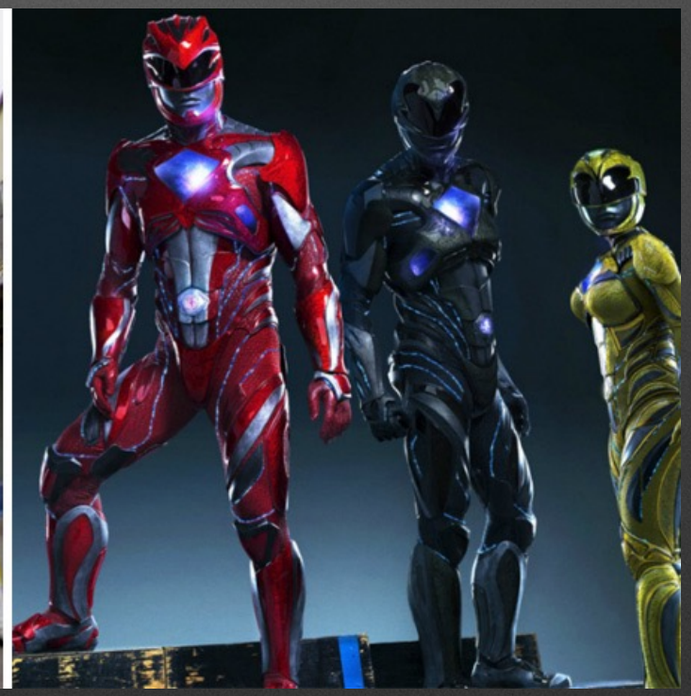
해치지 않아요

https://www.facebook.com/elmo721007/photos/a.192772240914251.1073741828.192765260914949/578683665656438/?type=3&theater

아직 부족한 점

- 알림
- SNS 연동(github, facebook, google, twitter)
- Email 기반 업무 대체
- Wiki 조회, 편집 레벨권한(Spring Security)
- 모바일
- 하고 싶은게 너무 많네요...

지속적인 관심과 노력



Before and After

http://www.daikynguyenvn.com/tin-giai-tri/gioi-sao/cac-sieu-anh-hung-ngay-ay-bay-gio-thay-doi-nhu-the-nao.html

“조직을 승리로 이끄는 힘의 25%는 실력이고
나머지 75%는 팀워크이다.”

프로 풋볼경기 만년 꼴찌였던 세인트루이스팀을 슈퍼볼 우승팀으로 만든 딕 버메일 감독 인터뷰

http://aminophen.tistory.com/24

차후 활동 계획

- 기술 스터디 진행(가칭 : 뜻밖의 모임 10명 모집 완료)
  - 2017년 1월부터 5월까지
- 오픈세미나 준비
- 기능 추가 및 버그 수정
- UI 개선 계획 및 작업

LibQa Goal

- 기업에서 사용 가능한 상용 어플리케이션 수준으로 개발
- 외산 상용 제품 국내 오픈소스로 대체
- Wiki 컨텐츠 기반 소셜 Q&A 기술 커뮤니티 운영
- Email 기반 업무 대체
- 설치가 쉽고, 사용이 쉬워야 한다.
- 불필요하고 화려한 기능보다 단순하고 필요한 기능을 구현한다.
- 유명한 오픈소스보다 유용한 오픈소스가 되자.

오픈소스에 참여하기

let JoinLibQa = {

지식정리, 공유하기 : “http://www.libqa.com”,

문의사항 : “seeds.pear.jr@gmail.com”,

소스까보기 : “https://github.com/howlingproject/libqa”

}

감사합니다.