

표준프레임워크 핵심 이해와 활용 (v3.5 ~ v3.8)

2019. 06. 25
장동한(jdh830@gmail.com)

목차

1. 표준프레임워크 3.5

- 1) 특징 및 핵심기능
- 2) 신기능 · 개발팁 및 활용

2. 표준프레임워크 3.6

- 1) 특징 및 핵심기능
- 2) 신기능 · 개발팁 및 활용

3. 표준프레임워크 3.7

- 1) 특징 및 핵심기능
- 2) 신기능 · 개발팁 및 활용

4. 표준프레임워크 3.8

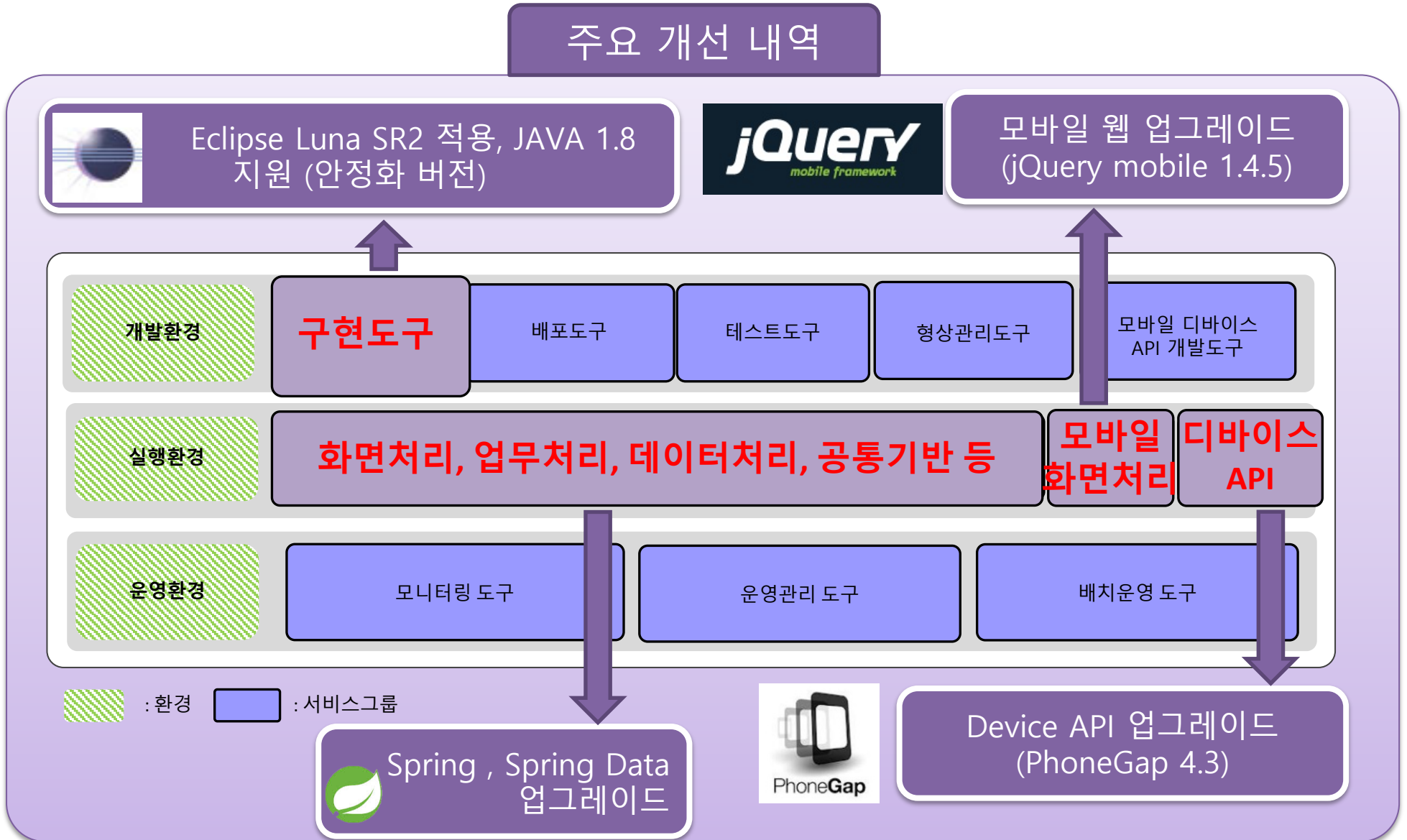
- 1) 특징 및 핵심기능
- 2) 신기능 · 개발팁 및 활용

1. 표준프레임워크 3.5

- 1) 특징 및 핵심기능
- 2) 신기능 · 개발팁 및 활용

특징 및 핵심기능 (1/3)

1. 표준프레임워크 v3.5



특징 및 핵심기능 (2/3)

- 전자정부 표준프레임워크를 구성하는 실행환경 오픈소스 업그레이드 최신화 반영
- 개발환경 개선 통해 개발 생산성 향상
- 모바일/공통 컴포넌트 버그 패치를 통하여 안정성 향상
- 실행환경 새로운 기능에 대해 예제 추가 및 보강

오픈소스 업그레이드

- ✓ 표준프레임워크 실행환경 오픈소스 중 6종 오픈소스 업그레이드 반영

주요 개선 기능


- ✓ 개발환경 Eclipse Luna SR2 적용으로 생산성 향상
- ✓ Eclipse 최신 버전 플러그인(pmd, ant, maven, emma, android configurator) 도입으로 기능 향상
- ✓ JDK 1.8 도입으로 최신화 보장

모바일/공통 컴포넌트 버그패치

- ✓ 모바일공통컴포넌트, 공통컴포넌트에서 리포팅된 버그를 패치하여 안정성 보장

실행환경 예제 보강

- ✓ 실행환경 새로운 기능에 대해서 8종 예제 추가



표준
프레임워크
3.5

특징 및 핵심기능 (2/3)

구분	서비스그룹	오픈소스SW	버전 3.0	버전 3.5
개발환경	구현도구	Eclipse	4.3.2	4.4.2
	테스트도구	Emma	1.5.2	2.3.2
	배포도구	Maven	3.0.4	3.2.1
실행환경	공통기반	Spring Framework	3.2.9	4.0.9
		Log4j	2	2.1
		PhoneGap	2.9	4.3
		jQueryMobile	1.3.2	1.4.5
	데이터처리	Spring Data JPA	1.4.3	1.4.3

- 빠른 실행 기능, 클린업기능으로 실행 속도 향상
- JAVA 8 지원
- eGovframe Template Project 웹 보안취약점 보완

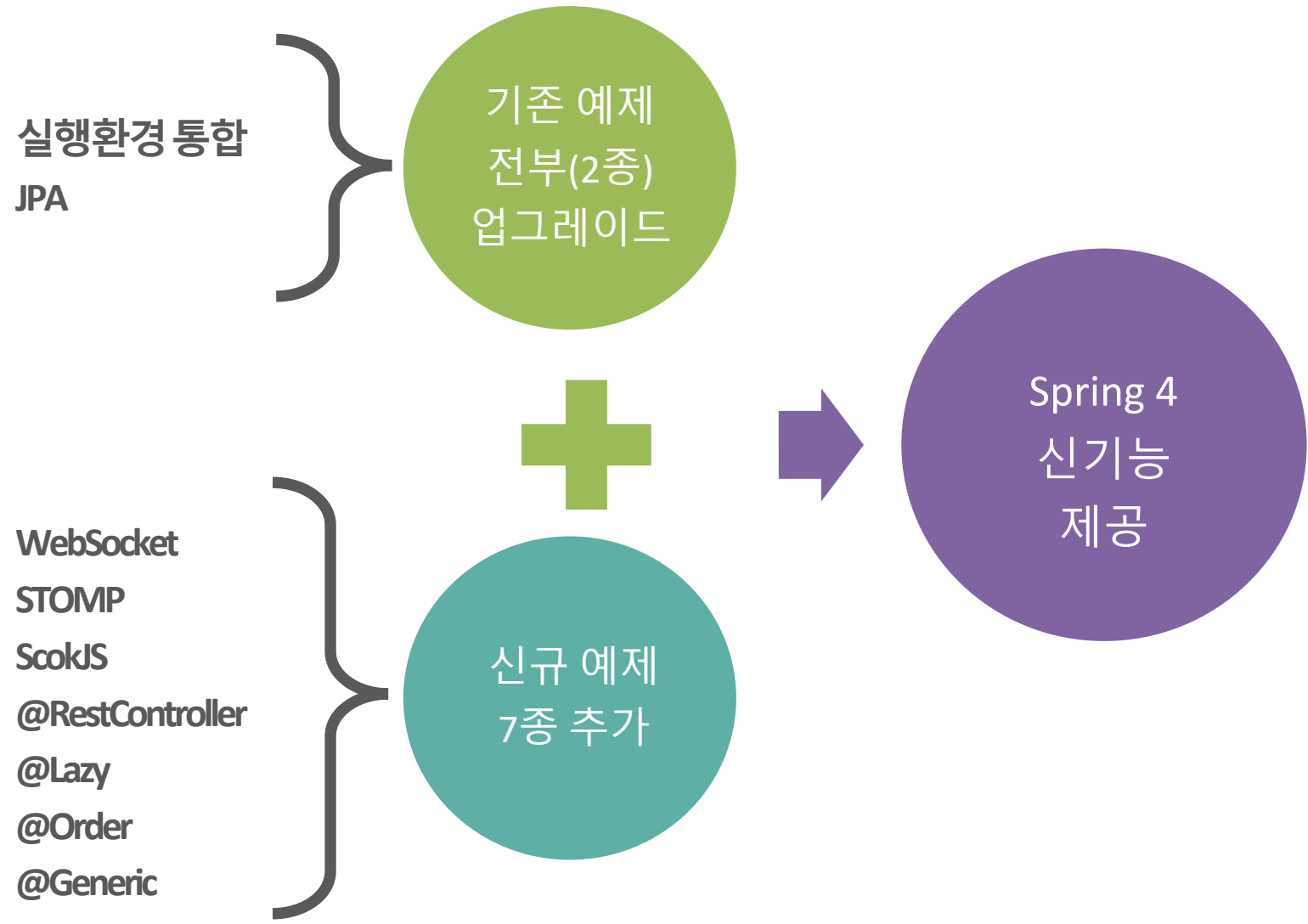
Eclipse 4.4.2 주요 개선 내역

- @Generic, @Order, @Lazy 기능이 강화 되어 Bean 식별성 증대
- WebSocket, SocketJS, STOMP 지원을 통한 외부 통신 기능 강화
- @RestController 도입으로 MVC 기능 강화

Spring 4.0.9 주요 개선 내역

PhoneGap 4.3 주요 개선 내역

- 최신의 다양한 스마트폰 플랫폼 지원
(iOS, Android, Windows Phone, Balckberry, Ubuntu, Firefox OS)
- CLI를 통한 PhoneGap 관리 기능 지원
- PhoneGap 구성 기본 플러그인 추가 제공
- PhoneGap 내장 Library가 소스 프로젝트 형태로 지원



● Web Socket

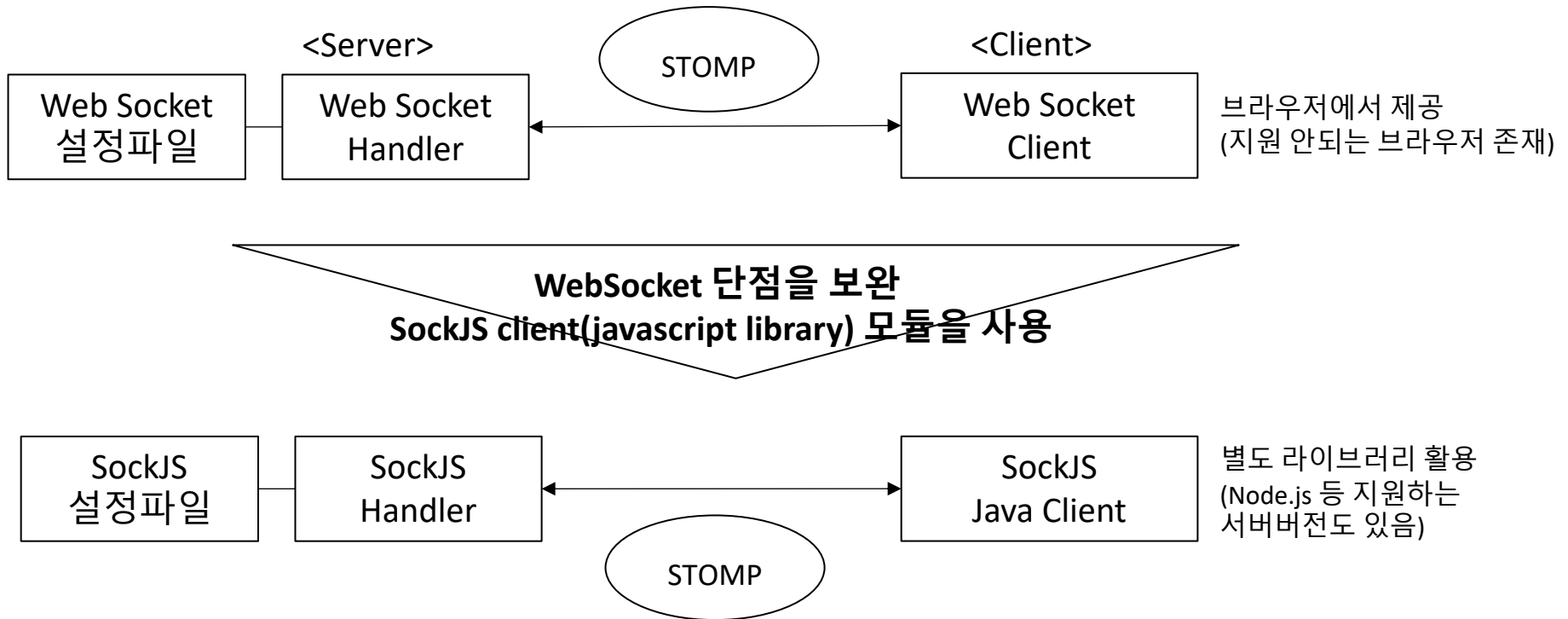
- W3C(World Wide Web Consortium)와 IETF 제정
- 웹 서버와 웹 브라우저 간의 통신을 위한 규정을 정의한 쌍방향통신(Duplex)용 기술 규약
- HTTP에 비해 간략한 header로 이루어진 overhead가 작은 가벼운 TCP 프로토콜 API로 활용

● SockJS

- SockJS 는 application 으로 하여금 WebSocket API 를 사용하는데 있다.
- 약 WebSocket 사용이 불가능한 경우에도 이를 fallback option 으로 제공하여 어떠한 코드 변화없이 WebSocket API 를 사용
- 브라우저의 지원 여부에 따라 WebSocket, HTTP Streaming, HTTP Long Polling 기술을 사용하여 목적지에 통신

● STOMP

- STOMP는 text 지향의 message protocol 기반으로 Websocket을 이용한 Message Handling을 쉽게 할수 있는 서비스를 제공
- WebSocket이나 SockJS의 경우 onmessage function에서 받는 메시지를 모두 handling해야되는 단점을 가지고 있다. 이러한 단점을 구독(subscription)과 사용자 구독(user)를 통해서 처리할 수 있다는 장점을 가지고 있다.



● Web Socket – Handler 예제

```
public class EchoHandler extends TextWebSocketHandler {
private static final Logger LOGGER = LoggerFactory.getLogger(EchoHandler.class);
    @Override
    public void afterConnectionEstablished(WebSocketSession session) throws Exception { LOGGER.info("%s 연결 됨\n",
session.getId()); }

    @Override
    protected void handleTextMessage(
        WebSocketSession session, TextMessage message) throws Exception { LOGGER.info("%s로부터 [%s] 받음\n",
session.getId(), message.getPayload());          session.sendMessage(new TextMessage("echo: " +
message.getPayload())); }

    @Override
    public void afterConnectionClosed(
        WebSocketSession session, CloseStatus status) throws Exception {
        LOGGER.info("%s 연결 끊김\n", session.getId());    }}

```

```
<beans xmlns=http://www.springframework.org/schema/beans
<websocket:handlers>
    <websocket:mapping path="/myHandler" handler="myHandler"/> </websocket:handlers>
<bean id="myHandler" class="org.springframework.samples.EchoHandler"/>
</beans>
```

● Web Socket – 클라이언트

```
<script type="text/javascript">
$(document).ready(function() {
    $('#sendBtn').click(function() { sendMessage(); });    });

var wsocket;
function sendMessage() {
    wsocket = new WebSocket("ws://localhost:8080/example/echo-ws");
    wsocket.onmessage = onMessage;
    wsocket.onclose = onClose;
    wsocket.onopen = function() {
        wsocket.send( $("#message").val() );    };
}

function onMessage(evt) {
    var data = evt.data;
    alert("서버에서 데이터 받음: " + data);
    wsocket.close();
}

function onClose(evt) {
    alert("연결 끊김");    }
</script>
```

- Socket JS – 클라이언트

```
<script type="text/javascript">
var socket;

$(document).ready(function() {
});

function sendMessage() {
    socket = new SockJS('http://localhost:8080/example/echo-ws');
    socket.onmessage = onMessage;
    socket.onclose = onClose;
    socket.onopen = function() {
        socket.send( $("#message").val() );
    }
}

function onMessage(message) {
    alert("서버에서 데이터 받음: " + message.data);
    socket.close();
}

function onClose(evt) {
    alert("연결 끊김");
}
</script>
```

● RestController

- 스프링3 버전에서는 Controller > 메소드에 @ResponseBody 어노테이션을 적용하여
- 스프링4 버전부터 지원하는 @RestController라는 어노테이션을 사용하여 컨트롤러의 모든 메소드는 자동적으로 @ResponseBody 어노테이션이 적용된 것처럼 동작

```
@RestController
public class RestExampleController {

    @RequestMapping("/hello")
    public String sayHello() {

        return "Hello Rest Controller !!!";

    }

}
```

- @Lazy 개선

- @Lazy 어노테이션은 이제 @Bean 정의뿐만 아니라 주입 지점에도 사용할 수 있음

```
@Configuration
public class FooConfig {
    @Bean
    @Lazy //자연로딩, Bean > 실제사용될 때 로딩
    public Bar bar(){
        return new Bar();
    }

    @Bean //즉시로딩, FooConfig 로드될 때 Baz는 생성됨
    public Baz baz(){
        return new Baz();
    }
}
```

- @Order 개선

- @Order 어노테이션을 통해 빈의 Autowired 순서를 지정
- org.springframework.core.Ordered interface 통해 Order 정보이 확인 기능 제공

```
@Component
@Order(1)
public class RankOne implements Ranks{
// Same code as given above
}
```

- Generic 개선

- 빈 주입시 Generic Type을 지원

```
@Autowired  
Repository<User> userRepository;
```

- @Description

- 빈에 대한 주석 및 설명을 기입하기 위한 어노테이션

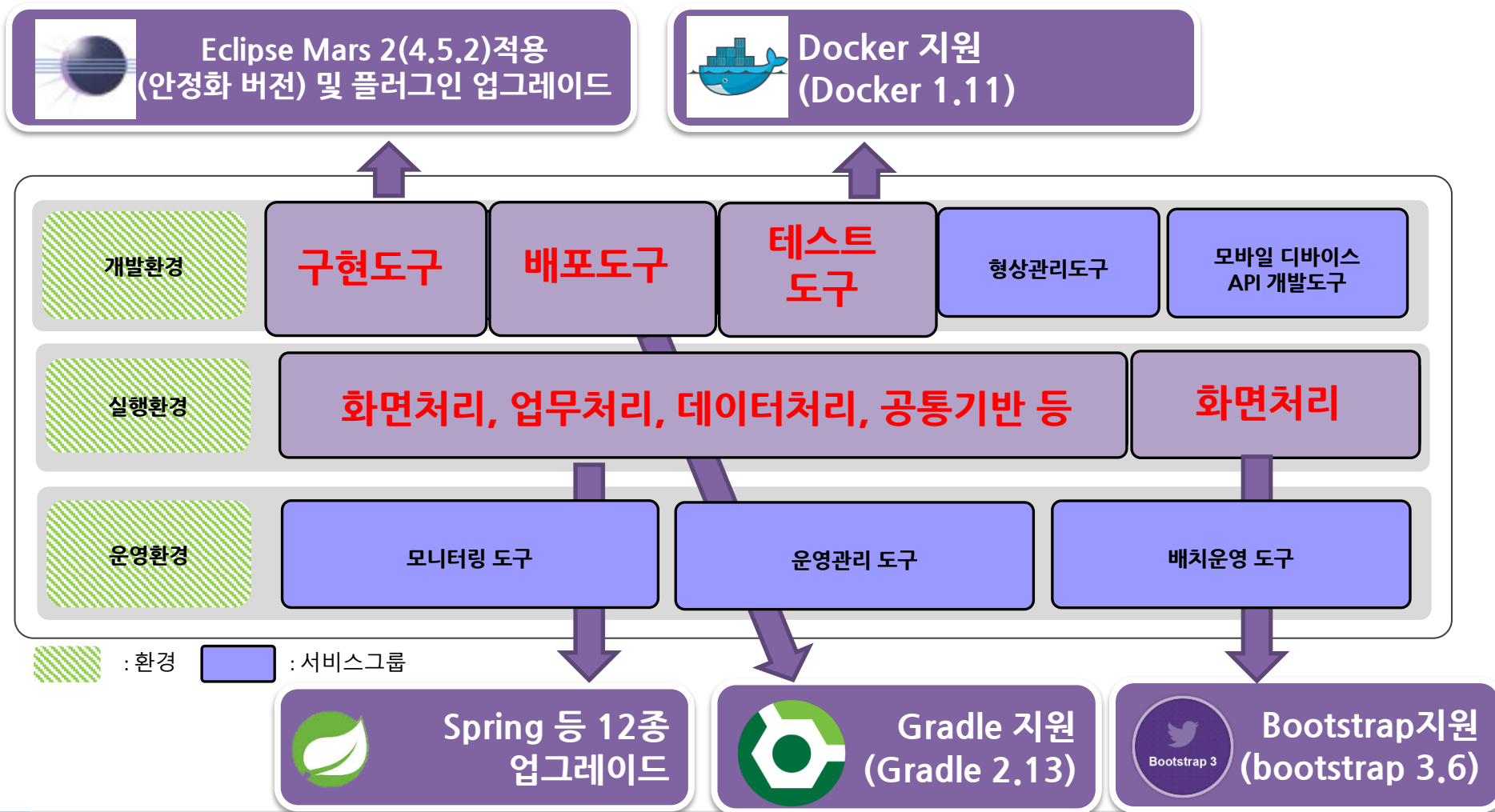
```
@Configuration  
public class AppConfig {  
    @Bean  
    @Description("Login bean")  
    public Login login() {  
        return new Login();  
    }  
}
```


2. 표준프레임워크 3.6

- 1) 특징 및 핵심기능
- 2) 신기능 · 개발팁

특징 및 핵심기능 (1/4)

주요 개선 내역



특징 및 핵심기능 (2/4)

주요 업그레이드 현황 : 실행환경 12종, 개발환경 3종 오픈소스 업그레이드

구분	서비스그룹	오픈소스SW	버전 3.5	버전 3.6
개발 환경	구현도구	Eclipse	4.4.2	4.5.2
	테스트도구	Emma	2.3.2	2.3.2
	배포도구	Maven	3.2.1	3.3.3
실행 환경	공통기반	Spring Framework	4.0.9	4.1.2
		Log4j	2.1	2.5
		PhoneGap	(3.5.1) 6.1	6.4.x
		jQueryMobile	1.4.5	1.4.5
	데이터처리	Hibernate	3.4	4.3.11

Eclipse 4.5.2 주요 개선 내역

- ✓ Eclipse 4.5.2(Mars)버전 도입으로 이전 버전 대비 실행 속도 향상
- ✓ Gradle plugin지원으로 Build 및 Library구성 가능
- ✓ Maven 3.3.3지원 및 POM 에디터 자동완성 기능 강화

Spring 4.1.2 주요 개선 내역

- ✓ 분산mvc:resources기능 지원으로 웹 리소스 맵핑으로 식별성 증대
- ✓ Servlet 3.1 도입으로 web.xml을 Java Bean으로 설정 가능
- ✓ Jackson2 ObjectMapper 지원으로 개발 편의성 증대

Log4j 2.5 주요 개선 내역

- ✓ Java 8 Lambda식 로그 출력 기능 지원
- ✓ Log4j설정 정보를 Automatic Reloading 기능 지원

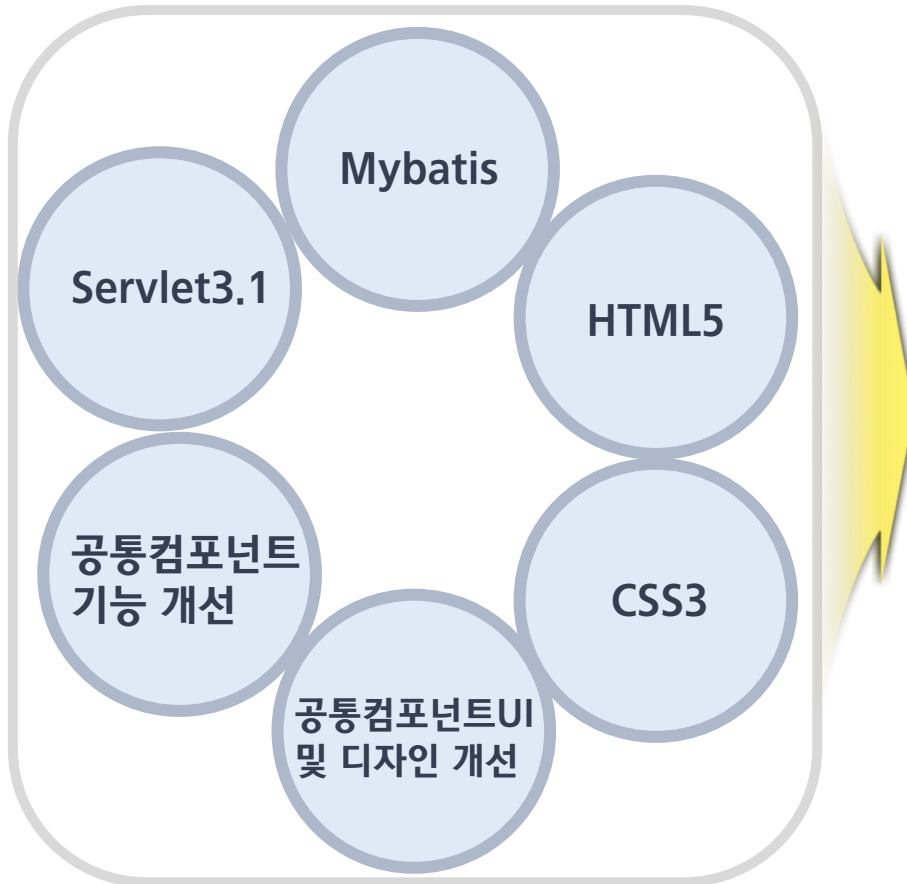
Hibernate 4.3.11주요 개선 내역

- ✓ SessionFactory 기능 향상
- ✓ Multi-tenancy 기능 지원

특징 및 핵심기능 (3/4)

최신 트렌드와 기술을 반영하여 기존 공통컴포넌트 30종을 개선하여 제공

공통컴포넌트 30종 선별



구분	공통컴포넌트
현업(15)	게시판(6) : 게시판, 공지사항, 유효게시판, 익명게시판, 댓글, 스크랩기능 동호회(3) : 커뮤니티, 커뮤니티 방명록, 커뮤니티 사진첩 약관관리(1) : 약관관리 주소록/명함록(1) : 주소록 일정관리(4) : 부서일정관리, 일정관리, 일지관리, 전체일정
사용자 디렉토리/ 통합인증(1)	일반로그인
사용자지원 (14)	온라인참여(2) : 상담관리, 온라인Poll 온라인헬프(6) : 도움말, 용어사전, FAQ, Q&A, 행정전문용어사전, 온라인매뉴얼 정보제공/알림(4) : 뉴스, 사이트관리, 추천사이트, 행사/이벤트/캠페인 개인화(1) : 마이페이지 사용자관리(1):사용자관리

특징 및 핵심기능 (4/4)

- 모바일 시장의 수요 확대로 필요한 기능을 모바일 Device API 공통컴포넌트로 추가하여 10종을 iOS, Android OS 환경에 맞게 개발하여 제공
- 향상된 HTML5 Runtime 기능으로 크로스플랫폼, 크로스브라우징 처리 가능

순번	API 가이드 프로그램	설명
1	앱 푸시	안드로이드 및 iOS 푸시 서버와 연계하여 단말에서 정보 푸시
2	문서뷰어	문서 다운로드 및 문서 뷰어와 연결
3	멀티미디어 재생	음악이나 동영상 등 멀티미디어 파일 재생지원
4	QR코드/바코드	QR코드 인식하고 링크 바로가기 제공
5	웹 소스 버전 관리 및 배포	하이브리드 앱의 HTML, CSS, JS 등의 웹 리소스를 업데이트 할 수 있도록 지원
6	파일관리	단말 내부 파일 관리 지원
7	탈옥 및 루팅 감지	Ios 탈옥 및 Android 루팅 감지 *
8	소켓통신	단말과 FTP 송수신 통신을 지원
9	SQLite	기기 로컬 DB 생성 및 활용지원 *
10	ZIP파일 압축 / 해제	파일을 zip파일로 압축 및 해제 기능

● JMS Improvement

- @JmsListener 어노테이션을 통해 간편하게 Jms를 등록 가능
- jms:annotation-driven 통해 jms 설정 가능
- <jms:listener/>를 통해서 리스너의 우선순위 및 factory 설정 가능

```
@Component
public class MyService {
    @JmsListener(containerFactory = "myContainerFactory", destination = "myQueue")
    public void processOrder(String data) { ... }
}
```

```
<jms:annotation-driven/>
<bean id="myContainerFactory"
    class="org.springframework.jms.config.DefaultJmsListenerContainerFactory">
    <property name="connectionFactory" ref="connectionFactory"/>
    <property name="destinationResolver" ref="destinationResolver"/>
    <property name="concurrency" value="3-10"/>
</bean>
```

```
<jms:listener-container factory-id="myContainerFactory"
    connection-factory="connectionFactory"
    destination-resolver="destinationResolver"
    concurrency="3-10"/>
```

● Caching Improvement

- Spring의 기존 캐시 구성 및 인프라 추상화를 사용하여 JCache(JSR-107) 기반 어노테이션을 지원

번호	Spring	JSR-107	설명
1	@Cacheable	@CacheResult	캐시할 수 있는 캐싱 대상 메서드를 지정 하기 위한 어노테이션
2	@CachePut	@CachePut	캐시에 대한 내용을 갱신 하기 위한 어노테이션
3	@CacheEvict	@CacheRemove	특정 캐시에 대한 내역을 삭제 하기 위한 어노테이션
4	@CacheEvict(allEntries=true)	@CacheRemoveAll	전체 캐시에 대한 내역을 삭제 하기 위한 어노테이션
5	@CacheConfig	@CacheDefaults	캐시 환경설정을 위한 어노테이션

- @CacheConfig 지원으로 자바 클래스를 통해서 캐시 환경설정 지원

● Web Improvement

- @RequestParam, @RequestHeader 및 @MatrixVariable 어노테이션에서 JDK 1.8의 java.util.Optional 기능 지원
- HttpMessageConverter 3가지 옵션 추가(Gson, Google Protocol Buffers, Jackson based XML serialization)

```
@RequestMapping (value = "/submit/id/{id}", method = RequestMethod.GET, produces="text/xml")
public String myLogin(@PathVariable("id") String id,
    @RequestParam(value = "logout") Optional<String> logout,
    @RequestParam("name") Optional<String> username,
    @RequestParam("password") Optional<String> password,
    @ModelAttribute("myLoginModel") MyLoginModel model,
    BindingResult errors) throws MyLoginException {...}
```


신기능 · 개발팁 및 활용 - 개선공통컴포넌트(30종) (공통컴포넌트)

● CSS3, HTML5 적용

- CSS3 기반 새로운 UI에 따른 style sheet 적용
- HTML5 기반 크로스 브라우징, 웹표준, 웹호환성을 높여 웹 페이지 리뉴얼

자유게시판 - 글조회

제목	게시판 글 제목				
작성자	일반회원	작성시간	2017-11-17	조회수	3
글내용	<p>영국에체 대일리페일은 지난 9일(현지시간) 미국 방콕에 사는 어린 긴꼬리원숭이 이야기를 전했다. 미국의 길거리에서는 야생 원숭이를 흔히 볼 수 있다.</p> <p>이날 역시 6개월 된 어린 야생 원숭이는 나무에 올라 관광객의 모습을 지켜보고 있었다.</p> <p>나무 위에 매달려 있던 원숭이는 목이 말랐는지 이내 밑으로 내려와 한 관광객의 오토바이 핸들에 매달린 바닐라봉투 속 커피잔 잔을 핥았다. 원숭이는 빠르게 도망쳤고 관광객들이 음료수를 마시는 모습을 많이 봤는지 익숙하게 커피를 마시기 시작했다.</p>				

수정 삭제 답글작성 목록 스크랩



게시판 상세보기

제목	게시판 글 제목				
작성자	홍길동	작성시간	2016-06-01	조회수	17
글내용	<p>해마다 늘어나던 국민연금 가입자가 내년부터 감소세로 돌아설 것이라는 전망이 나왔다.</p> <p>저출산의 영향으로 생산 가능 활동인구가 줄어든데 따른 것으로 분석된다.</p> <p>2일 국민연금연구원의 '국민연금 중기재정전망(2016~2020)' 연구보고서를 보면, 경제활동참가율과 국민연금가입률 증가 등으로 지속해서 늘던 전체 가입자는 내년 감소할 것으로 전망됐다.</p>				
첨부파일	없음				

수정 삭제 답글작성 목록 스크랩

전체 댓글 2

홍길동 | 2016-06-01 10:37:43

안녕하세요. 공부프레임워크 쓰면서 잘 안되는 부분이 있어서 문의합니다. 공통컴포넌트 설치를 하였고 local에서 실행되는것을 확인하였습니다.

수정 삭제

내용

댓글등록

● Mybatis 적용

- Ibatis 기반에서 Mybatis 기반으로 SQL ORM OSS 변경

context-sqlMap.xml(공통컴포넌트 v3.5)

```
<?xml version="1.0" encoding="UTF-8"?>
<beans xmlns="http://www.springframework.org/schema/beans"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://www.springframework.org/schema/beans
http://www.springframework.org/schema/beans/spring-beans-3.2.xsd">
.
.
.
<!-- SqlMap setup for iBatis Database Layer -->
<bean id="egov.sqlMapClient"
class="egovframework.rte.psl.orm.ibatis.SqlMapClientFactoryBean">
<property name="configLocations">
<list>
<value>classpath:/egovframework/sqlmap/config/${Globals.DbType}/*
.xml</value>
</list>
</property>
<property name="dataSource" ref="egov.dataSource"/>
<property name="lobHandler" ref="egov.lobHandler"/>
</bean>
</beans>
```



context-sqlMap.xml(공통컴포넌트 v3.6)

```
<?xml version="1.0" encoding="UTF-8"?>
<beans xmlns="http://www.springframework.org/schema/beans"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://www.springframework.org/schema/beans
http://www.springframework.org/schema/beans/spring-beans-4.0.xsd">
.
.
.
<!-- Mybatis setup for Mybatis Database Layer -->
<bean id="egov.sqlSession"
class="org.mybatis.spring.SqlSessionFactoryBean">
<property name="dataSource" ref="egov.dataSource"/>
<property name="configLocation"
value="classpath:/egovframework/sqlmap/config/sql-map-config.xml" />
<property name="mapperLocations">
<list>
<value>classpath:/egovframework/sqlmap/com/**/*_${Globals.DbType}.xml</value>
</list>
</property>
</bean>
</beans>
```

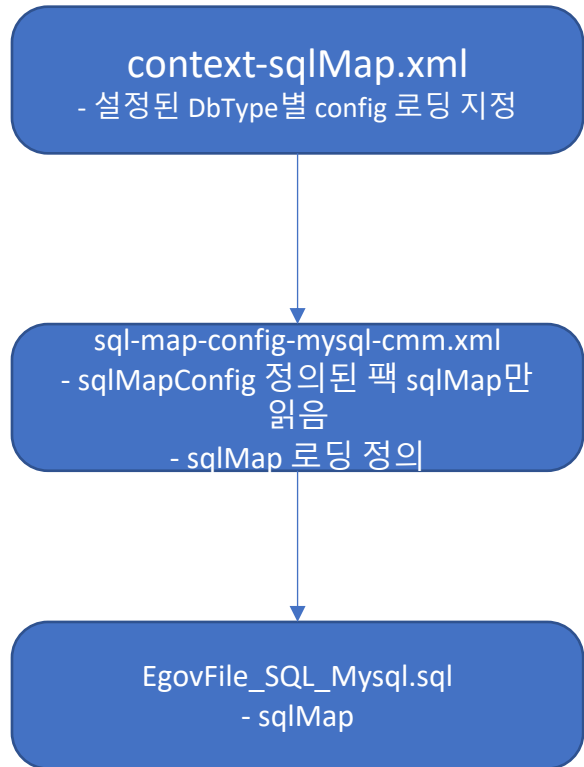
신기능 · 개발팁 및 활용 - 개선공통컴포넌트(30종) (공통컴포넌트)

2. 표준프레임워크 v3.6

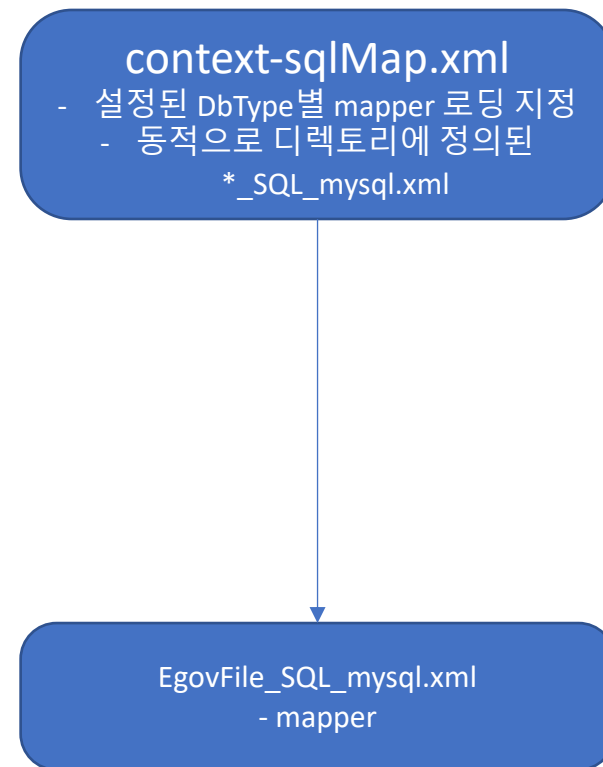
● ORM 로딩 방식 변경

- Ibatis sqlMap 로딩 방식(sqlMapConfig:정적)에서 mapper(동적) 로딩 방식으로 변경

공통컴포넌트 3.5



개선 공통컴포넌트 3.6



● Servlet 3.1 적용

- Servlet 2.5 기반에서 Servlet 3.1로 WAS 실행환경 변경
- WebApplicationInitializer 적용을 통한 동적인 web.xml 설정
 - > globals.properties 설정 파일의 Globals.Auth 속성을 통해(security, session, dummy) 권한 인증 처리

```
globals.properties
```

```
.  
. .  
. .
```

```
# 권한 인증방식(dummy, session, security) - 사용자의  
로그인시 인증 방식을 결정함  
# dummy : 더미 방식으로 사용자 권한을 인증함  
# session : 세션 방식으로 사용자 권한을 인증함  
# security : spring security 방식으로 사용자 권한을 인증함  
Globals.Auth = session
```

```
.  
. .  
. .
```



```
EgovWebApplicationInitializer
```

```
public class EgovWebApplicationInitializer  
implements WebApplicationInitializer {
```

```
.  
. .  
. .
```

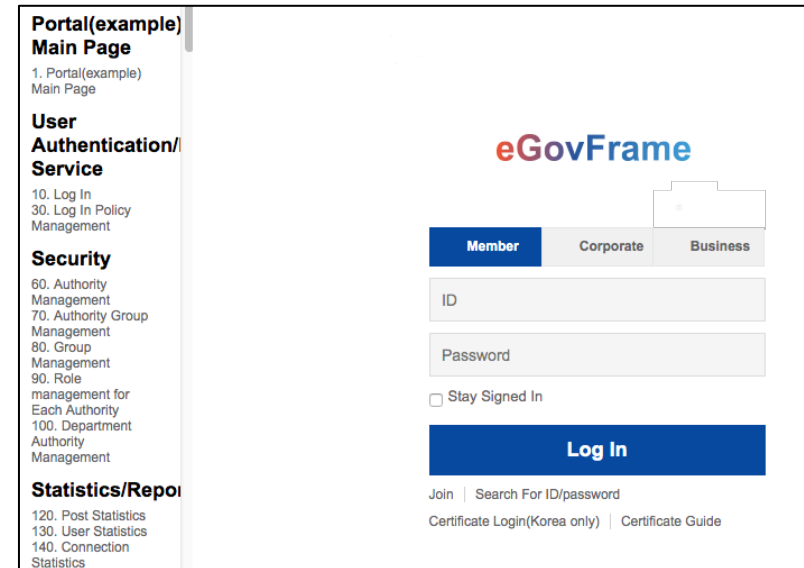
```
if("security".equals(EgovProperties.getProperty("Gl  
obals.Auth").trim())) {
```

```
...  
}
```

```
.  
. .  
. .
```

신기능 · 개발팁 및 활용 - 개선공통컴포넌트(30종) (공통컴포넌트)

- Internationalization(국제화) 적용
 - 공통컴포넌트 20종 전체 화면에 대해 영문 국제화 적용



3. 표준프레임워크 3.7



- 1) 특징 및 핵심기능
- 2) 신기능 · 개발팁

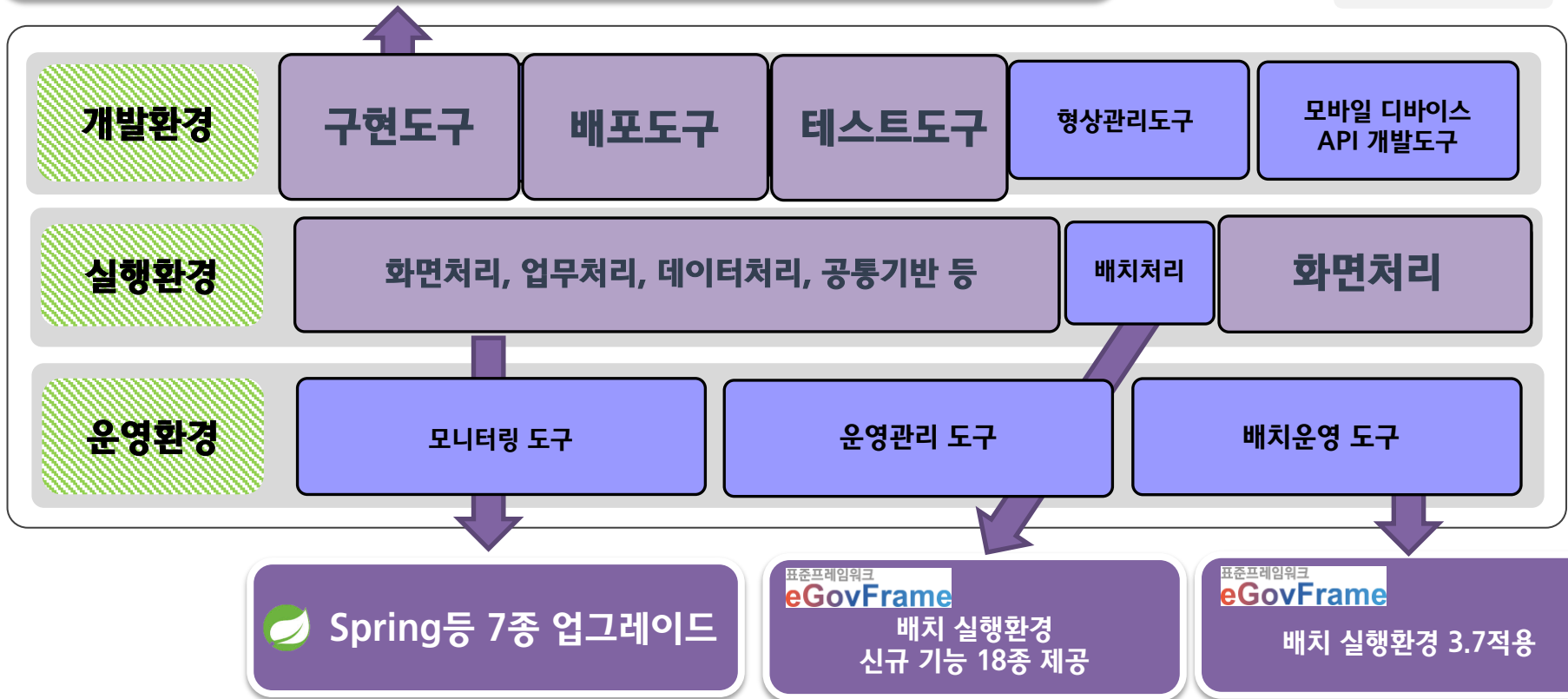
특징 및 핵심기능 (1/2)

주요 개선사항



Eclipse Neon 3 (4.6.3) 적용 (안정화 버전) 및
플러그인 업그레이드 · 배치 템플릿프로젝트 개선,
개선 된 50종 공통컴포넌트 탑재

 : 환경
 : 서비스그룹



특징 및 핵심기능 (1/4)

주요 업그레이드 현황 : 실행환경 7종, 개발환경 2종 오픈소스 업그레이드

구분	서비스그룹	오픈소스SW	버전 3.6	버전 3.7
개발 환경	구현도구	Eclipse	4.5.2	4.6.3
	테스트도구	Emma	2.3.2	2.3.2
	배포도구	Maven	3.3.3	3.3.9
실행 환경	공통기반	Spring Framework	4.1.2	4.2.4
		Spring Security	3.2.8	4.0.3
		Spring Batch	2.1.9	3.0.6
		Log4j	2.5	2.8.2
	데이터처리	Hibernate	4.3.11	4.3.11

Eclipse 4.6.3 주요 개선 내역

- ✓ Java 9 지원
- ✓ Java 에디터 기능 개선
- ✓ Code Generator, 디버그 모드 기능 강화, High-DPI 기능 지원

Spring 4.2.4 주요 개선 내역

- ✓ @interface annotation 기능 지원
- ✓ JUnit 기반 통합 테스트 규칙 설정 통해 사용 가능
- ✓ 데이터 액세스 jdbc : embedded-database 기능 강화

Spring Security 4.0.3 주요 개선 내역

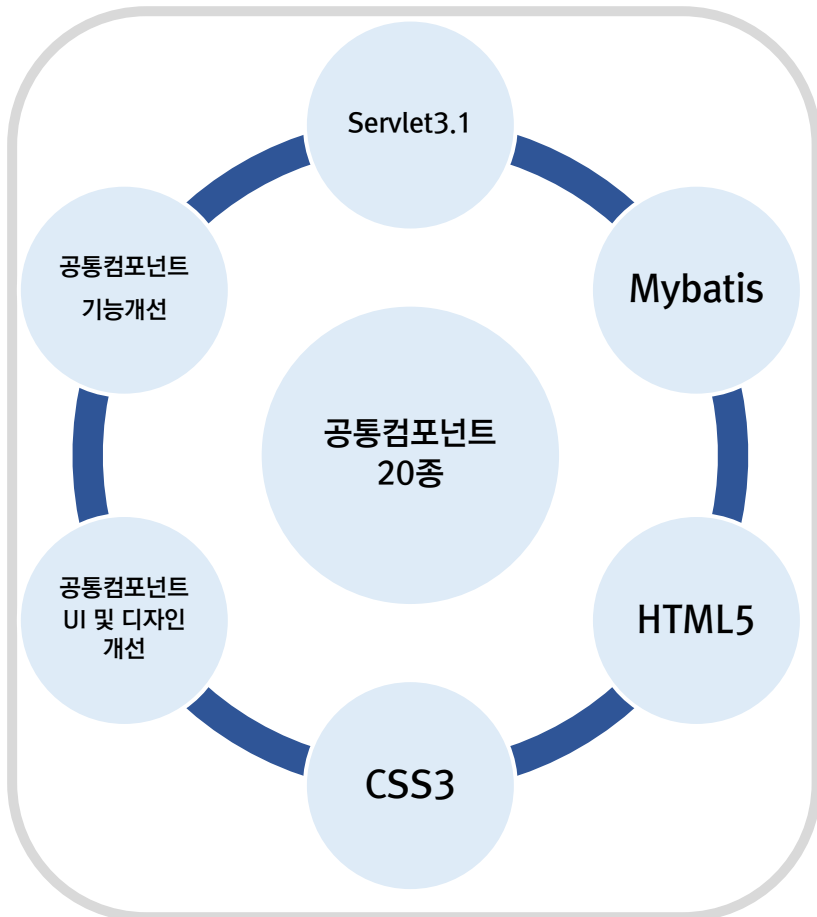
- ✓ Sniff, Clickjacking Protection 기능 제공
- ✓ XFrameOptions, XSSProtection 기능 제공

Spring Batch 3.0.6 주요 개선 내역

- ✓ Java 8 지원 및 JSR-352 기능 사용 제공
- ✓ JobScope, SQLite 지원

특징 및 핵심기능 (2/4)

최신 트렌드와 기술을 반영하여 기존 공통컴포넌트 20종을 개선하여 제공



공통컴포넌트 20종 선별

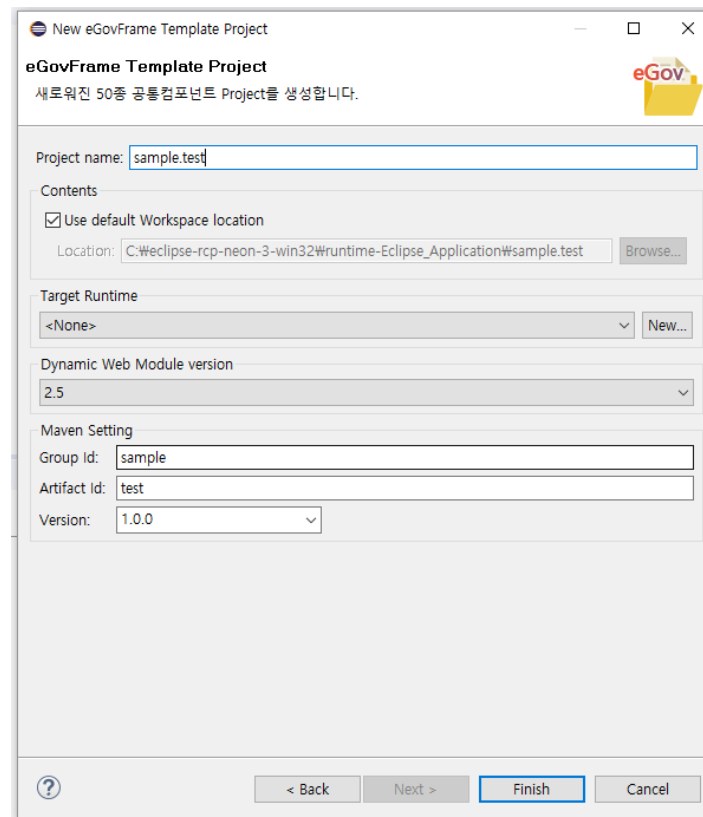
구분		공통컴포넌트
사용자지원(9)	온라인참여(6)	설문관리, 설문조사, 설문템플릿, 응답자관리, 질문관리, 항목관리
	정보제공/알림(3)	쪽지관리, 받은 쪽지함 관리, 보낸 쪽지함 관리
시스템관리(11)	공통코드관리(6)	공통분류코드, 공통상세코드, 공통코드, 우편번호 주소찾기, 우편번호관리, 도로명 주소관리
	로그관리 (5)	로그관리, 사용 로그관리, 웹 로그관리, 접속 로그관리, 개인정보 로그관리

특징 및 핵심기능 (3/4)

3. 표준프레임워크 v3.7

- 표준프레임워크 개선 공통컴포넌트 50종을 개발환경 New eGovFrame Template Project 기능으로 제공

※ 최신화 웹 호환성, 웹 접근성, UI 및 디자인 개선에 따른 생산성·가독성, 유지보수 편의성의 효과 기대



실행환경에서
모듈화된 UI제공

실행환경에서
UI 레이어 제공으로
지속적인
업그레이드 용이

UI 레이어의
컴포넌트들 간의
유기적인 결합과
상호 호환성

특징 및 핵심기능 (4/4)

• 배치 실행환경 신규 기능 18종 목록

순번	기능 명	설명
1	Remote JobLauncher	Remote 기반의 job 기능 제공
2	VSAM Processing	VSAM 파일에 대한 처리 기능 제공
3	Shell Processing	Shell 스크립트에 대한 처리 기능 제공
4	Center Cut	대용량 데이터 처리를 위해 메세지 기반 일괄 처리 기능 제공
5	File Delete	STEP에서 파일 삭제 기능 제공
6	Proxy Processing	파일 데이터와 JAVA 객체간의 IO를 Proxy 방식으로 처리하는 기능
7	Step Scope	STEP에 대한 Scope 기능 제공
8	Step Variable	STEP에서 변수를 사용할 수 있는 기능 제공
9	Job Variable	JOB에서 변수를 사용할 수 있는 기능 제공
10	Resource Variable	리소스 경로에 대한 변수를 사용할 수 있는 기능 제공
11	Code Base Exception	에러코드에 대한 에러메시지를 출력하는 기능 제공
12	Logback logging	Logback에 대한 지원으로 대용량 로깅 처리 제공
13	GDG Sequence	Flat 파일에 대한 자동 순번 처리 제공
14	JobScope	JOB에 대한 Scope 기능 제공
15	Async Item Processors	Item에 대한 비 동기 처리 기능 제공
16	SQLite	소규모에서 DB 처리 기능 OSS 제공
17	Mybatis Item Reader	Mybatis를 통해 데이터 Read 기능 제공
18	Mybatis Item Writer	Mybatis를 통해 데이터 Write 기능 제공

● Server Security

- Spring Security 3.2.4 -> 4.0.3 메이저 버전 OSS 업그레이드
- sniff : 선언된 콘텐츠 유형으로부터 벗어난 응답에 대한 브라우저의 MIME 가로채기를 방지 여부(true, false)
- xframeOptions : sniff 옵션 이 ture 일때 X-Frame-Options 범위설정(SAMEORIGIN, DENY)
- xssProtection : XSS Protection 기능의 사용 여부(true, false)
- csrf : spring security의 csrf 기능 사용 여부(false, ture)

Spring Security 처리 URL 및 파라미터 명 변경

V3.6	
로그인 처리URL	/j_spring_security_check
사용자 파라미터	j_username
비밀번호 파라미터	j_password
로그아웃 URL	/j_spring_security_logout

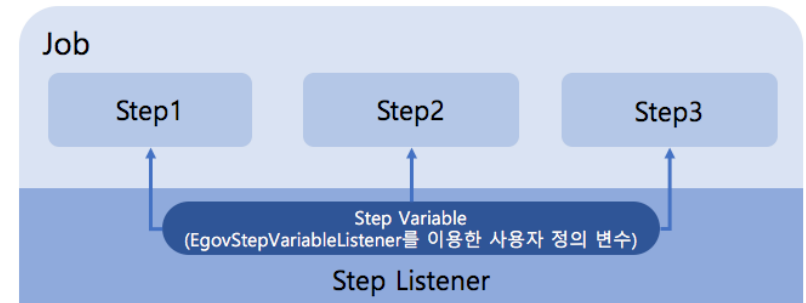


V3.7	
로그인 처리URL	/egov_security_login
사용자 파라미터	egov_security_username
비밀번호 파라미터	egov_security_password
로그아웃 URL	/egov_security_logout

신기능 · 개발팁 및 활용 - Step Variable (배치 실행환경)

● Step Variable

- Step에서 사용자 정의 변수를 사용할 수 있는 기능
- EgovStepVariableListener를 통해서 변수 선언
- Step의 listeners를 통해 변수 처리 Listener 등록



Step Variable 설정

```
<bean id="egovStepVariableListener" class="egovframework.rte.bat.support.EgovStepVariableListener">
<property name="pros">
<props>
    <prop key="StepVariableKey1">StepVariableValue1</prop>
    <prop key="StepVariableKey2">StepVariableValue2</prop>
    <prop key="StepVariableKey3">StepVariableValue3</prop>
</props>
</property>
</bean>
```

step에서 tasklet 선언시 Step Variable 사용

```
<job id="delimitedToDelimitedJob-StepVariable" parent="eGovBaseJob" xmlns="http://www.springframework.org/schema/batch">
    <step id="step1">
        <tasklet ref="taskletStep" />
        <listeners>
            <listener ref="egovStepVariableListener" />
        </listeners>
    </step>
</job>
```

● Job Variable

- Job에서 사용자 정의 변수를 사용할 수 있는 기능
- EgovJobVariableListener를 통해서 변수 선언
- Job의 listeners를 통해 변수 처리 Listener 등록



Job Variable 설정

```
<bean id="egovJobVariableListener" class="egovframework.rte.bat.support.EgovJobVariableListener">
<property name="pros">
<props>
    <prop key="JobVariableKey1">JobVariableValue1</prop>
    <prop key="JobVariableKey2">JobVariableValue2</prop>
    <prop key="JobVariableKey3">JobVariableValue3</prop>
</props>
</property>
</bean>
```

Job에서 tasklet 선언시 Job Variable 사용

```
<bean id="taskletJob" class="egovframework.example.bat.step.TaskletJob" scope="step">
    <property name="jobVariable" value="#{jobExecutionContext[JobVariableKey1]}" />
</bean>>
```

신기능 · 개발팁 및 활용 - Resource Variable (배치 실행환경)

- Resource Variable
 - step에서 사용자 리소스 정의 변수를 사용할 수 있는 기능
 - EgovResourceVariable를 통해서 변수 선언
 - ItemReader, ItemWriter에서 사용가능

EgovResourceVariable 설정

```
<bean id="egovResourceVariable" class="egovframework.rte.bat.support.EgovResourceVariable">
  <property name="pros">
    <props>
      <prop key="input.resource">file:./src/main/resources/egovframework/batch/data/inputs/csvData.csv</prop>
      <prop key="writer.resource">file:./target/test-outputs/csvOutput_ResourceVariable_{new
java.text.SimpleDateFormat('yyyyMMddHHmmssSS').format(new java.util.Date())}.csv</prop>
    </props>
  </property>
</bean>
```



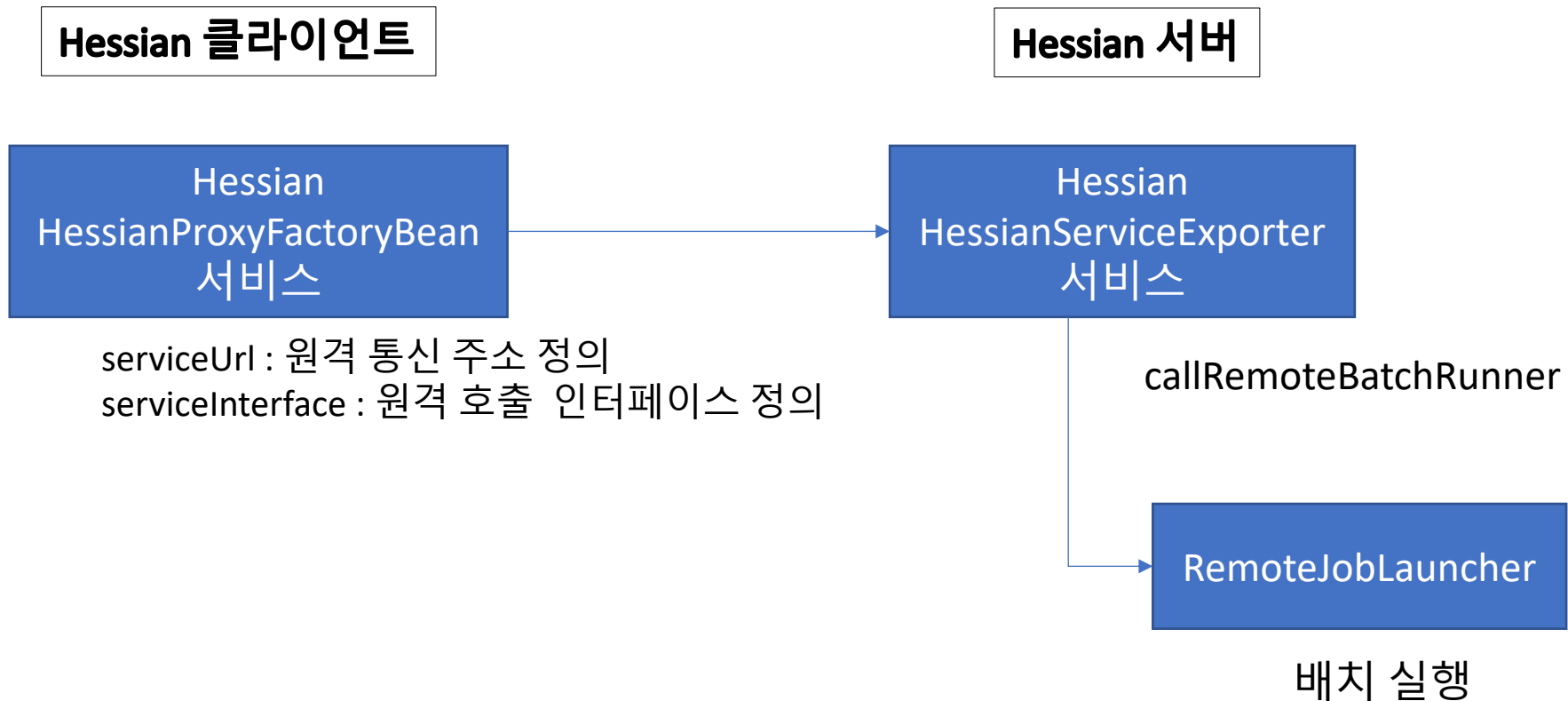
step 정의 시 리소스 사용

```
<bean id="delimitedToDelimitedJob-ResourceVariable.delimitedToDelimitedStep.delimitedItemReader"
  class="org.springframework.batch.item.file.FlatFileItemReader" scope="step">
  <property name="resource" value="#{egovResourceVariable.getVariable('input.resource')}" />
  ...
</bean>
<bean id="delimitedToDelimitedJob-ResourceVariable.delimitedToDelimitedStep.delimitedItemWriter"
  class="org.springframework.batch.item.file.FlatFileItemWriter" scope="step">
  <property name="resource" value="#{egovResourceVariable.getVariable('writer.resource')}" />
  ...
</bean>
```

신기능 · 개발팁 및 활용 - Remote JobLauncher (배치 실행환경)

3. 표준프레임워크 v3.7

- Remote JobLauncher
 - 온라인 상에서 원격으로 배치 서버의 Batch Job을 실행하는 기능
 - Hessian 오픈소스 사용



● Shell Processing

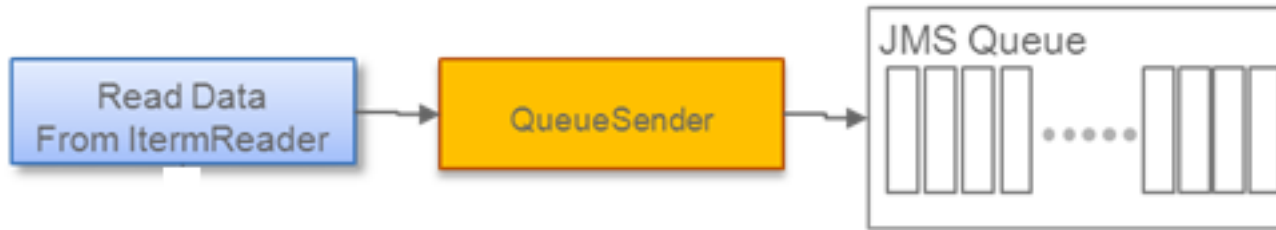
- STEP에서 리눅스의 쉘, 윈도우 배치를 실행할 수 있는 기능을 제공

```
<bean id="egovJobVariableListener" class="egovframework.rte.bat.support.EgovJobVariableListener">
<property name="pros">
  <props>
    <!-- # Windows의 경우 사용 -->
    <prop key="encoding">MS949</prop>
    <prop key="shellScript">
      <![CDATA[
        dir /a
        echo "Shell Script Execution Completed !!!"
        rem copy aa bb
        exit 0
      ]]>
    </prop>
    <!-- # Unix, Linux, MacOS 등의 경우 사용 -->
    <!-- <prop key="encoding">UTF-8</prop>
    <prop key="shellScript">
      <![CDATA[
        ls -al
        echo "Shell Script Execution Completed !!!"
        echo cp aa bb
        echo exit 0
      ]]>
    </prop> -->
  </props>
</property>
<bean id="taskletShell" class="egovframework.rte.bat.core.step.TaskletShellStep" scope="step">
  <property name="shellScript" value="#{jobExecutionContext[shellScript]}" />
  <property name="encoding" value="#{jobExecutionContext[encoding]}" />
</bean>
<job id="delimitedToDelimitedJob-ShellStep" xmlns="http://www.springframework.org/schema/batch">
  <listeners>
    <listener ref="egovJobVariableListener" />
  </listeners>
  <step id="stepShell">
    <tasklet ref="taskletShell" />
  </step>
</job>
```

신기능 · 개발팁 및 활용 - Center Cut (배치 실행환경)

- Center Cut
 - 대용량 데이터 처리를 위해 큐(Queue)를 사용하여 처리하는 기능
 - ActiveMq 오픈소스를 허용하여 큐(Queue) 처리

Unordered List Item처음 ItemReader를 사용하여 데이터를 읽고 큐에 넣은 Center-Cut Reading Step과, 읽어온 데이터를 가공 후 QueueSender를 통해 Queue에 넣는 구조이다.



Center-Cut Process Step은 큐에서 들어온 데이터를 읽고 처리 모듈(Business Proc)를 활용하여 데이터를 처리하는 구조이다.



● Code Base Exception

- 데이터베이스에 에러코드를 정의하고 이를 사용하여 에러메시지를 출력하는 기능 제공
- 에러처리코드 테이블 필요

Code Base Exception 데이터베이스 설정

```
CREATE TABLE BATCH_EXCEPTION_MESSAGE (  
    EX_ID BIGINT NOT NULL PRIMARY KEY,  
    EX_KEY VARCHAR(255) NOT NULL,  
    EX_MESSAGE VARCHAR(2500) NOT NULL  
);  
INSERT INTO BATCH_EXCEPTION_MESSAGE VALUES(1,'EGOVBATCH000001','배치실행 중 업무 관련 에러가 발생 하였습니다.');
```

Code Base Exception 사용

에러처리 생성자 파라미터 데이터베이스소스, 에러코드를 사용하여 에러처리를 할 수 있음

```
try{  
    ...  
}catch(Exception e){  
    throw new EgovBatchException(dataSource,"EGOVBATCH000001");  
    //Sql 설정시 EgovBatchException 생성자 파라미터 추가  
    //throw new EgovBatchException(dataSource,"EGOVBATCH000001","SELECT EX_MESSAGE FROM BATCH_EXCEPTION_MESSAGE WHERE EX_KEY = ?");  
}
```

● GDG Sequence

- 파일 액세스시 Resource 엘리먼트의 shell step에 shell script에 포함된 파일명에서 일련번호(index)를 사용할 수 있는 Reader를 제공
- 파일 이름이 "[이름]_NDX_[YYYYMMDDhhmmss]" 형식으로 이루어진 파일

NDX File : 파일 이름이 "[이름]_NDX_[YYYYMMDDhhmmss]" 형식으로 이루어진 파일
ex) Sample_NDX_20121126151237

NDX 일련번호 : 파일명 끝에 14자리 수의 생성시간(년월일시분초)

NDX 파일명 치환 : "[이름]_NDX(Index)" 형식의 파일명은 해당 디렉터리의 NDX 파일에 대해 Index에 해당하는 실제 파일명으로 치환됨

(-2) : 일련번호 기준 마지막 파일에서 두 번째 이전 파일명으로 치환됨

(-1) : 일련번호 기준 마지막 파일에서 첫 번째 이전 파일명으로 치환됨

(0) : 일련번호 기준 마지막 파일명으로 치환됨

(+1) : 일련번호 기준 마지막 파일에서 Index를 1 증가시켜 새로운 파일 생성

```
<bean id="fileIndex-delimitedItemReader" class="egovframework.rte.bat.core.item.file.EgovIndexFileReader">
  <property name="indexResource" value="file:./src/main/resources/egovframework/batch/data/inputs/csvData_NDX(0)" />
  <property name="lineMapper">
    <bean class="org.springframework.batch.item.file.mapping.DefaultLineMapper">
      <property name="lineTokenizer">
        <bean class="org.springframework.batch.item.file.transform.DelimitedLineTokenizer">
          <property name="delimiter" value=";" />
          <property name="names" value="name,credit" />
        </bean>
      </property>
      <property name="fieldSetMapper">
        <bean class="org.springframework.batch.item.file.mapping.BeanWrapperFieldSetMapper">
          <property name="targetType" value="egovframework.example.bat.domain.trade.CustomerCredit" />
        </bean>
      </property>
    </bean>
  </property>
</bean>
```

신기능 · 개발팁 및 활용 - Core Container Improvement (실행환경)

3. 표준프레임워크 v3.7

- @AliasFor 도입
 - Annotation 을 재사용하여 속성 값을 지정 하는 기능
 - Annotation 재활용 하여 새로운 Annotation을 만들때

```
public @interface ContextConfiguration {  
    @AliasFor ( "locations")  
    String [] value () default {};  
  
    @AliasFor ( "value")  
    String [] locations () default {};  
    // ...  
}
```

```
@ContextConfiguration  
public @interface MyTestConfig {  
  
    @AliasFor (annotation = ContextConfiguration.class, attribute = "value")  
    String [] xmlFiles ();  
  
    // ...  
}
```

신기능 · 개발팁 및 활용 - Data Access Improvement (실행환경)

3. 표준프레임워크 v3.7

- AspectJ 내에 javax.transaction.Transactional 지원
 - AspectJ에서 javax.transaction.Transactional을 지원
 - 자바빈의 annotation 통해 transaction 설정 가능

```
@Configuration
@EnableTransactionManagement(proxyTargetClass = true, mode = AdviceMode.ASPECTJ)
@EnableAspectJAutoProxy
@ComponentScan({"com.example"})
public class AppConfig {
    ...
}
```

신기능 · 개발팁 및 활용 - Data Access Improvement (실행환경)

- AspectJ 내에 javax.transaction.Transactional 지원
 - AspectJ에서 javax.transaction.Transactional을 지원
 - 자바빈의 annotation 통해 transaction 설정 가능

```
@Configuration
@EnableTransactionManagement(proxyTargetClass = true, mode = AdviceMode.ASPECTJ)
@EnableAspectJAutoProxy
@ComponentScan({ "com.example"})
public class AppConfig {
    ...
}
```

[신기능 · 개발팁 및 활용 - Testing Improvement (실행환경)

- @Commit 도입
 - 테스트시 Transaction에 대한 커밋 기능 제공
 - 특정 상황에서 데이터에 대한 저장 기능 제공

```
@Test
@Transactional
@Commit
public void commitJUnitTest () {
    //...
}
```


- @Sql 도입
 - 데이터 베이스 단위 테스트 시 선행 sql 작업 실행
 - 데이터 초기화, 특정 데이터 적재

```
@Test
@Sql("drop_schema.sql")
@Sql({ "schema.sql", "override_data.sql" })
public void sqlAnnotationTest() {
    List empNames = jdbcTemplate.queryForList("select name from emp", String.class);
    assertEquals(3, empNames.size());
}
```

4. 표준프레임워크 3.8

- 1) 특징 및 핵심기능
- 2) 신기능 · 개발팁

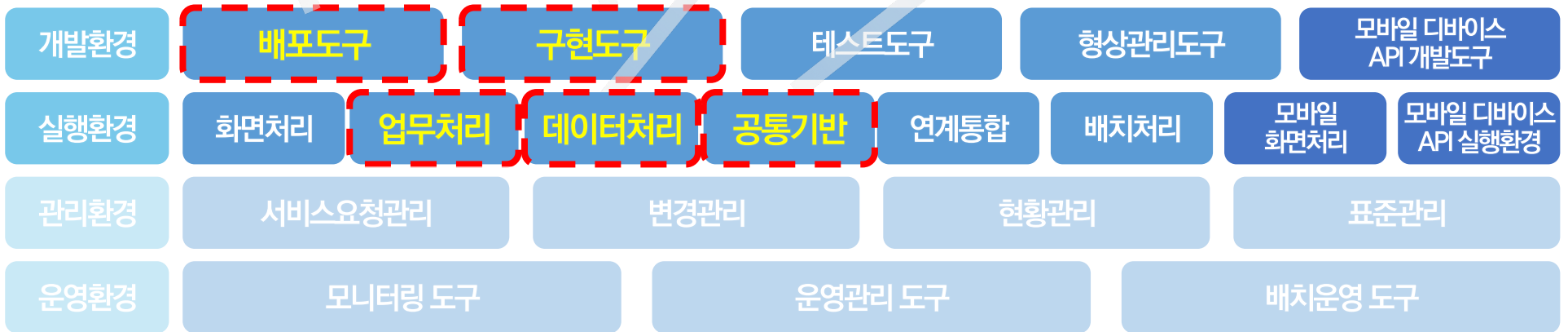
특징 및 핵심기능 (1/7)

4. 표준프레임워크 v3.8



공통컴포넌트 251종

표준프레임워크



특징 및 핵심기능 (2/7) - 개발환경

4. 표준프레임워크 v3.8

Eclipse Oxygen (4.7.3a)

- 2018년 4월 11일 출시 최신 안정화 버전
- Java 10 지원
- Java 에디터 기능 개선 (이미지파일, Generic Editor)
- 디버그 모드 기능 강화



Jenkins 2.138.4

- Hudson 에서 Jenkins으로 변경
- Hudson 라이선스 정책 변경에 따른 이용 제한 발생
- 웹 인터페이스를 통한 간편한 설정
- 지속적인 자동화 빌드 및 테스트
- 커버리지 감시 및 코드 품질 감시



표준프레임워크 3.8 플러그인 탑재

- 실행환경 v3.8
- 공통컴포넌트 251종 (신규 개선된 97종, 기 개선된 50종, UI 없는 요소기술 104종)
- DBMS 7종
- 템플릿
- 코드 자동생성 기능 (MyBatis)



특징 및 핵심기능 (3/7) - 개발환경

그룹	오픈소스 SW	표준프레임워크		비고
		3.7	3.8	
구현 도구	Eclipse	4.6.3.	4.7.3a	- 최신 자바 지원 등 이클립스 Oxygen 기반으로 최신화 •Java 8 이상에서 작동, Java 에디터 기능 개선 •Code Generator •디버그 모드 기능 강화 •Dark 테마 제공 등
배포 도구	Maven	1.7.0	1.9.0	- 최신 안정화 버전 적용
CI 도구	Hudson	3.2.2	-	- Hudson에서 Jenkins으로 변경 ※Hudson 라이선스 정책 변경에 따른 이용 제한 발생
	Jenkins	-	2.138.4	

특징 및 핵심기능 (4/7) - 개발환경

그룹	오픈소스 SW	표준프레임워크		비고
		3.7	3.8	
플러그인	Spring Core	3.8.4	3.9.4	- 최신 안정화 버전 적용
	UML2 Extension	5.2.3	5.3.0	- 최신 안정화 버전 적용
	eGovFramework 1) 실행환경 2) 공통컴포넌트 3) DBMS 지원 4) 웹사이트 템플릿 5) 자동생성 기능	3.7	3.8	1) 최신 표준프레임워크 실행환경 탑재 2) 기능이 개선된 공통컴포넌트 탑재 3) 기존 5종(Tibero, Mysql, Oracle, CUBRID, ALTIBASE) 외 추가 2종(PostgreSQL, MariaDB) 지원 4) 웹사이트 템플릿에 실행환경 v3.8 적용 및 DBMS 2종(PostgreSQL, MariaDB) 추가 지원 5) 기본 UI, 소스코드, DB스키마 자동생성 기능 반영
	MyBatipse	1.1.0	1.1.1	- 최신 안정화 버전 적용
	PMD	4.0.15	4.0.16	- 최신 안정화 버전 적용

Spring Framework (4.3.16)

- ✓ @RequestMapping Annotation 기능 강화
- ✓ Web Scopes 기능 강화
- ✓ Caching 기능 강화
- ✓ Spring Security 4.2.5와 가장 호환

Spring Security 최신 안정화버전 업그레이드 (4.2.5)

- ✓ Sniff, Clickjacking Protection 등 보안 기능 강화

MyBatis 3.4.1

- ✓ JSR-310(Java Date-Time) 지원
- ✓ YearTypeHandler, MonthTypeHandler 기능 제공

EhCache 안정화버전 업그레이드 2.10.3

- ✓ JSR-107(JCache) 지원으로 호환성 증대

서버 환경설정 파일 암호화

- ✓ 서버 환경설정 파일 암호화 통한 중요 정보 은닉성 제공
- ✓ 암호/복호화 간소화 설정을 통한 개발 편의성 향상

특징 및 핵심기능 (6/7) - 실행환경

그룹	오픈소스 SW	표준프레임워크		비고
		3.7	3.8	
공통 기반	Spring Framework	4.2.4	4.3.16	- 스프링 코어 모듈 최신 안정화 버전 업그레이드 • Spring Core 컨테이너 기능, 데이터 액세스, Web 처리 기능, Testing 기능 개선 및 강화로 개발 편의성과 시스템 처리 능력 향상
	Spring Security	4.0.3	4.2.5	- 스프링 보안 모듈 최신 안정화 버전 업그레이드 • Sniff, Clickjacking Protection 등 보안 기능 강화
	Log4j	2.8.2	2.10.0	- 최신 안정화 버전 적용 • Apache Log4j 역직렬화 취약점 이슈에 대응하여 시스템 가용성과 안정성을 향상
	MyBatis	3.3.0	3.4.1	- 최신 안정화 버전의 데이터처리 모듈 업그레이드와 공통컴포넌트 적용
	MyBatis-Spring	1.2.3	1.3.0	- 최신 안정화 버전의 데이터처리 모듈 업그레이드와 공통컴포넌트 적용
	EhCache	2.6.11	2.10.3	- 웹 데이터 캐쉬 모듈 최신 안정화 버전 적용
	JUnit	4.11	4.12	- 최신 안정화 버전의 데이터처리 모듈 업그레이드와 공통컴포넌트 적용

2018년도 개선된 UI 97종

- ✓ 통합인증 (2) : 인증서로그인, 로그인정책관리
- ✓ 보안 (5) : 실명확인, 공공i-Pin 실명확인, 그룹관리 등
- ✓ 통계/리포팅 (6) : 게시물통계, 사용자통계, 접속통계등
- ✓ 협업 (24) : 공지사항관리, 디자인템플릿, 유효게시판 등
- ✓ 사용자지원 (29) : 출퇴근관리, 저작권보호정책 등
- ✓ 시스템관리 (16) : 행정코드관리, 기관코드수신 등
- ✓ 시스템/서비스연계 (4) : 시스템연계관리, 연계현황관리, 연계메시지관리, 연계기관관리
- ✓ 디지털 자산 관리 (8) : 개인지식관리, 지식맵관리 등
- ✓ 시스템/서비스연계 (2) : OPEN-API연계서비스, 모바일MMS서비스
- ✓ 사용자지원/개인화 (1) : 모바일 메뉴관리

DBMS 7종 지원 (MyBatis)

- ✓ Tiberio, Mysql, Oracle, Cubrid, Altibase, PostgreSQL, MariaDB

HTML5, CSS3 적용

다국어 처리 (영어)

● Interface default methods 지원

- bean property 설정시 interface default methods의 getters/setters(java8 기준)

```
public interface IMyDateHolder {  
    void setLocalDate(LocalDate localDate);  
    LocalDate getLocalDate();  
    default void setStringDate(String stringDate) {  
        setLocalDate(LocalDate.parse(stringDate,  
            DateTimeFormatter.ofPattern("dd.MM.yyyy")));  
    }  
}
```

```
public MyDateHolder implements IMyDateHolder{  
    public void myShowMessage(String msg) {  
        System.out.println(msg);  
    }  
}
```

```
<bean id="dateHolder"  
    class="com.sample.MyDateHolder">  
    <property name="stringDate" value="15.10.1982"/>  
</bean>
```

신기능 · 개발팁 및 활용 - Web Improvements (실행환경)

- @GetMapping 지원
- @PostMapping 지원
- @PutMapping 지원
- @DeleteMapping 지원
- @PatchMapping 지원

번호	Annotation	설명
1	@GetMapping	웹요청 Method:GET에 대한 RequestMapping @RequestMapping(method = RequestMethod.GET)의 축약형
2	@PostMapping	웹요청 Method:POST에 대한 RequestMapping @RequestMapping(method = RequestMethod.POST)의 축약형
3	@PutMapping	웹요청 Method:PUT에 대한 RequestMapping @RequestMapping(method = RequestMethod.PUT)의 축약형
4	@DeleteMapping	웹요청 Method:DELETE에 대한 RequestMapping @RequestMapping(method = RequestMethod.DELETE)의 축약형
5	@PatchMapping	웹요청 Method:PATCH에 대한 RequestMapping @RequestMapping(method = RequestMethod.PATCH)의 축약형

```
@Controller
@RequestMapping("/appointments")
public class AppointmentsController {

    private final AppointmentBook appointmentBook;

    @Autowired
    public AppointmentsController(AppointmentBook appointmentBook) {
        this.appointmentBook = appointmentBook;
    }

    @GetMapping
    public Map<String, Appointment> get() {
        return appointmentBook.getAppointmentsForToday();
    }

    @GetMapping("/{day}")
    public Map<String, Appointment> getForDay(@PathVariable @DateTimeFormat(iso=ISO.DATE) Date day, Model model) {
        return appointmentBook.getAppointmentsForDay(day);
    }

    @GetMapping("/new")
    public AppointmentForm getNewForm() {
        return new AppointmentForm();
    }

    @PostMapping
    public String add(@Valid AppointmentForm appointment, BindingResult result) {
        if (result.hasErrors()) {
            return "appointments/new";
        }
        appointmentBook.addAppointment(appointment);
        return "redirect:/appointments";
    }
}
```

- @SessionAttribute

- 클래스 레벨에서 @SessionAttribute 사용하던 Annotation을 메소드 레벨에서 사용
- Session Attribute를 가져와 바인딩 해주는 Annotation

```
@RequestMapping("/")  
public String handle(@SessionAttribute User user) {  
    // ...  
}
```

신기능 · 개발팁 및 활용 - Core Container Improvement (실행환경)

4. 표준프레임워크 v3.8

- @RequestAttribute

- Request Attribute를 가져와 바인딩 해주는 어노테이션

```
@RequestMapping("/")
public String handle(@RequestAttribute Client client) {
    // ...
}
```

신기능 · 개발팁 및 활용 - 서버 환경설정 파일 암호화 (실행환경)

● 서버 환경설정 파일 암호화

- 서버, 웹 취약점에 의한 파일 탈취 대비하여 globals.properties 중요정보 암호화
- 서버 환경설정 파일 암호화 통한 중요 정보 은닉성 제공

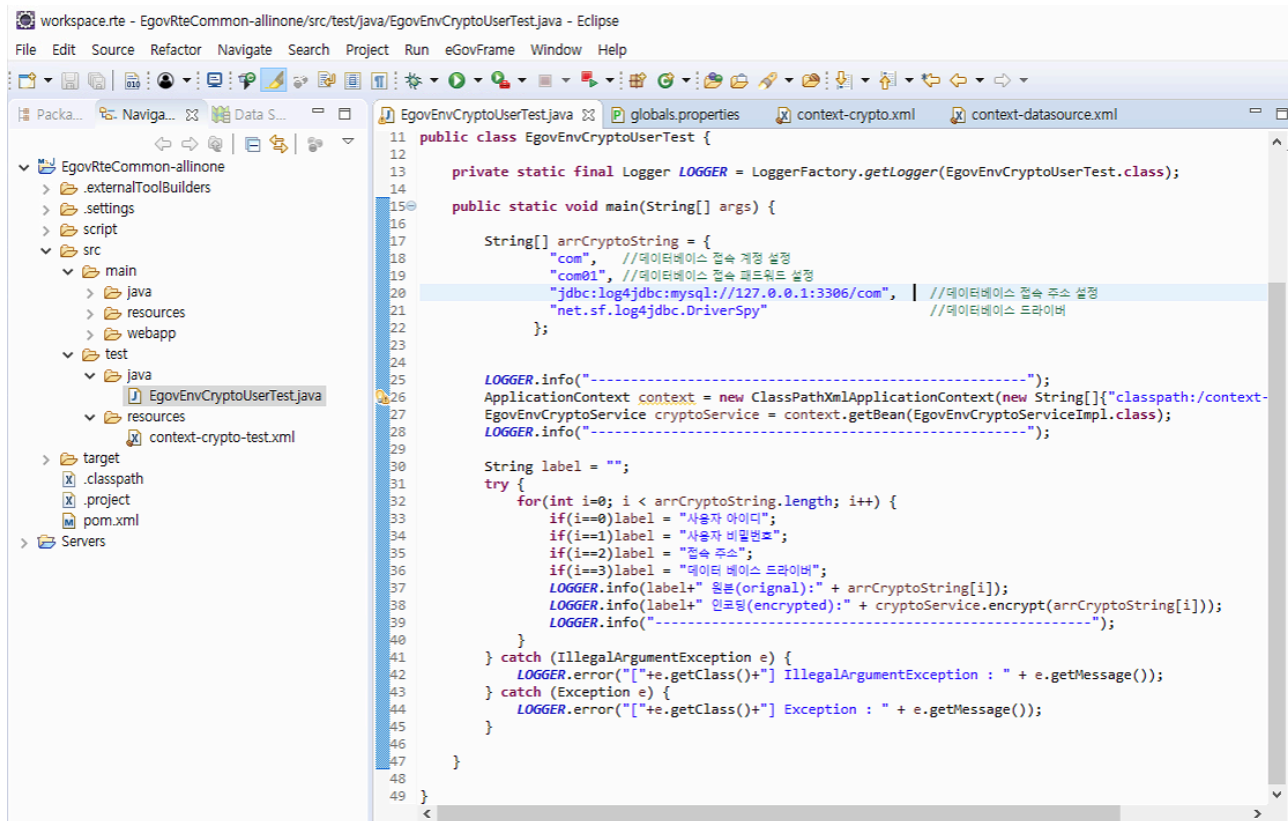
globals.properties 암호화 대상 확인

```
#mysql
Globals.mysql.DriverClassName = net.sf.log4jdbc.DriverSpy
Globals.mysql.Url = jdbc:log4jdbc:mysql://127.0.0.1:3306/com
Globals.mysql.UserName = com
Globals.mysql.Password = com01
```

● 서버 환경설정 파일 암호화

암호화 대상 키값을 암호화

(가이드를 참고하여 원하는 값을 넣어 설정 한다.)



```
workspace.rte - EgovRteCommon-allinone/src/test/java/EgovEnvCryptoUserTest.java - Eclipse
File Edit Source Refactor Navigate Search Project Run eGovFrame Window Help
EgovEnvCryptoUserTest.java | globals.properties | context-crypto.xml | context-datasource.xml
11 public class EgovEnvCryptoUserTest {
12
13     private static final Logger LOGGER = LoggerFactory.getLogger(EgovEnvCryptoUserTest.class);
14
15     public static void main(String[] args) {
16
17         String[] arrCryptoString = {
18             "com", //데이터베이스 접속 계정 설정
19             "com01", //데이터베이스 접속 패드워드 설정
20             "jdbc:log4jdbc:mysql://127.0.0.1:3306/com", //데이터베이스 접속 주소 설정
21             "net.sf.log4jdbc.DriverSpy" //데이터베이스 드라이버
22         };
23
24         LOGGER.info("-----");
25         ApplicationContext context = new ClassPathXmlApplicationContext(new String[]{"classpath:/context-
26             EgovEnvCryptoService cryptoService = context.getBean(EgovEnvCryptoServiceImpl.class);
27         LOGGER.info("-----");
28
29         String label = "";
30         try {
31             for(int i=0; i < arrCryptoString.length; i++) {
32                 if(i==0)label = "사용자 아이디";
33                 if(i==1)label = "사용자 비밀번호";
34                 if(i==2)label = "접속 주소";
35                 if(i==3)label = "데이터 베이스 드라이버";
36                 LOGGER.info(label+" 원본(original):" + arrCryptoString[i]);
37                 LOGGER.info(label+" 인코딩(encrypted):" + cryptoService.encrypt(arrCryptoString[i]));
38                 LOGGER.info("-----");
39             }
40         } catch (IllegalArgumentException e) {
41             LOGGER.error("[ "+e.getClass()+" ] IllegalArgumentException : " + e.getMessage());
42         } catch (Exception e) {
43             LOGGER.error("[ "+e.getClass()+" ] Exception : " + e.getMessage());
44         }
45     }
46 }
47
48 }
49 }
```


신기능 · 개발팁 및 활용 - 서버 환경설정 파일 암호화 (실행환경)

- 서버 환경설정 파일 암호화

globals.properties파일 mysql 접속 항목 암호화 설정

```
Globals.mysql.DriverClassName = RC2Xpum1xR5Djc9LD1tsx%2B6RmBK%2Fli2vgmOkLbTft2k%3D
Globals.mysql.Url = IPnSE1%2BSpbvetos96I7vdReyNce6grYg5BLkDzGdt9%2FbdnGiC4enjlq5upRZK%2BOR
Globals.mysql.UserName = McNplxbxwVO7%2BHNm5TfKw%3D%3D
Globals.mysql.Password = xz4fmrSdr1vGGI6UtwPLwA%3D%
```

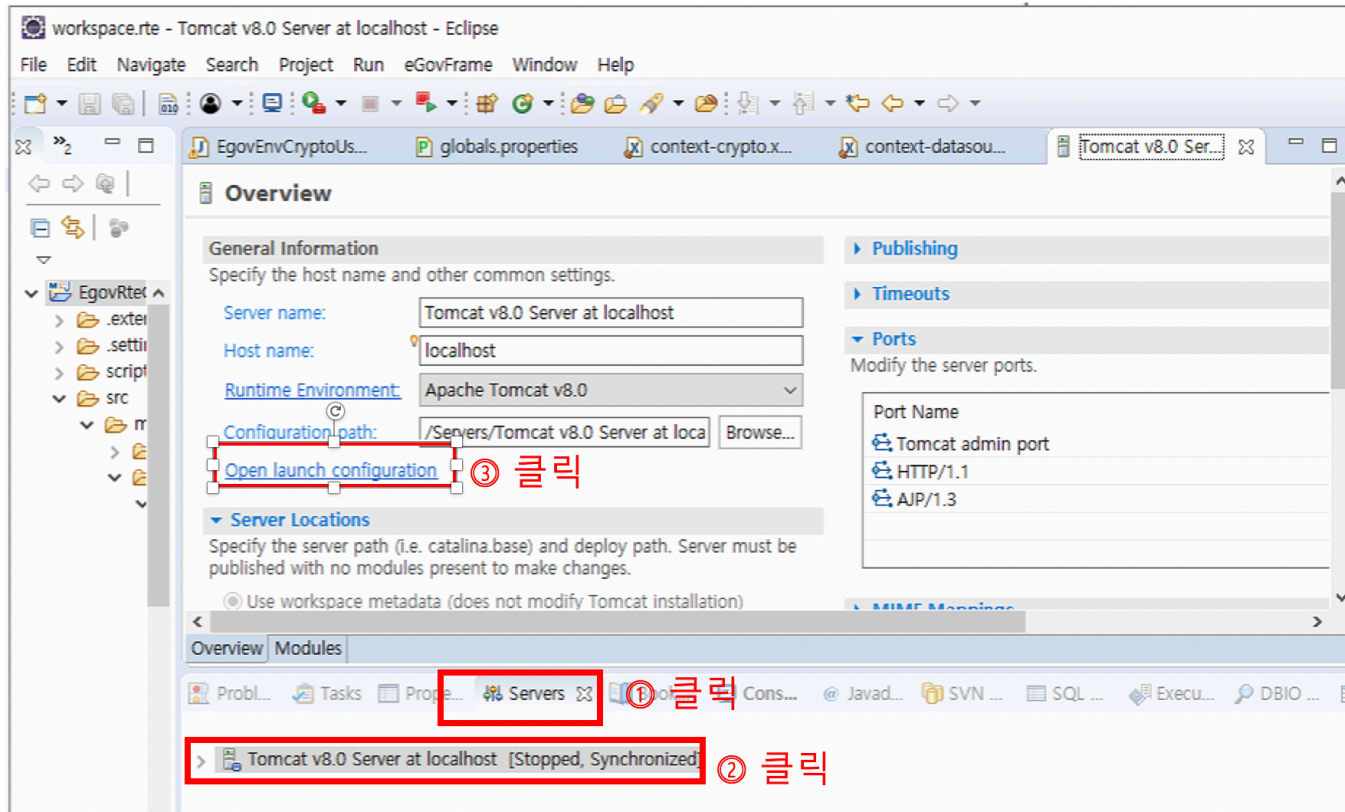
context-crypto.xml 설정파일의 crypto 간소화 설정

```
<egov-crypto:config id="egovCryptoConfig"
initial="true"
crypto="true"
algorithm="SHA-256"
cryptoBlockSize="1024"
/>
```

신기능 · 개발팁 및 활용 - 서버 환경설정 파일 암호화 (실행환경)

- 서버 환경설정 파일 암호화

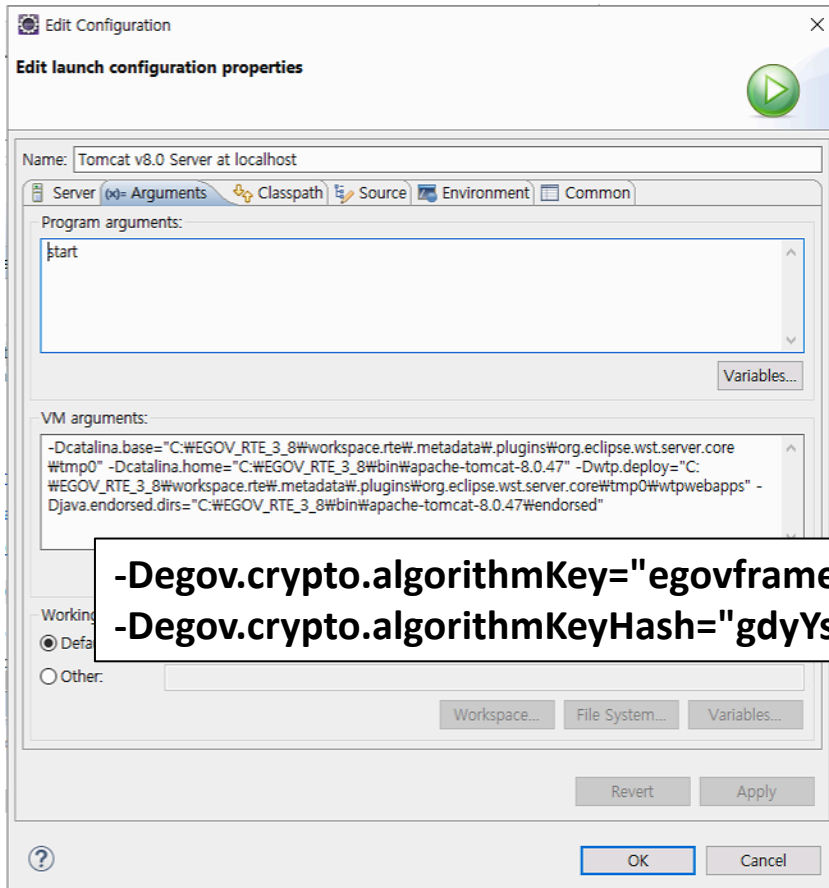
이클립스 톰캣 설정의 Open Launch Configuration 환경 설정 파일 오픈



신기능 · 개발팁 및 활용 - 서버 환경설정 파일 암호화 (실행환경)

- 서버 환경설정 파일 암호화

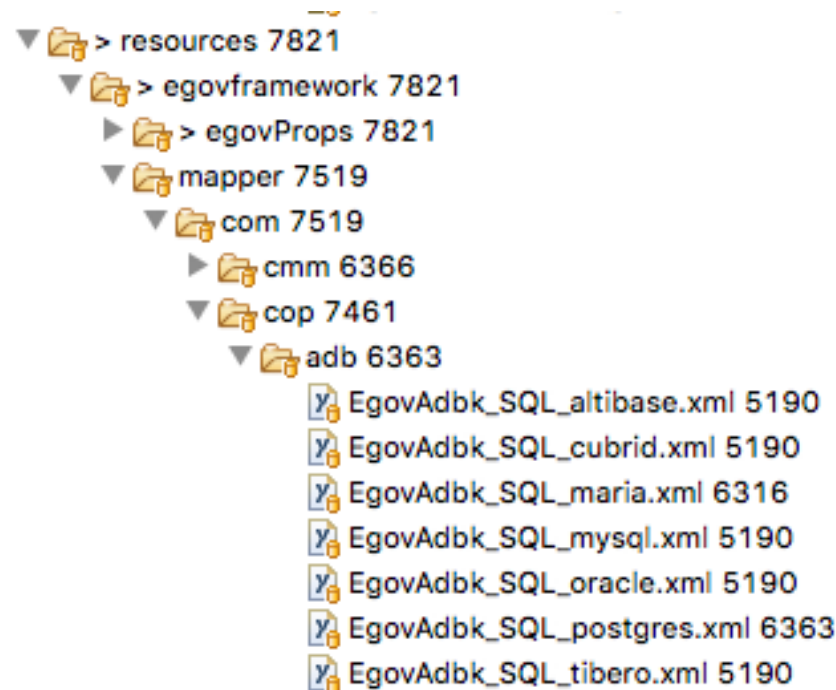
Edit Raunch Configuration의 vm arguments에서 Degov.crypto.algorithmKey, Degov.crypto.algorithmKeyHash 옵션값 추가



신기능 · 개발팁 및 활용 - DB 2종 추가 (공통컴포넌트)

● DB 2종 추가

- 기존 지원하는 DB 5종(altibase, cubrid, mysql, oracle, tiber)에서 2종(Postgresql, MariaDB)을 추가 지원
- 공통컴포넌트 기본 50종 + 신규 (97)종 : 147종 지원



신기능 · 개발팁 및 활용 - Jenkins (서버 개발환경)

● Jenkins 도입

- Hudson 라이선스 정책 변경에 따른 사용 제한 발생(Eclipse Public License: EPL)
- Hudson은 Sun에서 관리하던 java.net에서 출시 되어 관리되는 오픈소스 CI툴
Sun -> 오라클로 인수 되면서 허드슨 오픈소스 커뮤니티 간의 분쟁으로 분기되어
젠킨스란 새로운 이름을 서비스를 시작
- Jenkins : MIT License 채택

