

# 전자정부 표준프레임워크 신규버전 4.0 소개

eGovFrame



# Contents

1. 소개 및 추진성과
2. 신규 버전 4.0 소개
3. 향후 추진 과제

# I. 소개 및 추진 성과



## “ 전자정부 서비스 품질향상 및 정보화 투자 효율성 제고 ”



국가 정보화  
투자 효율성 제고



중소 SI업체  
경쟁력 확보



선진 국가정보화  
추진 기반 환경 제공

공통컴포넌트 재사용성 제고

공통컴포넌트 및  
표준프레임워크 개선

표준프레임워크 이용 활성화

### 차세대 전자정부 공통컴포넌트 및 표준프레임워크 구축 및 적용 요구

각 사업별로  
공통컴포넌트 중복 개발

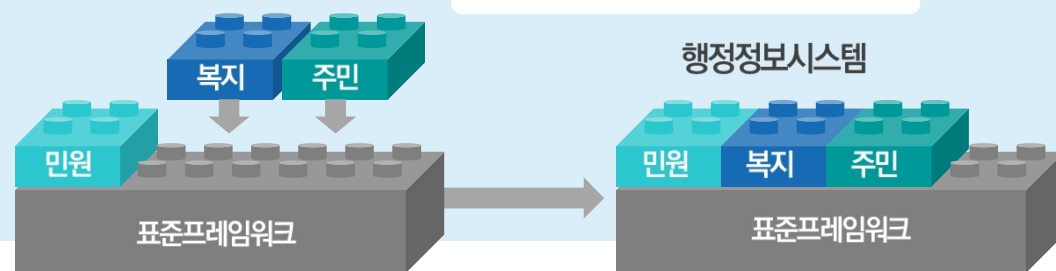
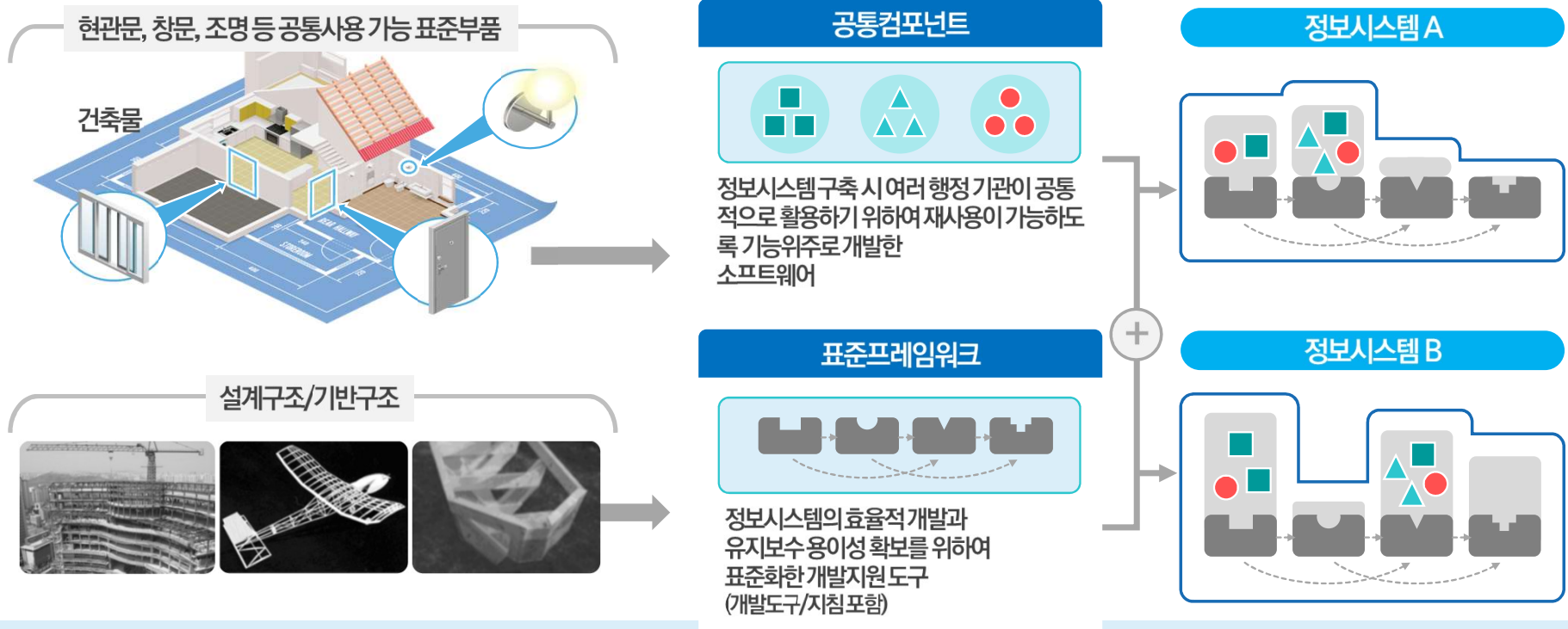
기관별/사업별  
개별적인 정보화 사업추진

최적화/표준화된  
공통개발기반 요구

특정업체 종속성 발생으로  
인한 공정경쟁 저하 및  
사업자 변경 시 예산낭비

개발표준 부재로  
시스템간 상호운용성 및  
재사용성 저하

# 전자정부 표준프레임워크 개념



# 표준프레임워크 추진경과

07.12~'08.6

## ISP 수립

- 표준프레임워크 목표모델 정립 및 공통컴포넌트 219종 도출  
※ 전문가 협의회(총7회), 공청회('08.4), 이해관계자간담회('08.10) 개최

08.11~'10.11

## 1~3단계 구축 사업

- 표준프레임워크 실행환경, 개발환경, 관리환경, 운영환경 구축
- 로그인, 게시판 등 공통컴포넌트 219종 구축
- 표준프레임워크 포털 시스템 및 영문 포털
- 국가대표포털, 통합인증체계, 여수엑스포, 국토부 공간정보 등 92개 사업 적용
- '09년 대한민국 SW 기술대상 우수상 수상(지경부)
- 싱가포르 주관 미래정부시상(FutureGov Awards) 3개 부문 수상  
- 올해의정부상(대상), 기술선도상, 정부혁신상
- 2010 공개SW 활용상 지식경제부장관상(대상) 수상 (지경부)
- 국제공인인증 CMMI 획득

11.4~'11.11

## 4단계 및 모바일

- 표준프레임워크 오픈소스 버전 업그레이드
- 표준프레임워크 경량화 및 모바일 표준프레임워크 구축
- 모바일 웹 공통컴포넌트 40종 구축

12.4~'12.11

## 5단계 구축 사업

- 표준프레임워크 배치프레임워크 개발
- 모바일 표준프레임워크 디바이스 API 개발

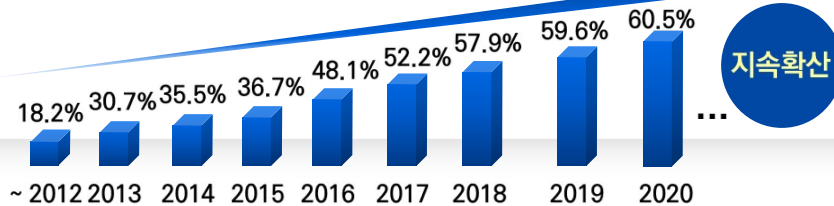
'11.1~현재

## 운영관리 · 기술지원

- '21년 9말 현재 4,972개(약 5.9조원) 사업에 적용
- 표준프레임워크 교육 10,808명 수료
- 297개 SW 호환성 확인
- 17개국 33개 해외사업 적용

## 공공부문 활용

- 공공 및 민간 833개 사업 기술지원  
\* 기술지원 464건, 교육지원 98건, 적용점검 208건 진행
- 전자정부 표준프레임워크 공공부문 적용률 지속확산  
\* 전자정부 표준프레임워크 적용 사업 수 4,972개 사업



## 사용자 교육

- 표준프레임워크 개발자 및 발주자 교육  
\* 월 1회 정기교육 실시, '09년부터 11,540명 교육 실시
- 사용자 맞춤형 교육  
\* 국방부, 건강보험심사평가원, 성균관대학교, 부산대학교, 한국산업기술대학교 등 3,052명 ('09~)
- 해외 확산 교육  
\* 베트남, 멕시코, 나이지리아, 우즈베키스탄, 남아공, 수단, 코트디부아르 등 14개국

## 생태계 조성

- 개방형 상생 생태계 조성 (오픈커뮤니티 설립·운영)
- 표준프레임워크 SW 호환성 확인 342건  
\* 20개 분야, 227개 상용SW 제품

## 글로벌 확산

- 글로벌 확산 협력을 위한 MoU체결 및 기술지원  
\* 멕시코, 나이지리아, WeGO, 남아공, 베트남 다낭시, 우즈베크, 수단, 코트디부아르와 MoU  
\* 필리핀, 튀니지, 태국, 사우디 등 기술지원
- 해외 17개국 33개 사업 적용(1,549억 규모)
- 해외 표준프레임워크센터 설립·운영  
\* 멕시코 표준프레임워크센터 설립('13.10) 및 현재 운영 중

## II. 신규버전 4.0 소개



# 표준프레임워크 4.0 주요 개선 사항

II. 신규버전 4.0 소개



표준프레임워크 MSA 포털  
MICROSERVICES ARCHITECTURE

MSA 템플릿 (신규)  
+  
템플릿 3종 UI/UX 개선

spring 5.3.6.RELEASE

spring boot 2.4.5.RELEASE

Microservices Cloud

Batch

디지털 원패스

GitHub 컨트리뷰션

공통컴포넌트 252종



기존

v3.10

- Eclipse 2020.06 (4.16.0)
- 표준프레임워크 3.10 플러그인 탑재
- Spring IDE 3.9.13
- Nexus 2.14.17
- SpotBugs 4.0.6
- EGit 5.8.0

신규

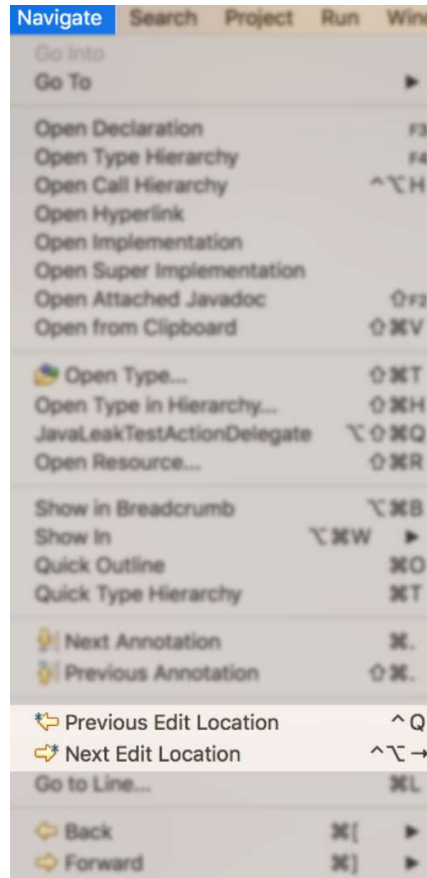
v4.0

- Eclipse 2021.03 (4.19.0)
  - JDK compliance Java 14 지원 등 개발환경 최신화
  - New record를 이용한 템플릿 생성 기능
  - Record 생성 위자드 제공
- 표준프레임워크 4.0 플러그인 탑재
  - 실행환경 v4. 0
- Spring IDE 3.9.16
  - JDK 11 (base line)
- Spring Tools 4.10
  - Spring Boot 대시보드
- Nexus 2.14.17
  - Remote Code Execution (RCE) 이슈 개선
- SpotBugs 4.2.3
  - FindBugs가 SpotBugs로 계승됨
- EGit 5.11.0
  - Eclipse용 Git 통합 툴

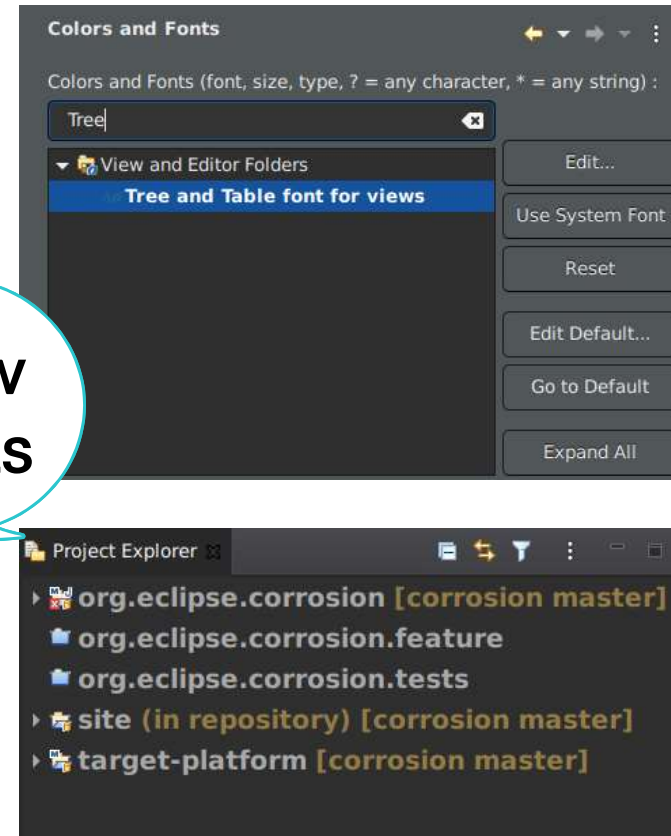
입체감



v4. 0

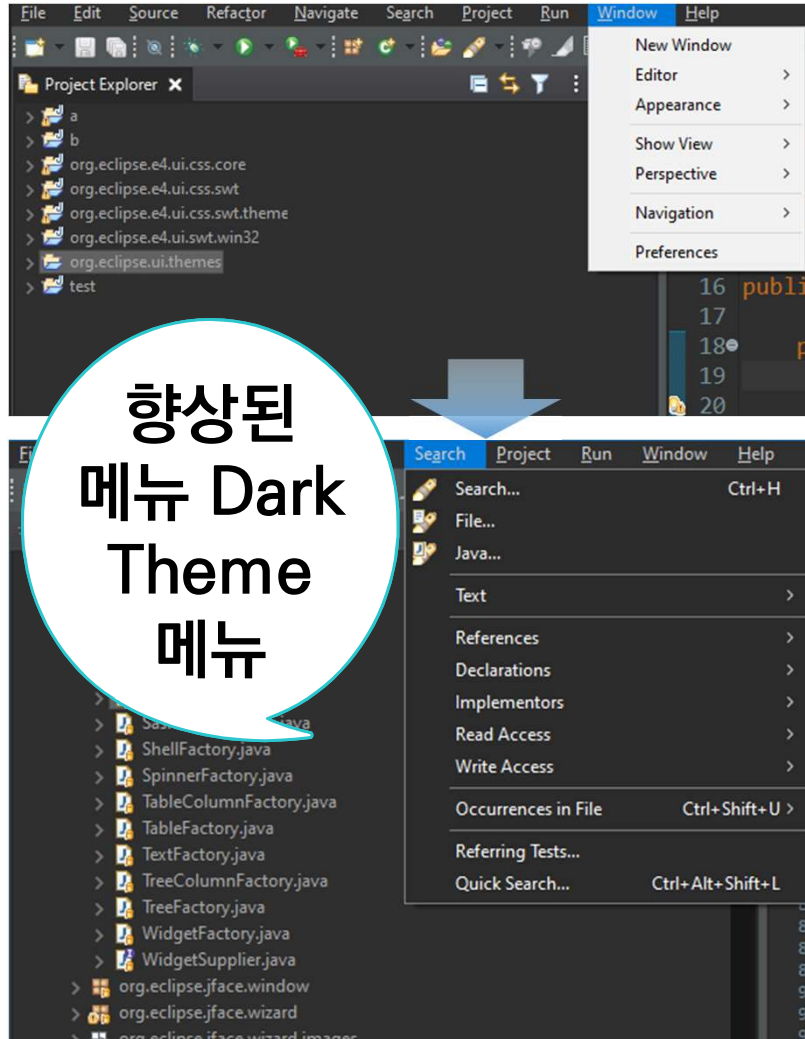


v4. 0



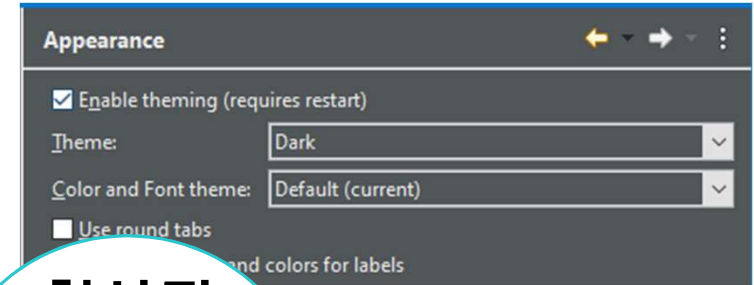
Multiple  
Edit  
Locations

View  
Fonts



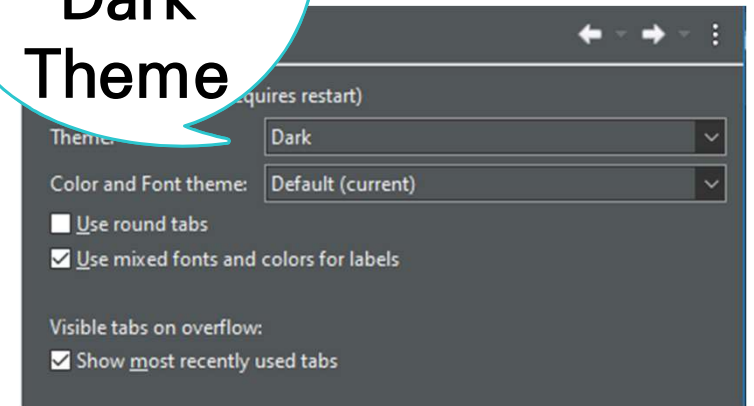
향상된  
메뉴 Dark  
Theme  
메뉴

v3.10

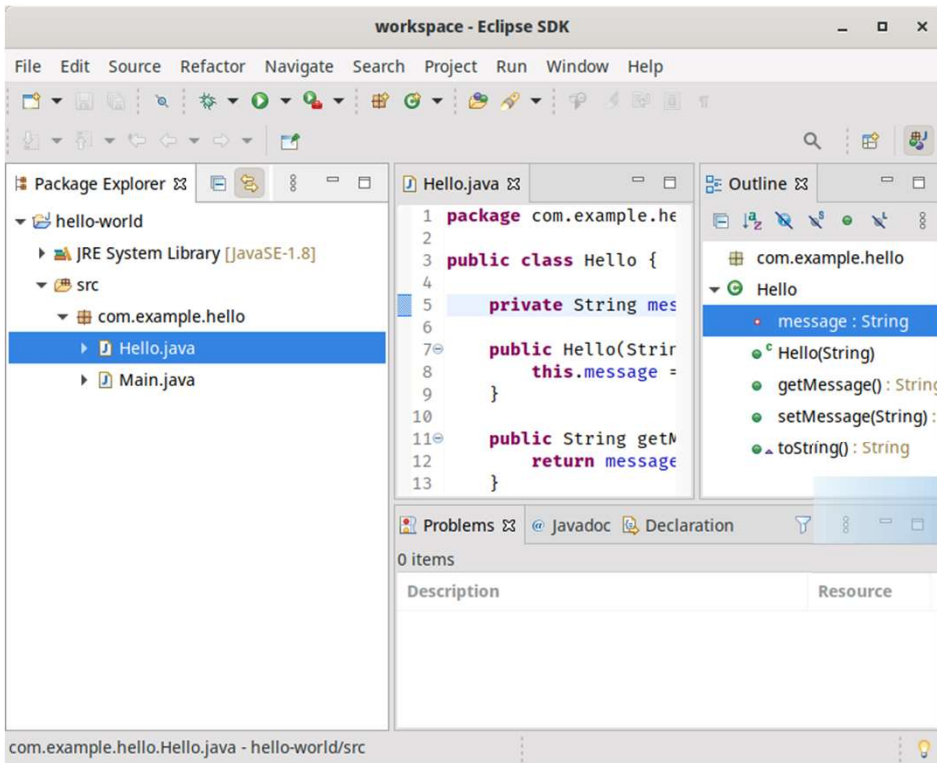


향상된  
Combo  
Dark  
Theme

v4.0

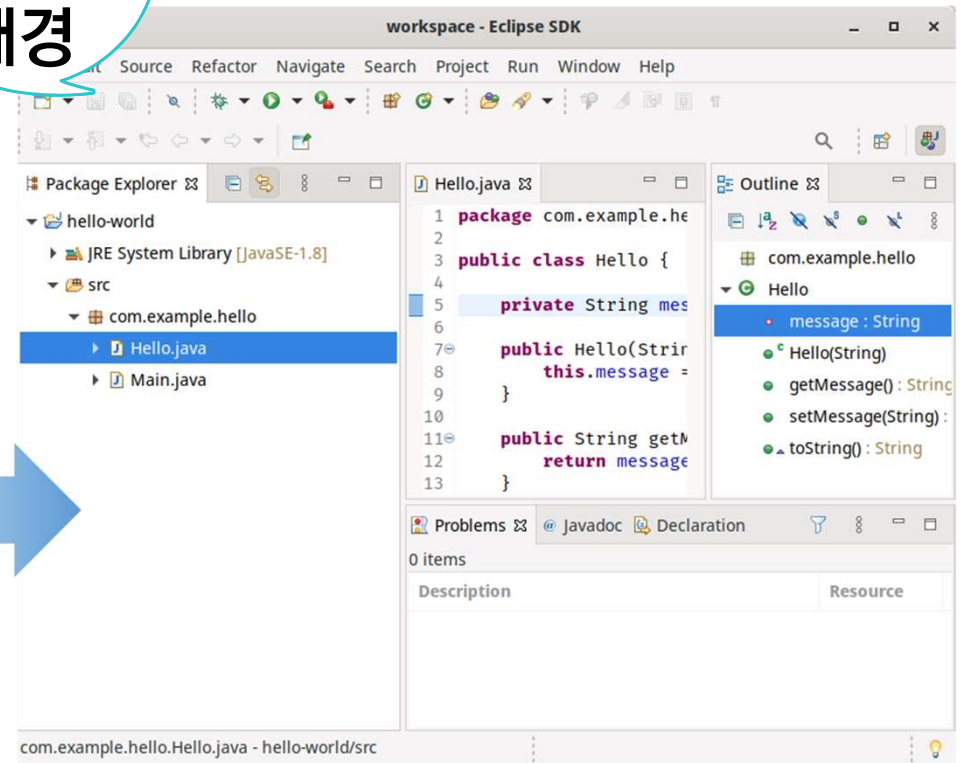


v3.10

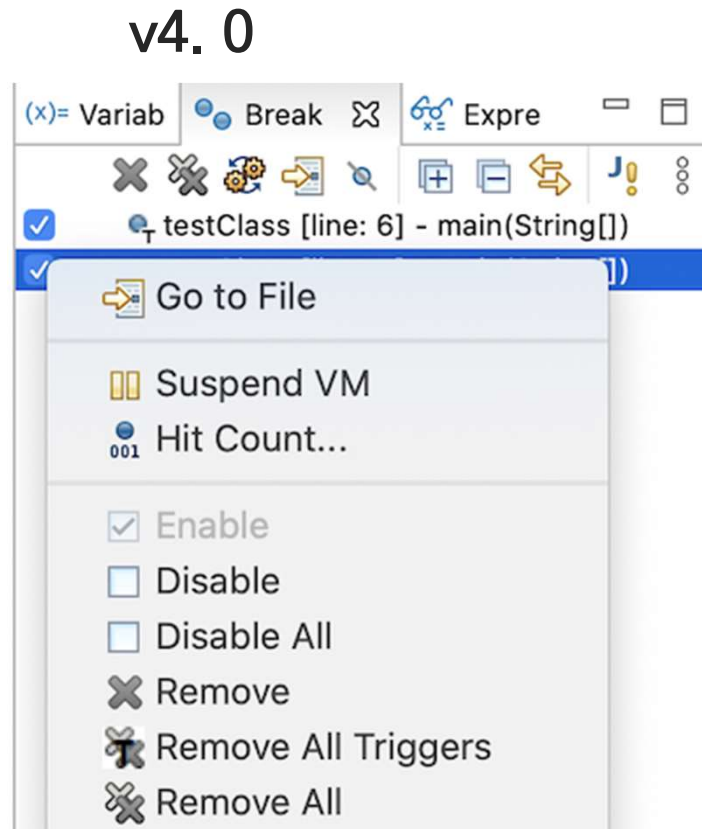


밝아진  
틀바  
배경

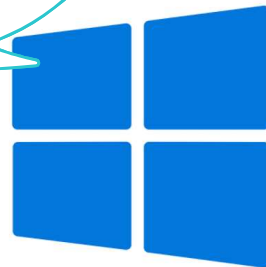
v4.0



Disable All  
in  
Breakpoints  
view



Horizontal  
Scrolling  
in Text Editor



v4.0

Assign to  
new local var  
in try-with..

```
public boolean loadFile(File file) throws IOException {  
    new FileInputStream(file);  
}
```

- Assign statement to new local variable (Ctrl+2, L)
- Assign to new local variable in try-with-resources (Ctrl+2, T)
- Assign statement to new field (Ctrl+2, F)
- Return the allocated object

```
public boolean loadFile(File file) throws IOException {  
    try (FileInputStream fileInputStream = new FileInputStream(file)) {  
    };  
    return true;  
}
```

fileInputStream  
inputStream  
stream



v4.0

try-with-resources  
기능

```
TestTryWithResources.java ✖
1 package trywithresources;
2
3 import java.io.FileInputStream;
4
5
6 public class TestTryWithResources {
7
8     public void foo() throws FileNotFoundException {
9         FileInputStream f = new FileInputStream("a.b");
10    }
11
12 }
13
```

```
TestTryWithResources.java ✖
1 package trywithresources;
2
3 import java.io.FileInputStream;
4 import java.io.FileNotFoundException;
5 import java.io.IOException;
6
7 public class TestTryWithResources {
8
9     public void foo() throws FileNotFoundException {
10        try (FileInputStream f = new FileInputStream("a.b")) {
11        } catch (FileNotFoundException e) {
12            throw e;
13        } catch (IOException e) {
14            // TODO Auto-generated catch block
15            e.printStackTrace();
16        }
17    }
18
19 }
20
```

Type hierarchy of java.io.FileNotFoundException:

- Object - java.lang
  - Throwable - java.lang
    - Exception - java.lang
      - IOException - java.io
        - FileNotFoundException - java.io

Press 'Ctrl+T' to see the supertype hierarchy

Clean Ups >  
Java Feature  
추가됨

The screenshot shows the 'Clean Ups' dialog in an IDE, with the 'Java Feature' tab selected. The dialog is organized into sections for different Java versions:

- Java 15:**  Pattern matching for instanceof (Java 15 or higher with feature preview)
- Java 14:**  Convert to switch expression
- Java 10:**  Use local variable type inference
- Java 8:**
  - Convert functional interface instances
    - Use lambda where possible
    - Use anonymous class
  - Use Comparator.comparing()
  - Use String.join() when possible
- Java 7:**
  - Use try-with-resource
  - Use Multi-catch
  - Use diamond operator
  - Use Objects.hash()
  - Use Objects.equals() in the equals method implementation
- Java 5:**
  - Convert 'for' loops to enhanced
    - Only if loop variable used
  - Use Autoboxing
  - Use Unboxing

The 'Preview' pane on the right shows the following code:

```
IF (object instanceof Integer i) {  
    return i.intValue();  
}  
  
int i;  
switch(j) {  
    case 1:  
        i = 3;  
        break;  
    case 2:  
        i = 4;  
        break;  
    default:  
        i = 0;  
        break;  
}  
  
int number = 0;  
ArrayList<String> list = new ArrayList<String>();  
HashMap<Integer, String> map = new HashMap<>();  
  
IntConsumer c = new IntConsumer() {  
    @Override public void accept(int value) {  
        System.out.println(i);  
    }  
};  
Runnable r = () -> { /* do something */ };  
  
Comparator<Date> comparator = Comparator.nullsFirst(Compara
```

# 표준프레임워크 4.0 개발환경

## II. 신규버전 4.0 소개

### 템플릿 UI 개선

**표준프레임워크 포털 eGovFrame** **상용 홈페이지** | 사이트소개 | 정보마당 | 고객센터 | 알림마당

**공지사항** 갤러리 + 더보기  
검색된 결과가 없습니다.

**자료실**  
다양한 자료를 다운로드 받으실 수 있습니다.

**표준프레임워크센터**  
표준프레임워크센터의 약도 등의 정보를 제공합니다.

**단순 홈페이지**  
전자정부 표준프레임워크의 경량환경 내부업무에 대한 최신 정보와 기술을 제공하고 있습니다.

- 주요사업 소개**  
표준프레임워크가 제공하는 주요 사업을 소개합니다.
- 대표서비스 소개**  
표준프레임워크 실행환경의 서비스 그룹에서 제공하는 대표서비스입니다.
- 서비스 신청**  
표준프레임워크 경량환경 홈페이지의 다양한 서비스를 신청 하실 수 있습니다.
- 일정 현황**  
표준프레임워크 경량환경 홈페이지의 전체적인 일정 현황을 조회하실 수 있습니다.

**표준프레임워크 포털 eGovFrame** **내부업무 사이트** | 로그인

**표준프레임워크 경량환경 내부업무**  
표준프레임워크 경량환경 내부업무에 대한 전반적인 지원을 약속합니다.

**오늘의 할일** +

홈페이지 샘플업무1	2021-11-22		
홈페이지 샘플업무2	2021-11-22		
홈페이지 샘플업무3	2021-11-22		
홈페이지 샘플업무4	2021-11-22		
홈페이지 샘플업무5	2021-11-22		

**최신 업무공지 정보** +

홈페이지 샘플공지1	NEW	관리자	2021-11-22
홈페이지 샘플공지2	NEW	관리자	2021-11-22
홈페이지 샘플공지3	NEW	관리자	2021-11-22
홈페이지 샘플공지4	NEW	관리자	2021-11-22
홈페이지 샘플공지5	NEW	관리자	2021-11-22

**표준프레임워크 포털 eGovFrame** **상용 포털** | 로그인 | 회원가입 | 고객센터 | 정보마당

**표준프레임워크 경량환경 포털**  
표준프레임워크 경량환경 포털에 대한 전반적인 지원을 약속합니다.

**공지사항** | 자유게시판

**포털사이트 샘플공지1**  
포털사이트 샘플공지1  
2021-07-02

**포털사이트 샘플공지2**  
포털사이트 샘플공지2  
2021-07-02

**포털사이트 샘플공지3**  
포털사이트 샘플공지3  
2021-07-02

**자주하는 질문FAQ** +  
표준프레임워크 경량환경 서비스에 대한 자주하는 질문의 답변들을 볼 수 있습니다.

- FAQ 샘플3
- FAQ 샘플3 - 답변내용
- FAQ 샘플2
- FAQ 샘플2 - 답변내용
- FAQ 샘플1
- FAQ 샘플1 - 답변내용

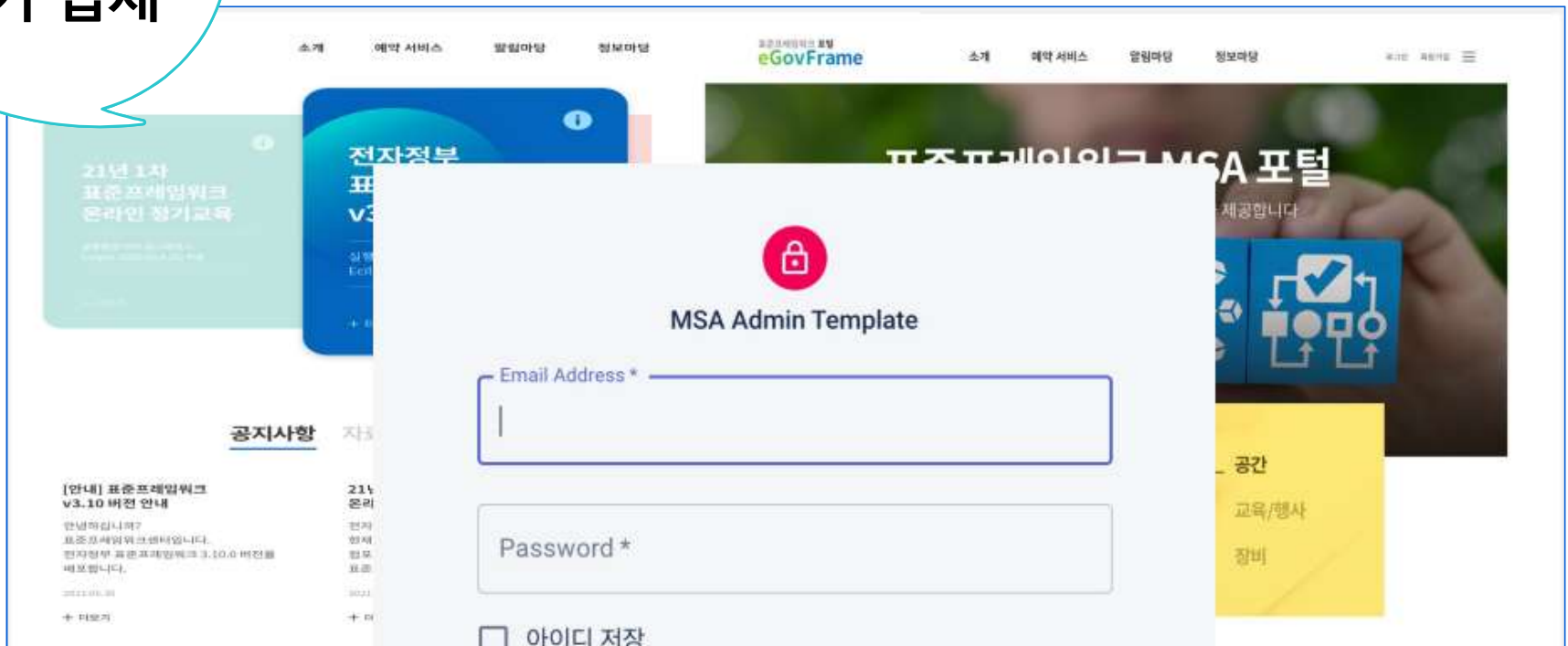
**한편에 신청하는 민원** +

기술지원 필요시  
유지보수 민원

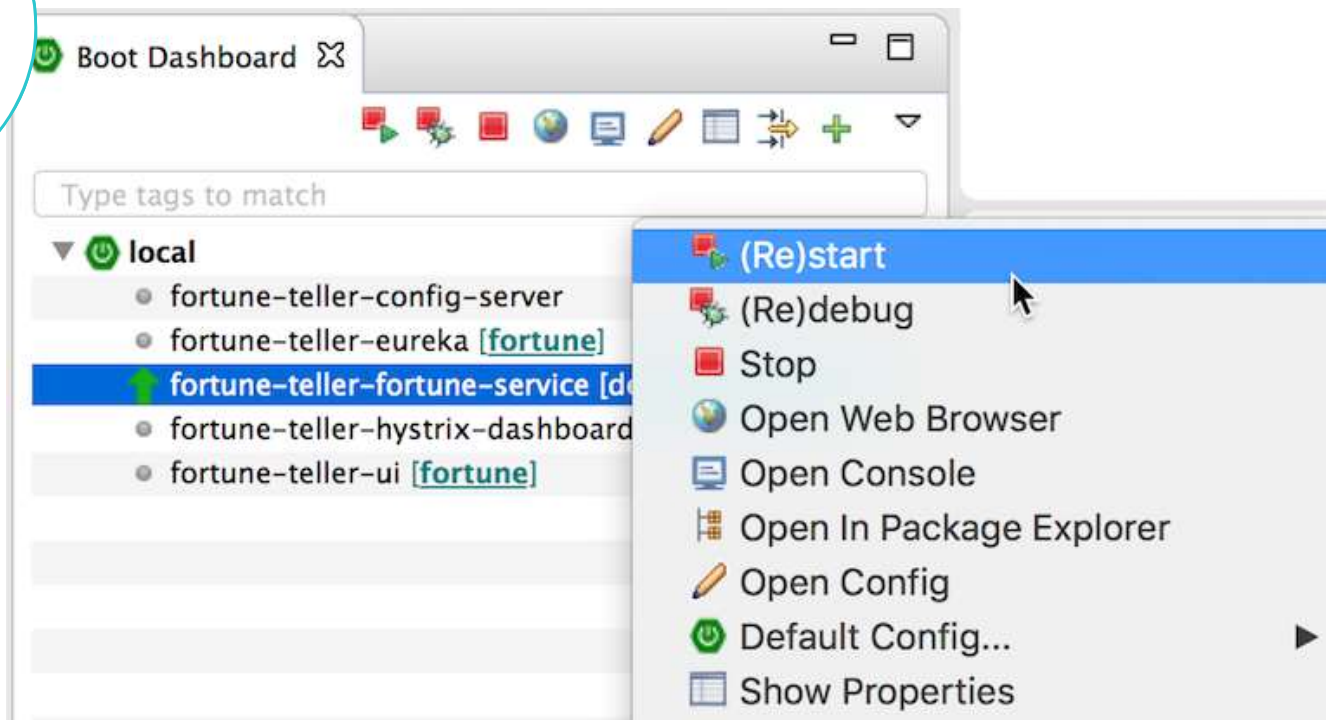
구매 제품  
A/S 민원

**포털 설명 참여**  
표준프레임워크 경량환경 포털 홈페이지

MSA 템플릿  
추가 탑재



### 스프링부트 대시보드



기존

v3.10

- Spring Framework (4.3.25)
  - SpEL 기능 향상
  - FlashMapManager 오류 개선
- Spring Security (4.2.13)
  - 인증 처리 기능 개선
- Spring Batch 3.0.10
  - XML 외부 참조 취약점 개선
- Spring Data JPA 1.11.23
  - 중간 심각도 보안 취약점 개선 버전 적용

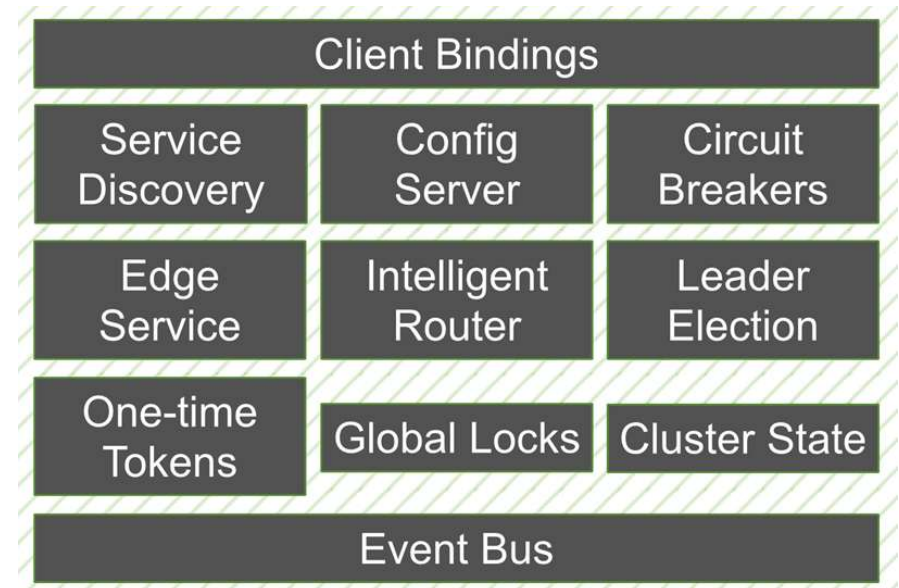
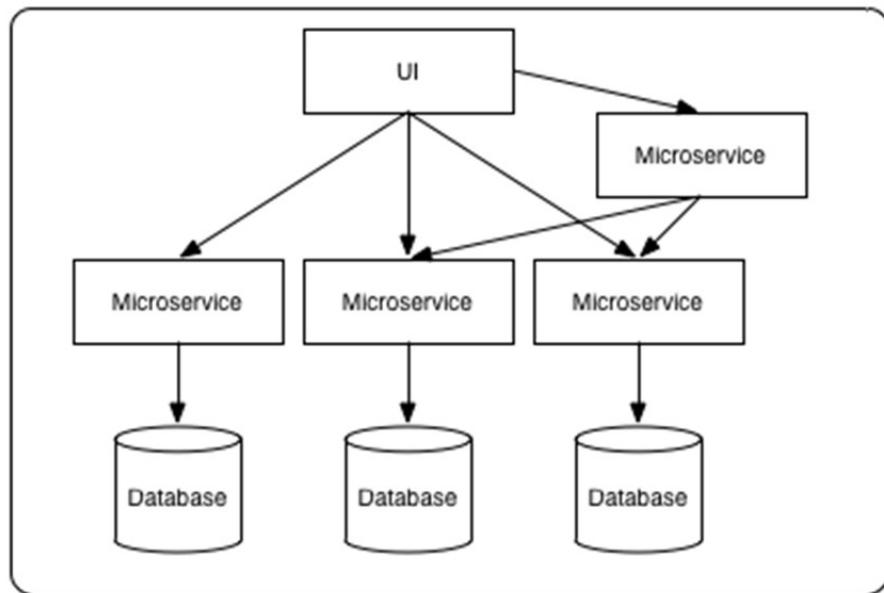
신규

v4.0

- Spring Framework (5.3.6)
  - JDK 8을 기반으로 하며 JDK 14 까지 호환
  - 스프링 부트 2.4.5 지원 (독립적 구동 및 배포)
  - 스프링 프레임워크 코드에서 JDK 8 기능 사용
  - 리액티브 프로그래밍 지원
  - 함수형 웹 프레임워크 제공
  - Kotlin 지원
- Spring Security (5.4.6)
  - OAuth 2.0 지원
  - 현대화된 암호 인코딩 (단방향 암호화 기능)
- Spring Batch 4.3.2
  - Spring Framework 5 지원
  - ItemReader 및 ItemWriter를 위한 빌더 지원
  - Micrometer(모니터링 도구) 기반의 배치 모니터링 지원
- Spring Data JPA 2.2.3

## 클라우드 네이티브 애플리케이션

- 마이크로서비스 아키텍처 구현
- MSA 적용 개발가이드 제공





- 리액티브 프로그래밍 지원
  - Reactive Stream, Reactor
  - WebFlux 도입으로 비동기 프로세스 애플리케이션 구현 가능
- JAVA EE 8 API와 통합
  - Java Reflection, 함수형 자바 랴다 지원
- JDK 9+, Servlet 4+, JUnit 5+ 지원
  - 어플리케이션용 업그레이드 모듈 시스템
- 융합 아키텍처 구축 지원
  - Spring MVC + Spring WebFlux



- 라이브러리 관리 자동화
  - Spring Boot Starter를 이용한 의존성 처리
- 설정의 자동화
  - 실행에 필요한 환경을 자동으로 설정
- 라이브러리 버전 자동 관리
  - 의존관계에 있는 서드파티 라이브러리 호환 버전 자동 관리
- 테스트 환경과 내장 Tomcat, Jetty, Underdow
  - JUnit 5와 내장 서버 제공
- 독립적으로 실행 가능
  - 웹 어플리케이션 jar 파일 패키징하여 독립적 실행 가능
- OAuth 2.0 도입
  - 인증·인가
  - 간단한 구현
  - 안정성 향상 (HTTPS 필수)
- 현대화된 비밀번호 인코딩
  - DelegatingPasswordEncoder





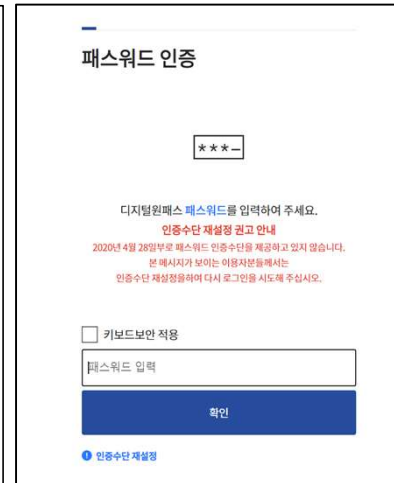
### ▪ 디지털 원패스 연동 추가

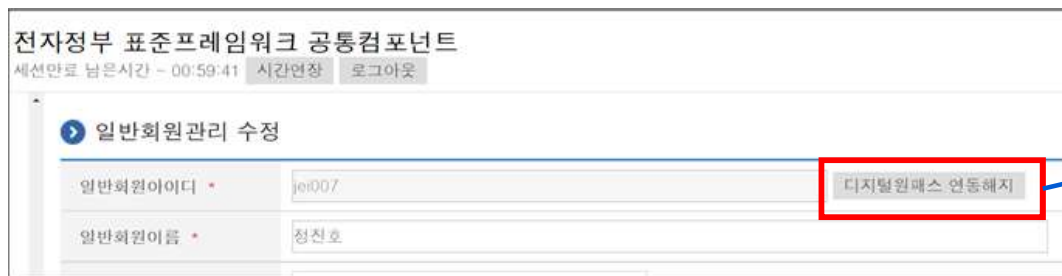


전자정부 서비스  
로그인 화면

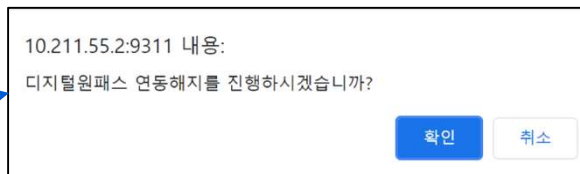



디지털 원패스  
회원 정보 확인

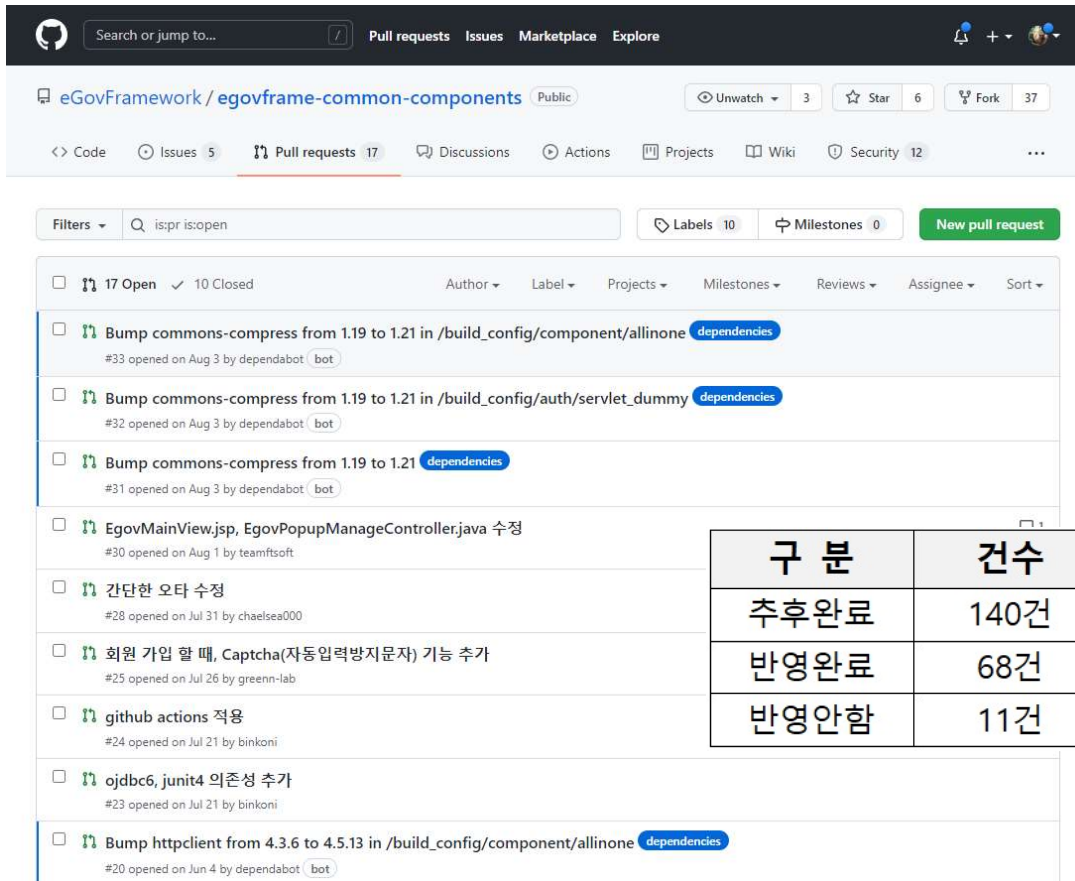




전자정부 서비스  
회원 관리 화면



## 컨트리뷰션 적용



2021.3.31. ~  
7.31(4개월)  
18명 참여  
(219건)



구분	건수	주요내용
추후완료	140건	공통컴포넌트 테스트코드 제공 및 기능추가 오픈소스 연동 시 오류 등
반영완료	68건	소스코드 수정 및 정리, 오류 제보, 테스트 추가 등
반영안함	11건	이미 제공되는 예제 요청, 이상없는 오류 제보 등

그룹	오픈소스 SW	표준프레임워크		비고
		3.10	4.0	
공통 기반	Spring Framework	4.3.25	5.3.6	- JDK8+, 함수형 웹 프레임워크 제공 - Spring WebFlux, Spring Boot
	Spring Boot		2.4.5	- Tomcat, Netty, Jetty, Undertow 내장
	Spring Security	4.2.13	5.4.6	- OAuth 2.0, 비밀번호 인코딩 향상
	Spring Batch	3.0.10	4.3.2	- Serialization 메카니즘 기본값 변경 Jackson - Micrometer(모니터링 도구) 기반의 배치 모니터링 지원
	Log4j	2.12.1	2.14.0	- Java 8 이상지원 - 비동기 처리 성능 향상, MongoDB 4 지원 추가
	Hibernate	5.1.7	5.4.10	- Java 8 이상지원 - Gradle 플러그인 개선, 바이트 코드 향상
	MyBatis	3.4.6	3.5.3	- CDATA 구문에서 대체변수 지원 - 다중 행 삽입 구문 지원

그룹	오픈소스 SW	표준프레임워크		비고
		3.10	4.0	
구현 도구	Eclipse	4.16.0	4.19.0	<ul style="list-style-type: none"> <li>- JDK compliance Java 16 지원 등 개발환경 최신화</li> <li>- Try-with-resources Block 이 자동 생성되는 기능</li> <li>- Java 버전별 코드 Clean Up 기능</li> <li>- Non blocking 코드 completion 기능</li> </ul>
CI 도구	Nexus2 Repository	2.14.17	3.36.0	- 최신 안정화버전 적용
	Jenkins	2.242	2.288	- 최신 안정화버전 적용
	VisualSVN	4.2	4.3.4	- 최신 안정화버전 적용

그룹	오픈소스 SW	표준프레임워크		비고
		3.10	4.0	
플러그인	Spring Tools	3.9.13	4.10.0	- Spring Boot 지원, JDK11+ 필요
	UML2 Extension	5.5.1	5.5.2	- 최신 안정화 버전 적용
	eGovFramework 1) 공통컴포넌트 2) 템플릿 3) 소스자동생성	3.10.0	4.0.0	1) 최신 4.0.0 표준프레임워크 실행환경 탑재 2) 기능이 개선된 공통컴포넌트 탑재 3) 4.0.0 실행환경이 적용된 템플릿 탑재
	PMD	4.16	4.24.0	- 최신 안정화 버전 적용
	EclEmma	3.1.3	3.1.4	- 최신 안정화 버전 적용
	Gradle	3.1.4	3.1.5	- 최신 안정화 버전 적용
	JUnit	4.13	4.13	-

그룹	오픈소스 SW	표준프레임워크		비고
		3.10	4.0	
플러그인	SpotBugs	4.1.2	4.2.3	- 최신 안정화 버전
	EGit	5.8.0	5.11.0	- GPG 인증방식 추가
	Grep Console	3.7.0	3.7.0	- 디버깅 정보 및 로그 정보 시각화

### III. 향후 추진 과제



클라우드 활용을 위한  
표준프레임워크 개선



## 수요자 중심 표준프레임워크 기능개선

- ✓ Non-Blocking 리액티브 프로그래밍 지원
- ✓ 마이크로서비스 아키텍처 템플릿 고도화

최신 기술 및  
정보화 사업 동향에  
대응하는 기술지원



## 최신 기술 및 사업 동향을 통한 발전

- ✓ 스트리밍 템플릿, 빅데이터 템플릿 등 확대 지원
- ✓ 모바일 운전면허증 정부 공통 서비스 연계 지원

국내·외 표준프레임워크  
적용 및 확산 지속력 강화



## 국내외 표준프레임워크 지속확산 체계 강화

- ✓ 국내확산을 위한 전문가 확보와 체계 강화
- ✓ 글로벌 정책 수립 및 지원 강화



# 감사합니다

전자정부 표준프레임워크 센터

E-Mail : [egovframesupport@gmail.com](mailto:egovframesupport@gmail.com)

Tel : +82-1566-3598

Home page : <https://egovframe.go.kr>

