

표준프레임워크 컨트리뷰터 되기

개발가이드 - 실행환경
편

2024.08.29



—
시작하면서

컨트리뷰션은 오픈소스
프로젝트에 참여하고
기여하는 모든 활동을
의미함

사람들이 왜 컨트리뷰션 활동을 하는지,

사용하는 소프트웨어를
개선하기 위해

작은 변화라도 스스로 만들 수 있다는
점에서 힘을 얻기 위해

기존 기술을 향상시키기
위해

비슷한 것에 관심이 있는
사람들을 만나기 위해

사람들과의 소통 능력을
만들기 위해

멘토를 찾고 다른 사람들을
가르치기 위해

Build public artifacts that help you grow a
reputation (and a career)

컨트리뷰션 유형

소스코드 수정만이
컨트리뷰션이 아니다.

2016년 발표된 "More Common Than You Think: An In-Depth Study of Casual Contributors"에 따르면 다음과 같이 컨트리뷰션의 30% 이상이 소스 코드 수정 이외의 영역에서 이루어졌습니다.

THE CATEGORIZATION OF THE CASUAL CONTRIBUTIONS.

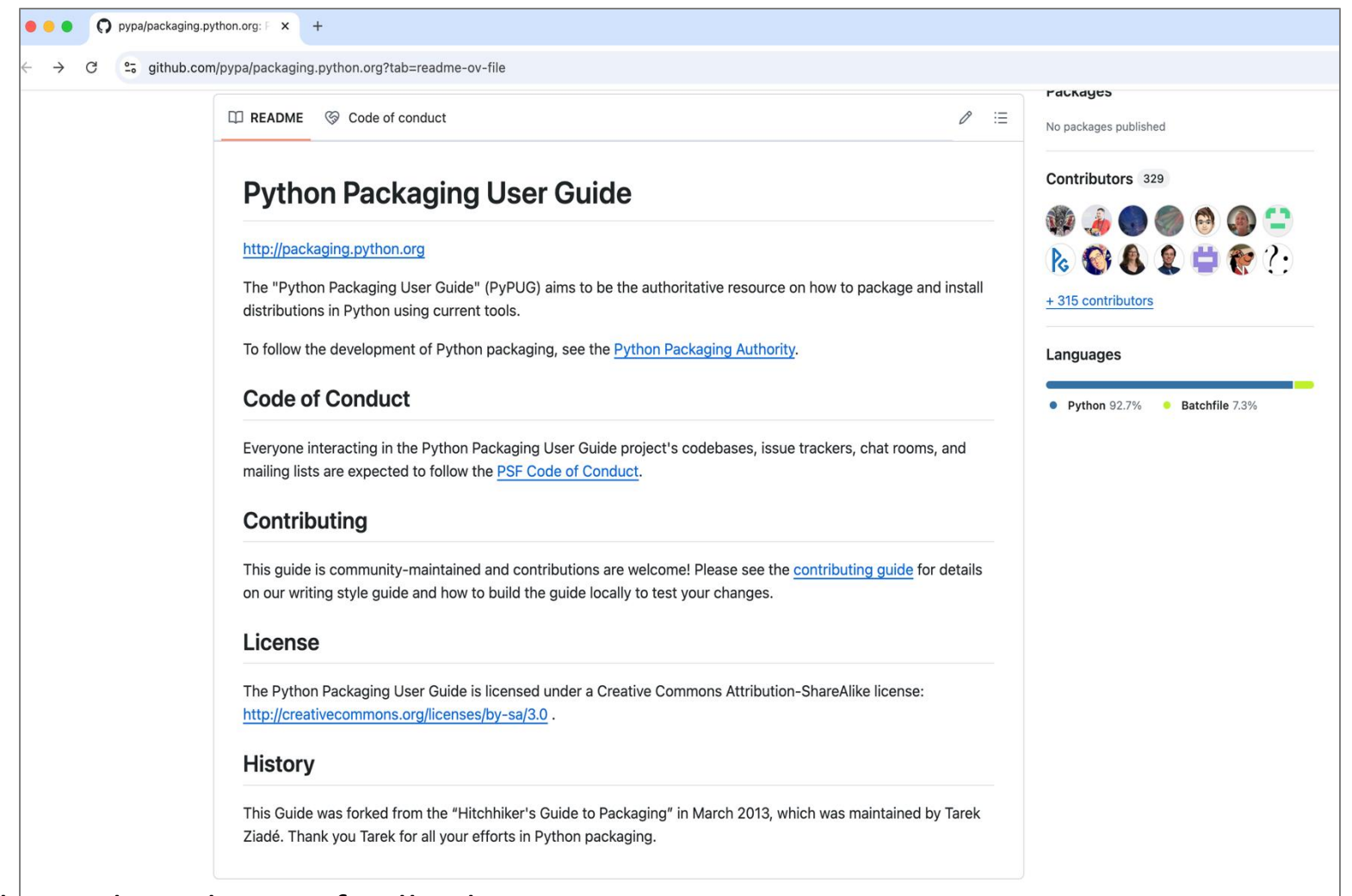
Category	#	%
Bug Fix	116	30.20%
Documentation	110	28.64%
Add New Feature	72	18.75%
Refactoring	34	8.85%
Update Version/Dependencies	25	6.51%
Improve Error/Help Messages	14	3.64%
Improve Resource Usage	8	2.08%
Add Test Cases	5	1.30%

컨트리뷰션 유형

사례

.사례

<https://github.com/pypa/packaging.python.org?tab=readme-ov-file>



- Reading the guide and giving feedback
- Reviewing new contributions
- Revising existing content
- Writing new content
- Translate the guide

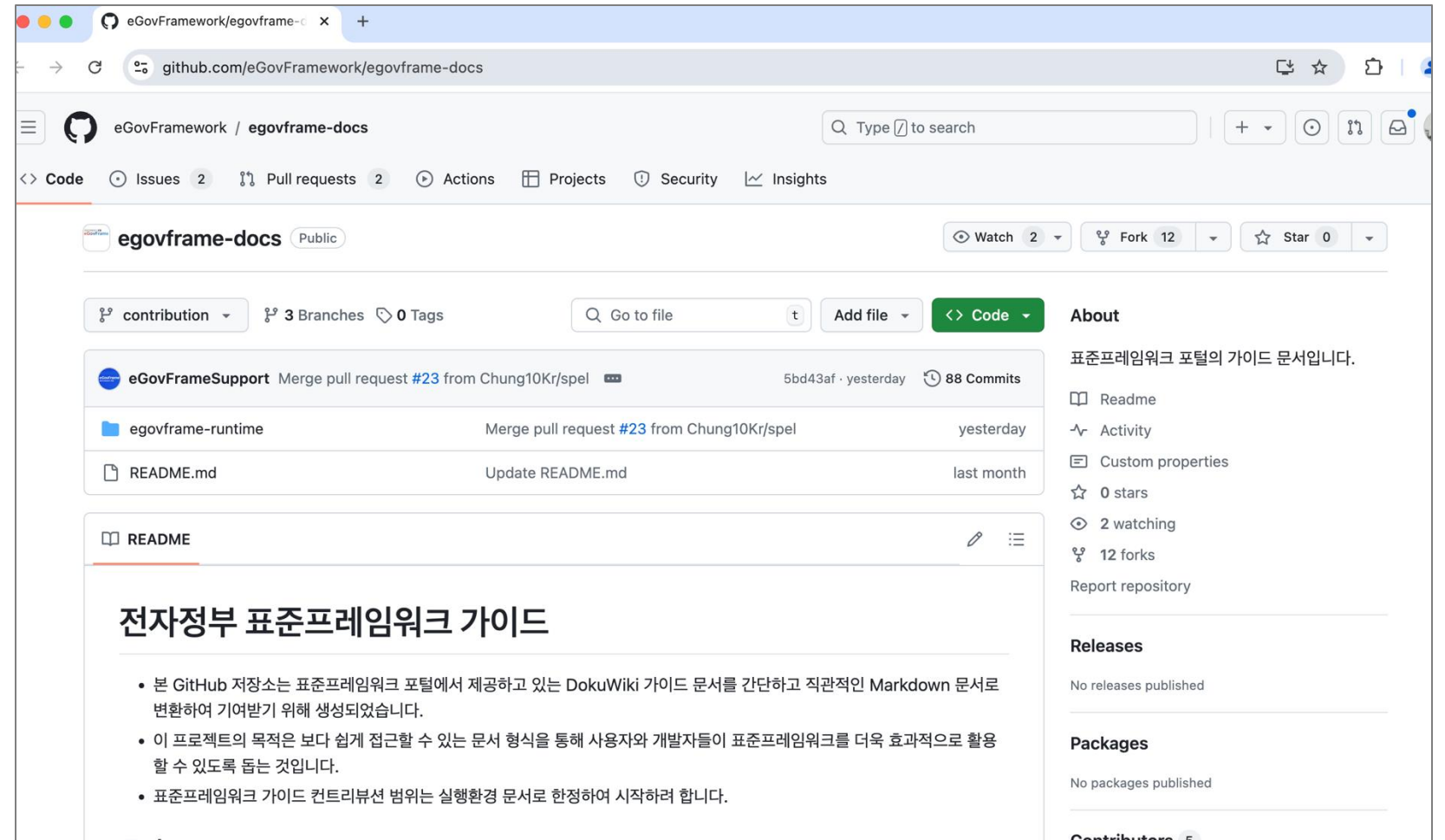
그래서

표준프레임워크 개발가이드 컨트리뷰션



사례

<https://github.com/eGovFramework/egovframe-docs>



표준프레임워크 구성

1. 표준프레임워크 구성

2. 표준프레임워크 구성

구성도

표준프레임워크는 실행, 개발, 관리, 운영 등 4개의 환경과 모바일 표준프레임워크, 공통컴포넌트로 구성

공통컴포넌트

모바일

모바일전용 공통컴포넌트

기존 공통컴포넌트 모바일 전환

웹

공통기술 컴포넌트

요소기술 컴포넌트 (유틸리티)

표준프레임워크

개발환경

구현도구

배포도구

테스트도구

형상관리도구

모바일 디바이스
API 개발도구

실행환경

화면처리

업무처리

데이터처리

연계통합

공통기반

배치처리

모바일
화면처리

모바일 디바이스
API 실행환경

관리환경

서비스요청관리

변경관리

현황관리

표준관리

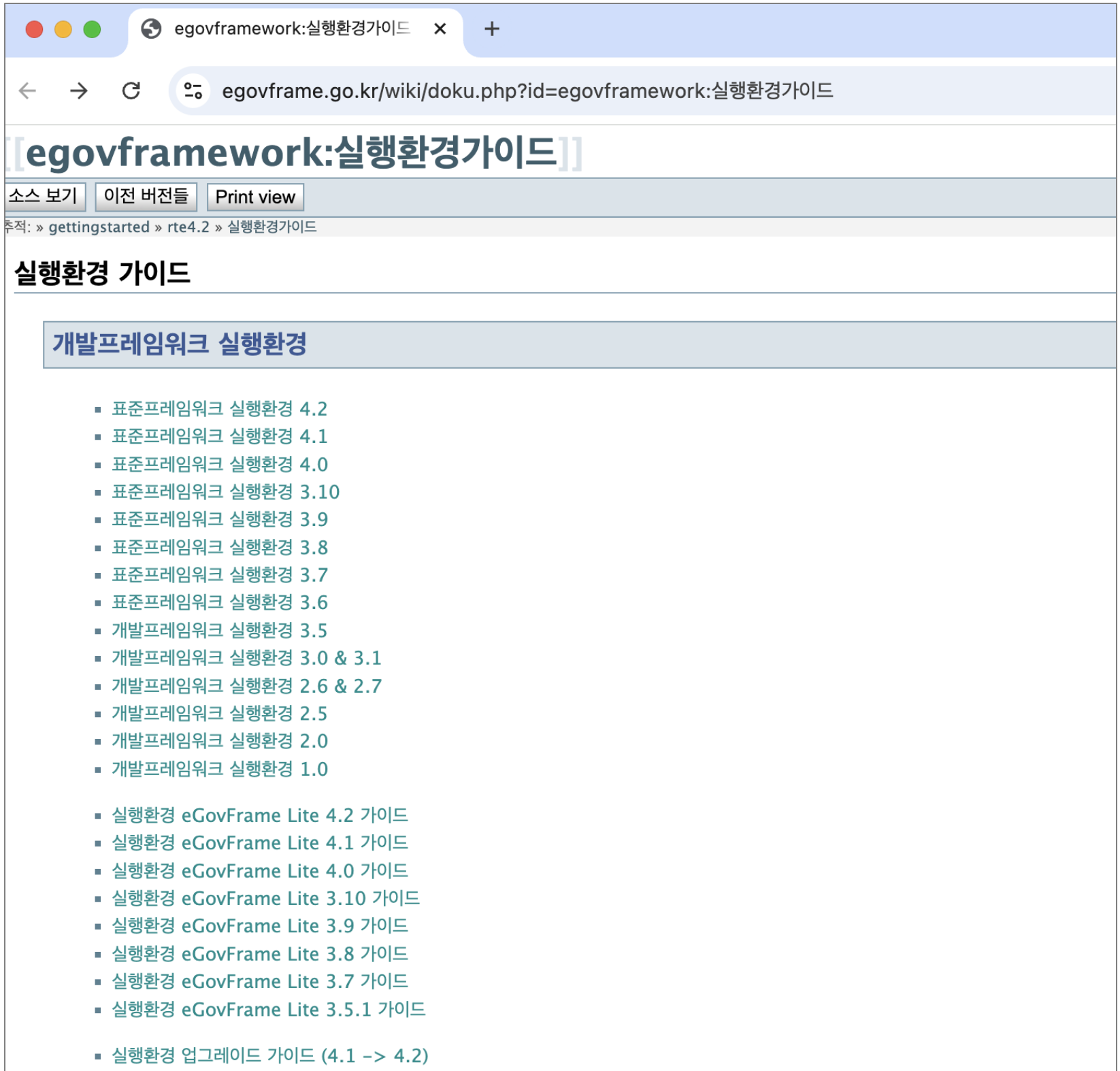
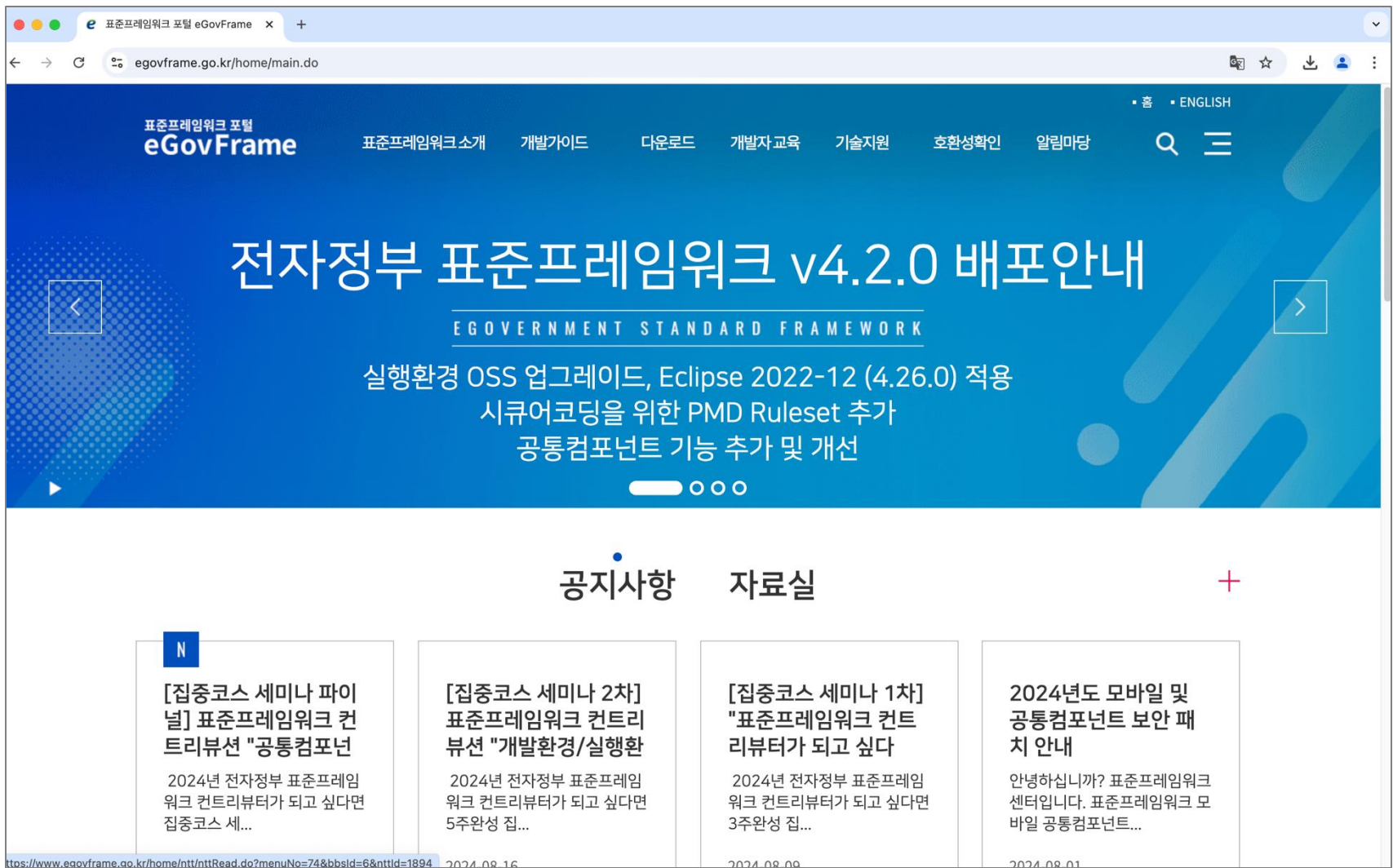
운영환경

모니터링 도구

운영관리 도구

배치운영 도구

표준프레임워크 개발가이드



—
지금까지는

개발가이드 (지금)

1. https://www.egovframe.go.kr/wiki/doku.php?id=egovframework:rte4.2:fdl:ioc_container

egovframework:rte4.2:fdl:ioc

✕

+

← → ↺ egovframe.go.kr/wiki/doku.php?id=egovframework:rte4.2:fdl:ioc_container

[[egovframework:rte4.2:fdl:ioc_container]]

소스 보기

이전 버전들

Print view

추적: » overview » brte_v3.7 » 실행환경가이드 » rte4.2 » ioc_container

IoC Container

개요

프레임워크의 기본적인 기능인 Inversion of Control(IoC) Container 기능을 제공하는 서비스이다.
객체의 생성 시, 객체가 참조하고 있는 타 객체에 대한 종속성을 소스 코드 내부에서 하드 코딩하는 것이 아닌, 소스 코드 외부에서 설정하게 함으로써, 유연성 및 확장성을 향상시킨다.

주요 개념

Inversion of Control(IoC)

IoC는 Inversion of Control의 약자이다. 우리나라 말로 직역해 보면 “역제어”라고 할 수 있다. 제어의 역전 현상이 무엇인지 살펴본다.
기존에 자바 기반으로 어플리케이션을 개발할 때 자바 객체를 생성하고 서로간의 의존 관계를 연결시키는 작업에 대한 제어권은 보통 개발되는 어플리케이션에 있었다.
그러나, Servlet, EJB 등을 사용하는 경우 Servlet Container, EJB Container에게 제어권이 넘어가서 객체의 생명주기(Life Cycle)를 Container들이 전담하게 된다.
이처럼 IoC에서 이야기하는 제어권의 역전이란 객체의 생성에서부터 생명주기의 관리까지 모든 객체에 대한 제어권이 바뀌었다는 것을 의미한다.

관련문서

▪ Martin Fowler가 저술한 Inversion of Control

▪ [Inversion of Control 한글 번역](#)

Dependency Injection

각 클래스 사이의 의존관계를 빈 설정(Bean Definition)정보를 바탕으로 컨테이너가 자동적으로 연결해주는 것을 말한다.
컨테이너가 의존관계를 자동적으로 연결시켜주기 때문에 개발자들이 컨테이너 API를 이용하여 의존관계에 관여할 필요가 없게 되므로 컨테이너 API에 종속되는 것을 줄일 수 있다.
개발자들은 단지 빈 설정파일(저장소 관리 파일)에서 의존관계가 필요하다는 정보를 추가하기만 하면 된다.

관련문서

▪ Martin Fowler가 저술한 Inversion of Control Containers and the Dependency Injection pattern

▪ Inversion of Control Containers and the Dependency Injection pattern 한글 번역 by 최범균(Blog 자바캔(Java Can Do It))

앞으로



Markdown Guide

Markdown

1. <https://www.markdownguide.org/basic-syntax/>

Electron Vue Window Toolkit

EVWT provides Vue components, plugins and other tools

Installation

New app:

```
```bash
npm install -g @vue/cli
vue create my-evwt-app
cd my-evwt-app
vue add electron-builder
npm install evwt
```
```

Existing Electron/Vue app:

```
`npm install evwt` (or `yarn add evwt`)
```

Docs

* [Documentation](https://evwt.net/)

* [Cookbook](https://github.com/evwt/evwt/blob/ma

Example Apps

* [Markdown Editor](https://github.com/evwt/evwt-

Electron Vue Window Toolkit

EVWT provides Vue components, plugins and other tools to assist in building great Electron apps.

Installation

New app:

```
npm install -g @vue/cli
vue create my-evwt-app
cd my-evwt-app
vue add electron-builder
npm install evwt
```

Existing Electron/Vue app:

```
npm install evwt (or yarn add evwt )
```

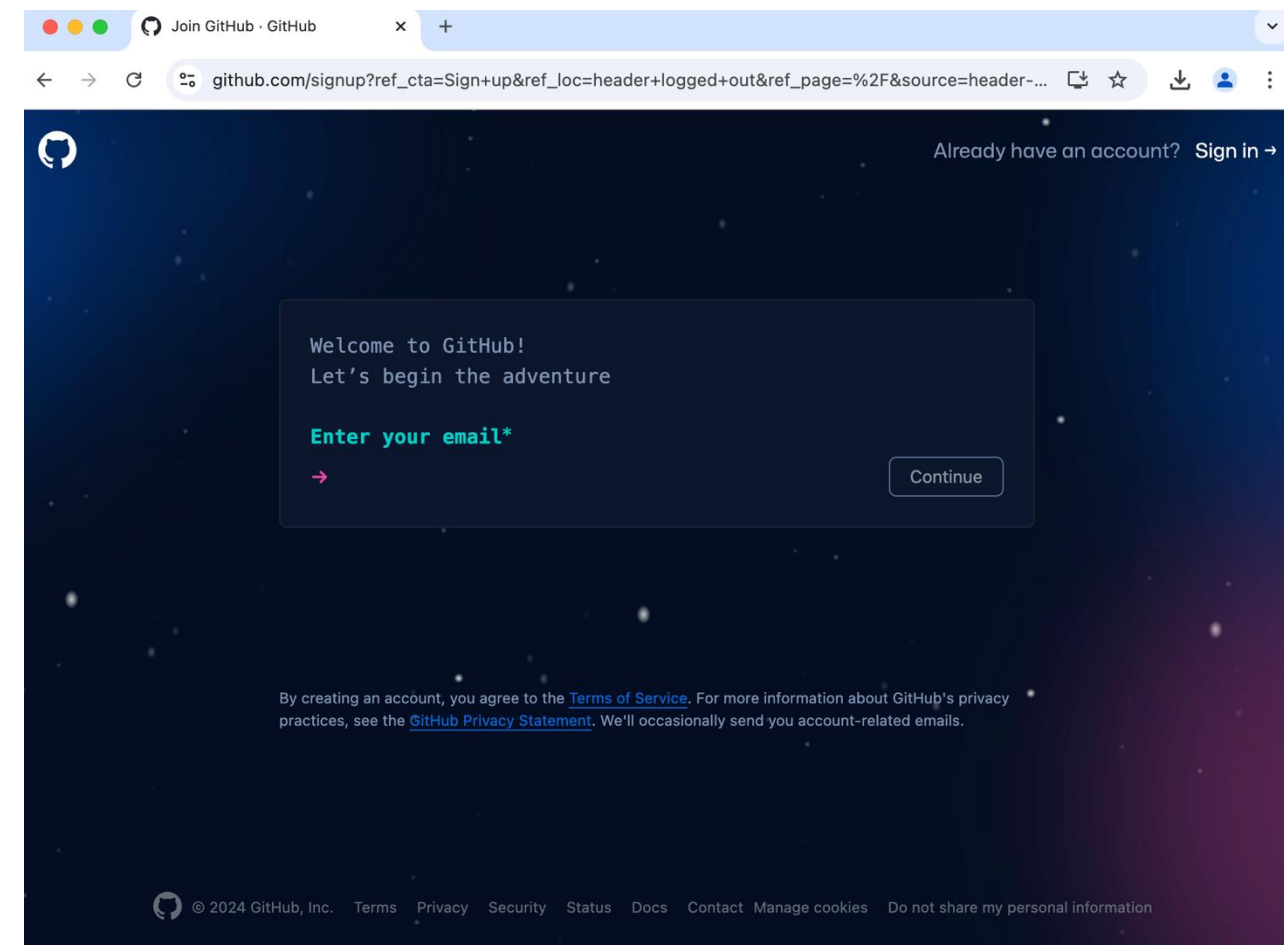
Docs

컨트리뷰션 환경 준비

처음이라면

Github 가입하기

1. <https://github.com/>
2. Sign up
 1. <https://docs.github.com/ko/get-started/start-your-journey/creating-an-account-on-github> (가이드 참고)

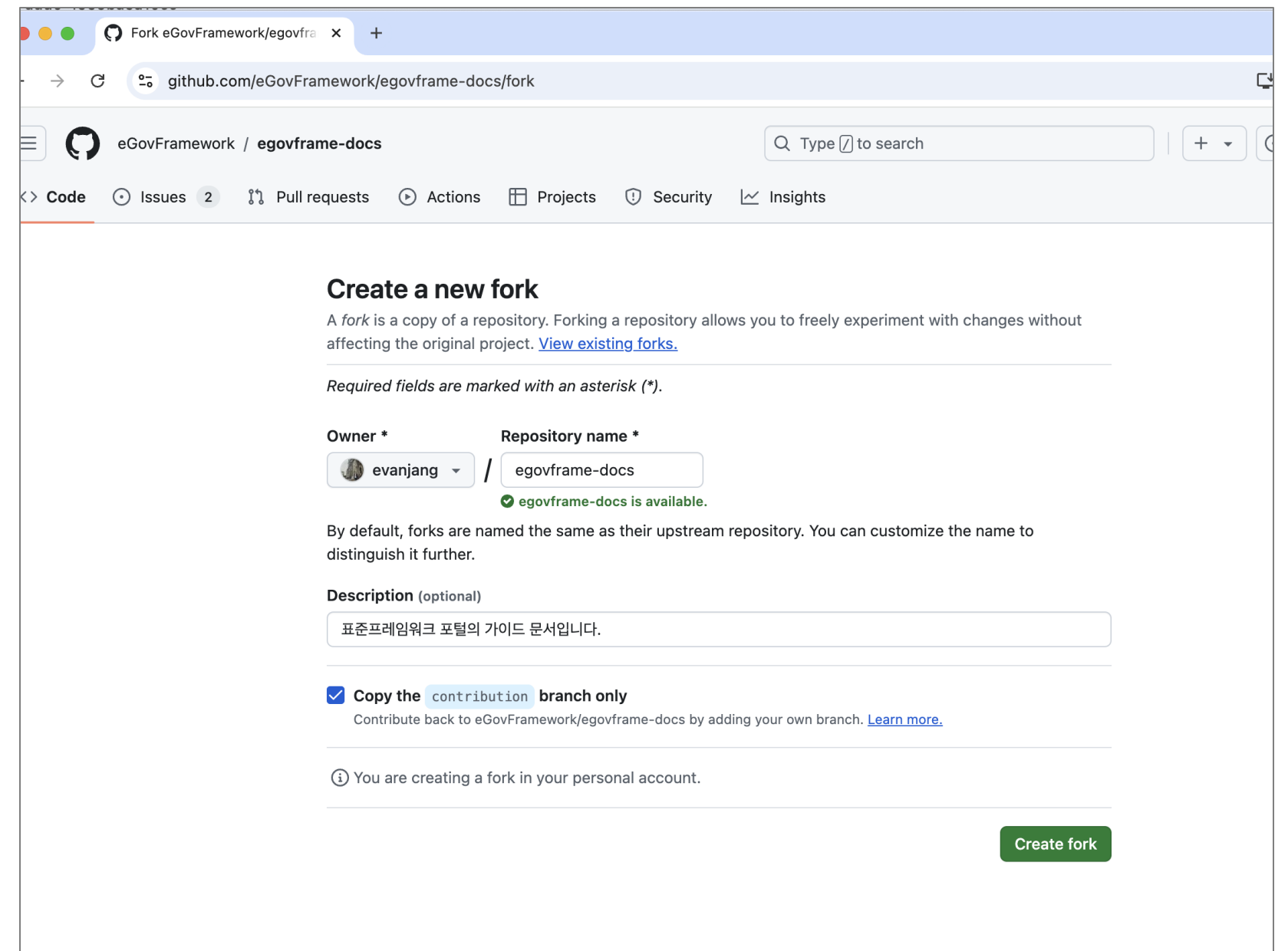


컨트리뷰션 환경 준비

계정이 있다면

표준프레임워크 포털 가이드 문서 Repository

1. <https://github.com/eGovFramework/egovframe-docs>
2. fork

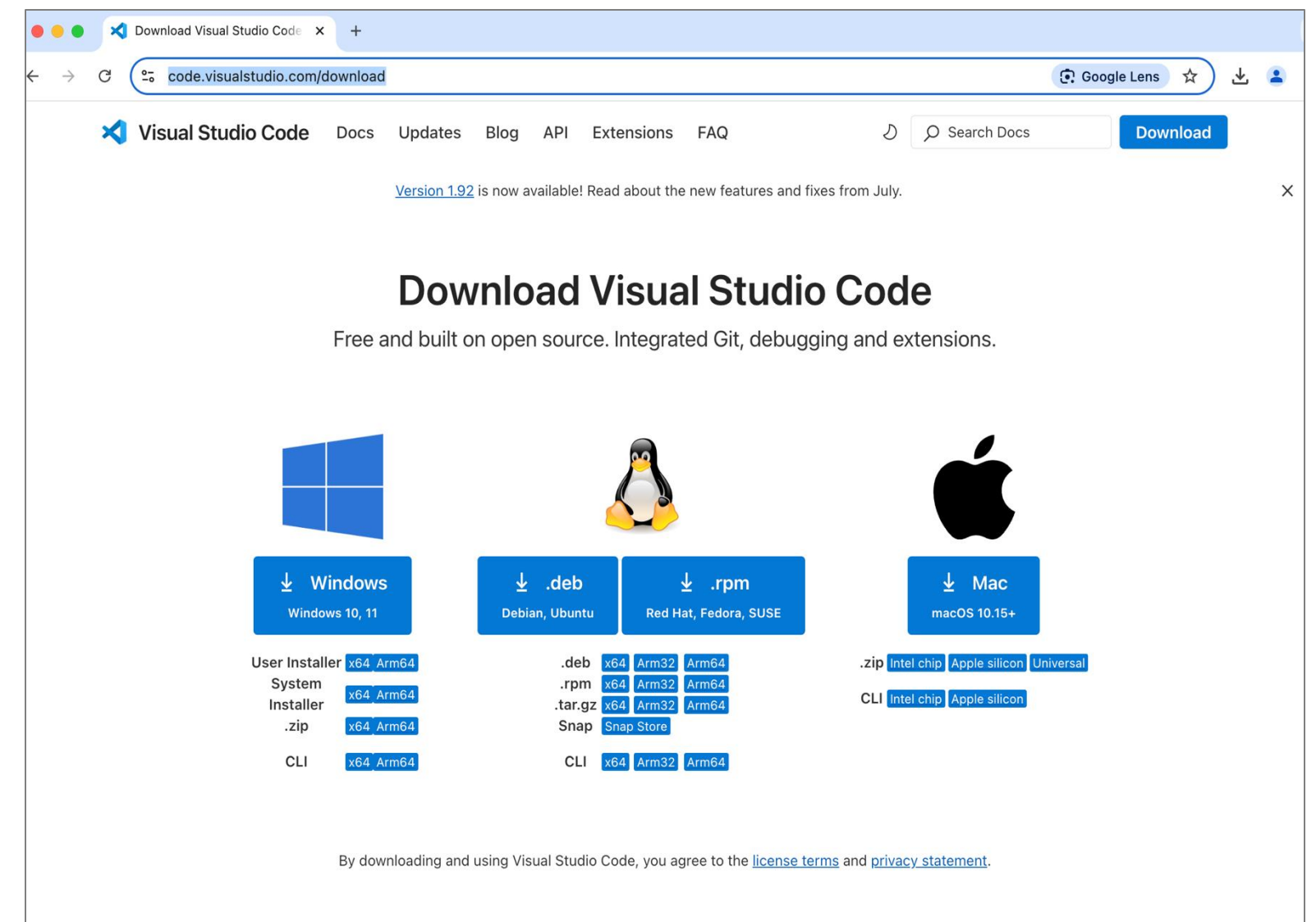


컨트리뷰션 환경 준비

개발환경이 없다면
VSCode 실행/설치

VSCode

1. <https://code.visualstudio.com/download>



컨트리뷰션 환경 준비

clone

EXPLORER

WORKSPACE-EDU

egovframe-docs

egovframe-runtime

batch-layer

business-logic-layer

images

hello-world.jpg

hello-world.png

exception-handling.md

foundation-layer

images

xml_manipulation.jpg

server-security.md

foundation-layer-core

images

egovframework:rte:fdl:aop_001.jpg

egovframework:rte:fdl:aop_002.jpg

egovframework:rte:fdl:aop_003.jpg

egovframework:rte:fdl:ioc_container:container-magic.png

egovframework:rte:fdl:ioc_container:prototype.png

egovframework:rte:fdl:ioc_container:singleton.png

egovframework:rte:ptl:spel1.jpg

egovframework:rte:ptl:spel3.jpg

aop-aspectj.md

aop-egovrteaopguide.md

aop-xmlschema.md

aop.md

ioc-container-annotation-based_configuration.md

ioc-container-basics.md

OUTLINE

TIMELINE

workspace-edu

README.md

Preview ioc-container-basics.md

ioc-container-basics.md

Basics

개요

Spring Framework에서 객체가 생성자 인수, 팩토리 메서드에 대한 인수 또는 객체 인스턴스가 생성되거나 팩토리 메서드에서 반환된 후 객체 인스턴스에 설정된 속성을 통해서만 종속성(함께 작업하는 다른 객체)을 정의하는 프로세스를 제어의 역전(Inversion of Control, IoC)라고 한다. 의존성 주입(Dependency Injection, DI)은 모듈간의 의존성을 모듈의 외부 컨테이너 에서 주입시켜주는 기능으로 IoC 의 한 종류이다.

설명

Spring Framework에서 Bean은 어플리케이션을 구성하고, IoC Container에 의해 관리되어지는 객체로 간단히 말해 IoC Container 에 의해 객체화되고, 조립되고, 또는 관리되는 객체를 의미한다.
Bean들과 Bean들간의 종속성은 Container가 사용하는 설정 메타데이터에 의해 결정된다.

The Container

org.springframework.beans.factory.BeanFactory 인터페이스는 Spring IoC Container의 핵심 인터페이스로 Spring IoC Container는 객체를 생성하고, 객체간의 종속성을 이어주는 역할을 한다.

Configuration Metadata

The Spring Container


Your Business Objects (POJOs)

produces

컨트리뷰션 환경 준비

마크다운 in VSCode

Extension: Markdown All in One



Markdown All in One

v3.6.2

Yu Zhang

9,205,750

★★★★★ (153)

All you need to write Markdown (keyboard shortcuts, table of contents, auto p

Enable

Uninstall

Auto Update

This extension is disabled globally by the user.

DETAILS

FEATURES

CHANGELOG

Markdown Support for Visual Studio Code

vscode marketplace

v3.6.2

downloads

27M

build

passing

github stars

2.9k

contributors

64

All you need for Markdown (keyboard shortcuts, table of contents, auto preview and more).

Note: VS Code has basic Markdown support out-of-the-box (e.g, **Markdown preview**), please see the [official documentation](#) for more information.

Table of Contents


• Features

- Keyboard shortcuts
- Table of contents
- List editing
- Print Markdown to HTML
- GitHub Flavored Markdown
- Math
- Auto completions
- Others

• Available Commands

컨트리뷰션 환경 준비

마크다운 in Chrome




Markdown Preview

Preview markdown text.

Details

Remove



Showdown - Markdown in JavaScript

Markdown Preview

chrome-extension://hpdcheebhjfkdbialimbhoopehil/showdown/showdown-gui.html

☆ | 👤 | ⋮

Showdown

a javascript port of Markdown

Preview

Input

Spring Framework API를 참조한다.

IoC Container of Spring Framework
org.springframework.beans과 org.springframework.context 패키지는 Spring Framework의 IoC Container의 기반을 제공한다.
BeanFactory 인터페이스는 객체를 관리하기 위한 보다 진보된 설정 메커니즘을 제공한다.
BeanFactory 인터페이스를 기반으로 작성된 ApplicationContext 인터페이스(BeaFactory 인터페이스의 sub-interface이다)는 BeanFactory가 제공하는 기능 외에 Spring AOP, 메시지 리소스 처리(국제화에서 사용됨), 이벤트 전파, 웹 어플리케이션을 위한 WebSpplicationContext 등 어플리케이션 레이어에 특화된 context 등의 기능을 제공한다.

요약하면, BeanFactory는 프레임워크와 기본적인 기능에 대한 설정 기능을 제공하는 반면에, ApplicationContext는 좀더 Enterprise 환경에 맞는 기능들을 추가로 제공한다.
ApplicationContext는 BeanFacatory의 완전한 superset이므로, BeanFactory의 기능 및 행동에 대한 설명은 ApplicationContext에도 모두 해당된다.

본 문서는 크게 두 부분으로 나뉘어지는데, 첫번째 부분은 BeanFactory와 ApplicationContext 모두에 적용되는 기본적인 원리를 설명하고, 두번째 부분은 ApplicationContext에만 적용되는 특징들을 설명한다.

* [Basics](https://github.com/eGovFramework/egovframe-docs/blob/main/egovframe-runtime/foundation-layer-core/ioc-container-basics.md)
IoC Container를 설명하기 위해 필요한 기본적인 개념 및 사용 방법을 설명한다.
* [Dependencies]()
IoC Container의 핵심 기능인 Dependency Injection의 사용 방식 및 설정 방법을 설명한다.
* [Bean scope]()
IoC Container에 의해 관리되는 Bean의 생성 방식 및 적용 범위를 설명한다.
* [Customizing the nature of a bean]()
Bean의 생명주기 관리, Bean이 속한 Container 참조 등 Bean의 성질을 변화시키는 방법을 설명한다.
* [Bean definition inheritance]()
Bean 정의 상속에 대해서 설명한다.
* [IoC Container:Container extension points]()
IoC Container의 기능을 확장하는 방법을 설명한다.
* [IoC Container:The ApplicationContext]()
ApplicationContext만이 제공하는 기능을 설명한다.
* [IoC Container:Annotation-based configuration]()
Java Annotation을 기반으로 Bean을 정의하는 방법을 설명한다.
* [IoC Container:Classpath scanning for managed components]()
Dependency Injection에 의해 상입되는 base Bean에 대한 Java Annotation 기반 설정 방법을 설명한다.
* [IoC Container:JSR 330 Standard Annotations]()
JSR-330 표준 Annotation에 대해서 설명한다.
* [IoC Container:Java-based configuration]()
Java Annotation을 기반으로 Container를 구성하는 방법을 설명한다.
* [IoC Container:Environment Abstraction]()
환경 설정을 추상화하는 방법을 설명한다.

참고자료

* [The Spring Framework - Reference Documentation / 1. The IoC Container]
(https://docs.spring.io/spring-framework/docs/5.3.27/reference/html/core.html#beans)
* [Inversion of Control](https://martinfowler.com/bliki/InversionOfControl.html)

IoC Container

개요

프레임워크의 기본적인 기능인 Inversion of Control(IoC) Container 기능을 제공하는 서비스이다.
객체의 생성 시, 객체가 참조하고 있는 타 객체에 대한 종속성을 소스 코드 내부에서 하드 코딩하는 것이 아닌, 소스 코드 외부에서 설정하게 함으로써, 유연성 및 확장성을 향상시킨다.

주요 개념

Inversion of Control(IoC)

IoC는 Inversion of Control의 약자이다. 우리나라 말로 직역해 보면 "역제어"라고 할 수 있다. 제어의 역전 현상이 무엇인지 살펴본다.
기존에 자바 기반으로 어플리케이션을 개발할 때 자바 객체를 생성하고 서로간의 의존 관계를 연결시키는 작업에 대한 제어권은 보통 개발되는 어플리케이션에 있었다.
그러나, Servlet, EJB 등을 사용하는 경우 Servlet Container, EJB Container에게 제어권이 넘어서서 객체의 생명주기(Life Cycle)를 Container들이 전달하게 된다.
이처럼 IoC에서 이야기하는 제어권의 역전이란 객체의 생성에서부터 생명주기의 관리까지 모든 객체에 대한 제어권이 바뀌었다는 것을 의미한다.

관련문서

- Martin Fowler가 저술한 [Inversion of Control](#)
- [Inversion of Control 한글 번역](#)

Dependency Injection

각 클래스 사이의 의존관계를 빈 설정(Bea Definition)정보를 바탕으로 컨테이너가 자동적으로 연결해주는 것을 말한다.
컨테이너가 의존관계를 자동적으로 연결시켜주기 때문에 개발자들이 컨테이너 API를 이용하여 의존관계에 관여할 필요가 없게 되므로 컨테이너 API에 종속되는 것을 줄일 수 있다.
개발자들은 단지 빈 설정파일(자장소 관리 파일)에서 의존관계가 필요하다는 정보를 추가하기만 하면 된다.

관련문서

- Martin Fowler가 저술한 [Inversion of Control Containers and the Dependency Injection pattern](#)
- [Inversion of Control Containers and the Dependency Injection pattern 한글 번역](#) by 최범균(Blog 자바권(Java Can Do It))

사용된 오픈 소스

- IoC Container는 <https://spring.io/projects/spring-framework>를 수정없이 사용한다.

설명

컨트리뷰션 방법은

egovframework:rte4.2:fdl:ioc

egovframe.go.kr/wiki/doku.php?id=egovframework:rte4.2:fdl:ioc_container

egovframework:rte4.2:fdl:ioc_container

소스 보기

이전 버전들

Print view

주최: » overview » brte_v3.7 » 실행환경가이드 » rte4.2 » ioc_container

IoC Container

개요

프레임워크의 기본적인 기능인 Inversion of Control(IoC) Container 기능을 제공하는 서비스이다. 객체의 생성 시, 객체가 참조하고 있는 타 객체에 대한 종속성을 소스 코드 내부에서 하드 코딩하는 것이 아닌, 소스 코드 외부에서 설정하게 함으로써, 유연성 및 확장성을 향상시킨다.

주요 개념

Inversion of Control(IoC)

IoC는 Inversion of Control의 약자이다. 우리나라 말로 직역해 보면 “역제어”라고 할 수 있다. 제어의 역전 현상이 무엇인지 살펴본다. 기존에 자바 기반으로 어플리케이션을 개발할 때 자바 객체를 생성하고 서로간의 의존 관계를 연결시키는 작업에 대한 제어권은 보통 개발되는 어플리케이션에 있었다. 그러나, Servlet, EJB 등을 사용하는 경우 Servlet Container, EJB Container에게 제어권이 넘어가서 객체의 생명주기(Life Cycle)를 Container들이 전달하게 된다. 이처럼 IoC에서 이야기하는 제어권의 역전이란 객체의 생성에서부터 생명주기의 관리까지 모든 객체에 대한 제어권이 바뀌었다는 것을 의미한다.

관련문서

- Martin Fowler가 저술한 Inversion of Control
- Inversion of Control 한글 번역

Dependency Injection

각 클래스 사이의 의존관계를 빈 설정(Bean Definition)정보를 바탕으로 컨테이너가 자동적으로 연결해주는 것을 말한다. 컨테이너가 의존관계를 자동적으로 연결시켜주기 때문에 개발자들이 컨테이너 API를 이용하여 의존관계에 관여할 필요가 없게 되므로 컨테이너 API에 종속되는 것을 줄일 수 있다. 개발자들은 단지 빈 설정파일(저장소 관리 파일)에서 의존관계가 필요하다는 정보를 추가하기만 하면 된다.

관련문서

- Martin Fowler가 저술한 Inversion of Control Containers and the Dependency Injection pattern
- Inversion of Control Containers and the Dependency Injection pattern 한글 번역 by 최병균(Blog 자바캔(Java Can Do It))

EXPLORER

EGOVFRAME-DOCS

egovframe-runtime

batch-layer

images

batch_process.png

intro-outline.png

xml_manipulation.jpg

intro.md

business-logic-layer

images

exception-handling.md

foundation-layer

images

xml_manipulation.jpg

server-security.md

foundation-layer-core

images

aop-aspectj.md

aop-egovrteaopguide.md

aop-xmlschema.md

aop.md

intro.md

Preview intro.md

aop-egovrteaopg

egovframe-runtime > batch-layer > intro.md > # 배치 실행환경

배치 실행환경 소개

개요

전자정부 표준프레임워크에서의 대용량 데이터 처리 지원을 위해 작업수

목표

배치 실행환경은 대용량 데이터를 처리를 위한 기반 환경을 제공함으로써

배치 실행환경 수행 과정

전자정부 표준프레임워크 실행환경에 추가된 배치 실행환경은 3-Tier(

제공한다.

! [내용] (./images/batch_process.png)

주요개념

EXPLORER

EGOVFRAME-DOCS

egovframe-runtime

batch-layer

images

batch_process.png

intro-outline.png

xml_manipulation.jpg

intro.md

business-logic-layer

images

exception-handling.md

foundation-layer

images

xml_manipulation.jpg

server-security.md

foundation-layer-core

images

aop-aspectj.md

aop-egovrteaopguide.md

aop-xmlschema.md

aop.md

intro.md

Preview intro.md

aop-egovrteaopguide.md

overview.md

배치 실행환경 소개

개요

전자정부 표준프레임워크에서의 대용량 데이터 처리 지원을 위해 작업수행 및 결과 관리 및 스케줄링 관리 기능을 제공한다.

목표

배치 실행환경은 대용량 데이터를 처리를 위한 기반 환경을 제공함으로써 배치 실행에 필요한 핵심 기능을 제공한다.

배치 실행환경 수행 과정

전자정부 표준프레임워크 실행환경에 추가된 배치 실행환경은 3-Tier(Run, Job, Application Tier)로 구성되며, 대용량 데이터를 처리를 위한 기반 환경을 제공

Run Tier

Job Tier

Application Tier

Scheduler

Http, Web Service

Command Line

JobRunner <Java>

JobLauncher

eGovJob

Job <XML>

eGovStep

Step <XML>

Item Reader

ItemProcessor <Java>

Item Writer

