




2023 오픈소스 컨트리뷰션 아카데미

Open Source Contribution Academy

A type-safe, composable library **ZIO**

FINAL EVALUATION

주최  과학기술정보통신부

주관  nipa 정보통신산업진흥원 

운영  ipa Small Business Development Association
한국기비즈니스진흥협회

목차

01 프로젝트
소개

02 목표

03 진행 방법

04 오픈소스
기여

05 성과

프로젝트 소개

ZIO란?

- scala 기반
- 타입 안전성(ZIO type)
- 명령형 + 함수형

ZIO 타입 **ZIO[-R, +E, +A]**

R**Environment** 의존성**E****Failure** 에러**A****Success** 결과

프로젝트 소개

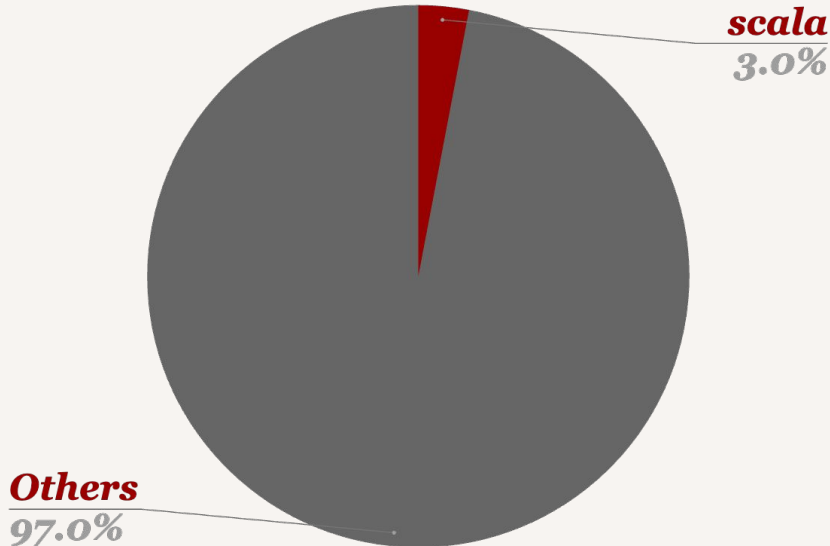
ZIO 생태계

- zio-http
- zio-json
- zio-quill
- zio-kafka

실용적 라이브러리

최근 1년 내 사용한 프로그래밍 언어

Published by Lionel Sujay Vailshery , Feb 20, 2023



목표

초기

ZIO 생태계 기여



변경 후

함수형 디자인 응용을
통하여

실무에서 유용한
프로그래밍 스킬 향상

진행 방법

주 2회 오프라인 + 1회 온라인

정규모임

실무에서 사용되는 기법

- JSON parsing
 - 핵심 요구사항 추출
- Database 설정
 - 의존성 주입 + CRUD Application 제작
- 테스트 주도 개발(TDD)
 - E2E & Unit Test 작성



Deep ZIO

함수형 프로그래밍

- ADT를 이용한 데이터 모델링
 - 합타입과 곱타입
- flatMap과 map 이해
 - 모나드와 펄터
- 선언형 프로그래밍
 - 선언적 인코딩과 에러 처리
 - Effect system



오픈소스 기여

1. **working-scala**

모임자료 오픈소스화

2. **ZIO 생태계에 기여**

zio-http, zio-json

3. **cheese-shop**

도메인 로직을 가진 scala 프로젝트

1. **working-scala** 모임자료 오픈소스화

db-connection

declarative-prog

dependency-injection

error-handling

file-handling

for-comprehension-in-scala

json-handling

multi-project

testing

2. ZIO 생태계에 기여

refactor zio-http

- 1) `element`의 길이를 직접 비교하는 로직을 단순화
- 2) 스키마를 길이로 판단하는 로직 변경

bug-fix zio

- 저장소 내 행동 지침 및 기여 가이드라인에 대한 심볼릭 링크 수정

bug-fix zio-query

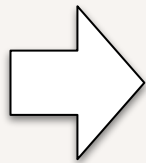
- ZIO 산하 저장소들이 문서 생성, CI 작업 등을 위해 공통적으로 사용하는
플러그인 버전 업데이트

3. cheese-shop 도메인 로직을 가진 scala 프로젝트

문제

- 모임 참여자 동일
- 예제코드 크기 방대, 관리 어려움
- 고도화된 주제*
- 도메인 로직의 필요성

*) resource control, mocking, scheduling, streaming



조건

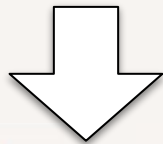
- 여러 자원을 이용
- 연속성 있는 코드
- 하나의 프로젝트로 관리
- 여러 도메인이 연합

실무에 가까운 협업 경험

성과

함수형 디자인 응용을 통하여 실무에서 유용한 프로그래밍

스킬 향상



하드 스킬

- ZIO 타입으로 타입 안전한 에러 처리
- 발생 가능한 에러 추상화
- 선언형 프로그래밍
- ZLayer로 의존성 주입
- 아키텍처 설계하기
- ZIO의 runtime
- Fiber로 자원 관리
- ZIO로 테스트 작성
- 공식 문서 읽는 방법
- 오픈소스 레포지토리에서 예제코드를 찾는 방법
- 패턴매칭으로 ADT 분해하기

소프트 스킬

- 페어 프로그래밍 하는 방법
- 온보딩 전략
- GitHub를 통해 협업하기
- 코드 리뷰
- 핵심 요구사항 추출

**THANK
YOU**

Q&A