





ZeroDay Patch Monitor

실시간 게임 패치노트 자동화 시스템

 웹 스크래핑  자동 번역  데이터 비교  알림 시스템

48시간 챌린지로 구현한 Python 기반 게임 정보 크롤링 및 알림 서비스

 2025-11-17

프로젝트 배경 및 문제점



게임 정보의 파편화

전 세계적으로 방대한 정보가 파편화되어 존재하며, 이는 게이머와 개발자 모두에게 큰 어려움을 야기합니다. 다양한 플랫폼과 언어로 분산된 정보는 접근성을 떨어뜨리고, 중요한 업데이트를 놓칠 수 있습니다.



언어 장벽

패치 및 업데이트 내용은 영어, 한국어 등 다양한 언어로 제공되며, 이로 인해 정보 접근성이 저하됩니다. 언어 장벽은 게임 커뮤니티의 활성화를 저해하고, 사용자 경험을 전략 수립의 어려움으로 이끕니다.



서버 간 업데이트 시차

국내외 서버 간의 업데이트 시차는 정보의 불균형을 심화시켜 사용자 경험 저하를 초래합니다. 이러한 문제들은 게임 커뮤니티의 활성화를 저해하고, 중요한 게임 정보를 놓치게 만드는 주요 원인이 됩니다.

PatchCatch 솔루션 개요

48시간 내에 Python(Flask) 기반의 실시간 게임 패치노트 자동화 시스템으로, 웹 스크래핑, 자동 번역, 데이터 저장 및 비교, 이메일 알림 기능을 통합하여 게임 정보 수집부터 전달까지의 전 과정을 자동화한 서비스입니다.



웹 스크래핑

Requests & BeautifulSoup을 활용하여 대상 게임 웹사이트에서 패치노트 정보를 실시간 수집합니다.



자동 번역

Googletrans 라이브러리를 사용하여 북미 서버의 영어 패치노트를 자동으로 한국어로 번역합니다.



데이터 저장 및 비교

JSON 파일 기반 데이터베이스를 사용하여 패치 데이터를 저장하고, 버전 비교 로직으로 새로운 패치를 감지합니다.



이메일 알림

smtplib 라이브러리를 활용하여 새로운 패치가 감지될 경우 사용자에게 즉시 이메일 알림을 발송합니다.

48h

48시간 내에 구현한 Flask 기반 웹 서비스

시스템 아키텍처 및 데이터 파이프라인

🔗 MVC 패턴 기반 Flask 프레임워크



Model

데이터베이스 및 business logic



View

Frontend 템플릿 및 렌더링



Controller

Request 처리 및 로직 흐름

≡ 자동화 데이터 파이프라인

1. 수집

Requests, BeautifulSoup

게임사 웹사이트에서 패치노트 정보를
실시간으로 수집



2. 가공

Googletrans, re

영어 패치노트를 자동으로 한국어로 번
역하고 데이터를 정제



3. 저장

JSON

패치 데이터를 파일 기반 JSON DB에 저
장하여 관리



4. 알림

smtpplib

새로운 패치 감지 시 사용자에게 이메일
알림 발송



핵심 기술 구현 - 필터링 및 버전 비교



패치노트 이중 검증 필터링 로직

게임사 공지사항 페이지에 다양한 정보가 혼재되어 있습니다. PatchCatch는 순수한 패치노트만을 정확하게 선별하기 위해 '이중 검증' 필터링 로직을 구현했습니다.

```
# Python 반복문과 조건문으로 패치노트 필터링
def filter_patch_notes(title_list):
    filtered_titles = []
    for title in title_list:
        # 'Patch'와 'Notes' 키워드 포함, 'TFT' 제외
        if "Patch" in title and "Notes" in title and "TFT" not in title:
            filtered_titles.append(title)
    return filtered_titles
```



정규표현식을 활용한 버전 비교

북미(NA) 서버와 한국(KR) 서버 간의 패치 버전 동기화 상태를 실시간으로 판별하기 위해, 정규표현식을 활용한 버전 비교 알고리즘을 개발했습니다.

```
# re 라이브러리를 사용한 버전 추출 및 비교
import re

def extract_version(title):
    match = re.search(r'(\d+\.\d+)', title)
    if match:
        return float(match.group(1))
    return 0.0 # 버전 없을 경우 기본값

def compare_server_versions(na_version_title,
                             kr_version_title):
    na_version = extract_version(na_version_title)
    kr_version = extract_version(kr_version_title)

    if na_version > kr_version:
        return "북미 선행 공개"
    elif na_version == kr_version and na_version != 0.0:
        return "동기화 완료"
    else:
        return "상태 불일치"
```

킬러 피처 - LoL 서버 비교 시연

 북미(NA) 서버

14.23

 선행 공개

 최신 패치: 2시간 전

VS

 한국(KR) 서버

14.21

 동기화 대기

 최신 패치: 4시간 전

실시간 동기화 감지

서버 간 패치 버전 차이를 실시간으로 감지합니다.

정보 시차 해결

국내외 서버 간 업데이트 시차로 인한 정보 불균형을 해결합니다.

게임 전략 지원

업데이트 정보로 사용자가 정확한 게임 전략을 수립합니다.

최종 결과물 및 대시보드 UI

주요 기능



반응형 디자인

Bootstrap 프레임워크를 기반으로 한 반응형 웹 디자인으로, 다양한 장치에서 최적의 화면 크기를 제공합니다.



NEW 배지 자동 활성화

새로운 패치가 감지되면 해당 게임 카드에 자동으로 "[NEW]" 배지가 활성화되어 사용자가 최신 정보를 즉시 인지할 수 있도록 돕습니다.



서버 버전 비교

북미(NA) 서버와 한국(KR) 서버의 패치 버전을 실시간으로 비교하여 "북미 선행 공개" 또는 "동기화 완료" 상태를 직관적으로 표시합니다.

PatchCatch 대시보드

[자동 새로고침](#)

리그 오브 레전드 **NEW** NA: 14.23 KR: 14.22 **북미 선행 공개**

발로란트 **NEW** NA: 2.02 KR: 2.02 **동기화 완료**

이터널 리턴 NA: 1.5 KR: 1.5 **동기화 완료**

결론 및 향후 계획



성과 요약

48시간이라는 제한된 시간 내에 웹 크롤링을 통한 정보 수집, 데이터 처리 및 번역, JSON 기반의 경량 DB를 활용한 상태 관리, 그리고 SMTP를 통한 알림 발송에 이르는 풀스택 자동화 파이프라인을 성공적으로 구축했습니다.

- 게임 정보의 파편화 문제 해결
- 언어 장벽 극복 (영어 → 한국어 자동 번역)
- 서버 간 업데이트 시차로 인한 정보 불균형 해소



향후 계획

- **게임 확장:** LoL, 발로란트, 이터널 리턴 외에 더 많은 인기 게임의 패치노트를 통합하여 서비스 범위를 확장할 예정입니다.
- **모바일 푸시 알림:** 이메일 알림을 넘어 모바일 푸시 알림 기능을 연동하여, 사용자가 언제 어디서든 최신 게임 패치 정보를 실시간으로 받아볼 수 있도록 할 것입니다.
- **사용자 경험 향상:** 직관적인 대시보드 디자인과 필터링 기능을 통해 사용자에게 보다 나은 서비스를 제공할 예정입니다.

감사합니다