

# Price scraper

Autor: Tomasz Załęski

# Idea

Utworzenie narzędzia do wyszukiwania produktów i automatycznego obserwowania ich cen

Użytkownik posiada jedno miejsce gdzie może szukać produktu na wszystkich interesujących go stronach jednocześnie



# Problem

Każda strona w inny sposób prezentuje swoje produkty użytkownikowi - nie można utworzyć jednej klasy, która byłaby w stanie zdobyć informacje z każdej strony.

Aby dodać obsługę kolejnego sklepu trzeba by ingerować w kod programu



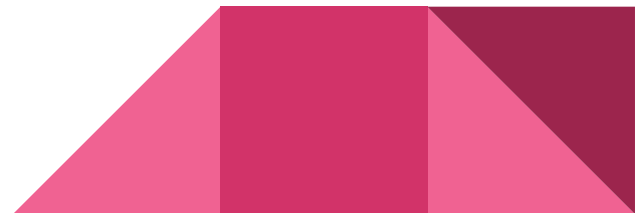
# Rozwiązanie

Odwrócenie kontroli poprzez wstrzykiwanie zależności

Każdy sklep jest reprezentowany przez obiekt w JSON'ie, który jest wczytywany przy uruchomieniu programu

Sklepy posiadają podstawowe informacje jak: nazwa, strona wyszukiwania, tagi; oraz atrybuty, które są obiektami specjalnie przystosowanymi do wyciągania informacji ze strony za pomocą selektorów CSS

Umożliwia to łatwe dodawanie nowych/edycję obecnych sklepów w przyszłości



# Wykonanie - karta search

Użytkownik po uruchomieniu aplikacji okienkowej jest na karcie search, gdzie może po lewej stronie aktywować i deaktywować wyszukiwanie na poszczególnych sklepach

Po wpisaniu dowolnej frazy i naciśnięciu przycisku search prezentowane są wyniki z pierwszych stron wyszukiwania ze wszystkich wybranych sklepów. Można otworzyć stronę produktu w przeglądarce, aby dowiedzieć się więcej, albo dodać do listy obserwowanych produktów



# Wykonanie - karta watch

Na karcie watch są umieszczone wszystkie obserwowane produkty

Możemy:

- aktywować / deaktywować obserwowanie zaznaczając pole wyboru
- zmienić czas co jaki dany produkt będzie odświeżany (w minutach, domyślnie 24 godziny, może wymagać naciśnięcia przycisku enter po wpisaniu nowego czasu)
- obserwować wykres zależności ceny od czasu



# Wykonanie - obserwator

Po uruchomieniu programu w trybie obserwowania program automatycznie określa co jaki czas powinien się obudzić, aby pobrać obecną cenę

Aby wczytać zmiany w liście obserwowanych produktów trzeba program zabić i uruchomić ponownie.



# Pomysły na przyszłość

Dodanie ceny granicznej, po przekroczeniu której zostanie automatycznie wysłany email informujący o promocji.

Dodanie obsługi filtrowania sklepów po tagach

Dodanie kategorii produktów - z każdą kategorią są powiązane specjalne atrybuty. Jeśli atrybut istnieje na stronie to jest wyciągany i wyświetlany w aplikacji

Ulepszenie UI





# Wykorzystane biblioteki

ScalaFX - Do aplikacji okienkowej

Scala-scrapers - Do wczytywania i wyciągania danych ze stron internetowych

Spray - Do wczytywania plików JSON

