

# Схема работы сервиса

## Инициализация зависимостей

*scrapper.NewDep() Dep*

Создается структура, которая хранит:

1. Подключение к базе данных
2. Значений системных переменных (данные почты-отправителя, таймауты между парсингом ссылок)
3. Http.Client и интерфейс функции отправителя писем для возможности тестирования.

## Оформление подписки

*subscribe.Handler(Dep) http.Handler*

1. Запускается http Handler, который принимает POST запросы, содержащие поля email и link.
2. Запускаем Парсинг полученной от пользователя ссылки, для получения стоимости товара. Если стоимость получить не удалось- return.
3. Сохраняем в базу данных (табл. link) ссылку и ее цену.
4. Сохраняем в базу данных (табл. email) емэйл и ссылку.  
Если емэйл уже есть в базе данных- переписываем его ссылку. Если перезаписанная(старая) ссылка была связана только с этой почтой, то удаляем ее из таблицы link.

## Проверка обновлений цены + рассылка

*go scrapper.ComparePrices(Dep)*

1. Получаем все ссылки и их цены из базы данных.
2. Для полученных ссылок в цикле запускаем парсинг и сверяем значение их цены с ценой полученной во время парсинга. Если цены не совпадают- перезаписываем значения в базе данных на актуальные и сохраняем ссылку в список (слайс). Делаем таймауты после парсинга каждой ссылки, для избежания блокировки .
3. Для ссылок с не актуальной информацией находим почты из базы данных.
4. Отправляем письма на почты о том что цена изменилась.
5. После проверки всех ссылок ждем таймаут и начинаем цикл сначала.

### Структура бд

link	
PRIMARY KEY(link)	price
<a href="https://www.avito.ru/exampe_ad_link">https://www.avito.ru/exampe_ad_link</a>	100000

email	
PRIMARY KEY(email)	link
<a href="mailto:example@mail.com">example@mail.com</a>	<a href="https://www.avito.ru/exampe_ad_link">https://www.avito.ru/exampe_ad_link</a>