## Способы парсинга [wyscout.com](https://wyscout.com/)

1. Ручное сохранение страниц на локальный диск. Парсинг сохраненных страниц.
2. Написание расширения для автоматического парсинга во время серфинга по [wyscout.com](https://wyscout.com/).

## Полезные ссылки

[Google Chrome Extensions Documentation](https://developer.chrome.com/extensions)

## Что такое расширения? [#](https://developer.chrome.com/extensions)

Расширения являются маленькими программами, которые могут изменять и улучшать функциональные возможности браузера Chrome. Вы пишите их, используя веб технологии, такие как HTML, JavaScript и CSS.

Расширения имеют минимальный пользовательский интерфейс или вообще его не имеют. Например, на изображении справа показана иконка, которая представляет UI для [Google Mail Checker extension](https://developer.chrome.com/samples#google-mail-checker).

Расширения связывают все свои файлы в один файл, который пользователь скачивает и устанавливает. Это связывание означает, что в отличие от обычных веб-приложений, расширения не должны зависеть от веб-контента.

Вы можете распространять ваше расширение с помощью [Chrome Developer Dashboard](https://chrome.google.com/webstore/developer/dashboard) опубликовывая на [Chrome Web Store](http://chrome.google.com/webstore). Для получения дополнительной информации смотрите документацию для store разработчика.

## Беглый обзор [#](https://developer.chrome.com/extensions/overview)

После того, как вы закончите чтение этой страницы и [Руководства по началу работы](https://developer.chrome.com/extensions/getstarted), вы будете готовы к написанию расширений.

### Основы [#](https://developer.chrome.com/extensions/overview#what)

Расширения представляют собой сжатую пачку файлов – HTML, CSS, JavaScript, изображения и другие необходимые файлы – они добавляют функциональность в браузер Google Chrome. Расширения – это в основном веб-страницы, и они могут использовать все [API, которые браузер предоставляет для веб-страниц](https://developer.chrome.com/extensions/api_other), от XMLHttpRequest до JSON в HTML5.

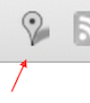
Расширения могут взаимодействовать с веб-страницами или серверами, используя [сценарии содержания](https://developer.chrome.com/extensions/content_scripts) или [XMLHttpRequests с перекрестными источниками](https://developer.chrome.com/extensions/xhr). Расширения также могут взаимодействовать с элементами браузера, такими как [закладки](https://developer.chrome.com/extensions/bookmarks) и [вкладки](https://developer.chrome.com/extensions/tabs).

#### UI расширений [#](https://developer.chrome.com/extensions/overview#extension-ui)

Многие расширения, но не Chrome Apps, добавляют UI в Google Chrome в действия браузера или действия страницы. У каждого расширения может быть не более одного действия браузера или действия страницы. Выберите действие браузера, если расширение применимо к большинству страниц. Выберите действие страницы, если иконка расширения должна быть активной или неактивной (серой), в зависимости от страницы.



Это [расширение Google Mail Checker](https://developer.chrome.com/extensions/samples#google-mail-checker) использует действие браузера.



Это расширение [Mappy](https://developer.chrome.com/extensions/samples#mappy) использует действие страницы и сценарии содержания (код встраивается в веб-страницу).



Это расширение [Set Page Color](https://developer.chrome.com/extensions/samples#set_page_color) содержит действие браузера, при нажатии на которое отображается раскрывающийся список.

Расширения (и Chrome Apps) могут также отображать UI другими способами, такими как добавления в контекстное меню Chrome, предоставления страниц параметров или использование сценариев содержания, изменяющих способ просмотра страниц. Полный список функций смотрите в [Developer’s Guide](https://developer.chrome.com/extensions/devguide) со ссылками на подробные описания реализации каждой из них.

### Файлы [#](https://developer.chrome.com/extensions/overview#files)

Каждое расширение содержит следующие файлы:

* Файл манифеста.
* Один или несколько HTML-файлов (если расширение не является темой).
* Опционально: один или несколько файлов JavaScript.
* Опционально: любые другие файлы, необходимые вашему расширению, например, изображения.

Работая над своим расширением, вы помещаете все эти файлы в единственную папку. Когда вы распространяете свое расширение, содержимое папки упаковано в специальный zip-файл с суффиксом .crx. Если вы загружаете свое расширение с помощью [Панели разработчика Chrome](https://chrome.google.com/webstore/developer/dashboard), то файл .crx создается для вас. Подробные сведения о распространении расширений смотрите в разделе [Hosting](https://developer.chrome.com/extensions/hosting).

#### Обращение к файлам [#](https://developer.chrome.com/extensions/overview#relative-urls)

Вы можете поместить любой файл, который вам нравится, в расширение, но как вы его будете использовать? Как правило, вы можете ссылаться на файл, используя относительный URL, как на обычной веб-странице. Вот пример ссылки на файл с именем myimage.png, находящийся в папке с именем images.

<img src="images/myimage.png">

Как вы могли заметить, когда вы используете отладчик Google Chrome, каждый файл в расширении также доступен по абсолютному URL, например,

chrome-extension://<extensionID>/<pathToFile>

В этом URL,

<extensionID> – это уникальный идентификатор, который система расширений генерирует для каждого расширения. Вы можете посмотреть идентификаторы всех загруженных расширений, перейдя по URL chrome://extensions.

<pathToFile> – местоположение файла в верхней папке расширения, это то же самое, что и относительный URL.

Пока вы работаете над расширением (до его упаковки), идентификатор расширения может измениться. В частности, идентификатор распакованного расширения изменится, если вы загрузите расширение из другого каталога; идентификатор снова изменится, когда вы упакуете расширение. Если код расширения должен указать полный путь к файлу внутри расширения, вы можете использовать [предопределенное сообщение](https://developer.chrome.com/extensions/i18n#overview-predefined) @@extension\_id, чтобы избежать жесткого кодирования идентификатора во время разработки.

Когда вы упаковываете расширение (как правило, загружая его с панели мониторинга), расширение получает постоянный идентификатор, который остается неизменным даже после обновления расширения. Если идентификатор расширения постоянен, вы можете заменить все вхождения @@extension\_id на реальный идентификатор.

#### Файл манифеста [#](https://developer.chrome.com/extensions/overview#manifest)

Файл манифеста, именуемый manifest.json, содержит информацию о расширении, такую как информация о наиболее важных файлах и возможностях, которые может использовать расширение. Вот типичный файл манифеста для action браузера, использующего информацию из google.com:

Для получения дополнительной информации смотрите [Файлы манифеста](https://developer.chrome.com/extensions/manifest).

### Архитектура [#](https://developer.chrome.com/extensions/overview#arch)

У многих расширений есть фоновая страница, невидимая страница, которая содержит основную логику расширения. Расширения может также содержать другие страницы, предоставляющие пользовательский интерфейс расширения. Если расширение должно взаимодействовать с веб-страницами, которые загружает пользователь (в отличие от страниц, которые включены в расширение), расширение должно использовать сценарии содержания.

#### Фоновая страница [#](https://developer.chrome.com/extensions/overview#background_page)

Фоновые страницы, определенные как background.html, могут включать в себя код JavaScript, который контролирует поведение расширения. Существует два типа фоновых страниц: [постоянные фоновые страницы](https://developer.chrome.com/extensions/background_pages) и [страницы событий](https://developer.chrome.com/extensions/event_pages). Постоянные фоновые страницы, как следует из названия, постоянно открыты. Страницы событий открываются и закрываются по мере необходимости. Если вам совершенно не нужна фоновая страница, работающая все время, то используйте страницу события.

Для получения дополнительной информации смотрите [страницы событий](https://developer.chrome.com/extensions/event_pages) и [фоновые страницы](https://developer.chrome.com/extensions/background_pages).

#### Страницы пользовательского интерфейса [#](https://developer.chrome.com/extensions/overview#pages)

Расширения могут содержать обычные HTML-страницы, отображающие пользовательский интерфейс расширений. Например, действие браузера может иметь всплывающее окно, образованное HTML-файлом. Любое расширение может иметь страницу параметров, которая позволит пользователю настроить работу расширения. Другой тип специальной страницы – переопределенная страница. И, наконец, вы можете использовать [tabs.create](https://developer.chrome.com/extensions/tabs#method-create) или window.open для отображения любых других HTML-файлов, находящихся в расширении.

HTML-страницы внутри расширения имеют полный доступ к DOM друг друга, и они могут вызывать функции друг друга.

На следующем рисунке показана архитектура всплывающего окна действия браузера. Содержимое всплывающего окна – это веб-страница, определяемая HTML-файлом (popup.html). Это расширение имеет фоновую страницу (background.html). Всплывающему окну не нужно дублировать код, который находится в фоновой странице, потому что всплывающее окно может вызвать функции фоновой страницы.

