Coleta Online de Dados

Usando recursos do Chrome Extension

WebMedia 2019 – Minicurso 2

Instrutor e Co-Autores

- André Paulino de Lima (instrutor)
  Doutorando em Ciência da Computação no ICMC/USP
- Prof. Dr. Marcelo Garcia Manzato (coordenador)
  Professor e Pesquisador do ICMC/USP
- Profa.Dra. Maria da Graça Pimentel (coordenadora)
  Professora e Pesquisadora do ICMC/USP

Como surgiu a ideia desse curso

- Pesquisa na área de Sistemas de Recomendação
  - Interesse no problema de superespecialização
- Na revisão bibliográfica:

  Ronald E. Robertson, David Lazer, and Christo Wilson. 2018. ***Auditing the Personalization and Composition of Politically-Related Search Engine Results Pages****. In Proceedings of the 2018 World Wide Web Conference (WWW '18). International World Wide Web Conferences Steering Committee, Republic and Canton of Geneva, Switzerland, 955-965. DOI: https://doi.org/10.1145/3178876.3186143*
- Útil para pesquisa em IHC, comportamento online
- Baixo custo de desenvolvimento

Objetivos do curso

- Introdução ao uso do mecanismo de extensão do navegador Chrome para coleta de dados de pesquisa acadêmica
- Nível básico: primeiro contato com os principais componentes e APIs
- Organizado em torno de quatro atividades:
  - Atividade 1 - Coletar dados do histórico
  - Atividade 2 - Coletar resultados de busca (usando o Google *search engine*)
  - Atividade 3 - Coletar dados de identificação do usuário participante da pesquisa
  - Atividade 4 - Salvar os dados coletados em um repositório no GitHub
  - Cada atividade tem três segmentos: **introdução, hands-on, revisão**

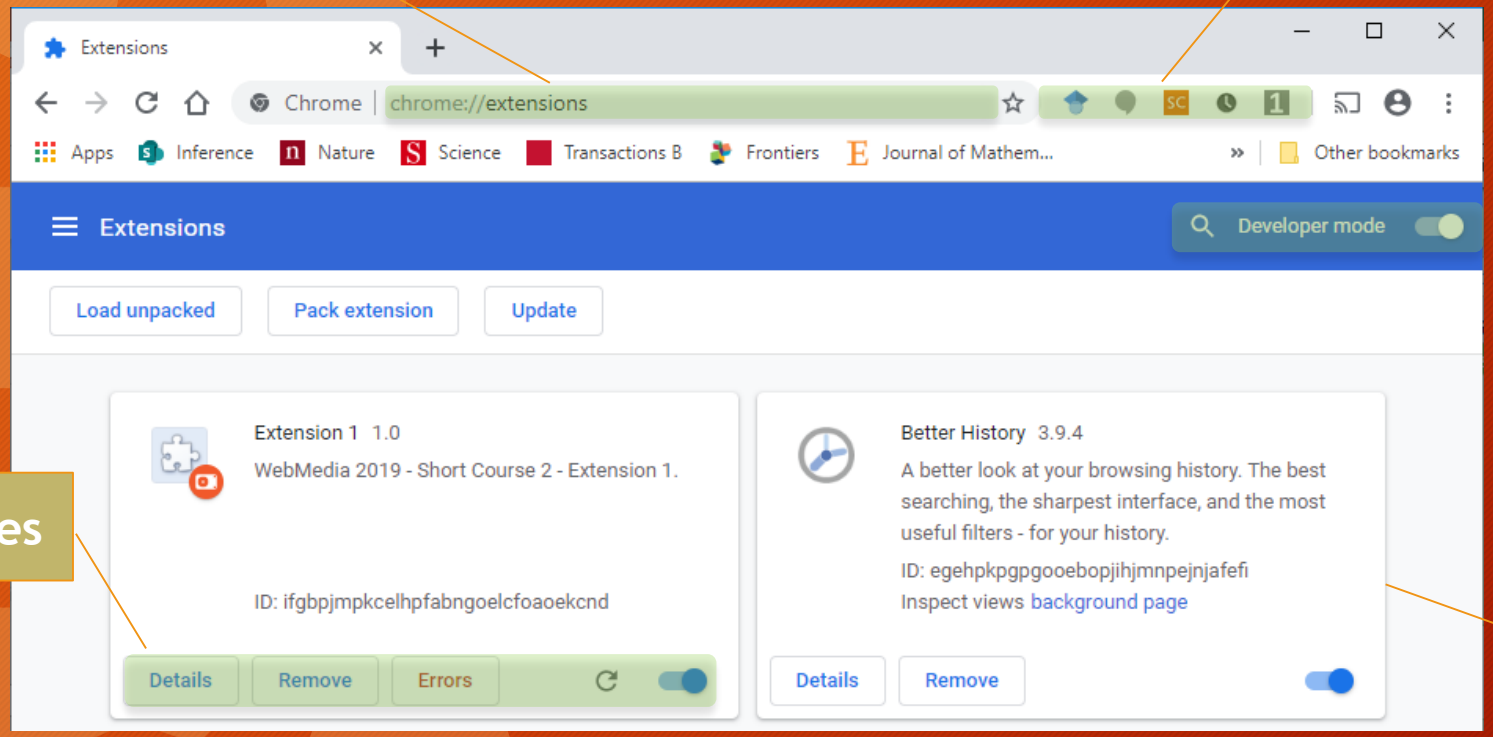
Preliminares - Perspectivas

- Da Google:
  - Personalizar a experiência do usuário na interação com o navegador
  - Implementa uma funcionalidade simples (e bem definida)
- Do usuário da extensão:
  - Um botão que aparece do lado da barra de endereço
  - Simples de instalar – é só entrar na loja e escolher uma
- Do desenvolvedor da extensão (pesquisador)
  - Uma forma barata de coletar dados para pesquisa
  - Usa somente tecnologias web comuns (HTML, JavaScript)

Preliminares - Terminologia básica

omnibox

extensões ativas



Página de gestão de extensões

controles

Painel de gestão da extensão

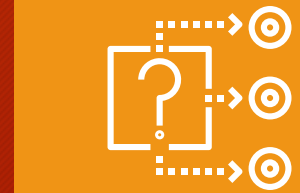
Preliminares - Componentes

- Manifesto
  Especificação da extensão (quais componentes e APIs são usados)
- Elementos da interface com o usuário (*UI elements*)
  Interfaces para interação com o usuário
- Script de Background (*background script*)
  Tratador de eventos
- Script de Conteúdo *(content script)*
  Acessa o conteúdo de uma página (DOM)
- Página de opções (*options page*)
  Interface para a configuração de parâmetros da extensão

Preliminares - Abordagem incremental

Atividade 1	Atividade 2	Atividade 3	Atividade 4
manifest	manifest	manifest	manifest
UI elements	UI elements	UI elements	UI elements
background script	background script	background script	background script
content script	content script	content script	content script
options page	options page	options page	options page
APIs	APIs	APIs	APIs

Atividade 1 - Coleta de Dados do Histórico



Atividade 1 - Introdução

Atividade 1

- manifest
- UI elements
- background script
- content script
- options page
- APIs

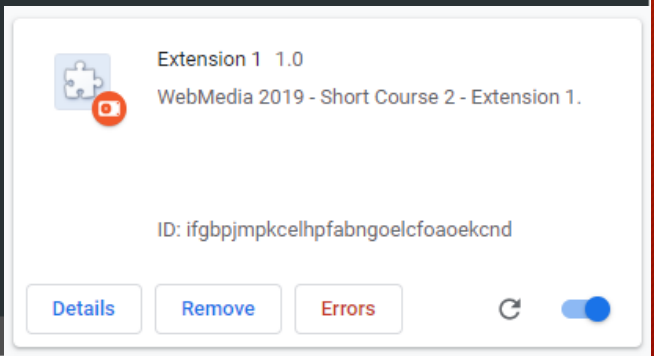
Manifesto

- Especificação da extensão (quais componentes e APIs são usados)

manifest.json

```
{
    "name": "Extension 1 ",
    "version": "1.0",
    "description": "WebMedia 2019 - Short Course 2 - Extension 1.",
    "permissions": ["history"],
    "browser_action": {
        "default_popup": "urls.html",
        "default_icon": "icon1.png"
    },
    "manifest_version": 1
}
```

Ln 12, Col 1, C0 | DOS | UTF-8 | JSON



Atividade 1 - Introdução

Atividade 1

- manifest
- UI elements
- background script
- content script
- options page
- APIs

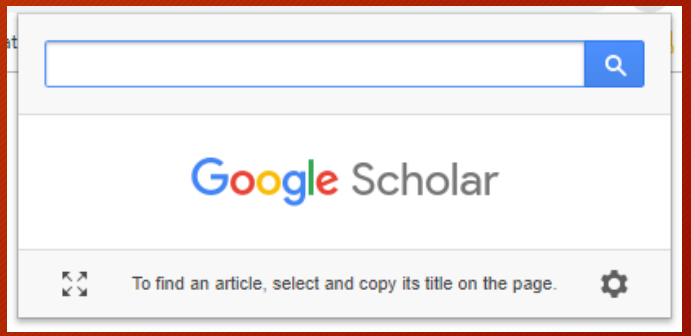
Elementos da interface com o usuário (*UI elements*)

- Interfaces para interação com o usuário
- HTML, CSS, JavaScript

popup.html

```
1  <!doctype html>
2  <html>
3  <head>
4    <meta charset="utf-8">
5    <title>Google Scholar Button</title>
6    <link rel="stylesheet" href="popup.css">
7    <script src="popup-compiled.js" defer></script>
8  </head>
9  <body></body>
10 </html>
11
```

Ln 8, Col 8, C0 | UNIX | UTF-8 | HTML | Mod: 9/29/20



Atividade 1 - Introdução

Atividade 1

- manifest
- UI elements
- background script
- content script
- options page
- APIs

APIs

- Acesso à funcionalidades do navegador (nativas ou não)
- Em geral, necessitam de permissão informada no manifesto
- Acesso via JavaScript
- APIs nativas documentadas em developer.chrome.com/extensions/devguide

Atividade 1

- chrome.history

Coleta de dados do histórico

Atividade 2

- chrome.runtime
- chrome.tabs

Coleta de dados de busca

Atividade 3

- chrome.identity
- chrome.storage

Coleta de dados de identificação

Atividade 4

- Github.js

Salvar dados coletados

Atividade 1 - Preparação

Atividade 1

chrome.history

Coleta de dados do histórico

search

```
chrome.history.search(object query, function callback)
```

Searches the history for the last visit time of each page matching the query.

Parameters

object	query	

string	text	A free-text query to the history service. Leave empty to retrieve all pages.
double	(optional) startTime	Limit results to those visited after this date, represented in milliseconds since the epoch. If not specified, this defaults to 24 hours in the past.
double	(optional) endTime	Limit results to those visited before this date, represented in milliseconds since the epoch.
integer	(optional) maxResults	The maximum number of results to retrieve. Defaults to 100.

function	callback	The *callback* parameter should be a function that looks like this:

```
function(array of HistoryItem results) {...};
```

array of **HistoryItem**	results

```
33
34  chrome.history.search({
35      'text': '',
36      'startTime': timeReference
37    },
38    function(historyItems) {
55    });
```

Atividade 1 - Preparação

Atividade 1

chrome.history

Coleta de dados do histórico

search

```
chrome.history.search(object query, function callback
```

Searches the history for the last visit time of each page matching

Parameters

object	query	

string	text	A free-text all pages.
double	(optional) startTime	Limit resul millisecon 24 hours i
double	(optional) endTime	Limit resul millisecon
integer	(optional) maxResults	The maxin

function	callback	The *callback* parameter should be a function that looks like this: `function(array of HistoryItem results) {...};`

array of HistoryItem	results

HistoryItem

An object encapsulating one result of a history query.

properties

string	id	The unique identifier for the item.
string	(optional) url	The URL navigated to by a user.
string	(optional) title	The title of the page when it was last loaded.
double	(optional) lastVisitTime	When this page was last loaded, represented in milliseconds since the epoch.
integer	(optional) visitCount	The number of times the user has navigated to this page.
integer	(optional) typedCount	The number of times the user has navigated to this page by typing in the address.

Atividade 1 - Preparação

Atividade 1

chrome.history

Coleta de dados do histórico

getVisits

```
chrome.history.getVisits(object details, function callback)
```

Retrieves information about visits to a URL.

Parameters

object	details	string	url	The URL for which to retrieve visit information. It must be in the format as returned from a call to history.search.
function	callback	The *callback* parameter should be a function that looks like this: `function(array of VisitItem results) {...};`		
		array of **VisitItem**	results	

```
49 | chrome.history.getVisits({url: url}, processVisitsWithUrl(url));
```

Ln 42, Col 43, C0 | DOS | UTF-8 | JavaScript | Mod: 10/17/2019 8:18:54 PM | File size: 3504 | R/W

Atividade 1 - Preparação

Atividade 1

chrome.history

Coleta de dados do histórico

getVisits

`chrome.history.getVisits(object details, function ca`

Retrieves information about visits to a URL.

Parameters		
object	details	string url The URL for which as returned from a
function	callback	The *callback* parameter should be a `function(array of VisitItem` array of **VisitItem**

VisitItem

An object encapsulating one visit to a URL.

properties		
string	id	The unique identifier for the item.
string	visitId	The unique identifier for this visit.
double	(optional) visitTime	When this visit occurred, represented in milliseconds since the epoch.
string	referringVisitId	The visit ID of the referrer.
TransitionType	transition	The **transition type** for this visit from its referrer.

```
49 | chrome.history.getVisits({url: url}, processVisitsWithUrl(url));
```

Ln 42, Col 43, C0 | DOS | UTF-8 | JavaScript | Mod: 10/17/2019 8:18:54 PM | File size: 3504 | R/W

Atividade 1 - Hands On

- Siga as instruções da Atividade 1 na apostila
- Tempo estimado: 30 minutos
- Nesta atividade, você vai:

  Criar uma extensão que coleta o histórico de navegação que fica registrado no navegador. O histórico corresponde ao conjunto de endereços que o usuário digitou no omnibox ou o conjunto de endereços acessados por meio de links.

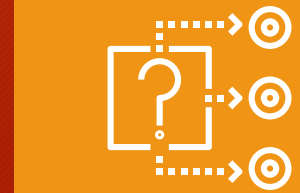
Atividade 1 - Revisão

Nesta atividade, você:

- Carregou uma extensão não empacotada.
- Corrigiu erros na extensão usando os recursos do navegador.
- Empregou a API chome.history para recuperar o histórico de navegação do usuário.
- Configurou a extensão para operar no navegador em modo incógnito.

Como publicar uma extensão na Chrome Web Store (CWS)?

Atividade 2 – Coleta de Dados de Busca



Atividade 2 - Introdução

Atividade 2

- manifest
- UI elements
- content script
- background script
- options page
- APIs

manifest.json

```
{
  "name" : "Extension 2",
  "version" : "1.0",
  "description": "WebMedia 2019 - Short Course 2 - Extension 2.",
  "permissions" : ["tabs"],
  "browser_action" : {
    "default_popup": "popup.html",
    "default_icon" : "icon2.png"
  },
  "content_scripts": [
      {
        "matches": ["https://www.google.com/*"],
        "js": ["content.js"]
      }
    ],
  "background" : {
    "scripts" : ["background.js"],
    "persistent" : false
  },
  "manifest_version" : 2
}
```

Ln 1, Col 1, C0 | DOS | UTF-8 | JSON | Mod: 10/22/2019 7:23:37 PM | File size: 97

Atividade 2 - Introdução

Atividade 2

- manifest
- UI elements
- content script
- background script
- options page
- APIs

Script de Background (*background script*)

- Tratador de eventos relevantes para a aplicação (JavaScript)

background.js

```
chrome.runtime.onInstalled.addListener(function() {
    chrome.storage.sync.set({color: '#3aa757'}, function() {
    });
    chrome.declarativeContent.onPageChanged.removeRules(undefined, function() {
    });
});
```

background.js

```
var googleSearch = "https://www.google.com/search?q=";
var searchTerms = "o efeito do consumo de café";

chrome.runtime.onMessage.addListener(
  function(request, sender, sendResponse) {

```

Atividade 2 - Introdução

Atividade 2

- manifest
- UI elements
- content script
- background script
- options page
- APIs

Script de Conteúdo *(content script)*

- JavaScript que acessa o conteúdo de uma página (DOM)

content.js

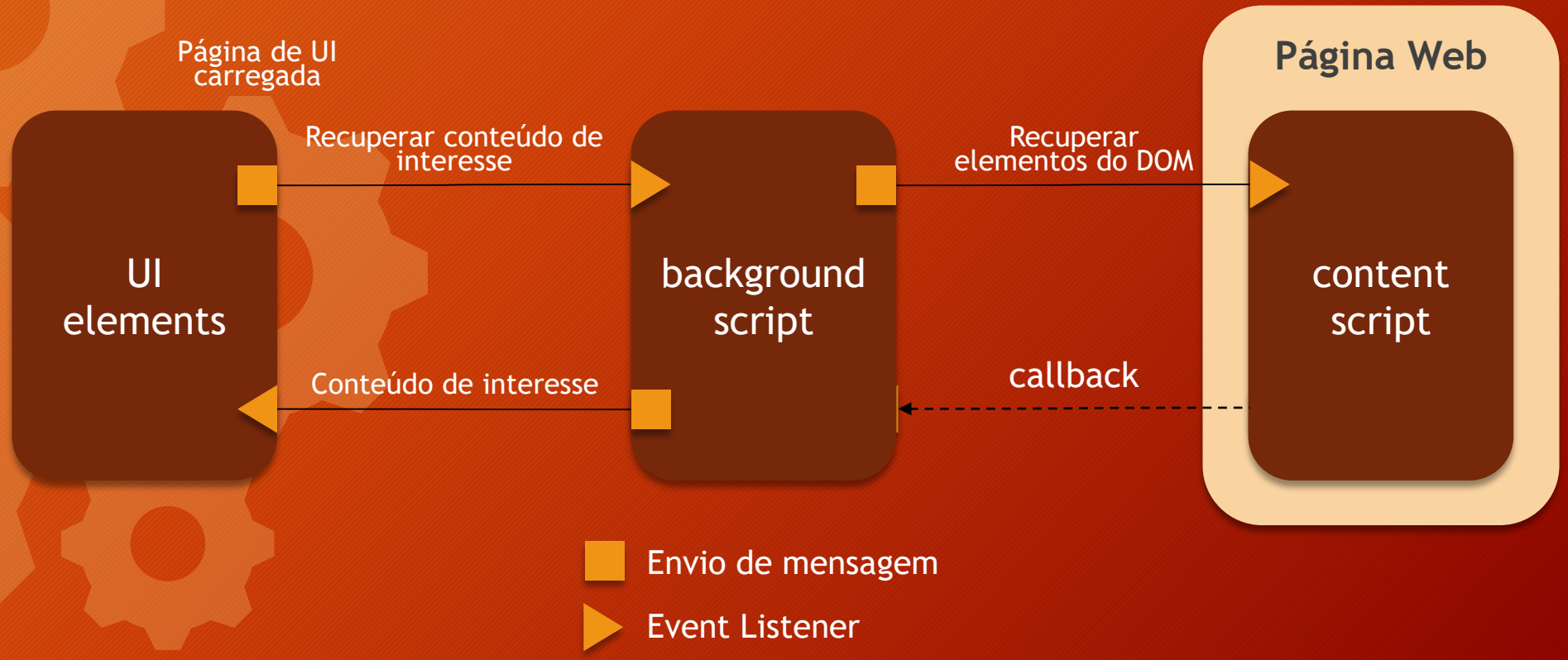
```
1  chrome.runtime.onMessage.addListener(
2    function(request, sender, sendResponse) {
3    // If the received message has the expected format...
4      if (request.subject === 'parsing_search_results') {
5        data = [];
6        var sers = document.evaluate("//div[@class=\"srg\"]//a/h3", document, null
7        for (var i=0; i < sers.snapshotLength; i++)
8        {
9          data[i] = sers.snapshotItem(i).textContent;
10       }
11       sendResponse(
12         {
13           "sers": data
14         }
15       )
16     };
```

Atividade 2 - Introdução

Troca de mensagens entre os componentes da extensão

Atividade 2

- manifest
- UI elements
- content script
- background script
- options page
- APIs



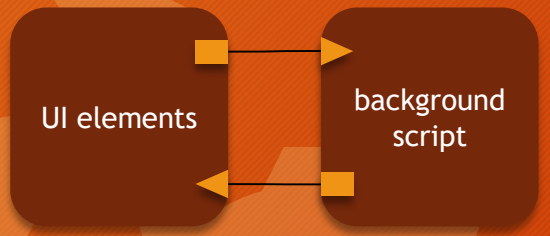
Atividade 2 - Preparação

Troca de mensagens entre os componentes da extensão

Atividade 2

- chrome.runtime
- chrome.tabs

Coleta de dados de busca



chrome.runtime.sendMessage

sendMessage

`chrome.runtime.sendMessage(string extensionId, any message, object options, function responseCallback)`

Parameters

any	message	The message to send. This message should be a JSON-ifiable object.
function	(optional) responseCallback	If you specify the *responseCallback* parameter, it should be a function that looks like this: `function(any response) {...};` any — response — The JSON response object sent by the handler of the message. If an error occurs while connecting to the extension, the callback will be called with no arguments and **runtime.lastError** will be set to the error message.

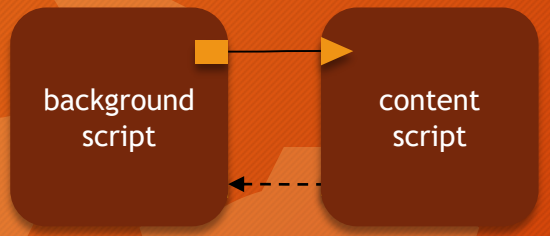
Atividade 2 - Preparação

Troca de mensagens entre os componentes da extensão

Atividade 2

- chrome.runtime
- chrome.tabs

Coleta de dados de busca



chrome.tabs.sendMessage

sendMessage

```
chrome.tabs.sendMessage(integer tabId, any message, object options, function responseCallback)
```

Parameters

integer	tabId	Uma extensão pode criar (e consultar DOMs em) mais de uma guia no navegador.
any	message	The message to send. This message should be a JSON-ifiable object.
function	(optional) responseCallback	If you specify the *responseCallback* parameter, it should be a function that looks like this: `function(any response) {...};` any — response — The JSON response object sent by the handler of the message. If an error occurs while connecting to the specified tab, the callback is called with no arguments and **runtime.lastError** is set to the error message.

Atividade 2 - Preparação

Atividade 2

chrome.runtime

chrome.tabs

Coleta de dados de busca

background script

Criação de nova guia

create

`chrome.tabs.create(object createProperties, function callback)`

Creates a new tab.

Parameters

object	createProperties	string	(optional) url	The URL to initially navigate the tab to. Fully-qualified URLs must include a scheme (i.e., 'http://www.google.com', not 'www.google.com'). Relative URLs are relative to the current page within the extension. Defaults to the New Tab Page.
		boolean	(optional) active	Whether the tab should become the active tab in the window. Does not affect whether the window is focused (see **windows.update**). Defaults to *true*.
function	(optional) callback	If you specify the *callback* parameter, it should be a function that looks like this: `function( Tab tab) {...};`		
		Tab	tab	The created tab.

Atividade 2 - Hands On

- Siga as instruções da Atividade 2 na apostila
- Tempo estimado: 20 minutos
- Nesta atividade, você vai:

  Criar uma extensão que recupera o conteúdo de uma busca pelo Google *web search engine*. A extensão abre uma nova guia no navegador, faz uma busca pela string “o efeito do consumo de café”, parseia os resultados e os apresenta em uma janela popup.

Atividade 2 - Revisão

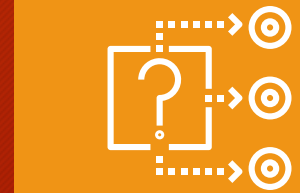
Nesta atividade, você:

- Corrigiu erros na extensão usando os recursos do navegador.
- Empregou a API chrome.tabs para abrir uma nova tab e realizar uma busca usando o Google web search engine.
- Empregou componentes do tipo background e content script.
- Empregou as APIs chrome.runtime e chrome.tabs para trocar mensagens entre componentes da extensão.

Solução da Atividade 2

Coffee Break! Retornamos às 10:30.

Atividade 3 - Coleta de dados do usuário



Atividade 3 - Introdução

Atividade 3

- manifest
- UI elements
- background script
- content script
- options page
- APIs

manifest.json | options.html

```
{
  "name": "Extension 3",
  "version": "1.0",
  "description": "WebMedia 2019 - Short Course 2 - Extension 3.",
  "permissions": ["storage", "identity", "identity.email"],
  "options_page": "options.html",
  "browser_action": {
    "default_popup": "popup.html",
    "default_icon": "icon3.png"
  },
  "manifest_version": 2
}
```

Ln 1, Col 1, C0 | DOS | UTF-8 | JSON | Mod: 10/25/2019 7:06:40 PM | File size: 33

Atividade 3 - Introdução

Atividade 3

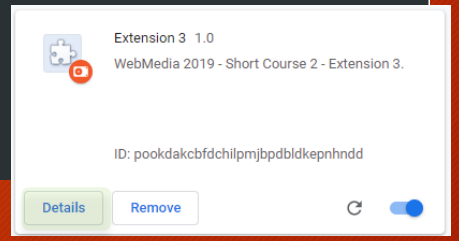
- manifest
- UI elements
- background script
- content script
- options page
- APIs

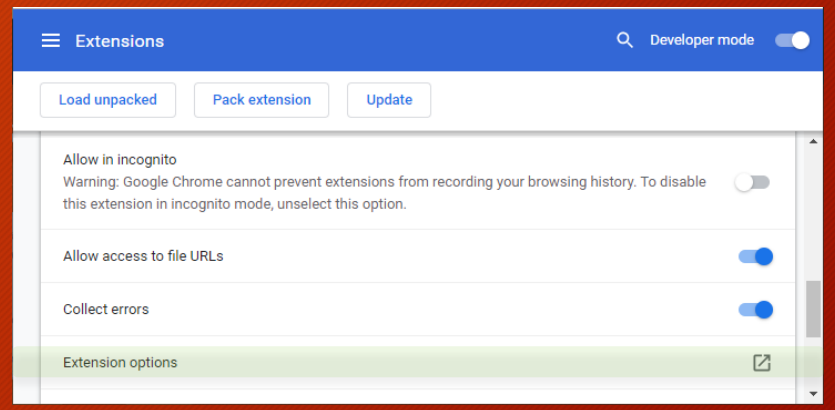
Página de opções (*options page*)

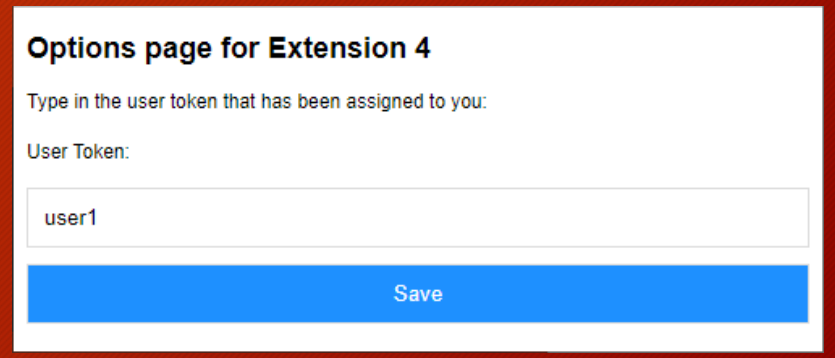
Interface para a configuração de parâmetros da extensão

manifest.json | options.html

```
1  <!DOCTYPE HTML>
2  <html>
3  <head>
50 </head>
51 <body>
52
53 <h2>Options page for Extension 4</h2>
54 <p>Type in the user token that has been assigned to you:</p>
55
56 <form class="form-inline">
57   <label for="email">User Token:</label>
58   <input  id="user_token">
59   <button id="save_button">Save</button>
60 </form>
61
62 </body>
63 <script src="options.js"></script>
64
65 </html>
```







Atividade 3 - Preparação

Atividade 3

- chrome.identity
- chrome.storage

Coleta de dados de identificação

getProfileUserInfo

```
chrome.identity.getProfileUserInfo(function callback)
```

Retrieves email address and obfuscated gaia id of the user signed into a profile.

Parameters		GAIA (Google Accounts and ID Administration) ID
function	callback	The *callback* parameter should be a function that looks like this: `function(object userInfo) {...};`

object	userInfo			
		string	email	An email address for the user account signed into the current profile. Empty if the user is not signed in or the `identity.email` manifest permission is not specified.
		string	id	A unique identifier for the account. This ID will not change for the lifetime of the account. Empty if the user is not signed in or (in M41+) the `identity.email` manifest permission is not specified.

Atividade 3 - Preparação

Atividade 3

- chrome.identity
- chrome.storage

Coleta de dados de identificação

set

```
StorageArea.set(object items, function callback)
```

get

```
StorageArea.get(string or array of string or object keys, function callback)
```

```
chrome.storage.local.set({key: value}, function() {
  console.log('Value is set to ' + value);
});

chrome.storage.local.get(['key'], function(result) {
  console.log('Value currently is ' + result.key);
});
```

Informações confidenciais do usuário não devem ser armazenadas! A área de armazenamento não é criptografada

Atividade 3 - Hands On

- Siga as instruções da Atividade 3 na apostila
- Tempo estimado: 30 minutos
- Nesta atividade, você vai:

  Criar uma extensão que recupera diferentes formas de identificação do usuário participante: se o usuário estiver logado no navegador (signed in), é possível recuperar seu e-mail e seu ID da conta no Google. Alternativamente, durante o esforço de recrutamento, você pode enviar junto com o convite para participar da pesquisa um “token” de participante. Neste método, o usuário registra seu token na extensão como parte das tarefas designadas para ele.

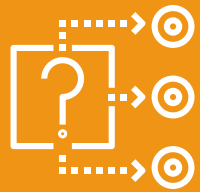
Atividade 3 - Revisão

Nesta atividade, você:

- Empregado a API **chrome.identity** para recuperar dados da identificação do usuário participante.
- Empregado o componente "*options page*" para permitir a configuração da extensão pelo usuário.
- Empregado a API **chrome.storage** para salvar dados na instância local do navegador.

Importante: informe ao comitê de ética de sua instituição sobre a coleta de dados de identificação do usuário, bem como as medidas de proteção dos dados coletados.

Atividade 4 - Salvando os dados coletados



Atividade 4 - Introdução

Atividade 4

- manifest
- UI elements
- background script
- content script
- options page
- APIs

manifest.json

```
{
  "name" : "Extension 4",
  "version" : "1.0",
  "description": "WebMedia 2019 - Short Course 2 - Extension 4.",
  "permissions" : ["tabs", "storage", "identity", "identity.email"],
  "browser_action" : {
    "default_popup": "popup.html",
    "default_icon" : "icon4.png"
  },
  "content_scripts": [
      {
        "matches": ["https://www.google.com/*"],
        "js": ["content.js"]
      }
    ],
  "background" : {
    "scripts" : ["background.js", "github.js"],
    "persistent" : false
  },
  "manifest_version" : 2
}
```

Atividade 4 - Preparação

Atividade 4

Github.js

Salvar dados coletados

```
1  // Creates a new instance of the Github object exposed by Github.js
2  var github = new Github({
3    username: 'YOUR_USERNAME',
4    password: 'YOUR_PASSWORD',
5    auth: 'basic'
6  });
7
8  // Creates an object representing the repository you want to work with
9  var repository = github.getRepo('A_USERNAME', 'A_REPOSITORY_NAME');
10
11 // Creates a new file (or updates it if the file already exists)
12 // with the content provided
13 repository.write(
14    'BRANCH_NAME', // e.g. 'master'
15    'path/to/file', // e.g. 'blog/index.md'
16    'THE_CONTENT', // e.g. 'Hello world, this is my new content'
17    'YOUR_COMMIT_MESSAGE', // e.g. 'Created new index'
18    function(err) {}
19 );
```

GitHub Account

Remote Repository

Branch

File to commit

(trecho extraído de "Upload files on GitHub using GitHub.js")

Atividade 4 - Hands On

- Siga as instruções da Atividade 4 na apostila
- Tempo estimado: 30 minutos
- Nesta atividade, você vai:

  Criar uma extensão que recupera o conteúdo de uma busca pelo Google web search engine e salva os resultados da busca em um repositório do GitHub, na forma de um arquivo JSON.

- Para esta atividade, você pode usar sua conta no GitHub. Caso não tenha uma, use as credenciais abaixo:
  - user “webmedia2019mc2”, password “WebMedia2019”.

Atividade 4 - Revisão

Nesta atividade, você:

- Empregou a API do Github para salvar os dados coletados em um repositório.
- Empregou a API chrome.tabs para abrir uma nova guia e realizar uma busca usando o Google web search engine.
- Empregou as APIs chrome.runtime e chrome.tabs para trocar mensagens entre componentes da extensão.

Importante: as credenciais de acesso ao repositório do GitHub podem, em princípio, ser recuperadas por meio da inspeção do código instalado no filesystem local.

Recap

- Coleta de dados do histórico
- Coleta de dados de busca
- Coleta de dados do usuário
- Envio dos dados coletados para repositório externo

Coleta Online de Dados

Usando recursos do Chrome Extension
WebMedia 2019 – Minicurso 2
andre.p.lima@usp.br

Obrigado pela sua participação!