

دليلك لاحتراق برمجة وتطوير إضافات وملحقات متصفح كروم



محمد مفيد البنا

مطور ويب وتطبيقات أندرويد

محمد مفيد البنا

تخرجت عام ٢٠١٢ من الجامعة الإسلامية بقطاع غزة - فلسطين بتخصص تطوير برمجيات من كلية تكنولوجيا المعلومات.

بدأت عام ٢٠١٤ التوجه لتطوير مُلحقات وإضافات متصفح كروم وفايرفوكس، ولي أربع إضافات حالياً موجودة على سوق كروم، وطوّرت العديد من الإضافات والمُلحقات لشركات ورؤاد أعمال.

أعمل حالياً كمستقل في مجال تطوير الويب وتطوير تطبيقات أندرويد، ومدّون في "مدونة خارج الصندوق".



فهرس المحتويات

١. مقدمة عن إضافات المتصفح ١
- ١.١ ما هي إضافات المتصفح ١
- ١.٢ التقنيات المستخدمة ١
- ٣.١ مفارقات بين إضافات متصفح كروم وفايرفوكس ٢
٢. إضافات متصفح كروم Chrome Extensiones ٣
- ٢.١ أين تعمل إضافات متصفح كروم ٣
- ٢.٢ أنواع وأشكال إضافات متصفح كروم ٣
- ٢.٣ الوصول للإضافات المُسطبة على متصفح كروم ٤
- ٢.٤ تركيبة الملفات في إضافات كروم ٥
- ٢.٥ تجربة الإضافات المطورة على المتصفح قبل رفعها على السوق ٨
٣. الواجهات البرمجية الخاصة بمتصفح كروم وكيفية التعامل معها ٩
- ٣.١ ما هي الواجهات البرمجية API ٩
- ٣.٢ الصلاحيات وطرق التعامل معها ٩
- ٣.٣ التعامل مع ذاكرة المتصفح Local Storage و Synchronized Storage ١١
- ٣.٤ الفرق بين Background Script و Content Script ١٤
- ٣.٥ تبادل الرسائل داخل الإضافة Message Passing ١٨
- ٣.٦ الشارات Badges ٢٠
- ٣.٧ تنبيهات سطح المكتب ٢١
٤. نشر المُلحق على سوق كروم ٢٣
- ٤.١ ما هو سوق متصفح كروم Chrome Web Store ٢٣
- ٤.٢ الصيغة النهائية للمُلحقات ٢٤
- ٤.٣ الرفع على سوق كروم أو على استضافتك الخاصة ٢٧

٢٨	٤.٤ من داخل صفحة المطورين
٣٠	٤.٥ رفع إضافة/ملحق جديد إلى السوق من صفحة المطورين
٣٥	٤.٦ مسببات لرفض إضافتك ومُلاحقاتك من السوق
٤٢	٥. الكسب المادي من الإضافات
٤٤	٥.١ شراء المُلحق لمرة واحدة، أو الاشتراك الشهري/السنوي
٤٧	٥.٢ البيع من داخل الإضافة In-App Payments
٤٩	٥.٣ الإعلانات وإعلانات Google AdSense
٤٩	٥.٣ وسائل أخرى للربح؟
٥٠	الخاتمة
٥٠	الحقوق

١. مقدمة عن إضافات المتصفح

١.١ ما هي إضافات المتصفح

إضافات المتصفح هي عبارة عن برمجيات صغيرة نقوم بتحميلها وتنصيبها على المتصفحات (كروم، فايرفوكس، سفاري... وغيرها) تقوم بأداء وظيفة محددة وواضحة قد لا يوفرها المتصفح ولا تكن مدمجة أثناء تحميلنا للمتصفح من المصدر، ومن الممكن كذلك تعديل وتحسين وظائف موجودة مسبقاً بالمتصفح.

وحتى لا يقوم المطورون بالتعديل على المتصفح ذاته لإضافة وظائف أخرى معينة (كإدارة المواقع المفضلة مثلاً، أو إضافة لمنع الإعلانات من المواقع الإلكترونية)، يوفر صانعو المتصفحات ما يُسمى بالإضافات أو المُلحقات يتم تطويرها على أيدي مطورين آخرين تسمح للمستخدمين بتحميلها على متصفحاتهم لأداء وظائف معينة.

١.٢ التقنيات المستخدمة

يعتمد بناء إضافات أو ملحقات المتصفح على تقنيات الويب الأساسية وهي:

- **HTML**: لغة ترميز وتعني مجموعة الأكواد الوصفية المستخدمة لإنشاء وتصميم صفحات ومواقع الويب وهي اختصار لـ **Hyper Text Markup Language** ولقراءة المزيد حول هذه اللغة من ويكيبيديا.
 - **CSS**: لغة تهتم بتنسيق صفحات الويب وتصميم المواقع لتظهر بصور جميلة، كالألوان والصور والخلفيات. فـ **HTML** يركز على المحتوى الخاص بالصفحة، والـ **CSS** مهتمة بالتنسيق وترتيب هذا المحتوى الذي يظهر لزائر صفحة الويب، ولقراءة المزيد حول هذه اللغة من ويكيبيديا.
 - **JavaScript**: وهي لغة برمجة مخصصة لبث الحياة على صفحات الويب، فهي تجعل صفحة الويب أكثر تفاعلية مع المستخدم. ماذا تريد أن يحصل عندما يقوم المستخدم بالضغط على زرٍ ما؟ أو ماذا تريد أن يحصل عندما تتأكد من تحميل صفحة ويب كاملة مع الصور وغيرها عند المستخدم؟ كل هذه الأحداث تستطيع توليها باستخدام لغة **JavaScript** وهنا المزيد حول هذه اللغة من ويكيبيديا.
- بالإضافة للتقنيات أعلاه، فعلى المطور أن يملك خلفية ولو كانت صغيرة في التعامل مع هياكل البيانات ك:

- **JSON**: وهي اختصار لـ **JavaScript Object Notation** وهي صيغة تمثيل بيانات يسهل نقلها وتخزينها.
- **XML**: وهي اختصار لـ **Extensible Markup Language** وهي لغة ترميز مخصصة لوصف البيانات ونقلها وتخزينها كذلك. مثال لاستخدام **JSON** وما يوازيه في **XML** من ويكيبيديا

٣.١ مفارقات بين إضافات متصفح كروم وفايرفوكس

هذا الكتاب الإلكتروني يتحدث بشكل من التركيز على متصفح كروم وإضافاته، وهنا جدول المفارقات بين متصفح كروم وفايرفوكس من واقع العمل على تطوير إضافات لكل المتصفحين:

المزايا	كروم	فايرفوكس
سهولة التوثيق	سهل وبسيط	سهل ممتنع
سهولة التطوير	بسيط وغير معقد	به نوع من التعقيد وتحتاج للإلمام بالعديد من الأمور البرمجية الأخرى
مجانية العضوية في السوق	غير مجانية (تُكلف ٥ دولارات لمرة واحدة)	مجانية
الرفع خارج السوق	لا يسمح	يسمح (بشيء من التعقيد)
أسلوب مراجعة الإضافات	آلي في البداية ما لم تُثير الإضافة الشك. تُحوّل للمراجعة اليدوية إذا كانت هناك مشكلة ما بالإضافة ويُرسَل لك بريد إلكتروني عن المشكلة، ولن تُقبل الإضافة بالسوق إلا بعد مراجعتها يدوياً وخلّوها من المشاكل	يدوي دائماً
سرعة قبول الإضافات في السوق	بمعدل ساعة أو ساعتين على الأكثر بعد نشرك للإضافة ما لم تُحوّل للمراجعة اليدوية	من ثلاث أيام إلى عشرة أيام حسب النمط الذي قمت باختياره للفحص، لأن الفحص يدوي دائماً

٢. إضافات متصفح كروم Chrome Extensiones

٢.١ أين تعمل إضافات متصفح كروم

من المنطقي أن تعمل إضافات وملحقات متصفح كروم بوجود متصفح كروم، حيث يتوافر متصفح كروم على العديد من أنظمة التشغيل ك ويندوز، ماك، ولينكس. هذا يعني أن ملحقاتك ستعمل على أكثر من نظام تشغيل بفضل وجود متصفح كروم ودعمه على أكثر من نظام تشغيل.

لكن للأسف لا تعمل الإضافات بوجود متصفح كروم على الأجهزة الذكية التي تعمل بنظام Android أو iOS، فالمتصفح لا يدعم عمل ملحقاتك حتى بوجود المتصفح.

٢.٢ أنواع وأشكال إضافات متصفح كروم

يوجد ثلاث أنواع لإضافات متصفح كروم، يمكنك اختيار إحداها بناءً على وظيفة الإضافة ذاتها:

١- **Browser Action**: وهو النوع الذي يظهر في شريط الأدوات الخاص بمتصفح كروم بجانب شريط العناوين كما هو موضح بالصورة ٢.٢.١ وتسمح بوضع أيقونة ليتفاعل معها المستخدم بالضغط عليها لتظهر نافذة منبثقة مثلاً.

يستخدم هذا النوع بالعادة من الإضافات التي تقدم وظائف عامة وغير مقتصرة للعمل بوجود صفحة ويب برابط مخصص، كإضافة منع الإعلانات التي تعمل على كافة صفحات الويب وليس فقط صفحات فيسبوك على سبيل المثال لا الحصر.



صورة رقم ٢.٢.١ مثال عن إضافات من نوع Browser Action

٢- **Page Action**: وهو النوع الذي يظهر في شريط العناوين كما هو موضح بالصورة ٢.٢.٢ وتسمح كذلك بوضع أيقونة من اختيارك وتسمح كذلك بتفاعل المستخدم مع الأيقونة لتظهر له نافذة منبثقة في حال قام بالضغط عليها.



صورة رقم ٢.٢.٢ مثال عن إضافات من نوع Page Action

يُستخدم هذا النوع من الإضافات التي تقدم وظائف مخصصة للصفحة المفتوحة حالياً وليس على كل الصفحات (الألسنة) المفتوحة حالياً على متصفح كروم بعكس الإضافات من نوع Browser Action.

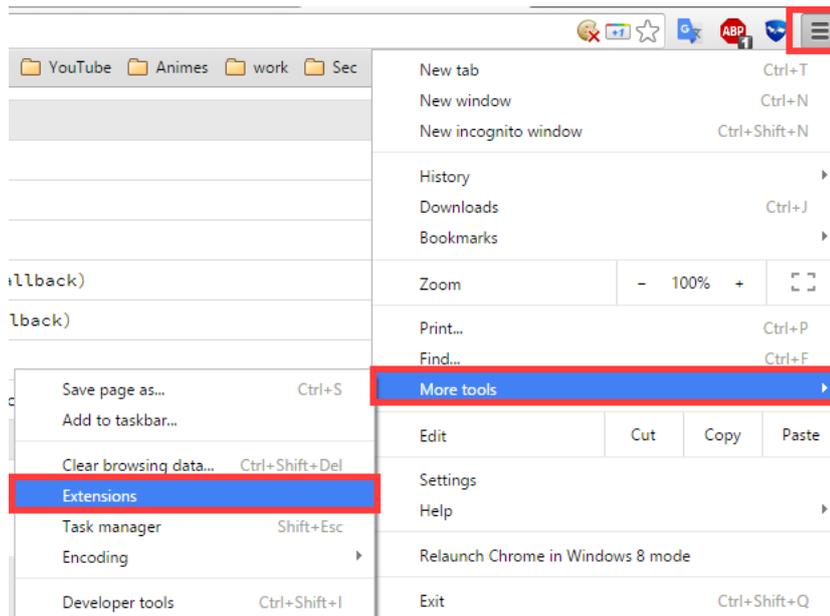
أمثلة لهذا النوع من الإضافات، الإضافات التي تُظهر أسماء المكتبات المُستخدمة في الموقع المفتوح حالياً، أو إضافة لإظهار نوع الخط المُستخدم في الفقرة المحددة في هذه الصفحة الحالية وهكذا.

٣- لا هذا ولا ذلك: في حال كنت تريد أن تعمل الإضافة في الخلفية دون حاجة لتفاعل المستخدم، فلا داعي لتحديد أي نوع من الأنواع السابق ذكرها.

٢.٣ الوصول للإضافات المُسببة على متصفح كروم

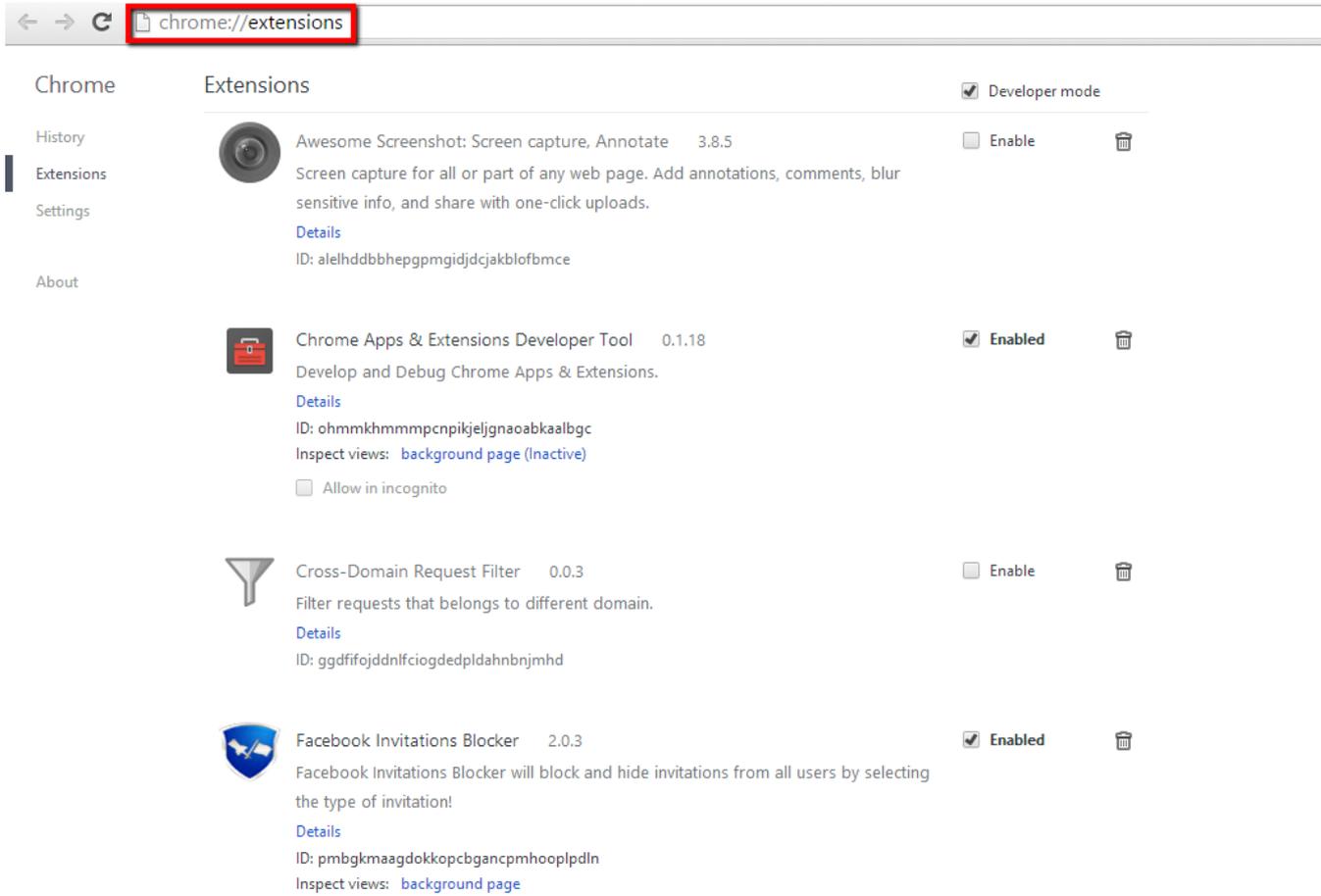
رؤيتك للإضافات الموجودة في شريط الأدوات أو في شريط العناوين لا يعني أن هذه هي كل الإضافات المُسببة لديك! فكما أشرت للنوع الأخير وهو النوع الذي يعمل في الخلفية ولا حاجة ليتفاعل المستخدم مع الإضافة! لذلك ولرؤية كافة الإضافات المُسببة في متصفح كروم من خلال:

- ١- الضغط على أيقونة (الإعدادات)، ثم الذهاب لـ (المزيد من الأدوات - More Tools)، ومن ثم (الإضافات - Extensions) كما تشير الصورة رقم ٢.٣.١:



صورة رقم ٢.٣.١ للذهاب إلى صفحة الإضافات المُسببة

٢- من خلال كتابة في شريط العناوين هذا الرابط Chrome://Extensions سينقلك مباشرة لصفحة الإضافات المُسطبة لديك كما هو موضح في الصورة رقم ٢.٣.٢:



صورة رقم ٢.٣.٢ لعرض كافة الإضافات المُسطبة في متصفح كروم

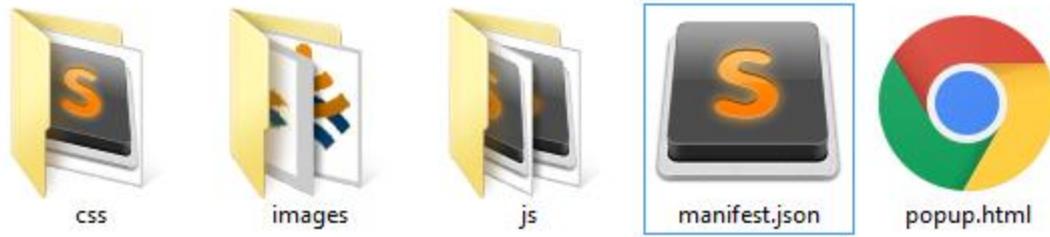
هذه هي صفحة الإضافات المُسطبة لدي، يمكنك تعطيل أو تفعيل الإضافة التي تريد من خلال وضع علامة صح على Enabled أو تفعيل بجانب كل إضافة أو إزالة علامة الصح من على Enabled.

٢.٤ تركيبة الملفات في إضافات كروم

كما أشرنا سابقاً أن التقنيات المُستخدمة في تطوير إضافات المُتصفح هي تقنيات الويب، أي عندما تقوم بتصميم صفحة ويب فإنك تقوم بتقسيم الملفات بالشكل الذي تريد. فكافة ملفات CSS تقوم بوضعها في مجلد Style مثلاً، وكافة ملفات JavaScript تقوم بوضعها في مجلد بعنوان JS وهكذا.

على مستوى متصفح كروم، وحتى يفهم المتصفح أن هذه هي إضافة أو ملحق وليست مجرد صفحة ويب عادية، فيجب عليك أن تُعرّف ذلك في ملف أساسي بإسم Manifest.json وهو عبارة عن ملف بهيكلية بيانات من نوع JSON لتمثيل صيغة بيانات الإضافة، كإسم الإضافة، أيقونتها، وصف الإضافة، نوع الإضافة وغيرها من المعلومات الأخرى التي يقرأها متصفح كروم ويظهرها للمُستخدم في صفحة الإضافات التي تعلمنا كيفية الوصول إليها.

إذًا، الملف الأساسي لدينا هو Manifest.json وهذه هي صورة تشرح توزيع الملفات والمجلدات:



صورة رقم ٢.٤.١ توضح طبيعة الملفات وأهمها لأي إضافة

الملفات والمُجلدات:

- مجلد CSS: يحتوي على كافة ملفات من نوع CSS لتنسيق محتوى وصفحات الإضافة التي تعرض للمستخدم (كصفحة popup.html وتنسيقاتها).
- مجلد Images: يحتوي على كافة الصور من أيقونات وغيرها من الصور.
- مجلد JS: يحتوي على كافة ملفات وأكواد جافا سكريبت.
- صفحة ويب Popup.html: وهي الصفحة التي ستحمل محتوى النافذة المنبثقة عند ضغط المستخدم لأيقونة الإضافة (نستنتج من هذا الكلام أن نوع الإضافة هو Browser Action).
- وملفنا الأساسي هو Manifest.json: ويحتوي على كافة تفاصيل الإضافة من إسم، وصف، الأيقونات ومساراتها وغيرها، وصورة رقم ٢.٤.٢ ستوضح طبيعة محتوى واحدة من الإضافات لملف Manifest.json:

```
{ 1 {
2   "manifest_version": 2,
3   "name": "My First Extension",
4   "version": "1.0.0",
5   "description": "Description",
6   "icons": {
7     "16": "images/16.png",
8     "48": "images/48.png",
9     "128": "images/128.png"
10  },
11  "author": "www.mbanna.info",
12  "browser_action": {
13    "default_icon": "images/48.png",
14    "default_popup": "popup.html"
15  }
16 }
```

صورة رقم ٢.٤.٢ توضح أهم القيم التي يجب أن تتوفر في ملف Manifest.json الأساسي

شرح بعض المصطلحات:

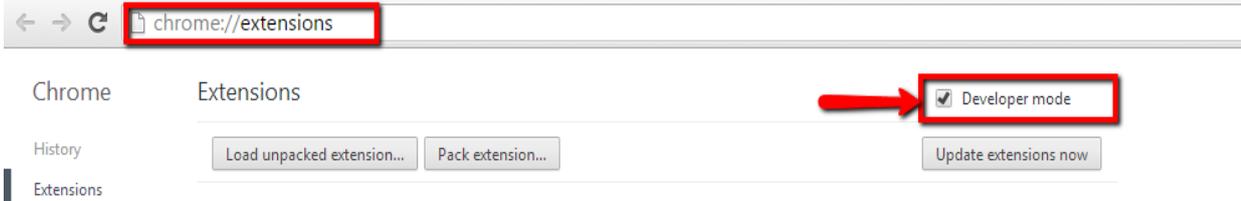
- **manifest_version**: أتركها دائماً كما هي بإصدارها الثاني.
 - **Name**: يحتوي على اسم الإضافة وهو الاسم الذي سيظهر للمستخدم عند عرضه لكافة الإضافات المُسببة في المتصفح أو في صفحة الإضافة في السوق الخاص بمتصفح كروم للإضافات.
 - **Version**: وهو رقم نسخة الإضافة، وسيساعد هذا الفيديو بعنوان "كيف تقوم بتصدير برمجياتك بطريقة صحيحة" أو Software Versioning لتعطي دلالة للمستخدم عن طبيعة هذه النسخة.
 - **Description**: يحتوي على وصف الإضافة وهو ذاته الذي سيظهر للمستخدم في صفحة الإضافات المُسببة أو في صفحة الإضافة في السوق الخاص بمتصفح كروم للإضافات.
 - **Icons**: يحتوي على مسار وأسماء الأيقونات بأحجامها المختلفة ويفضل أن يحتوي على ثلاث جودات للأيقونات.
 - **Author**: قيمة غير ضرورية ويمكن الاستغناء عنها، يمكن الإشارة لموقع مطور هذه الإضافة.
 - **browser_action**: وهو لإعلام المتصفح بنوع الإضافة أنها من نوع Browser Action، أي ستظهر أيقونة الإضافة في شريط الأدوات بجانب شريط العناوين، قمنا بتحديد مسار وإسم ملف الأيقونة الذي سيظهر بجانب شريط العناوين، وكذلك حدّدنا مسار الصفحة التي يجب أن تُعرض عند ضغط المُستخدم على أيقونة إضافتك.
- يوجد العديد من القيم الأخرى الإضافية التي يمكن وضعها في ملف Manifest.json، يمكن الإستعانة بهذا المصدر الرسمي من جوجل لإيجاد المزيد، ومن أهم الأمور التي يجب الإطلاع عليها هي قيمة **Permissions** حيث تتطلب بعض الإضافات صلاحيات معينة للعمل بشكل صحيح، كصلاحية الوصول للموقع الجغرافي الحالي للمستخدم Physical Location أو الـ GeoLocation على سبيل المثال! وليس بالضرورة أن تحتوي كل إضافة على صلاحيات! لكن سيتم عرض الصلاحيات المطلوبة للمستخدم عند تحميله لإضافة ما من السوق.

٢.٥ تجربة الإضافات المطورة على المتصفح قبل رفعها على السوق

بالطبع، وقبل رفع الإضافة على السوق، يجب أن تقوم بتجربتها لديك على المتصفح والقيام بالتعديلات اللازمة والتأكد من كافة الوظائف، وحتى تقوم بتجربة الإضافة على المتصفح يجب القيام بهذه الخطوات:

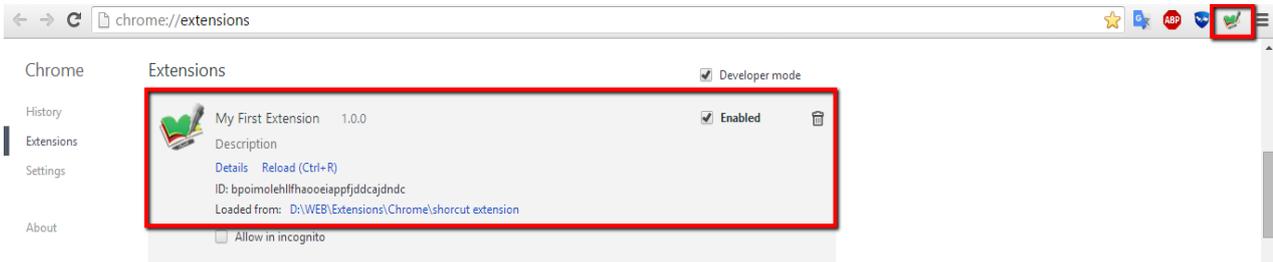
١- قم بالذهاب لصفحة الإضافة المُسطبة بأيٍ من الطرق التي تم ذكرها آنفاً.

٢- تأكد من تفعيل وضع المطورين كما هو موضح بالصورة رقم ٢.٥.١:



صورة رقم ٢.٥.١ توضح طريقة تفعيل وضع المطورين

٣- يمكنك بعدها سحب مجلد الإضافة (بما يحتويه من مجلدات وملفات وأهمها ملف Manifest.json) وإفلاته في هذه الصفحة، أو ببساطة الضغط على Load Unpacked Extension وتحديد مسار مجلد الإضافة (هذه الإضافة ستكون مرفقة مع الكتاب الإلكتروني للتجربة).



صورة رقم ٢.٥.٢ توضح استيراد مجلد الإضافة وتركيبها في متصفح كروم

٤- أي تعديل تقوم به في ملفات الإضافة، كإضافة صور وأيقونات جديدة، أو تحديث في ملفات جافاسكريبت أو الستايل، يجب عليك أن تقوم بتحديث الإضافة من خلال الضغط على Reload في الصورة رقم ٢.٥.٢ وترى تعديلاتك فعلياً على المتصفح.

٣. الواجهات البرمجية الخاصة بمتصفح كروم وكيفية التعامل معها

٣.١ ما هي الواجهات البرمجية API

عندما نقوم ببناء خدمة برمجية معينة، ونريد من مستخدمين آخرين الاستفادة من هذه الخدمة، فقد يتبادر في ذهنك لأول مرة بأن يجب علينا مشاركتهم الكود البرمجي الخاص بهذه الخدمة للاستفادة من مزاياها! هذا بالطبع خاطيء تماماً، ومن هنا ظهر ما يُسمى بالواجهات البرمجية أو API وهي اختصار لـ Application Programming Interface، من خلال صنع واجهة برمجية معينة يمكن للمستخدم التعامل معها للاستفادة من الخدمات دون أي حاجة أو ضرورة لرؤية الكود الخاص بالخدمة.

يشرح هذا الفيديو المتواضع بأسلوب كرتوني ومبسط عن مفهوم الواجهات البرمجية، ومثلها بأنك عندما تذهب إلى مطعم ما، فأنت لا تتعامل مباشرة مع الطباخ، بل تسأل النادل وتطلب منه ما تشاء من طلبات ويقوم هو بإيصالها للمطبخ لتجهيز طلبياتك ليعود لك النادل بما طلبت عند جهوزية الطلب! فالنادل هنا لعب دور الوسيط أو ما يُسمى بالواجهة التي تعاملت أنت معها دون الخوض بالتفاصيل فيما يحصل داخل المطعم ومطبخه.

عودة إلى متصفح كروم، وحتى يتسنى لنا التعامل مع الخدمات التي يقدمها المتصفح، كاستخدام ذاكرة المتصفح أو عرض المواقع المفضلة للمستخدم أو مراقبة طلبات الصفحات المفتوحة حالياً، التعامل مع الكوكيز. إذاً هذا يتطلب منا معرفة الكود الداخلي الخاص بالمتصفح؟ بالطبع لا، جوجل توفر لنا واجهة برمجية بسيطة لاستدعائها واستخدامها للوصول لموارد المتصفح دون رؤية كود المتصفح من الداخل.

٣.٢ الصلاحيات وطرق التعامل معها

تطرقنا سابقاً في ملف Manifest.json حول نقطة الصلاحيات، وأنه يجب تعريف الصلاحيات التي نريد وليس بالضرورة كل إضافة يجب أن تملك صلاحيات! لكن ما علاقة هذا بالواجهات البرمجية الخاصة بمتصفح كروم؟

لنفترض مثلاً بأنني أريد الحصول على الموقع الجغرافي الخاص بمستخدم إضافتي حالياً، إذا بالطبع أنا بحاجة للتعامل مع الواجهة البرمجية الخاصة بجوجل، وبالطبع لفعل ذلك يجب أن أضيف صلاحية الوصول للموقع الجغرافي GeoLocation عبر ملف Manifest.json ليخرج لنا بهذا الشكل:

```
{ 1 {
2     "manifest_version": 2,
3     "name": "My First Extension",
4     "version": "1.0.0",
5     "description": "Description",
6     "icons": {
7         "16": "images/16.png",
8         "48": "images/48.png",
9         "128": "images/128.png"
10    },
11    "author": "http://www.mbanna.info",
12    "browser_action": {
13        "default_icon": "images/48.png",
14        "default_popup": "popup.html"
15    },
16    "permissions": ["geolocation"]
17 }
```

صورة رقم ٣.٢.١ توضح إضافة صلاحية GeoLocation في ملف الـ Manifest.json

تم إضافة كلمة Permissions في ملف Manifest.json وكما نرى أنها تحتوي على مصفوفة، أي يمكن أن تحتوي الإضافة على مجموعة من الصلاحيات الأخرى التي نحتاجها.

الآن وعبر كود جافا سكريبت صغير يمكننا الحصول على الموقع الجغرافي لمستخدم الإضافة:

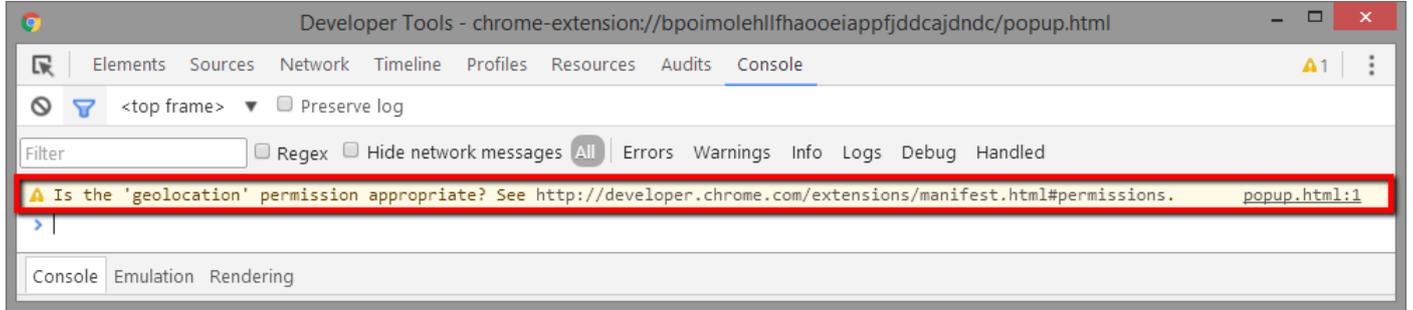
```
1 function getLocation() {
2     if (navigator.geolocation) {
3         navigator.geolocation.getCurrentPosition(function(position){
4             console.log("The Latitude is: " +position.coords.latitude);
5             console.log("The Longitude is: " +position.coords.longitude);
6         });
7     }
8 }
```

صورة رقم ٣.٢.٢ توضح دالة الحصول على الموقع الجغرافي للمستخدم عند استدعاء الدالة

سنلاحظ عبر هذا الكود الصغير بعرض نقاط الطول والعرض لمستخدم الإضافة، وطبعهم في شاشة الكونسول في متصفح كروم.

ماذا لو لا أملك صلاحية الوصول لـ **GeoLocation** معرفاً في ملف **Manifest.json**:

بكل بساطة لن يتم تنفيذ هذه التعليمات وتظهر رسالة خطأ في شاشة الكونسول بأن الإضافة لا تملك صلاحيات للوصول الـ **GeoLocation**:



صورة رقم ٣.٢.٣ توضح رسالة التحذير بأن صلاحية **GeoLocation** غير متوفرة ويجب إضافتها

٣.٣ التعامل مع ذاكرة المتصفح **LocalStorage** و **Synchronized Storage**

إن كنت لازلت تتابع قراءة الكتاب ولا يوجد بجانبك كوباً من القهوة، فعليك أن تُجهز نفسك لقليلاً من الصداع وكثيراً من الكود يا صديقي!

ولازلنا في جزئية التعامل مع الواجهة البرمجية الخاصة بمتصفح كروم للوصول لأهم الخدمات والمميزات في متصفح كروم، والآن وبالطبع أي إضافة لابد لها من أن تقوم بتخزين بعض البيانات حتى تعود لها لاحقاً، كإعدادات الإضافة على سبيل المثال؟

قبل كل شيء، يجب علينا إضافة صلاحية الدخول للذاكرة في ملف **Manifest.json** كما أشرنا سابقاً:

```
{ 1 {
2   "manifest_version": 2,
3   "name": "My First Extension",
4   "version": "1.0.0",
5   "description": "Description",
6   "icons": {
7     "16": "images/16.png",
8     "48": "images/48.png",
9     "128": "images/128.png"
10  },
11  "author": "http://www.mbanna.info",
12  "browser_action": {
13    "default_icon": "images/48.png",
14    "default_popup": "popup.html"
15  },
16  "permissions": ["geolocation", "storage"]
17 }
```

صورة رقم ٣.٣.١ توضح إضافة صلاحية **Storage** للتعامل مع ذاكرة المتصفح

الآن، بإمكاننا الاستفادة من الواجهة البرمجية والتعامل مع الذاكرة بكل سهولة، توفر لنا الواجهة البرمجية API أسلوبين جميلين في تخزين البيانات عبر المتصفح:

- Local Storage

تمكننا من حفظ البيانات والاستعلام عنها في المتصفح حتى بعد إغلاقه، وهنا كود بسيط لتخزين معلومات بسيطة عبر استخدام الواجهة البرمجية الخاصة بجوجل:

```
15 function saveData(){
16     chrome.storage.local.set({'value': 'First Value To Save'}, function() {
17         //What do you want to do after that
18         console.log("Data Saved!");
19     });
20 }
```

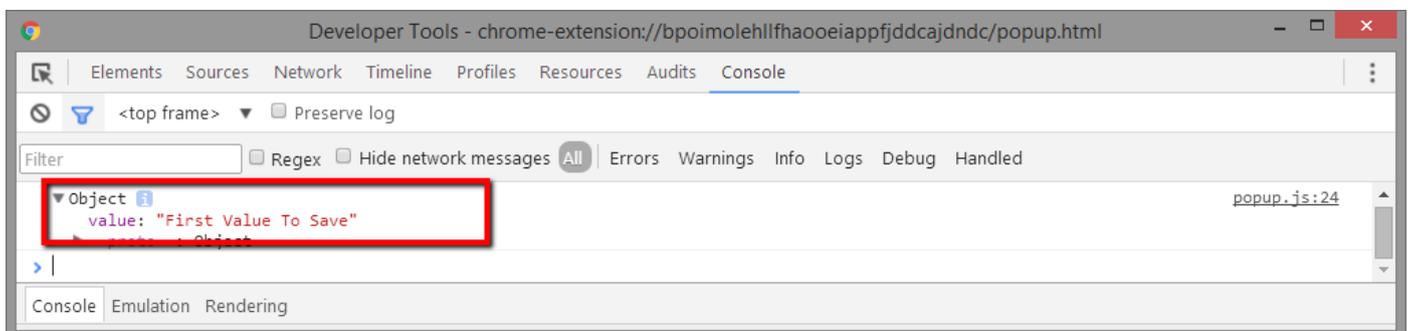
صورة رقم ٣.٣.٢ عبارة عن دالة لتخزين كائن أو قيمة معينة في ذاكرة المتصفح

بالإستعانة بـ `chrome.storage.local` استطعنا الوصول إلى الذاكرة، وتخزين كائن Object، وإظهار رسالة بعد عملية الحفظ، أردت هنا تخزين رسالة "First Value To Save" في مفتاح "value"، حتى أستطيع الوصول للرسالة من خلال هذا المفتاح. الآن، دعنا نستعلم عن ما قمنا بحفظه:

```
22 function loadData(){
23     chrome.storage.local.get('value', function(data) {
24         console.log(data);
25     });
26 }
```

صورة رقم ٣.٣.٣ عبارة عن دالة للاستعلام عن الكائن الذي قمنا بتخزينه

عبر هذه الدالة، سنستعلم عن البيانات التي قمنا بتخزينها مسبقاً في المفتاح Value وعرضها على شاشة الكونسول، وهذه هي النتيجة:



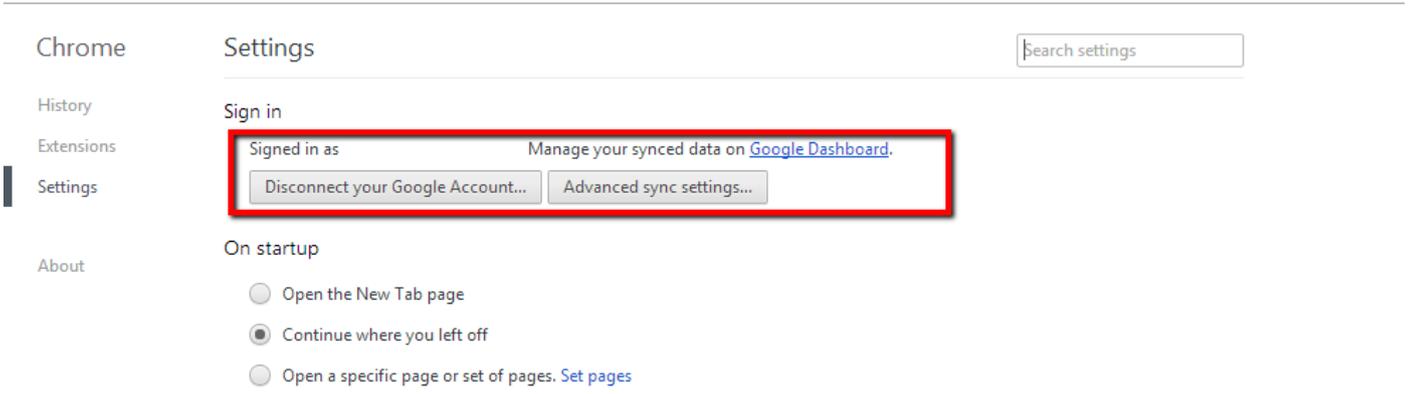
صورة رقم ٣.٣.٤ توضح القيمة التي استعلمنا عنها وعرضها في شاشة الكونسول

- Synchronized Storage

نفس التي سبقتها بالإستعمال تماماً، فقط علينا تبديل `chrome.storage.local` بـ `chrome.storage.sync`

لكن، ما الفرق الجوهرى بينهما؟ نعلم جميعاً أن جوجل توقّر لنا ربط المتصفح بحسابنا عبر البريد الإلكتروني لتخزين المواقع المفضلة، وإعدادات المتصفح وحتى الإضافات المسطبة...حتى نستطيع من جهاز حاسوب آخر مزامنة مواقعنا المفضلة وإعدادات المتصفح والإضافات وغيرها من الأشياء التي يمكن إدارتها دون الحاجة للقيام بذلك يدوياً كل مرة نقوم بتحميل متصفح كروم!

فلو قمت بإعادة تهيئة لنظامك وتمت إزالة متصفح كروم من حاسوبك، لا تياس إذا قمت سابقاً بربط متصفحك عبر بريدك الإلكتروني لعمل مزامنة لكل ذلك! فكل شيء بالحفظ والصون كما هو مُشار بالصورة التالية:



صورة رقم ٣.٣.٥ من شاشة الإعدادات الخاصة بمتصفح كروم توضح ربط المتصفح بالبريد الإلكتروني لعمل مزامنة للإعدادات

الآن، عودة الى واجهتنا البرمجية `chrome.storage.sync` وما قمت بتخزينه هنا، سيتم كذلك عمل مزامنة له بحساب المستخدم المربوط ببيده الإلكتروني. بالعادة، يجب عليك أن تقوم بتخزين إعدادات إضافتك من خلال `chrome.storage.sync` وليس `chrome.storage.local` حتى يتم مزامنة إعدادات إضافتك في حال اضطر المُستخدم لإعادة تسطيب متصفح كروم وإضافتك مرة أخرى!

لكن احذرا! جوجل تقول بأن المطور يستطيع مزامنة أي شيء فعلاً، لكن هناك حجم محدد لما يمكنك مزامنته وحفظه، فالحجم الكلي المسموح لك به هو ١٠٢,٤٠٠ بايت فقط، وحجم كل كائن يمكنك تخزينه بمفتاحه هو ٨,١٩٢ بايت فقط، وأقصى عدد من الكائنات يمكنك تخزينها هو ٥١٢ فقط، وإلا، سيحصل خطأ أثناء حفظك للبيانات.

٣.٤ الفرق بين Content Script و Background Script

أشرنا في قسم أنواع إضافات متصفح كروم أن هناك ثلاث أنواع، وكان النوع الأخير هو "لا هذا ولا ذاك" وهو نوع يعمل فقط في الخلفية! أي بمجرد أن يقوم المُستخدم بتشغيل متصفح كروم وتبدأ الإضافات بالعمل، تريد أن تقوم إضافتك بعمل شيء ما بالخلفية حتى وإن لم يقم المُستخدم بأي حركة أو فتح أي صفحة! إذاً نحن بحاجة لـ Script يتم تنفيذه في Background فور تشغيل المتصفح وتبدأ الإضافة الخاصة بنا بالعمل.

حتى نقوم بذلك، نقوم بتجهيز السكريبت وليكن background.js وهو يحتوي على الكود الذي أريد تنفيذه ليعمل في الخلفية، لكن يتبقى أن أعطي التعليمات للمتصفح بأن هذا الملف يجب تنفيذه فور تشغيل إضافتي...وهو ما يجب أن أقوم بتحديدده في ملف Manifest.js كما توضح الصورة رقم ٣.٤.١:

```
{ 1 {
2   "manifest_version": 2,
3   "name": "My First Extension",
4   "version": "1.0.0",
5   "description": "Description",
6   "icons": {
7     "16": "images/16.png",
8     "48": "images/48.png",
9     "128": "images/128.png"
10  },
11  "author": "http://www.mbanna.info",
12  "browser_action": {
13    "default_icon": "images/48.png",
14    "default_popup": "popup.html"
15  },
16  "permissions": ["geolocation", "storage"],
17  "background": {
18    "scripts": ["js/background.js"],
19    "persistent": true
20  }
21 }
```

صورة رقم ٣.٤.١ توضح تعريف لسكريبت أريد تنفيذه ليعمل في الخلفية

قمنا بإضافة background وحددنا مسار السكريبت الذي نريد تنفيذه ليعمل في الخلفية وهو داخل مجلد js واسم الملف هو background.js ولكن، ما هو الـ persistent ولماذا يحمل قيمة true؟

لنفترض أن السكريبت الذي يعمل في الخلفية يقوم بفحص ما كل عشر دقائق، وبالتالي تنفيذ السكريبت لن ينتهي عند مرحلة معينة وإنما سيبقى يعمل بالخلفية دائماً وطول الوقت! وهو ما أشرنا إليه هنا من خلال قيمة `persistent` وتحمل `true` وتعني بأن السكريبت لا يُنفذ فقط لمرة واحدة بل سيبقى يعمل في الخلفية، لذلك أبقى يا متصفح كروم عينك على ما يعمل في الخلفية وتابع النتائج ولا تتكاسل عن ذلك!

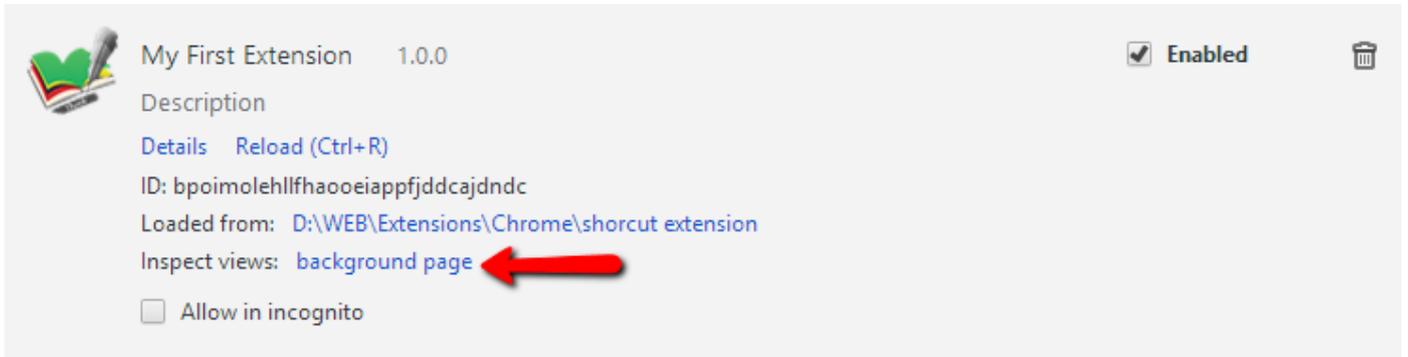
ماذا لو لم أضع هذه القيمة؟ أو قمت بتعريفها بـ `false` مثلاً؟ سيتم تنفيذ الكود لمرة واحدة فقط، ولن ترى النتائج بعد عشر دقائق لأن متصفح كروم ألقى نظرة واحدة وقام بالتنفيذ وأنهى بعد ذلك كل شيء له علاقة بالسكريبت!

قمت بإنشاء سكريبت `background.js` وهذا ما بداخله:

```
1 console.log("Hello! I'm background.js and I'm just working in background now!");
```

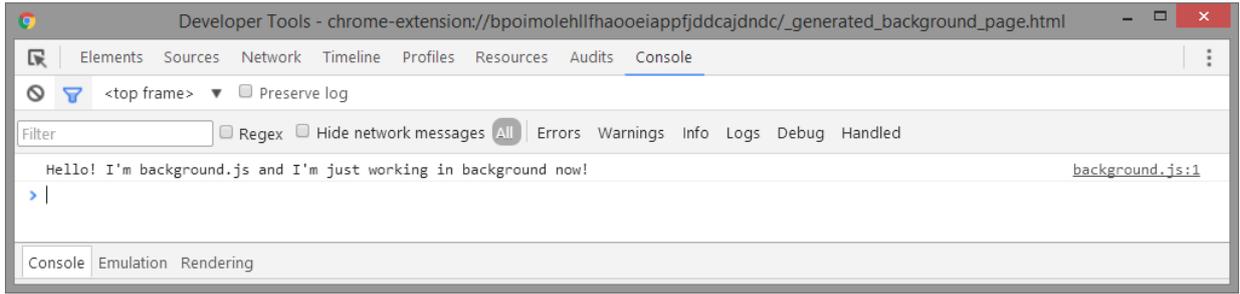
صورة رقم ٣.٤.٢ توضح الرسالة التي أريد إظهارها في الخلفية

مجرد رسالة تظهر فور تشغيل المتصفح يتم طباعتها في شاشة الكونسول، لكن يبقى السؤال... كيف أتابع مخرجات السكريبت الذي يعمل في الخلفية؟ كهذه الرسالة مثلاً؟ بمجرد زهابنا لصفحة الإضافات المُسطبة، سنرى شيئاً جديداً أُضيف في خانة إضافتنا:



صورة رقم ٣.٤.٣ توضح الوصول لنتائج السكريبت الذي يعمل بالخلفية

هذا يعني أن هناك سكريبت يُنفذ بالخلفية! وبمجرد الضغط على **background page**، سنعرض شاشة الكونسول الخاصة بالسكريبت الذي تم تنفيذه في الخلفية، وهذه هي المخرجات:



صورة رقم ٣.٤.٤ توضح نتائج السكريبت الذي يعمل في الخلفية

إذا كانت إضافتك تعمل في مواقع معينة، **Facebook Mention Tagging Blocker** إضافة المخصصة لمنع الإشارة إليك في منشور ما أو صورة ما على فيسبوك، فهذا النوع من الإضافات تضطر لأن تحقن كود معين داخل صفحات فيسبوك فقط! وهو ما يُسمى بـ Content Script أي الكود الذي قمنا بحقنه في صفحة فيسبوك أصبح جزءاً منها.

سأقوم الآن بإنشاء سكريبت خاص في مجلد js يحمل اسم `content_script.js` وأريد من هذا السكريبت أن يتم حقنه في صفحة فيسبوك، ما الذي أحتمل الآن حتى يعرف متصفح كروم بذلك؟ بالطبع! ملف `Manifest.json` عليه أن يحمل بعض التعليمات كما فعلنا مع `Background Scripts`:

```
{ 1 {
2   "manifest_version": 2,
3   "name": "My First Extension",
4   "version": "1.0.0",
5   "description": "Description",
6   "icons": {
7     "16": "images/16.png",
8     "48": "images/48.png",
9     "128": "images/128.png"
10  },
11  "author": "http://www.mbanna.info",
12  "browser_action": {
13    "default_icon": "images/48.png",
14    "default_popup": "popup.html"
15  },
16  "permissions": ["geolocation", "storage"],
17  "background": {
18    "scripts": ["js/background.js"],
19    "persistent": true
20  },
21  "content_scripts": [{
22    "matches": ["*://www.facebook.com/*"],
23    "js": ["js/content_script.js"],
24    "run_at": "document_start"
25  }
26 }
```

صورة رقم ٣.٤.٥ توضح آلية إضافة كود يتم حقنه في صفحات معينة

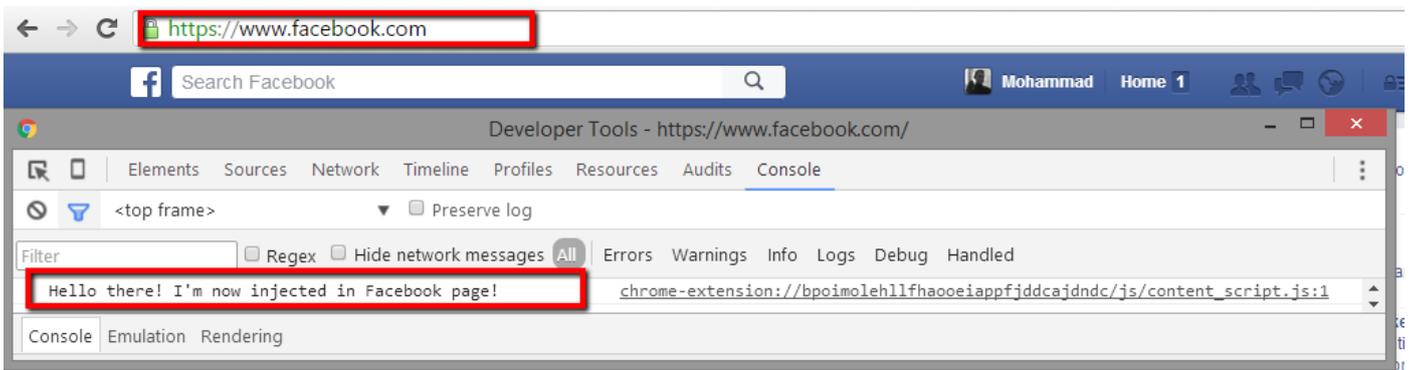
لا داعي للإرتباك يا صديقي! دعنا نوضح ما قمنا به هنا:

- قمنا بتعريف أن هناك Content Scripts سيتم حقنهم، لكن أين يجب أن يقوم كروم بحقنهم؟ قلنا أنه يجب أن يتم الحقن داخل صفحات فيسبوك فقط! ووضحنا ذلك داخل مصفوفة matches وهي من الممكن أن تحتوي على أكثر من موقع مثلاً أو صفحات معينة في فيسبوك.
 - حدّدنا في قيمة js والتي تحمل كذلك مصفوفة، أن السكريبت الذي أريد حقنه داخل صفحات فيسبوك هو content_script.js الموجود داخل مجلد js ومن الممكن كذلك أن أحقن أكثر من سكريبت في آنٍ واحد بلا أي مشكلة!
 - أما القيمة الأخيرة run_at فهي توضح متى أريد أن يتم حق السكريبت؟ بعد تحميل الصفحة كلياً أو فور تحميلها؟ تم تحديد فور تحميلها عبر قيمة document_start، ومن الممكن استبدالها ب document_end لو أردنا تنفيذ السكريبت فور الانتهاء من تحميل الصفحة.
 - هل يمكن حقن أكواد من CSS؟ بالطبع! من الممكن أن تصمم نافذة منبثقة داخل صفحة فيسبوك، ويمكن حقن أكواد CSS بنفس طريقة حقن أكواد js مع استبدالها ب CSS وتحديد مسار ملفات CSS.
- الآن، ماذا يحتوي الملف الذي قمنا بإنشائه في content_script.js؟

```
1 console.log("Hello there! I'm now injected in Facebook page!");
```

صورة رقم ٣.٤.٦ توضح محتوى content_script.js وما أريد إظهاره في شاشة الكونسول

رسالة صغيرة يتم طبعها في شاشة الكونسول توضح بأن الكود فعلاً تم حقنه بنجاح! الآن، دعنا نفتح صفحة فيسبوك ونرى هل فعلاً تم حقن الكود وتنفيذه داخل صفحة فيسبوك أم لا؟



صورة رقم ٣.٤.٧ توضح ناتج السكريبت الذي قمنا بحقنه داخل صفحة فيسبوك

رائع جداً! فور فتح شاشة الكونسول عبر الضغط على زر F12 وأنا داخل صفحة فيسبوك، شاشة الكونسول تحمل الرسالة التي قمنا بحققها عبر إضافتنا الصغيرة.

هل يجب علينا الإكتفاء فقط بـ Content Script دون الحاجة للتعامل مع Background Scripts حيث أنني لست بحاجة لأي شيء يُنفذ في الخلفية؟ هذا سؤال جيّد! هناك بعض القيود في Content Script حيث أنني لا أستطيع الوصول لكافة مميزات الواجهة البرمجية التي توفرها لي جوجل وأنا داخل Content Script، بعكس Background Scripts التي تسمح لي بالوصول للواجهة البرمجية بكافة تفاصيلها! إذاً أنا بحاجة لنوع من التواصل بين Content Script و Background Script حتى أستطيع الوصول للواجهة البرمجية بشكل الكامل!

٣.٥ تبادل الرسائل داخل الإضافة Message Passing

إن أنهيت فنجان قهوتك، سأقترح عليك بصب المزيد، لأن هذه النقطة تحديداً ذات أهمية كبيرة وستحتاجها بالتأكيد يوماً ما!

عندما نقرأ Message Passing أو تبادل الرسائل، هذا يعني أن هناك شيئان منفصلان يريدان التحدث أو التواصل فيما بينهم، أو إرسال رسالة من جانب إلى آخر. وكما تحدثنا عن الفرق بين Content Script و Background Script وأنا بحاجة لنوع من التواصل بينهما، إذاً هذا القسم سيُجيب بالتحديد عن كيفية القيام بهذا التواصل.

دعنا نقوم بإرسال رسالة صغيرة من سكريبت content_script.js إلى سكريبت background.js:

```
4 //Send message to background.js
5 chrome.runtime.sendMessage({greeting: "I'm the content Script!"}, function(response) {
6     //Show the response
7     console.log(response.farewell);
8 });
```

صورة رقم ٣.٥.١ من سكريبت content_script.js لإرسال رسالة إلى background.js

حتى نقوم بإرسال رسالة من أي سكريبت من داخل Content Script من خلال واجهة برمجية متصفح كروم، نستخدم chrome.runtime.sendMessage وتحتوي على الكائن (الرسالة) التي نريد إرسالها إلى المستقبل، وفي حالتنا تلك سيكون المستقبل هو background.js، ومنتظر الرد الذي سيتم إرساله من background.js في حال فعلاً أردنا الرد على هذه الرسالة.

الآن، كيف نجعل background.js يستمع لأي رسالة نرسلها له؟ ستوضح هذه الصورة رقم ٣.٥.٢ ذلك:

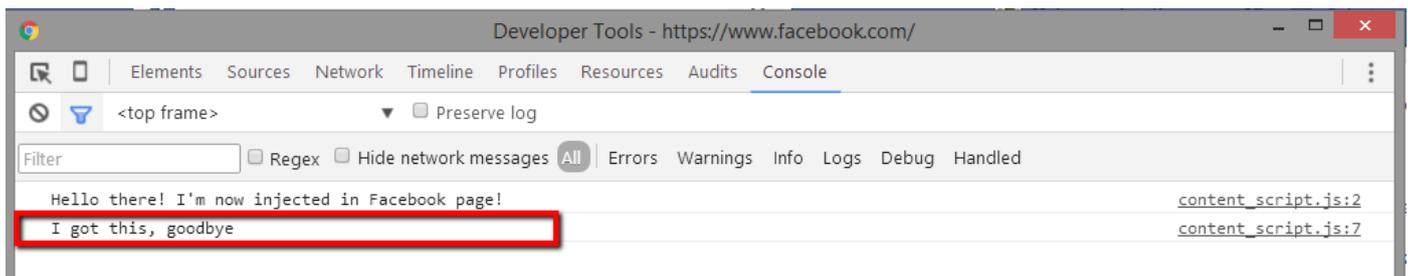
```
//Waiting any messages from content script
chrome.runtime.onMessage.addListener(function(request, sender, sendResponse) {
  if (request.greeting){
    sendResponse({farewell: "I got this, |goodbye"});
  }
});
```

صورة رقم ٣.٥.٢ من سكريبت background.js يوضح استماعه للرسائل المُرسلة من content_script.js والرد عليها

باستخدام الواجهة البرمجية بالطبع، وعبر chrome.runtime.onMessage.addListener سأضيف خاصية الاستماع لأي رسالة قادمة من Content Script وأنا داخل سكريبت background.js.

قمت بالفحص إذا كان الكائن المُرسَل يحمل اسم greeting وإذا كان كذلك أرسل له الرد " I got this, goodbye"! دعنا نرى الآن النتائج:

نفتح صفحة فيسبوك ونتأكد من تنفيذ السكريبت content_script.js، والذي بدوره سيرسل رسالة الى background.js ليُرد علينا بعدها السكريبت بالرسالة: "I got this, goodbye"



صورة رقم ٣.٥.٣ يوضح لنا الرد من background.js ونحن داخل content_script.js

وبالضغط على F1٢ ونحن في صفحة فيسبوك لعرض شاشة الكونسول، سنرى الرد الذي أتانا من background.js بنجاح! بهذه الطريقة استطعنا أن نرسل رسائل من Content Script الى Background Script واستقبل Content Script الرد بنجاح.

النوع الذي تطرقنا له قبل قليل يُسمى بـ Simple One-Time Requests أي نرسل رسالة وننتظر الرد وهذا كل شيء! [هذا المصدر من جوجل](#) يشرح أنواع أخرى لتبادل الرسائل، ماذا لو احتجنا لقناة مفتوحة بين Content Script و Background Script لتبادل أكثر من رسالة بنفس الوقت؟ أو أخذ إجراء معين عند رد معين؟ هذا النوع يُسمى بـ Long Live Connection أو الاتصال المفتوح.

يمكن كذلك تبادل رسائل بين أكثر من إضافة! بل ويمكنك أيضاً إرسال رسالة من موقعك الشخصي إلى إضافتك بكل سهولة. كل هذا يعتمد على طبيعة ووظيفة الإضافة التي تريد تطويرها.

٣.٦ الشارات Badges

يُصبح وثرديد أن تظهر رقم بجانب إضافتك التي هي من نوع Browser Action، أو تظهر نص ما أسفل أيقونة الإضافة كما تقوم إضافة حجب الإعلانات التي تخبرنا كم إعلان قامت بحذفه.

الواجهة البرمجية الخاصة بمتصفح كروم تسمح لنا بذلك، بإمكاننا وضح نص مكون من ٤ أحرف على الأكثر وتحديد اللون الذي نريد، دعنا نظهر رقم "١٢٣" أسفل أيقونة الإضافة كما في صورة رقم ٣.٦.١:



صورة رقم ٣.٦.١ تُظهر عرض أرقام كشارة أسفل أيقونة الإضافة

بالطبع للقيام بذلك، من غير المنطقي أن ندع Content Script يقوم بهذا العمل بل لا نستطيع أساساً الولوج لهذه المنطقة من واجهة جوجل البرمجية وأنا داخل Content Script، يمكننا الإستعانة بـ Background Script لتغيير القيمة بالخلفية والتحكم بالشارات كما نشاء كما هو موضح في الصورة:

```
11 //Badge text background color
12 chrome.browserAction.setBadgeText({ text: "123" });
13 chrome.browserAction.setBadgeBackgroundColor({color: "#E60000"});
```

صورة رقم ٣.٦.٢ من ملف background.js لعرض شارة أسفل أيقونة إضافتنا

باستخدام chrome.browserAction هذا يعني نحن قادرين الآن الوصول لخصائص الإضافة من نوع Browser Action، وحددنا النص الذي ستحملة الشارة Badge ولون الخلفية بكل سهولة! أي تغيير نريده أن يتم للشارة كتغيير للرقم أو النص وأنا داخل Content Script، فنحن بحاجة للتواصل الذي تطرقنا له بالنقطة السابقة بين Content Script و Background Script من خلال Message Passing.

٣.٧ تنبيهات سطح المكتب

قد تحتاج إضافتك بشكل أو بآخر لتعطي للمستخدم تنبيه معين، أو تُظهر له رسالة معينة بعد القيام بوظيفة معينة داخل إضافتك. جوجل توفّر لنا من خلال الواجهة البرمجية كذلك ما يُسمى بالتنبيهات أو Notification لتُظهر رسالة معينة للمستخدم.

في هذا القسم، سنتعلم كيف تُظهر هذا الصندوق الصغير المتواضع للمستخدم من خلال الإضافة:



صورة رقم ٣.٧.١ مثال عن إظهار تنبيه أو إشعار للمستخدم

لا شيء معقد في هذا الإشعار أو التنبيه، سيظهر هذا الصندوق الصغير أسفل الشاشة حتى لو كانت نافذة جوجل كروم مُصغرة، لذلك تُسمى بتنبيهات سطح المكتب! وكما نرى من التنبيه فإنه يحتوي على ٣ عناصر أساسية: أيقونة، عنوان للرسالة، ومحتوى للرسالة. هناك العديد من الأنواع الأخرى يمكننا القيام بها عن طريق الواجهة البرمجية التي توفرها لنا جوجل عبر هذا المصدر.

حتى يتسنى لنا استخدام الإشعارات أو التنبيهات، يجب أن نحصل على الصلاحيات اللازمة في ملف Manifest.json:

```
{ 1 {
2   "manifest_version": 2,
3   "name": "My First Extension",
4   "version": "1.0.0",
5   "description": "Description",
6   "icons": {
7     "16": "images/16.png",
8     "48": "images/48.png",
9     "128": "images/128.png"
10  },
11  "author": "http://www.mbanna.info",
12  "browser_action": {
13    "default_icon": "images/48.png",
14    "default_popup": "popup.html"
15  },
16  "permissions": ["geolocation", "storage", "notifications"],
17  "background": {
18    "scripts": ["js/background.js"],
19    "persistent": true
20  },
21  "content_scripts": [{
22    "matches": ["*://www.facebook.com/*"],
23    "js": ["js/content_script.js"],
24    "run_at": "document_start"
25  }]
26 }
```

صورة رقم ٣.٧.٢ توضح إضافة صلاحية الإشعارات في ملف Manifest.json

لا نستطيع الوصول لقسم الإشعارات أو التنبيهات من خلال الواجهة البرمجية ونحن داخل Content Script، يمكننا فقط استدعاء الإشعار أو التنبيه من خلال Background Script أو أي مكان آخر غير Content Script، وإذا أردنا من خلال Content Script يجب علينا استخدام التراسل أو Message Passing بينهما كما أشرنا بالسابق.

وهذا جزء من الكود الموضوع في ملف background.js وهو الذي سيتم تنفيذه في الخلفية، ويحتوي على:

```
16 //Message details object
17 var messageInfo = {
18     type:"basic",
19     title:"My First Title",
20     message:"Out Of Pal Box Blog Message",
21     imageUrl:"../images/128.png"
22 }
23 //Show notification after 2 seconds
24 setTimeout(function(){
25     chrome.notifications.create(messageInfo);
26 }, 2000);
```

صورة رقم ٣.٧.٣ من ملف background.js لإنشاء إشعار أو تنبيه ليظهر بعد ثانيتين من تشغيل المتصفح مثلاً

قمنا بإنشاء كائن Object يحمل مواصفات ومعلومات الإشعار، كنوع الإشعار (النوع الأساسي هو Basic، وهناك أنواع أخرى من هذا المصدر)، عنوان الرسالة، محتوى الرسالة، وأخيراً مسار الأيقونة.

الآن معلومات الإشعار أو التنبيه جاهزة، يتبقى لنا فقط إنشاء التنبيه واستدعائه عبر chrome.notifications.create وتمرير الكائن الذي يحمل بيانات التنبيه، ولكن، أردت هنا إظهار التنبيه بعد مرور ثانيتين.

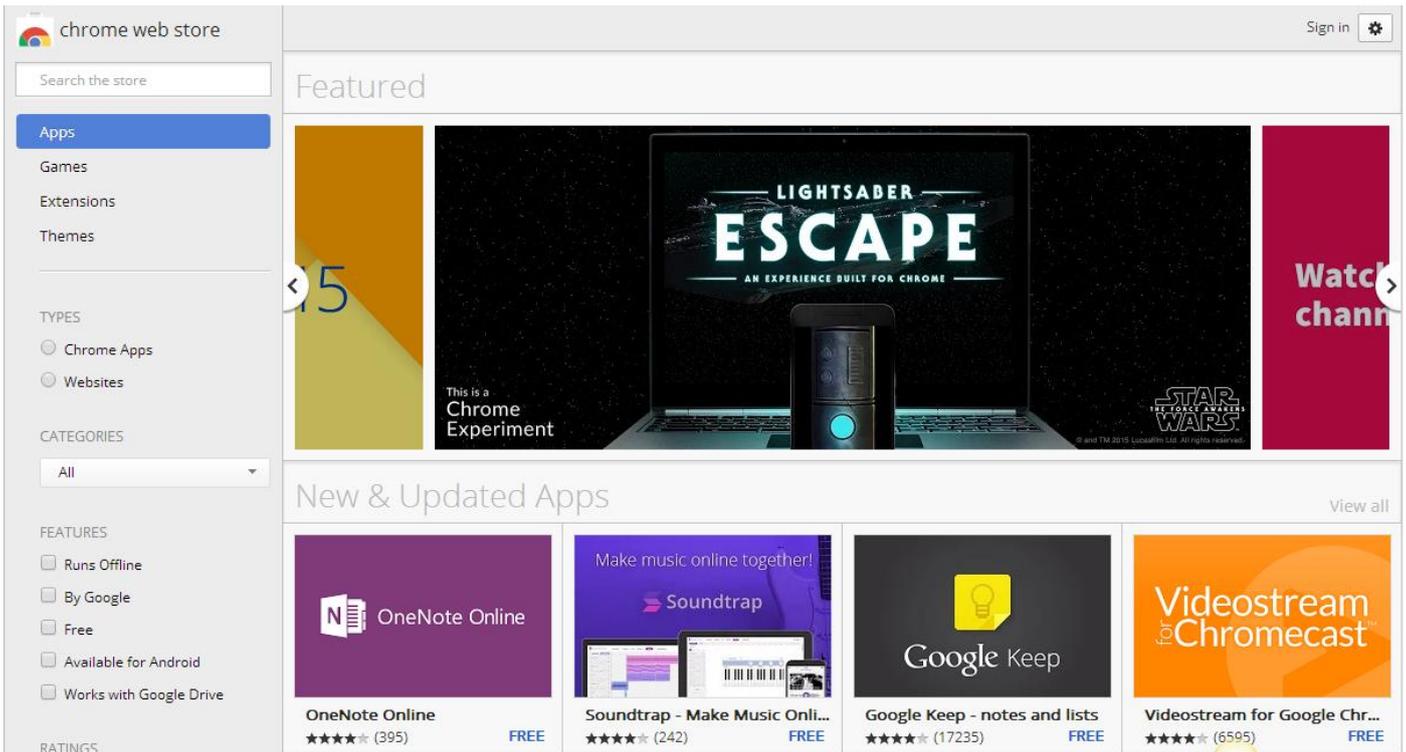
الآن، أنت قادر على إرسال تنبيهات للمستخدم، لكن لا تُكثر منها، فالمستخدمين يكرهون لفت الإنتباه أكثر من اللازم! يمكنك الآن إنشاء إضافة لقراءة إشعارات فيسبوك أو تويتر مثلاً وإظهارها للمستخدم...أطربونا بأفكاركم!

٤. نشر المُلحق على سوق كروم

بعد عمل متواصل وسهر الليالي في تطوير إضافتك، سيحين الوقت لتبدأ بنشر أعمالك على السوق حتى يستطيع الناس من حول العالم تحميلها والاستفادة منها والإعجاب بها وجني الأرباح، إن أمكن!

٤.١ ما هو سوق متصفح كروم Chrome Web Store

هو سوق إلكتروني تم افتتاحه في شهر ديسمبر من عام ٢٠١٠ ليصبح السوق الرسمي لتحميل تطبيقات الويب وملحقات متصفح جوجل كروم بما يشمله من إضافات أو ثيمات أو حتى ألعاب.



صورة رقم ٤.١.١ من سوق كروم

يستضيف السوق الملايين من الملحقات المجانية والمدفوعة، ويوجد ما لذ وطاب من الخدمات التي يمكن أن تجدها متوفرة كملحق يعمل بوجود المتصفح كمنتجات جوجل ذاتها، ليوفر سهولة الوصول للمستخدم لبعض خدمات الشركات.

يمكن للمستخدم تحميل ملحقاتك من السوق، تقييمها، ووضع تعليقات ما، وحديثاً أصبح يمكن مراسلة المُستخدم لمطور الإضافة لإخباره عن مشكلة ما في المُلحق.



صورة رقم ٤.٢.٢ لواحدة من الإضافات الموجودة على السوق

٤.٢ الصيغة النهائية للملحقات

قبل الرفع على السوق والذي سنتطرق له بالتفصيل في النقطة التالية، يجب أن تقوم بتجميع ملفات ومجلدات المُلحق الخاص بك على شكل ملف واحد مضغوط من نوع ZIP ومن ثم رفعه على السوق. لكن، هل هذه هي الصيغة النهائية للإضافة؟

الصيغة النهائية للإضافة هي **CRX** وهو ما يتعرف عليه متصفح كروم أن هذا عبارة عن مُلحق تابع له، يمكننا إنشاء هذه الصيغة النهائية ولكن بمجرد رفع الملف المضغوط الذي بهيئة ZIP على السوق سيقوم وبشكل تلقائي تحويله الى CRX ليكون جاهز للتحميل من قبل المُستخدمين دون أن يتدخل المطور بعملية التحويل تلك.

لماذا إذاً عليّ أن أعرف هذه المعلومة؟

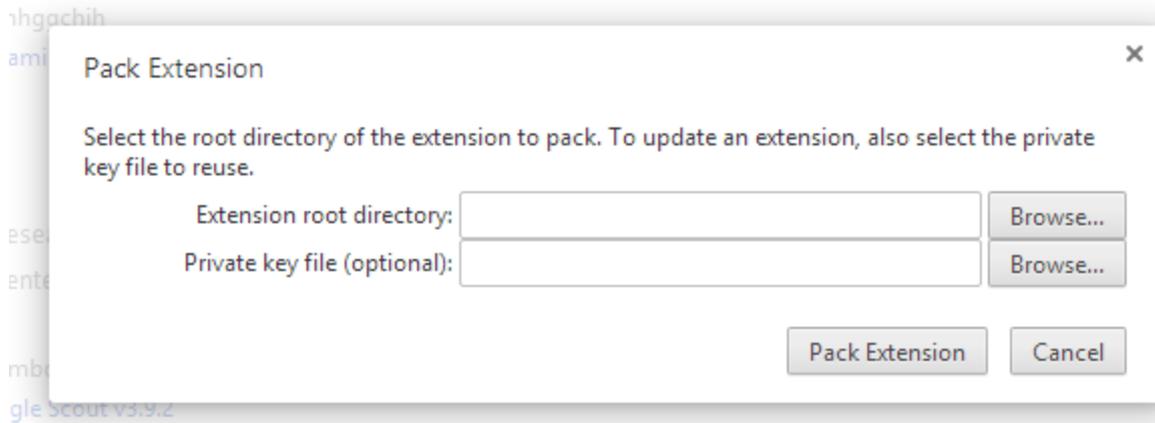
عودة إلى صفحة إضافات المُسطبة على الجهاز، وسبق وأن شرحنا كيف نقوم بتركيب إضافتنا أثناء التطوير على متصفح كروم، وكان هناك طريقتان لذلك: إما بسحب مجلد الإضافة الذي يحتوي على ملف Manifest.json وباقي الملفات إلى صفحة الإضافات، أو من خلال الضغط على Load Unpacked Extension وتحديد مسار مجلد الإضافة.

لكن ماذا لو أردنا إخراج الملحق أو الإضافة بالصيغة النهائية CRX لإعطائها لصديق أو لعميل ما لتجربتها مثلاً قبل رفعها للسوق؟ دعنا نذهب لصفحة الإضافات المُسطبة ونختار Pack Extension:



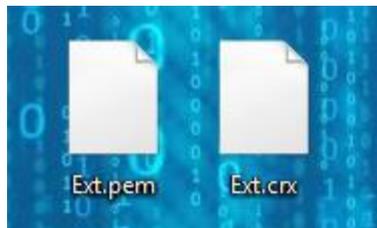
صورة رقم ٤.٢.١ لتصدير الإضافة إلى شكلها النهائي قبل الرفع إلى السوق

وهذه النافذة المنبثقة تخبرنا أين هو مسار إضافتنا التي نريد تحويلها للصيغة النهائية CRX كما تشير هذه الصورة:



صورة رقم ٤.٢.٢ لتصدير الإضافة إلى شكلها النهائي قبل الرفع إلى السوق

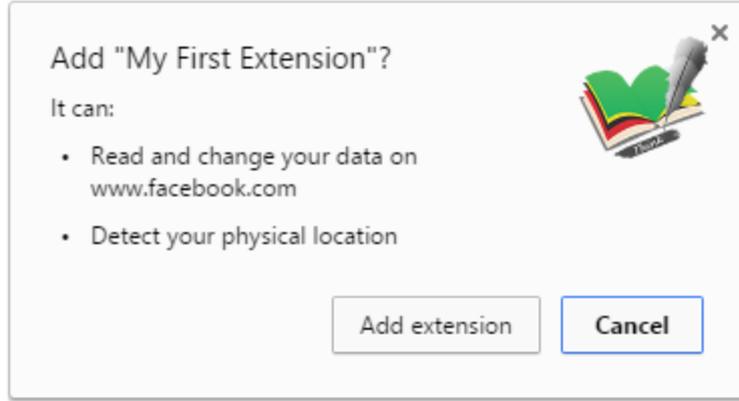
كوننا نقوم بهذه العملية لأول مرة، لا داعي لوضع مسار ملف Private Key لأننا لا نملكه بعد، لذلك بمجرد انتهائنا من تحديد مسار إضافتنا والضغط على Pack Extension سنرى الصيغة النهائية للملحق بالإضافة لملف آخر بجانبه بصيغة PEM:



صورة رقم ٤.٢.٣ توضح الملفات المُخرجة

هذان الملفان سنجدهما بنفس مسار مجلد الإضافة الأصلي ، Ext.crx هو ملف الذي يحمل الصيغة النهائية للإضافة، ويمكن سحبه وإفلاته مباشرة في صفحة الإضافات المُسطبة، ويمكن كذلك رفعه مباشرة على استضافتك الخاصة حتى يقوم آخرون بتجربته، وكأنك تحمّل الإضافة من السوق لأول مرة.

هذا ما سيحصل عندما تقوم بسحب Ext.crx إلى صفحة الإضافات:



صورة رقم ٤.٢.٤ عند تركيب إضافة لأول مرة بصيغتها النهائية CRX

نفس الصندوق الذي سيظهر فور تحميلنا للإضافة من السوق الرسمي لكروم، يُظهر الصندوق فور تحميلنا للإضافة: اسم الإضافة، أيقونتها، والصلاحيات التي تحتاجها الإضافة للعمل بشكل صحيح، يمكن للمستخدم الرفض (والذي ينتج بالطبع عدم تحميل الإضافة) أو القبول بالضغط على Add extension لتُضاف لقائمة الإضافات المُسطبة في صفحة الإضافات.

ماذا إذاً بخصوص ملف PEM الآخر؟

إذا أردنا أن نقوم بعمل تعديل آخر للإضافة أو إضافة المزيد من المزايا إلى مُلحقنا، ومن ثم التحويل للصيغة النهائية CRX مرة أخرى، فنحن الآن بحاجة لملف PEM الذي قام متصفح كروم بتوليده لنا أثناء قيامنا لأول مرة من تحويل ملفات الإضافة إلى CRX، فهو بمثابة الختم الذي ختمه لنا متصفح كروم عندما أنشأ لنا لأول مرة ملف CRX، وإذا أردت التعديل مرة أخرى على محتوى الإضافة وتوليد CRX جديد، أنا بحاجة للختم لأثبت هوية مُلحقي ويعطيها ما يُسمى بـ ID أو المعرّف الفريد الخاص بالإضافة.

كما أشرت سابقاً، نحن بالواقع لا نحتاج إلى كل هذه العمليات عند رفع الإضافة إلى السوق، فعند رفع الإضافة للسوق لأول مرة يقوم بشكل تلقائي بهذه العملية، ويحفظ لنا الختم لديه دون رفعه كل مرة إلى السوق! لكن أردت الإشارة إلى هذه النقطة وكيفية القيام بها في حال وجدت موقع من المواقع يوفر الإضافة خارج سوق متصفح كروم وتحميل ملف ينتهي بصيغة CRX.

٤.٣ الرفع على سوق كروم أو على استضافتك الخاصة

نهاية شهر فبراير من عام ٢٠١٤، والإصدار رقم ٣٣ من متصفح كروم، وبسبب المشاكل الأمنية المتزايدة التي تعرض لها المُستخدمين بسبب المُلحقات التي كان هدفها الضرر بالمُستخدمين، فقد أُجبرت جوجل المطورين على رفع ملحقاتهم و فقط على سوق كروم (المصدر)! ولا يمكن للمُستخدم تحميل أي إضافة مُستضافة خارج سوق كروم الإلكتروني، وإن تم، سيحصل على تحذير بذلك.

يمكنك رفع إضافتك أو ملحقاتك على استضافتك الخاصة في حالتين فقط:

- إذا قام المُستخدم بتفعيل وضع المطورين، وهو الأمر الذي قمنا نحن به سابقاً لتجربة الإضافة.

- ما يُسمى بـ Enterprise Policy وهو ما يوضّح تفاصيله بالضغط هنا.

ولكن، وكما رأينا في عملية تحويل الإضافة للصيغة النهائية CRX وأنها عملية مُنهكة بعض الشيء، فرفع الإضافة مباشرة إلى السوق على شكل ملف مضغوط ZIP ويقوم السوق لوحده بتحويل ملحقاتنا إلى الصيغة النهائية دون تدخل من المطور، قد زاح الكثير من الوقت لصالح المطورين.

كما ولن يقلق المطور من عملية تطوير الملحقات وإعلام المُستخدمين بأن هناك نسخة جديدة من إضافاتهم ويجب أن يقوموا بتحميلها يدوياً، فمتصفح كروم سيقوم تلقائياً بتحميل النسخة الأحدث من ملحقتك أو إضافتك دون تدخل من المُستخدم بمجرد رفعك للنسخة الأحدث إلى السوق، وبحلول ٢٤ ساعة تكون النسخة الأحدث في كافة متصفحات مُستخدميك.

٤.٤ من داخل صفحة المطورين

حتى يمكننا نشر إضافتنا الآن إلى السوق، يلزمنا شراء رخصة مطور والتي كما أسلفنا تُكلف لمرة واحدة ٥ دولارات فقط، بعد إتمام هذه العملية البسيطة يصبح لديك صفحة خاصة لمراقبة نشاط إضافاتك الحالية أو إضافة ملحقات جديدة. صورة رقم ٤.٤.١ تشرح لنا مجمل الإضافات التي قمت بنشرها على السوق وحالة كل واحدة منهم:

Developer Dashboard

Your Listings (1 - 9 of 9)

	Created	Last published	Weekly users	Status	
 Facebook Mention Tagging Blocker Version 2.0.6 ★★★★★ (9) Target users in other languages. More info ▼	5/23/14	7/29/15	804	Published	User Feedback (1) Stats Unpublish Edit More info
 RSS Timeliner Version 1.1.7 ★★★★★ (6) Target users in other languages. More info ▼	11/20/14	7/12/15	270	Published	User Feedback Stats Unpublish Edit More info
 Tabs Board Version 2.5 ★★★★★ (5) Target users in other languages. More info ▼	6/14/14	6/4/15	231	Published	User Feedback Stats Unpublish Edit More info
 Facebook Invitations Blocker Version 2.0.3 ★★★★★ (2) Target users in other languages. More info ▼	9/30/14	5/24/15	352	Published	User Feedback Stats Unpublish Edit More info

صورة رقم ٤.٤.١ من داخل صفحة المطورين الخاصة بالإضافات والملحقات

Add new item

Your Developer Account

- You have published 6 item(s) (excluding themes) out of your maximum allotted 20.
- Edit your developer display name: **Dev. Mohammad M. AlBanna**
- Edit your tester accounts: (not set)
These test accounts allow you to make your application visible only to specific users for testing. [Learn more](#)
- Edit your User Feedback preferences: **User Feedback is enabled**
- This publisher is currently linked to Google Wallet account [REDACTED] to receive payments for paid items.
[Access Google Wallet account](#) to review orders, get reports and edit your account information.
- Edit your communication preferences: **Communication not allowed**
- Google Group publishing: [Choose existing group](#) or [create new group](#) (you must be the owner or administrator).
Allows multiple developers to act on the publisher's behalf. [Learn more](#)

Save Changes

صورة رقم ٤.٤.٢ من صفحة المطورين الخاصة بالإضافات والملحقات

شرح لبعض الأمور:

- لإضافة مُلحقنا الجديد نقوم بالضغط على زر Add new item.

- جوجل تسمح لنا برفع حتى ٢٠ مُلحق أو إضافة، وإذا أردنا المزيد نرسل طلب آخر لجوجل حتى يتسنى لنا رفع المزيد، ستقوم جوجل بمراجعة حسابك وطبيعة إضافتك وإن كانت قيّمة قد تسمح لك بالمزيد!

- يمكننا تعديل بعض المعلومات الهامة كإسم المطور الذي سيظهر على صفحة الإضافة بالسوق، وكما هو موضح الإسم الذي يظهر هو Dev. Mohammad M. AlBanna، ومن جهة السوق سيظهر الإسم على هذا الشكل:

صورة رقم ٤.٤.٣ توضح عرض اسم المطور كموفر رسمي للإضافة

- يمكننا أن نحدد قائمة مختبرين لإضافاتنا ومُلحقاتنا عبر بريدهم الإلكتروني، لتصلهم دعوة فيما بعد لتجربة الإضافة من السوق، من الممكن أن يكون هؤلاء بقية أعضاء فريق التطوير مثلاً، من خلال تحديد Testers Accounts وبريدهم الإلكتروني.

٤.٥ رفع إضافة/ملحق جديد إلى السوق من صفحة المطورين

وبحماس شديد، دعنا نرفع أول إضافة لنا إلى السوق عن طريق الضغط على زر Add New Item أو إضافة عنصر جديد، وهذه هي الصفحة الخاصة في إضافة أول إضافة/ملحق لنا:

Developer Dashboard > Add new item

Upload an extension or app (.zip file)

Choose file Upload

Uploading an item:

- Upload a ZIP file of your item directory, not a packaged CRX file.
- Include a well-designed product icon in your manifest ([more info](#)).
- [Read the documentation](#) about creating and packaging apps.
- Need more help? Check out the [Chrome Web Store developer documentation](#).

صورة رقم ٤.٥.١ تعرض كيفية رفع أول إضافة لنا على السوق

كما نشاهد بالصورة أعلاه، لا داعي لرفع الإضافة بصيغتها النهائية التي أشرنا لها سابقاً بـ CRX، بل نرفعها كملف مضغوط على هيئة ZIP، نختار Choose File ونحدد موقع الملف المضغوط الخاص بالإضافة ومن ثم الضغط على Upload:

Developer dashboard > My First Extension

Edit Item In-app Products

Upload

**My First Extension**
Short name: Not specified in manifest
Version 1.0.0 by Dev. Mohammad M. AlBanna
Upload Updated Package
Description

Detailed description


Icon

صورة رقم ٤.٥.٢ وإضافة أولى مُلحقاتنا ومعلوماتها إلى سوق كروم

أهلاً بكم في أولى إضافاتنا وملحقاتنا! سنلاحظ في القسم الأول حول اسم الإضافة ورقم النسخة وغيرها من المعلومات تم أخذها تلقائياً من ملف Manifest.json الذي قمنا بإنشائه كأول خطوة. هذا القسم غير معقد البتة، قم بكتابة وصف الإضافة، وتحديد الأيقونات وغيرها من الأمور الروتينية! سأتطرق لنقاط هامة في هذه الصفحة:

Websites

Verify that this is an official item for a website you own:

None

[Add a new site](#) [Refresh list](#)

Link to website for your item (optional)

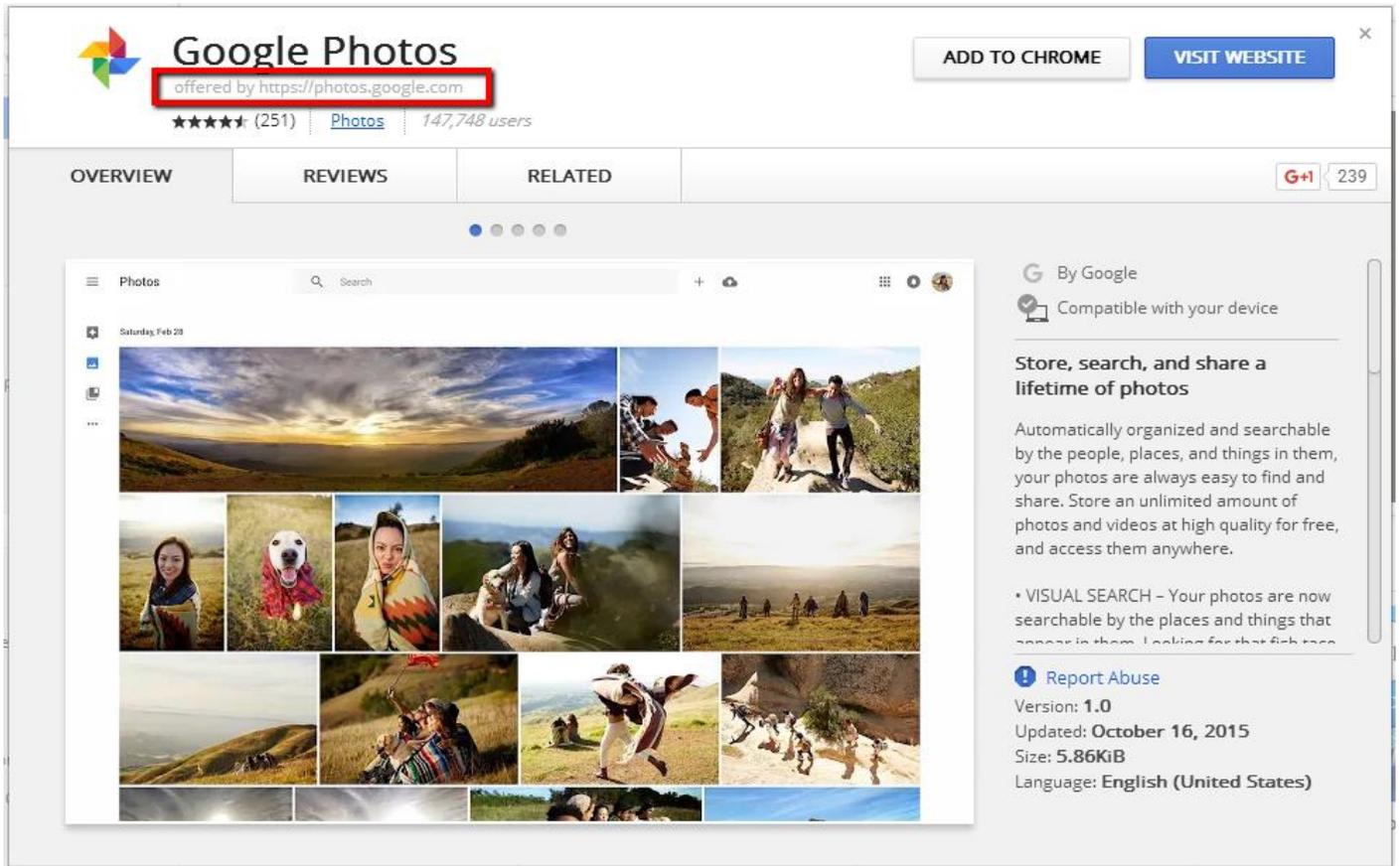
Link to support & FAQ for your item (optional)

If your item is associated with a website that you own, select that website from the list. You can register as the owner for a site using Google Webmaster Tools. [More info](#)

Providing the URLs for description and support pages can improve your users' experience and help make this item's ratings and comments more meaningful. [More info](#)

صورة رقم ٤.٥.٣ لربط الإضافة بموقع إلكتروني كَمقدم رسمي للإضافة

- في حال كان لإضافتك أو ملحقتك موقع شخصي، وتريد ربط الموقع الشخصي بالملحق بحيث من يستعرض الإضافة من السوق تُظهر له أن هذا الملحق مقدم من موقع كذا وكذا كما في صورة رقم ٤.٥.٤، ولا يعرض له إسم مطور الإضافة ذاتها، قم فقط بالضغط على Add a new site.



صورة رقم ٤.٥.٤ توضح أن مقدم إضافة Google Photos هو موقع photos.google.com

الضغط على Add a new site سينقلك لخدمة جوجل التي تُسمى Google Webmasters Tools وهي خدمة مُقدمة من جوجل تساعد أصحاب المواقع لمراقبة مواقعهم وأرشفتها في محركات البحث، وهو موضوع آخر تماماً! لكن بالمختصر المفيد سيطلب جوجل لإثبات ملكية الموقع برفع ملف صغير إلى مُستضيف الموقع أو وضع عنصر جديد في الصفحة الرئيسية داخل head.

بعد إتمامك لهذه العملية، والضغط على Refresh List ستجد رابط موقعك الإلكتروني أصبح مضاف في القائمة ويمكنك ببساطة اختياره كجهة رسمية مُقدمة للإضافة.

يمكن كذلك القيام بهذه العملية في حال قمت بتطوير إضافة خاصة لشركة ما، بالطبع إما ستطلب منهم إنشاء حساب مطور خاص بهم، أو رفع إضافتهم على حسابك الشخصي مع الإشارة أن مقدم الإضافة هو موقعهم الإلكتروني، وبالتالي يجب عليك إضافة موقعهم في قائمتك ليسمح لك سوق الكروم بالتحدث بإسمهم ونسب الإضافة لهم. وهذا يتم من خلال ربط حساب Google Webmasters Tools مع الموقع الإلكتروني للشركة أو الجهة المالكة للملحق.

- في حال كانت الإضافة مدفوعة وليست مجانية، هذا هو الخيار الأنسب لذلك:

Pricing and payments

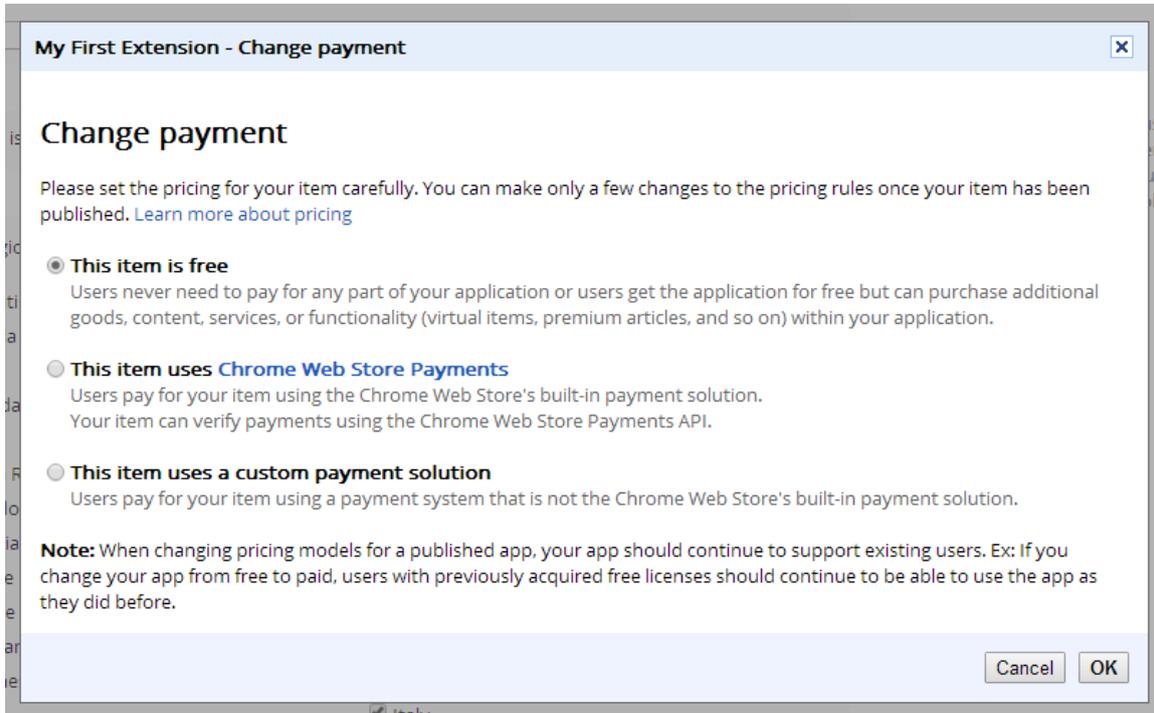
This item is free.

Change pricing

You can charge for your app using Chrome Web Store Payments or a custom payment solution. If your app is free, you can still make money by using the Google In-App

صورة رقم ٤.٥.٥ تحدد مجانية الإضافة أو جعلها مدفوعة للاستعمال

بالضغط على Change Pricing، يمكنك تحديد تسعيرة مُلحقك أو إضافتك كما هو موضح بالصورة التالية:



صورة رقم ٤.٥.٦ لتحديد وسائل الدفع (الافتراضي هو مجانية)

من هذه الشاشة يمكن تحديد وسيلة الدفع! إما أن تكون الإضافة مجانية، أو نستخدم وسيلة الدفع التي توفرها لنا جوجل عبر السوق، أو وسيلة دفع أخرى لا توفرها جوجل. وسنتطرق لهذا القسم بشكل مفصل لاحقاً في هذا الكتاب.

- في حال كانت الإضافة تطلب تسجيل دخول المُستخدمين من حساباتهم على جوجل، فيجب عليك أن تتأكد من تفعيل هذا الخيار:

OpenID

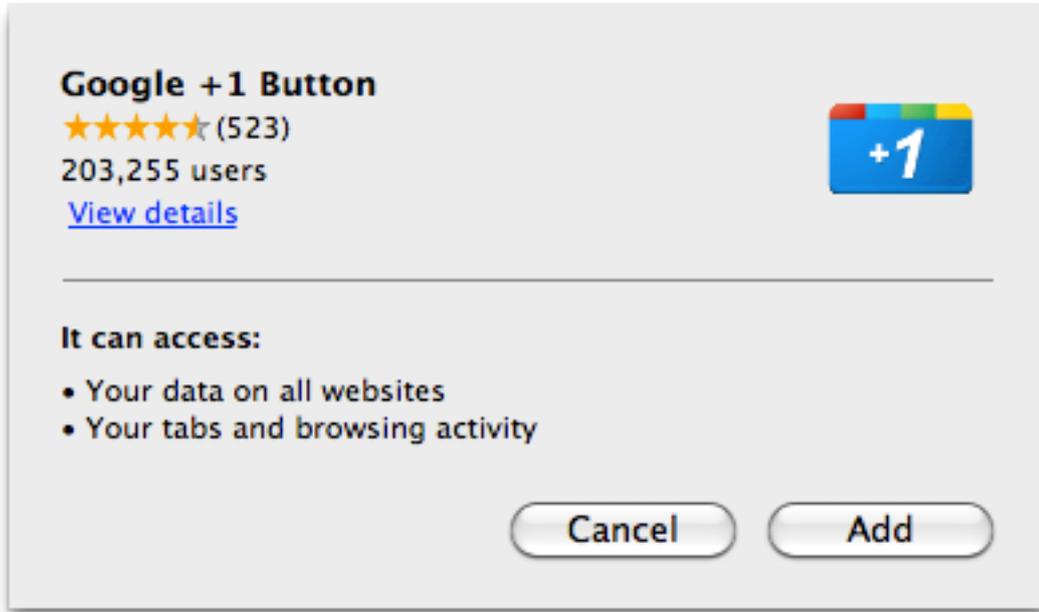
This item uses Google OpenID to authenticate users.

If your app needs to identify users, supporting Google Account identities (using Google's OpenID endpoint) provides a seamless transition from the store to your app. [More info](#)

صورة رقم ٤.٥.٧ لإجبار المُستخدمين قبل تحميل المُلحق تسجيل الدخول لحساب جوجل

هذا الخيار سيجبر أي مُستخدم قبل تحميل إضافتك على المتصفح، يجبرهم على تسجيل الدخول أولاً عبر حساب جوجل ليتمكنوا من تحميل إضافتك.

- هل سبق لك وزرت موقعاً إلكترونياً ما، وعرض عليك تحميل إضافة أو مُلحق خاص بهذا الموقع عبر الضغط على زرٍ ما؟ الضغط على هذا الزر لن ينقلك إلى صفحة الإضافة داخل السوق، بل الضغط على الزر سيُظهر صندوق تحميل مباشر لترغب الإضافة مباشرة على المتصفح وأنت داخل الموقع حتى!



صورة رقم ٤.٥.٨ مثال لتحميل الإضافات من مواقعها الرسمية دون التوجه لصفحة الإضافة في السوق

يُسمى هذا النوع من تحميل الإضافات دون الذهاب إلى صفحة الإضافة نفسها إلى السوق بـ **Inline Install**، أي التحميل من موقعك دون نقل المُستخدم إلى صفحة الإضافة إلى السوق، ولتفعيل هذه الخطوة أنت بحاجة لتفعيل هذا الخيار:

Inline install This item uses inline install. Select this checkbox to allow inline install for your item (note that this may delay publishing). [More info](#)

صورة رقم ٤.٥.٩ لتفعيل خيار **Inline Install**

بتفعيل هذا الخيار، أنت تُشير أن الإضافة من الممكن تحميلها عبر الضغط على زر ما، لكن هل هذا كل شيء؟ أنت تحتاج لإضافة بعض الأكواد البرمجية من جهة موقعك الشخصي حتى يتعرف المتصفح على الإضافة، وهذا المصدر سيشرح لك بالتفصيل كيف تقوم بذلك باستخدام واجهة جوجل البرمجية. - يمكنك كذلك تفعيل **Google Analytics** وهي خدمة من جوجل مجانية كذلك في إحصاء زوّار موقعك ومن أين أتوا وتفاصيل أخرى، يمكنك كذلك تطبيق نفس الأمر على زوّار صفحة إضافتك على السوق من خلال ال **ID** عندما تقوم بإنشاء متتبع جديد في خدمة **Google Analytics**:

Analytics Google Analytics ID UA- (example: 1234567-1)

صورة رقم ٤.٥.١٠ لإضافة رقم متتبع من خدمة جوجل **Google Analytics**

كل ما عليك القيام به بعد إنشاء المتتبع الجديد من خدمة **Google Analytics** هو وضع الرقم بعد **UA-**، وستحصل على تقارير زوّار إضافتك من على سوق جوجل كروم. - أخيراً وليس آخراً، هل الإضافة جاهزة للاستعمال من قبل كافة المُستخدمين من حول العالم؟ في آخر قسم يمكن أن تحدّد ذلك:

Visibility options

- Public**
Everyone can see it
- Unlisted**
Only people with the link can see it.
- Private**
Only trusted testers from your developer dashboard can see it.
You can also include members of a [Google Group](#) that you own or manage.

صورة رقم ٤.٥.١١ توضح خيار ظهور الإضافة في السوق من عدمه

الخيار الأول **Public**: الظهور لعموم الزوّار والمستخدمين لسوق كروم.

الخيار الثاني **Unlisted**: فقط لمن يملكون رابط الإضافة على السوق.

الخيار الثالث **Private**: يمكن أن تحدد فئة مخصصة لإختبار إضافتك والتي أشرنا لها سابقاً في بداية الدخول لصفحة المطورين، سيصل فقط دعوة للمدخل بريدهم الإلكتروني لتجربة الإضافة فقط.

- كل شيء على ما يُرام الآن؟ يمكنك أن تفعل التالي:

Discard draft Save draft and return to dashboard

Preview changes Publish changes

صورة رقم ٤.٥.١٢ توضح الإجراء الأخير

من اليمين إلى اليسار:

Publish Changes: أي النشر فوراً، وكما أشرنا سابقاً بالعادة لا تأخذ العملية أكثر من ساعتين كحد أقصى ما لم تُثر الإضافة الشك لتصبح قابلة للتحميل والعرض في السوق.

Preview Changes: عرض كافة التعديلات كما لو أن الإضافة تم قبولها من طرف جوجل وتعرض بالسوق الآن، سترأها أنت كما يراها المستخدمون بعد النشر على السوق.

Save draft and return to dashboard: أي حفظ التعديلات كمسودة يمكن الرجوع لها لاحقاً ولا تقم بتطبيق التعديلات على الإضافة في السوق، وعودة إلى صفحة المطورين.

Discard draft: أي لا تقوم بحفظ التعديلات التي قمت بها قبل قليل.

٤.٦ مسببات لرفض إضافتك ومُلاحقاتك من السوق

في هذا القسم، سأضع وأجمل لك ٩٠% من الأسباب لرفض الإضافات والمُلاحقات من السوق، حتى بعد قبولها! أجل، لا تتوتر إن أتك إشعار إزالة عبر البريد الإلكتروني ليخبرك عن إزالة منتجك البرمجي أو تحفتك الفنية من السوق لأسباب غير واضحة، ومن هذه الأسباب:

١ - الكود غامض ويخفي الكثير:

لا نتحدث عن أنّ الكود مشقّر، هناك مصطلح آخر يوازيه وهو **Obfuscated** ويعني أن الكود غامض، ويوجد العديد من الخدمات المجانية التي تجعل من الكود الخاص بك كود ممّوه أو صعب قراءته تحديداً في أكواد جافاسكريبت.

دعونا في البداية نرى الرسالة التي من الممكن أن تتلقاها من جوجل:

Chrome Web Store: Removal notification for Tabs Board

Dear Developer,

We have reviewed your item and it does not comply with our program policies. Please re-review this section of our policy:

"Where possible, make as much of your code visible in the package as you can. If some of your app's logic is hidden and it appears to be suspicious, we may remove it."

If you'd like to re-submit your item, please make the appropriate changes to the item so that it complies with our policies, then re-publish it in your developer dashboard. Please reply to this email for issues regarding this item removal.

Thank you for your cooperation,
Google Chrome Web Store team

صورة رقم ٤.٦.١ وإشعار بإزالة إحدى إضافاتي من السوق

هذا نموذج من رسالة ستصلك عبر البريد الإلكتروني يخبرك عن اسم الإضافة التي تمت إزالتها من السوق، والجزء المُشار إليه بالمستطيل الأحمر هو الجزء الذي تُزال بسببه العديد من الإضافات بالسوق. فما هو الكود الممّوه أو الغامض؟

هو الكود الذي يصعب قراءته أو تتبع تعليماته من قبل مطور آخر، كتغيير أسماء المتغيرات، والدوال، وبعض الخدمات الأخرى توفر لك ميزة تغيير أماكن النصوص أو ترميزها ليصعب تعديلها كذلك.

كل ما سبق هو ضد سياسة جوجل في تطوير إضافات المتصفح! فجوجل تريد أن ترفع الإضافة بالشكل الذي قمت أنت بتطويره، حتى يتسنى لهم مراجعتها يدوياً من قبل مطورين آخرين ليتأكدوا من سلامتها ولا تُشكل خطراً على المستخدمين. فاحذر من رفع إضافتك للسوق بكود ممّوه واجعله دائماً واضح وسهل القراءة.

مبرمجوا الويب تعوّدوا بالطبع عمل ضغط لأكوادهم البرمجية لتظهر كلها بسطر واحد فقط، لا وجود لأسطر فارغة ولا حتى وجود تعليقات، كما هو بالصورة التالية:

```

/*! jQuery v1.11.3 | (c) 2005, 2015 jQuery Foundation, Inc. | jquery.org/license */
!function(a,b){"object"==typeof module&&"object"==typeof module.exports?module.exports=a.document?b(a,!0):function(a){if(!a.document)throw new Error("jQuery requires a window with a document");return b(a)}:"undefined"!=typeof window?window:this,function(a,b){var c=[],d=c.slice,e=c.concat,f=c.push,g=c.indexOf,h={},i=h.toString,j=h.hasOwnProperty,k={},l="1.11.3",m=function(a,b){return new m.fn.init(a,b)},n="/^\s\uFFFF\u0000+|[\s\uFFFF\u0000]+$/g,o=/^-ms-/,p=-/([\da-z])/gi,q=function(a,b){return b.toUpperCase()};m.fn=m.prototype={jquery:l,constructor:m,selector:"",length:0,toArray:function(){return d.call(this)},get:function(a){return null==a?this[a+this.length]:this[a].call(this)},pushStack:function(a){var b=m.merge(this.constructor(),a);return b.prevObject=this,b.context=this.context,b},each:function(a,b){return m.each(this,a,b)},map:function(a){return this.pushStack(m.map(this,function(b,c){return a.call(b,c,b)})),slice:function(){return this.pushStack(d.apply(this,arguments))},first:function(){return this.eq(0)},last:function(){return this.eq(-1)},eq:function(a){var b=this.length,c=a+(0>a?b:0);return this.pushStack(c>=0&&b<c?[this[c]]:[])},end:function(){return this.prevObject||this.constructor(null)},push:f,sort:c.sort,slice:c.slice},m.extend=m.fn.extend=function(){var a,b,c,d,e,f,g=arguments[0]||{};h=1,i=arguments.length,j=1;for("boolean"==typeof g&&(j=g,g=arguments[h]||{}),h++),"object"==typeof g||m.isFunction(g)||({g:{},h===i&&(g=this,h--);>h;h++)if(null!=(e=arguments[h]))for(d in e)a=g[d],c=e[d],g!=="c&&(j&&c&&(m.isPlainObject(c)||b=m.isArray(c)))?(b=b||1,f=a&&m.isArray(a)?a:[]):f=a&&m.isPlainObject(a)?a: {},g[d]=m.extend(j,f,c);void 0!=="c&&(g[d]=c);return g},m.extend({expand:"jQuery"+(1+Math.random()).replace(/\D/g,""),isReady:!0,error:function(a){throw new Error(a)},noop:function(){},isFunction:function(a){return"function"===m.type(a)},isArray:Array.isArray||function(a){return"array"===m.type(a)},isWindow:function(a){return null!=a&&a=a.window},isNumeric:function(a){return!m.isArray(a)&&parseFloat(a)+1==0},isEmptyObject:function(a){var b;for(b in a)return!0},isPlainObject:function(a){var b;if(!a||"object"!==m.type(a)||a.nodeType||m.isWindow(a))return!1;try{if(a.constructor&&!j.call(a,"constructor")&&j.call(a.constructor.prototype,"isPrototypeOf"))return!1}catch(c)

```

صورة رقم ٤.٦.٢ عبارة عن كود مضغوط للمكتبة الشهيرة jQuery

هذا جزء من مكتبة jQuery أو ما يُسمى بـ jQuery Minimized، نرى أن الكود كله بجانب بعضه البعض ولا مسافات وغير سهل للقراءة بالملء! لذلك فور تحميلك لمكتبة jQuery تتيح لك خياران: إما تحميل المكتبة الكاملة دون ضغط أو نسخة مضغوطة Minimized وهي النسخة التي نقوم باستخدامها بالغالب لأنها توفر مساحة للملف ومنه محافظة على الـ Bandwidth الخاص بموقعنا الإلكتروني.

هذا أيضاً مخالف لسياسة جوجل، على الكود أن يكون كما هو واضح وسهل، حتى مع وجود مكاتب مشهورة ومعروفة كـ jQuery مثلاً فيجب أن ترفقها في ملحقاتك أو إضافاتك بشكلها الأصلي والكامل وليس الشكل المضغوط... وإلا **ستعريض إضافتك للإزالة من السوق**.

لا تقلق من ناحية حجم الإضافة في حال فعلاً استخدمت فيها مكتبات خارجية، سيصبح حجم إضافتك بالميجا بايت فعلاً لو استخدمت مكتبات غير مضغوطة، لكن لا تقلق! فكما قلنا أن الصيغة النهائية للإضافات هو CRX، هذه الصيغة ستضغط الكود بالشكل اللازم حتى لا تُشكل حجم الإضافة عائق للمستخدمين الآخرين، كل شيء معمول له حساب على ما يبدو.

٢ - الاتصال والتواصل مع API أو مع موقع ما:

قد يبدو هذا أمر غريب نوعاً ما ولكن حصل معي بالفعل، مجمل الرسالة التي ترسلها لنا جوجل هو "شيء ما مخفي وغير واضح بالكود"، لو فعلاً كانت إضافتك واضحة جداً وغير مضغوطة كما أسلفنا بالنقطتين السابقتين ووصلك إشعار بإزالة إضافتك من السوق، إذاً عليك النظر حول الطلبات التي تقوم بها إضافتك أو ملحقك الخاص مع الواجهات البرمجية والخدمات الأخرى. يجب أن يكون ما يُرسل وما يُستقبل واضح جداً حتى يتسنى لمراجعي إضافتك من جوجل معرفة ذلك.

أسباب سابقة طرحها جوجل في سياسة المحتوى:

ما سبق ذكره، لم توضّحه جوجل فعلاً في [سياسة المحتوى](#)، وهو ناتج عن محصلة تجارب عام كامل من تطوير الإضافات والملحقات لي ولعملائي. لكن جوجل تذكر في سياسة المحتوى الخاص بالإضافات بأن لا تشمل مثلاً المحتوى الإباحي، المحتوى الذي ينتهك حقوق ملكية ما، أو ملحقاتك غير قانونية. وغيرها العديد من الأمور التي يجب أن توافق سياسة جوجل في النشر كما هو مذكور من [هذا المصدر](#).

لذلك عزيزي المطور/ة، إذا قوبلت إضافتك للمرة الأولى على السوق فلا تفرح كثيراً، قد يتم إسقاطها فيما بعد إذا وُجدت أنها مثيرة للشك أو غير واضحة كما أسلفنا بالسابق، فبليلة وضحاها تم إسقاط آلاف الملحقات من السوق وبعضهم لم يتم إشعار مطورهم حتى! اليوم جوجل أصبحت أكثر صرامة في قبولها للملحقات ليستضيفها السوق الخاص بهم ويثق المستخدمين بما يُقدمه السوق لهم.

الأصل، جوجل ترسل رسالة توضّح فيها أسباب إزالة الإضافة، لو لديك استفسار يمكنك الرد على الرسالة ذاتها وستحصل على رد قد يكون متأخر (من يومين إلى ثلاثة أيام) بالغالب، ويمكن أن تطرح مشكلتك في مجموعات جوجل للمساعدة.

كيف أعرف حالة إضافتي إذا تم إسقاطها أم لا؟

سيصلك إشعار بذلك بالتأكيد، لكن عند دخولك لصفحة المطورين، ستجد حالة كل إضافة خاصة بك:

 TEST Version 1.0.0 ★★★★★ (0) Target users in other languages. More info ▼	10/17/13	0	Draft	User Feedback Publish Edit More info
 ★★★★★ (1) Target users in other languages. More info ▼	12/15/14	12/15/14	1	User Feedback Publish Edit More info
 ★★★★★ (1) Target users in other languages. More info ▼	12/12/14	12/12/14	1	User Feedback Stats Unpublish Edit More info

صورة رقم ٤.٦.٣ توضح حالة الإضافات من صفحة المطورين

كيف أتواصل مع جوجل حول مشكلة ما في مُلحقاتي؟

كما أشرت سابقاً، يمكنك الرد على رسالة إشعار الإزالة الذي وصلتك بالبريد الإلكتروني، لكن مع تزايد شكاوي المطورين حول الدعم الفني السيء في فترة من الفترات، فقد قامت جوجل بتوفير صفحة خاصة للمطورين بإرسال مشكلاتهم التي يواجهونها حول مُلحقاتهم، شكل النموذج:

Chrome Web Store Help

CHROME WEB STORE

Please submit your issue

Please submit your issue and we will get back to you as soon as we can.

First name *
Mohammad

Last name *
AlBanna

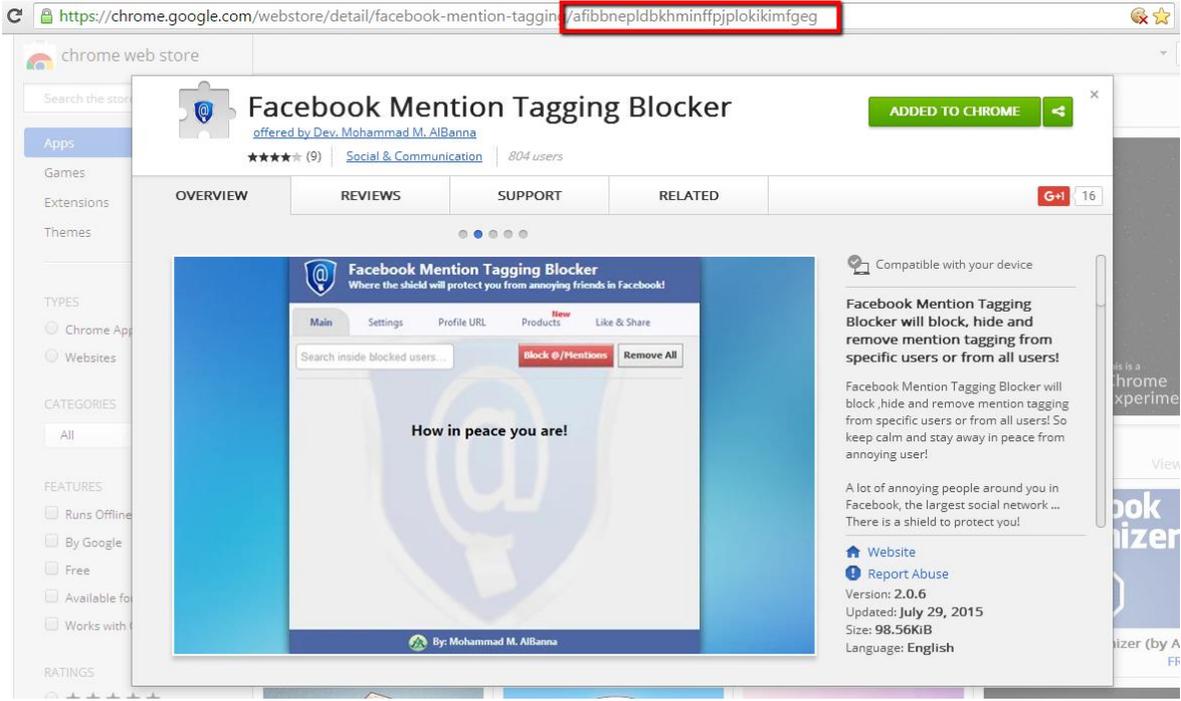
Email address *
|

Chrome Web Store Item URL or Item ID *
To determine the ID value for a specific item, go to the Chrome Developer Dashboard and click the name of the relevant app or extension. The ID value for that item will be shown in the browser address bar, it is a string of 32 Latin characters at the end of the URL in the address bar.

صورة رقم ٤.٦.٤ توضح نموذج الاتصال مع فريق الدعم الفني لسوق كروم

أرسل وصف المشكلة التي تواجهك (كأن تم إزالتها دون إشعارك مثلاً) وحدد ال ID أو ما يُسمى بالمعرّف الفريد الخاص بالملحق أو الإضافة الخاصة بك.

يمكن الحصول على المعرف الفريد بعدة طرق، فهذا هو معرّف الـ ID الخاص بأي إضافة:



صورة رقم ٤.٦.٥ توضح الحصول على المعرف من خلال الرابط الخاص بالإضافة على السوق

يمكن كذلك الحصول على ID من لوحة التحكم الخاصة بالمطورين، عبر الضغط على عرض المزيد من المعلومات حول أي إضافة من الإضافات المعروضة أمامك:



صورة رقم ٤.٦.٦ توضح آلية الحصول على المعرف لأي إضافة من صفحة المطورين

وهو ما ستضعه في خانة المعرّف المطلوبة في النموذج الذي سترسله لجوجل لمعرفة أي من الإضافات أو الملحقات التي واجهتك فيها المشكلة:

Last name *
AlBanna

Email address *

Chrome Web Store Item URL or Item ID *
To determine the ID value for a specific item, go to the Chrome Developer Dashboard and click the name of the relevant app or extension. The ID value for that item will be shown in the browser address bar; it is a string of 32 Latin characters at the end of the URL in the address bar.

Please describe your issue *

Submit a screenshot or crx file with your issue *
Choose File No file chosen

SUBMIT * Required field

صورة رقم ٤.٦.٧ توضح أين تضح المعرّف الفريد لإضافتك في نموذج التواصل مع الدعم الفني الخاص بسوق بكموم أخيراً وليس آخراً، يمكن وضع صورة للرسالة التي وصلتكم من الدعم الفني مثلاً، أو إرسال الصيغة النهائية للإضافة CRX بالطريقة التي تطرقنا لها سابقاً. بعد ذلك الضغط على Submit لإرسال النموذج وانتظر الرد... وتمنيتي بأن يكون ردّاً إيجابياً دائماً.

٥. الكسب المادي من الإضافات

أغلب المطورين والشركات يقدّمون ملحقاتهم بشكل مجاني تماماً، ويعتمدون على الكسب المادي من الإضافات عن طريق ربطها بخدمات مدفوعة أخرى تقدمها الشركات أو يقدمها المطورون. لكن، وكما تجري العادة في أي سوق إلكتروني يسمح لك بعرض منتجاتك إما بشكل مجاني أو مدفوع، سنتطرق في القسم الأخير من هذا الكتاب الإلكتروني حول الأساليب القانونية والمتاحة للربح من إضافاتك وملحقاتك.

توفر جوجل ٤ خيارات للحصول على العائد المادي من الإضافات مُلخصة في هذه الصورة:

Freemium	A free trial version of your product that prompts users for payment to unlock advanced features or regain access after a time-limit.
One-time Payment	Users buy your product by providing a one-time payment upfront. Note: This is the only monetization option for themes.
Subscription	Users purchase access to your product on a subscription basis.
Custom	You are free to implement a custom model that works best for you. For in-app payments, consider using the Chrome Web Store API .

صورة رقم ٥.١ توضح الخيارات المتاحة للكسب المادي من الإضافات والملحقات

- **Freemium**: وتعني توفير نسخة تجريبية لبعض الوقت بخصائص محددة ومقيّدة، وعلى المُستخدم أن يشتري النسخة بعد مرور الوقت. ومن الممكن كذلك توفير نسخة مجانية كلياً، لكن تسمح للمُستخدم بشراء مزايا داخلية جديدة للوصول للوظائف الجديدة داخل الإضافة.
- **One-time Payment**: وتعني الدفع لمرة واحدة، يشتري المُستخدم الإضافة الخاصة بك فقط.
- **Subscription**: الإشتراك بالإضافة، إما اشتراك شهري أو سنوي.
- **Custom**: تحديد نظام خاص بك بعيداً عن واجهة جوجل البرمجية أو التعامل مع سوق جوجل، كإنشاء متجر إلكتروني.

تحدثنا سابقاً حول تحديد ما إذا كانت الإضافة مجانية أو مدفوعة أثناء رفع إضافتنا أو مُلحقنا إلى السوق:

Pricing and payments

This item is free.

Change pricing

You can charge for your app using Chrome Web Store Payments or a custom payment solution. If your app is free, you can still make money by using the Google In-App

بالضغط على زر Change Pricing نبدأ بتحديد الآلية بشكل مخصص، وهو ما أشرنا إليه في هذه الصورة:

My First Extension - Change payment

Change payment

Please set the pricing for your item carefully. You can make only a few changes to the pricing rules once your item has been published. [Learn more about pricing](#)

- This item is free**
Users never need to pay for any part of your application or users get the application for free but can purchase additional goods, content, services, or functionality (virtual items, premium articles, and so on) within your application.
- This item uses Chrome Web Store Payments**
Users pay for your item using the Chrome Web Store's built-in payment solution. Your item can verify payments using the Chrome Web Store Payments API.
- This item uses a custom payment solution**
Users pay for your item using a payment system that is not the Chrome Web Store's built-in payment solution.

Note: When changing pricing models for a published app, your app should continue to support existing users. Ex: If you change your app from free to paid, users with previously acquired free licenses should continue to be able to use the app as they did before.

Cancel OK

٥.١ شراء المُلحق لمرة واحدة، أو الاشتراك الشهري/السنوي

جوجل ومن خلال الواجهة البرمجية كذلك تساعدنا في بناء إضافات أو مُلحقات وعمل نظام شراء أو حتى اشتراك شهري أو سنوي للمُلحق، من خلال تحديد وسيلة الدفع التي يوفرها Chrome Web Store كما هو موضح في الصورة التالية:

My First Extension - Change payment

Change payment

Please set the pricing for your item carefully. You can make only a few changes to the pricing rules once your item has been published. [Learn more about pricing](#)

- This item is free**
Users never need to pay for any part of your application or users get the application for free but can purchase additional goods, content, services, or functionality (virtual items, premium articles, and so on) within your application.
- This item uses Chrome Web Store Payments**

subscription charged per month ▾
one-time payment
subscription charged per month
subscription charged per year

ial mode (recommended).
limited functionality before purchasing it.

[Learn more about free trial applications](#)
- This item uses a custom payment solution**
Users pay for your item using a payment system that is not the Chrome Web Store's built-in payment solution.

Note: When changing pricing models for a published app, your app should continue to support existing users. Ex: If you change your app from free to paid, users with previously acquired free licenses should continue to be able to use the app as they did before.

Cancel OK

صورة رقم ٥.١.١ لتحديد نظام الاشتراك بالإضافة أو المُلحق

لذلك، لا عليك أن تقلق كثيراً من هذه الأمور المالية، ولا داعي لتوفير استضافة خاصة لذلك، لكن بالطبع هناك عدة قيود ومن أهمها الدول المسموح بها لبيع المُلحقات.

هذه هي القائمة التي تسمح بها جوجل لاستخدام نظام الدفع الإلكتروني التي تقدمه:

Who can use Chrome Web Store Payments

If you're in one of the following regions, you can sell apps using Chrome Web Store Payments:

• Argentina	France	Japan	Russia
• Australia	Germany	Malaysia	Singapore
• Austria	Hong Kong	Mexico	Spain
• Belgium	Hungary	Netherlands	Sweden
• Brazil	India	New Zealand	Switzerland
• Canada	Indonesia	Norway	Taiwan
• Czech Republic	Ireland	Philippines	Thailand
• Denmark	Israel	Poland	United Kingdom
• Finland	Italy	Portugal	United States

You can sell to buyers in any of the above regions, and to the following regions: United Arab Emirates, Saudi Arabia, Morocco, Egypt, and the Ukraine. In the future, we expect to support sales to the following additional regions: Argentina, India, Mexico, and Poland. Buyers in unsupported regions might be able to purchase apps, but they can't pay in their local currency and might have to pay international transaction fees.

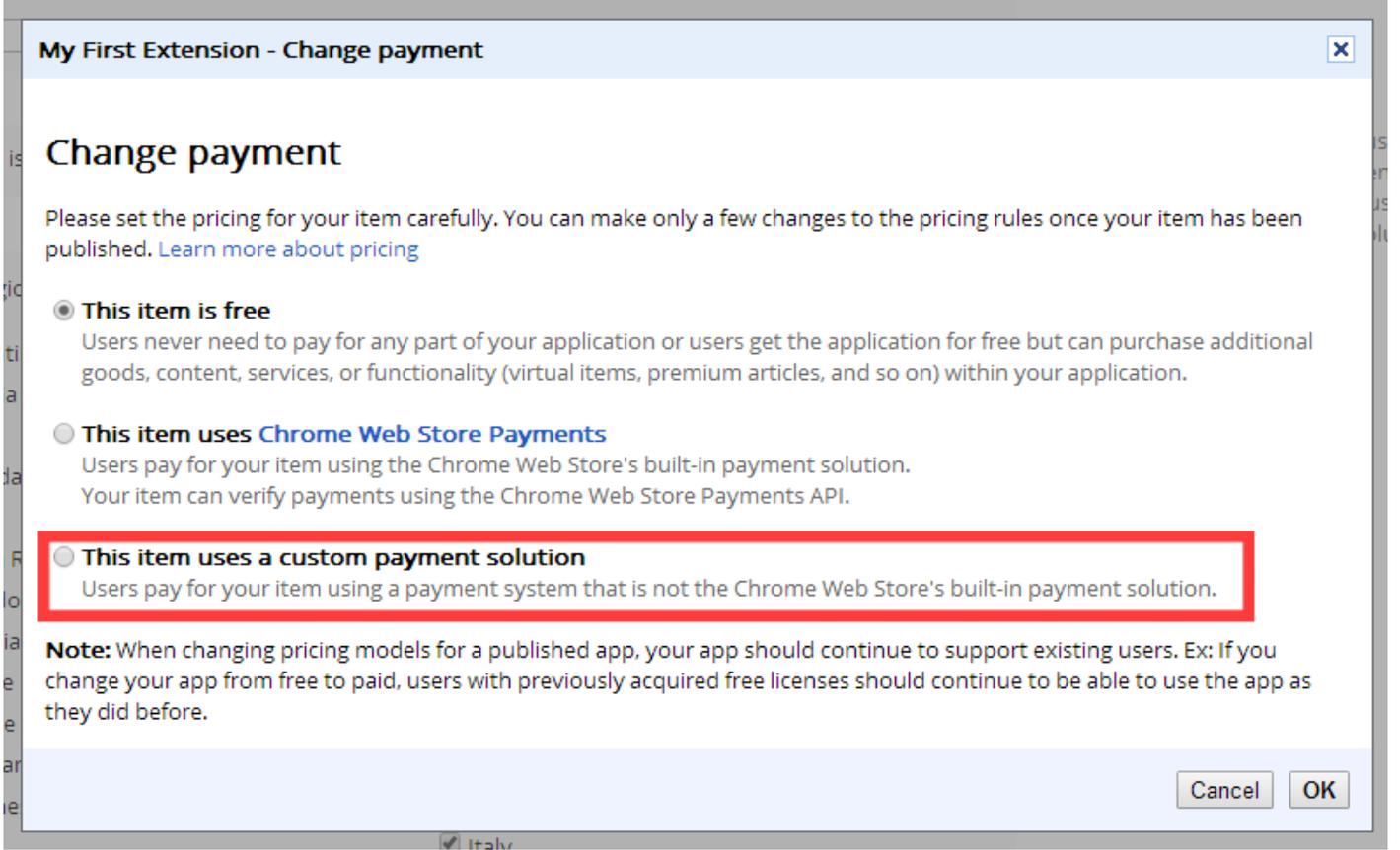
صورة رقم ٥.١.٢ تحدد الدول المدعومة التي تستطيع بيع ملحقاتك والدول التي تستطيع شراء الملحقات

يوضح هذا المصدر الدول المدعومة لاستخدام واجهة برمجية جوجل في البيع والشراء من السوق، حتى تستطيع بيع ملحقاتك يجب أن تكون ضمن الدول الموجودة داخل المُستطيل الأحمر، أما الدول التي تستطيع الشراء فهي الدول الموجودة داخل المُستطيل الأحمر بالإضافة للإمارات العربية المتحدة، المملكة العربية السعودية، المغرب، مصر، وأوكرانيا... وسيتم دعم بعض الدول الأخرى قريباً.

نستطيع أن نستنتج بأن العالم العربي لا يستطيع استخدام وسائل البيع الذي يوفره سوق كروم، لكن يستطيع العديد من المُستخدمين من بعض الدول العربية الشراء من داخل السوق.

إذا كنت تسكن بإحدى الدول المُتاحة لبيع الملحقات، إذناً أهلاً بك مع واجهة كروم البرمجية والتعامل مع الدفعات وتأكيد الاشتراكات وما إلى آخره، ويجدر الإشارة أن جوجل ستقتطع ٥% مما تبيعه من ملحقات كونها وفرت لك البيئة المناسبة والواجهة البرمجية المناسبة للتعامل مع نظام الدفع الخاص بهم.

في حال كنت تسكن خارج الدول المسموح بها للتعامل مع وسائل البيع التي يوفرها سوق كروم، بإمكانك إذاً التوجه للخيار الأخير ببناء نظامك الخاص في بيع مُلحقاتك:



صورة رقم ٥.١.٣ توضح اختيار نظام دفع وشراء خاص بك

أنت لست مُقيداً يا صديقي باستخدام نظام البيع والاشتراك الذي تقدمه جوجل لك، لكن بالطبع سيُسهل عليك الكثير والكثير. لو كنت تملك متجر إلكتروني الخاص بك لبيع مُلحقاتك، بالطبع جوجل تسمح بذلك! لكن عليك بناء واجهتك البرمجية للتأكد من عمليات الشراء، الاشتراك الشهري أو السنوي، وهو مكلف وقتاً وجهداً لك.

مثلاً، أحد العملاء قام ببناء متجره الإلكتروني لبيع مُلحقاته، وكانت وسيلة الدفع المتاحة من المُستخدمين هو **بي بال**. يوجد العديد من نُظم **إدارة المحتوى كوووردبريس** والعديد من المُلحقات لتحويل موقعك الإلكتروني لمتجر إلكتروني وليس مجرد محتوى! البدائل موجودة وعديدة.

٥.٢ البيع من داخل الإضافة In-App Payments

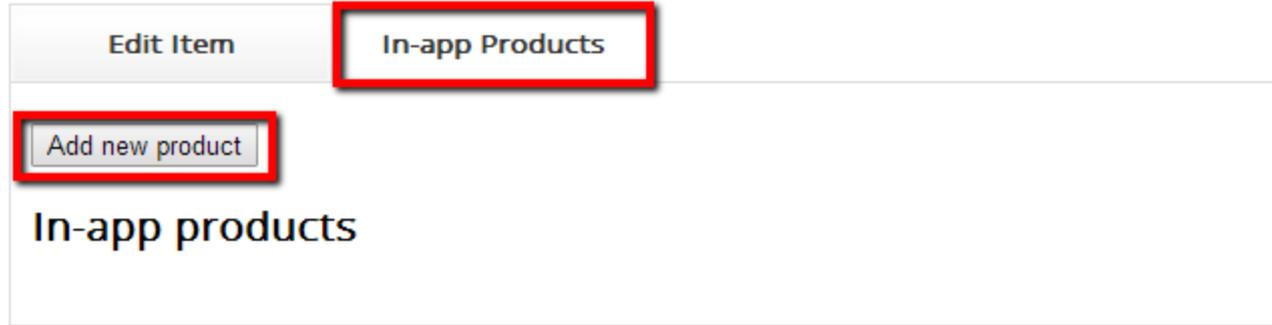
من الممكن القيام بعمليات بيع داخل الإضافة أو المُلحق وهو ما يُسمى In-App Payments، وهو مصطلح نسمع به كثيراً في تطبيقات الهواتف الذكية والأجهزة اللوحية، حيث يوفر هذا التطبيق على سبيل المثال شراء أدوات وأنت داخل التطبيق، أو لعبة ما وتريد شراء أسلحة ومعدات للشخصية التي تلعب بها.

جوجل توفر ذلك أيضاً من خلال واجهتها البرمجية، وهنا كافة التفاصيل البرمجية للقيام بذلك. بالطبع هذا يشمل القيود التي تحدثنا فيها بخصوص قيود الدول المتاحة لاستخدام البيع والشراء ومنصة جوجل لذلك.

كيف يمكن تطبيق مبيعات من داخل الإضافة In-App Purchases:

- أثناء دخولك لصفحة الإضافة من صفحة المطورين، اذهب الى خيار In-app Products:

Developer dashboard > My First Extension



صورة رقم ٥.٢.١ لعرض بعض المنتجات للبيع من داخل إضافتك

- قم بالضغط على Add new product ويمكنك إضافة أكثر من منتج لبيعه وأنت داخل الإضافة ذاتها.

- قم بإعطاء تفاصيل المنتج التي تود بيعه داخل الإضافة بهذه التفاصيل:

Developer dashboard > My First Extension > In-app Product

SAVE Inactive

Product ID *
0 of 100 characters

Product Type *
one-time subscription

English - en (default)

Title *
English - en
0 of 50 characters

Description *
English - en
0 of 100 characters

Price tier *
Tier 0

Local prices

United States	USD	0.99			
Argentina	ARS	6.00	includes tax	Australia	AUD 0.99 includes tax
Austria	EUR	0.74	includes tax	Belgium	EUR 0.74 includes tax
Brazil	USD	0.99		Bulgaria	BGN 1.49 includes tax

صورة رقم ٥.٢.٢ لإدخال بيانات المنتج الذي نود بيعه داخل الإضافة

ويمكن تحديد كذلك أن هذا المنتج يُدفع لمرة واحدة أو بنظام الاشتراك (الشهري أو السنوي).

عند زيارة المُستخدم لصفحة إضافتك على سوق كروم، يستطيع أن يلاحظ عبارة

" Offers in-app purchases " كما هو مُشار بالصورة التالية:

My First Extension
offered by Dev. Mohammad M. AlBanna
★★★★★ (0) | [Extensions](#) | **Offers in-app purchases**

OVERVIEW | REVIEWS | SUPPORT | RELATED

Compatible with your device

Description

Report Abuse
Version: 1.0.0
Updated: December 30, 2015
Size: 39.39KiB

The developer has not uploaded any screenshots.

صورة رقم ٥.٢.٣ توضح للمُستخدم بأن هذه الإضافة توفر منتجات داخل الإضافة يمكن شراؤها

٥.٣ الإعلانات وإعلانات Google AdSense

كما تجري العادة، لا يُحب المُستخدمين الدفع لقاء المُلحقات أو حتى تطبيقات الهواتف الذكية في العديد من الأحيان، لذلك يلجىء المطورون لاستخدام وسائل أخرى للربح مقابل جهدهم في تطوير خدمات لعموم المُستخدمين.

يلجىء العديد لتطبيق نظام إعلانات وقد يتجه العديد من المطورين لاستخدام [خدمة جوجل للإعلانات Google AdSense](#) في مواقعهم الإلكترونية، لكن... هل يصلح ذلك في مُلحقات جوجل؟

قد لا يروق ذلك لك عزيزي المطور، وأعتذر لنفسي ولك أيضاً بأن هذه الوسيلة غير مُتاحة بالمُطلق وقد تعرّضك لحظر حسابك لو حاولت الإلتفاف حول الموضوع!

فطبقاً لسياسة الاستخدام الخاصة في تطوير إضافات المُتصفح وكذلك سياسة الاستخدام في خدمة Google AdSense فلا يمكننا استخدامها في مُلحقاتنا، بل والواجهة البرمجية الخاصة بمتصفح كروم لا تدعم ذلك! هو ليس مجرد نفس الكود الذي نقوم بوضعه في مواقعنا الإلكترونية يمكن وضعه كذلك في مُلحقاتنا!

Currently, AdSense may not be used to serve ads in Chrome Extensions or Chrome Apps, per [AdSense policies](#).

صورة رقم ٥.٣.١ توضح عدم إمكانية استخدام خدمة إعلانات جوجل في إضافات متصفح كروم

لكن بالطبع، هناك العديد من الخدمات الأخرى أو ما تُسمى بأطراف ثالثة توفر واجهة برمجية أخرى لعرض إعلاناتهم داخل مُلحقاتك. لا يوجد لديّ مصادر موثوقة حول الأمر كوني لم أتعامل مع أيّ من هذه الخدمات، فتجربتها يقع على عاتقك أنت.

٥.٣ وسائل أخرى للربح؟

حسناً، يوجد العديد من أساليب الربح الأخرى، كبيع ملكية إضافتك لشركة ما وهو ما تسمح به جوجل، أو البدء بتطوير إضافات أخرى لعملائك بمقابل مادي، أو الإشراف على تطوير مُلحقات أخرى.

من الممكن كذلك عمل نظام إعلانات خاص بك يسمح لمواقع أخرى بنشر روابط أو مواضيع داخل إضافتك التي قمت بتطويرها، مع الأخذ بعين الاعتبار كافة الشروط والمعايير التي قدّمتها جوجل لك في سياسة الاستخدام.

هناك العديد من المطورين يجنون آلاف الدولارات فقط في تطوير إضافات المتصفح وتحديداً متصفح كروم لما يقدمه من سهولة في التوثيق وسهولة في التعامل مع الواجهة البرمجية لمتصفح كروم والتطوير الدائم عليها، وهناك من يبدع في ربط الخدمات بين إضافات متصفح كروم مع الهواتف الذكية وهو أمر ممكن وليس بمستحيل، فنحن نتحدث عن تقنيات ويب ويمكن استخدامها داخل مُلحقاتنا.

أسأل الله العليّ القدير أن أكون قد وُفقت في هذا الكتاب الإلكتروني بإصداره الأول، وسأسعى لتطويره إن جدّ أي جديد ليحتوي ما توفّره جوجل في إطار تطوير مُلحقات وإضافات متصفح كروم، ولا تنسوني من صالح دعاكم لي ولوالديّ وسائر المُسلمين من بقاع الأرض.

الحقوق

- تعود ملكية هذا الكتاب الإلكتروني لمؤلفه "محمد مفيد البنا" وكافة الحقوق محفوظة له، لا يحق لك نشر الكتاب دون أخذ إذن من مؤلفه.
- حقوق نقل المحتوى التابع للمصادر الخارجية الموجودة في الكتاب الإلكتروني فهي تابعة لكل مصدر على حدة.
- حقوق صورة غلاف هذا الكتاب تعود إلى موقع Freepik.com