

In order to publish, deploy and update a Chrome extension outside the Google Chrome Store you have to follow the guide below.

1. Architecture

Firstable it is necessary to define the CRX and XML names and the url where they would be deployed.

For what concerns this example:

- the CRX name would be `myCRX.crx` and it would be deployed at the url `https://my.server/resources/myCRX.crx`
- the XML name would be `myXML.xml` and it would be deployed at the url `https://my.server/resources/myXML.xml`

2. JSON Manifest

As specified in [this](#) link the `update_url` (where the update XML could be found) must be contained in the JSON manifest: in order to do that it is necessary to insert the following line into the JSON file.

```
{
  [...],
  "update_url": "https://my.server/resources/myXML.xml",
  [...]
```

3. CRX and private key creation

The creation of the CRX and the PEM can be performed following [this](#) guide.

At the end of the process two files would be created (a CRX and a PEM): after that it is necessary to rename them to `myCRX.crx` and `myPEM.pem` , respectively.

4. Public key extraction

The extraction of the public key can be performed by executing the following command from the folder where the PEM is located:

```
openssl.exe rsa -in myPEM.pem -pubout > myPEM_pub.pem
```

After that a new file containing the public key (named `myPEM_pub.pem`) would be created.

5. Extension ID extraction

The Extension ID extraction can be performed following [this](#) guide.

In particular, it is necessary to download the `extension_id.py` file, copy it in the folder containing the `myPEM_pub.pem` file and execute, from the same folder, the following command:

```
python extension_id.py myPEM_pub.pem
```

Note: if you're using Python 3 you have to modify the line 94 of the `extension_id.py` file from

```
with file(first_arg) as f:
```

to

```
with open(first_arg) as f:
```

The command output would be the following one:

```
[...]
Extension ID: <myExtensionID>
[...]
```

6. XML file creation

In order to properly deploy the extension it is necessary to create the update XML file (named, in this example, `myXML.xml`).

In this case its content would be:

```
<?xml version='1.0' encoding='UTF-8'?>
<gupdate xmlns='http://www.google.com/update2/response' protocol='2.0'>
  <app appid='<myExtensionID>'>
    <updatecheck codebase='https://my.server/resources/myCRX.crx'
version='<myExtensionVersion>' />
  </app>
</gupdate>
```

Note: it is necessary to replace the following entries with the right values

- `<myExtensionID>` , *output of paragraph 5*
- `https://my.server/resources/myCRX.crx` , *defined in paragraph 1*
- `<myExtensionVersion>` , *defined in the manifest file*

7. Publishing

In order to properly configure the environment it is necessary to publish the CRX created in paragraph 3 and the XML created in paragraph 6 at the urls defined in paragraph 1.

8. Deployment

The deployment of the extension can be performed following [this](#) guide.

In particular, it is necessary to add to the list of the force installed app and extension the following line:

```
<myExtensionID>;https://my.server/resources/myXML.xml
```

Note: it is necessary to replace the following entries with the right values

- `<myExtensionID>` , *output of paragraph 5*
- `https://my.server/resources/myCRX.crx` , *defined in paragraph 1*

9. Update - CRX creation

In order to update the extension it is necessary to create a new CRX package, with an updated version number (in this example `<myNewExtensionVersion>`).

The CRX creation can be performed following [this](#) guide.

Note: in order to make this process work it is necessary to select the key `myPEM.pem`, created in paragraph 3.

10. Update - Publishing

In order to publish the update it is necessary to rename the CRX created in paragraph 9 to `myCRX.crx` and to modify the version number in the XML created in paragraph 6 (see below).

```
<?xml version='1.0' encoding='UTF-8'?>
<gupdate xmlns='http://www.google.com/update2/response' protocol='2.0'>
  <app appid='<myExtensionID>'>
    <updatecheck codebase='https://my.server/resources/myCRX.crx'
version='<myNewExtensionVersion>' />
  </app>
</gupdate>
```

After that, the last thing that has to be done in order to perform the update is the publishing of the CRX and the XML at the urls defined in paragraph 1.

edited Nov 19 '18 at 14:05

answered Nov 19 '18 at 13:58



Daniele Milani

531 2 5 24