

الملخص:

نود عمل شيء مشابه ل Ionic ، اي شي مشابه لطريقة عمل Ionic بالضبط ولكن مع بايثون ، من أجل جعل عملية البرمجة أسهل وأسرع ، من خلال أن يكون UI هو نفسه مع جميع المنصات ويتم استخدام (Html - CSS - javascript) وتكون في الخلفية تعمل لغة برمجية واحدة وهي بايثون.

ملاحظة: هذا دليل سيكون للعمل البرمجي وليس للعمل الإداري

نقاط القوة:

يوجد مكتبة تدعى eel تمكننا من بناء تطبيق متعدد المنصات للكمبيوتر ، بنفس مفهوم المذكور اعلاه ، وكل هذا دون أن نكتب رمز برمجي واحد ؛ إنها ممتازة ولكن لدي بعض الملاحظات عليها ، إذن كل ما علينا فعله هو التعديل على الرمز البرمجي المصدر للمكتبة وجعلها أفضل وأكثر فائدة وأكثر خصائص.

أهم التحديات:

- إنشاء مكتبة فرعية تحتوي على أدوات ودوال تراعي الاختلافات بين أنظمة التشغيل ، من أجل إتمام التوافق التام واكمال الخصائص وهي بالضبط تشابه plugins الخاصة ب Ionic.
- مثل: دالة إنشاء الإشعارات تختلف من Windows أو linux أو Mac أو IOS أو Android.
- ومثل: في Android يوجد شيء يسمى Tost message أما في IOS لا يوجد.

طبعا يكون هذا في آخر مرحلة أو حتى بعد النشر ويمكن أن نتركه للمجتمع.

- إنشاء محول امتدادات لانظمة التشغيل المختلفة ، شيء مثل pyinstaller ، ولكن يقوم بتحويل الكود الخاص بنا ل (exe , mac , linux , apk , ios). وسيكون هناك توليد كود تلقائي

- إنشاء حزمة pypi لكي نتمكن من تثبيتها من خلال pip.

- توفير document على pypi و github.

- إنشاء امتداد للمشروع على github.

يقسم المشروع لمرحلتين:

1. cross platform desktops apps
2. cross platform mobile apps

:cross platform desktops apps


من المفترض أن يكون العمل هنا هو الأسهل ، لأننا فقط ، سنتعرف على كيفية استخدامها ثم سنطلع الرمز المصدري لمكتبة eel ونجري بعض التعديلات عليها.

الخصائص:

على أقل تقدير:

- حرية كتابة host
- عرض التطبيق على متصفح بدون نافذة

على تقديرات أفضل:

- قفل زر maximum minimum
- قفل resize في النافذة
- نافذة بدون الازرار الثلاث أعلاه  ، لأنها تعطي حرية أكبر في التصميم
- قد نقوم باستخدام dot net core لإنشاء نافذة عرض webview
- قد نقوم باستخدام dot net core من أجل الخصائص الإضافية plugins
- قد نقوم باستخدام dot net core و dot net من أجل تخصيص النافذة

ملاحظة: يفضل عدم استخدام لغات غير بايثون ، ولا نستخدم غيره إلا عندما نكون متطرين.

cross platform mobile apps:

هنا سنقوم باستخدام kivy ، عمل كود تطبيق ل Android و IOS ونقوم بعرض webview الذي يكون فيه رابط eel ، من أصعب المراحل هو عمل webview ل IOS وايضا تحتاج هاتف Iphone او محاكي لأجل التجربة.

الخصائص:

على أقل تقدير:

- خيار عرض أو إخفاء شريط الإشعارات

على تقديرات أفضل:

- خيار عرض التطبيق على متصفح بدون webview

مرحلة كتابة ال package:

بعد الإنتهاء من التجارب والمعايرة في الكود ، نقوم بجمعها تحت مكتبة واحدة ك package.

مرحلة إنشاء محول الإمتدادات:

بعد انتهاء من المكتبة يتم بناء محول الامتدادات ك حزمة اخرى او برنامج متفرد عن المكتبة ، وهو مزيج من كل أدوات التحويل مثل pyinstaller وغيرها ، يتم استخدام احدهما حسب الاحتياج، يحوي

خصائص التحويل المتوفرة في أغلب الأدوات (icon التطبيق - رقم الإصدار ...) ، وسنحاول تمكين التحويل خارج بيئة الحالية (عيب pyinstaller) مثل التحويل ل exe ضمن لينكس ، والتنصيب التلقائي للتطبيق على المحاكي Memu و خيار اختيار الهاتف الذي يود تنصيبته عليه.

الخصائص:

- على أقل تقدير:
- برنامج سكربت
- على تقديرات أفضل:
- برنامج ب واجهة رسومية رائعة

خطوات العمل:

1. التجريب والتعرف على eel و kivy

eel

- ☐ حرية كتابة host
- ☐ عرض التطبيق على متصفح بدون نافذة
- ☐ انبثاق المتصفح
- ☐ دون انبثاق المتصفح
- ☐ نافذة بدون الازرار الثلاث أعلاه
- ☐ قفل زر maximum minimum
- ☐ قفل resize في النافذة

2. بناء مكتبة package

3. بناء محول الامتدادات

4. عمل مشروع setup للمشروع

5. رفع المشروع على github و pypi

مشروع فرعي بعد الإنتهاء: