

در مطلب « [نحوه مشارکت در پروژه‌های GitHub به کمک Visual Studio](#) » با مفهوم pull request آشنا شدیم. اما ... یک pull request خوب چه خصوصیتی دارد و فرهنگ ارسال یک PR خوب چیست؟

اخلاق مشارکت در یک پروژه سورس باز

بعضی از توسعه دهنده‌ها در حین مشارکت در یک پروژه سورس باز، برای مثال جهت افزودن قابلیت جدید و یا رفع مشکلی، ابتدا سعی می‌کنند تا کدهای فعلی را برای خودشان «قابل فهم‌تر» کنند. این قابل فهم‌تر کردن پروژه، شامل تغییر نام متغیرها و متدهای فعلی، انتقال کدهای موجود به فایل‌هایی دیگر یا حتی یکی کردن چندین فایل با هم، مرتب سازی متدهای یک کلاس بر اساس حروف الفباء و امثال آن می‌شود. این کارها را نباید در حین مشارکت و توسعه‌ی پروژه‌های سورس باز دیگران انجام دهید! اگر هدفتان رفع مشکلی است یا افزودن قابلیت جدید، باید نحوه‌ی کدنویسی فعلی را حفظ کنید. از این جهت که نگهدارنده‌ی اصلی پروژه، پیش از شما این کار را شروع کرده‌است و زمانیکه شما به پروژه‌ای دیگر رجوع خواهید کرد، باز نیز باید همین کار را ادامه دهید. اگر refactoring گسترده‌ی شما به هر نحوی سبب بهبود پروژه‌ی اصلی می‌شود، ابتدا این مورد را با مسئول اصلی پروژه مطرح کنید. اگر او قبول کرد، سپس اقدام به چنین کاری نمائید.

بحث در مورد تغییرات پیش از ارسال PR

قبل از اینکه PR ایی را ارسال کنید، بهتر است یک issue یا ticket جدید را باز کرده و در مورد آن بحث کنید یا توضیح دهید. در این حالت ممکن است توضیحات بهتری را در مورد سازگار سازی تغییرات خود با کدهای فعلی دریافت کنید.

Pull request ها را کوچک نگاه دارید

برای اینکه شانس قبول شدن PR خود را بالا ببرید، حجم و تمرکز آن را کوچک نگه دارید. بسیاری از توسعه دهنده‌های سورس باز اگر با یک PR حجم روبرو شوند، آن را رد می‌کنند چون مشکل اصلی، مدت زمان بالایی است که باید جهت بررسی این PR اختصاص داد. هرچقدر حجم آن بیشتر باشد، زمان بیشتری را خواهد برد.

فقط یک کار را انجام دهید

شبهه به اصل تک مسئولیتی کلاس‌ها، یک PR نیز باید تنها یک کار را انجام دهد و بر روی یک موضوع خاص تمرکز داشته باشد. فرض کنید PR ایی را ارسال کرده‌اید که سه مشکل A، B و C را برطرف می‌کند. از دیدگاه مسئول اصلی پروژه، موارد A و C قابل قبول هستند؛ اما نه مورد C مطرح شده. در این حالت کل PR شما برگشت خواهد خورد. به همین جهت بهتر است بجای یک PR، سه PR مختلف و مجزا را جهت رفع مشکلات A، B و C ارسال کنید.

سازگاری تغییرات ارسالی را بررسی کنید

حداقل کاری را که پیش از ارسال PR باید انجام دهید این است که بررسی کنید آیا این تغییرات قابل Build هستند یا خیر. همچنین اگر پروژه دارای یک سری Unit tests است، حتماً آن‌ها را یکبار بررسی کنید تا مطمئن شوید جای دیگری را به هم نریخته‌اید. ضمناً وجود این تست‌ها به صورت ضمنی به این معنا است که تغییرات جدید شما نیز باید به همراه تست‌های مرتبطی باشند تا پذیرفته شوند.

PR ایی را بر روی شاخه‌ی master ارسال نکنید

پس از اینکه یک fork از پروژه‌ای سورس باز را ایجاد کردید و سپس آنرا clone نمودید تا به صورت Local بتوانید با آن کار کنید، فراموش نکنید که در همینجا باید یک branch و انشعاب جدید را جهت کار بر روی ویژگی مدنظر خود ایجاد کنید (برای مثال feature-X, fix-Y). بسیاری از پروژه‌های سورس باز به هیچ عنوان PRهای کار شده‌ی بر روی انشعاب master را قبول نمی‌کنند.

برای مطالعه بیشتر

[Open Source Contribution Etiquette](#)

[ten tips for better Pull Requests](#)

[Getting a Pull Request Accepted](#)

[Optimize Your Pull-request](#)

نظرات خوانندگان

نویسنده: جلال

تاریخ: ۱۳۹۳/۱۲/۱۴ ۱۹:۱۱

کاش زودتر خونده بودمش P: