بسم الله الرحمن الرحیم



**سید امیرمهدی میرشریفی**

**9831105**

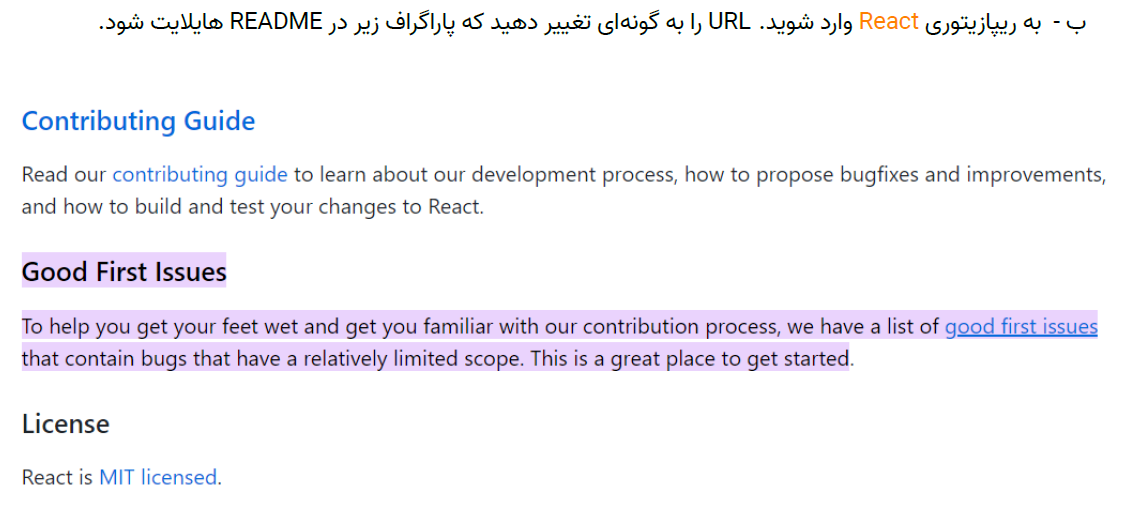
**پاسخنامه تکلیف یک درس طراحی وب**

**استاد علیزاده**



پاسخ:

با استفاده از لینک : <https://www.digikala.com/#sScdOJOzhFxtcEqkjP7P> میتوان به قسمت مورد نظر انتقال یافت . بعد از علامت # آی دی عکس مورد نظر قرار دارد تا مرورگر با استفاده از آن به المنت مورد نظر در صفحه برسد.



پاسخ:

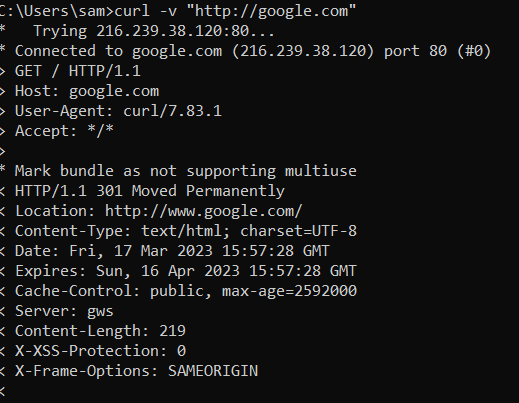
با کپی کردن لینک زیر میتوان به تصویر بالا دست یافت. همانطور که در لینک مشاهده می کنید پس از علامت شارپ آی دی المنت مورد نظر ( good first issues) را از inspect به دست آورده و اضافه میکنیم و سپس با استفاده از :~:text=first , last بازه متنی که می خواهیم هایلایت شود را مشخص کنیم.

<https://github.com/facebook/react/#user-content-good-first-issues:~:text=Good%20First%20Issues,to%20get%20started>

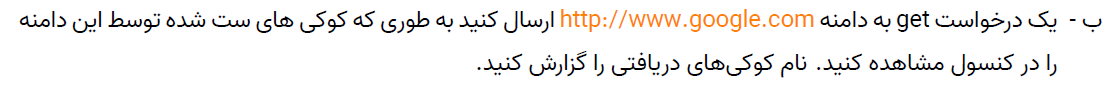


پاسخ:

تصویر زیر دستور مربوطه و نتیجه را نمایش میدهد که بخش اول آن هدرهای ریکوئیست و بخش دوم هدر های ریسپانس است:

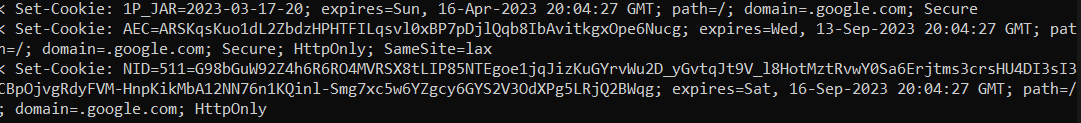


همانطور که در بخش هدرهای ریسپانس مشاهده میشود ، وضعیت با کد 301 مشخص شده است که بدین معنا است که منبع درخواستی به محل دیگری مشخص شده است که آدرس جدید آن را میتوان در Location مشاهده کرد.



پاسخ: در ادامه لیستی از کوکی های دریافتی ارائه میشود. همچنین در مقابل هرکدام در صورت نیاز لینکی قرار می گیرد که توضیح مختصری را از آنها ارائه می دهد.

* **1P\_JAR (**[**https://replyify.com/cookie-policy.html#:~:text=Purpose-,1P\_JAR,-google.com**](https://replyify.com/cookie-policy.html#:~:text=Purpose-,1P_JAR,-google.com)**)**
* **Expires**
* **Domain**
* **Samesite(**[**https://developer.mozilla.org/en-US/docs/Web/HTTP/Headers/Set-Cookie/SameSite#:~:text=See%20also-,SameSite%20cookies,-Secure%20context%3A**](https://developer.mozilla.org/en-US/docs/Web/HTTP/Headers/Set-Cookie/SameSite#:~:text=See%20also-,SameSite%20cookies,-Secure%20context%3A)**)**
* **NID (**[**https://cookiedatabase.org/cookie/google-ads-optimization/nid/#:~:text=With%20Cookiedatabase.Org-,NID,-Summary**](https://cookiedatabase.org/cookie/google-ads-optimization/nid/#:~:text=With%20Cookiedatabase.Org-,NID,-Summary)**)**

****

پاسخ:

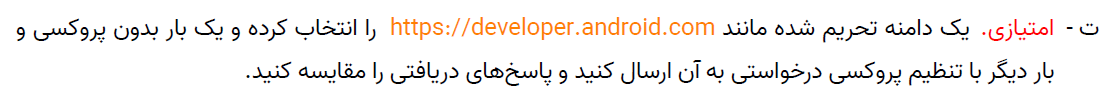
برای ارسال فایل جیسون باید به شکل زیر عمل کنیم:

>curl -X POST <https://reqres.in/api/users>

-H "Content-type:application/json"

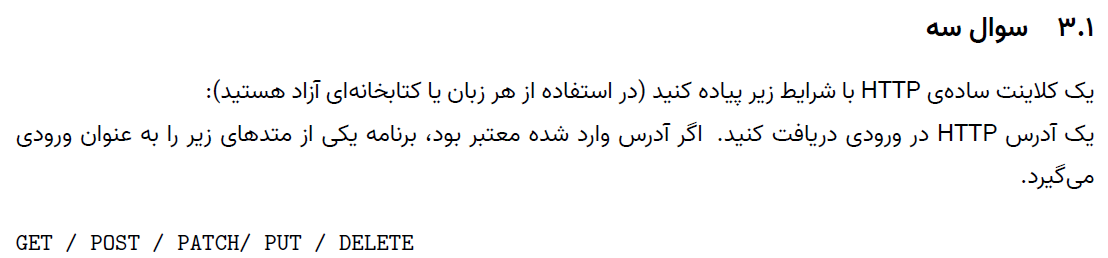
-d "{\"first\_name\":\"amirmahdi\",\"last\_name\":\"mirsharifi\"}"

و پاسخ دریافت شده به شکل زیر است:

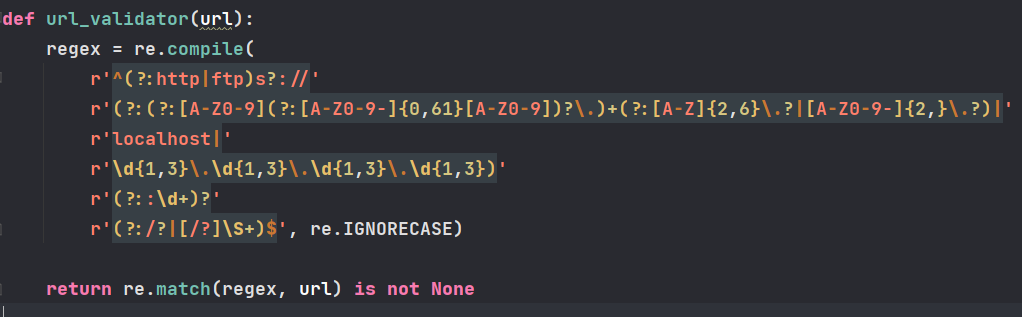
{"first\_name":"amirmahdi","last\_name":"mirsharifi","id":"917","createdAt":"2023-03-17T20:48:26.275Z"}

پاسخ: در ابتدا با استفاده از دستور کرل که به این دامنه پیام بزنیم با وضعیت 403 که به معنای ممنوع است مواجه میشویم. در این زمان اگر دستور : curl [https://developer.android.com](https://developer.android.com/) - - proxy [fodev.org:8118](http://fodev.org:8118/)

که –proxy نشانگر استفاده از پراکسی سرور و دامنه و پورتی که در ادامه آمده است نشانگر سرور مدنظر است استفاده کنیم وضعیت 200 را دریافت میکنیم. زیرا در مرحله قبلی دامنه تحریم شده بود و به آیدی ما خدمات نمیداد اما پراکسی این خدمات را در اختیار ما قرار داد.

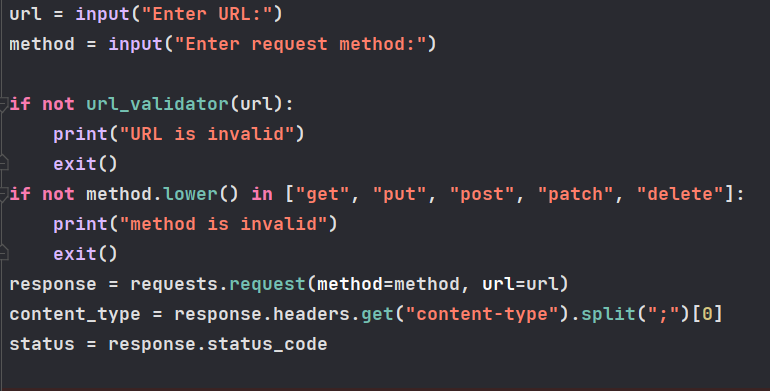


پاسخ:



با تابع url\_validator میتوان کد داده شده را اعتبار سنجی کرد تا در موقع ارسال درخواست با ارور روبه رو نشویم. در ادامه url & method را از کاربر میگیریم و اگر url or method نامعتبر باشد آنگاه از برنامه خارج میشویم.

سپس اطلاعات لازم مثل نوع فایل دریافتی و کد و وضعیت را استخراج میکنیم.



در آخر با توجه به حالات مختلف اعمال مختلف را اعمال میکنیم.برای مثال تفاوت در ذخیره سازی انواع فایل ها یا پیامی که در مواجه با ارور های مختلف نشان میدهد.همچنین ارور ها صرفا بر اساس نوع که همان عدد شروع شونده آن است تخفیف میشوند یعنی :2xx,3xx,4xx,5xx



۲ بخش Git

بخش اول – amend

برای اصلاح کامیت آخر و اضافه کردن فایل bar.txt

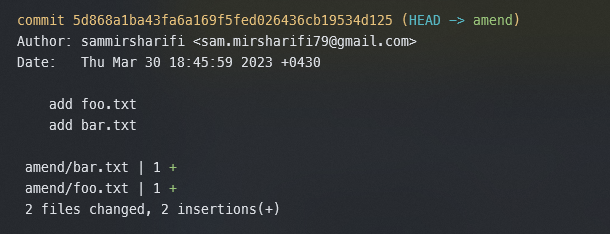
با استفاده از فرمان

فایل bar.txt را stage کرده و سپس کامیت آخر (به همراه پیام آن) را با استفاده از فرمان

اصلاح و آپدیت می‌کنیم.

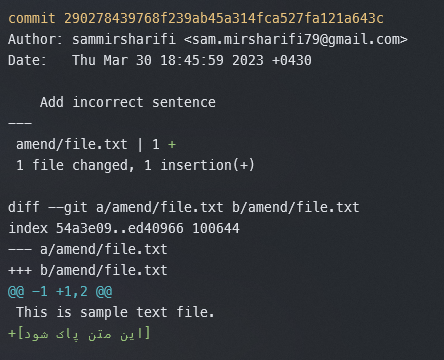
با استفاده از

می‌توان مشاهده کرد که فایل bar.txt به کامیت اضافه شده است. (پیام کامیت نیز قابل مشاهده خواهد بود)



برای اصلاح کامیت قبل‌تر با استفاده از فرمان

و طبق تصویر پایین می‌بینیم که در کامیت محتوای اشتباهی وارد فایل شده که لازم است آنرا پاک کنیم.



برای اینکار از به شکل تعاملی (interactive) استفاده می‌کنیم.

فرمان

را اجرا کرده و rebase را از گره پدر این کامیت آغاز می‌کنیم.

پس از اجرای فرمان در ادیتور، فرمانِ اولین کامیت را از pick به edit تغییر می‌دهیم و ذخیره کرده و از ادیتور خارج می‌شویم. با اینکار عملیات rebase به شکل ناتمام (سر کامیتی که edit قرار دادیم) باقی‌مانده تا فرمان rebase --continue اجرا شود.

در این لحظه آخرین کامیت همان کامیت اشتباه می‌باشد بنابراین مانند بخش اول ابتدا فایل‌های اصلاح شده را وارد مرحله staging کرده و سپس amend می‌کنیم.

به ترتیب

* edit file.txt
* با توجه به اینکه با حذف خط اشتباه، دیگر کامیت خالی از تغییرات خواهد بود، برای مجاز کردن این فرایند فلگ --allow-empty اضافه شده است
* با اینکار عملیات rebase ادامه می‌یابد ولی در ادامه و در کامیت‌های بعدی کانفلیکت ایجاد می‌شود و نیاز است کانفلیکت را برطرف کرده و دوباره   
  rebse --continue را اجرا کنیم.
* در نهایت با استفاده از git push -f origin amend تغییرات را روی ریپازیتوری خود قرار می‌دهیم.

استفاده از -f یا --force به این علت است که پس از rebase برنچ ریموت قابلیت fast-forward به کامیت آخر را نخواهد داشت (مثلا در صورتی که کامیت بخش قبلی را push کرده باشیم)

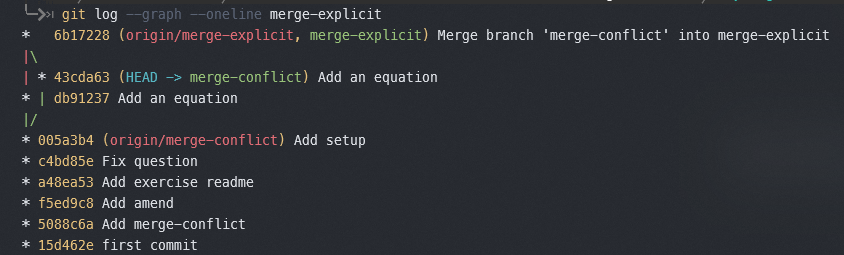
با توجه به اینکه گفته می‌شود بعد از تغییرات ابتدا git commit کنیم و سپس   
rebase --continue را اجرا کنیم، در این صورت به نظر کامیت خالی حذف می‌شود. برای جلوگیری از این اتفاق، بدون git commit اینکار را انجام می‌دهیم و git rebase خود این کامیت را انجام می‌دهد. به نظر این اتفاق مربوط به حالت interactive می‌باشد.

در نهایت از روش دوم استفاده شد و پس از stage کردن تغییرات مستقیما   
rebase --continue صدا زده شد.

۲.۳ بخش دوم – conflict merge

برای explicit-merge از وارد برنچ merge-explicit شده و سپس فرمان

را اجرا می‌کنیم، conflictها را برطرف کرده و سپس کامیت می‌کنیم. درخت برنچ به شکل زیر می‌شود.



همانطور که می‌بینیم برای مرج، یک کامیت جدید ایجاد شده است.

حال برای implicit-merge وارد برنچ merge-implicit شده و سپس فرمان

را اجرا می‌کنیم. کانفلیکت‌ها را برطرف کرده، stage و

را اجرا می‌کنیم. درخت به شکل زیر خواهد بود که یعنی کامیت‌های برنچ ما در ادامه کامیت‌های merge-conflict اضافه شده‌اند (merge-conflict به عنوان base برنچ فعلی قرار گرفته است).

