NO BODY'S BUSINESS BUT MINE: Vol. 2

May 2025

privacyinternational.org

Table of Contents

1. Introduction

Many menstruating individuals today use period-tracking apps as a convenient digital tool on their smart phone to track their menstruation. In the UK, researchers have found that more women are using smartphone apps to track their menstrual cycles for fertility-tracking purposes. In the U.S., the overturning of Roe v. Wade has slightly shifted the tides, with some users seeking out privacy-enhancing period tracking apps over the mainstream apps and other users outright deleting their period tracking apps over privacy fears. And in the current technological climate, it's worth asking: how responsible with user data *are* these apps?

In 2019, Privacy International (PI) investigated several popular period-tracking apps across the world to examine how they handle users' privacy, particularly the sharing of users' period data with Facebook. We performed a dynamic analysis of ten popular period tracking apps using our data interception environment (DIE), which allowed us to see whether and where these apps were sharing user data. The most popular apps we looked at did not appear to share data with Facebook, but the other apps we examined that still boasted millions of downloads appeared to engage in some extensive sharing of sensitive user cycle data with third parties including Facebook. Our research exposed serious concerns around these apps' compliance with data protection laws, as well as around companies' responsibility and accountability when it comes to third-party data-sharing.

Since then, data protection and privacy regulations have been ramping up, with increased expectations for user privacy protection in the form of regulations like the European Union (EU) Digital Services Act, the AI Act and continued enforcement of the General Data Protection Regulation (GDPR).

However, these privacy regulations have also been coupled with setbacks in the women's health sector, such as the repealing of Roe v. Wade in the U.S. that has put women's sexual and reproductive health data in a more precarious position than ever before. Not to mention in recent years numerous examples of law enforcement using people's online data for investigation purposes, such as U.S. law enforcement using Facebook chat logs to prosecute an abortion seeker in Nebraska or UK law enforcement reportedly obtaining a woman's Google search history and sentencing her for taking abortion pills beyond the legal limit.

Considering these changes over the past several years in the privacy and political landscape, as well as technological changes such as the expansion of cloud-based services and the AI industry, we undertook another technical investigation into how period tracking apps are handling user data five years later and the implications for users' privacy.

As we will expand on below, our research found that, overall, period tracking apps were not sharing users' cycle data as egregiously with third parties as we found for some apps in 2019. Though in the course of our investigation, we did observe several categories of third parties that many apps were integrating for different purposes, such as advertising software development kits (SDKs) or application programming interfaces (APIs) to service certain app functionalities,

and these third parties often processed some degree of the user's personal or device data. The various technical approaches that period tracking apps utilise to service their app warrant scrutiny in a politically volatile realm. In our report, we explore the various technical methods built into period tracking apps, such as integrating third party deployers and storing user data on servers, and we conclude with how these practices raise crucial questions for the future of privacy in the femtech space.

2. Our research

Building on our findings from our previous research, we sought to investigate the data management and sharing practices of menstruation apps with third parties beyond Facebook, as well as to assess whether some of the apps we looked at the first time around had improved their practices as they have claimed.

Methodology

We looked at the top period tracking apps downloaded in the Google Play Store, some of which we had examined in our original research, and some of which are newly emerging apps that have since grown in popularity. The top period-tracking apps with the most downloads included Flo; Period Tracker by Simple Design; and apps we'd tested in our previous research that still exist such as Maya, Period Tracker by GP Apps, as well as several apps popular in other global markets such as WomanLog and Wocute. We also included an app that saw an uptake in downloads post the overturning of Roe v Wade, which has claimed to be 'privacy-enhancing' (https://www.wired.com/story/period-tracking-apps-flo-clue-stardust-ranked-data-privacy/) (Stardust), and finally an open-source period tracker developed by non-profit researchers (Euki).

To conduct our research, we first ran every app through Exodus Privacy for a static analysis. Exodus Privacy is a static analysis tool that allows anyone to check the trackers (e.g., ad trackers) and permissions (e.g., access to precise GPS location) embedded into Android apps. We entered into Exodus a number of popular period tracking apps beyond those listed above, with the goal to narrow down our list and select the apps with the greater number of trackers and permissions enabled. This static analysis allowed us to reach our final list of apps to test: Flo, Period Tracker by Simple Design, Maya by Plackal Tech, Period Tracker by GP Apps, WomanLog, Wocute, Stardust and Euki. Note that we found no trackers and permissions enabled for Euki, but we decided to put Euki to the test anyway for the purposes of demonstration and comparison.

Then, we deployed a two-pronged analysis: firstly, we examined the dynamic web traffic analysis via our in-house Data Interception as a Service (DIAAS) tool; and secondly, we compared these findings with the apps privacy policies.

The DIAAS is a web traffic interception tool that allows us to view the requests and responses sent by apps over the web (note the DIAAS only shows client-server interactions, not server-to-server interactions). When running an app in the DIAAS environment we can view the web traffic of an app, for example the web requests they send to URLs, which can help to reveal any calls to third party deployers. The first step in the research involved running each app through the DIAAS to view the web requests being sent to various third parties as we interacted with the app, from the moment we set up our accounts to logging our period cycle data.

The second step, which we conducted alongside our dynamic analysis, was to compare our web traffic findings (e.g., which third parties appeared) with the information provided in the apps' privacy policies about their data management and third-party processing practices.

Definitions

Before diving into the details of the methodology and our findings, here are some definitions of technical vocabulary we will be using throughout this report:

- Software Development Kit (SDK): a set of software tools provided to developers that can be used for building applications for specific platforms (essentially the building blocks for a software application); SDKs include documentation, APIs (below), libraries and other tools
- Application Programming Interface (API): a software intermediary in code that enables two software programs to communicate with each other; API documentation provides common coding calls and functions for how developers can make requests and responses across applications (e.g., developer making an API call to its third-party cloud service provider which will respond with the infrastructure requested to power the application)
- Cloud computing: computing services like servers, storage, databases, networking, etc. delivered across the "cloud" (the Internet), generally describing objects abstracted from the underlying infrastructure
- Content delivery networks (CDNs): a network of proxy servers (intermediary between client requests and the servers providing that resource) and their data centres that deliver the requested content (e.g., videos, images, web and mobile content. etc.) at a higher speed (via caching) and scalability necessary for complex apps
- Caching: Storing data temporarily to improve performance
- Data minimalist: collecting only the data that is needed for the functioning of the app
- Open source: non-proprietary software programmes whose source code is publicly available for anyone to use, modify, or collaborate on
- Mitmproxy: An open-source tool for analysing encrypted datastreams by sitting in the middle of a connection between a client and a server, allowing the data to be examined
- Server-side: the processing takes place on an external web server
- Client-side: the processing takes place on the user's device

The app setup

1. First, we created a Google account for our research subject ('User'), which was required to download apps from the Google Play Store.

2. Then we downloaded each app from the Play Store onto our virtual Android machine deployed through the DIAAS.

3. We opened each app and completed their individual onboarding processes with the required personal information, the degrees of which varied across apps. Some apps required logging in with our Google or other email account, while others required providing basic information such as the user's name and date of birth. Some of the apps did not require any personal information before directing us to the cycle dashboard.

In the process of setting up our user onto the apps and populating their personal data and period cycle data, we took a data minimalist approach by providing the minimum amount of information needed for the proper functioning of the app. This meant we would not provide personal data like our birthday, height or weight, where doing so was optional, We also would not allow notifications in the app, location-sharing or other optional requests from the app. The purpose of this approach is to showcase how much extensive data each app requires of the user and how functional (or not functional) an app is when a user does not share everything.

Setting up the DIAAS

General practices we abided by when using the DIAAS to maintain a clean research environment are as follows:

- The mitmproxy window in the DIAAS displays the URL path on the left side, which shows where the request is being sent (e.g., the app's API, a third party website, etc.), and on the right side it shows the request and response information, such as what information in the app is being passed through (e.g., user input data about their cycle, date of birth during the onboarding questionnaire, device dimensions, device location).

- Every time we were done with a web traffic analysis session for an app, we went into the Android's Settings and clicked 'Force Stop' for the specific app. We then cleared the mitmproxy window via 'File > Clear all' to maintain a clean page.

- If we needed to restart an app to begin with a fresh slate (such as to redo the onboarding process), we went to the Android's Settings, selected the specific app and clicked on 'Clear storage', which would completely clear the storage and cache on the app to facilitate a fresh start.

We will be releasing publicly available documentation about the DIAAS in the near future, which we have developed in house as open-source software.

3. Research findings

Overall, our research found that apps were deploying less egregious data sharing practices compared to our findings last time. This time, we did not find the period tracking apps to be sharing users' period data with Facebook, though we did find that several apps were still integrating third parties such as advertising networks and analytics platforms to power certain functionalities of their app, and these third parties were often processing some degree of the user's personal and/or device data. Below we provide a summary of our findings for each app.

Flo

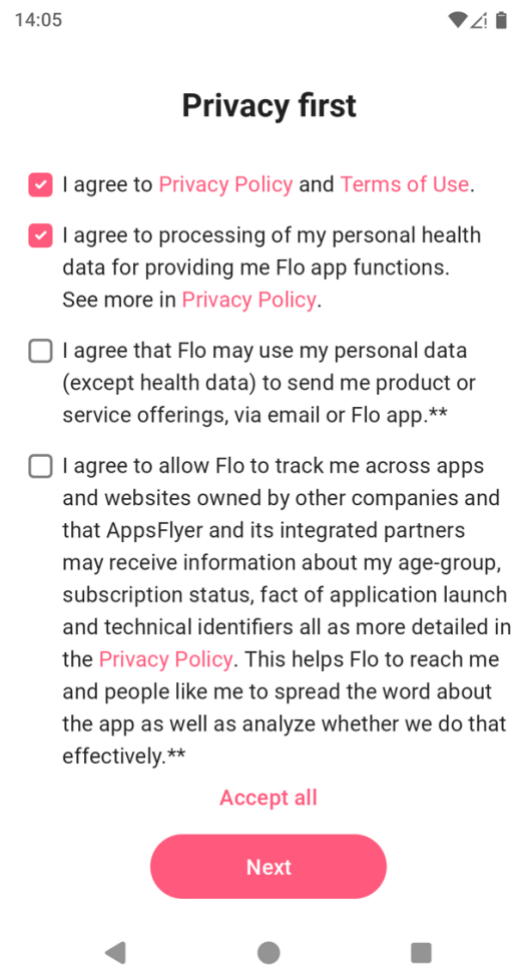
Flo, headquartered in London, UK, is one of the most popular period-tracking apps on the market with over 380 million downloads. The app was previously accused of sharing data with Facebook, such as informing Facebook whenever a Flo user was on their period or if they intended to get pregnant. In 2021 the U.S. Federal Trade Commission (FTC) reached a settlement with Flo to undergo an audit of their privacy policy and to obtain user permissions before sharing personal health data.

Flo also launched in 2022 an open-source 'Anonymous Mode' version of the app that allows users to use the app without having to create an account, and the company clarified in a public statement that Anonymous Mode ensures that the app won't be able to identify specific users if asked to do so by law enforcement.

For the purposes of our research, we tested Flo's default mode instead of Anonymous Mode; the app has clarified in its Privacy Policy that regardless of the mode selected, all user data remains safe and secure.

Flo had one of the lengthiest set-up processes, which presented the user with a questionnaire to personalise their profile before accessing their cycle dashboard.

Before delving into the questionnaire, we note that upon first launching the app, we were presented with the below consent agreements (Figure 1.1). We were able to proceed with ticking off just the first two boxes. It's worth noting that Flo took a positive opt-in approach to the consent form, rather than the default pre-ticking of boxes in the opt-out format. That is, the user can affirmatively choose what to consent to, rather than deselecting to indicate consent.

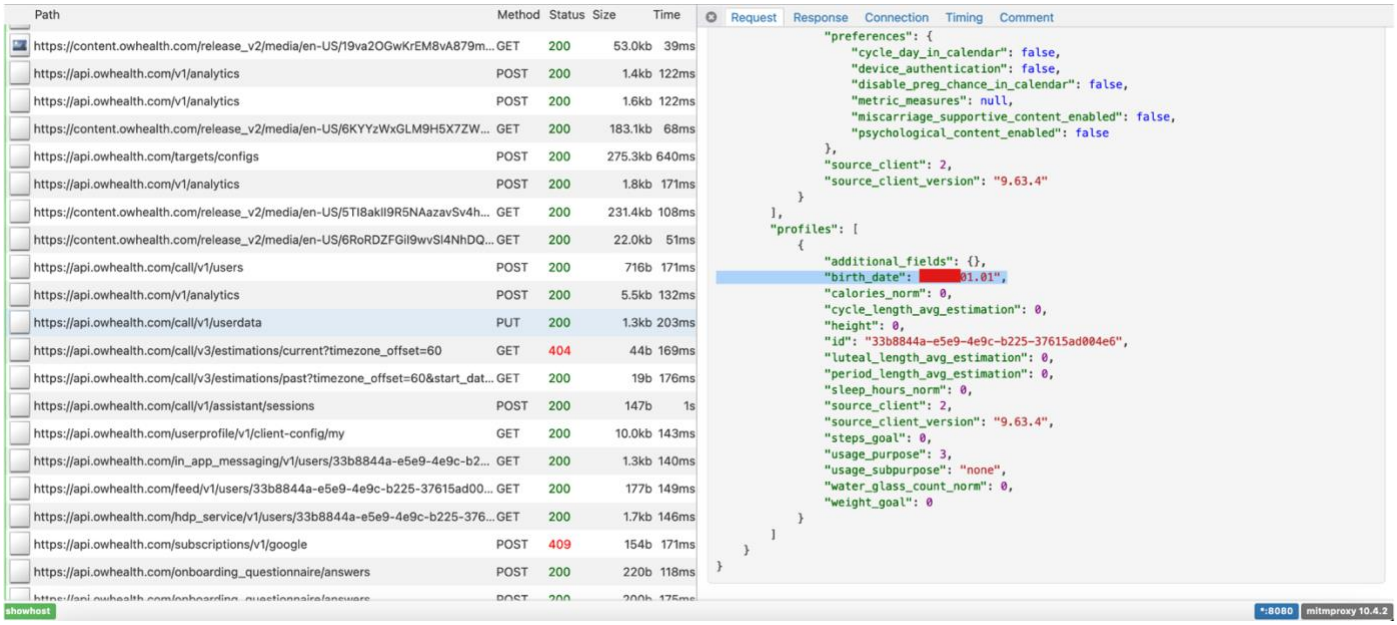


[Figure 1.1. In-app screenshot of Flo's privacy agreements for the user. Note the Privacy Policy is presented here at the very start of launching the app and before the questionnaire.]

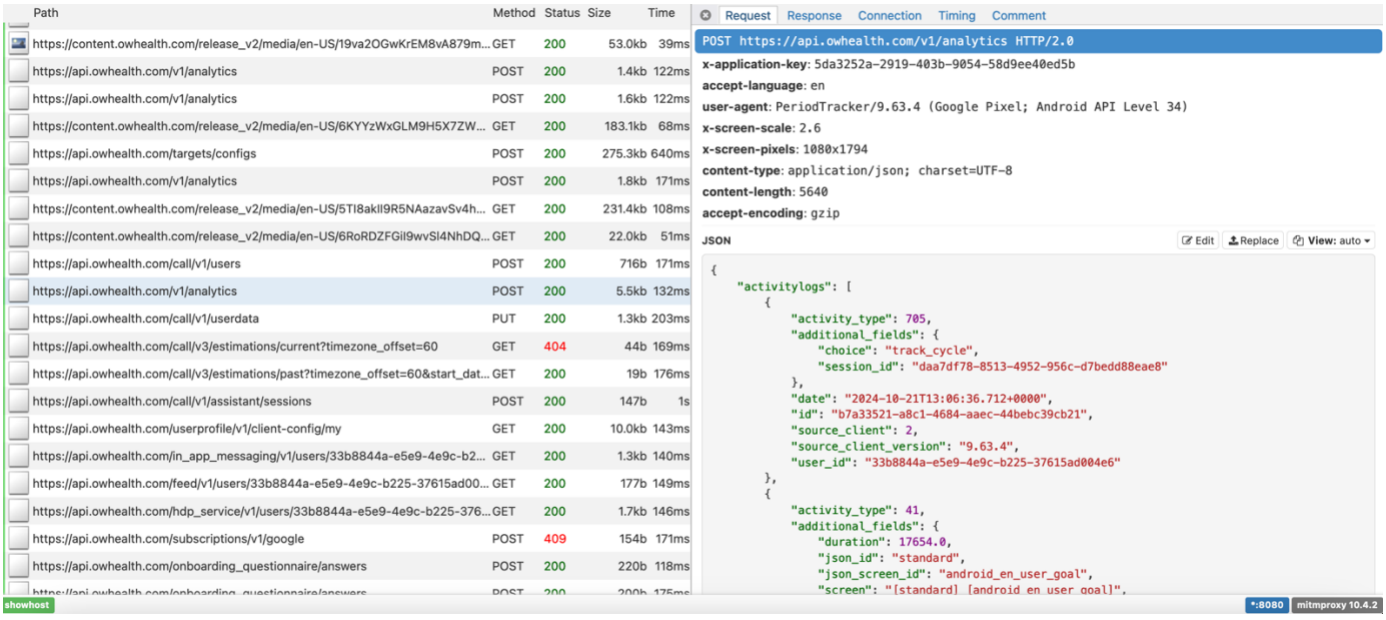
We then proceeded to the onboarding questionnaire, in which the user was asked a variety of questions about their cycle and what they are looking for in the app. The questionnaire allowed us the option to 'Skip' most questions (e.g., cycle duration, last period, height and weight), which we skipped to pursue a data minimalist approach (provide as little personal data as possible to use the app). We were not allowed to skip the question for our year of birth. In their response to our findings, Flo clarified that the purpose of this information is to verify the user’s age (only users 16 or older may use the app).

All our responses to the questionnaire were recorded in the web traffic and sent to the app's application programming interface (API). We noticed this occurred for several of the apps below, as an API facilitates the connection between a computer program (the app) and a computer (the system running the app). In effect, an API is an intermediary that communicates the rules, protocols, and tools that app developers use to interact with external services and databases, or their own libraries for the functionality of the app.

In the web traffic, we saw that our required answer for birth year and why we were using the app was sent to the API:



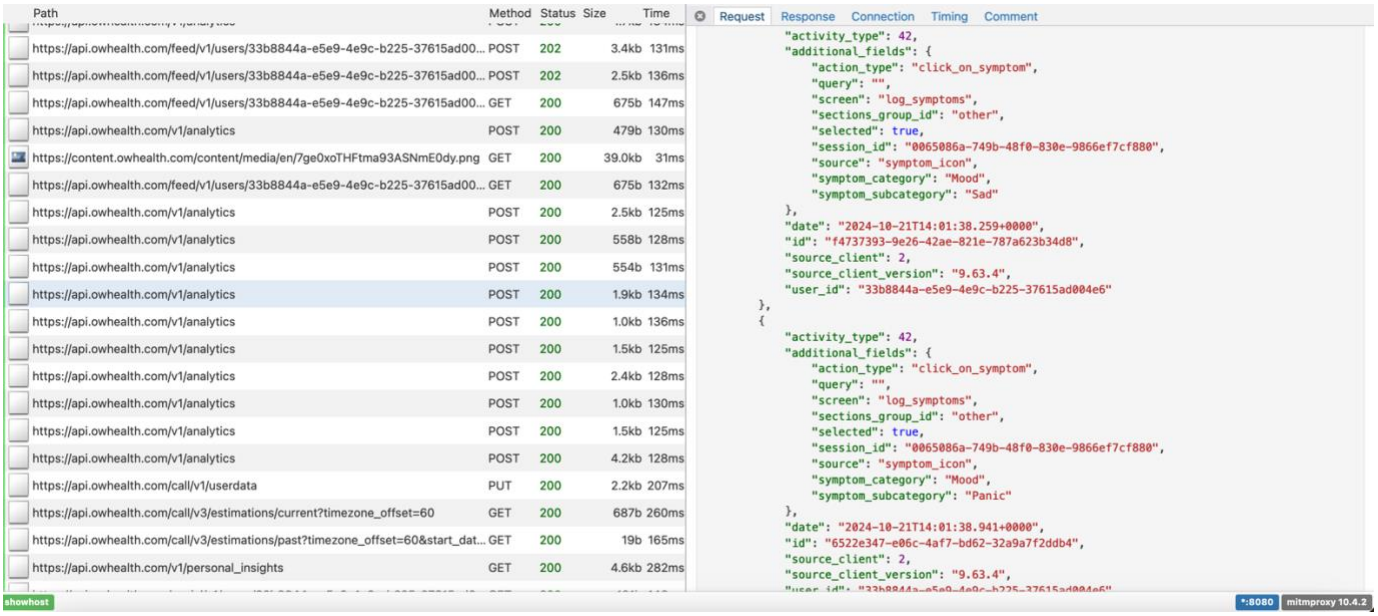
[Figure 1.2. See the 'birthdate' field. Note we were only asked for our birth year, not the month and day, hence the automatic population of 01/01. The 'Path' column in the left-hand side shows all the URL paths the information on the right-hand side is being requested by and sent to, and we see 'birth_date' is being sent to 'api.owhealth', which is Flo's developer's (OW Health) API.]



[Figure 1.3. See the 'choice' field, populated with 'track_cycle', which we selected among other options like tracking pregnancy.]

After completing the onboarding questionnaire, we were able to proceed on the app without having to create an account with our name or email.

Throughout our use of the app, we could see all our cycle inputs also being sent across the web traffic to Flo's API. For instance, we could see our symptoms being requested by the API:



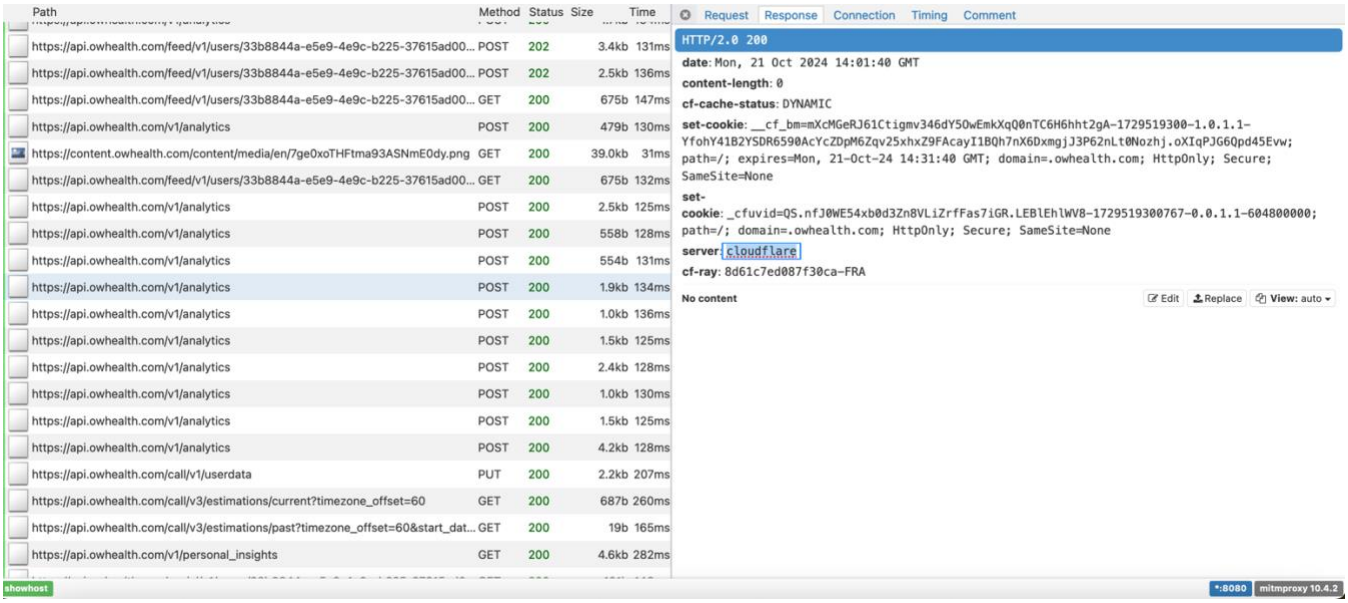
[Figure 1.4. See 'symptom_category' and 'symptom_subcategory'].

We then noticed that all these logs of our input data (e.g., 'Sad', 'Panic' under 'symptom_subcategory') were being sent over the Cloudflare server (Figure 1.5), a cloud-based content delivery network (CDN) used by many websites and services across various industries. CDNs are increasingly deployed as websites and apps grow their capacity and userbase, and cloud-based servers are highly scalable to accommodate for this growing infrastructure. Flo did disclose by name their use of Cloudflare in their Privacy Policy and labels Cloudflare as a data processor.

As the CDN, Cloudflare facilitates as a 'man-in-the-middle' all the communications of the Flo app to its API, such as forwarding the user's cycle inputs to the API and servicing the app's chatbot. Technically, Cloudflare operates as a blind forwarder (i.e., a proxy), and Cloudflare confirmed in their response to our report that it is not technically possible for them to turn over any content transiting across their network, nor does it access the data customers choose to send, which is protected under TLS encryption for all data in transit between an end user and any Cloudflare data center. Flo echoed in their response to us that Cloudflare does not access content traveling over its network, and its security-by-design structure 'significantly limits any potential for unauthorized or excessive access to personal data'.

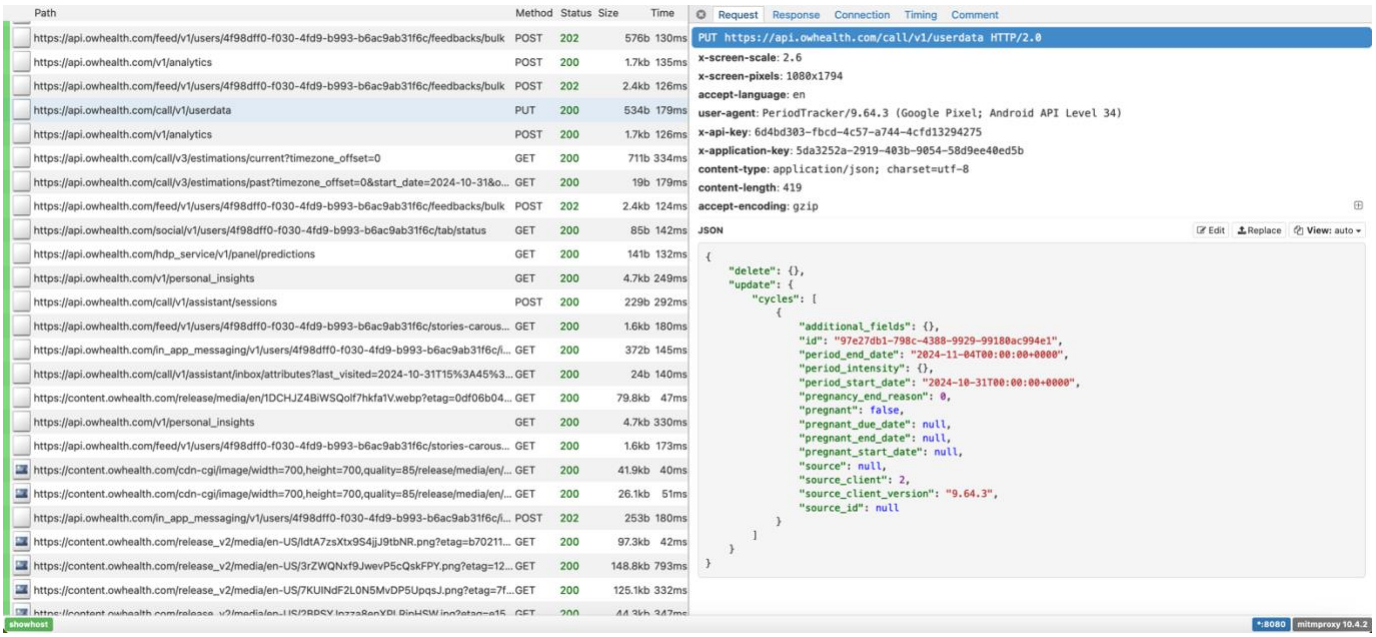
It's also worth noting that Cloudflare said they deploy their 'Privacy Gateway' for Flo's Anonymous Mode (which we didn't test). 'Privacy Gateway' encrypts HTTP requests and responses between a client and application server. In practice, this means that request data for Anonymous Mode users is encrypted by Cloudflare between the user and Flo, which prevents

Flo from seeing the IP addresses of those users and Cloudflare from seeing the contents of that request data.

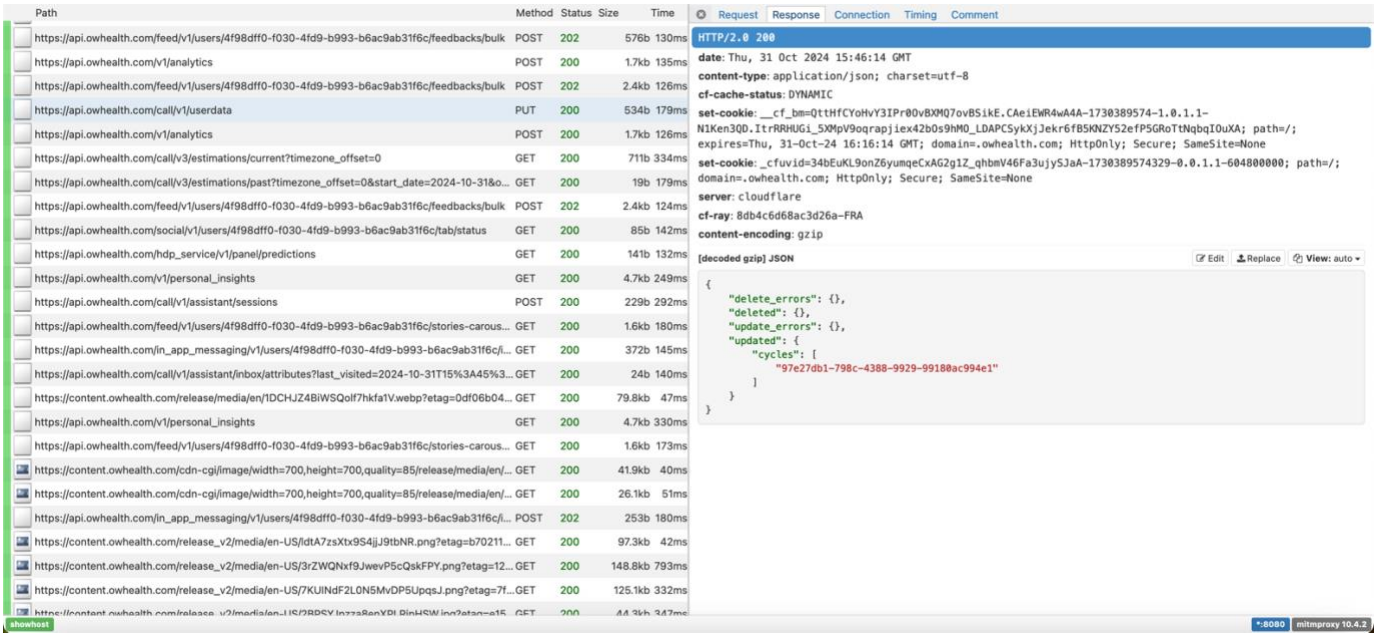


[Figure 1.5. See server: 'cloudflare' in the Response, with the data in Figure 1.4 being sent in the Request.]

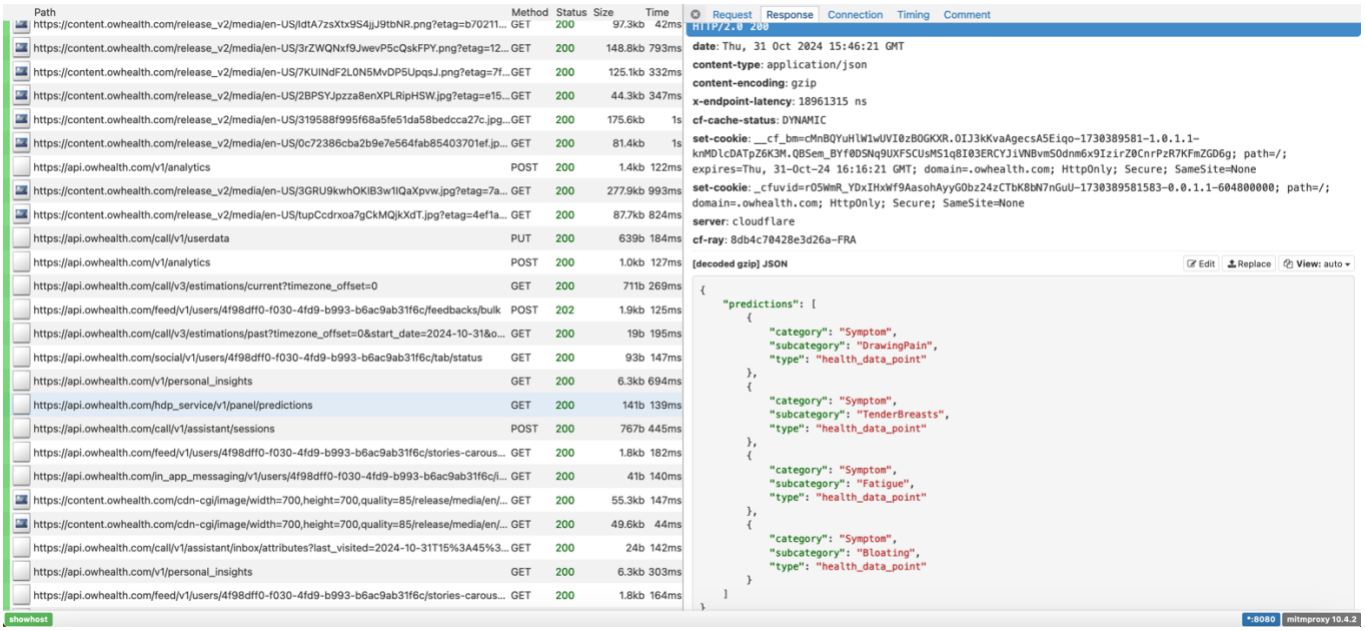
We similarly saw our period date updates, and our predictions sent over the Cloudflare server to the API:



[Figure 1.6. See the requested input data (e.g., 'period_start_date').]

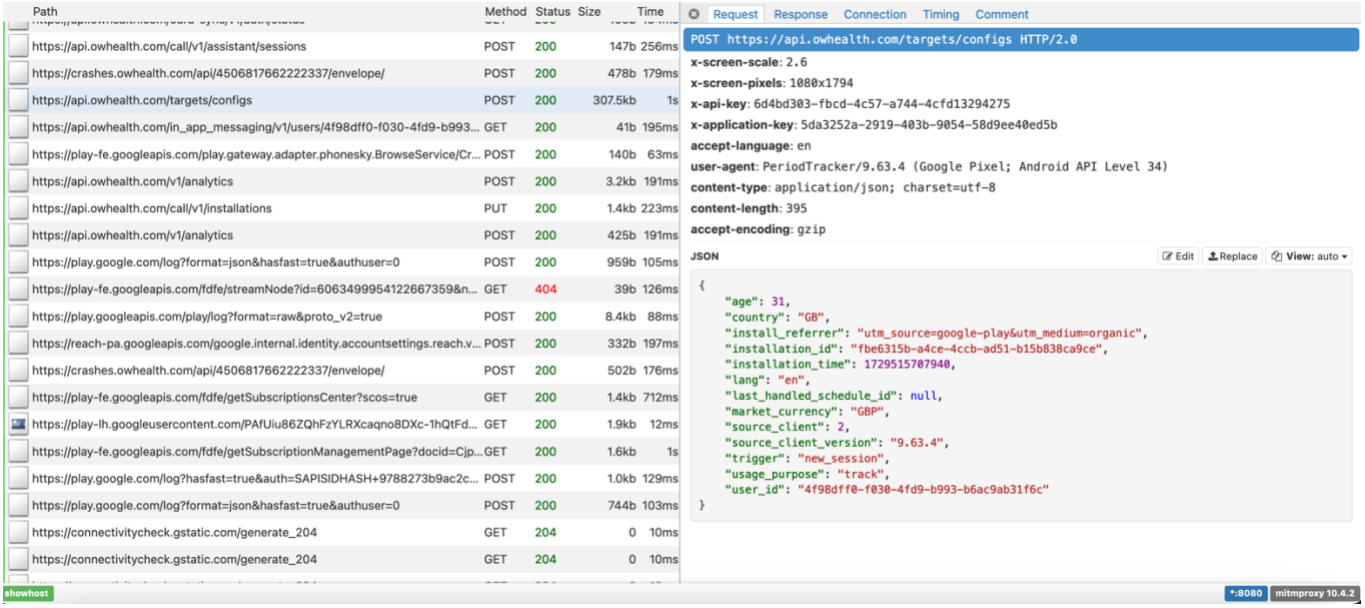


[Figure 1.7. The Response for the above data Request being received over the Cloudflare server].

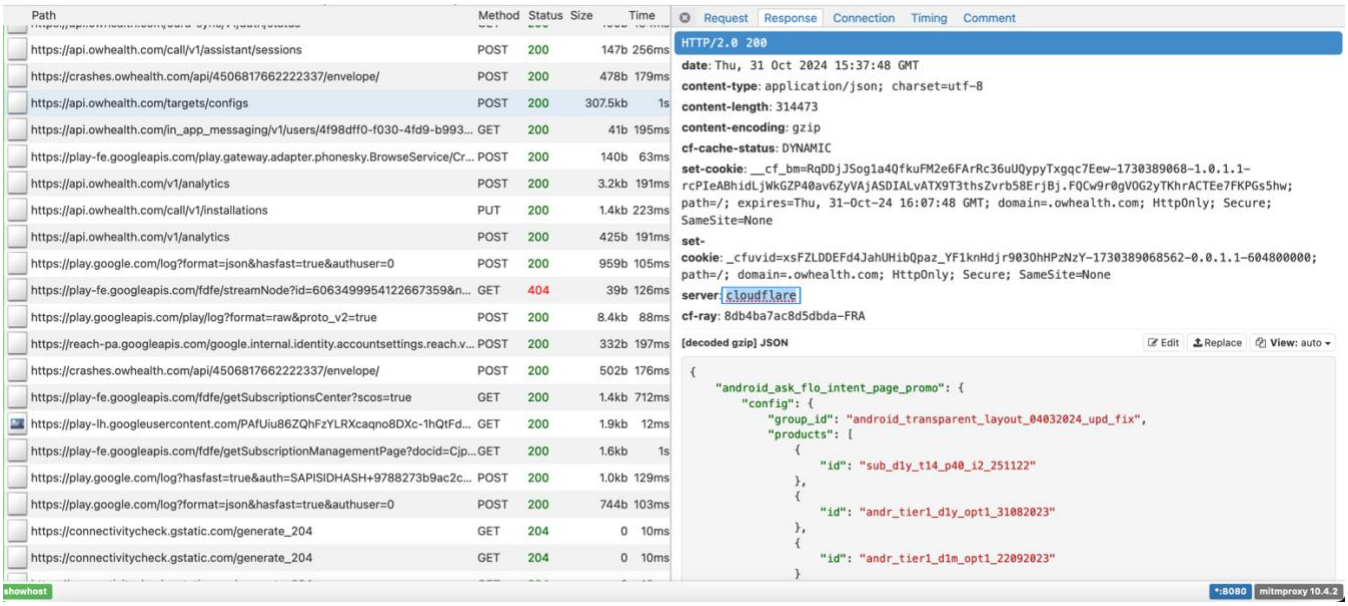


[Figure 1.8. Symptom Predictions generated and being processed over the Cloudflare server.]

We also saw that the algorithm calculated our age based on our birth year entry and sent this to the API (Figure 1.9) over the Cloudflare server (Figure 1.10):

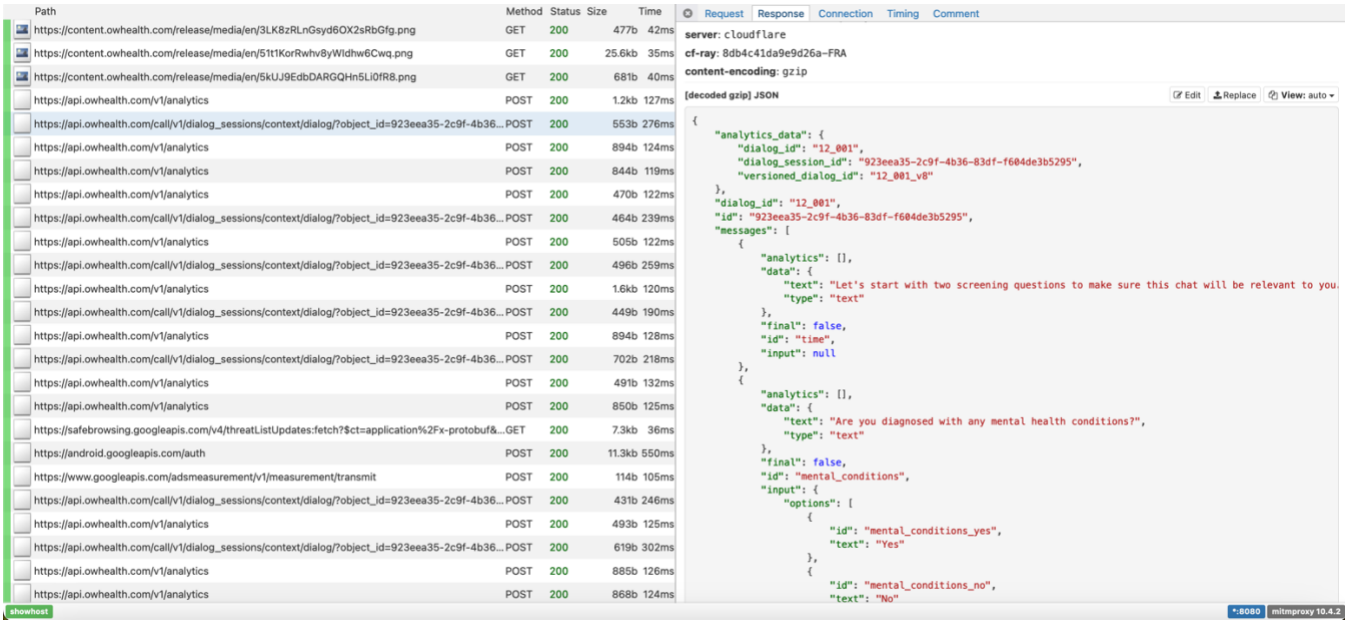


[Figure 1.9. See 'age' field. Recall we only provided the birth year, so the age itself was calculated by Flo.]



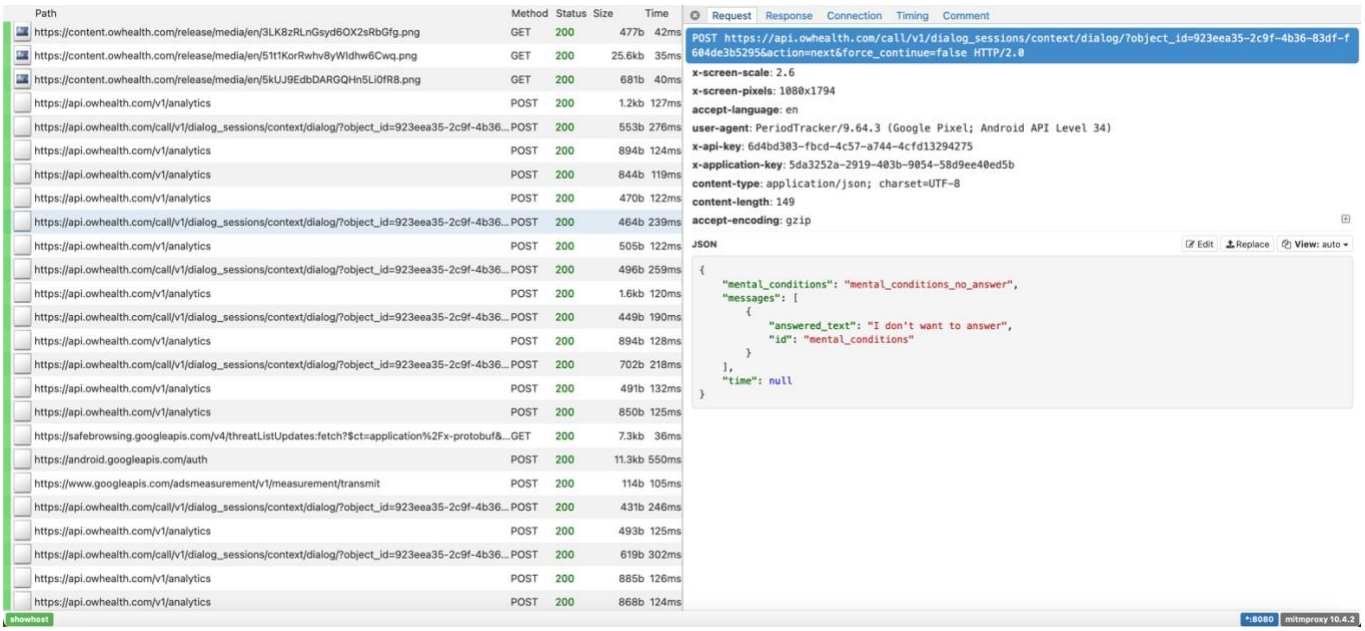
[Figure 1.10. The above data Request is processed here over the Cloudflare server.]

From here, we sought to try out Flo's chatbot tool, which primarily served as a Q&A screening tool for different health concerns (e.g., acne). These were all controlled multiple-choice chats (the user could only select from provided multiple choice options). We note that this chatbot was also hosted over the Cloudflare network, and each chat session was logged under a unique chat ID:

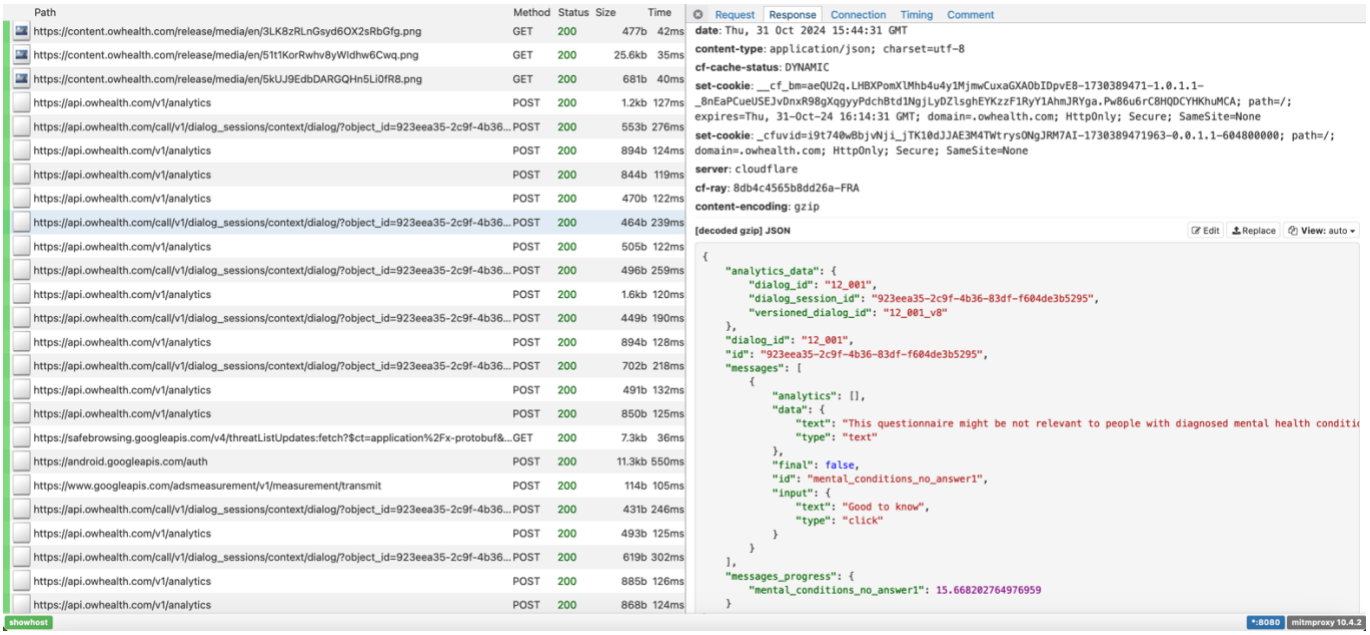


[Figure 1.11. See the chatbot's outputs above, linked to the unique 'id' assigned for this chat.]

All our responses in the chat were recorded in the web traffic request, perhaps for the API to provide a response call to the logged user selection. For instance, we can see the API logged our answer of 'mental_conditions_no_answer' over the Cloudflare server:



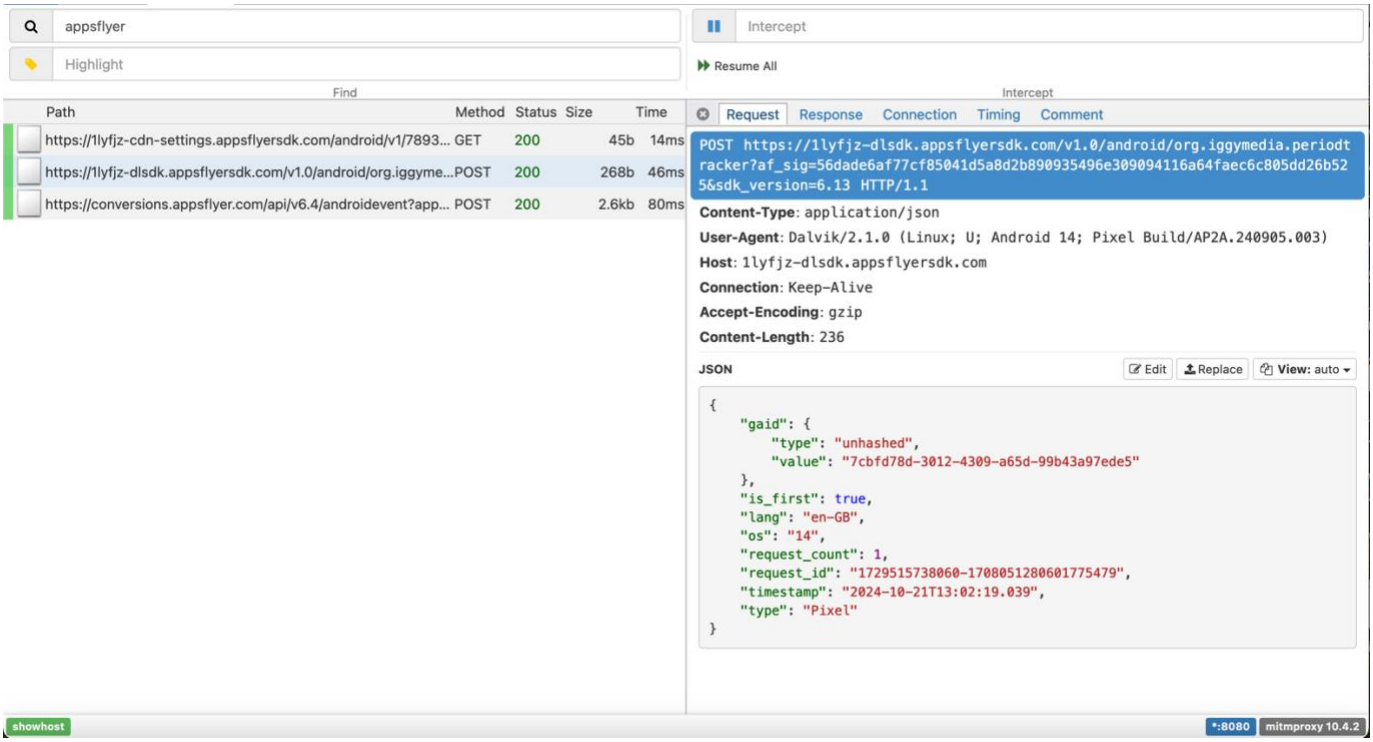
[1.12. In response to the chatbot's question of what mental health conditions we have, we responded 'I don't want to answer' which the chatbot set as the variable 'mental_conditions_no_answer'.]



[1.13. We can see the above answer of 'mental_conditions_no_answer' passed to this response here over the Cloudflare server, which returns the chatbot's output of 'Good to know'.]

We also noticed throughout the web traffic some instances of third-party URL paths from AppsFlyer, which is a San Francisco-based marketing analytics platform that allows developers to measure the performance of their app and/or analyse their marketing activity.

In its response to our research, AppsFlyer clarified that it ‘operates strictly as a data processor on behalf of its customers,’ and that ‘all personal data collected by customers through the AppsFlyer SDK is controlled and owned by the customer’. AppsFlyer also stated that any form of data collection on its part occurs under ‘instruction’ by the customer (developer) themselves. That is, any configurations through AppsFlyer’s SDK for data collection as seen below is configured by the customer.



[Figure 1.14. AppsFlyer's software development kit (SDK) requesting language, OS, timestamp and device type information.]

We note that Flo disclosed to users in both its in-app consent agreement (Figure 1.1) and its Privacy Policy that it integrates third party AppsFlyer. In their response to our research, Flo further clarified that they utilize two different AppsFlyer products, and the AppsFlyer request seen in Figure 1.14 is related to a different purpose than stated in the consent form in Figure 1.1, which concerned marketing attributions. In the above scenario in Figure 1.14, Flo uses 'third party AppsFlyer to help us identify you as an existing user when you use the App'. (Note that this is disclosed by Flo in their Privacy Policy in a diagram representing their use of AppsFlyer). In scenarios we've identified when our technical information was sent to AppsFlyer, this was done 'for the purposes of determining if the user has already completed the Flo website onboarding journey prior to downloading the Flo App', and this processing activity is not connected to the marketing purposes in the consent form. Flo clarified that by not consenting to the marketing purposes of AppsFlyer as in Figure 1.1, our device data would not be shared with AppsFlyer for these marketing purposes.

It's worth noting here that Flo had one of the more robust Privacy Policies that disclosed the use of third parties by name, such as AppsFlyer, as well as disclosing their use of cloud-based services like Cloudflare.

Period Tracker by Simple Design

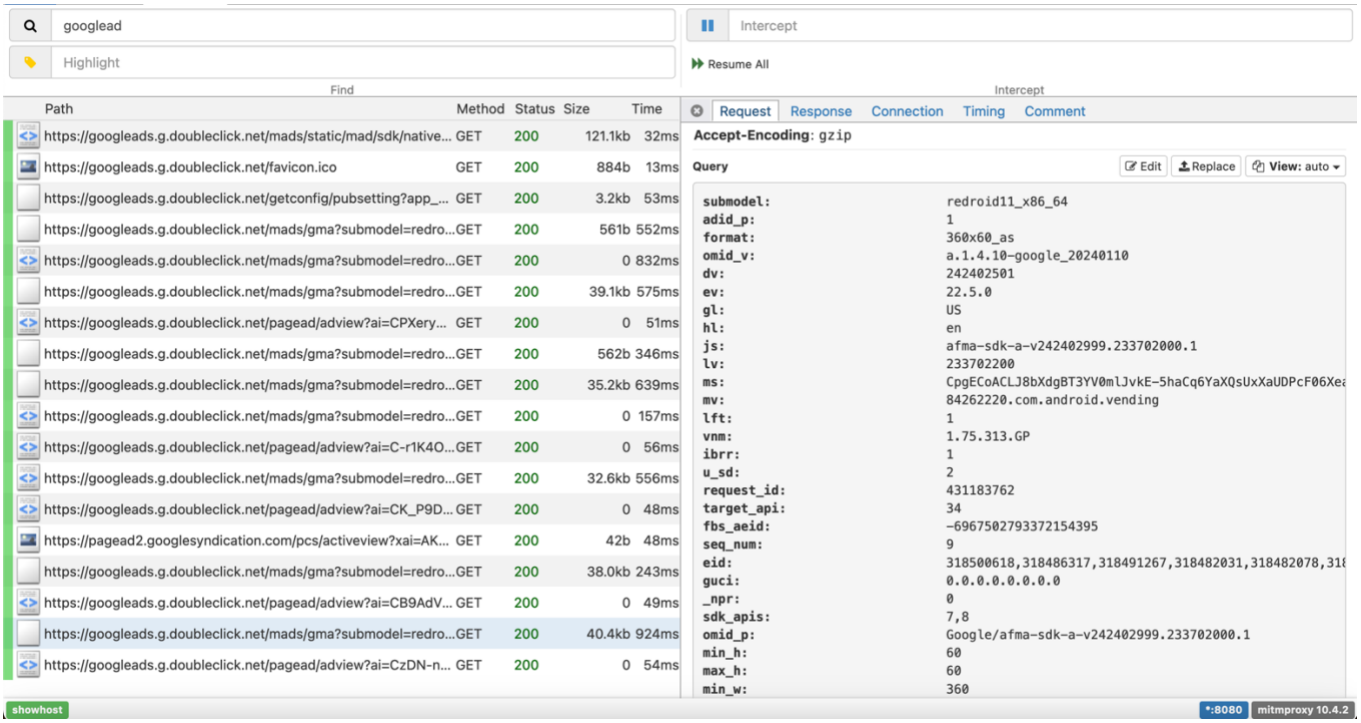
Period Tracker by Simple Design is another popular period tracking app that has over 150 million users. To begin using this app the user answers a set of three onboarding questions for about their cycle pattern. The user has the option to answer 'I'm not sure' for each question.

After answering 'I'm not sure' for these three questions, we were able to proceed on the app without having to create an account.

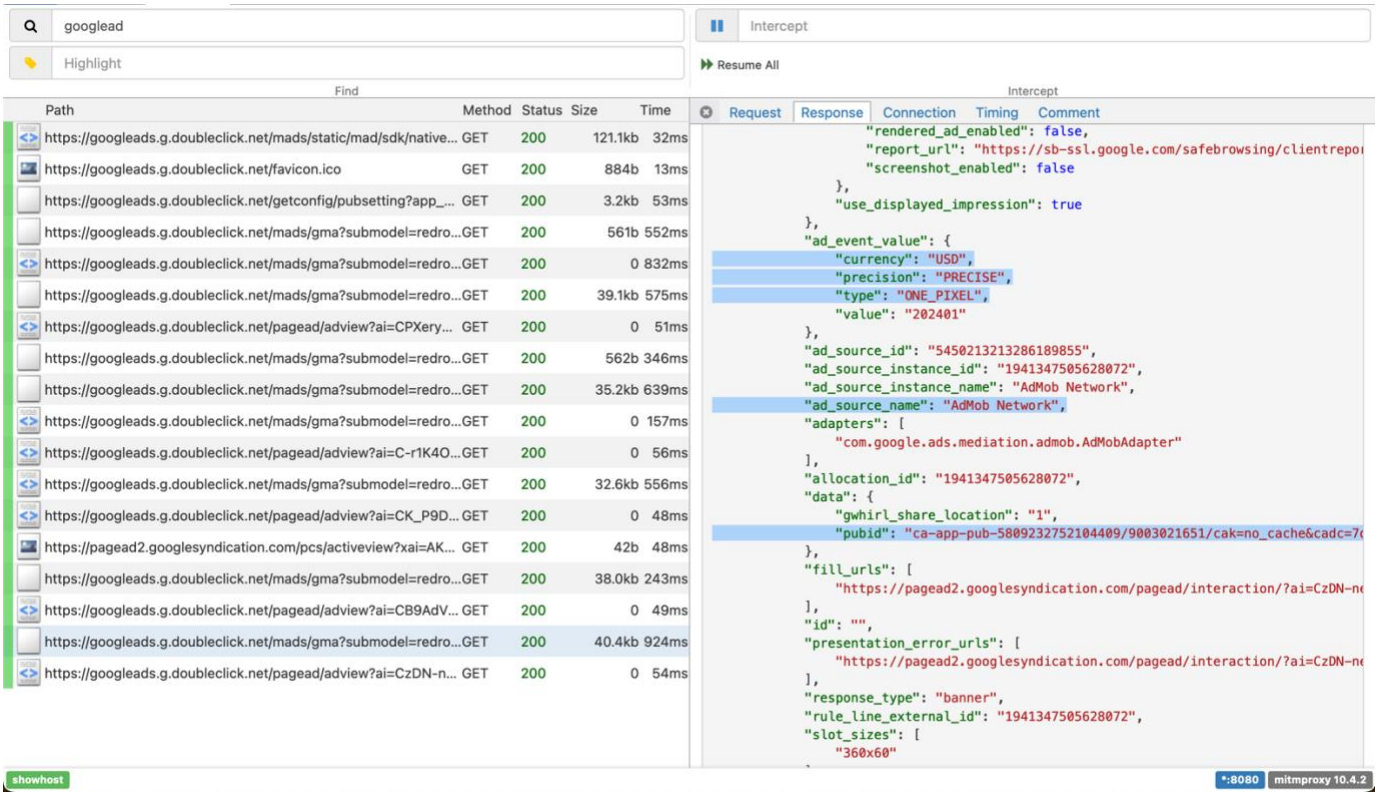
Throughout our experience inputting our cycle data, we did not appear to see our cycle inputs (e.g., blood flow, symptoms) logged anywhere in the web traffic (as was the case in our 2019 investigation), therefore the app did not appear to be communicating with an API. The app developers also confirmed in their response to us that all period input data is stored locally on the device.

However, we did notice a flood of third-party web traffic in the form of advertising and analytics SDKs. Upon opening the app, the user is first presented with a consent dialog box for which they can select their preferences on personalized advertising, which appears to be managed by Google Ads. We note that, at the time of our testing in August-September 2024, a consent dialog box did not appear when we set up the app. The app developers clarified and provided evidence that a consent dialog box does indeed appear upon launching the app, but when we redownloaded the latest version of the app and retested it, we still did not appear to see a consent dialog box. Separately, the developers also clarified that user consent for ads is managed by Google and that users can modify their preferences by navigating to Settings > Personalised Advertising.

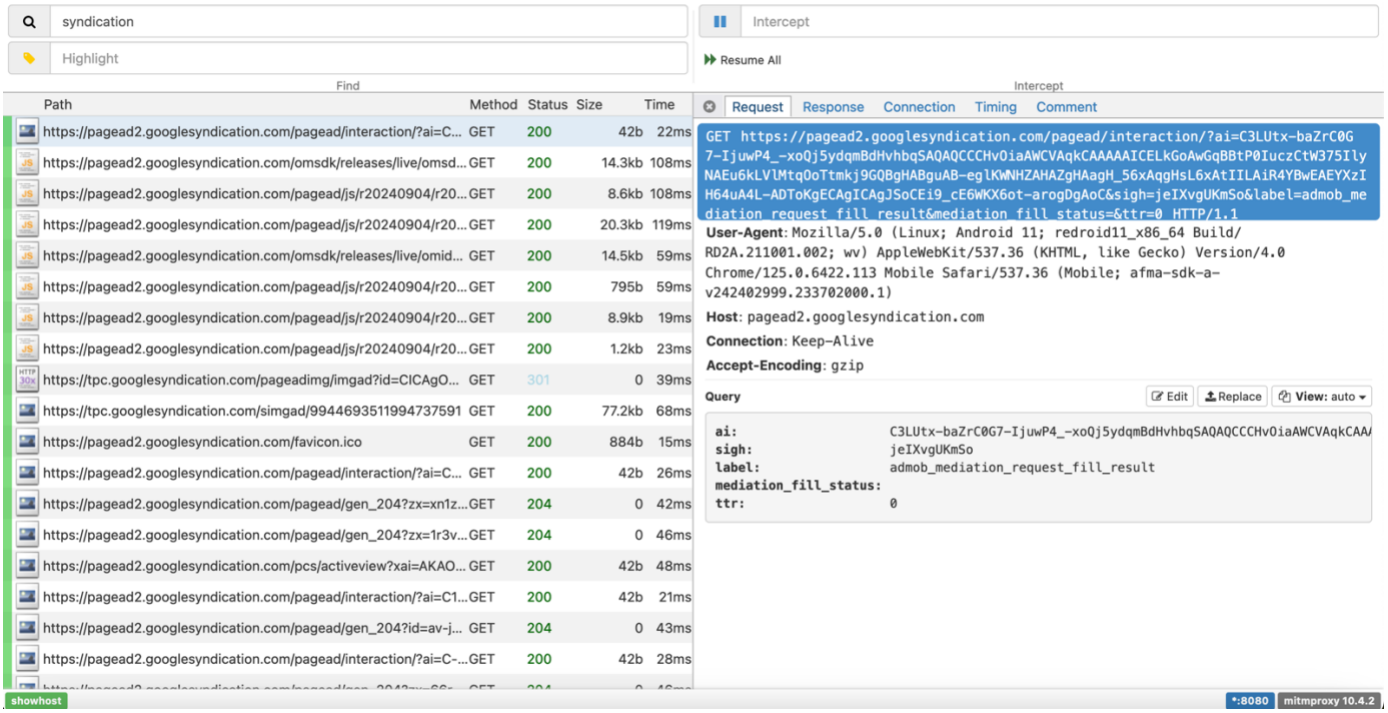
Throughout our use of the app, the web traffic was flooded with Google Ads URLs that suggested the app was a Google Ads customer. For instance, we observed numerous URL paths with 'GoogleAds' and DoubleClick (Figures 2.1, 2.2), and Google Syndication (Figures 2.3, 2.4).



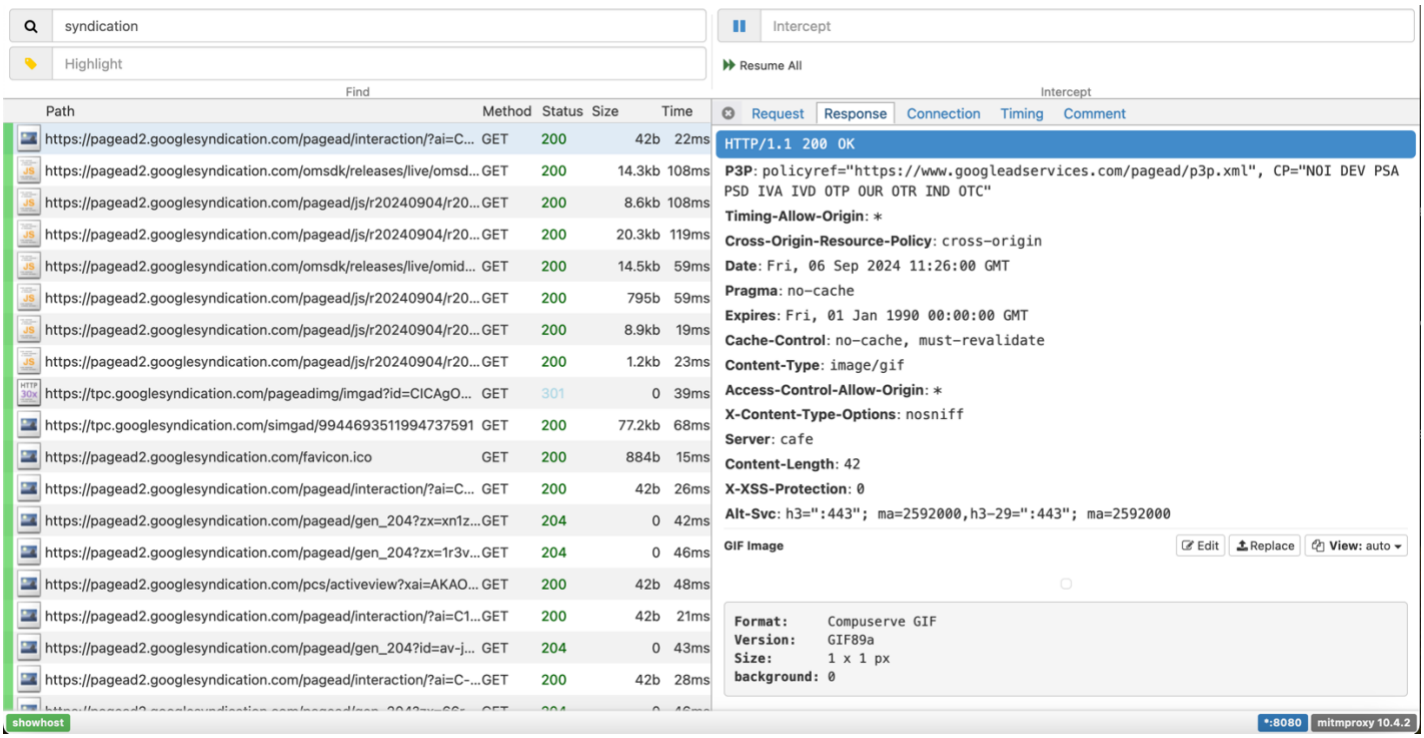
[Figure 2.1. Here is a request from Google Ads containing information about the device (e.g., 'format', 'submodel' of the device, etc.). In summary, this likely appears to be requesting information about the device to decide what format of ads (e.g., ads for the height and width of this device) fits this client best.]



[Figure 2.2. In the response for the above request, we see Google Ads has returned the appropriate currency, which it understands to be USD (which is not entirely accurate), as well as the source of the ad ('ad_source_name').]



[Figure 2.3. Google Ads call.]

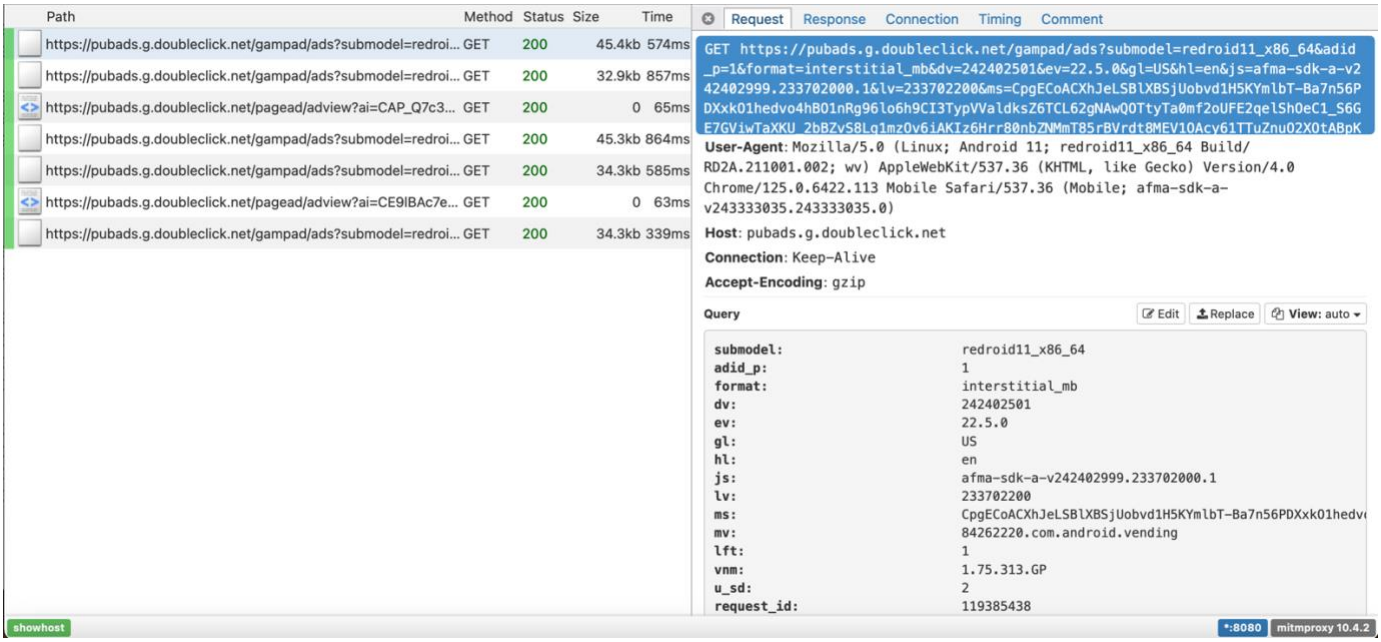


[Figure 2.4. The response to the Google Ads call is a 1x1 GIF, which represents a tracking pixel that can track user activity/behavior in the app.]

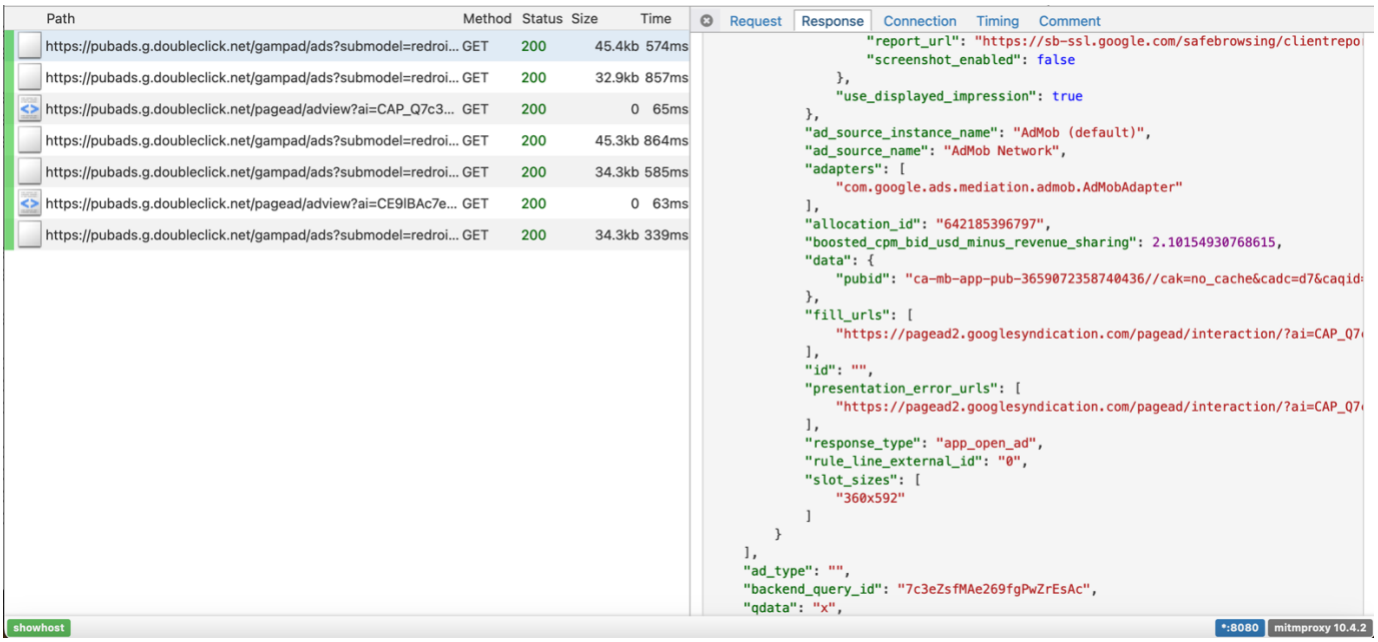
In Figures 2.1 and 2.2, we can see that DoubleClick, a Google-owned advertising tool that places ad banners, requested information about the user's device such as their device type, the screen's height and width, etc. The response shows a variety of ads-related information, such as the 'pubid' (publisher ID), currency, type of device and the ad source, which is 'AdMob Network', a Google tool that matches paid-for ads to apps based on certain criteria a developer can set, such as the appropriate ad for the specific height and width of a screen.

The presence of Google Syndication URLs in Figures 2.3 and 2.4 also point to this app's integration of Google's advertising network as a Google Ads customer. The Google-owned domain 'googlesyndication' is used for storing and loading ad content and other ads-related resources for Google AdSense and DoubleClick. The response returns a 1x1 tracker pixel, which is a tiny, invisible image embedded into the app that will automatically send information about the user's activity to the tracker owner (in this case, Google Ads).

These Google Ads paths continued to populate the web traffic every time we interacted with the app, and we saw additional URLs linked to googlesyndication.com like 'pubads' that made similar requests for device data (device dimensions, time in session, etc.) and receiving similar responses (AdMob Network as the ad source, publisher ID, etc.):



[Figure 2.5. Information requested by Google Ads is device information like above.]

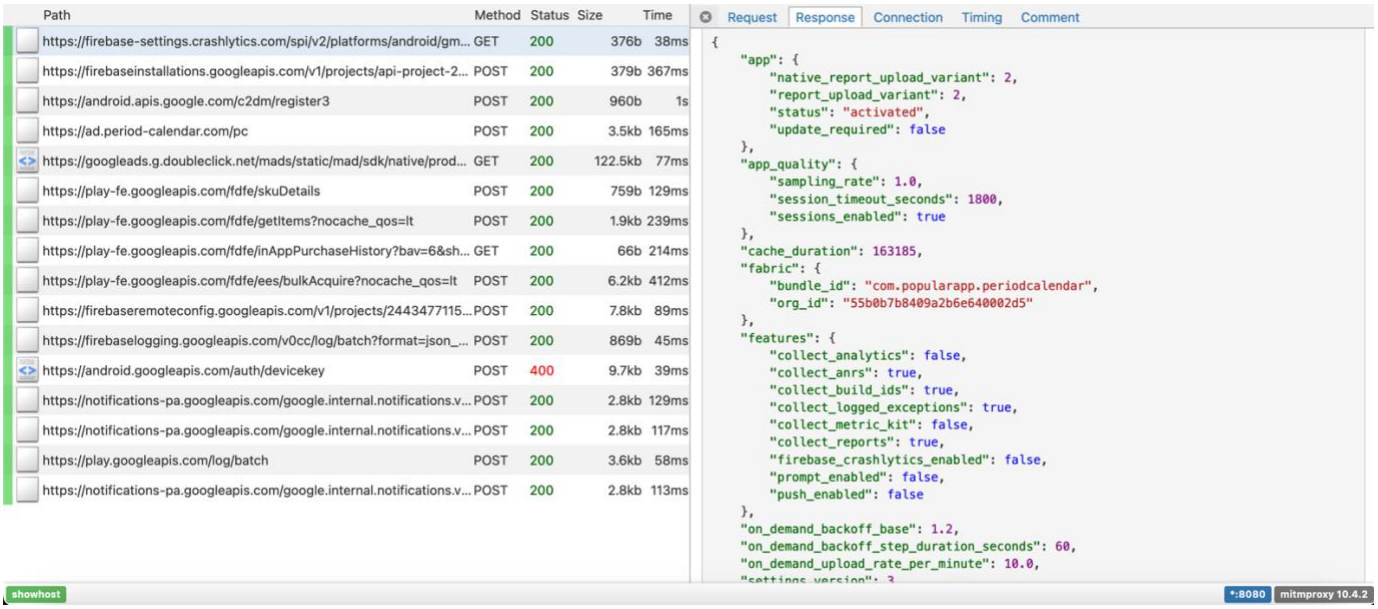


[Figure 2.6. The response also locates 'Ad Mob Network' as the source of ads to place and fill in the app.]

While we did not see our period input data being sent across the web traffic, we are seeing our device data being sent continuously to Google's ad network for the purposes of serving us ads in the app. Note that the app discloses in its Privacy Policy that it utilises third party service providers for analytics and displaying interest-based ads drawn from users' online behavior and device data, and the Privacy Policy does mention by name Google Analytics, Google AdSense, DoubleClick and AdMob.

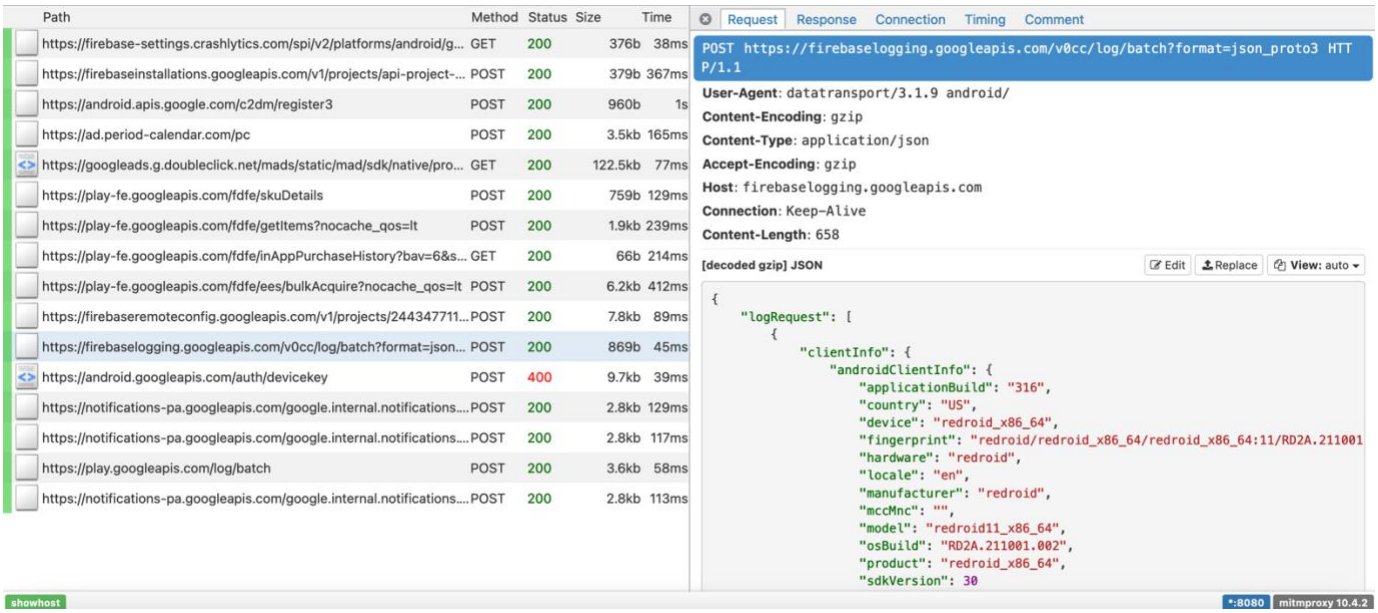
Additionally, we saw third party URLs from Firebase, which is Google's (acquired in 2014) mobile and web app development platform for building apps.

In the web traffic for this Period Tracker app, Firebase appears numerous times, such as in the form of its Crashlytics crash reporting feature (Figure 2.8), which collects telemetry data, such as when the app encounters an error or crashes.



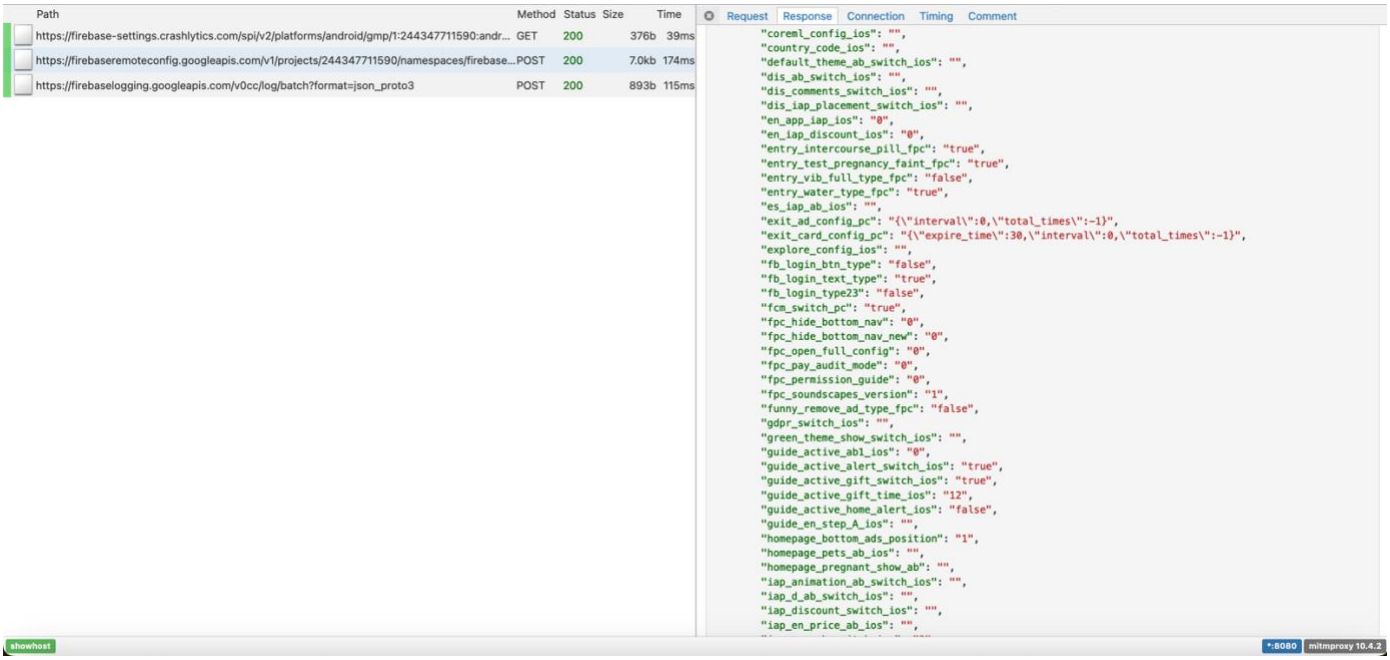
[Figure 2.7. The settings for Firebase's Crashlytics feature.]

Other appearances of the Firebase configuration beyond Crashlytics show 'clientInfo' being requested, which contains the user’s device and OS information among other device data:



[Figure 2.8. See above a range of device data is being requested by Firebase, from 'country' (incorrectly detected as the US) to 'device' to 'hardware'.]

Firebase's remote configuration path also included allowances for 'fb_login_text_type:true' in the response for the above request (Figure 2.10). This could likely be for scenarios when a user might choose to log in via a third-party social network, in which case the Privacy Policy states that it may collect some third-party data automatically, such as the user's account name, email address and public profile.



[Figure 2.9. See 'fb_login_text_type'.]

Firebase enables a range of app development and user data management activities for app developers. In general, it is commonly deployed (as we'll see below) because it is easily configurable with the Android SDK and thus is a relatively straightforward analytics tool for developers to integrate into their apps. We note that the app’s Privacy Policy states that device data may be collected by the app to inform new features and perform crash testing, but there was a lack of clarity that Firebase would be the third party collecting this data. When responding to our findings, the app clarified that the reason they did not explicitly name Firebase when providing their disclosures about analytics practices is because Firebase does not track users personally (‘Firebase is used solely for analytics and does not collect personally identifiable information’). However, the app has now said they will update their disclosures to specify the tool name.

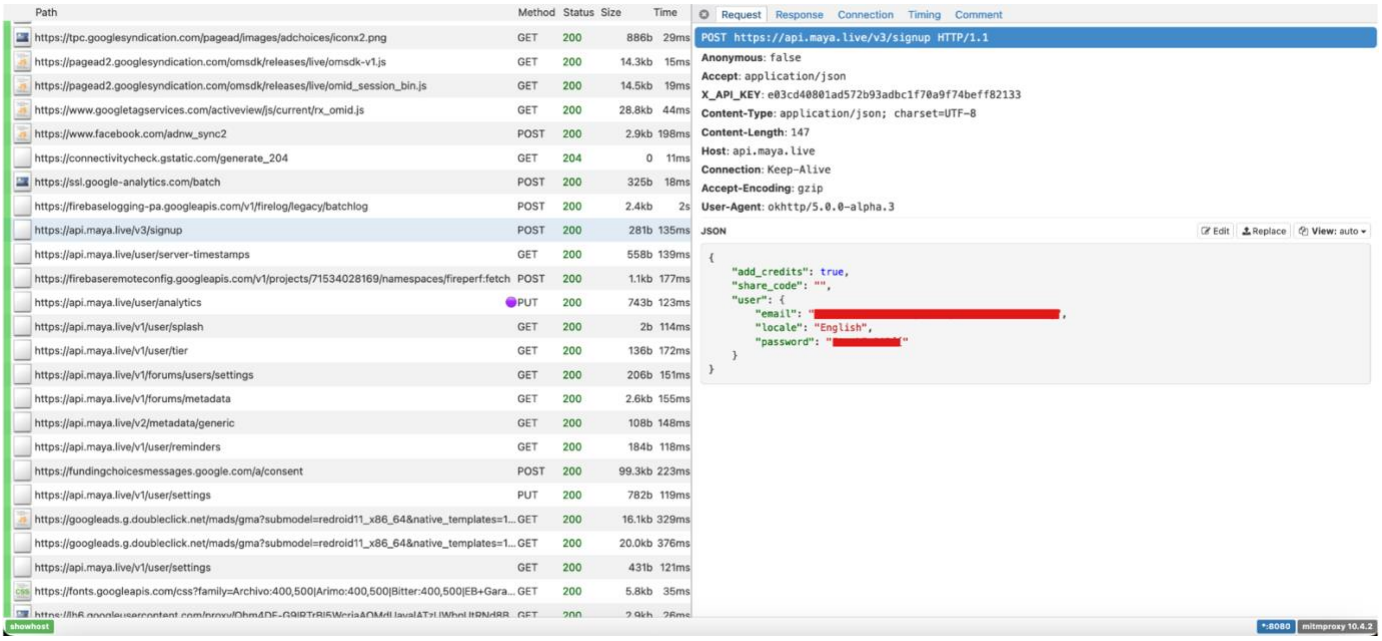
As a general note on Firebase, the main purpose of developers using Firebase, as we will see for many apps below, is to collect analytics information to aggregate into larger conclusions for developers (e.g., how many Android devices use this app, where in the world are most of these users, what screen sizes are commonly used). However, recall that Firebase is owned by Google; in theory, Google might have access to the device data collected by Firebase and, under the guise of the data being 'aggregated' and thus potentially anonymised, use this data for its own purposes. In fact, Firebase's Privacy Policy discloses that this further sharing with Google can happen in some cases (more on this below).

Maya

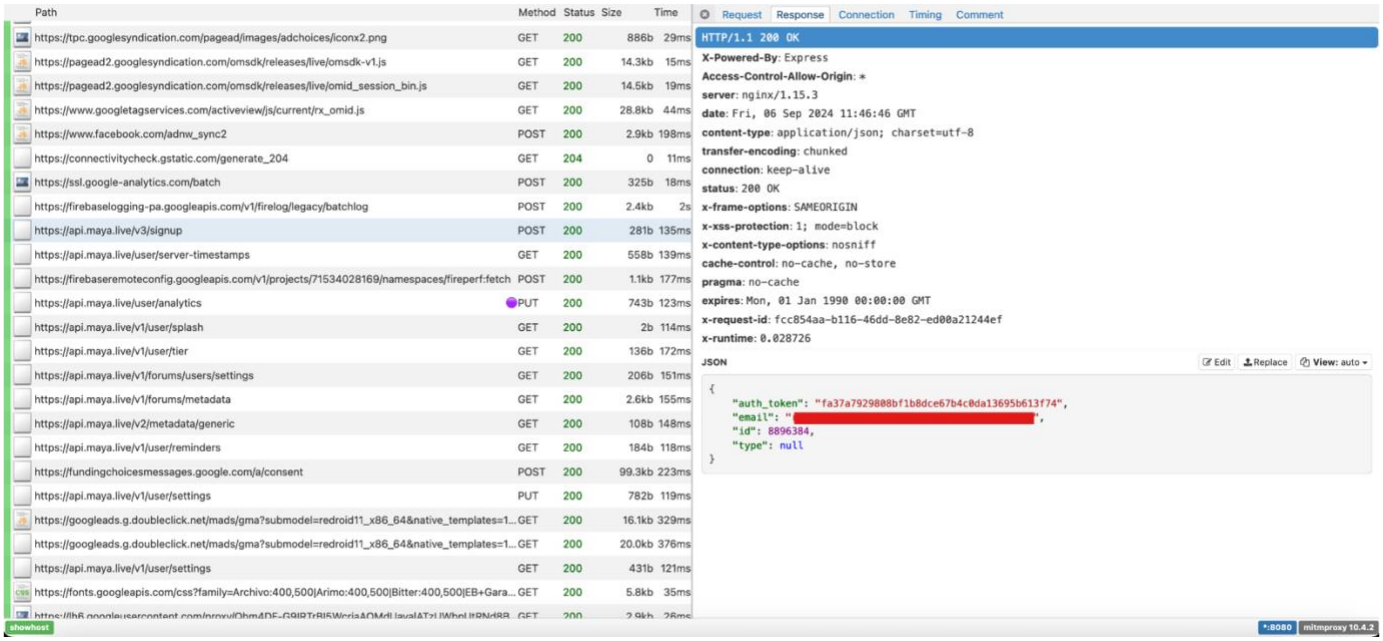
The Maya app is a period tracker app by Plackal Tech based in India. In our previous investigation, we revealed Maya was sharing a plethora of user input data to Facebook. However, in response to our 2019 research, the app claimed it had since ‘removed both the Facebook core SDK and Analytics SDK from Maya’ while ‘continu[ing] to use the Facebook Ad SDK, post opt-in to our terms and conditions and privacy policy’ for revenue purposes, the latter of which 'does not share any personally identifiable data or medical data with the Facebook Ad SDK.'

To get started on the Maya app, we were required to create an account using an email address and password. Note Maya's Privacy Policy states that 'the Application also automatically syncs the data entered by You to Your registered e-mail address.'

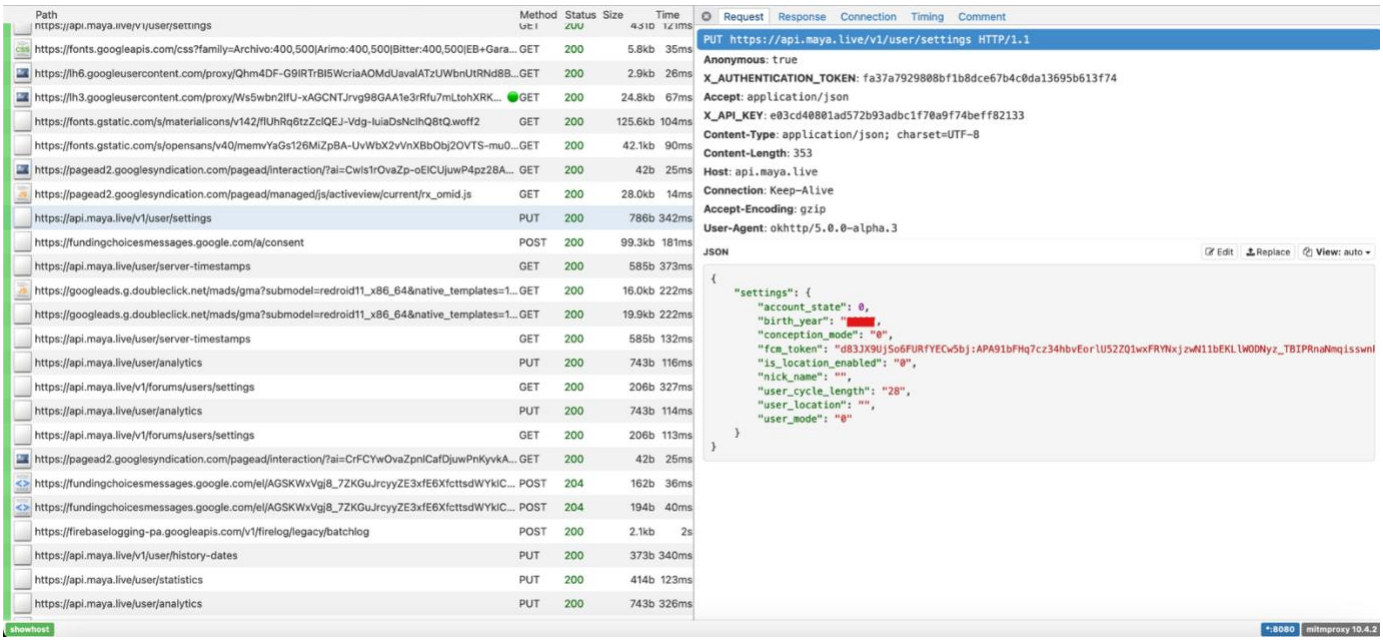
After creating our account, we completed the app's onboarding questionnaire about our cycle patterns. The app communicated all our inputs to its API, such as the email we used to sign up and our answers to the questionnaire:



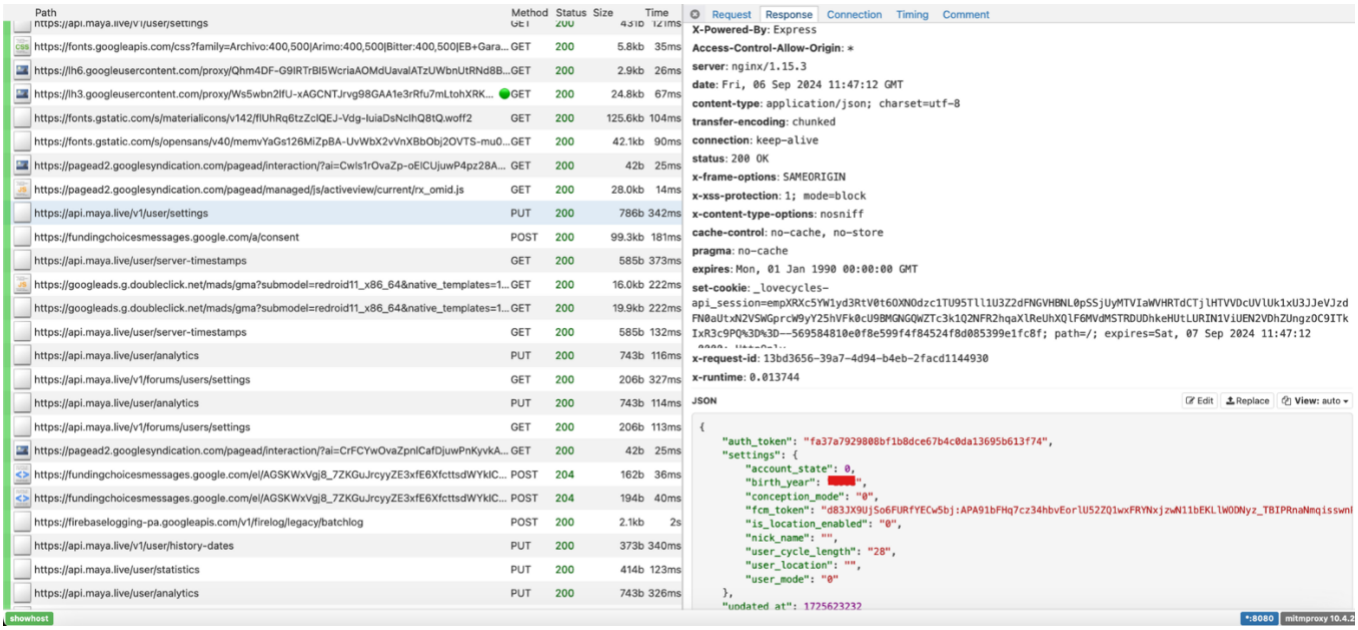
[Figure 3.1. Our email was logged in the web traffic and sent to the API.]



[Figure 3.2. In response, we were assigned a unique ID.]

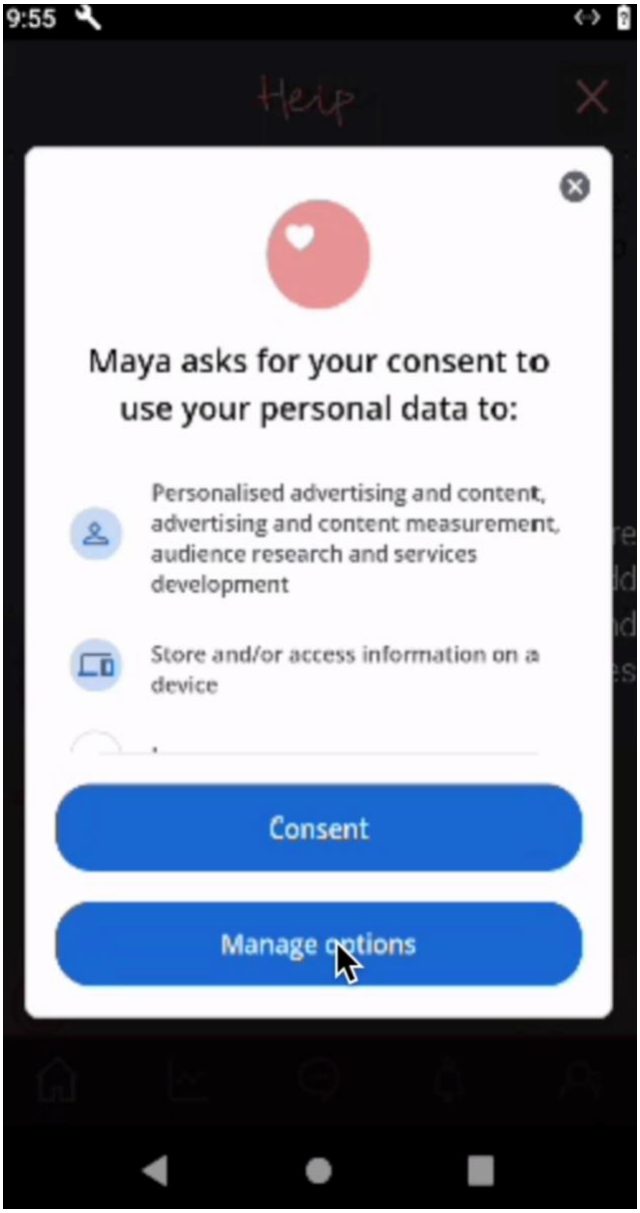


[Figure 3.3. Our birth year and cycle information was also sent to the API to update our settings.]

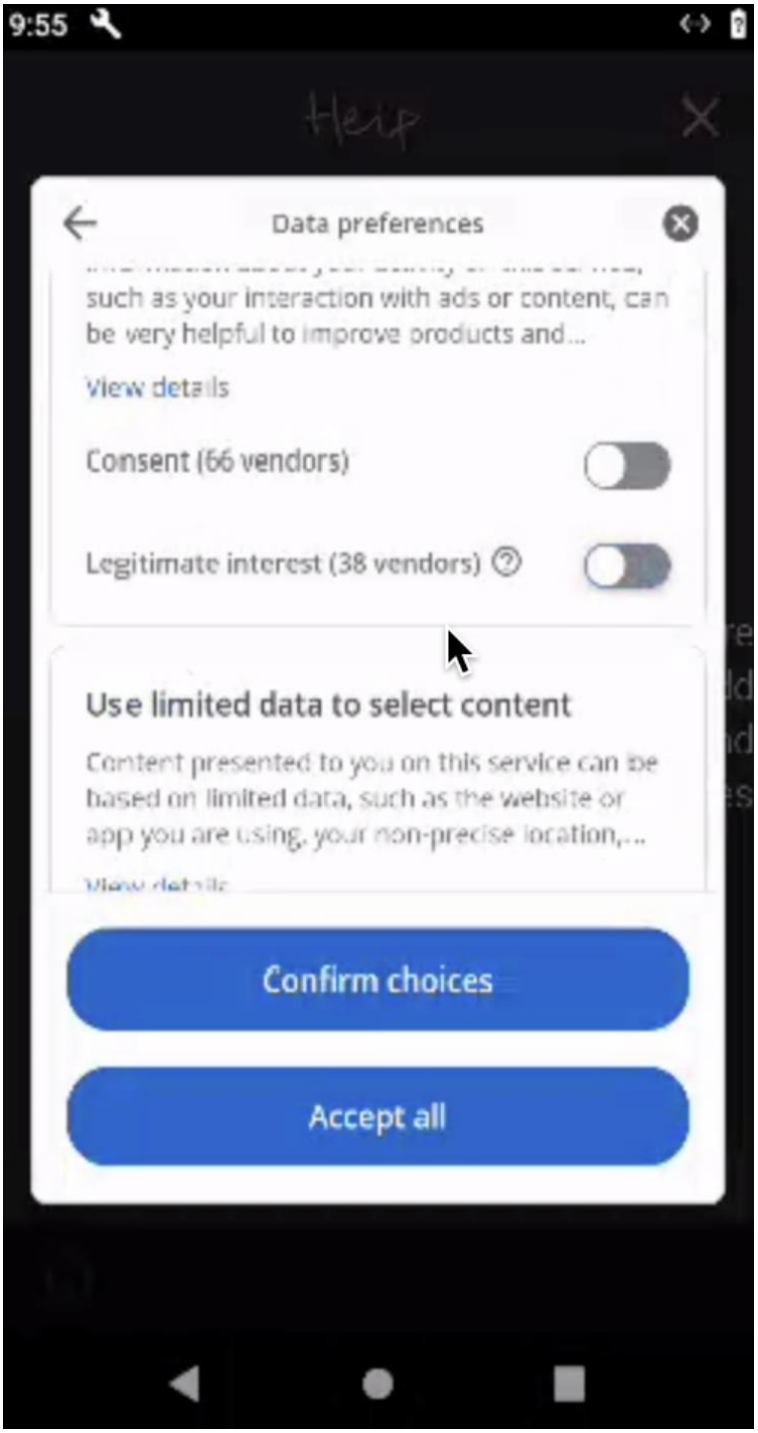


[Figure 3.4. This is also recorded in the response.]

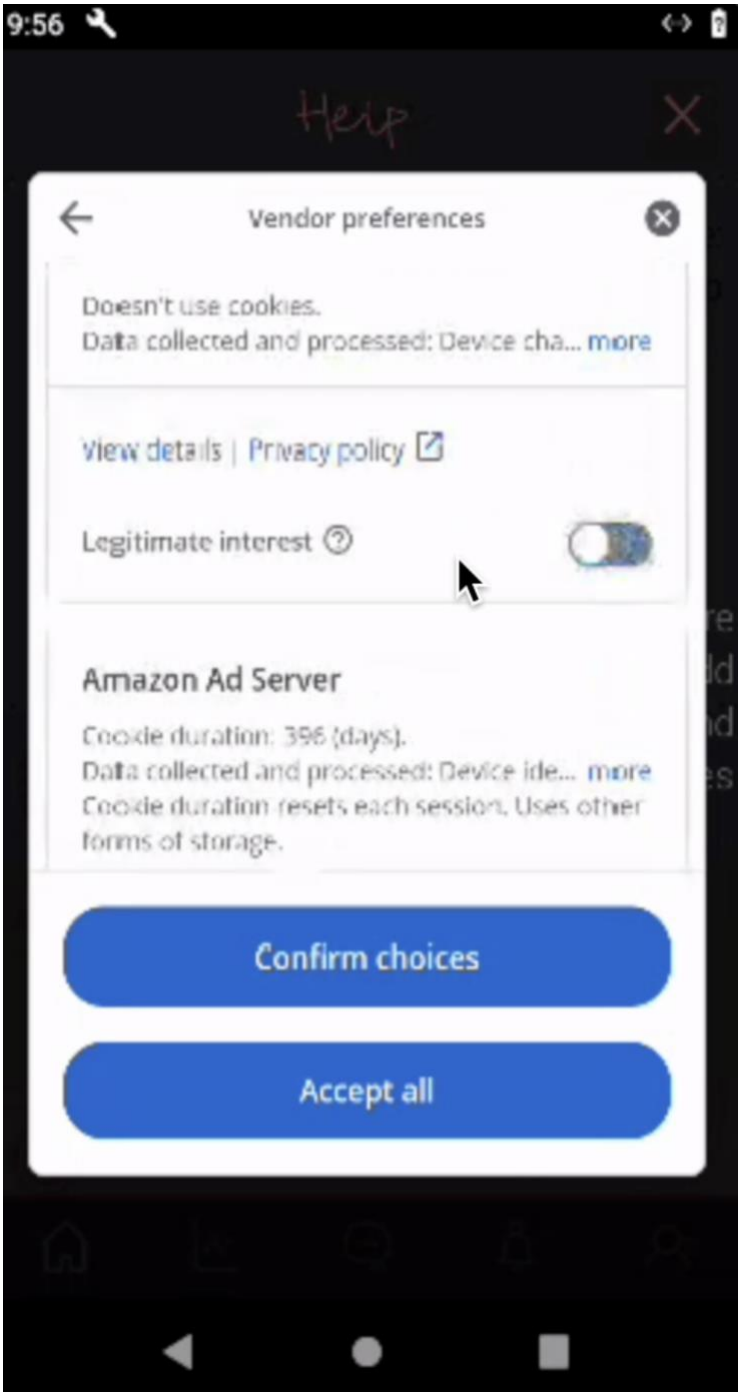
We were then presented with a lengthy advertising network's consent form, which asked the user for their consent to use their personal data for a range of advertising and analytics services, including enabling these vendors to 'store and access information on a device'. We clicked 'Manage options’ and were presented with a long list of purposes for data collection (e.g., analytics) and specific vendors (e.g., Amazon Ad Server, OpenX, Criteo SA, Genius Sports UK Limited, etc.), for which we deselected our consent (it was deselected by default) and deselected 'Legitimate interests' (it was selected by default).



[Figure 3.5. The consent form from the advertising network Maya partners with, for which we clicked 'Manage Options'.]

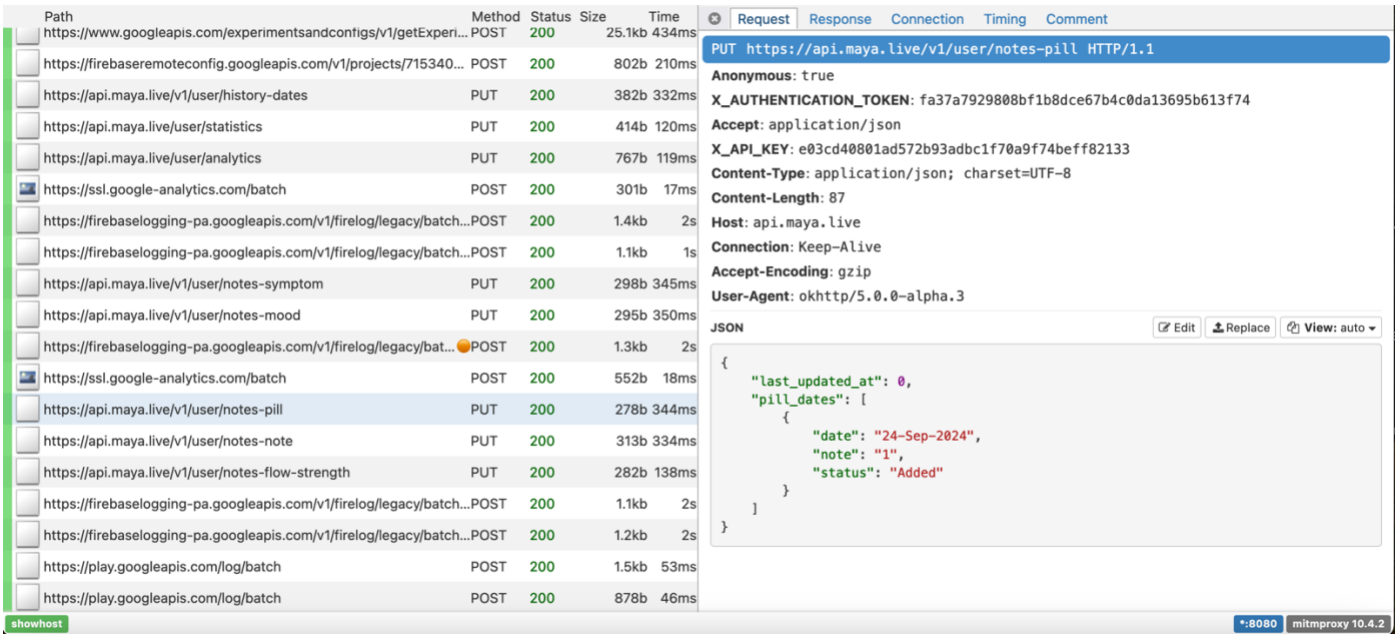


[Figure 3.6. We had to manually deselect all the processing purposes, including for legitimate interest, in the lengthy advertising network's consent form.]

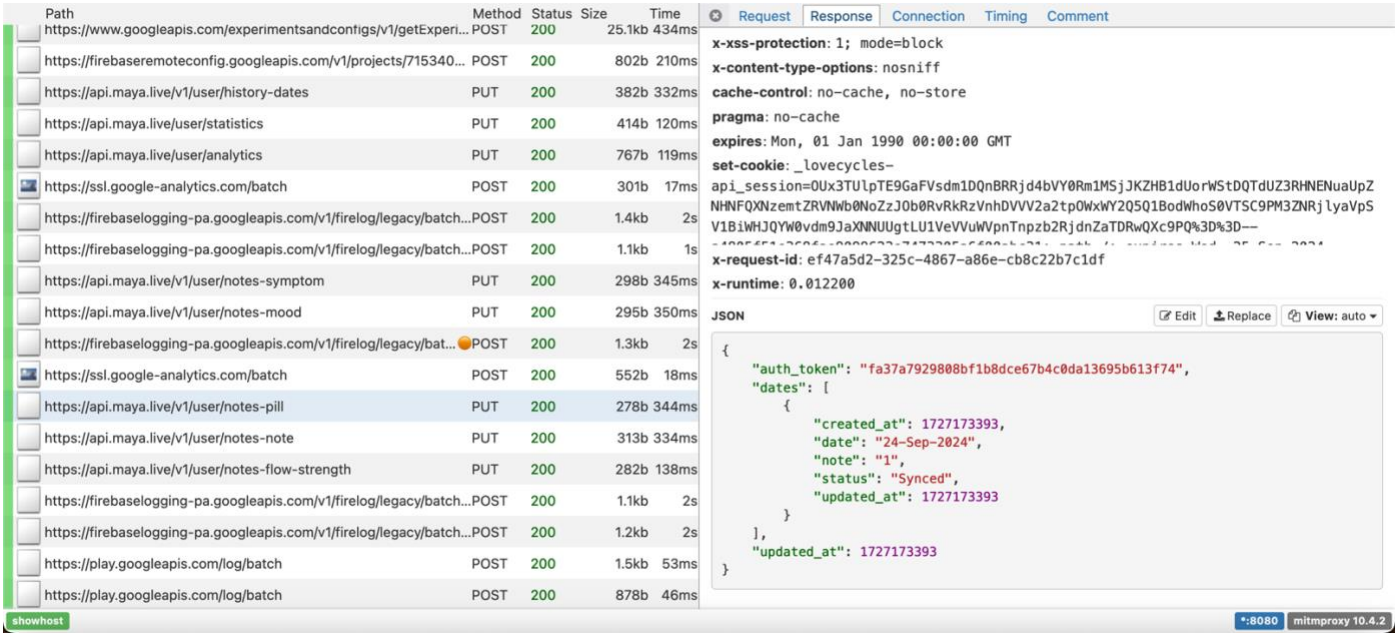


[Figure 3.7. We then had to manually deselect all the vendors in the lengthy advertising network consent form.]

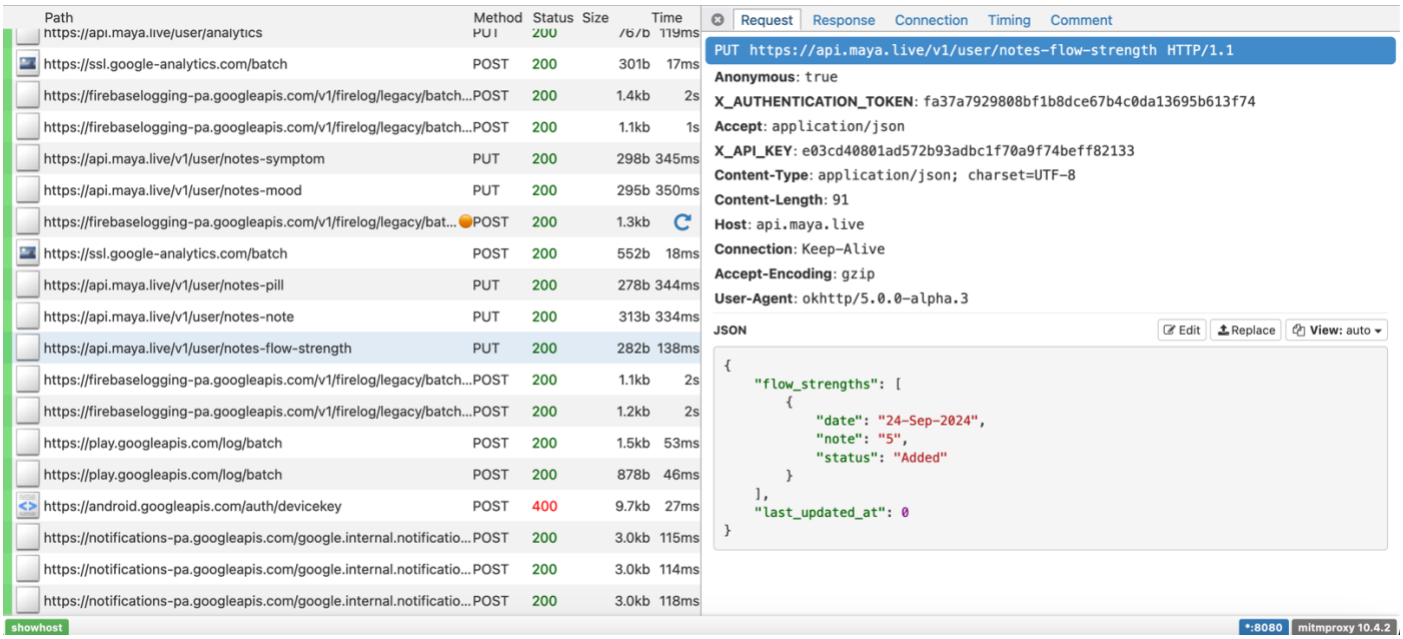
While using the app to input our cycle information over several sessions, we saw in the web traffic every instance our inputs were sent to Maya's API, such as our birth control input (Figures 3.8 and 3.9) and flow strength (Figures 3.7 and 3.8). This is likely the app syncing our inputs with our account.



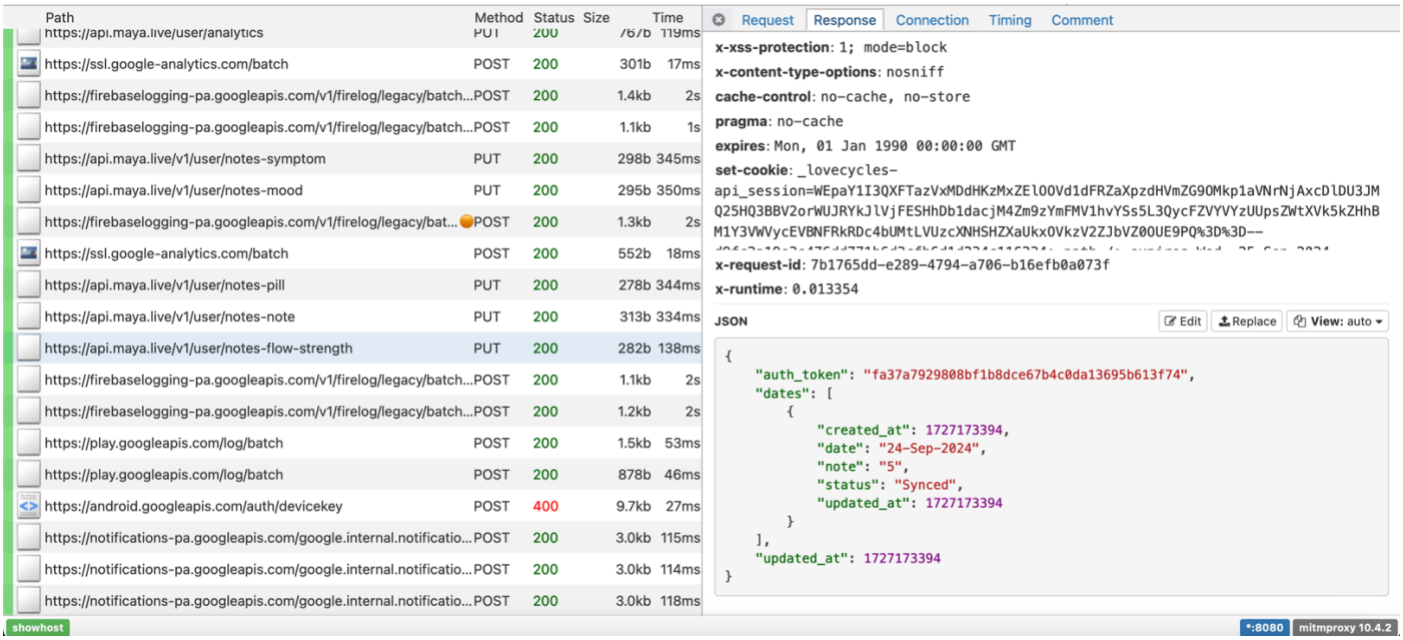
[Figure 3.8. The API requested the date of our pill entry and what the entry was (denoted by '1').]



[Figure 3.9. The API synced this newly logged information.]

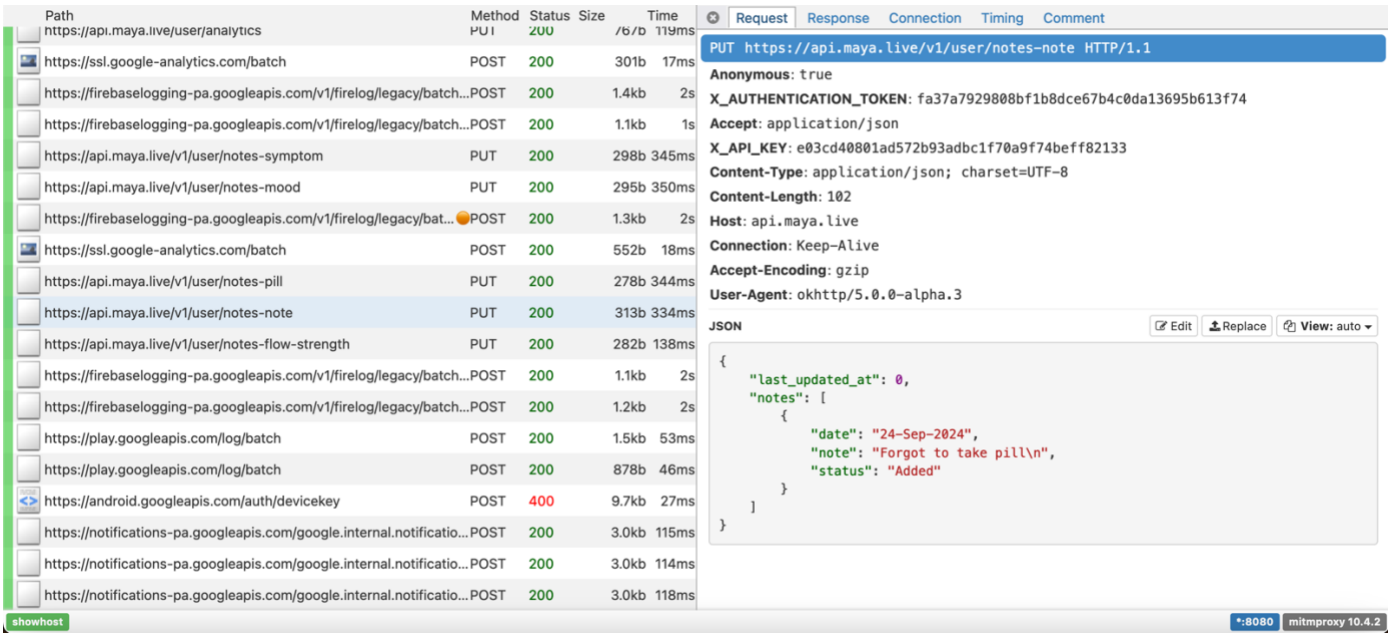


[Figure 3.10. The API requested the date of our flow strength entry and what that entry was (denoted by '5').]

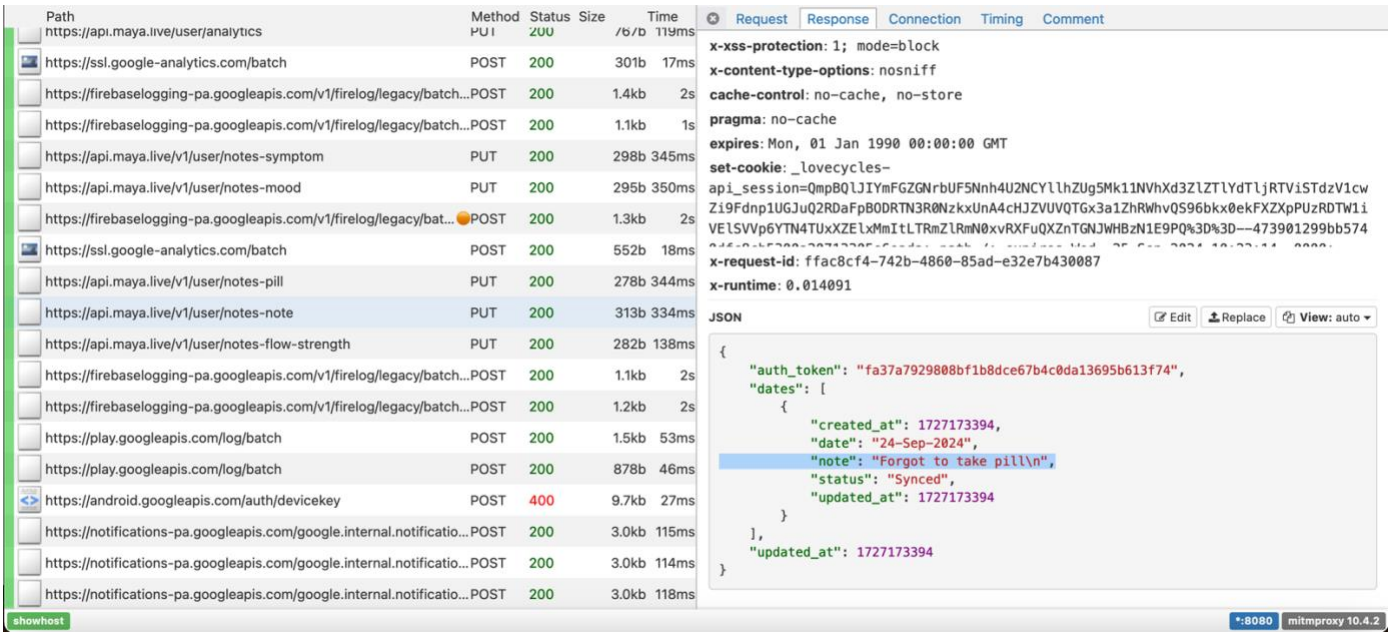


[Figure 3.11. The API synced this newly logged information.]

We also observed that the full text of the personal notes we wrote were also sent to the API:

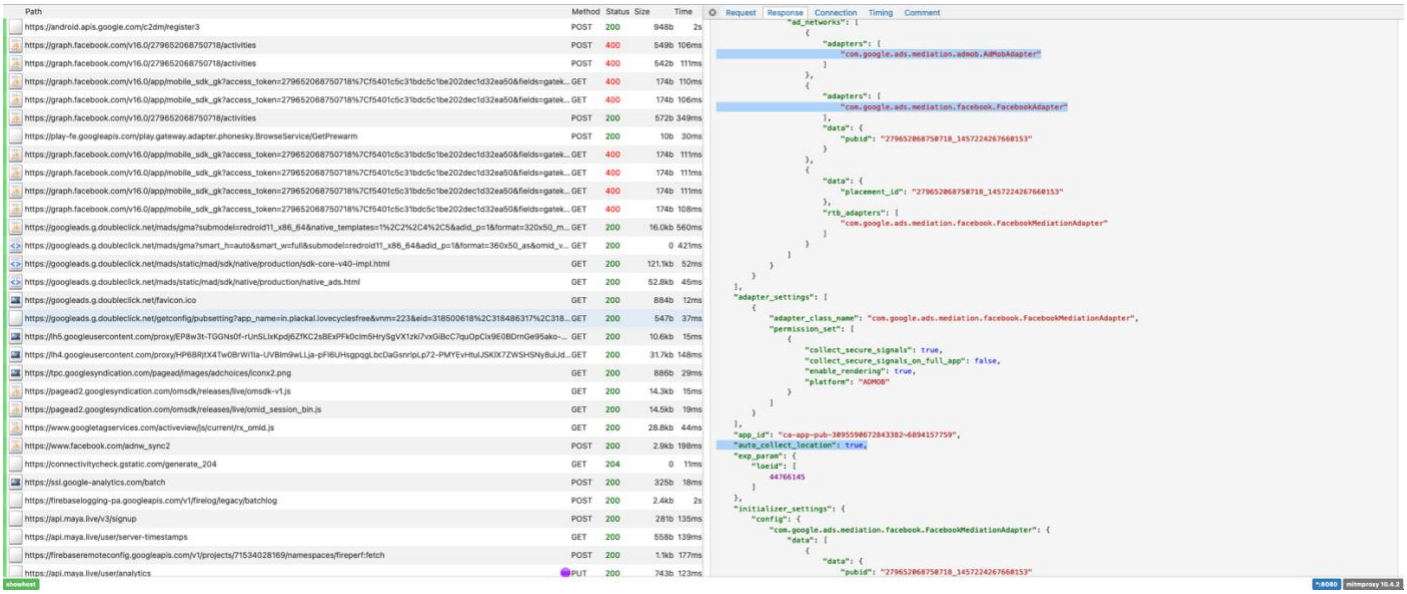


[Figure 3.12. In the request here we can see the text of our note.]



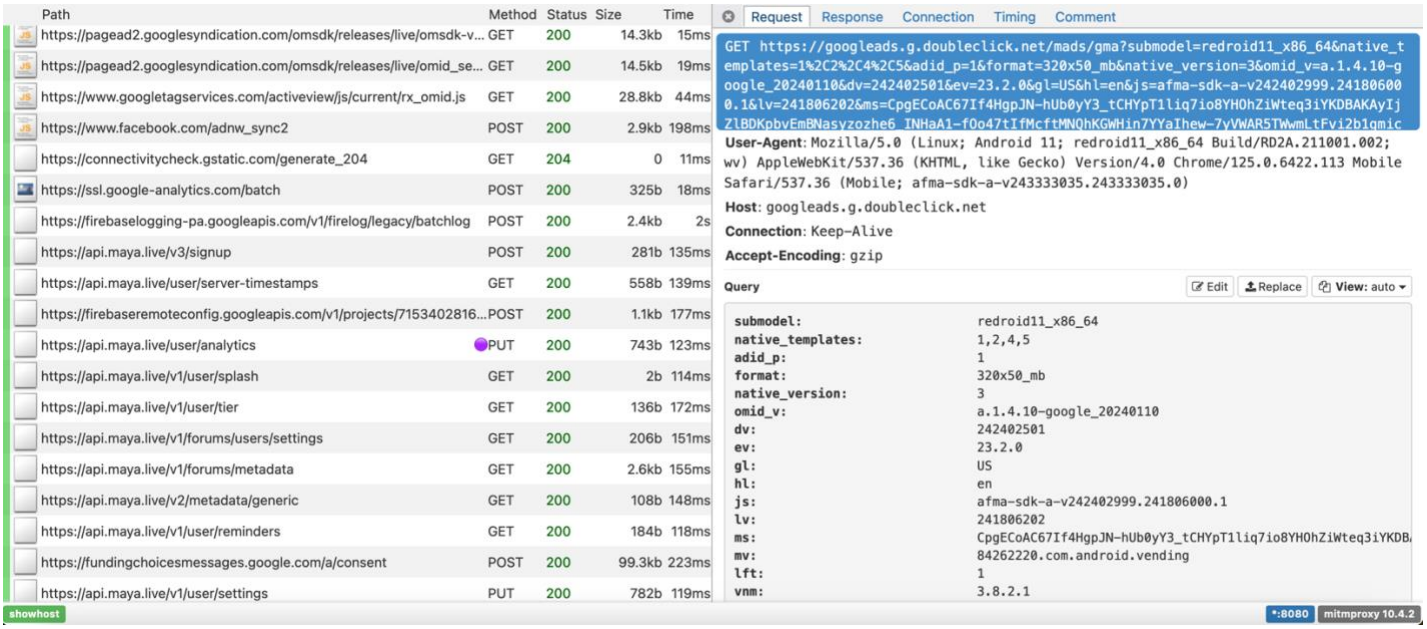
[Figure 3.13. In the response, the API synced the text of this note to the date updated. Note that all this is being communicated to the API, where it might have been the erroneous belief of some users that notes are local to their device and not stored remotely.]

Beyond this first-party data syncing across the API, we also saw numerous third-party advertising SDKs in Maya's web traffic, particularly Google's DoubleClick and Google Ads:

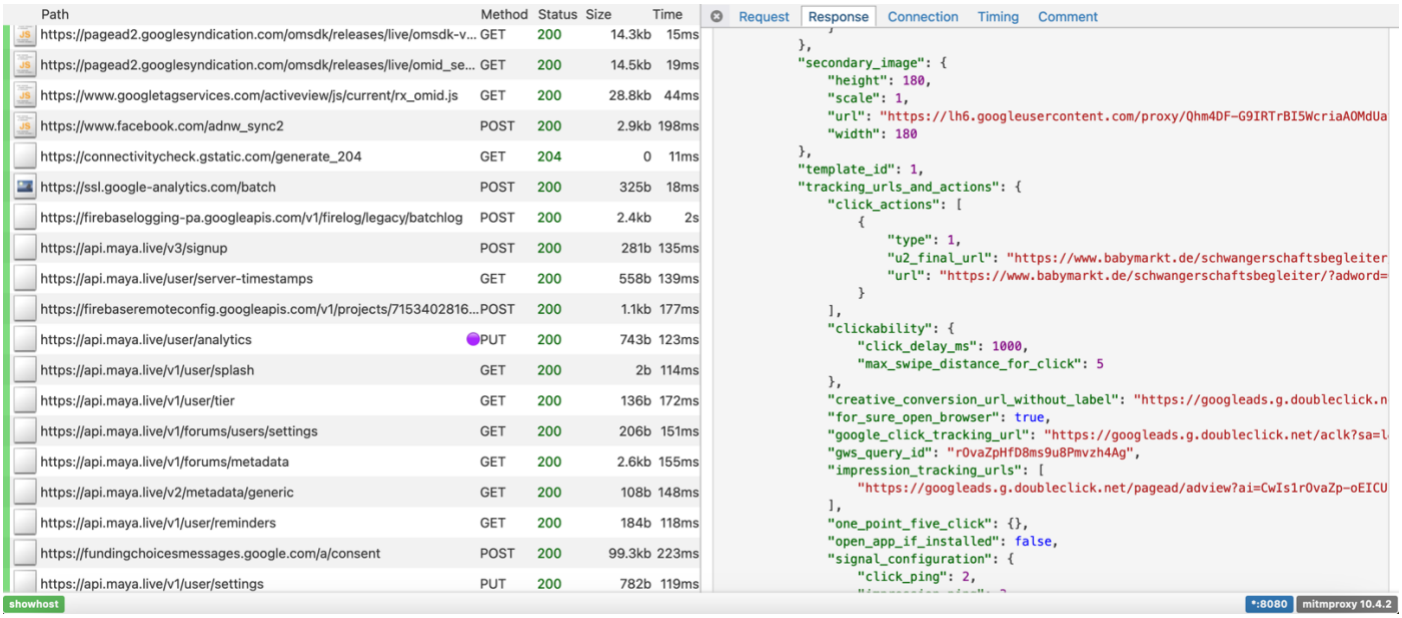


[Figure 3.14. The response from Google Ads with the advertising settings for this app, such as 'auto_collect_location: true'.]

Google Ads also requested device information to know what format of ads to display, and the response outputted an ad, 'babymarkt.de':



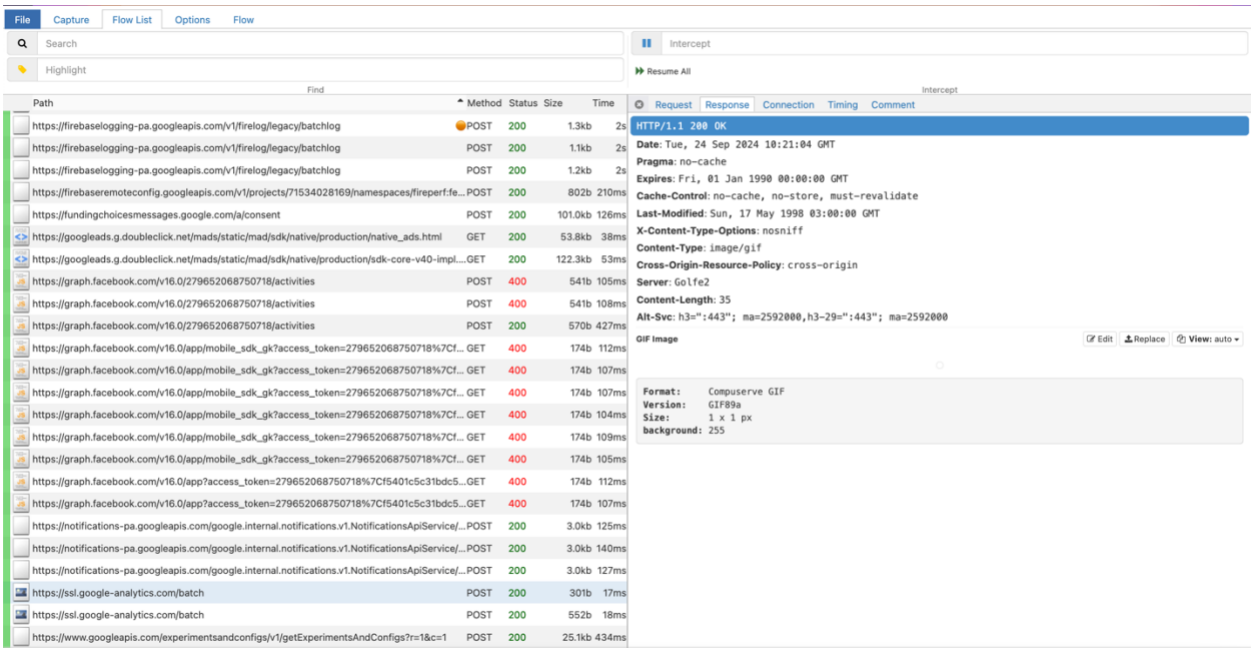
[Figure 3.15. Device data such as the 'submodel' and the 'format' was requested here.]



[Figure 3.16. Based on the requested device information, the advertising network responded with an ad for the company 'babymarkt'.]

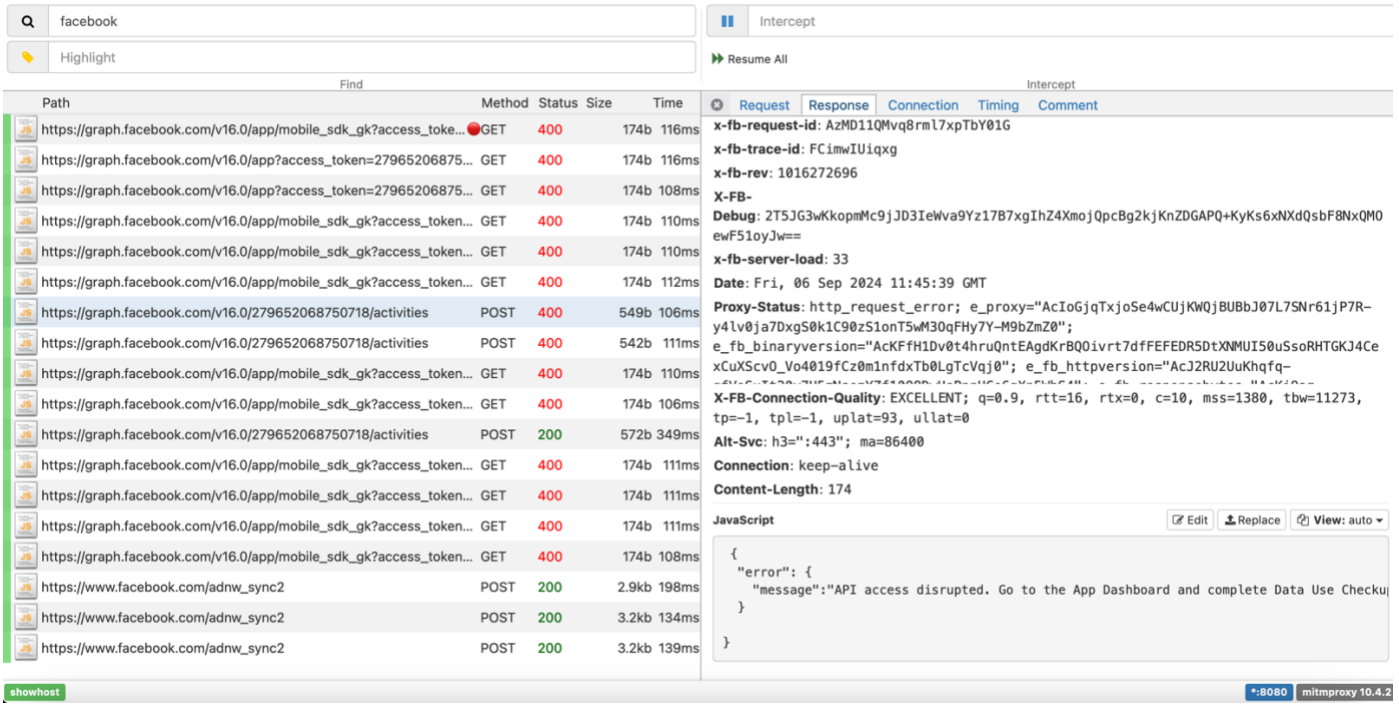
The above ad does not appear to be personalised to the user’s period input data but rather to the device info (e.g., likely customised to the type and operating service (OS) of the device). It is also an ad targeted to the nature of a period tracking app, as it is a baby supplies website. Maya's Privacy Policy discloses that they use third party advertising companies, but it does not name the companies (e.g., Google Ads).

We also noticed below a 1x1 tracker pixel that popped up in the web traffic while we interacted with the app, similar to what we saw in the previous app, that can automatically send information about the user's device and activity to the tracker owner, which appears to be Google Analytics here:

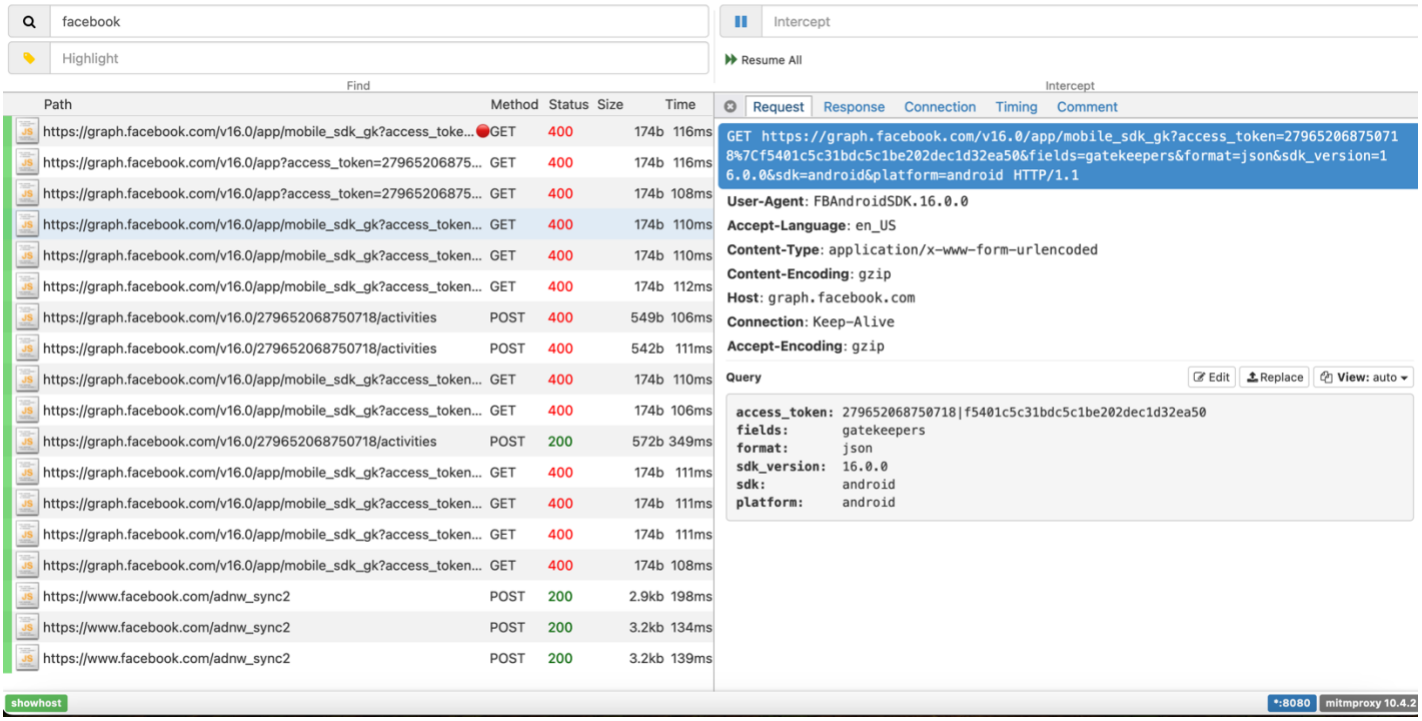


[Figure 3.17. We can see above for a URL path associated with ‘google-analytics’ a 1x1 GIF, which represents a tracking pixel that can track user activity/behavior in the app.]

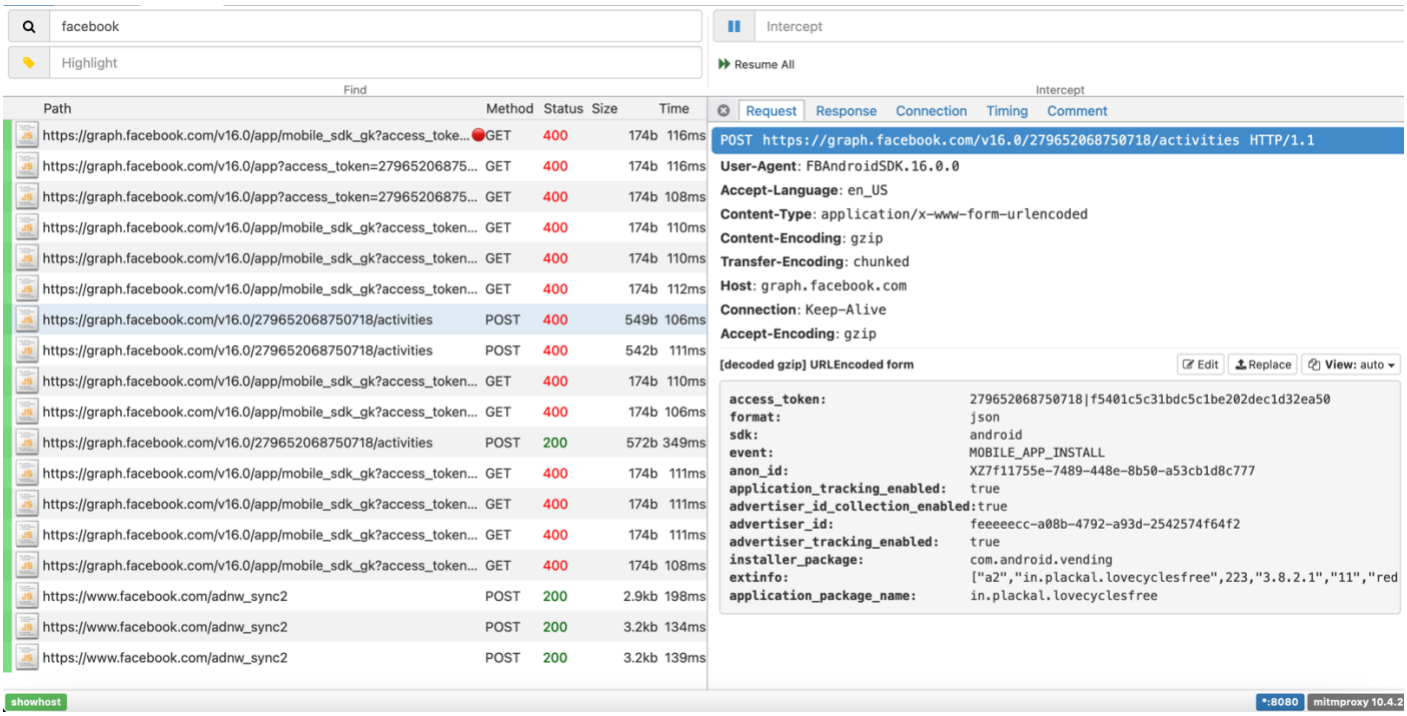
We also observed third party URL paths pointing to Facebook's Graph API (Figures 3.18, 3.19, 3.20) and Facebook's ad network (Figure 3.21). Note that while the Facebook API might be called by an app for log-in integration, we recall that the sign-up page for Maya only had the option to sign up with email, not with Facebook. Additionally, recall that Maya had said in its response to our 2019 research that it had removed the Facebook core SDK (and kept the Facebook Ads SDK), though its pushing of the Facebook log-in perhaps suggests that the SDK is still integrated in the application itself (while it is Facebook that denies the requests to its API).



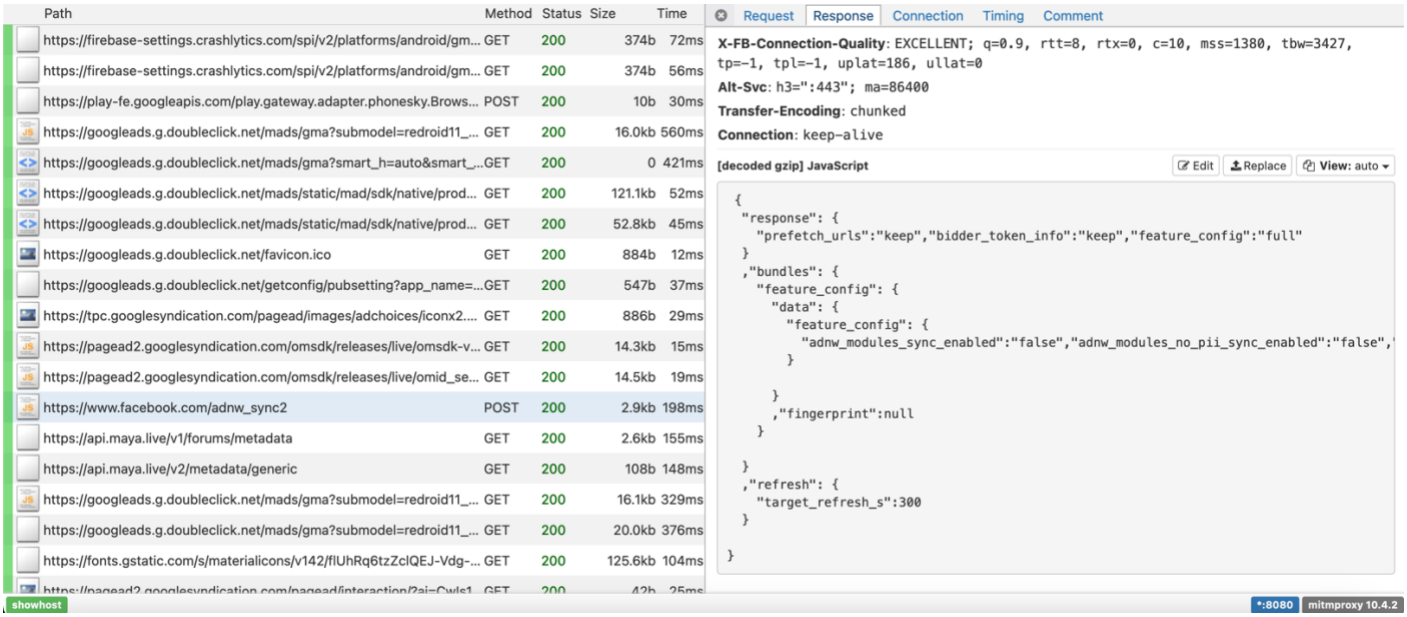
[Figure 3.18. Facebook denies Maya's calls to integrate the Facebook API.]



[Figure 3.19. The requests for these rejected calls contain the field 'gatekeepers'.]



[Figure 3.20. We see 'advertiser_tracking_enabled: true' for calls to the Facebook API, which suggests Maya is attempting (unsuccessfully) to integrate Facebook for advertising purposes.]

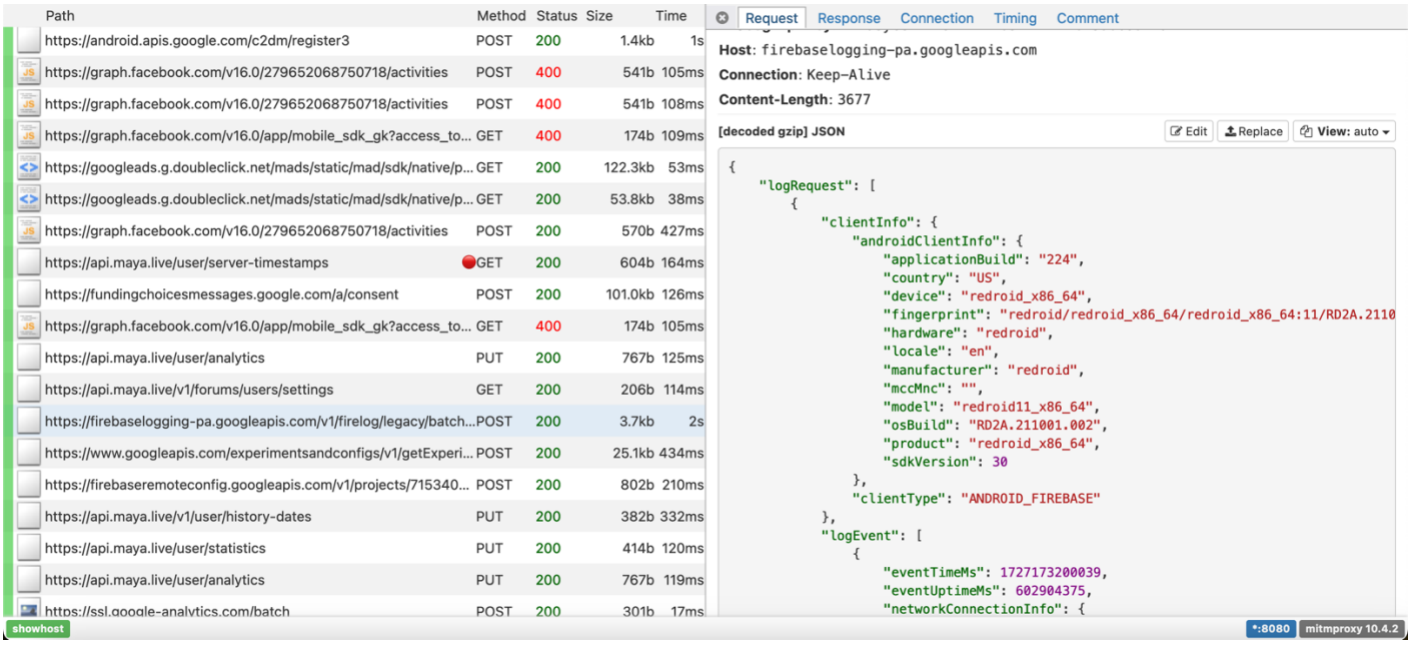


[Figure 3.21. We can see here that Maya had called Facebook's ad network, but Facebook responded false for all the bidding entries, likely because the Facebook API rejected Maya's calls.]

Above, we see that as Maya consistently makes calls to Facebook's Graph API, Facebook consistently rejects these API calls due to the 'gatekeepers' check. Even though Facebook is disrupting access by Maya to its API, Maya still appears to be sending requests (or at least

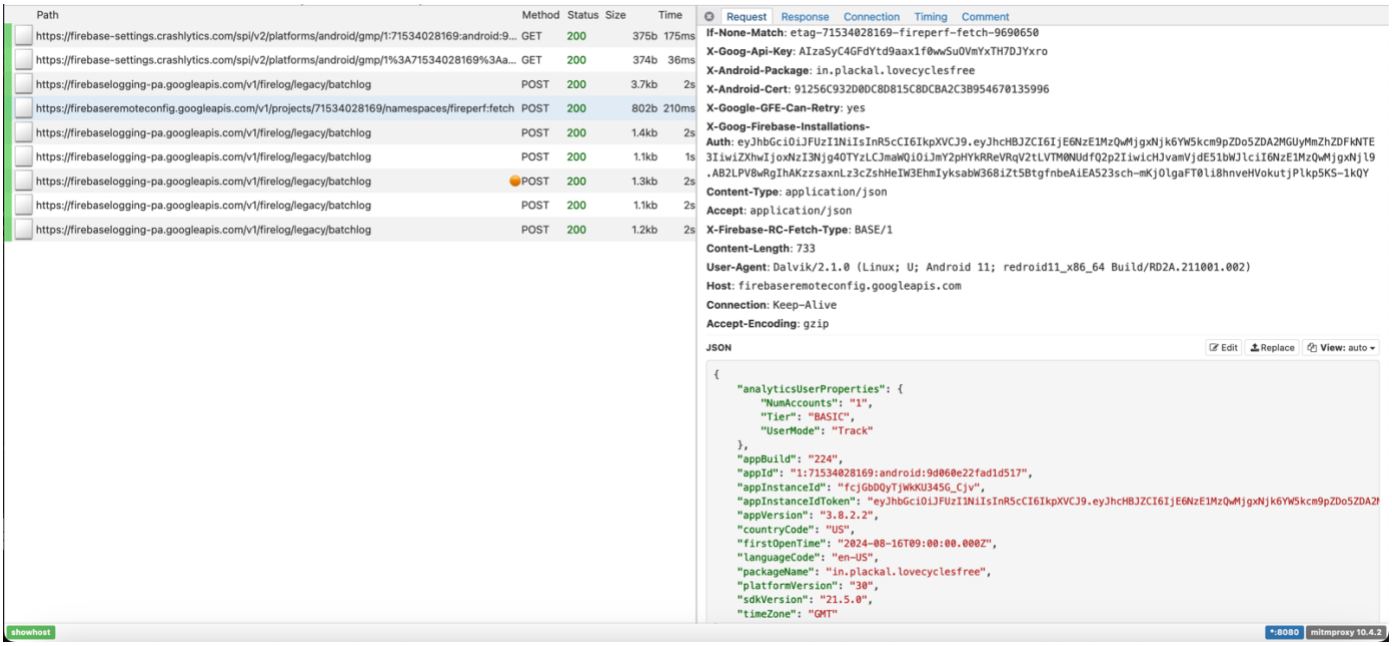
leaving this Facebook configuration in the app). On the one hand, user input data is not being sent to Facebook as it had in our previous investigation; however, we are still observing calls to third parties where there shouldn't be, and these calls are nonetheless sending device data (i.e., serving the ‘babymarkt’ ad).

Maya also appeared to integrate Firebase, which requested a range of data about the device:



[Figure 3.22. See device information in 'androidClientInfo'.]

We also noticed Firebase requested specific information about the app itself, such as 'Usermode: Track' and app version information, which is likely collected for app-specific monitoring and performance analytics:



[Figure 3.23. The variable 'UserMode: Track' is likely from our response in the onboarding questionnaire.]

We can speculate that collecting the above information may be for generating aggregated analytics for the developer (i.e., how many Android devices are using the app to 'track' menstruation), though this includes additional risks for consideration we will discuss further below, not to mention Firebase was not disclosed in the app's rather vague Privacy Policy.

Period Tracker by GP Apps

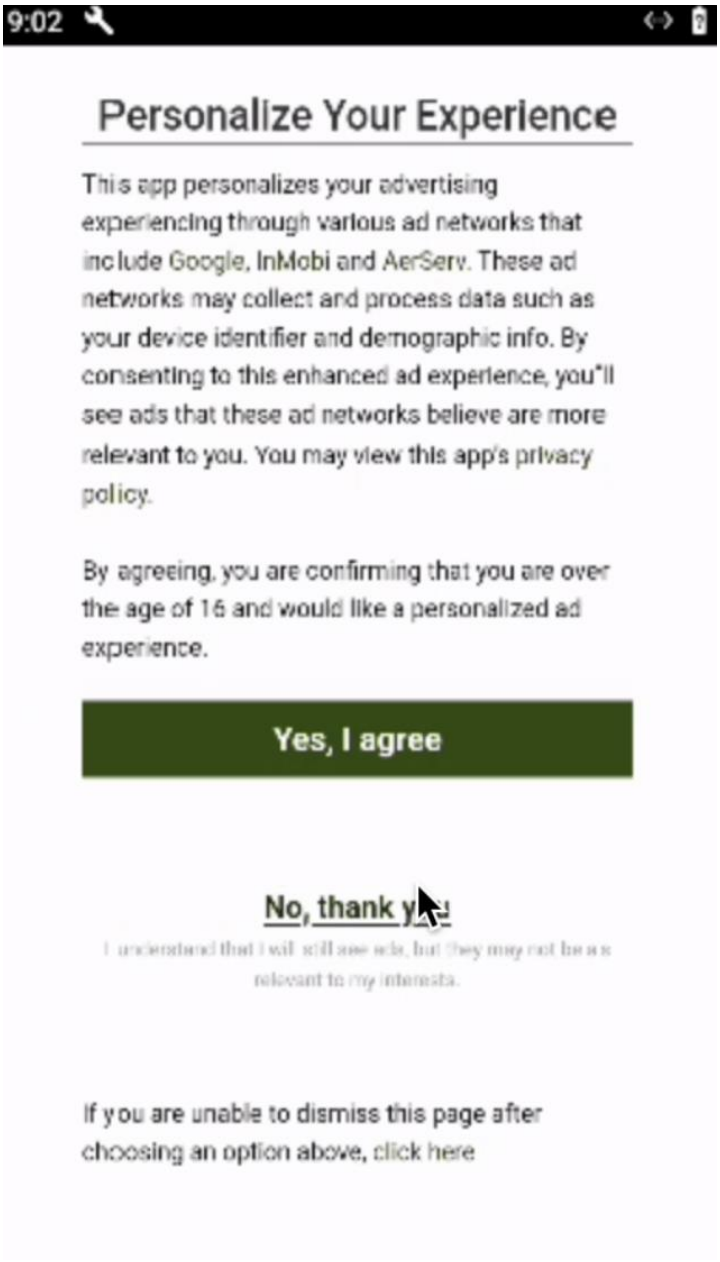
Period Tracker by GP Apps is another popularly downloaded app we previously looked at in 2019. In our original research, we determined that this app did not appear to share any user input data with Facebook.

This time, we examined the third parties that the app appeared to integrate and what kind of data was being shared with these third parties, as well as what user data the app was storing on its own or external services. It's worth noting that the developers of the app released a statement following the overturning of Roe v. Wade that:

> *'We are adamantly opposed to government overreach, and we believe that a hypothetical situation where the government subpoenas private user data from health apps to convict people for having an abortion is a gross human rights violation. In such a scenario, we will do all we can to protect our users from such an act. We would rather close down the company than be accomplice to this type of government overreach and privacy violation.'*

The statement also explained that users could use the app without an online account, and that, in this case, their data would be stored only locally on their device, rather than backed up to a cloud-based account.

We indeed got started on the app without having to create an account and without having to complete any onboarding questionnaire. Then, we were presented with a consent pop-up for the processing of user data for networks like Google, InMobi and AerServ. We clicked 'No, thank you' (Figure 4.1), which confirmed that 'Ad networks won't collect data to personalise advertising for you in this app' (Figure 4.2).



[Figure 4.1. Screenshot of the in-app consent page.]

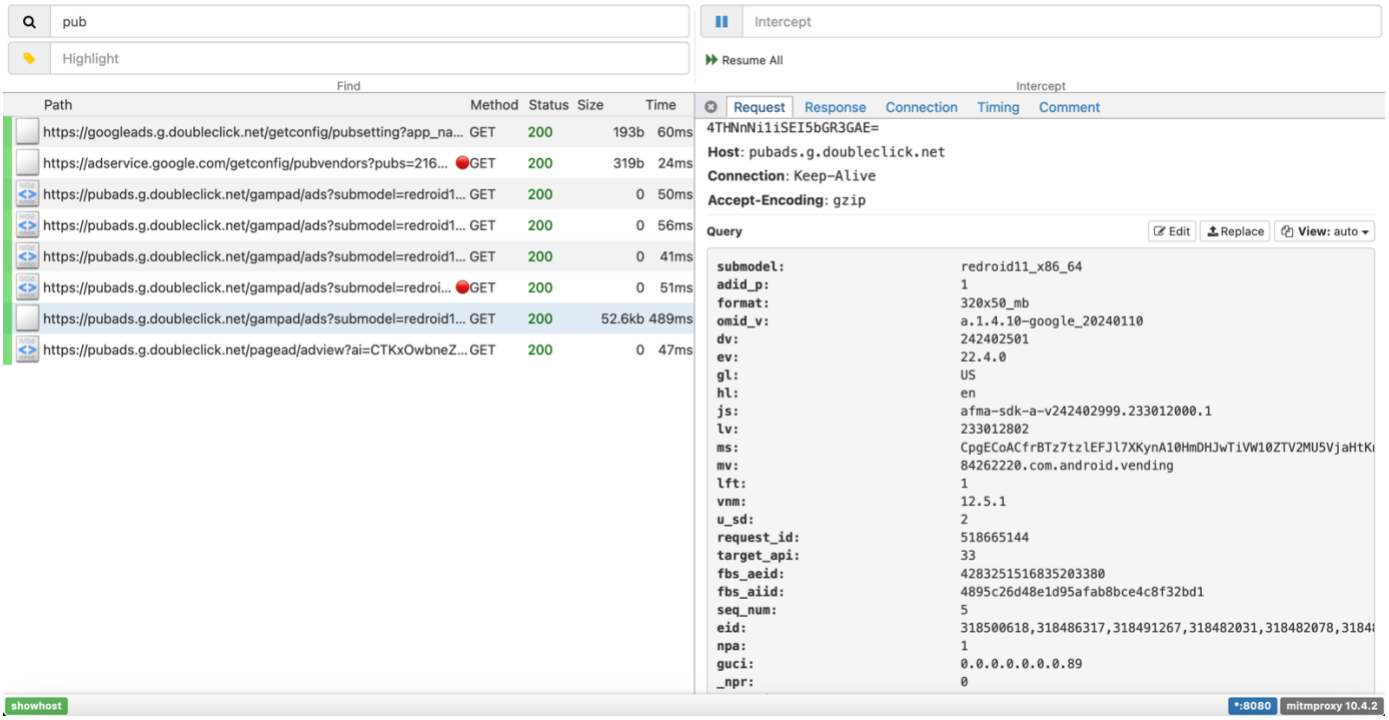
[Figure 4.2. Screenshot of the app's response to our 'No, thank you' selection.]

We were then directed to the cycle dashboard, where we began inputting our cycle data. Throughout our use, we did not observe our input data being sent across the web traffic to any API, which perhaps suggests a confirmation of the app's claims that user input data would be stored locally on the device only and not on the cloud for users who choose not to create an account.

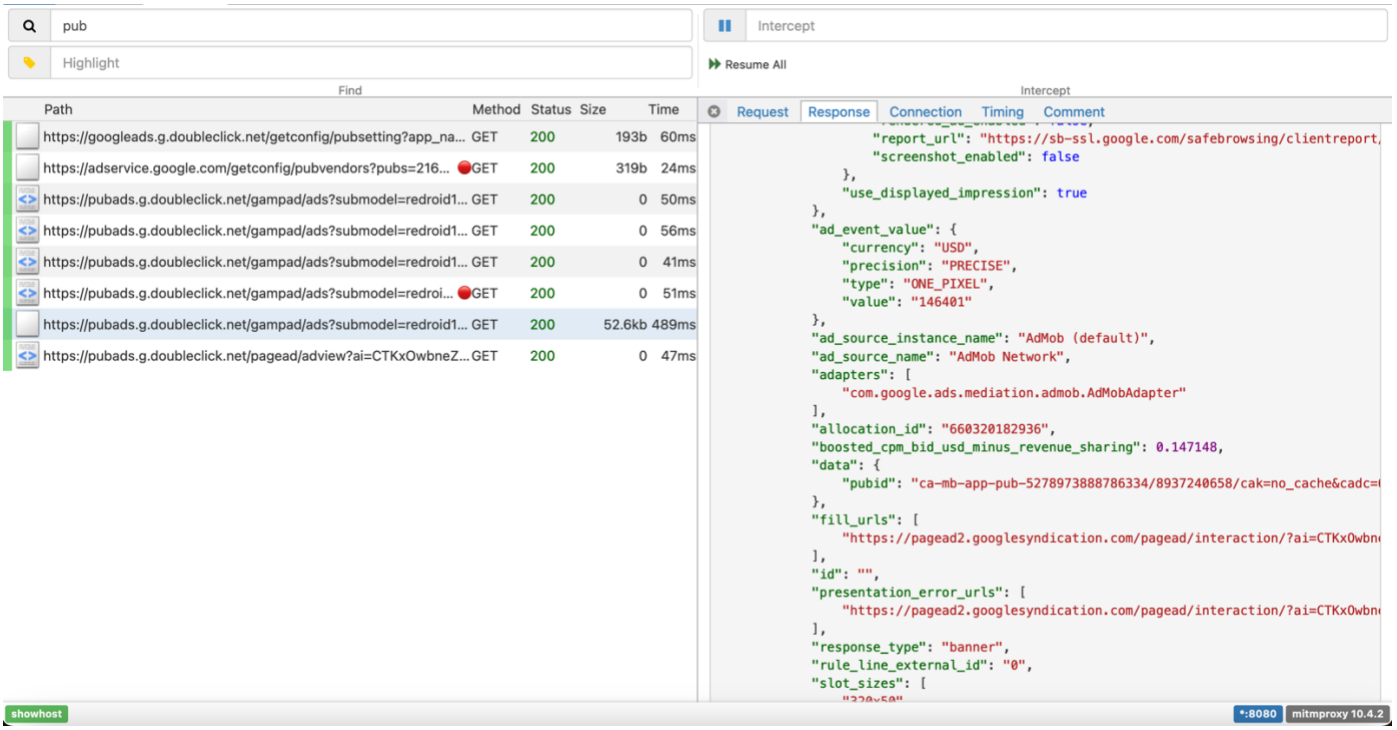
We nonetheless noticed some calls to third-party advertising and analytics SDKs. Most of the ads SDKs that we saw, such as Nexage and Moat Ads, were not properly functional and returned errors in the web traffic. We did some digging and discovered that these advertising networks no longer exist in the same state as they appeared in the web traffic; Nexage was acquired by AOL Advertising, which has since been absorbed into the larger Yahoo Advertising network, and Moat Ads, formerly an AOL customer, has since been acquired by Oracle.

It appears that this period tracker app has left outdated ads integrations in its implementation. While personal data does not appear to be shared in these ad requests, and these third-party URL paths do not appear to be functional SDKs anymore, it is nonetheless concerning that the app's developers have not removed these outdated calls. There is the risk, for example, that these URLs could in theory be hijacked by malicious actors (a threat called ‘broken link hijacking’).

Other ads SDKs we observed were advertising network calls through Pub Ads (acquired by Google), which requested device data and responded with a variety of ad placement information:



[Figure 4.3. Pub Ads requested generic device information, such as 'format'.]

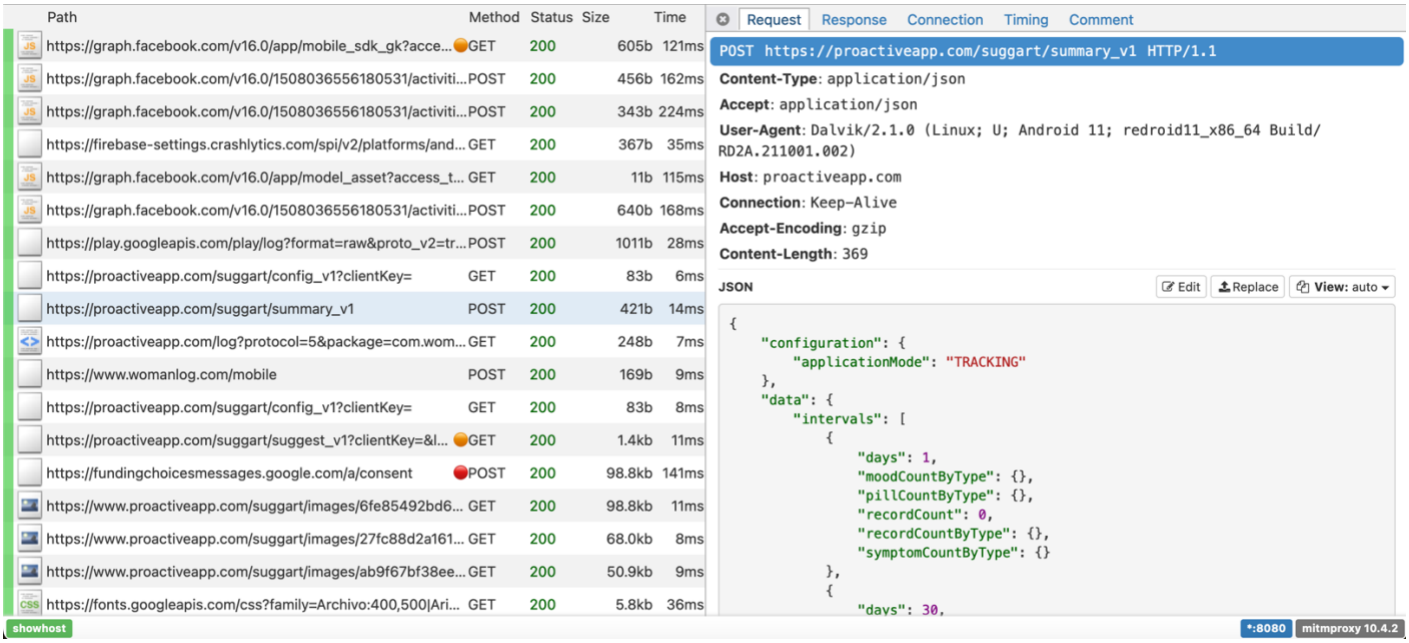


[Figure 4.4. Pub Ads then responded to that request with a variety of information for the ad placement, such as the device 'type: ONE_PIXEL' and the 'ad_source_name: AdMob Network'.]

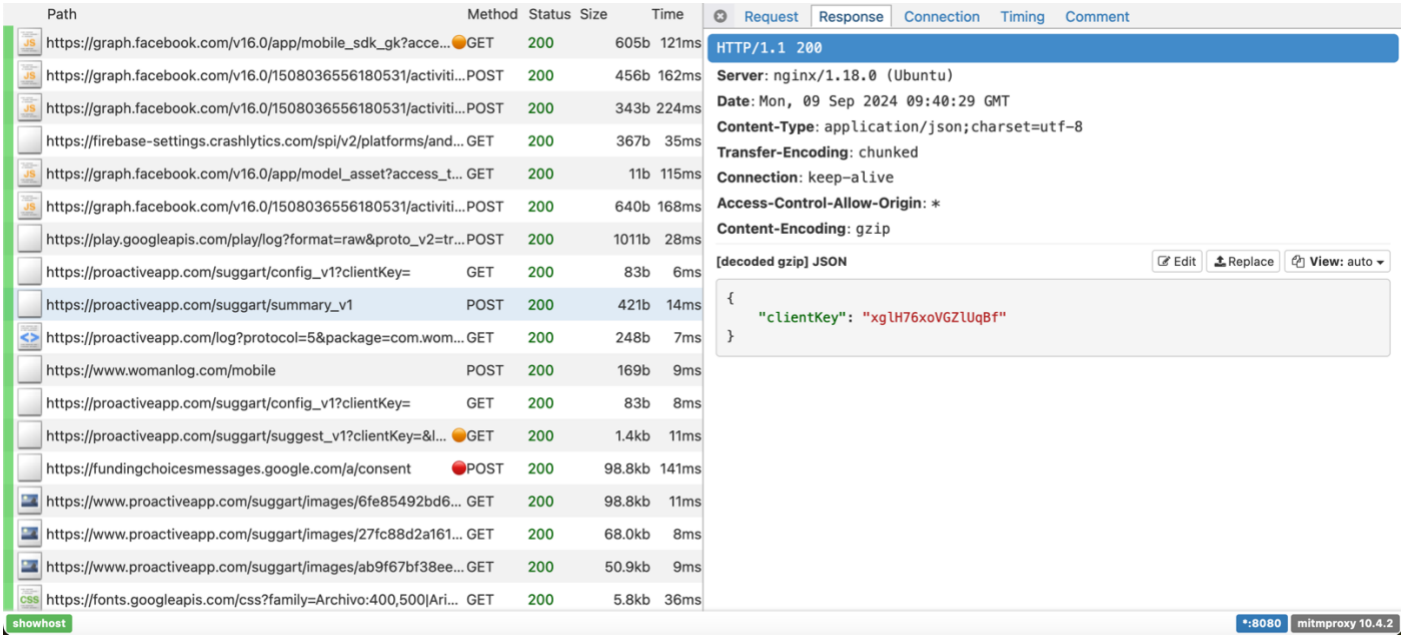
Neither Pub Ads nor any of the above advertising SDKs (Moat, Nexage) were named in the app’s Privacy Policy; the only mention of advertising SDKs in the privacy policy was a general statement that automatically collected information (e.g., device type, IP address, OS, other device data) would be sent to third party advertising networks and analytics companies (not specified by name). As the app claims in its public statement, it indeed does not appear to store any user input data (at least as far as our DIAS environment can see). The app does appear to utilise some third-party ads SDKs like Pub Ads, as well as some outdated or non-functional ads SDKs.

WomanLog

The WomanLog app, developed by Pro Active App SIA, is a Latvia-based period tracking app with over 10 million downloads that features an 'Intelligent Assistant' chatbot (more on this below). To get started on the app, we completed a short onboarding questionnaire about which app mode we intended to use (e.g., standard) and the length of our cycle and period. Our answers to these questions were sent across the web traffic to the app developer's API:



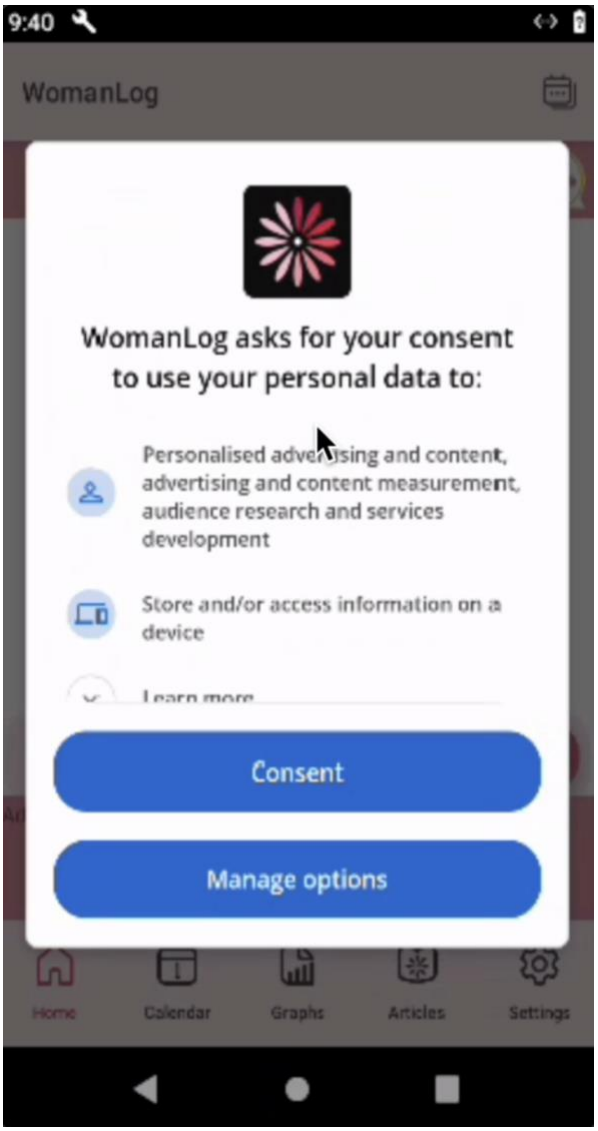
[Figure 5.1. The developer API, represented by the 'proactiveapp.com' URL, requested the app mode ('Tracking').]



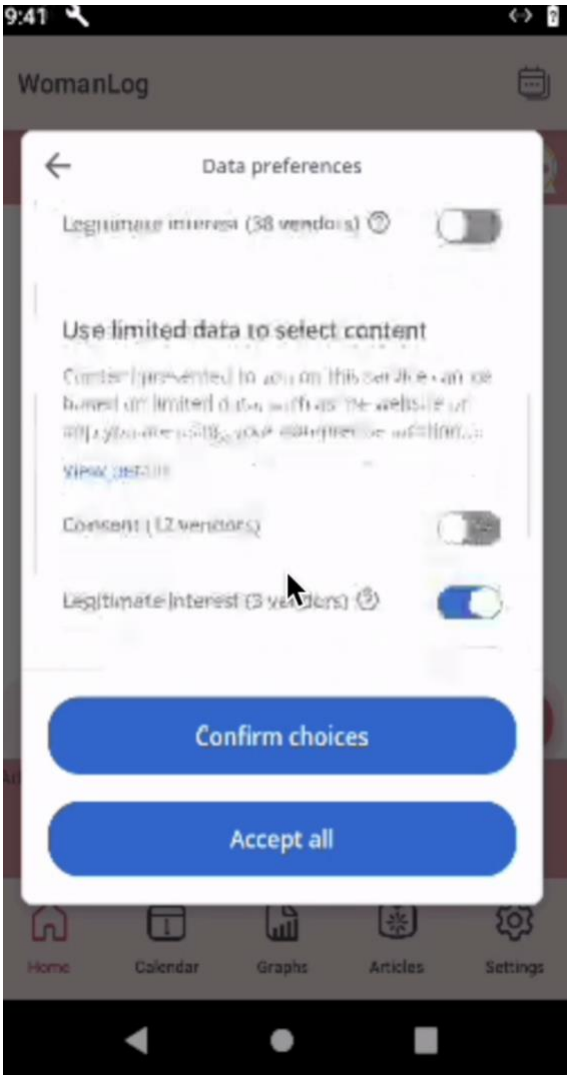
[Figure 5.2. The response assigned a unique 'clientKey' to this particular user.]

We were also asked to provide additional information about birth date, weight and height, which was optional but recommended (we skipped this).

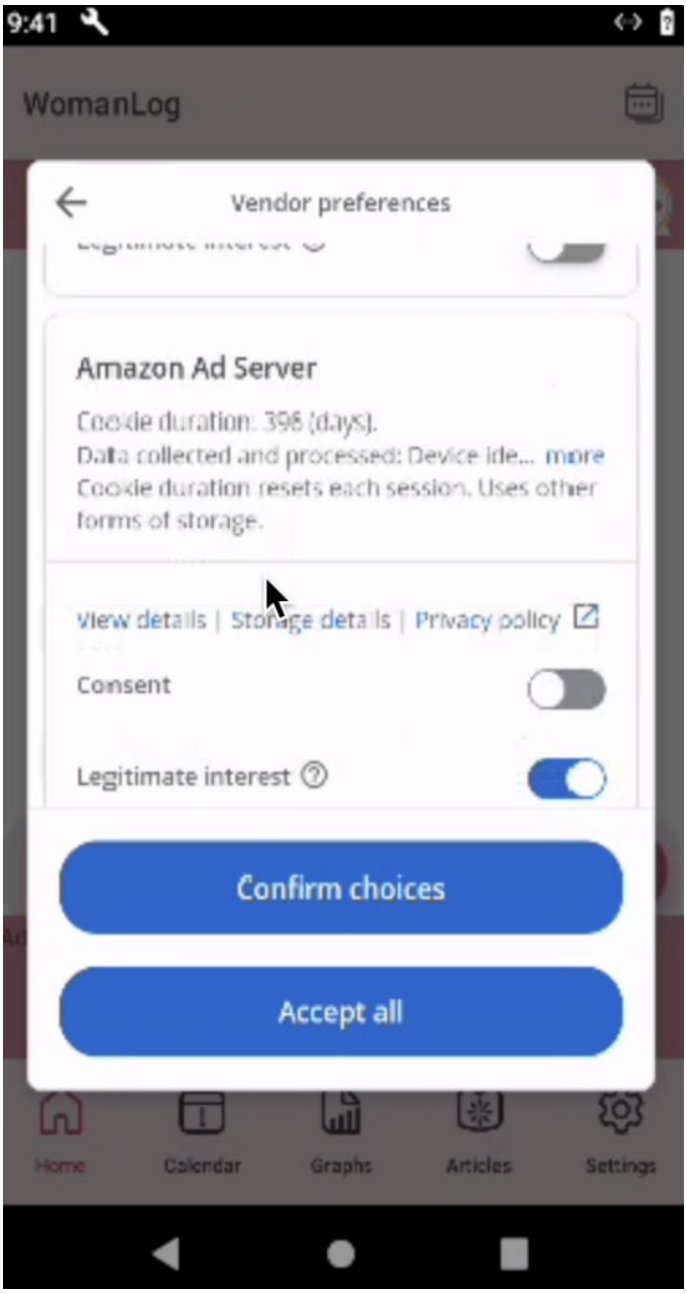
After the questionnaire, we were presented with an advertising network's lengthy consent form similar to Maya’s, for which we manually deselected our consent to the listed data-sharing activities and vendors for the purposes of personalised advertising and analytics:



[Figure 5.3. The consent form pop-up from the advertising network for which we selected ‘Manage options’.]



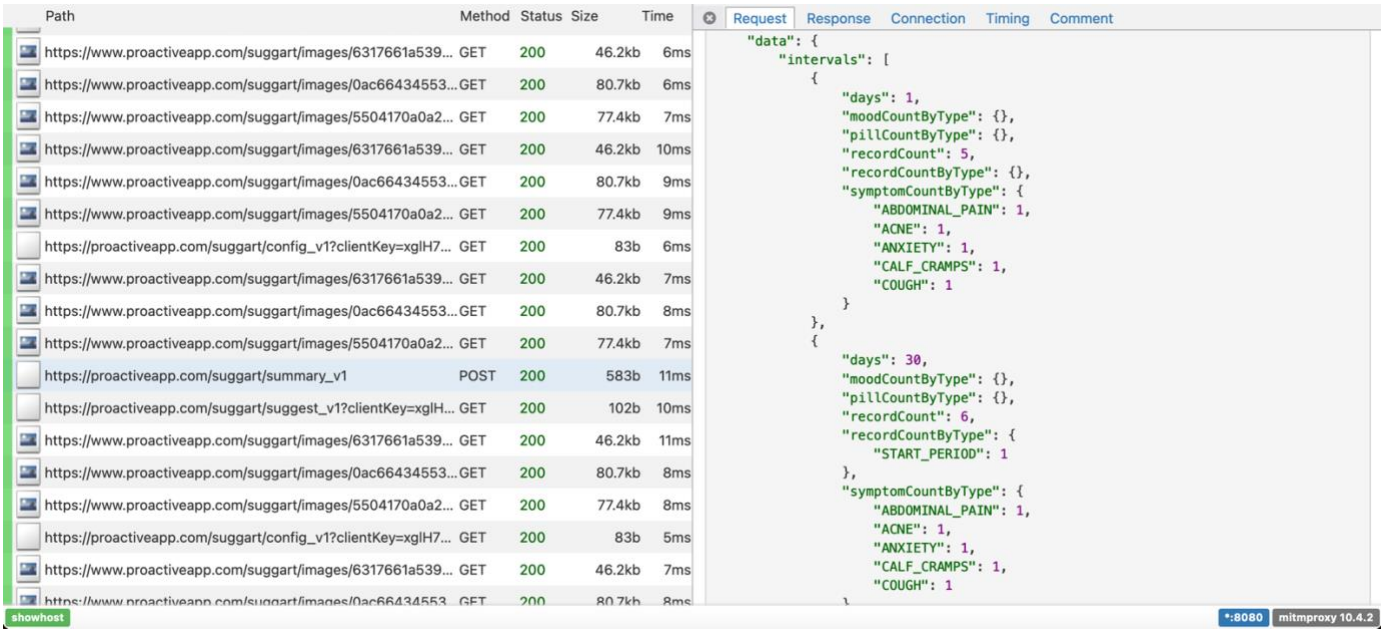
[Figure 5.4. We manually deselected our consent, including for legitimate interests, for the processing purposes in the advertising network’s consent form.]



[Figure 5.5. Screenshot of the vendors (e.g., Amazon Ad Server, Pubmatic, Inc.) requesting data processing permission.]

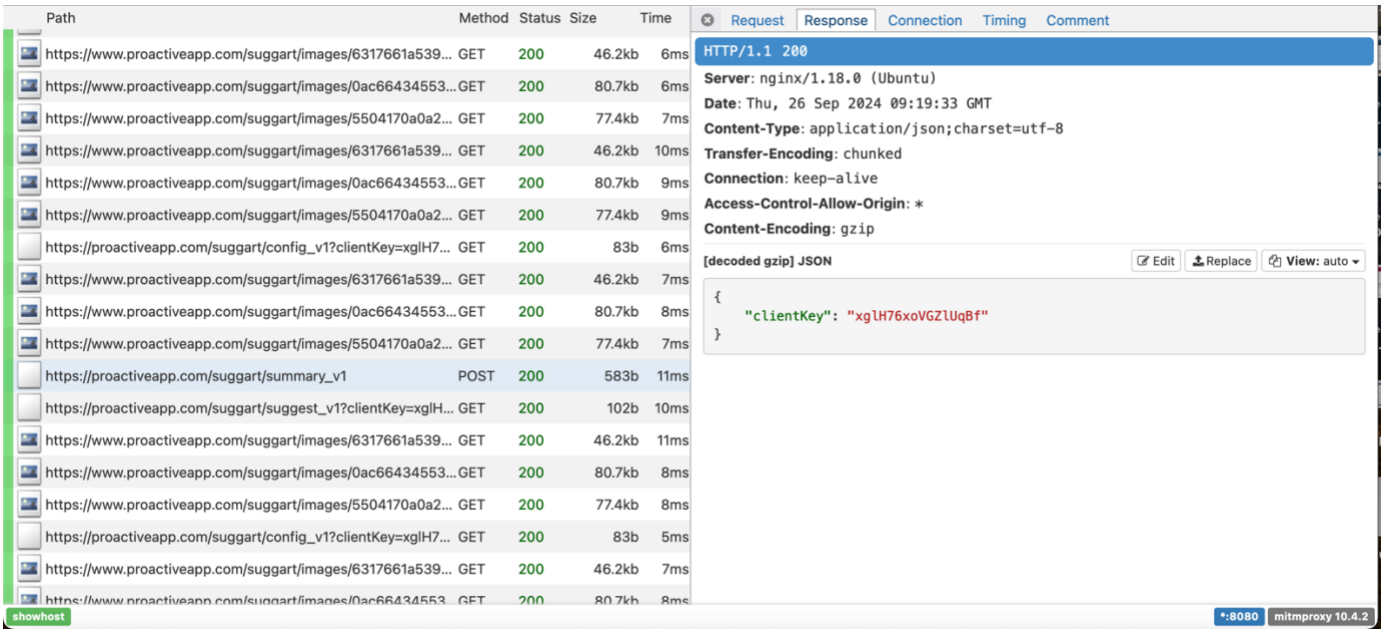
After manually deselecting all these options, we were directed to the cycle dashboard to utilise the app, without having to create an account.

As with our onboarding questionnaire responses, every time we inputted information about our cycle, this data was sent to the app developer's API. This included information such as which medication the user took, which symptoms they logged and the start of their cycle:



[Figure 5.6. We can see the API requesting our inputted data, for which we'd marked several 'symptomCountByType' entries.]

For all of these entries we inputted, the web traffic response returned the user's uniquely identifiable 'clientKey' (Figure 5.7), which links all our inputs with this uniquely identifiable key:

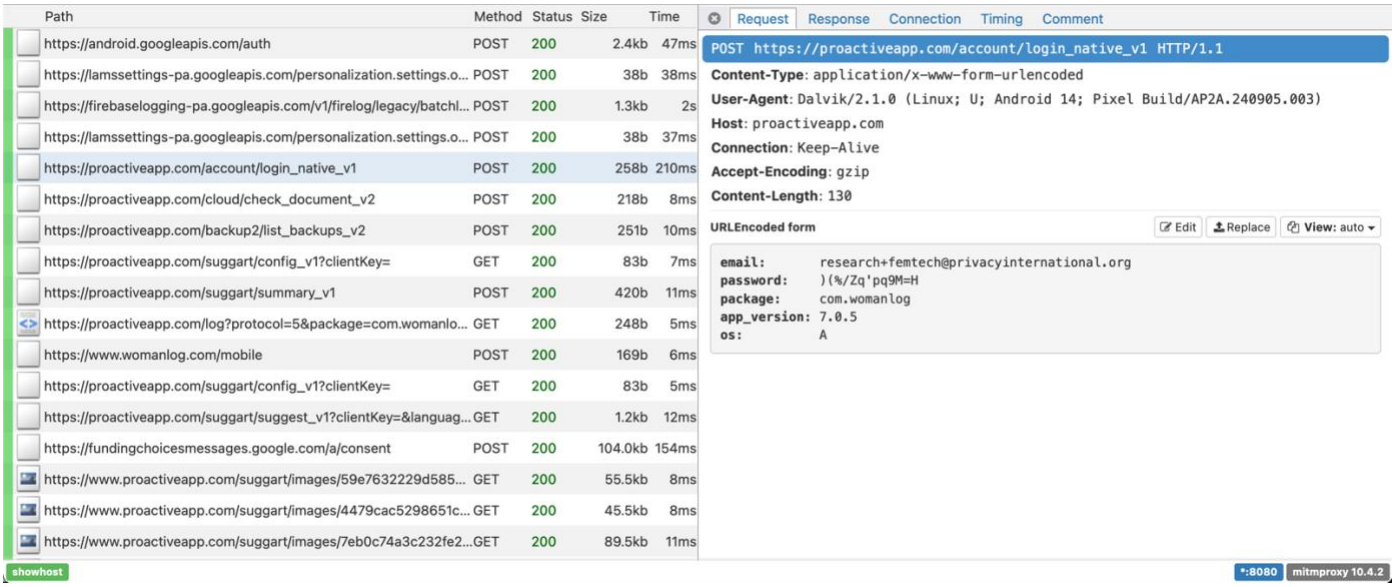


[Figure 5.7. This is the same 'clientKey' as the one assigned from the start in Figure 5.2.]

We note that in response to our findings, WomanLog stated that all communication between the app and their servers is HTTPS encrypted.

Next, we tested the Intelligent Assistant feature, which is a paid-for period prediction and chatbot service powered by OpenAI (note the disclosure of OpenAI was only provided separately in WomanLog's Privacy Policy and not in the app itself). Using this feature required us to create an account.

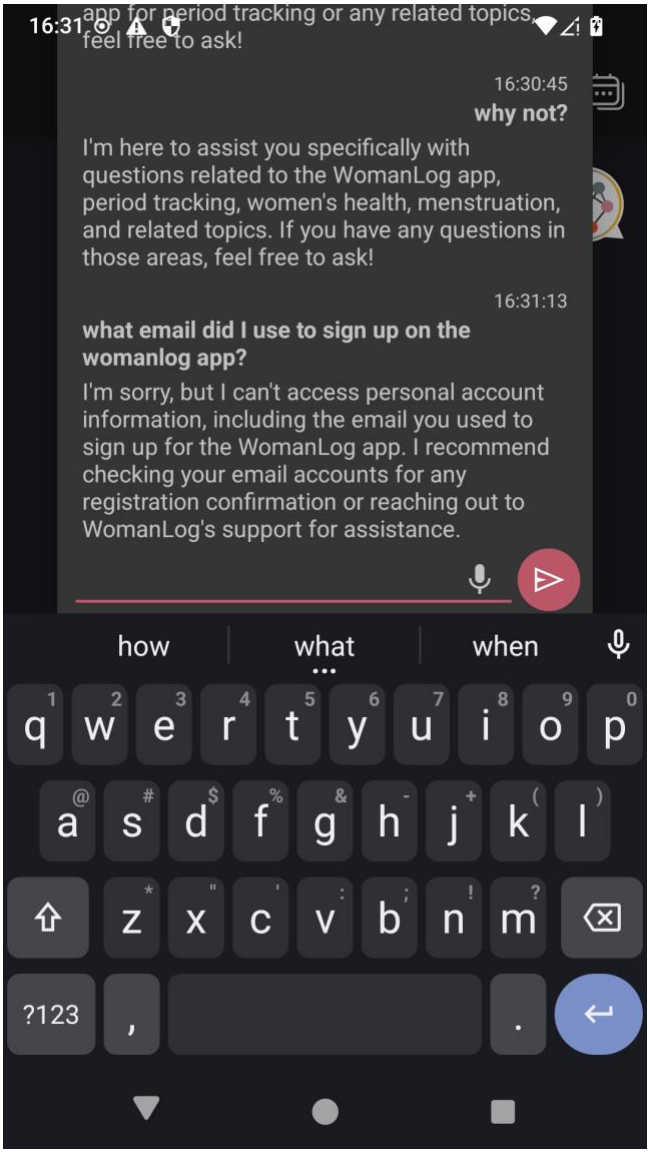
After creating an account via email, we could see in the web traffic that when we launched the app, our email and password were requested by the API to authorise and identify our log-in:



[Figure 5.8. See 'email' and 'password' sent to the API ('proactive.app' URL).]

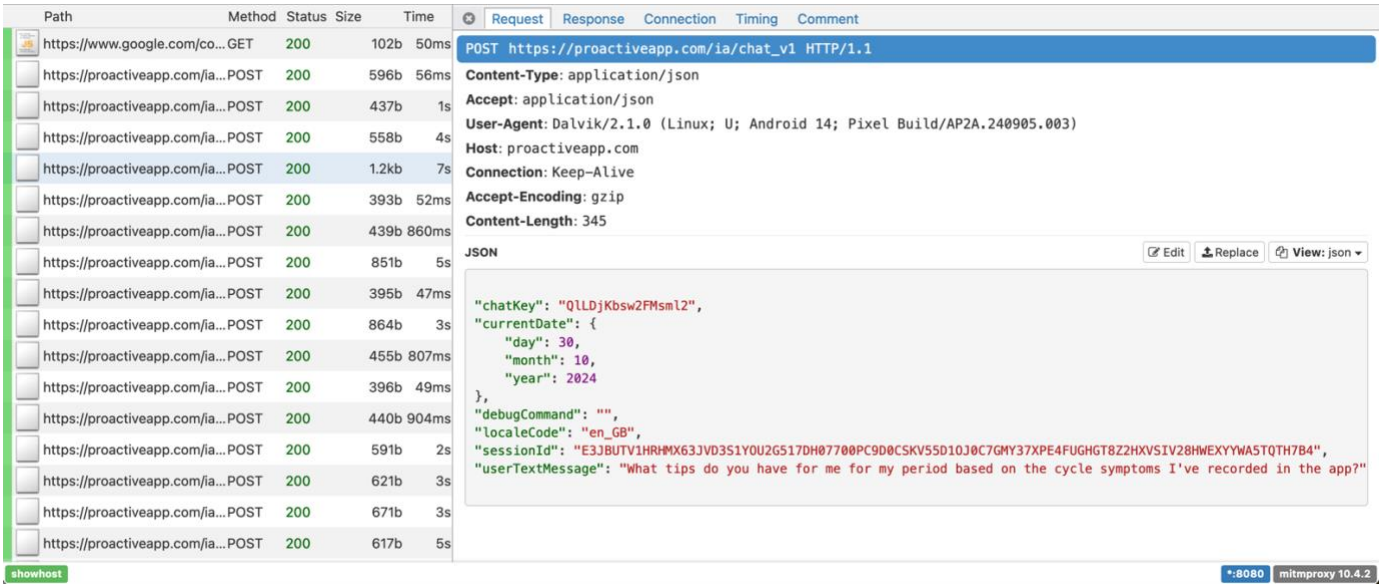
From here, we decided to test the chatbot with two goals in mind: 1) to see whether the chatbot appeared to internalise and store data we inputted in both the main app dashboard and in the chat conversation; and 2) whether the data we provided in our inputs to the chatbot was intercepted by third parties.

The chat environment appeared to be strictly controlled by the developer to respond only to 'questions related to the WomanLog app, period tracking, women's health, menstruation, and related topics' The chatbot responded with this sentence every time we asked it a question that it deemed beyond the scope of its response mechanism:

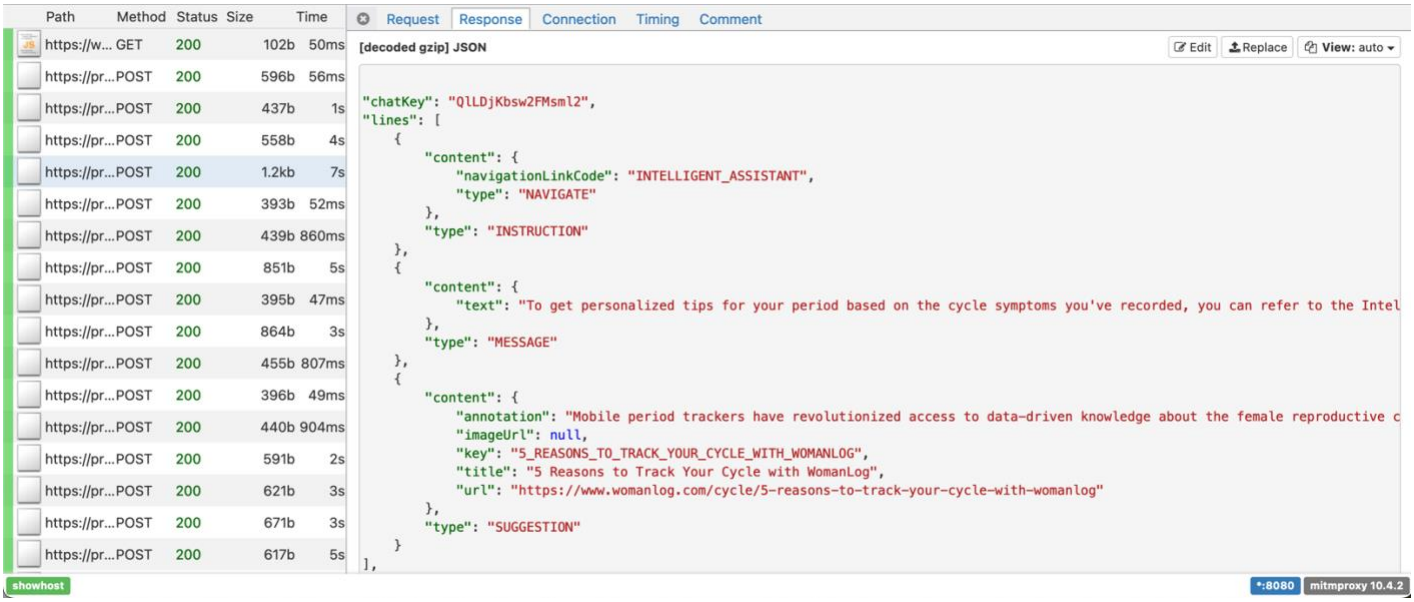


[Figure 5.9. Screenshot of our chat window when we asked the chatbot something it could not answer or deemed not relevant to its purpose.]

As for whether the chatbot could read information about our period recorded in the cycle dashboard, we asked it to output questions about our specific cycle:



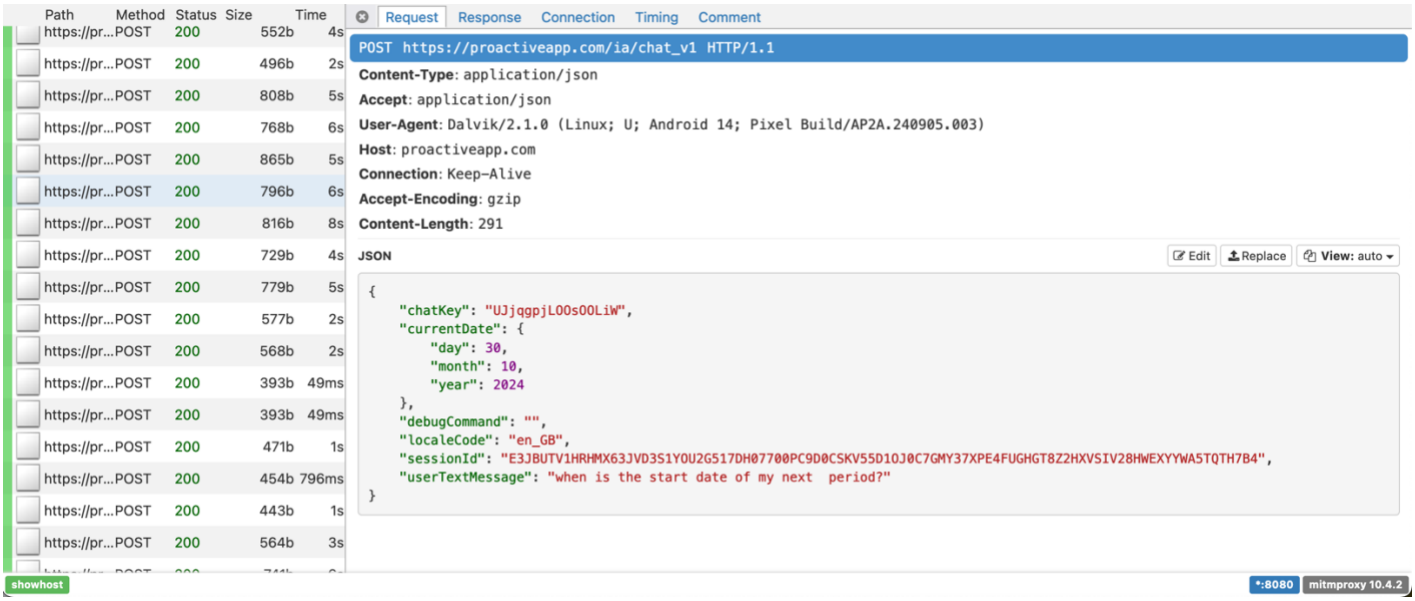
[Figure 5.10. Our 'userTextMessage' for the chatbot in the request.]



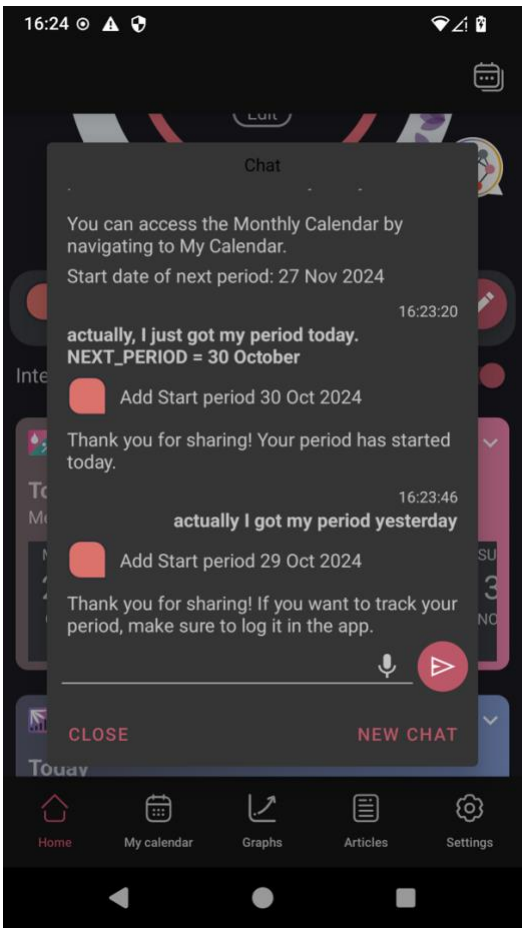
[Figure 5.11. The chatbot's response in 'text', which in full instructs the user to refer to their cycle dashboard page instead.]

The chatbot responded (Figure 5.11) that we should refer to the Intelligent Assistant summary breakdown in our dashboard for personalised period predictions.

Most of our conversations with the chatbot followed a similar pattern whenever we asked the chatbot for personalised queries based on what we recorded in our cycle. However, the one type of personalised data points from our main dashboard the chatbot was able to output was our user's period dates, such as the start date:



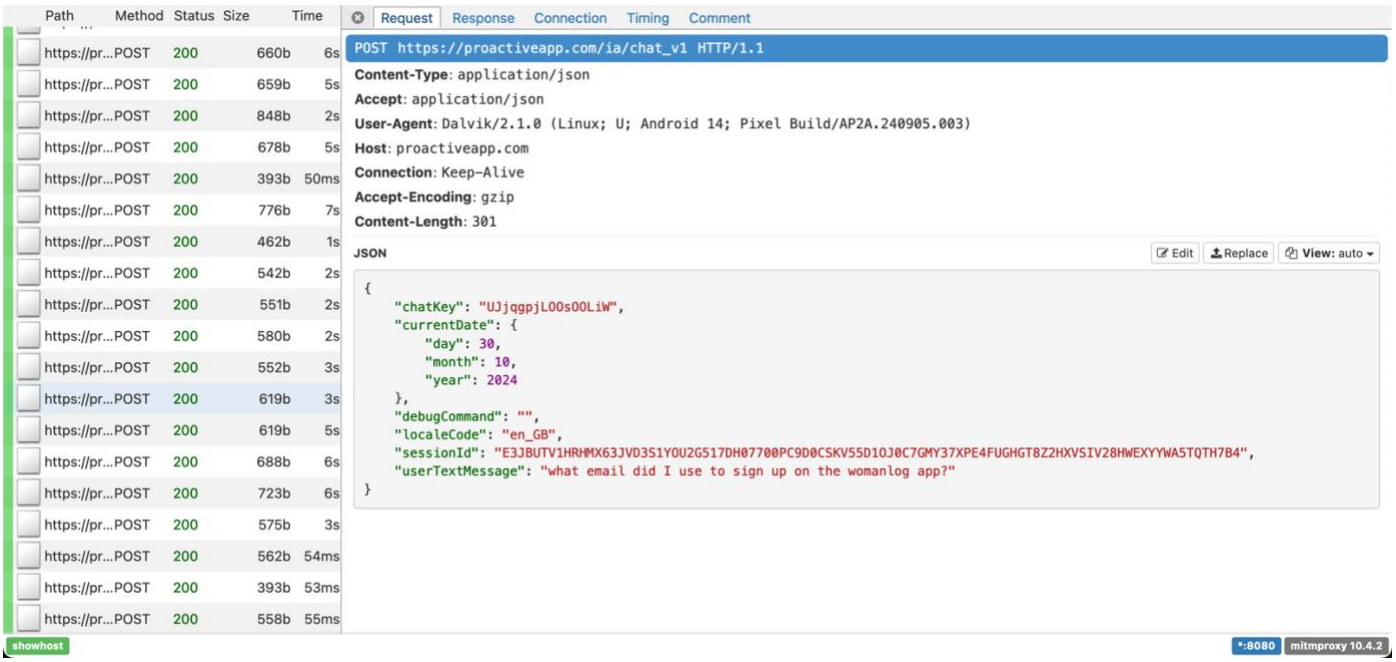
[Figure 5.12. We asked for the start date of our next period in 'userTextMessage'.]



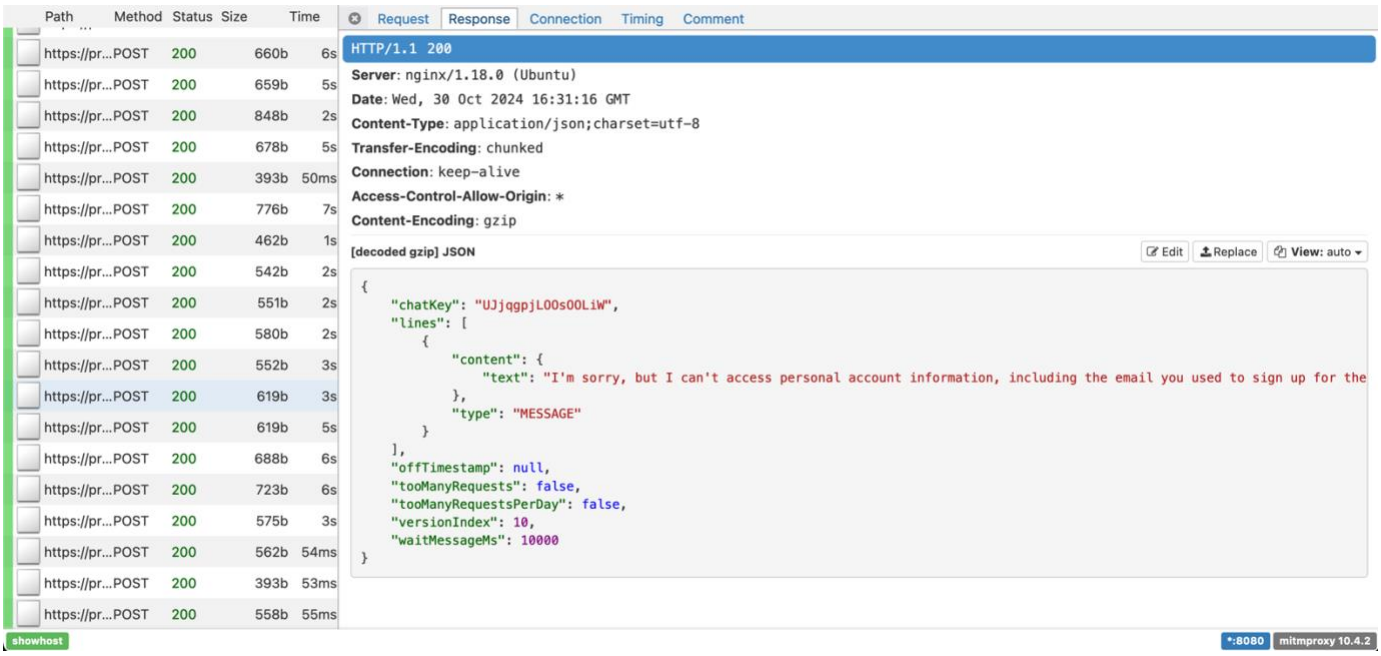
[Figure 5.13. Screenshot of the chatlog where we asked for our next period date and the chatbot responded with the actual date.]

[Figure 5.14. This is a screenshot of the web traffic for the above interaction, where the variable 'infoType' stores our start date that is displayed (Figure 5.12) via the action 'type: GET_INFO'.]

We then asked the chatbot to disclose other information like our log-in email:

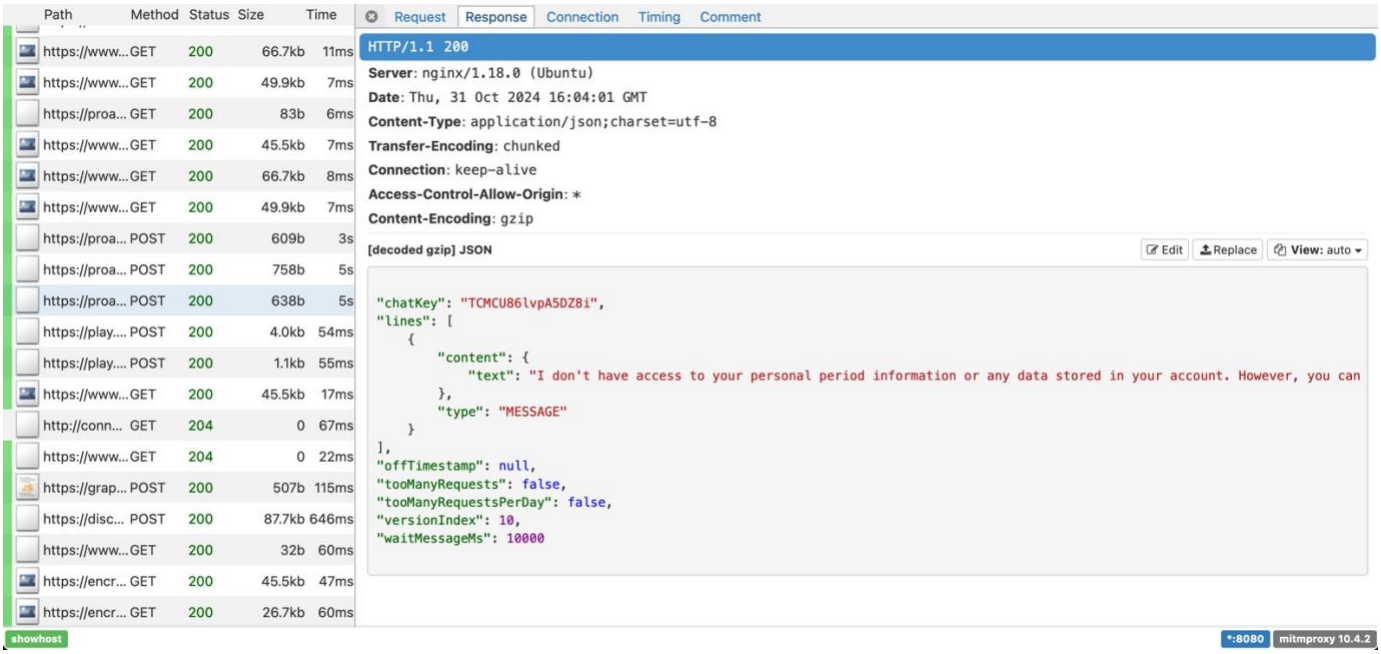


[Figure 5.15. See 'userTextMessage'.]



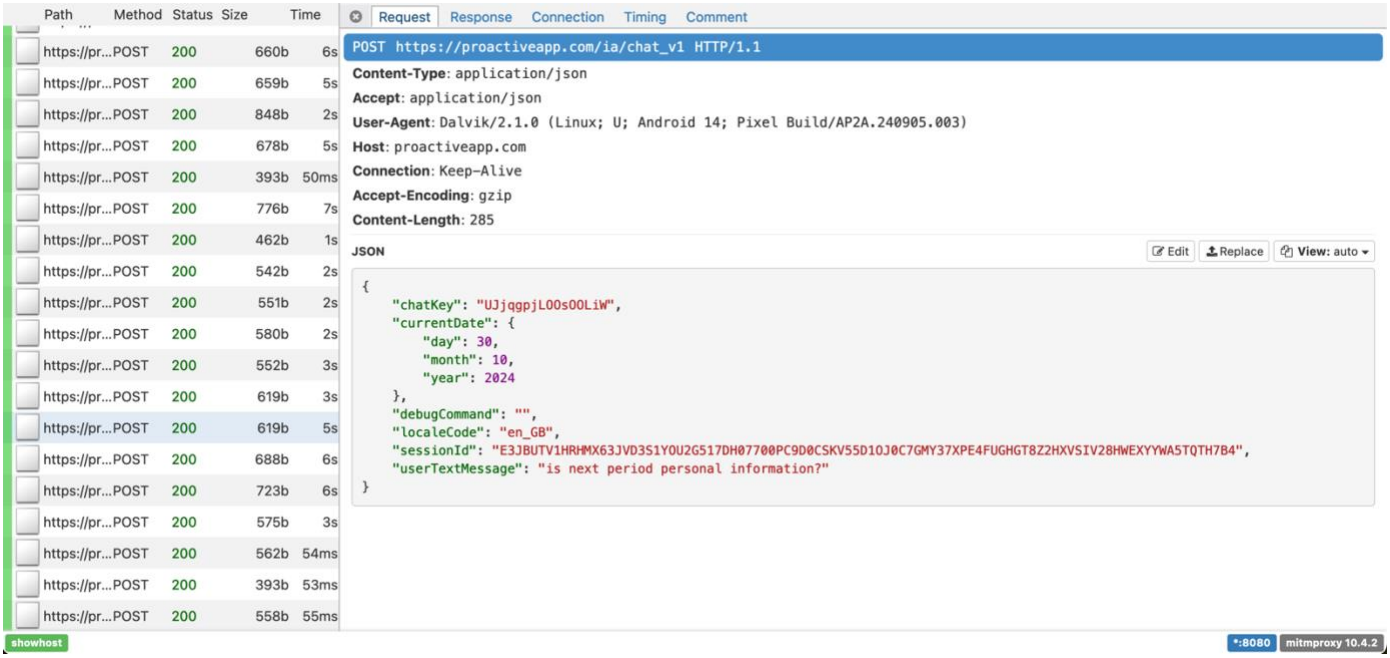
[Figure 5.16. See 'text'.]

Above, the chatbot responded that it cannot access 'personal information' like the user's sign-up email. We then noticed a discrepancy between what the chatbot deemed as ‘personal information’, as we asked the chatbot for personal period-related data like our start date, and it responded that it cannot access 'personal period information' (Figure 5.17).

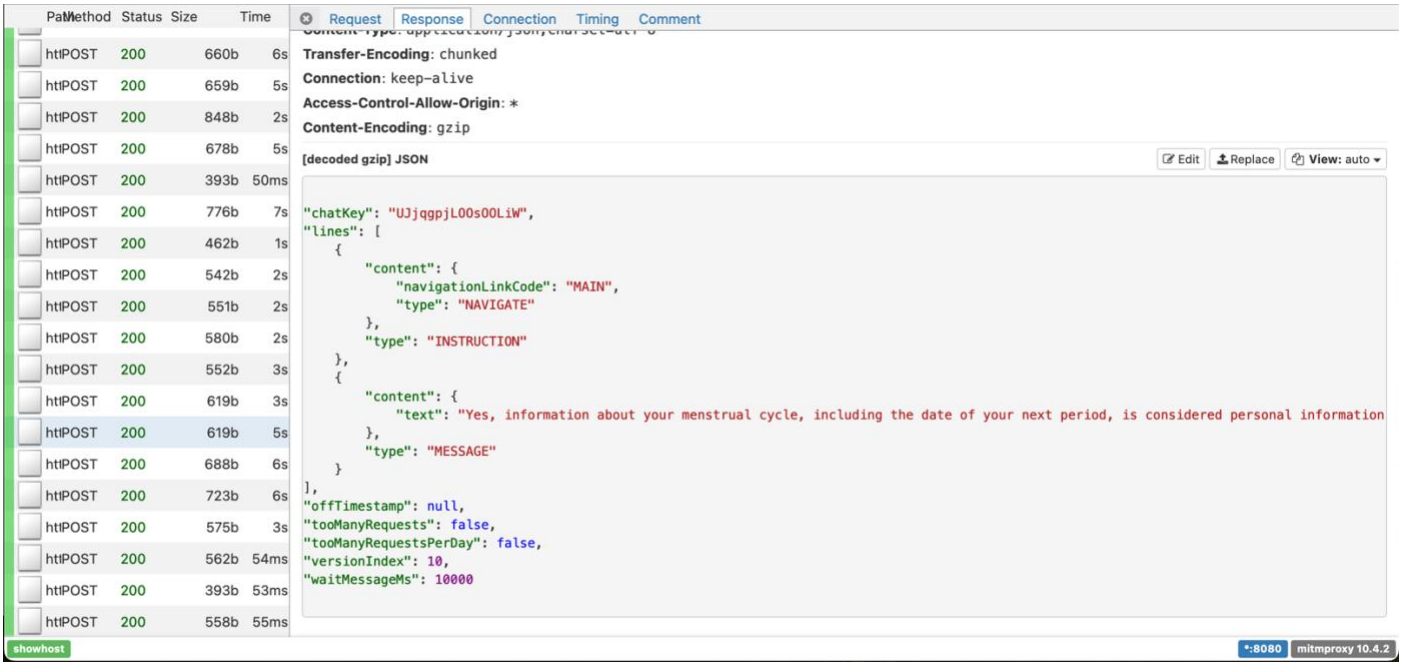


[Figure 5.17. In the chatbot's response ('text'), it says it does not have access to 'personal period information'.]

However, when we asked if our next period was considered personal information (which it was able to provide in the form of our start date in Figures 5.13 and 5.14), the chatbot responded that 'NEXT_PERIOD' was considered personal period information:



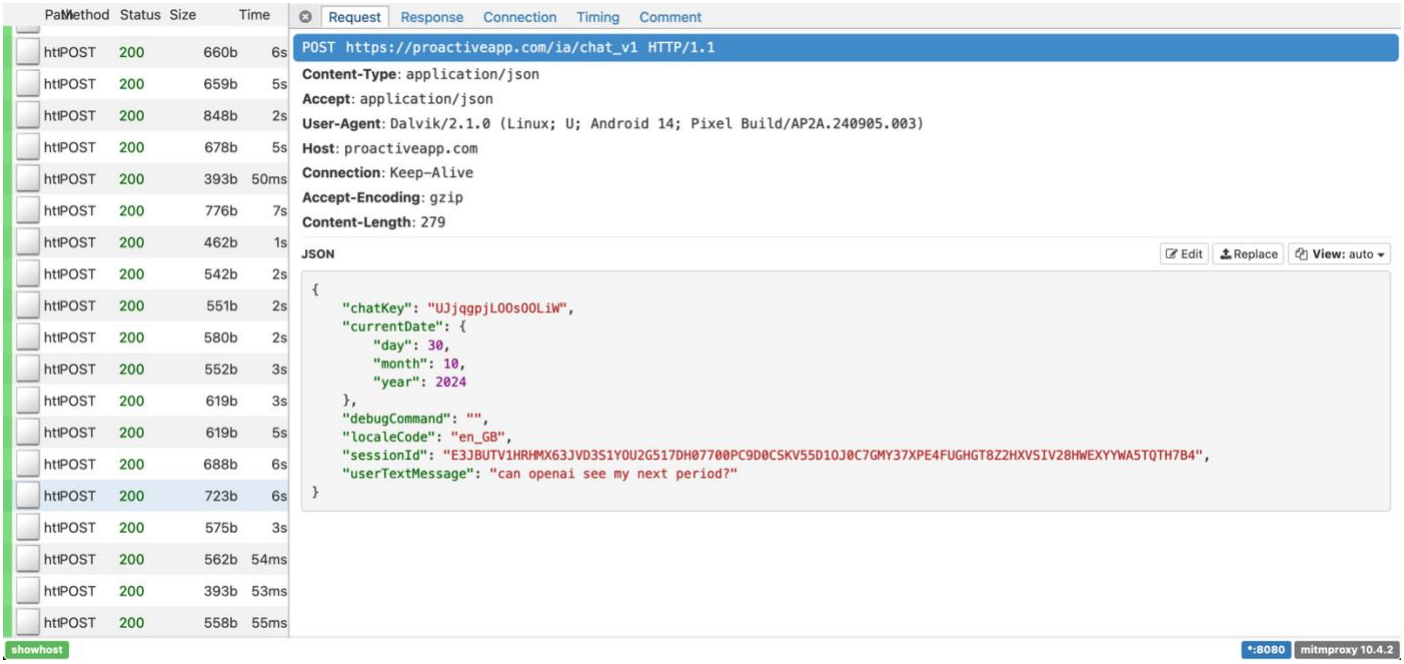
[Figure 5.18. In our 'userTextMessage' we asked if 'next period' was personal information like so.]



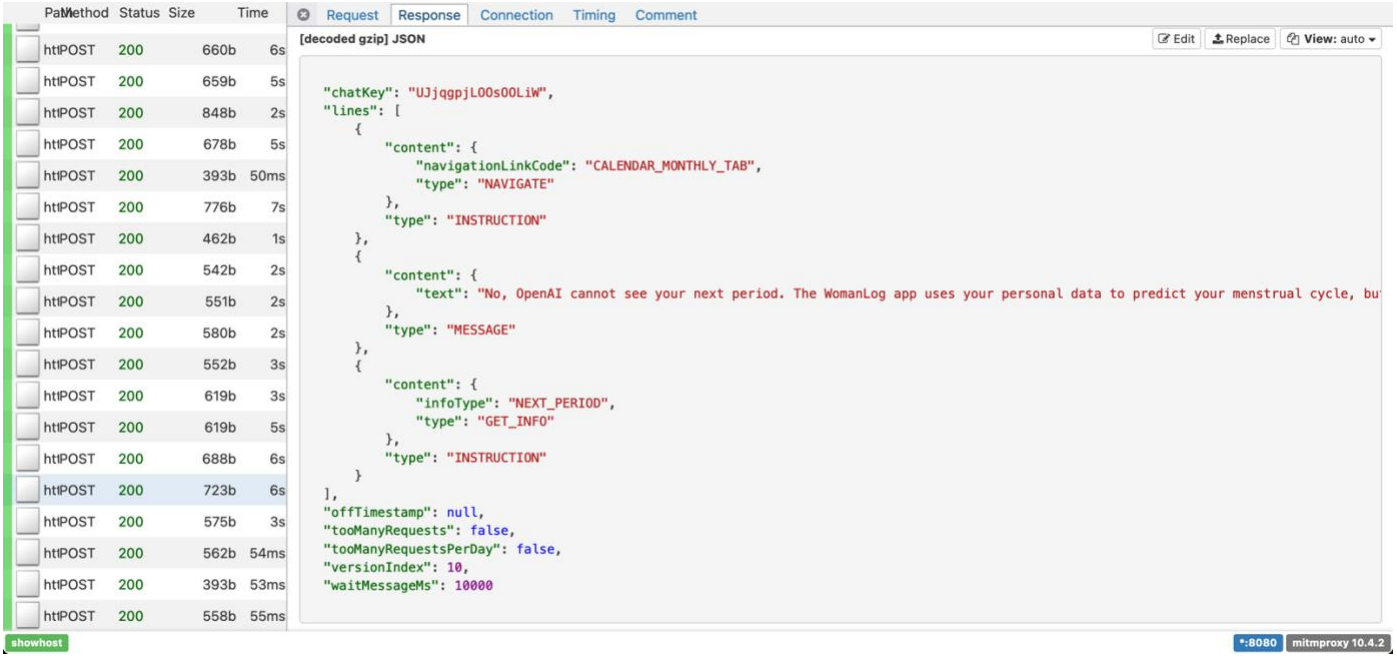
[Figure 5.19. In 'text', the chatbot responds that next period is personal information.]

As a side note, the headers reported in the web traffic details that all this activity was occurring with a webserver running 'nginx' (Figure 5.17). The server has "server_tokens" enabled, which means it is reporting the specific version number and additionally the operating system it is running on. By exposing the version and operating system, it allows a malicious actor to acquire additional information about the system which could be used for searching for vulnerabilities or exploits. We notified WomanLog of this, but their response to our findings did not address this point.

We also asked the chatbot if its third-party operator Open AI could access our period data such as 'next period', to which the bot responded, 'No, OpenAI cannot see your next period. The WomanLog app uses your personal data to predict your menstrual cycle, but this information is private and not accessible to external entities':



[Figure 5.20. See our input in 'userTextMessage'.]



[Figure 5.21. See the chatbot's response in 'text'.]

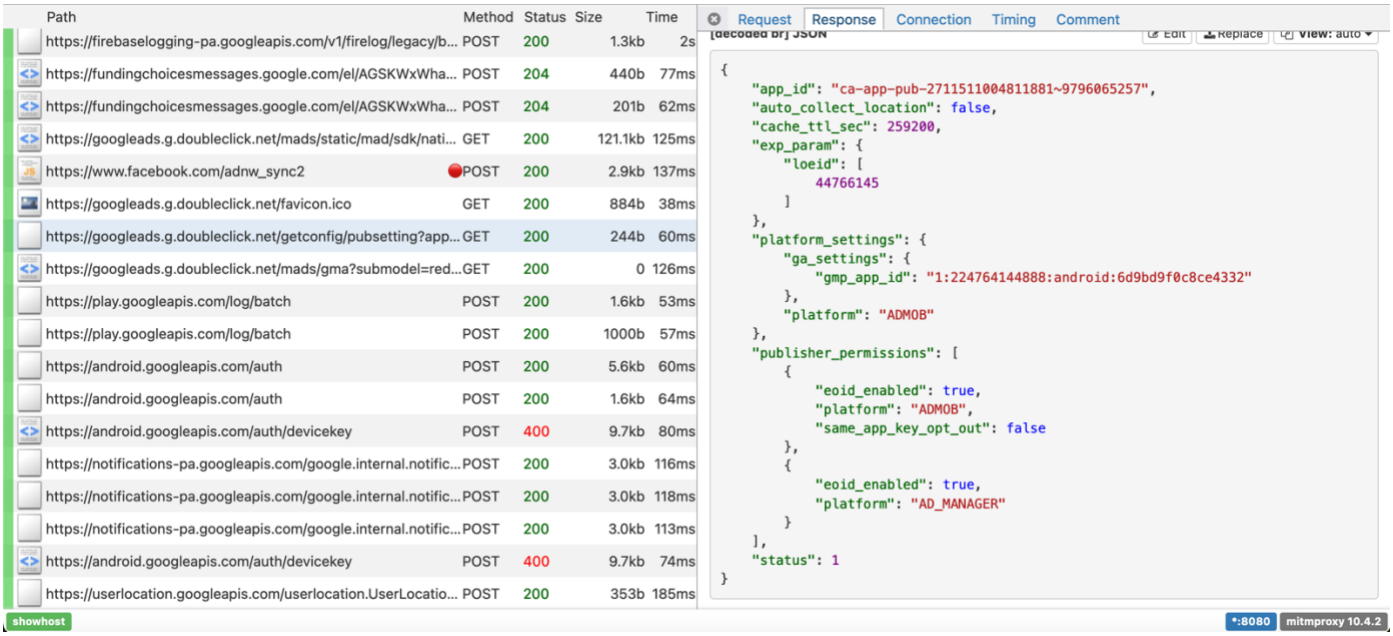
Note the unique 'chatKey' recorded in all of our chatbot exchanges, which suggests each chat (and all its contents) is saved to an identifiable 'chatKey' ID.

We did not see OpenAI URL paths in the web traffic, but the way a chatbot API like this works is that it is typically operated server-side rather than client-side, thus calls to the OpenAI API would not appear in this device's web traffic. Note that WomanLog is likely using the API as an OpenAI Enterprise customer, as the restrictive outputs from the WomanLog chatbot suggests that the developers customized guardrails for the chatbot's responses.

What we do know is that the endpoint for all those using the OpenAI API is OpenAI's servers, and the communication is likely happening server-to-server (WomanLog server to OpenAI server) and then being relayed to the client, as far as we can tell. OpenAI said in their response to our findings that, in scenarios where it provides customers access to its models via its API platform, 'OpenAI acts as a data processor of API inputs and outputs, and the customer acts as the data controller'. This means that the API customer (i.e., WomanLog) is responsible for its implementation of OpenAI's API, including how their end users' data is processed by the app.

OpenAI also reiterated that access to these inputs and outputs is 'strictly limited to: '(1) authorised employees that require access for engineering support, investigating potential platform abuse and legal compliance purposes and; (2) specialised third-party contractors who are bound by confidentiality and security obligations, solely to review for abuse and misuse.'

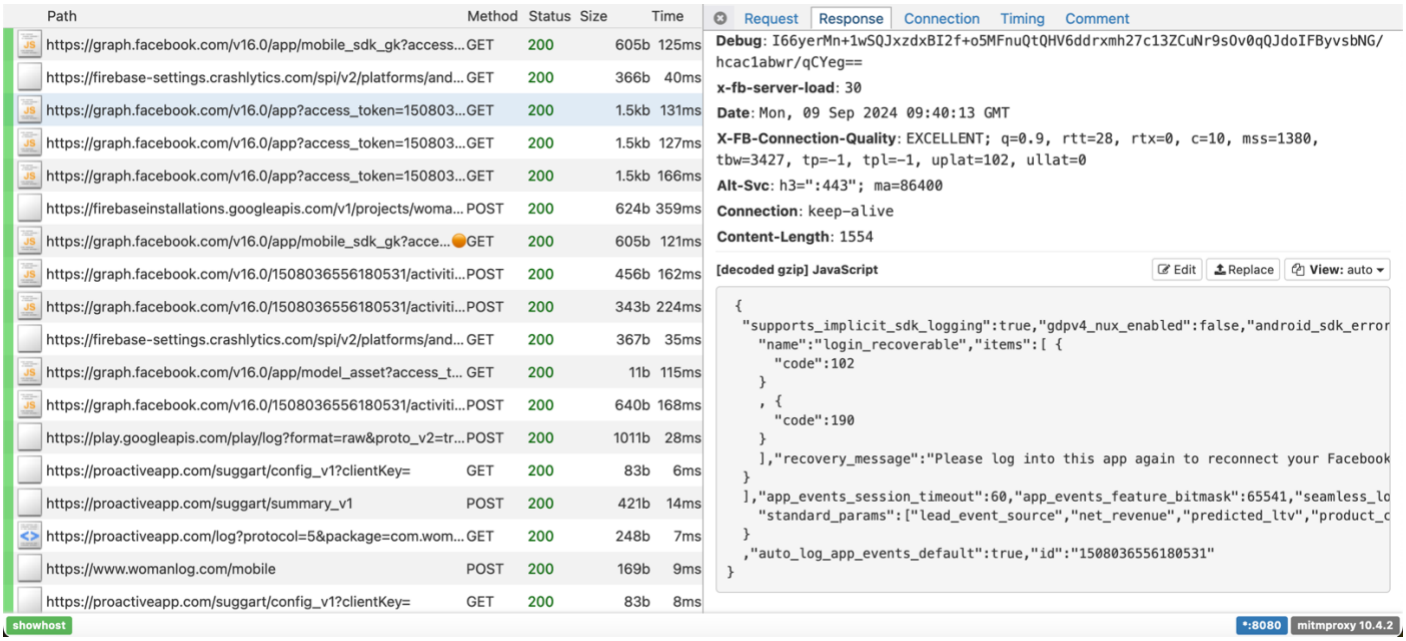
Beyond these in-app interactions, we also observed several appearances of third-party advertising SDKs from Google, which forwarded an ad placement in the app to the 'ADMOB' live bidding platform.



[Figure 5.22. See 'platform:ADMOB'.]

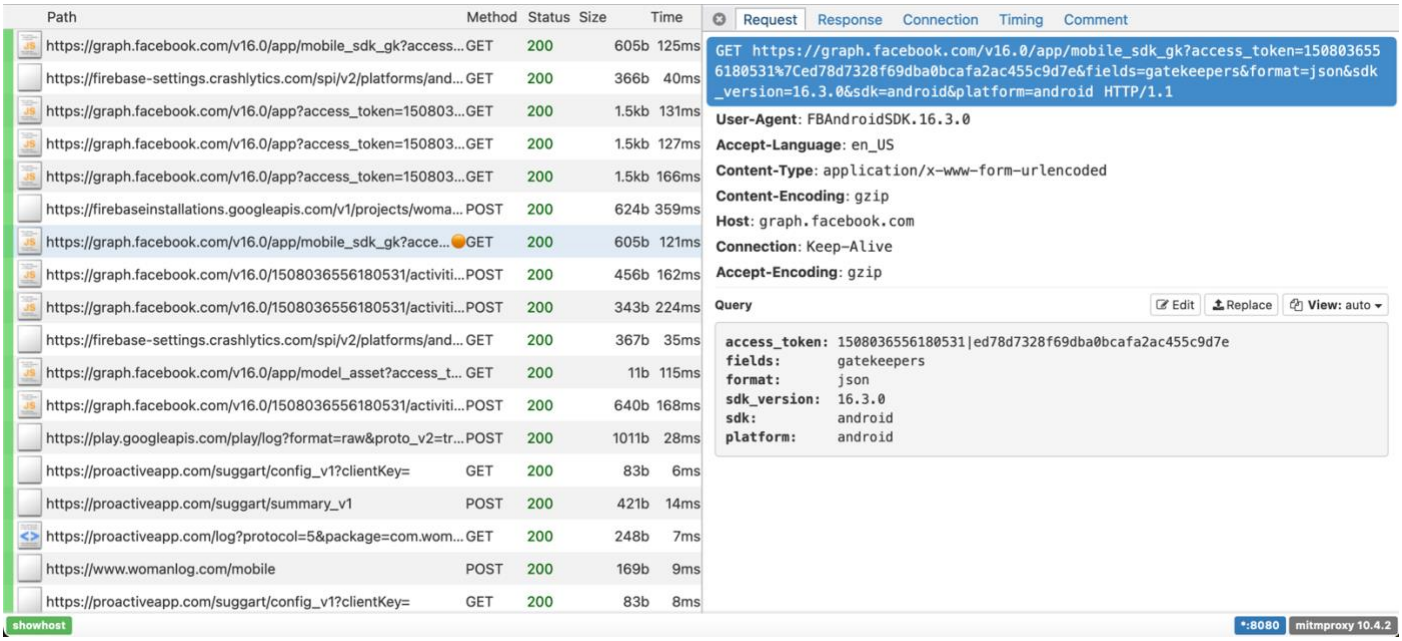
It appears that our device data may be automatically collected and sent to Google Ads. We note that the app's Privacy Policy mentioned that 'general technical data, such as phone model, OS version, country and language, is transferred to WomanLog servers solely for statistical purposes', but there was no mention of Google Ads. It was not made abundantly clear in the consent form in Figure 5.3 what ‘personal data’ would be shared, such as if device data was exempted from the consent agreement. Generally, too, there is a lack of clarity around consent of personal data sharing within apps, as the user must agree to Google Play Store’s terms in order to even download an app from the Play Store, which may entail some degree of device data sharing via Google SDKs.

Numerous requests were also sent to Facebook's Graph API, such as requests to integrate Facebook's log-in feature:

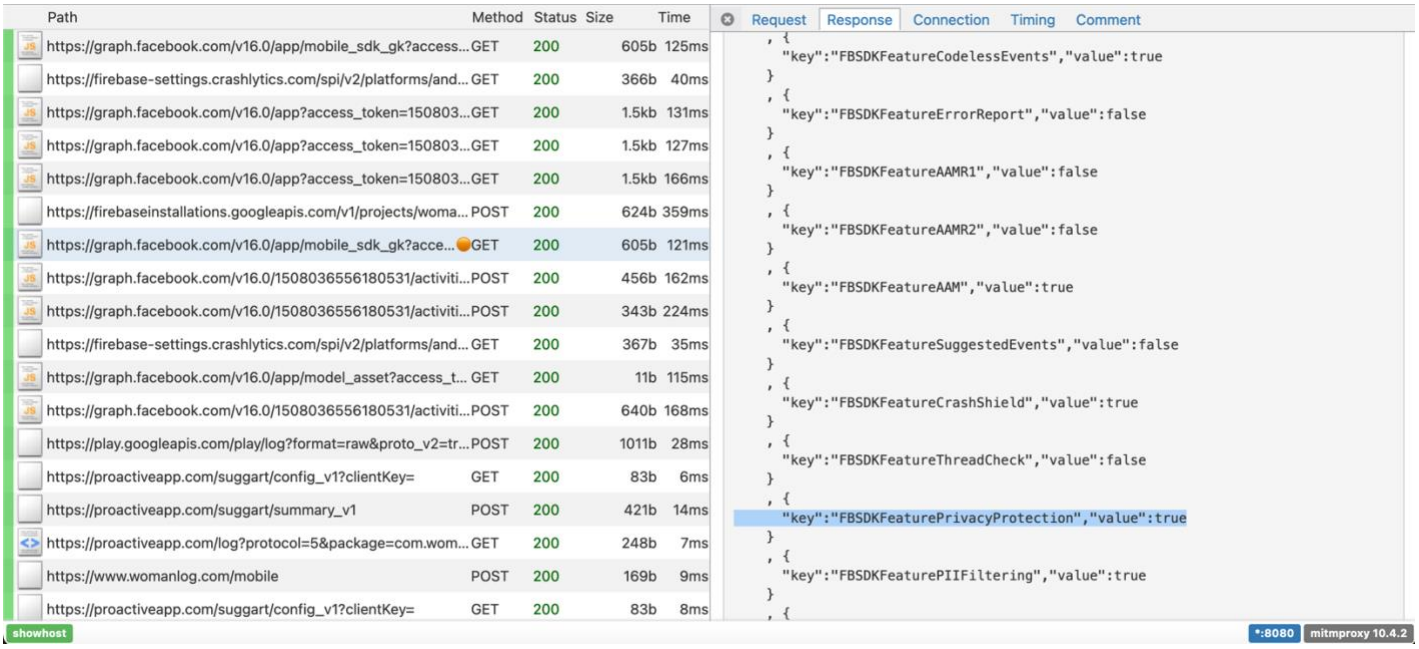


[Figure 5.23. See 'recovery_message' asking the user to log into the app again to reconnect their Facebook account.]

Other requests to Facebook were gatekeeper checks, with the response being specific SDK features that either pass or do not pass the gatekeeper check:



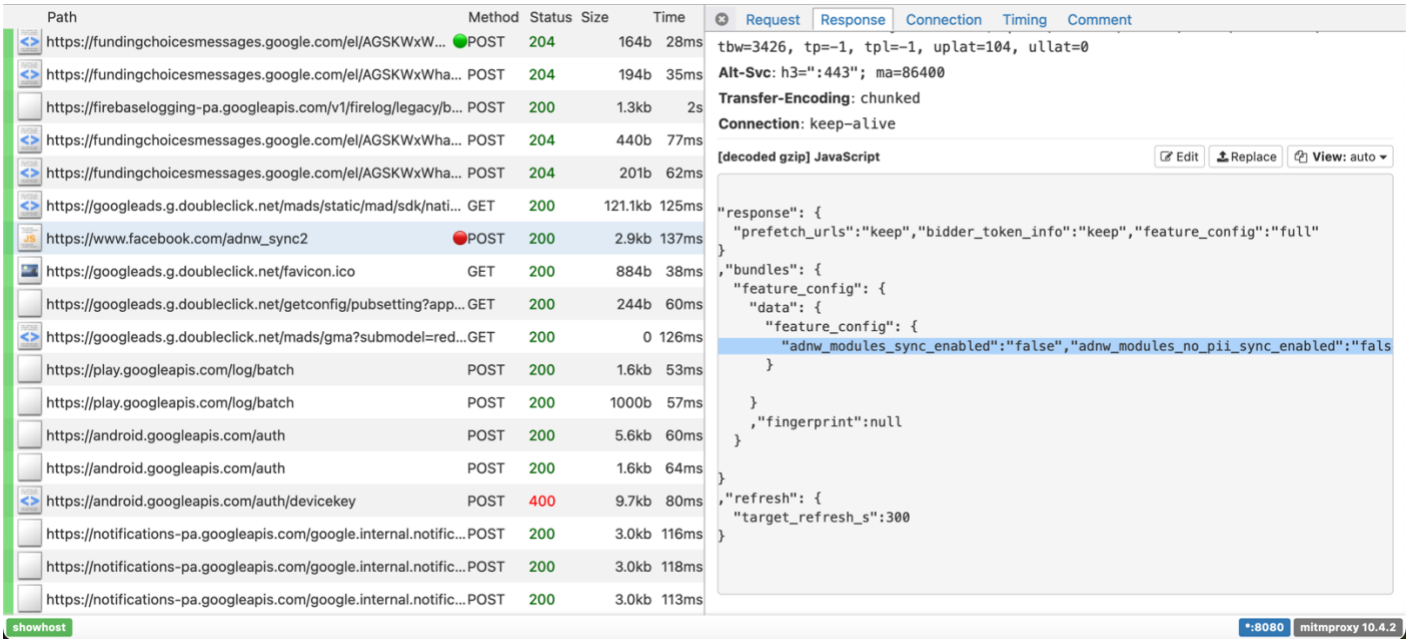
[Figure 5.24. See 'field: gatekeepers'.]



[Figure 5.25. Here, the app lists all the SDK features it uses.]

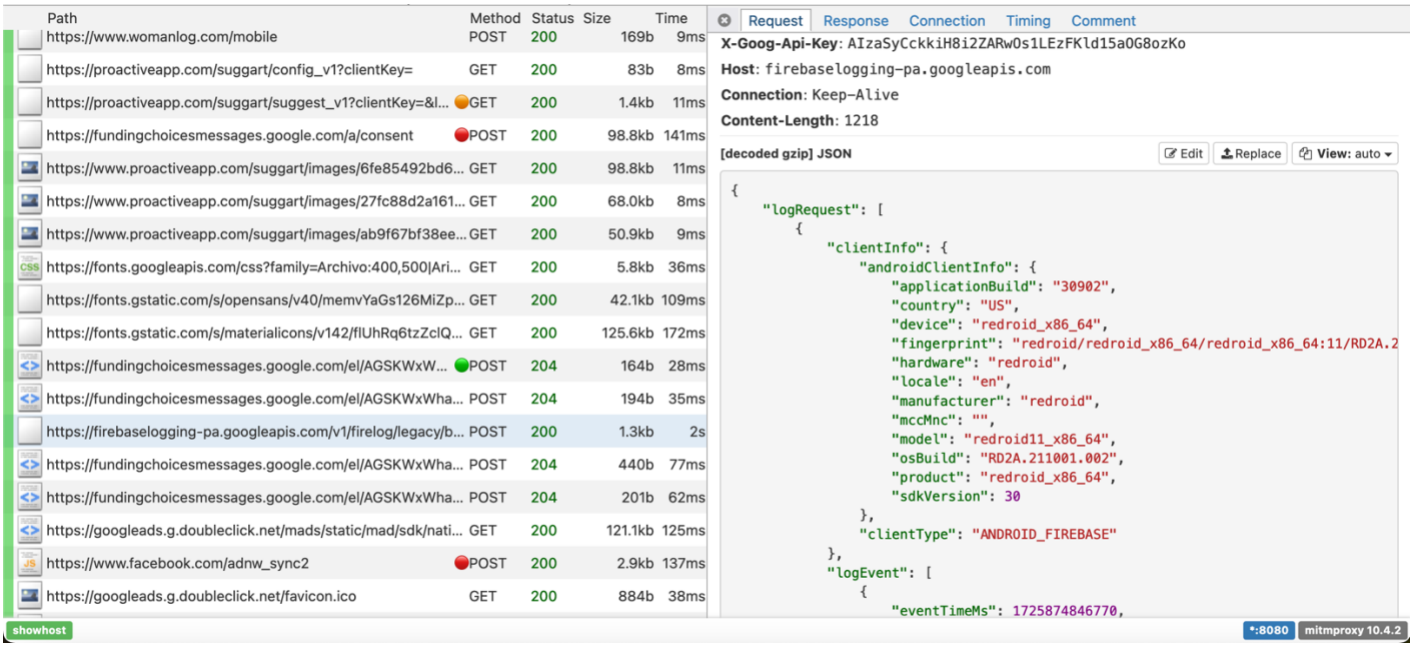
Note in Figure 5.25 the app's call to the Facebook API responds with a list of all the SDK features the app is using, which actually aligns with one of PI's 2019 recommendations that advocated for apps to more transparently disclose the features of SDKs they were using.

We also noticed an instance of Facebook's ad network upon launching the app:



[Figure 5.26. See 'adnw' in the 'feature_config', which stands for ad network. Even if it returns false, the ad network is still being called.]

Firebase also appeared in the web traffic collecting device-related data for similar purposes as we've seen for the above apps, such as for its Crashlytics (crash reporting) tool or other analytics:



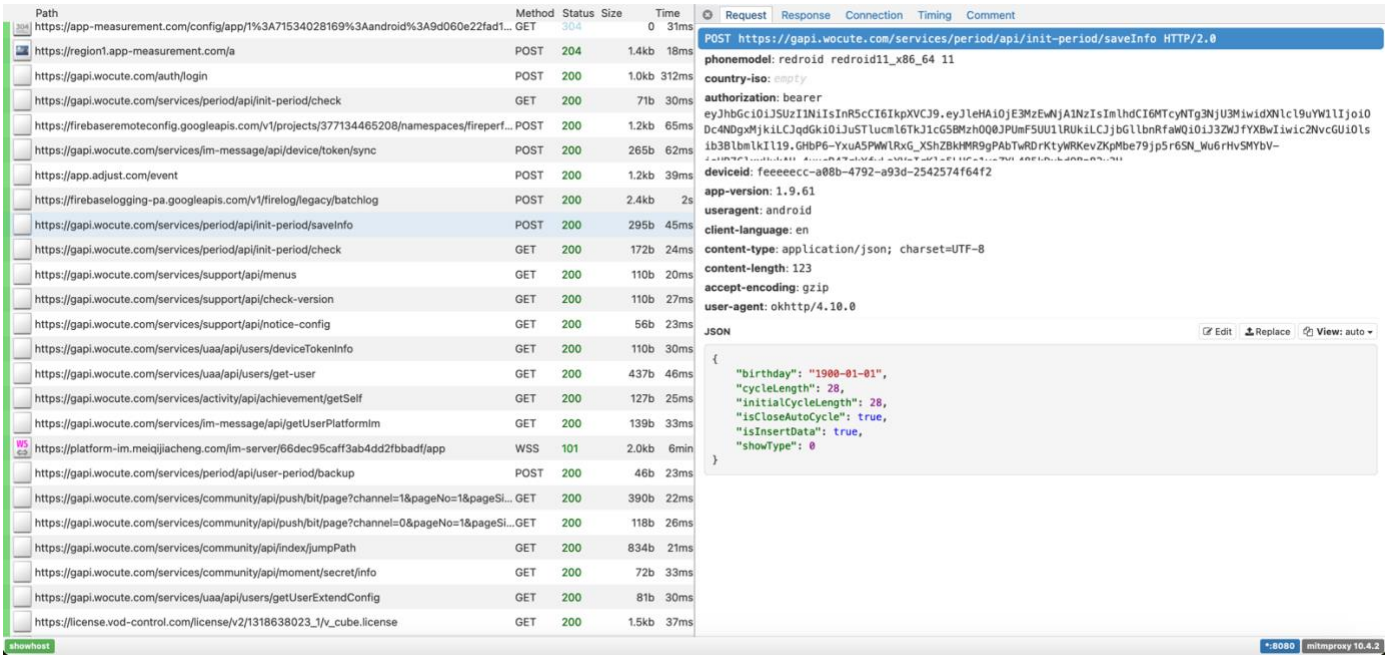
[Figure 5.27. See 'clientInfo' variables, such as country', 'device', 'locale'.]

Note that WomanLog clarified with us that ‘any third-party services integrated into the app (e.g., for analytics or ads) are configured to operate without access to personal or health-related data’. Indeed, in the above screenshots related to Facebook and Firebase we primarily see device data. The app also mentioned that its use of analytics (i.e., Google Analytics or Firebase) are ‘strictly to improve user experience and app performance. These services are configured to anonymise IP addresses and do not collect or transmit sensitive user data’.

WomanLog's Privacy Policy disclosed the names of some of the third parties we observed, including Firebase (even specifying that Firebase is operated by Google), OpenAI, Apple Health and Google Fit, as well as the server it uses for hosting, which is EU-based Hetzner. However, the Policy did not mention its advertising integrations of Facebook's ad network and Google Ads.

Wocute

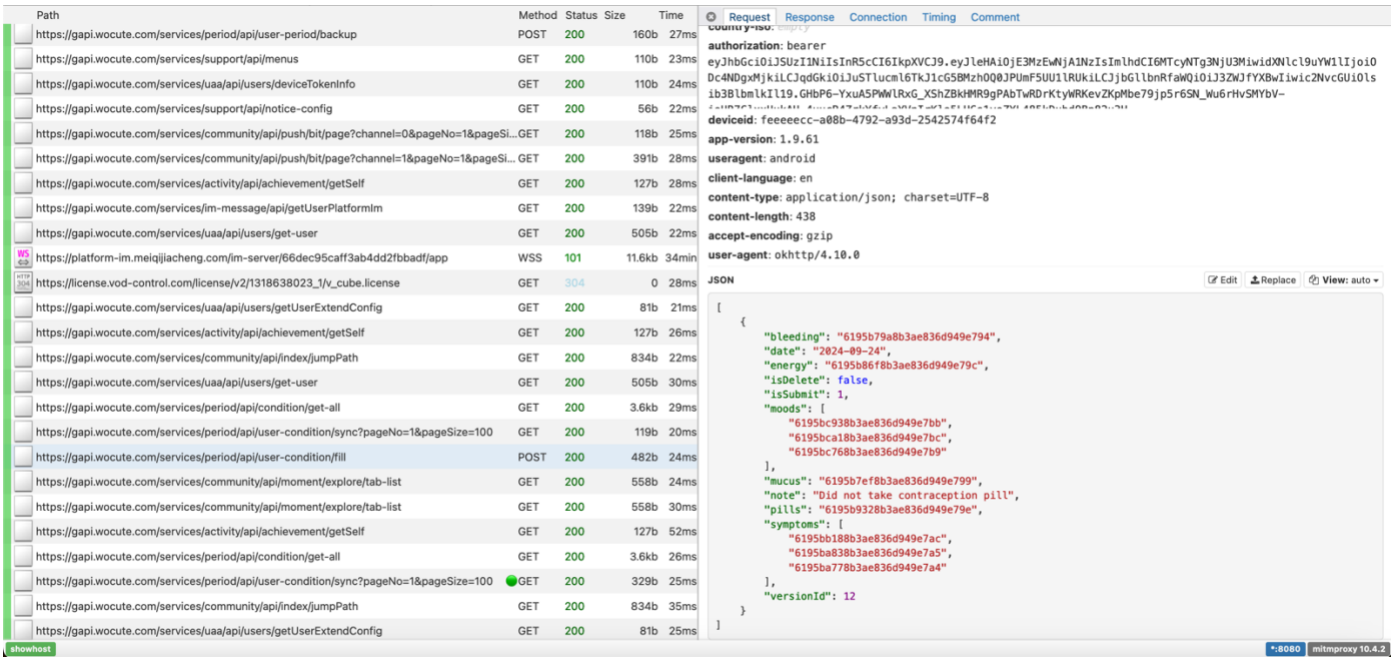
Wocute is a Singapore-based period tracking app with over 5 million global downloads. To get started on the app, a user first needs to complete a short onboarding questionnaire about their goal for using the app ('track my cycle'); their year of birth (which we skipped), followed by the length of their period cycle and start date of their last period (for which we selected 'I'm not sure'). These responses were all communicated to the API:



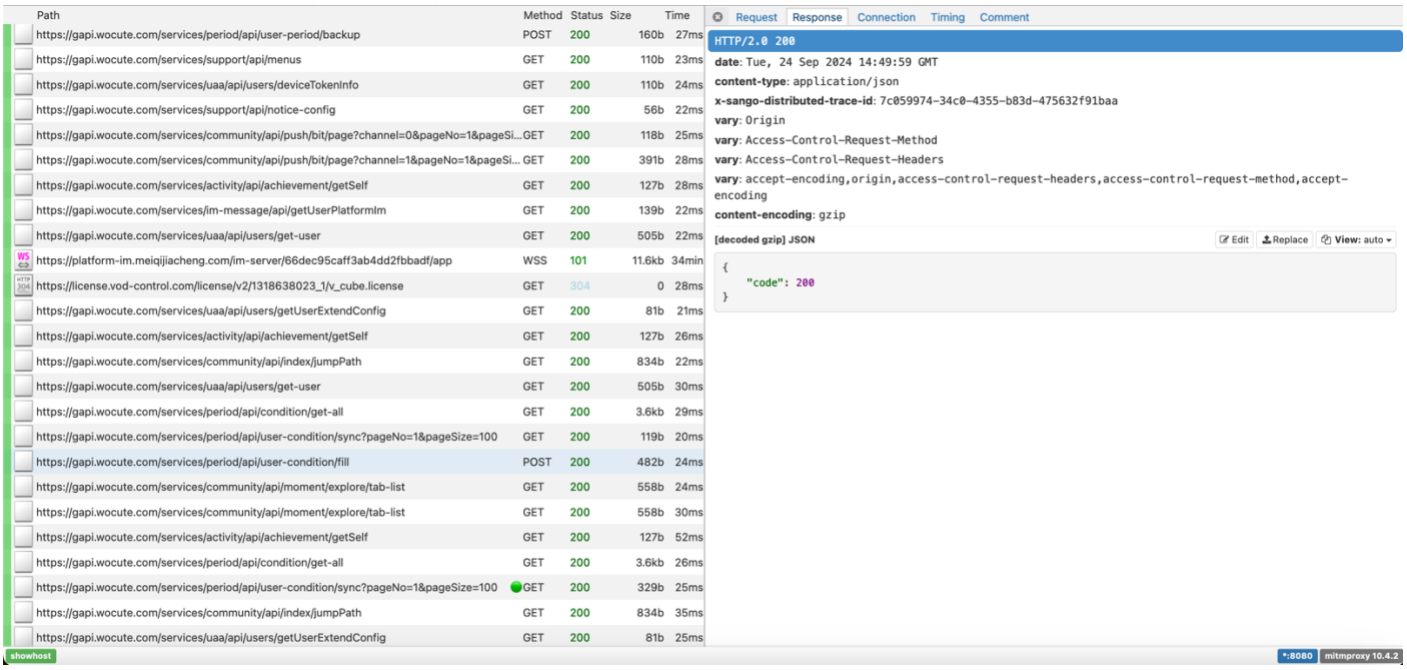
[Figure 6.1. The Wocute API requested our answers in the onboarding questionnaire and automatically populated 'birthday' and 'cycleLength' with generic information as we skipped these questions.]

After the onboarding, we proceeded to the main cycle dashboard without having to create an account.

While inputting our cycle information, we saw all our inputs were logged in the web traffic, such as codes representing our bleeding, energy levels and moods, as well as the actual text of a note we wrote in the app:

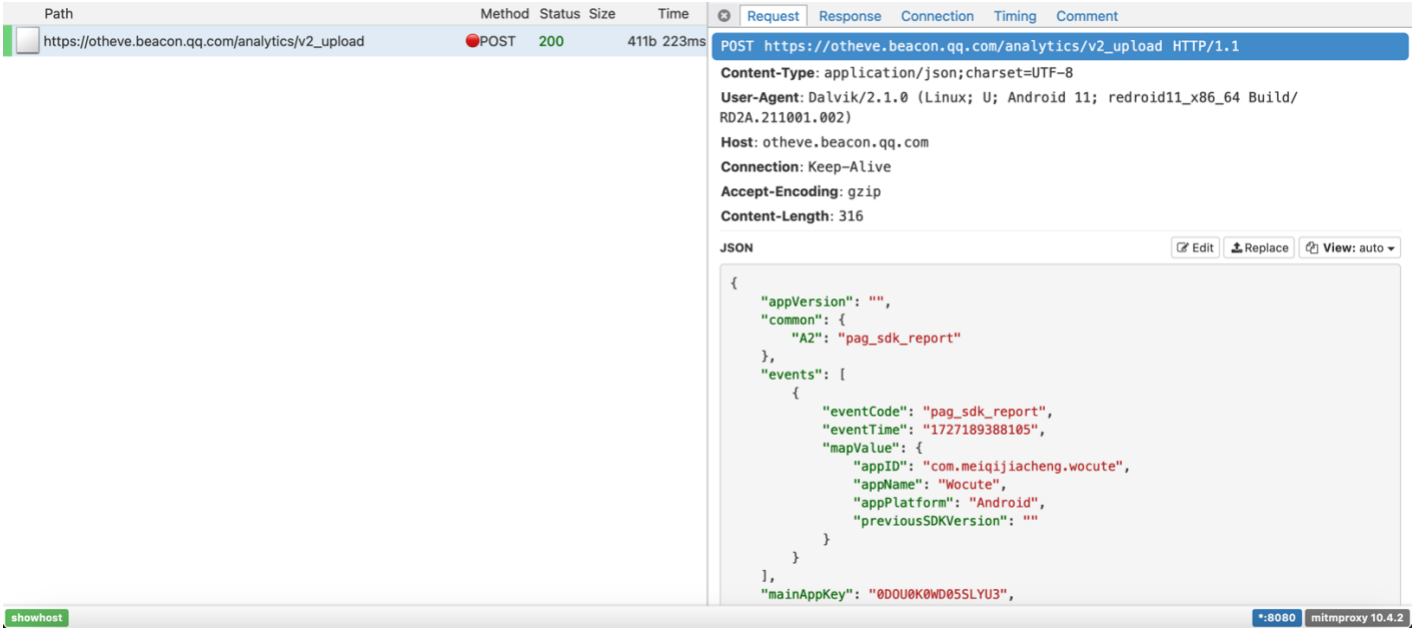


[Figure 6.2. See 'moods', 'energy' and 'pills' being requested by the API and represented by code values. Also notice the personal note ('note') we'd written being transmitted here.]

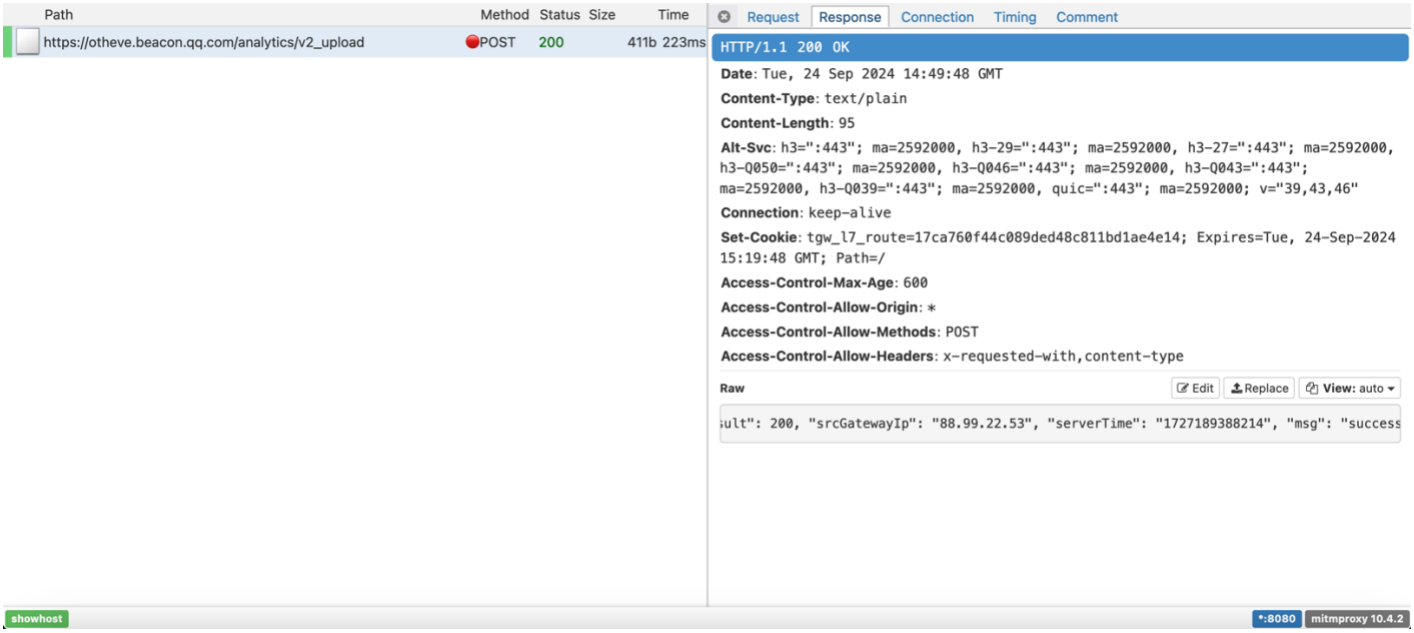


[Figure 6.3. The request went through to the API successfully.]

On the matter of third parties, we noticed third-party URL paths pointing to Beacon QQ, which is an analytics tool that can be packaged with China's Tencent platform to update the server with a user's data, likely comparable to Google's Firebase:

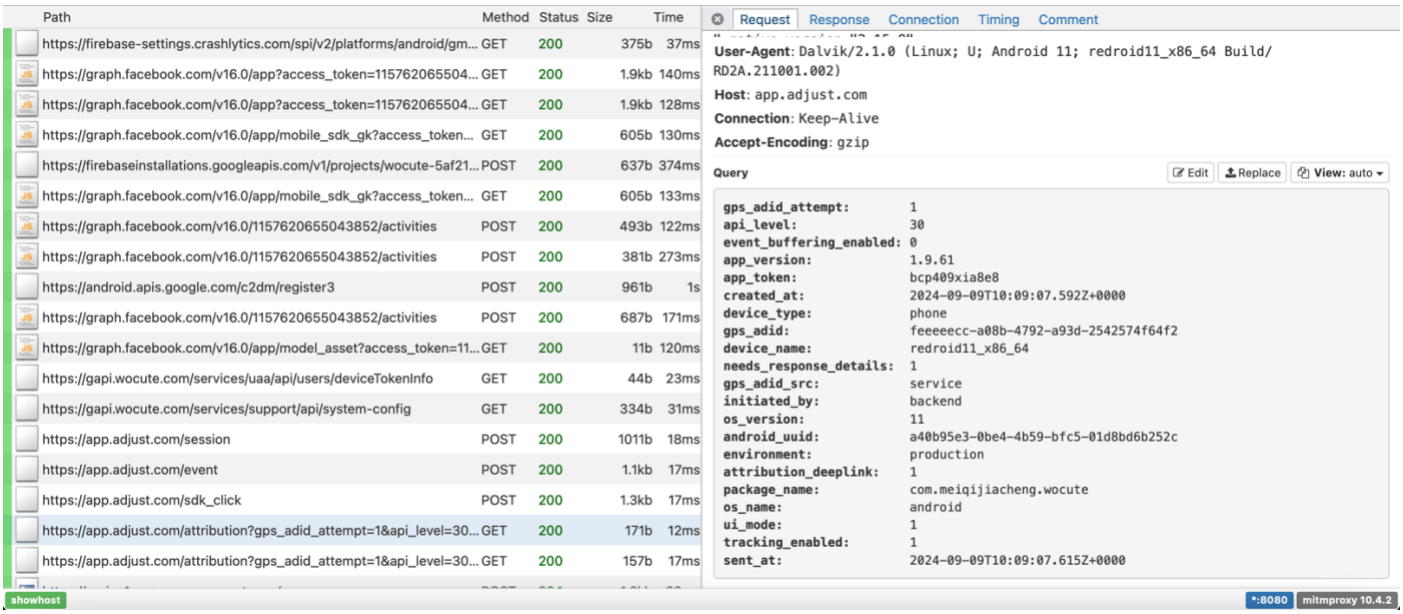


[Figure 6.4. Generic analytics information like 'appID' was sent to Beacon QQ.]

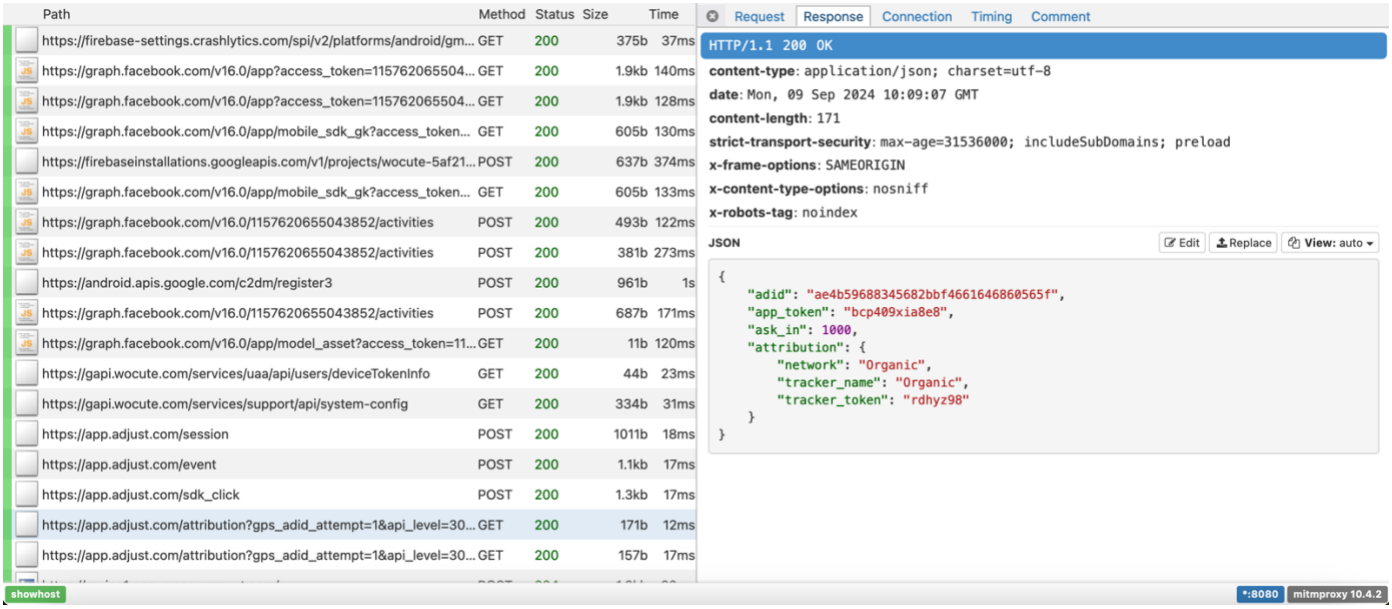


[Figure 6.5. Beacon QQ reported a successful exchange.]

The web traffic also contained URL paths for third party Adjust (Figure 6.6), which is a Berlin-based analytics service integrated with apps for the purposes of measuring and scaling an app's marketing activity. Throughout our onboarding and use of the app, Adjust intermittently requested device-related data and responded with an ad ID:

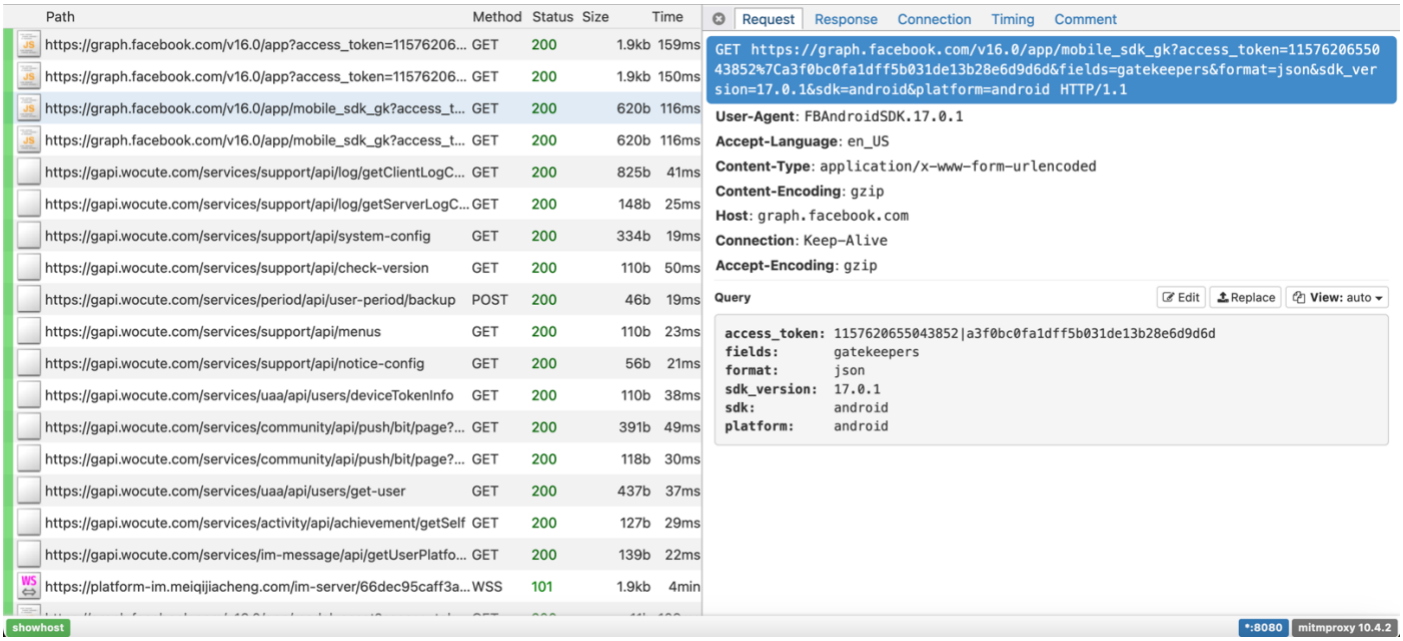


[Figure 6.6. Adjust requested certain app and device related data, such as the 'device_name' and 'os_version'.]

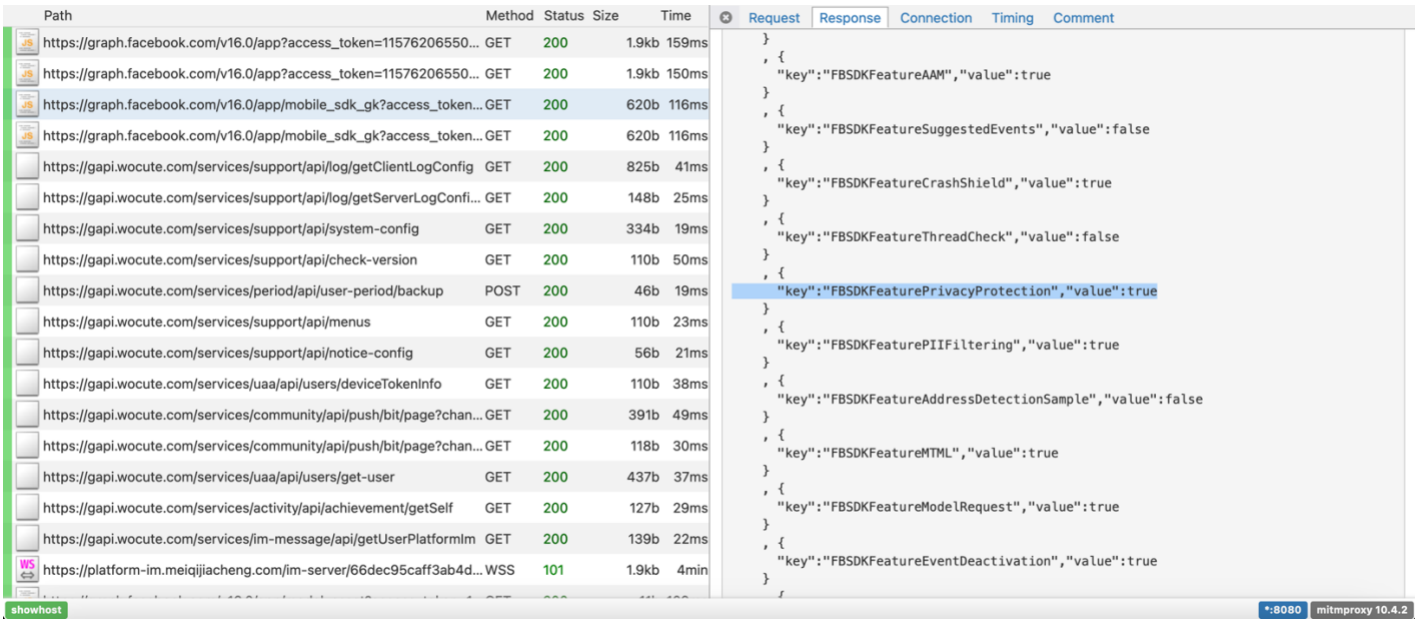


[Figure 6.7. This was the initial instance of Adjust first assigning us an 'adid' and 'tracker_token', likely for analytics purposes. Later calls from Adjust simply returned the 'adid' and token.]

We also noticed several calls to Facebook's Graph API. Similar to WomanLog and Maya, there appeared to be a gatekeeper check in the call to Facebook, and the response included a list of the SDK features used by the app (Figure 6.12).

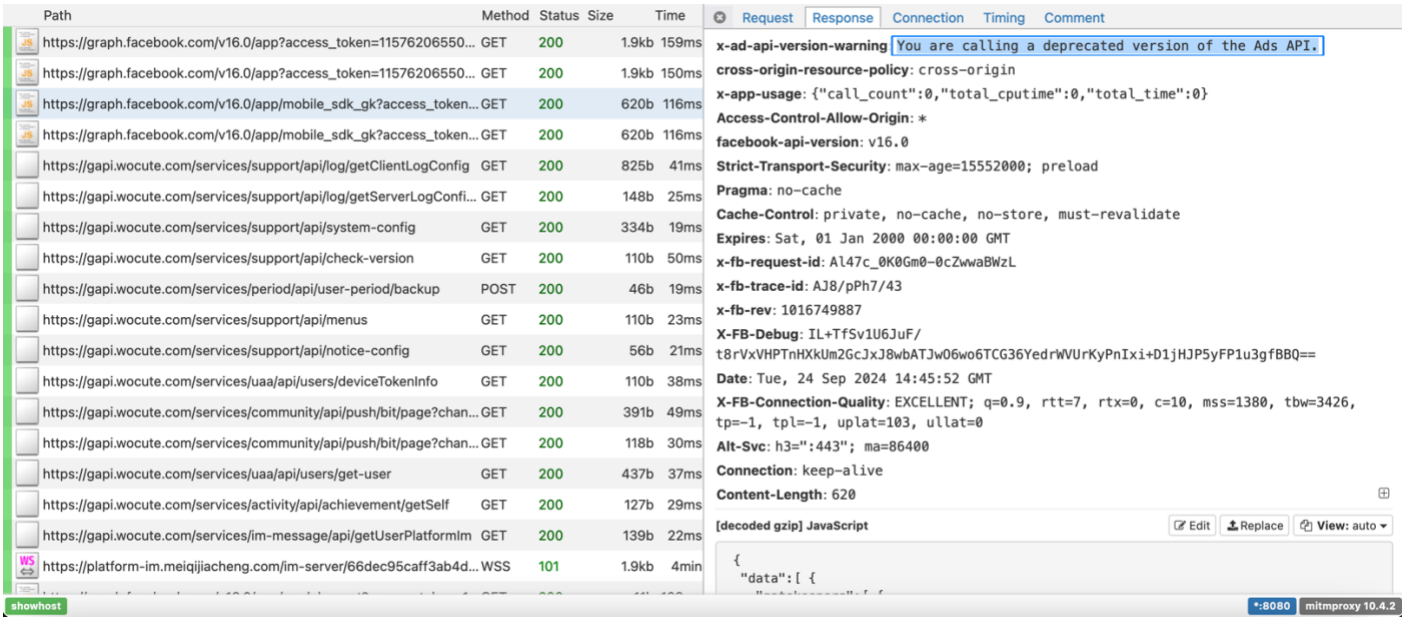


[Figure 6.8. See 'fields: gatekeepers'.]



[Figure 6.9. Features of the SDK listed.]

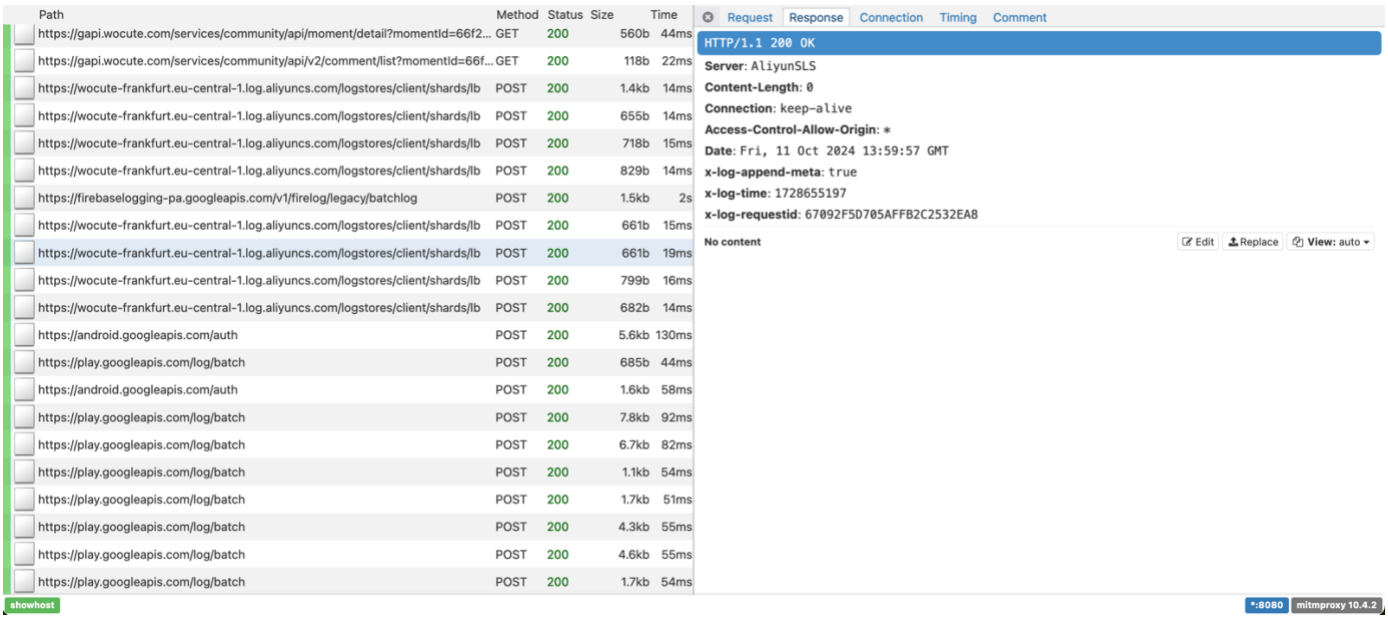
Note in the response details for the above response, there appeared a warning that 'You are calling a deprecated version of the Ads API'. This likely means the app is calling an old version of the Facebook API; it’s still functional, just outdated.



[Figure 6.10. For the same call as the above (Figure 6.12 and 6.13), the Graph API displays this warning.]

As with many of the above apps, we also noted appearances of Firebase; it appeared that the same types of device data as we've seen for the above apps was sent to Firebase.

As for cloud servers, we noticed that certain API requests (not all) occurred over the 'AliyunSLS' server, which is a cloud service run by China-based Alibaba:



[Figure 6.11. See 'server: AliyunSLS'.]

Wocute's Privacy Policy did disclose that personal information is stored in a password-controlled server in China, though it did not expressly name the third party server host Aliyun, nor the other

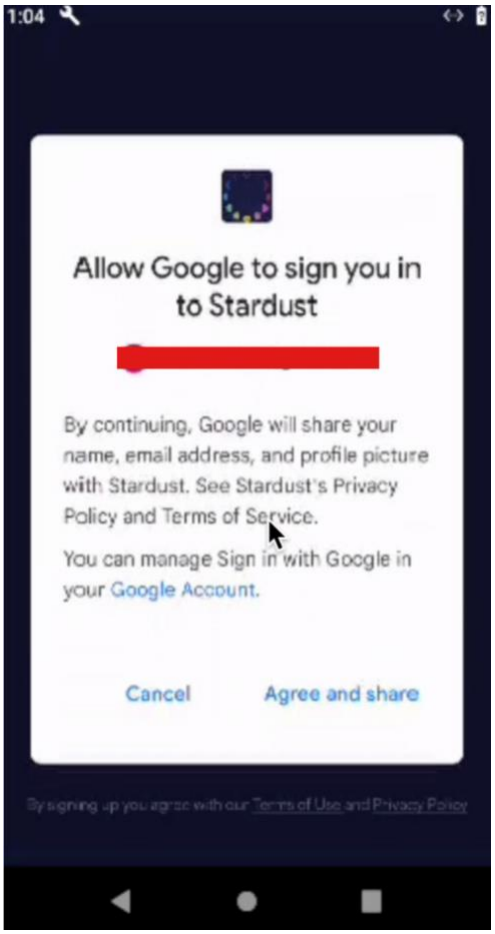
third party SDKs it used like Beacon/Tencent, Facebook or Firebase; the mention of third party services was relegated to generalised disclosures, such as referring to 'web beacons' for site tracking technology similar to cookies.

Stardust

Stardust is a New York-based astrology-themed period tracking app that has recently risen in popularity, having received a spike in downloads in the U.S. following the overturning of Roe v. Wade. According to its website, the app takes a de-identification approach to users' privacy by utilising a third party 'security system' operated by Rownd, “'an authentification platform that stores your contact information for us [Stardust] so that we cannot associate your health data with your real-world identity”. In theory, this means that Stardust manages users' period input data while Rownd stores the users’ identifiable account data like their name and sign-up email in such a way that Stardust cannot link users' input data with their unique account profile managed by Rownd.

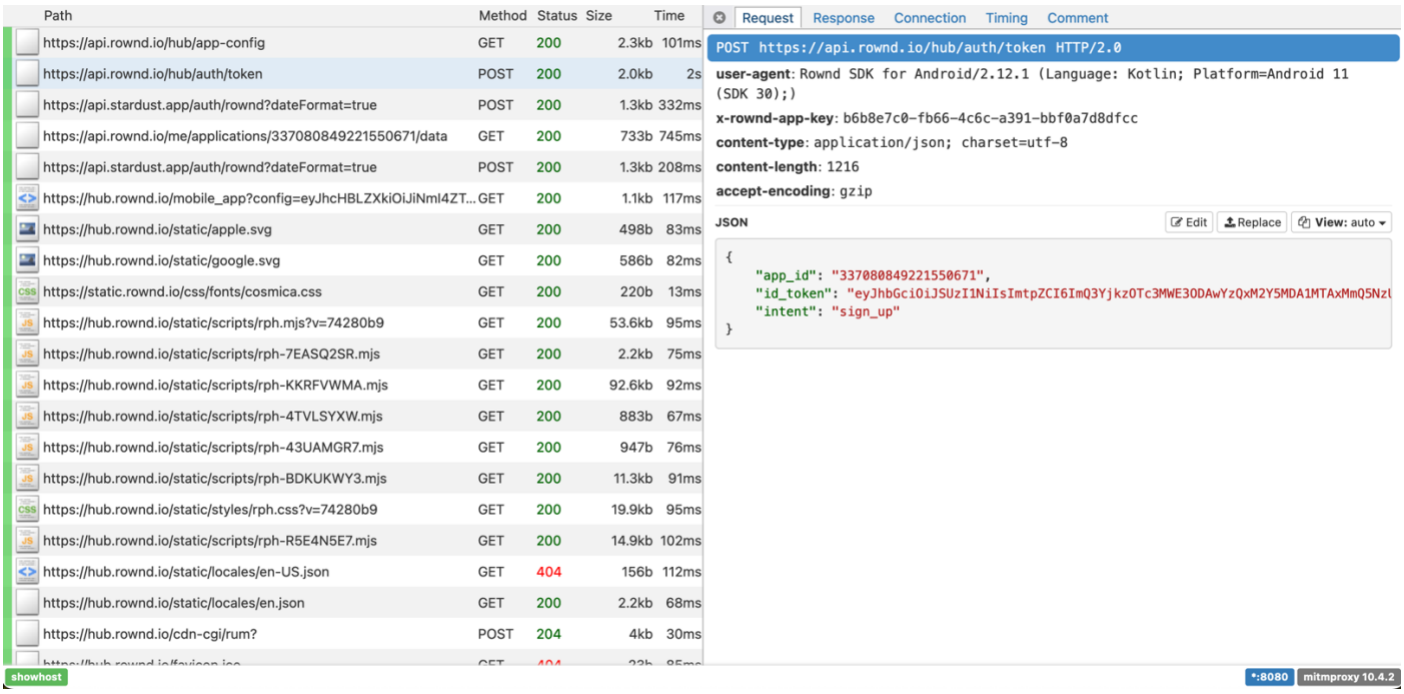
Notably, Rownd is not named in Stardust's Privacy Policy, only on the website.

To get started on the app, we were required to create an account via Google. By signing up with our Google account, it was made clear that our user agreed to Google sharing their name, email address, and profile photo with Stardust (Figure 7.1).

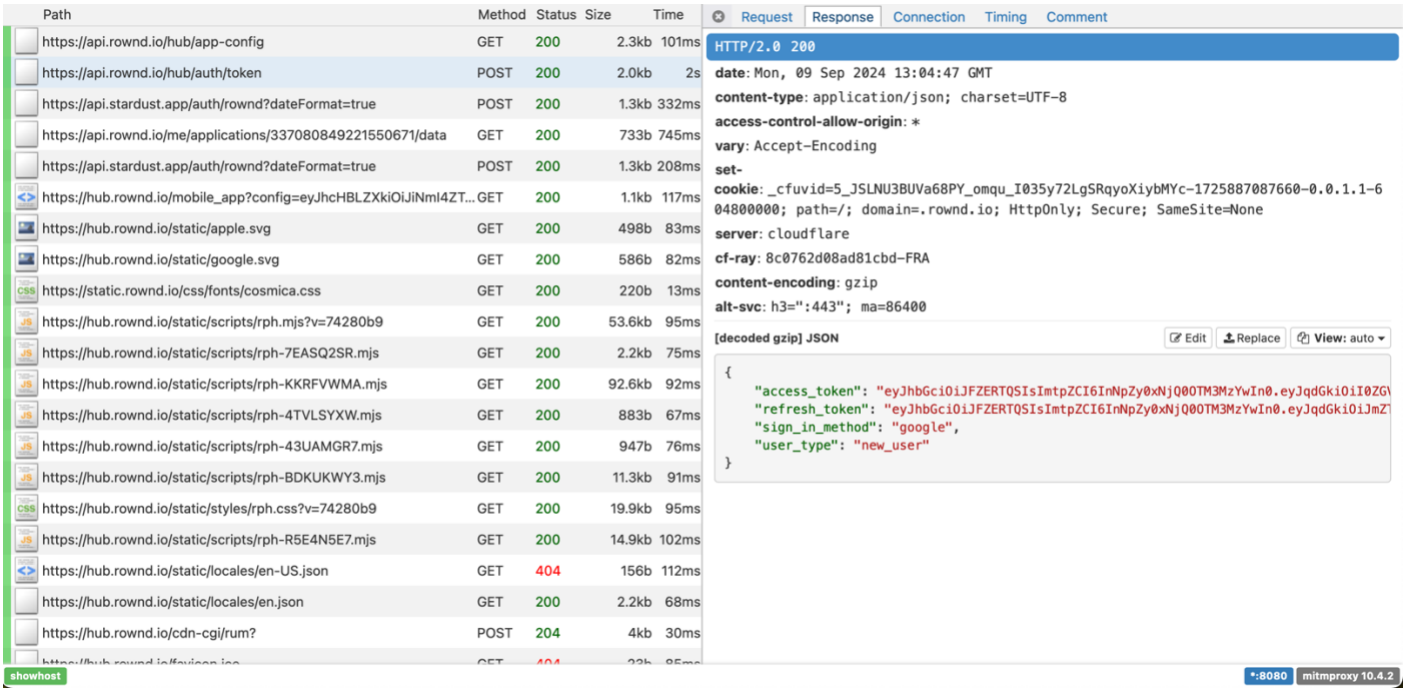


[Figure 7.1. Screenshot of the sign-up page with Google allowances.]

After connecting our account with Google, we completed Stardust's onboarding questionnaire, which asked for our purpose for using the app (e.g., for period tracking), our date of birth (which was required) and a range of period information like our latest period and birth control method. During this process, we noticed a plethora of URLs from Rownd's API in the web traffic:



[Figure 7.2. The Request shows Rownd is processing our 'intent: sign_up'.]

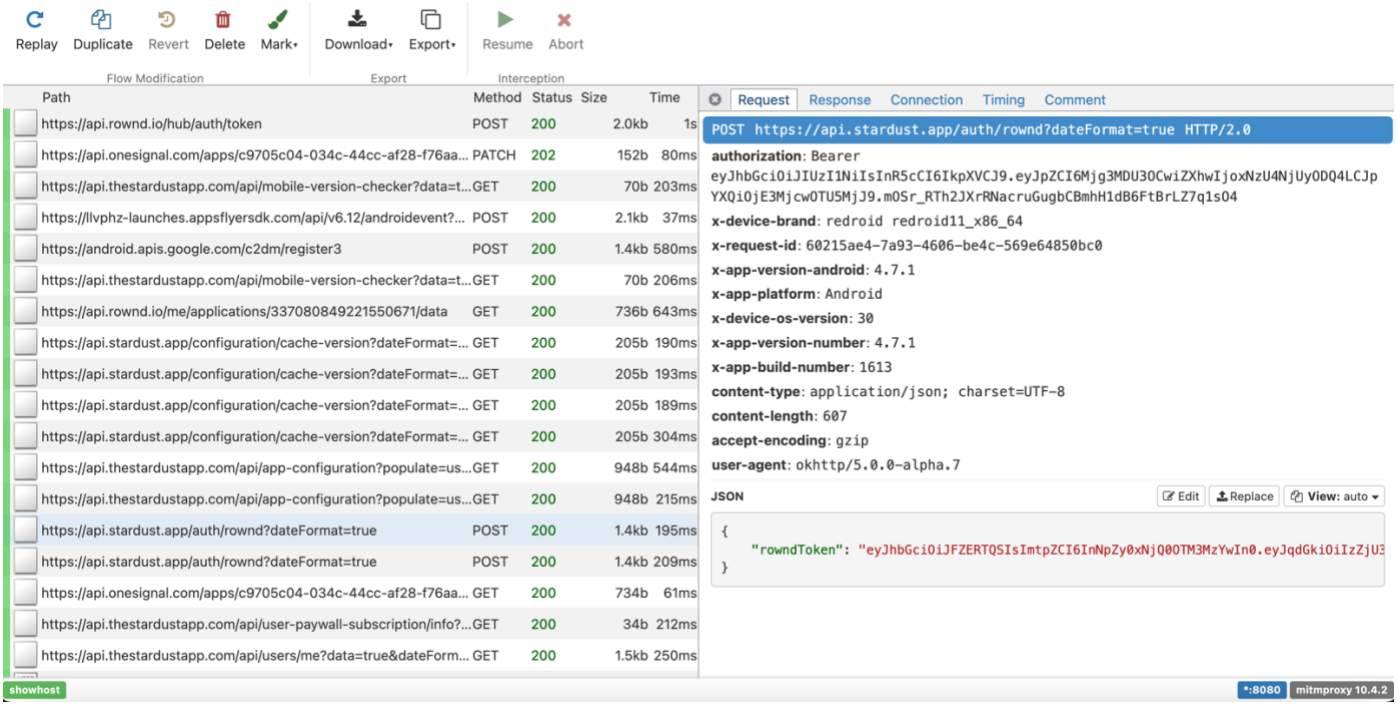


[Figure 7.3. Rownd's API responds with 'sign_in_method: google', as we signed up via our Google account.]

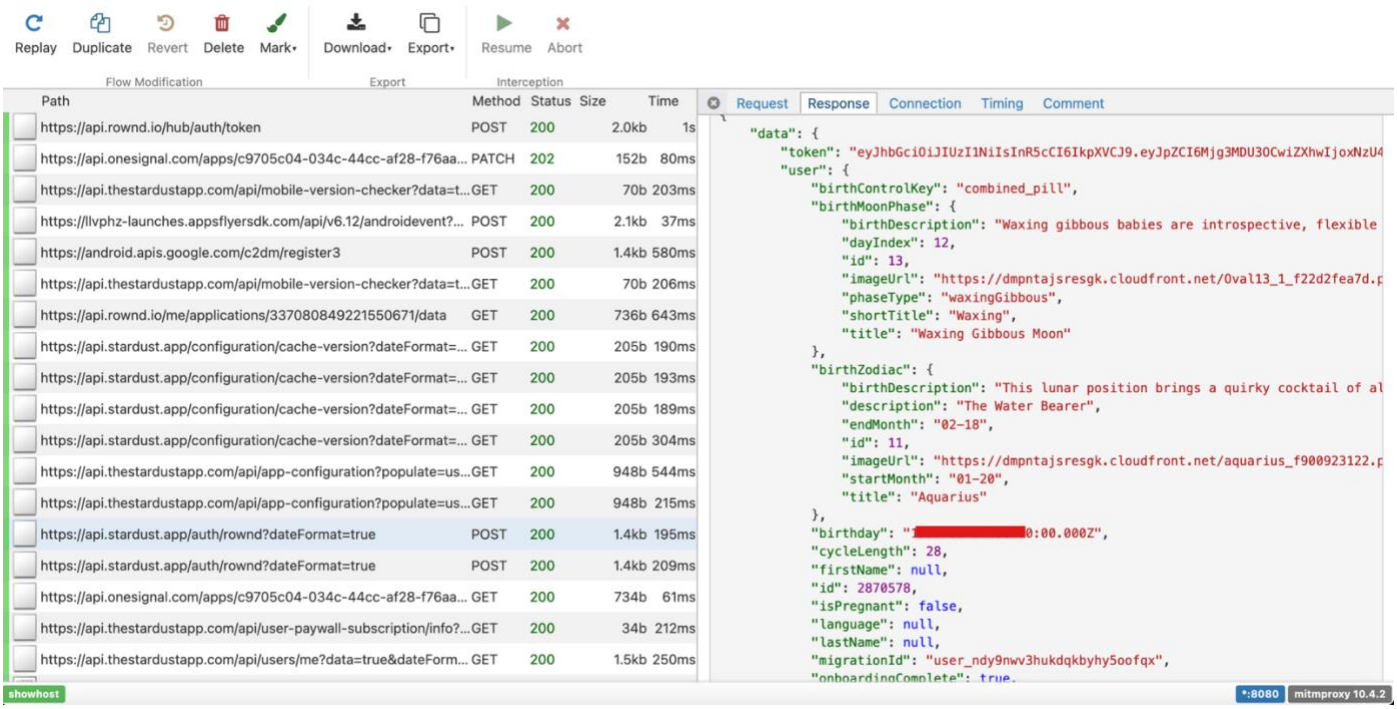
Note that all these and following calls to Rownd by Stardust occurred over the Cloudflare CDN, which is not named in Stardust’s Privacy Policy. Rownd utilizes Cloudflare for the delivery of their authentication services. We’ve already discussed Cloudflare’s security layer above, and

Rownd also clarified that robust TLS (transport layer security) encryption is used by Cloudflare between Rownd and its communications with the client, including between Cloudflare.

While using the app, with every update we inputted, the Stardust API requested a 'rowndToken' associated with our account. In theory, this links all our account data managed by Rownd (e.g., first and last name, log-in email) with that unique token:



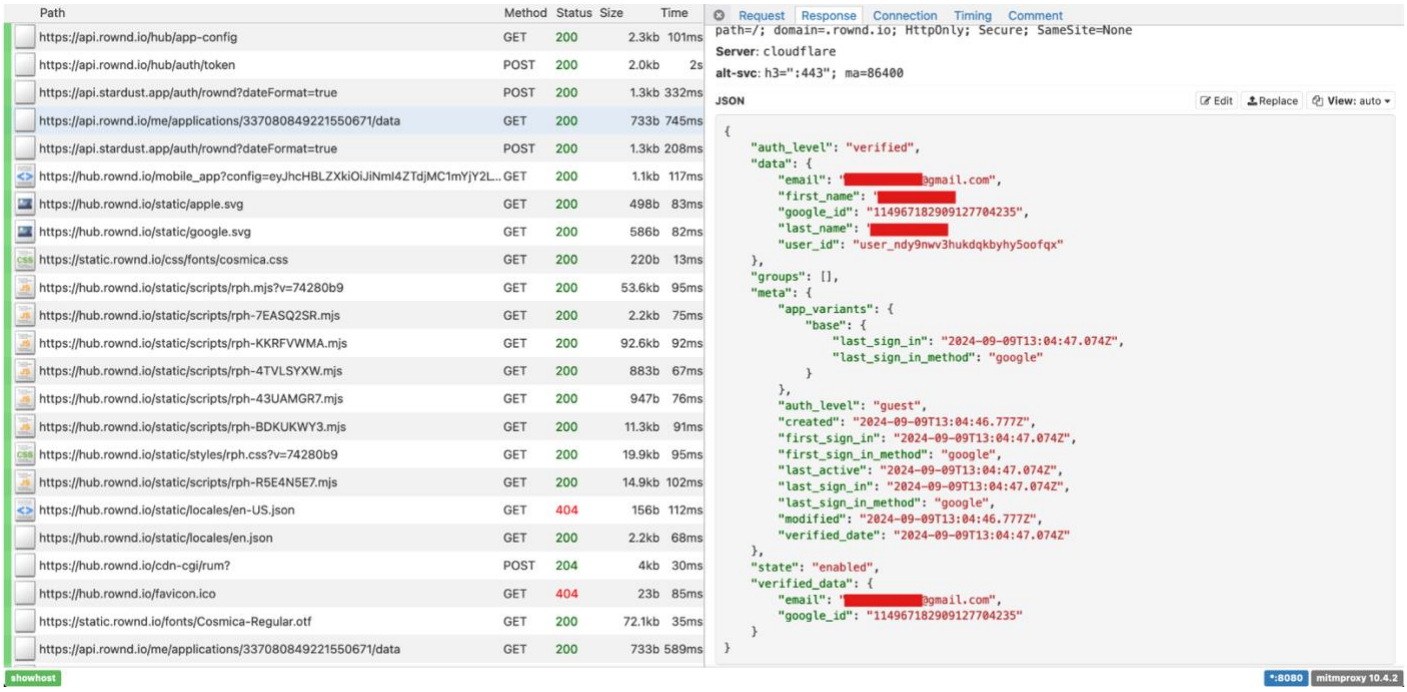
[Figure 7.4. The Stardust API requested a 'rowndToken' for this user.]



[Figure 7.5. The response shows the birth control entry we inputted for this unique rowndToken. Note that the Stardust API here reads 'null' for the data points for 'firstName' and 'lastName' but did seem to have access to the ‘birthday’ data point.]

In Figure 7.5, we can see that the Stardust API did not appear to have access to the first and last name of the user for the specific ‘rowndToken’ (‘null’ for these fields), but it did appear to output the accurate birthday of the user provided during our sign-up.

As for processing with the Rownd API, everything from our Google email to our last sign in time to our Google ID was being processed by the Rownd API over Cloudflare:

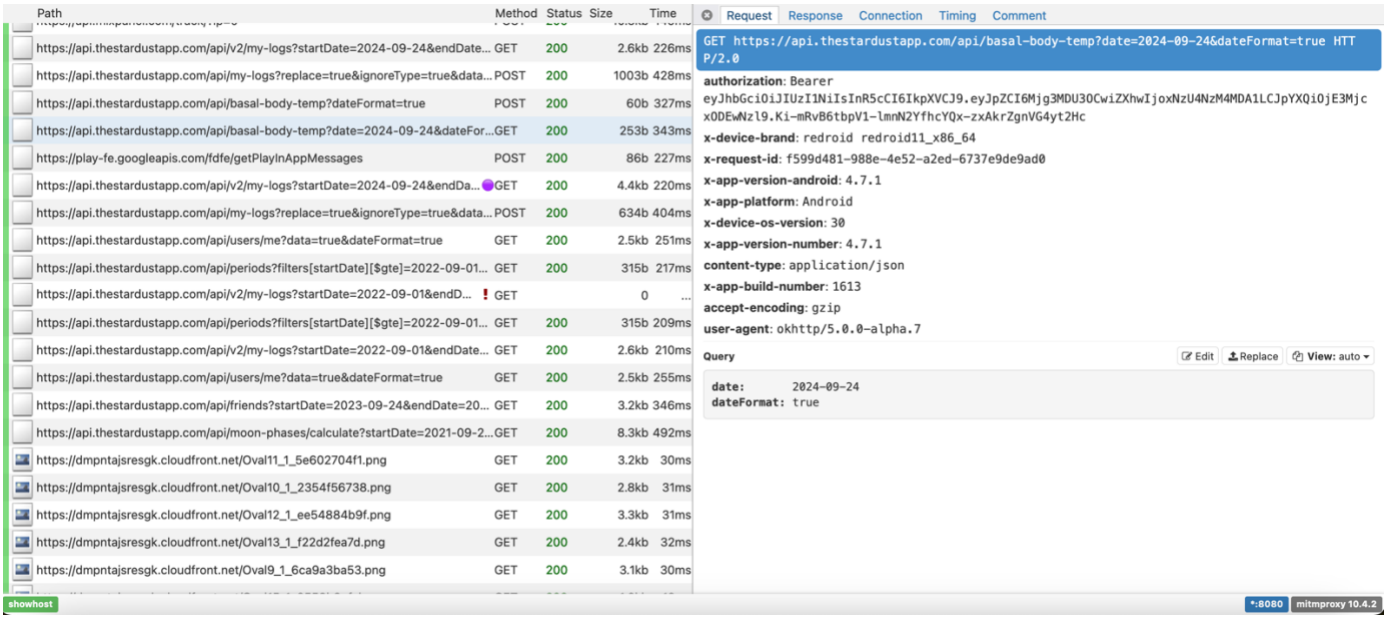


[Figure 7.6. See 'email', 'first_name' and 'last_name' which were extracted from our Google account after we linked the account to Stardust.]

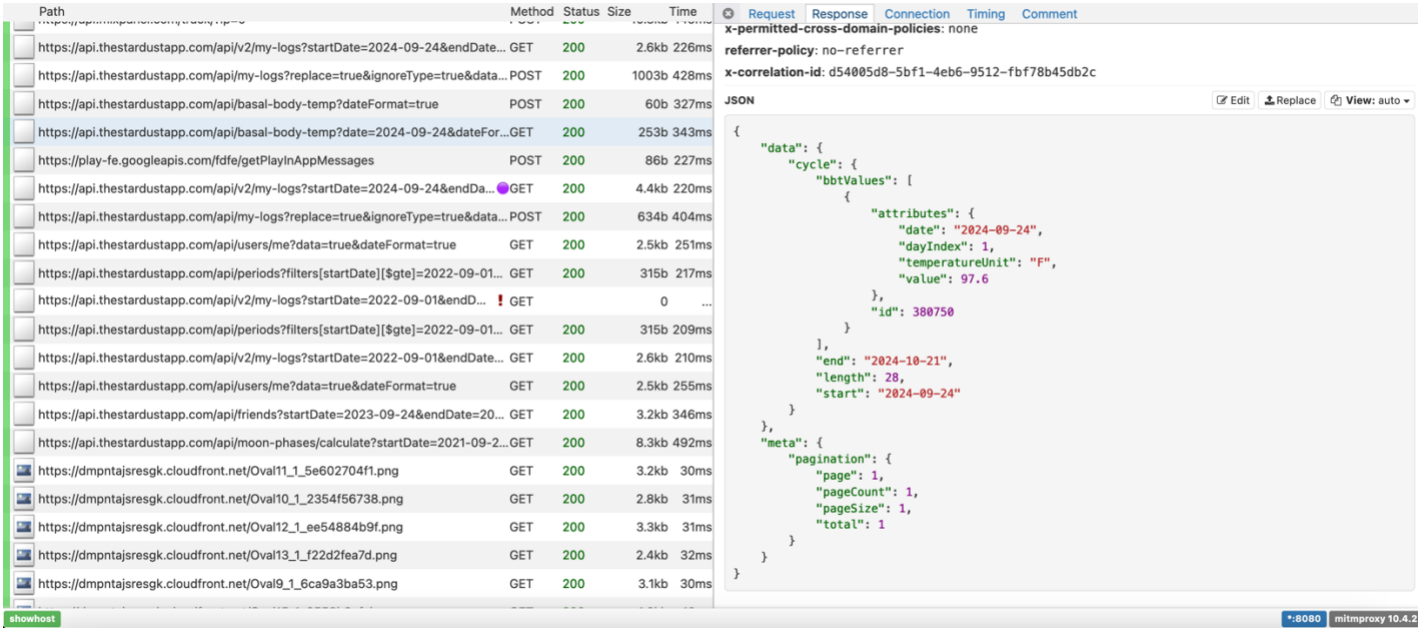
In Figure 7.6, we can also see detailed information about our sign-in and active times. These fields appeared in the web traffic for calls to Rownd every time we launched the Stardust app and logged in via Google.

As we'll discuss later, the offshoring of user authentication to a third party is an interesting approach to de-identification where, rather than not collecting account-based user data (e.g., email for sign-up), Stardust assigns this process off to a third party.

When it came to our in-app use of inputting our cycle information, our cycle data indeed appeared to be logged in the web traffic and sent to Stardust's own API as we exhibited above in Figure 7.5. In the below example, too, we can see our temperature information we recorded for the specific start date being sent to Stardust’s API:

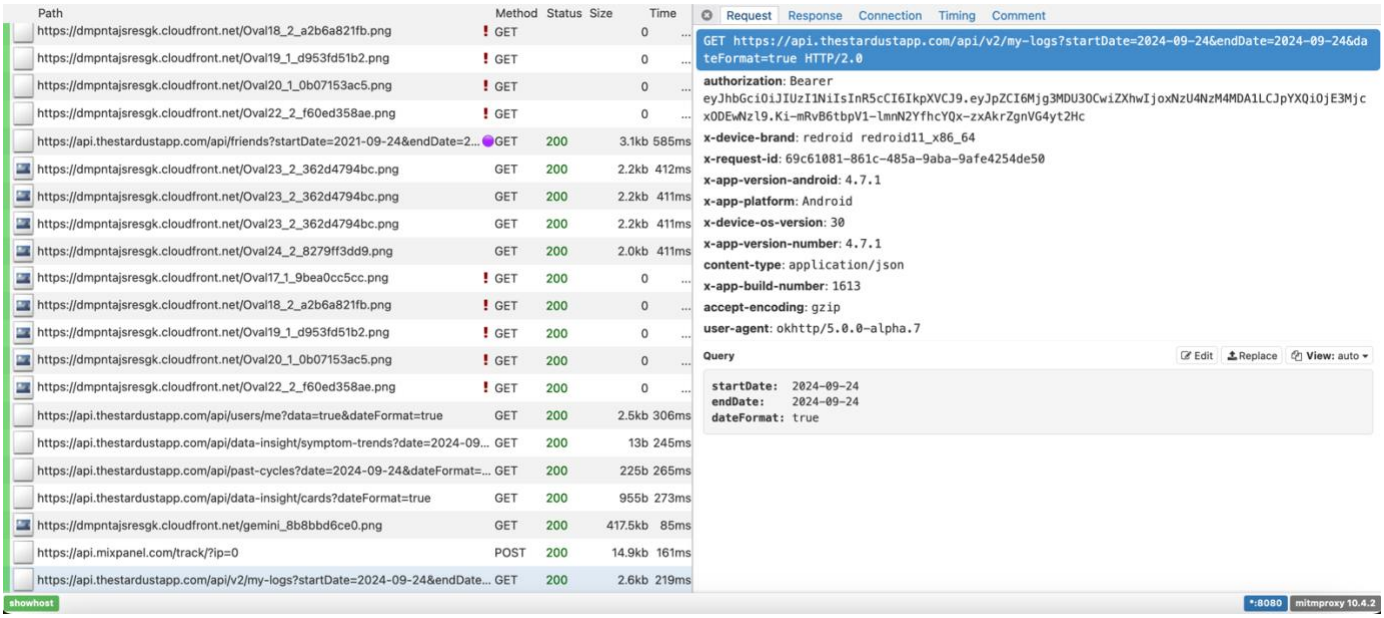


[Figure 7.7. The API requested the date for our entry.]

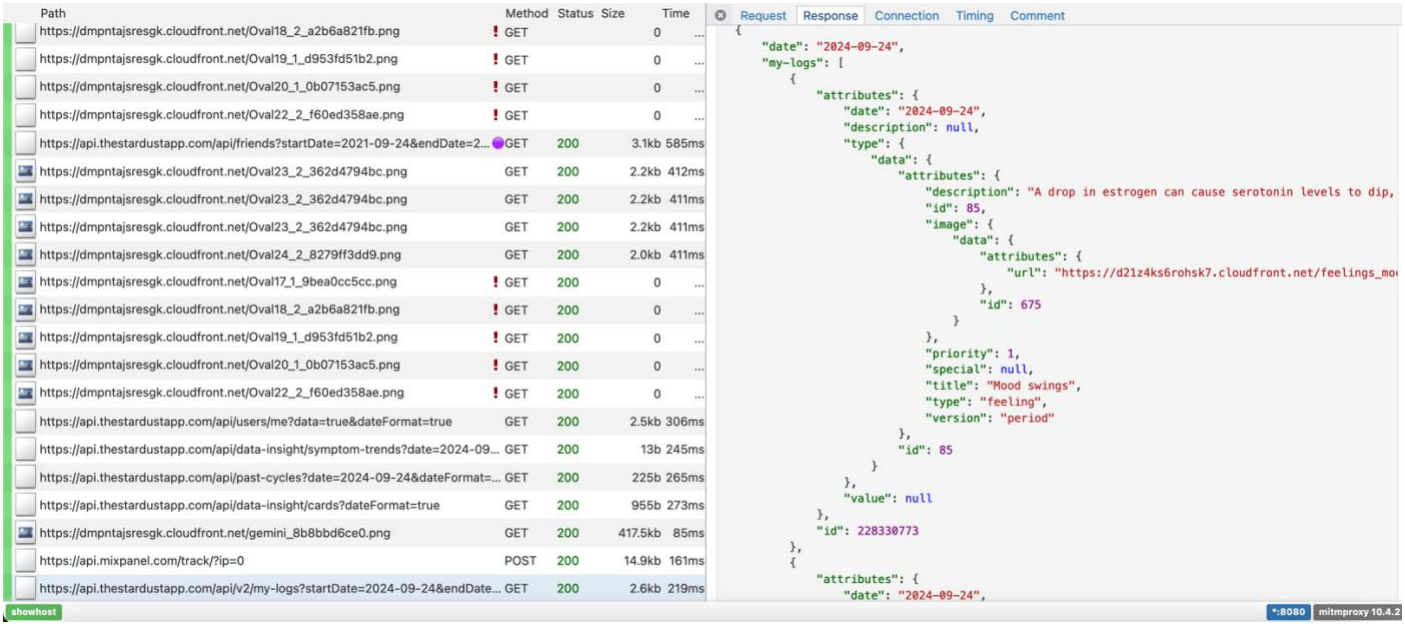


[Figure 7.8. In response, our inputted 'temperatureUnit' and 'value' was logged.]

We could also see all the symptoms we recorded and our period start date as well as predicted dates we'd recorded under 'my-logs':



[Figure 7.9. The dates requested for the entries to be logged for this month's cycle.]

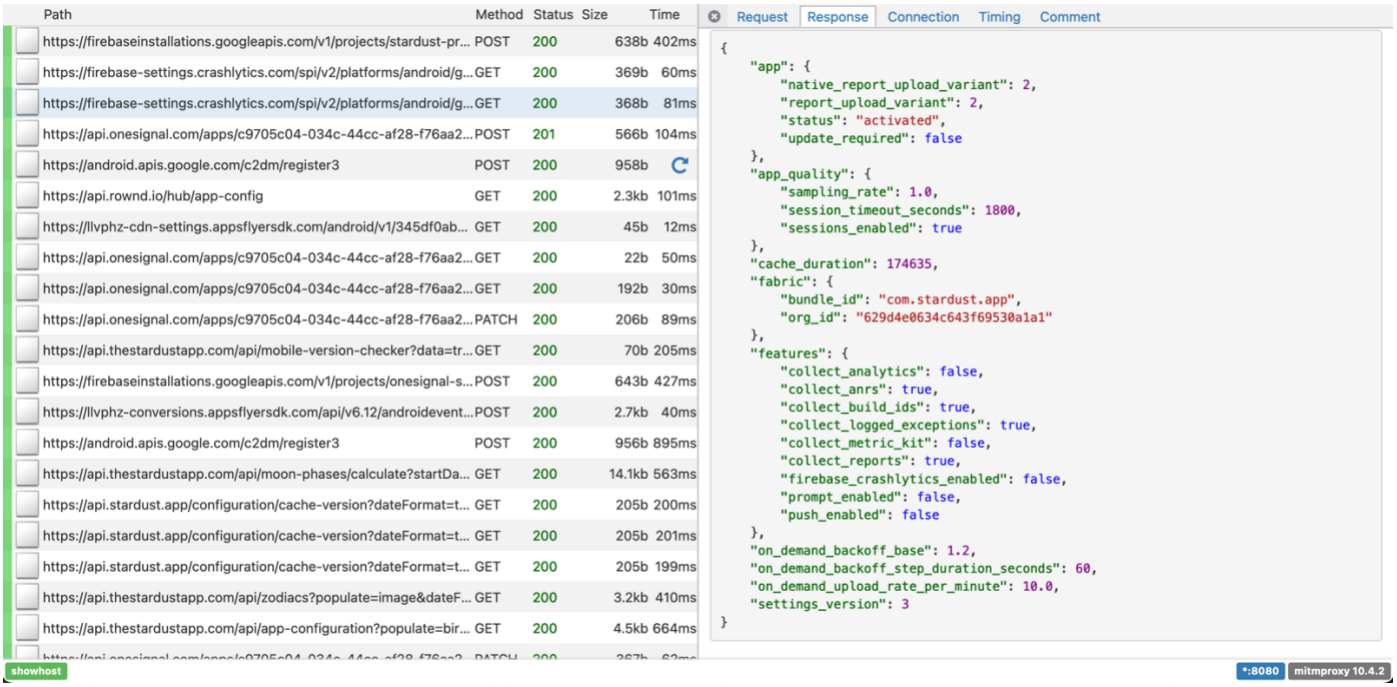


[Figure 7.10. The response that includes a list of all the symptoms ('attributes') logged for those dates under the entry 'my-logs'.]

We noticed a few appearances of other third parties, such as US-based MixPanel, which is an analytics service for developers to measure the performance of their product and marketing, with 'all your data, in one place'. It was difficult to discern in the web traffic what data was being requested, as the data in the request and response was encoded, so we cannot conclusively say whether the data sent to MixPanel represented device-related data or other data. We also noticed one appearance of AppsFlyer in the web traffic, though it did not appear that personal

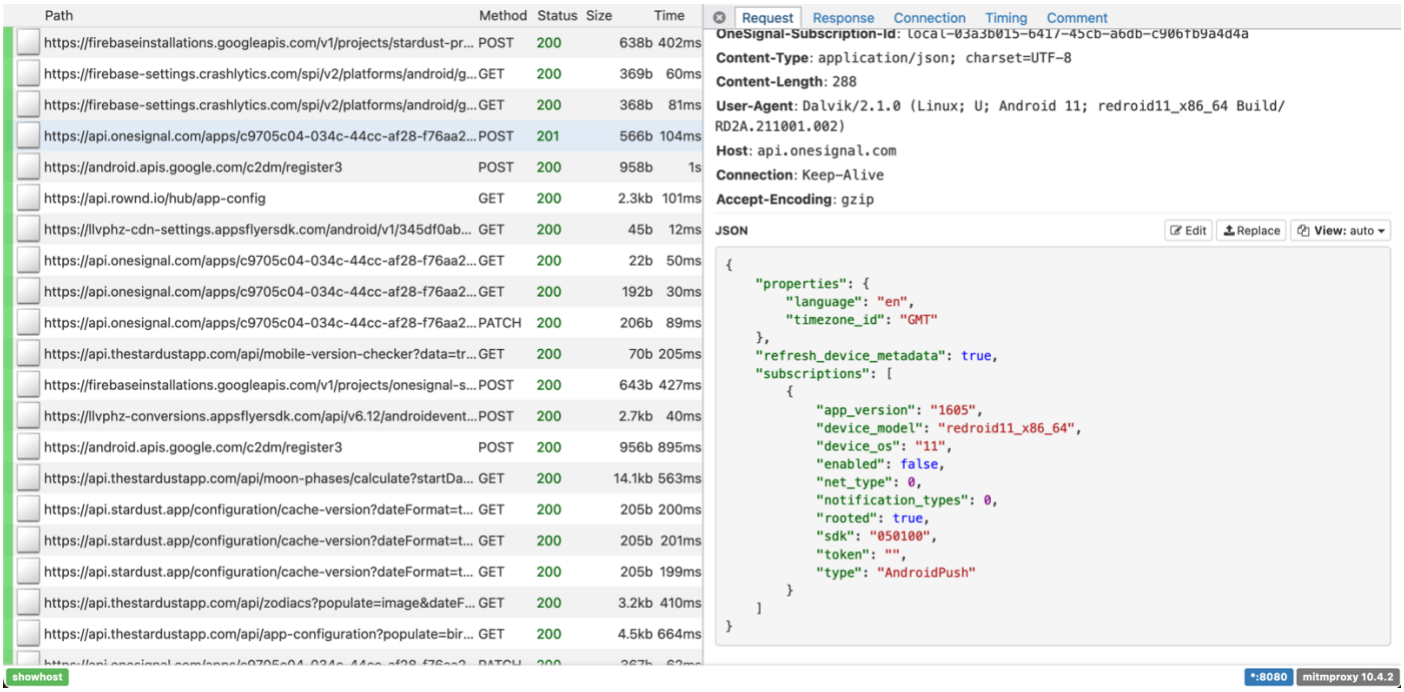
data of note was being requested. Notably, AppsFlyer was one of the few third parties mentioned by name in Stardust's Privacy Policy.

Like many of the above apps, Stardust also appeared to have integrated with Firebase, which meant the collection of certain device-related data for its Crashlytics crash reporting tool (see URL on the left-hand side):

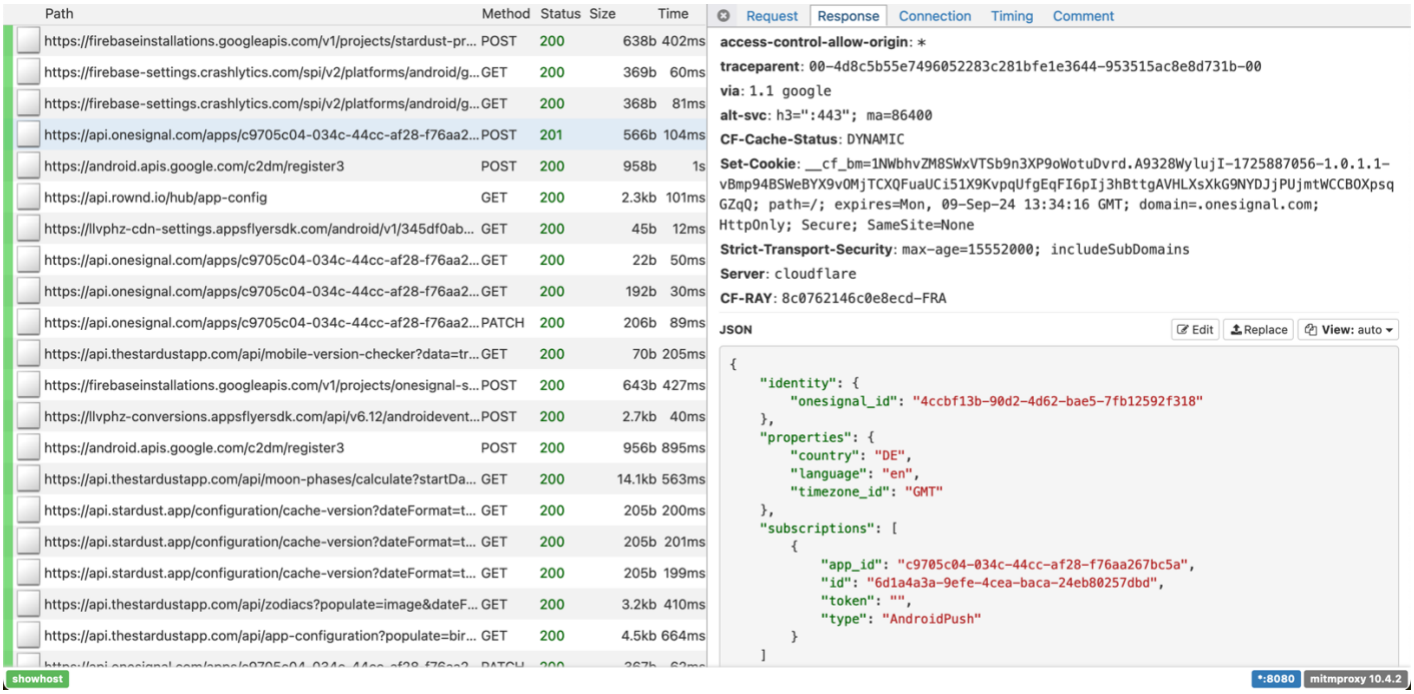


[Figure 7.11. See 'features' and 'crashlytics' in the URL.]

We also saw several appearances of US-based OneSignal, which is a third-party push notification integration for sending app notifications to a user. We did not enable notifications in our test, so the request from OneSignal occurred only when we first launched the app (likely as part of the app's build). However, even in this case the third party did appear to extract some device-related data:

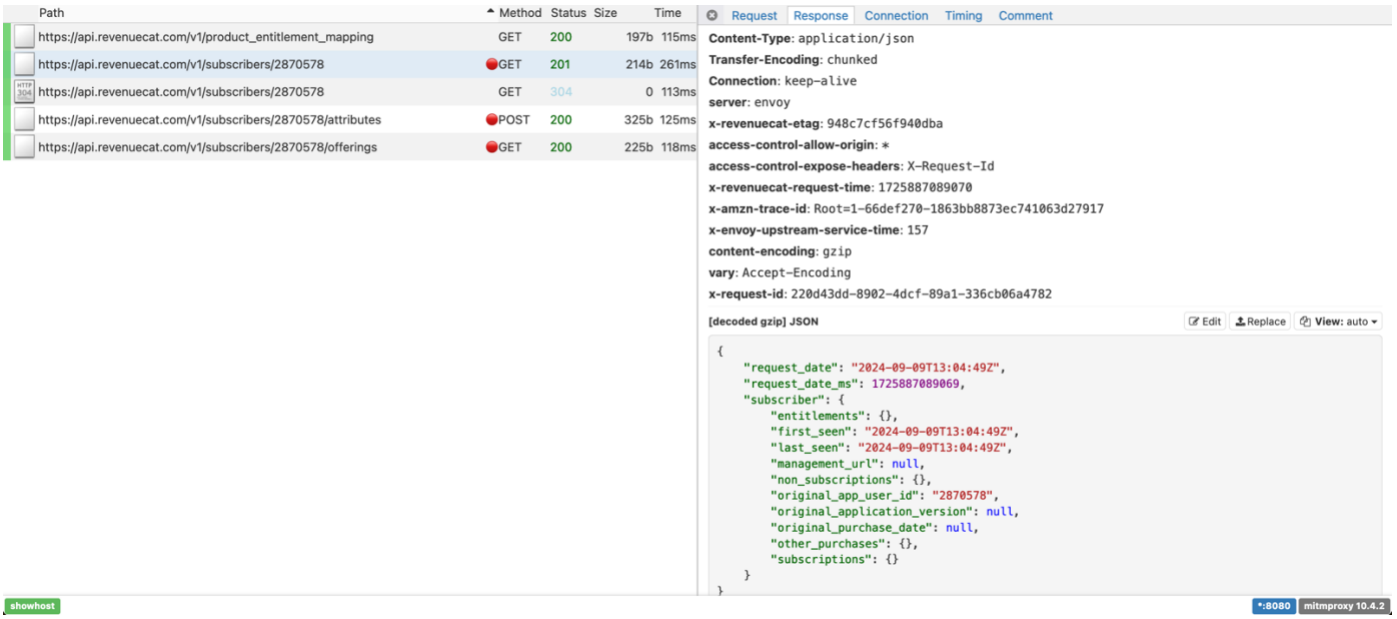


[Figure 7.12. See 'language', 'timezone', 'device_model'. Note we see 'enabled:false', which means we have not yet enabled push notifications. This does not mean the data is not being collected, just that notifications for this device have not been enabled.]



[Figure 7.13. The response for the above request assigned us a 'onesignal_id' for our device and other related information for when notifications are enabled.]

Another third party that appeared in the web traffic was Revenue Cat, which facilitates in-app purchases. We did not subscribe to anything in Stardust, so we cannot say with certainty what would be shared with Revenue Cat beyond the standard subscription-related information (e.g., payment info) and what we see below, such as sign in times. However, we did notice that Revenue Cat operated over third party cloud server 'envoy', which is run by US-based ride-sharing company Lyft:



[Figure 7.14. See 'server: envoy'.]

Note that none of these third parties (Rownd, Firebase, Mix Panel, OneSignal, Revenue Cat, envoy), nor Cloudflare, were named in the Privacy Policy. The only third parties disclosed were AppsFlyer and optional integrations like Google Fit, Apple Health and Oura Ring. There was general mention of tracking technologies like 'cookies, pixels and SDKs' being used to collect interaction information about users, but no specifics were provided. For an app that claims to engineer privacy into its implementation, it appears that its Privacy Policy perhaps falls short of comprehensive, transparent disclosures.

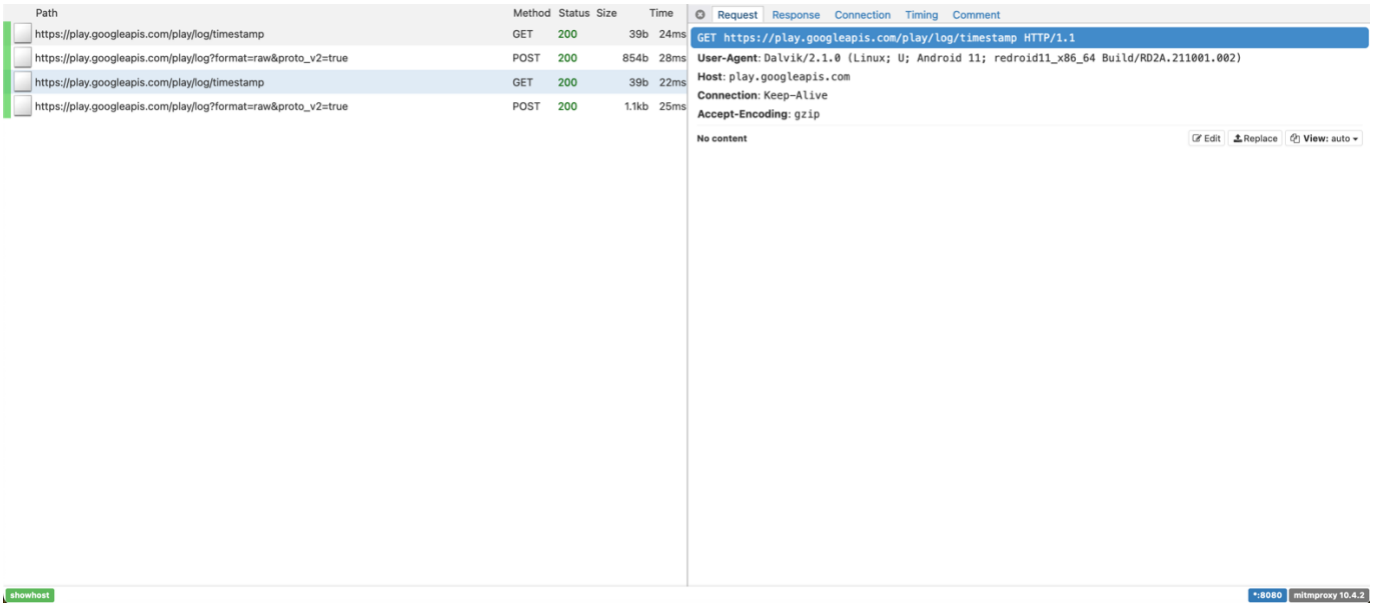
Euki

The last app we looked at was Euki, which has been recognised among privacy advocates. The app is a U.S. non-profit privacy-by-default period tracker app founded by a group of social tech and sexual and reproductive health organisations like Digital Defense Fund and Ibis Reproductive Health. The app has recently become open-source.

We ran the Euki app through the DIAAS environment to observe its web traffic as for the above apps. There were no onboarding questions to get started on the app except for agreeing to the

Terms of Services before use, and we were able to use the app without having to create an account (nor were we prompted anywhere on the app throughout our use to create an account).

Consequently, we did not find much in the web traffic while setting up and using the app. The only calls that were made across the web were the following calls that appear to be from the Play Store running in the background, not calls from the app itself:



[Figure 8.1. Screenshot of our web traffic window while using Euki. See 'connection: keep-alive' for Google API, which could be related to an instruction for the Play Store to keep running in the background.]

At this point, we had already inputted our blood flow and emotion symptoms. The only calls that appeared in the web traffic pertained to the Google Play Store API; these calls are from Google's Play Store which is running in the background. We did not observe any user data - nor device data - being sent to any parties.

Our findings mimic the disclosures in Euki's Privacy Policy, which states that the app does not collect personally identifiable information or anonymous information automatically, such as the type of mobile device the user uses, their unique device ID, the IP address of the device, or the operating system and web browsers the user is using.

4. Limitations

Before our analysis, we note the technical limitations (and the scope of our research) meant we did not test certain features mentioned, such as Google Fit integrations offered by some apps.

We also mention the limitations of our DIAS environment, which only allows us to see web (client-side) interactions, rather than server-side interactions, the latter of which are increasingly common among more advanced platforms that utilise cloud computing (e.g., server-side operations of OpenAI’s ChatGPT for WomanLog's chatbot). There may be some data-sharing activity server-side that our DIAS is unable to see and that can only be revealed by the developers themselves.

5. Analysis: What does this all mean?

Compared to our research in 2019, this time around we did not see instances of user’s personal data about their cycle being sent to Facebook for the above period tracking apps. However, in the web traffic of several apps we investigated, we found a significant number of third parties, from advertising software development kits (SDKs) to third party development tools.

We also found that the majority of platforms stored users’ cycle input data either on the app‘s own server or external servers, rather than storing this data locally on the user’s device. This introduces interesting questions about the risks to users when such intimate data might potentially be subpoenaed by law enforcement.

Advertising and analytics SDKs

Throughout our investigation, we found at least some, if not many, appearances of third-party URLs in the apps' web traffic. Most apps appeared to integrate some form of an advertising SDK (e.g., Google Ads, Facebook Ads) or an analytics SDK (e.g., Firebase, AppsFlyer). While the web traffic exchanges with these third parties did not appear to include the user’s period data, they did share significant technical identifier data about the user's device, such as the device they were using (e.g., ‘redroid_x86_46’, which was the virtual Android we used in our experiment) with associated fingerprint data (see Figure 3.22 in the ‘EN’ locale for one of many examples). This automatic collection of device data was disclosed in all the apps' privacy policies, all of which mentioned that device identifiers and IP addresses would be automatically collected through the use of the app.

There are two privacy issues we raise in relation to technical device data being automatically collected and shared with third parties: 1) the device data's identification properties, and 2) the way third parties might use this potentially sensitive data.

By definition, aggregate data about a device like its model or dimensions would not qualify as personal data under data protection laws. However, there are special cases in which device data may be considered as personal data, for example if the device becomes uniquely identifiable to a user via fingerprinting, which is a method that combines several different attributes of a device (e.g., screen resolution, IP address, operating system, device ID) to identify the unique device. While these individual pieces of data are not uniquely identifiable on their own, the probability of two different devices having the exact same combination of these attributes is statistically low, which thus makes it possible to identify a device. We note that not every app we looked at extracted such a wide web of information about the user's device, but some of them collected a fair amount of device data that was shared with third parties beyond those disclosed to users in the Privacy Policy.

Third-party Firebase appeared to collect device-related data like the device model, operating system and country of use for the purposes of app analytics. Analytics platforms typically generate aggregate analytics in order to provide developers insight into their app engagement (e.g., how many Android users downloaded the app, how many app crashes were reported in a specific country, etc.). An interesting scenario arises when we consider that Firebase Analytics is a Google-owned platform. The majority of global Firebase services run on Google infrastructure, which means the data could be processed by Google Cloud or Google data centers and Google could, in theory, use this data for its own purposes. Firebase's Privacy Policy discloses that 'Firebase Service Data', or personal information (excluding customer data) that 'Google collects and generates during the provision and administration of the Firebase services', might be used by non-Firebase Google services, such as to 'understand your use of Firebase and other Google services'. Firebase clarifies that customers (developers) do have the option to control whether their Firebase Service Data may be used by Google, and the Firebase Service Data that may be used for non-Firebase Google analytics is unlikely to be personally identifiable data. Regardless, the conclusion we draw here is concerned with the fact that device data pulled from Firebase through the use of its period tracking app clients could potentially be processed beyond the app itself and even beyond third party partner Firebase.

Overall, we were pleased to see that third party SDKs did not appear to be collecting period input data while we used the app (as far as our DIAS environment has been able to see). However, it is nonetheless worth noting that some of the observed third parties like Firebase are involved to some degree in the handling of troves of user device data unless the app developers manually configure these sharing settings.

We also note that some apps' privacy policies were more transparent than others (refer to our Findings for specific apps); several apps merely noted generic disclosures that 'third party service providers' might receive device data and other analytics data. And, where privacy policies did name third parties, there were often inconsistencies in what third parties appeared in the web traffic and which were disclosed by name in privacy policies (e.g., Google Ads

appeared in the web traffic for the Maya and WomanLog apps, but it was not explicitly named in their Privacy Policies).

Other third-party developer tools

Several apps also outsourced parts of their app functionality/development to third parties, such as to Facebook's Graph API (Maya, Wocute), push notifications integrations with OneSignal (Stardust), onboarding and authenticating users via Rownd (Stardust), health integration options with Google Fit (WomanLog) and AI features supported by OpenAI (WomanLog). Some of these apps linked their Privacy Policy somewhere on the landing page, but users nonetheless were not explicitly told within the apps themselves that their data (e.g., timestamps of opening the app, device data, onboarding information, chatlogs) would be shared with or processed by the specific third parties we saw in the web traffic. Additionally, only some of these third parties were explicitly named in the privacy policies or website, if at all (the privacy policies for Maya, Period Tracker by GP Apps, Wocute, and Stardust mentioned the use of third parties but did not mention by name the third parties we discovered in the web traffic). While such disclosures are not strictly required under GDPR, it certainly would be considered good practice to do so.

The outsourcing of certain app functionalities to third parties is concerning, as there is a lack of clarity around what data is being accessed or stored by these third parties. For instance, there may be a question of where WomanLog's chatlogs are stored and the potential privacy risks. Our findings above show that each chatlog in the WomanLog app was assigned a unique 'chatID', which suggests it may be stored according to that ID for some sort of referencing or for a certain time. This raises potential concerns for example if a user mentions in the chat that they've missed their period - might this become incriminating information that an app could potentially hand over to law enforcement as part of its profile on the user? It may become increasingly popular for apps to leverage AI to entice users to a more 'advanced' personalization experience, at the risk of potentially allowing this data to be handled by third party AI companies. Consequently, it is crucial to scrutinize the privacy ethos within these business partnerships and who holds access to what data.

It's worth noting that the outsourcing of certain functionalities like user authentication to third parties can be for the purposes of enhancing privacy, not diminishing it. Stardust has asserted that its use of third party Rownd for user authentication is for anonymising user's data and de-identifying users. In theory, according to Stardust's explanation on its website, by separating the user's personal account log-in data (stored by third party Rownd) from the user's period input data (stored by Stardust), Stardust cannot associate the unique user with their input data. As we'll discuss below, this is an interesting new approach to protecting privacy while still collecting and storing user data non-locally.

Nonetheless, we emphasise that involving third party deployers in any case – even in Stardust's case as a privacy-enhancing feature – means a higher threshold of privacy protection and accountability because there are now more parties involved that are processing user data.

Cloud-based content delivery networks (CDNs)

So far in our analysis, we've discussed the direct sharing of data with third party SDKs and deployers. A different category of entities we observed is cloud-based content delivery networks (CDNs), which appeared to facilitate the delivery of user data between the apps and the third parties mentioned above, as well as between apps and their APIs.

Cloud platforms are computing services (e.g., networking, data storage, etc.) that operate over virtual cloud servers. Cloud-based CDNs are functionally 'middle-men' that facilitate Internet requests between computers and servers over virtual servers (e.g., requesting specific images or icons to display in the app). Cloud-based CDNs, like Cloudflare, are becoming increasingly common as websites and apps grow their capacity and userbase. Cloud servers are highly scalable to accommodate for growing infrastructure needs that today’s apps increasingly require (e.g., a large number of users sending requests to an API all at the same time).

For many of the above apps, we saw cloud services either facilitating calls between apps and the third parties they're integrating (e.g., Aliyun for Wocute, Cloudflare for Rownd) or communicating user input data first-party across the API (e.g., Cloudflare for Flo).

In its response to our findings, Cloudflare clarified that its 1.1.1.1 public DNS resolver does not retain any personal data about requests made, and that its Oblivious DNS over HTTPs (ODoH) separates IP addresses from queries ‘so that no single entity can see both at the same time’. In theory, according to Cloudflare’s response, this approach ‘allows for legal compliance without detracting from Cloudflare’s or the app’s delivery of services’. On top of this, Cloudflare also stated that it does not access the data being transmitted in its everyday operations, nor is it possible for them to turn over any content transiting across their network anyway, as it would not be technically feasible to do so.

Most of the apps' privacy policies did not mention that user data would be processed by and passing through cloud-based services. Apart from Flo, all other apps that used cloud services did not explicitly mention the cloud service, and some did not even mention the use of cloud services at all. It is crucial that users be informed of all the entities that are processing their data at each stage.

Cloudflare's Privacy Policy clarifies that it is the responsibility of the business customers themselves (the apps) to 'establish policies for and ensure compliance with all applicable laws and regulations, including those relating to the collection of personal information, in connection with the use of our Services.' Cloudflare establishes its position as merely a 'conduit of information', thus the period tracking app itself, not Cloudflare, is responsible for what user data is being collected and passed through the server. As a man-in-the-middle proxy server, Cloudflare only handles the traffic that their customers elect to send across the network. And we reiterate that Cloudflare states it does not see or access the data being transmitted, nor is it technically feasible for them to hand over this data anyway due to the complex nature of their security-by-design architecture.

On the note of law enforcement subpoenas, Flo asserts that any data disclosure requests would be sent to Flo as the data controller and not Cloudflare as the data processor. Cloudflare also asserted that it has a long history of pushing back on government surveillance orders of traffic in their network. They state that:

> *'Cloudflare has never provided any government a feed of our customers' content transiting our network or installed any law enforcement software or equipment anywhere on our network…We would legally challenge a request to take those actions if we were to receive one.'*

Note that for the apps we tested above that show 'nginx' in their server rather than an explicit third-party server, it is possible the apps are using cloud services at other stages of their delivery that were not observable in the DIAS environment.

Non-local storage

Above, we saw that many apps (Flo, Maya, Wocute, WomanLog, Stardust) communicated user data (device data and input data) to their respective APIs, even in cases when the user did not create an account on the app. Recall that an API is a connection that allows software to communicate features with each other (e.g., actions on an app communicating with the computer servicing functions for the app). In these scenarios, it is possible that the data being communicated over the web to the API is being stored somewhere. Thus, all the user's input data we've observed above in our findings could be processed and stored somewhere by the developer. This data might be stored simply as part of a user's profile, or it might even be used for further purposes, such as, in WomanLog's case, to generate training data 'for predicting menstruation and fertility', according to their privacy disclosure. (We note that OpenAI asserted that it does not use inputs and outputs from deployments of its API Platform to train its models).

Other apps (Simple Design's Period Tracker, GP Apps's Period Tracker and Euki) did not appear to process user input data over the web. Both Simple Design and GP Apps have stated that the data users input about their cycle into the app is stored locally on the user's device only and thus not accessible nor processed by the apps themselves. Euki also clarified that the technical functionalities of the app have been engineered in such a way that the app is not capable of collecting user information in the app.

Although there is nothing inherently wrong with using an API to retain users' data, it does mean that all cycle input data is leaving the mobile device and being stored off-device on an external server overseen by the app. The significance of off-device (non-local) data storage as opposed to on-device local storage is that this data is not solely in the hands of the user and their physical device, but also stored and processed by the app developers. If complying with a law enforcement subpoena means turning over all the data an app has about one of its users, this could hypothetically mean the app turns over the cycle data requested that's in its possession. If data is stored locally and only on-device, the app would not have access to the health data about their users and thus would not have the data to hand over that law enforcement might be looking for.

While the most privacy-preserving practice may be to keep user data local to the device, some users may consider this a difficult choice to make for the convenience of backing up their data to an account that is accessible beyond their device in case they lose their phone or switch devices (when data is stored locally on a device, this means the data is not recoverable if a device is stolen or lost). When responding to our findings, Flo claimed that their use of an API rather than device-only storage is that the nature of lower-end Android devices such as those

used by individuals in 'locations where health literacy is low necessitates server-driven features' in order for Flo's optimal functioning of the app for these lower-end devices. While this is definitely an important consideration, it would be useful to separate – based on users' choice - between functionalities that can be delivered via APIs and functionalities that can be device based, associating only the latter with the processing of sensitive data. In any case, users may opt to use Flo's Anonymous Mode rather than this default mode, which still stores user data on Flo servers but will not store any personal identifiers such that the logged data would be difficult to connect or link to any user.

Storing personal identifiers is an interesting privacy topic to consider, as in contrast we saw a case (WomanLog) where a user was assigned a unique ID ('clientKey') despite not creating an account yet. While we inputted period flow and symptom information without an account, our inputs were being synced to this unique ID. By definition, such an ID is not personal information, but it might be considered so if it can be used to trace back to a unique user with that specific ID. Even when users have not created an account for themselves, they might still be potentially identified and their period patterns traceable if their input data is linkable to a unique identifier.

Data minimisation

One of our recommendations from our previous investigation urged menstruation app developers to limit the data they collect on users, as many apps appeared to request superfluous personal data despite the fact that not all this should be necessary for the purposes the app states (tracking menstruation). While we saw that some apps (Flo, Period Tracker by Simple Design, Period Tracker by GP Apps, WomanLog, Wocute) allowed users to use the app without having to create an account, others (Maya, Stardust) required users to do so in order to use the app. We've discussed above how users' input data can be exploited by apps who store their data whether or not the user creates an account; requiring users to create an account with personal details like emails and names only adds further exploitable personal data to the mix.

Recall for several apps above that the onboarding stage asked the user for their purpose for using the app, such as tracking their period or preventing pregnancy. While this information might be collected for the purposes of delivering the right application dashboard, this metric also carries risks, as it could be packaged with all the other data and future behaviours logged for this individual like when they've missed a period to infer certain conclusions about them without their knowledge. If law enforcement issues a subpoena to an app asking for user data, it is these packages of profile information that could be handed over to potentially criminialise individuals who have been using the app and are being investigated for accessing an abortion where abortion is criminalised or restricted, violating both their right to privacy and their right to health.

Many of the apps we observed also asked users to provide personal information like their name, date of birth (or year of birth) and height/weight information to get started, with some apps even requiring birth year information (Flo, Maya, Stardust). Recall that we'd seen birth year and other personal information logged in the web traffic and sent to the apps' APIs and even in some cases third parties involved in the processing (e.g., Rownd processing this onboarding data for Stardust). Flo clarified that asking for birth year information is for the purposes of verifying the

age of its users (only those aged 16 or over may use Flo). They state that asking for birth year but not the uniquely identifiable full birth date strikes a 'balance between age appropriate design and data minimization principles.'

The future of privacy

This brings us to the conversation about the future of privacy for period-tracking apps. On the one hand, menstruating individuals deserve to have technology that can assist their menstruation tracking and health monitoring. On the other hand, this technology in the form of apps may also have their own profit-driven purposes beyond providing health tracking services. Is there a future in which these paradoxes can be resolved?

Above, we observed a few different approaches to privacy from various apps. One such method was allowing users to use an app without creating an account, which helps to keep them potentially anonymous, as their input data may not be easily linked to their profile. However, certain other identifying information like their device information and even unique account IDs assigned to the user nonetheless established a form of unique identification that could potentially be traceable (not to mention that there have been studies that prove how anonymization is not actually entirely impenetrable).

Some apps were configured in such a way that stored user data locally on the device, rather than storing the data on servers managed by the developer and/or third parties. However, this local storage option means that user data is not recoverable because the data is not backed up to any account. This trade-off may not always be preferrable for individuals who are utilising a period tracing app for consistent monitoring of their sexual and reproductive health.

We also observed a fairly novel technique deployed by Stardust, which offshored their user authentication functionality to third party Rownd to store and process users' account data (e.g., name, sign-up email, etc.) that Stardust itself cannot access. With this method (the 'anonymous sign-in' approach), Stardust can require users to create an account while siphoning off account management to a third party such that the user's input data cannot be linked to their account, whose identifying data (e.g., name, email) is being managed separately by Rownd. Rownd clarified in their response that their platform is engineered in such a way that strictly isolates identifiable user data from sensitive personal data managed by Stardust, so that the risk of linking identifiable data with sensitive personal health information is significantly minimized.

A fourth privacy method to raise here is not so much something we've observed directly in the web traffic but something we're monitoring as an emerging trend: open-source apps. Non-profit apps like Euki are open source, as is Flo's Anonymous Mode. Open source means the source code is made available to the public, thus security vulnerabilities might be better observed by more people, and users might also be able to look into how their data is being handled according to the source code. There is plenty of trial and error in the open-source argument, though, and there is no right answer about whether open source is the best case when it comes to apps managing sensitive health information.

It is difficult to say what the future of privacy holds, whether menstruation apps might turn to more privacy-enhancing features and services with a privacy-forward mission due to public

pressure or whether the payoffs (and ease) of exploiting users' data are too lucrative for some apps to sacrifice. Some platforms have stated that they do not support law enforcement overreach for criminalizing abortion, such as Clue and Period Tracker by GP Apps, and Cloudflare has stated that they treat their customers' privacy with equal force to law enforcement requests. These privacy-forward goals should be the standard and not the outlier. The current regulatory landscape does not enforce enough accountability and responsibility upon apps to pursue better privacy practices; in order for developers to engineer robust privacy in their apps as a default, there must be explicit regulatory standards and safeguards that make privacy attractive to developers and exceptions permitting the sale and sharing of users' data to third parties should be reconsidered and narrowed.

6. Conclusion

Our research has introduced questions about the right to privacy when apps have the potential to share a range of user-related data. This is a particular concern for people using apps in countries where there are restrictions on access to abortion. In the US, after the overturning of Roe v Wade, concerns around the privacy practices of period-tracking apps have been raised in states that have introduced restrictions and bans on access to abortion. It could be very possible for some period tracking apps to hand over incriminating sexual health data they have on their users (e.g., weeks of missing period entries) for the purposes of complying with a law enforcement investigation, which could result in a violation of a woman's right to privacy and right to health.

Additionally, seeing such a wide range of third parties integrated into the apps' web traffic raised alarm bells due to how many different entities had potential access to various categories of user data, such as device data. Device data is potentially traceable to a particular person due to their unique device identifiers. Automatically collecting data without explicitly transparent disclosures to users, beyond just small mentions in the privacy policy, and before they can meaningfully consent, is bad practice. While we have seen improvements since our last investigation that this time user period input data was not being sent to third parties like Facebook, we also caution that other categories of data like device data are still being shared with third parties by some apps at different stages of their functionality and data management.

We have observed some privacy-protecting features in apps and services as discussed above, such as enhanced network security and encryption protecting user data and the anonymization and de-linking of users and their identifying account data. This suggests promising potential for the future of privacy, but in this increasingly hostile environment around women's health, privacy-forward techniques should be the standard for all apps.

The risks and trade-offs for companies opting to take a privacy-forward approach are in constant tension with the ability to make a profit, as data has become one of the richest currencies for advertising and analytics, among other third party uses. However, when it comes to menstruation data, which is sensitive health data, and given the increasingly hostile environment for reproductive rights and the risks of this data being used against an individual, period tracking apps should be held to a higher standard of privacy protection. Users should not have to sacrifice privacy for period tracking, nor should apps risk users' data in a way that leaves them vulnerable to violations of their rights. And the current regulatory landscape must be prepared to enforce a more privacy-forward deployment of period tracking apps in light of the shifting political tides that are putting women's right to health at risk.

Privacy International
62 Britton Street
London EC1M 5UY
United Kingdom

+44 (0)20 3422 4321

privacyinternational.org