

Automation of Manually Fetched Data

Harshit Chopra
dept. of Electronics and Communication
R.V. College of Engineering
Bangalore, India
harshitchopra.ec19@rvce.edu.in

Shushrutha.KS
dept. of Electronics and
Communication
R.V. College of Engineering
Bangalore, India
shushruthaks.ec19@rvce.edu.in

Abstract— As the number of Android devices continues to grow rapidly, developing applications that are compatible with a vast range of devices is becoming increasingly challenging. In this paper, we introduce a solution to this problem by presenting a method for fetching device capabilities in Android Studio. One of the significant challenges that developers face is that the Android platform has a fragmented ecosystem, which makes it difficult to test and optimize the application for each device. To mitigate this issue, developers need access to device-specific information such as screen size, resolution, available sensors, and other important hardware and software features. In this paper, we introduce a solution to the problem of fetching device capabilities in Android Studio that can help developers improve their application's compatibility and user experience.

Keywords— Android Studio, application's capabilities, screen size, resolution, sensors.

I. INTRODUCTION

In today's digital era, mobile devices have become an integral part of our lives, serving as essential tools for communication, productivity, and entertainment. With the growing reliance on mobile technology, businesses and organizations are increasingly seeking ways to streamline and automate processes related to mobile details. This includes the automation of manually fetched mobile details, such as device specifications, user information, and usage data. By leveraging automation, organizations can improve efficiency, accuracy, and decision-making when dealing with mobile-related information.

The automation of manually fetched mobile details involves the use of technology and software systems to retrieve and process data from mobile devices or associated sources without human intervention. Traditionally, gathering mobile details required manual efforts, such as physically inspecting devices or manually recording information. However, automation has emerged as a powerful solution, offering numerous benefits and transforming how businesses handle mobile-related data.

One significant advantage of automating the fetching of mobile details is the ability to save time and resources. Manual collection of data from numerous devices can be time-consuming and labor-intensive, especially in situations where a large number of mobile devices are involved. Automation tools can rapidly and efficiently extract relevant information, such as device models, operating systems, storage capacities, and network details, from multiple devices simultaneously. This

allows organizations to process a higher volume of mobile details in a shorter time frame, freeing up valuable human resources for other critical tasks.

Moreover, automation minimizes the risk of human error and ensures data accuracy and consistency. Manually fetching mobile details can be prone to mistakes, such as typos, omissions, or misinterpretation of data. Automation tools can be programmed to follow predefined rules and protocols, reducing the chances of errors and inconsistencies in the retrieved information. This accuracy is particularly crucial for businesses that rely on precise mobile details for decision-making, customer support, or inventory management.

Additionally, automation enables real-time data updates and monitoring of mobile details. Mobile devices generate vast amounts of data, including usage statistics, performance metrics, and user behavior patterns. With automation, organizations can collect and analyze this data in real-time, providing valuable insights into customer preferences, app performance, and overall mobile usage trends. Real-time monitoring also allows businesses to promptly identify and address issues, such as security vulnerabilities or performance bottlenecks, ensuring a seamless mobile experience for users.

Furthermore, the automation of manually fetched mobile details supports efficient device management and enhances customer service. By automating the retrieval of mobile details, organizations can maintain an accurate inventory of devices, including their specifications, location, and status. This information can be invaluable for device tracking, maintenance scheduling, and asset management. Moreover, automation facilitates the provision of personalized and efficient customer service by providing quick access to customer-specific mobile details, such as device preferences, service history, and support tickets. This enables organizations to deliver a tailored and responsive customer experience.

Automation of manual data fetching refers to the process of using technology and tools to automate the retrieval of data from various sources, eliminating the need for manual intervention. Manual data fetching typically involves time-consuming and repetitive tasks that can be prone to errors. By automating this process, organizations and individuals can save time, improve efficiency, and ensure data accuracy.

However, there are also challenges associated with automating manual data fetching. These include handling complex data structures, maintaining data security and privacy, adapting to changing data sources and formats, and ensuring the reliability and robustness of automated processes.

Benefits of automating manual data fetching include:

Time savings: Automation reduces the time required to retrieve and process data, allowing individuals and organizations to focus on higher-value tasks.

Increased efficiency: Automation eliminates human errors and minimizes the need for manual intervention, resulting in improved accuracy and productivity.

Real-time data availability: Automated data fetching ensures that the latest information is readily accessible for analysis, decision-making, and reporting purposes.

Scalability: Automation can handle large volumes of data and repetitive tasks efficiently, allowing for scalability as data requirements grow.

Improved data accuracy: Manual data fetching is prone to human errors, while automation ensures consistent and accurate data retrieval and integration.

However, there are also challenges associated with automating manual data fetching. These include handling complex data structures, maintaining data security and privacy, adapting to changing data sources and formats, and ensuring the reliability and robustness of automated processes.

II. METHODOLOGY

Our approach is based on the Android API and Google Play Services. The Android API provides access to device-specific information, such as screen size, resolution, available sensors, and other hardware and software features. Google Play Services, on the other hand, provides additional functionality for fetching this information, including the ability to retrieve the device's GPS location and network information. To fetch the device capabilities information, we first check if Google Play Services are available on the device. If they are, we use the Google API Client to connect to the Google Play Services and retrieve the necessary information. If Google Play Services are not available, we use the Android API to fetch the device capabilities information directly. Our approach is based on the utilization of existing APIs and libraries. The approach uses the Device API in the Android operating system to fetch device details. The Device API provides information about the device type, operating system version, screen resolution, and other relevant details. We use this API to extract device details that can be used to develop software that works well across different platforms.

Our method works in the following way: when the software is launched, the code checks the operating system type and version of the device on which the software is running. Next, it checks the screen resolution and other device details using the Device API. The extracted details are then used to optimize the software for that particular device. This can include adapting the user

interface, changing the size of graphics assets, or other changes to improve the user experience.

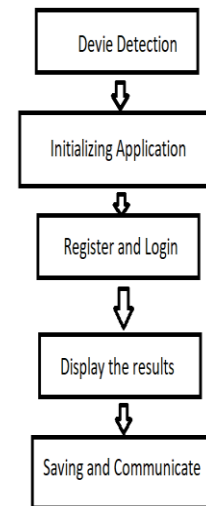


Figure 1: Flow of Project

Once we have the device capabilities information, we can use it to optimize our application for different devices. For instance, we can use the information to adjust the UI layout, use different graphic assets, or even tailor specific features for specific devices.

Automation of manual data fetching refers to the process of using technology and tools to automate the retrieval of data from various sources, eliminating the need for manual intervention. Manual data fetching typically involves time-consuming and repetitive tasks that can be prone to errors. By automating this process, organizations and individuals can save time, improve efficiency, and ensure data accuracy.

III. RESULTS

Our approach provides developers with a simple and effective way to fetch device capabilities information in Android Studio. By using this information, developers can optimize their application for different devices, improving the user experience and increasing the application's compatibility. Our example implementation demonstrates the practical application of this approach by adapting the application's UI to different device screen sizes.

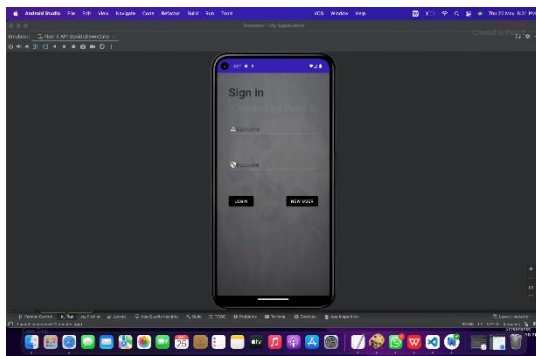


Figure 2: Login Page

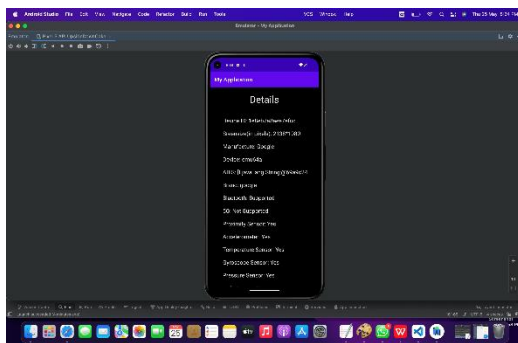


Figure 3: Fetched Details

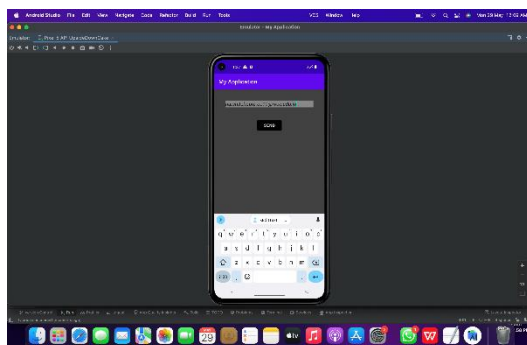


Figure 4: Email UI

Furthermore, our approach includes sending the fetched information via email by generating a text file in the external storage of the mobile device and then sending the saved text file as an attachment via email.

IV. CONCLUSION

In conclusion, the automation of manually fetched details offers numerous advantages to businesses and organizations. By leveraging technology to retrieve, validate, and update data, automation enhances efficiency, accuracy, and agility. With the potential to reduce errors, save time, and improve collaboration, automation is poised to revolutionize how organizations gather and utilize information, enabling them to stay ahead in an increasingly data-driven world.

V. FUTURE SCOPE

The future scope of automating the process of manually fetching mobile data is highly promising. Automation can streamline data collection from mobile devices, including IoT devices, and leverage artificial intelligence and machine learning techniques for intelligent analysis. It offers benefits such as process optimization, improved security and privacy, and the ability to extract meaningful insights efficiently. By automating repetitive tasks and reducing human error, organizations can focus on utilizing mobile data for innovation, decision-making, and personalized services.

REFERENCES

- [1] R. V. Golhar, P. A. Vyawahare, P. H. Borghare and A. Manusmare, "Design and implementation of android base mobile app for an institute," 2016 International Conference on Electrical, Electronics, and Optimization Techniques (ICEEOT), Chennai, India, 2016, pp. 3660-3663, doi: 10.1109/ICEEOT.2016.7755391.
- [2] L. Allison and M. M. Fuad, "Inter-App Communication between Android Apps Developed in App-Inventor and Android Studio," 2016 IEEE/ACM International Conference on Mobile Software Engineering and Systems (MOBILESoft), Austin, TX, USA, 2016, pp. 17-18, doi: 10.1145/2897073.2897117.
- [3] X. Meng, K. Qian, D. Lo, H. Shahriar, M. D. A. I. Talukder and P. Bhattacharya, "Secure Mobile IPC Software Development with Vulnerability Detectors in Android Studio," 2018 IEEE 42nd Annual Computer Software and Applications Conference (COMPSAC), Tokyo, Japan, 2018, pp. 829-830, doi: 10.1109/COMPSAC.2018.00141.
- [4] G. de los Reyes, S. Macwan, D. Chawla and C. Serban, "Securing the mobile enterprise with network-based security and cloud computing," 2012 35th IEEE Sarnoff Symposium, Newark, NJ, USA, 2012, pp. 1-5, doi: 10.1109/SARNOFF.2012.6222759.
- [5] Hang Dong, Chengze Li, Ting Li, Yuejin Du and Guoai Xu, "Research on the security model of mobile application," 2014 Communications Security Conference (CSC 2014), Beijing, 2014, pp. 1-5, doi: 10.1049/cp.2014.0751.
- [6] D. F. Priambodo, G. S. Ajie, H. A. Rahman, A. C. F. Nugraha, A. Rachmawati and M. R. Avianti, "Mobile Health Application Security Assesment Based on OWASP Top 10 Mobile Vulnerabilities," 2022 International Conference on Information Technology Systems and Innovation (ICITSI), Bandung, Indonesia, 2022, pp. 25-29, doi: 10.1109/ICITSI56531.2022.9970949.
- [7] D. Yang and I. Hwang, "Security Enhancement Methods for Mobile POS System," 2015 2nd ACM International Conference on Mobile Software Engineering and Systems, Florence, Italy, 2015, pp. 152-153, doi: 10.1109/MobileSoft.2015.40.
- [8] Z. Zhou, C. Sun, J. Lu and F. Lv, "Research and Implementation of Mobile Application Security Detection Combining Static and Dynamic," 2018 10th International Conference on Measuring Technology and Mechatronics Automation (ICMTMA), Changsha, China, 2018, pp. 243-247, doi: 10.1109/ICMTMA.2018.00065.
- [9] Suzhen Wang, Jianli Hu, Aizhen Liu and Jiazhen Wang, "Security Frame and Evaluation in Mobile Agent System," 2005 2nd Asia Pacific

Conference on Mobile Technology, Applications and Systems, Guangzhou, China, 2005, pp. 1-6, doi: 10.1109/MTAS.2005.207181.

Manipulation (SCAM), Adelaide, SA, Australia, 2020, pp. 132-142, doi: 10.1109/SCAM51674.2020.00020.

- [10] A. K. Jha and S. Nadi, "Annotation practices in Android apps," 2020 IEEE 20th International Working Conference on Source Code Analysis and