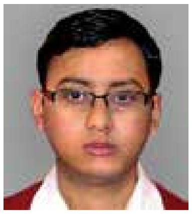
Employee Online App:
Single Point Resource for Employee Information

In near future, the 'Push Notification' service of EO App will be decoupled and provided as a component or a web service so that it can be resused by other applications.

RACHNA SRIVASTAVA
Sr.Technical Director & HoD
rachna_sri@nic.in

Y V RAMANA
Scientist-D
yvramana@nic.in

ANDREWS VARGHEESE
Scientist-C
andrews.varghese@nic.in



PANKAJ K. KHETWAL
Scientist-B
p.khetwal@nic.in

SUBRAMANIAN M
Scientist-B
ms.mani@nic.in

Edited by
MOHAN DAS VISWAM

Employees Online Mobile App (EO App) has been designed and developed by e-Office Project Division & NIC-DoP&T Division for providing various information such as about Senior officers' appointments & postings orders, as approved by ACC, "What is New" in D/o DoP&T, Holiday list and Directory Listing of all Ministries/ Departments (contact-us info) on a real time basis. Over 4900 IAS officers posted across the country and other Group A central services officers will be able to access details of their ER sheet, APAR, IPR, Postings apart from offer list, Officers at centre, training application status, domestic, foreign training details, Civil list (IAS), vacancy circulars, OMs & Orders etc.

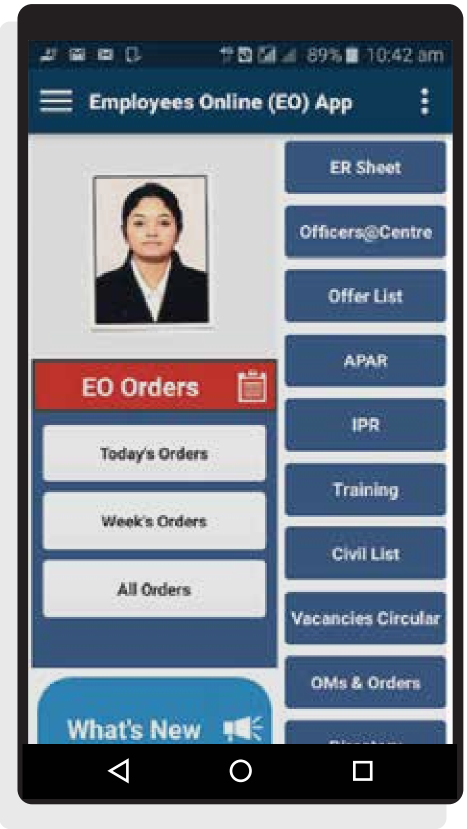
The EO App was launched on 28/10/2016 by Dr. Jitendra Singh, Hon'ble Minister of State of the Ministry of Development of North Eastern Region, Prime Minister Office, Personnel, Public Grievances and Pensions, Department of Atomic Energy and Department of Space, in the presence of Shri Bhaskar Khulbe, Secretary(PMO), Shri B. P. Sharma, Secretary(P), Shri Rajiv Kumar, EO & AS(DoP&T), Dr. Ajay Kumar, AS(MeitY), Smt. Neeta Verma DG(NIC), Shri G. K. Gaur, DDG(NIC), Smt. Rachna Srivastava, Sr.TD(NIC), Shri S. N. Sowpari, Sr.TD(NIC) and other senior officers from DoP&T.

The Hon'ble Minister appreciated the initiative and stated that "EO App is first of its kind to keep one updated about bureaucratic appointments, other orders and circulars of DoP&T and is a step towards transparency in eGovernance. It will also check on the number of repeated RTI Applications filed by citizens to seek Governance related information as most of the details will be put online for public on real time basis."

EO APP ARCHITECTURE

The data for EO App is being provided by several different backend application systems like DoP&T Personnel, DoP&T Trainings, Circulars and Orders database, eOffice SPARROW etc. Considering the fact, that some of these applications were already in use and all had different authentication mechanisms, it was decided that a centralized architecture (the Middle Layer) should be created to provide the following features.

- Unified authentication mechanism.
- Decoupling of mobile App from departmental web services from the data contributing Apps.



A sample UI Screen of EO App

Dr. Jitendra Singh, Hon'ble Minister of State (P) launching EO App. Others on the dais are Smt. Neeta Verma, DG(NIC), Shri Rajiv Kumar, EO, Shri Bhaskar Khulbe, Secretary, PMO and Shri B. P. Sharma, Secretary(DoPT)

- Single point of contact for better security.

EO APP SERVER (MIDDLE LAYER)

EO App middle layer had been designed as a RESTful API, using the latest technologies, in a secure and optimized manner, to address the three most important aspects of software development.

Security

- All the APIs are accessible only via encrypted channels, TLS.
- Once the user is logged in, using username/ password, each subsequent request will include the JSON Web Token, JWT, for authentication. The password is not stored in the mobile, thus it cannot be stolen.
- Developed as per OWASP guidelines.

Scalability

- The middle layer was developed as a stateless REST API, since there is no session it can be easily scaled, both vertically and horizontally.

Performance

- The middle layer has a caching mechanism, using Ehcache, to cache all the master data, thus reducing the turnaround time.
- The middle layer supports "Conditional GET" request, so that a resource is not returned unless it has been modified since the previous request. This feature will also reduce the data usage of the mobile.
- The response from the server is compressed using gzip.

BEST PRACTICES

As several backend systems were the source of data for EO mobile App, Web Services were written by individual application providers. Some of the best practices adopted while designing REST API for Backend Data Provider Web services were:

- The backend API should support "Conditional GET" request.
- Wherever the API is returning text response it is recommended to enable compression.
- It is recommended to expose the service URL via encrypted channel only, i.e. HTTPS.
- Maintain versioning of API.
- Use RESTful URLs.
- For text response, provide an option for the client to specify the response format, i.e. JSON or XML, both formats should be supported and JSON can be made as default format.
- Ensure proper usage of HTTP methods.
- Ensure proper usage of HTTP status code.
- Implement the concept of limit and offset when retrieving large set of data.
- Create proper documentation, documentation libraries like Swagger can be used to document your API.
- Application should be stateless.
- Use token based authentication, like JWT, instead of HTTP Basic authentication.
- For the data that serves multiple users, frequently accessed and doesn't change frequently, prefer caching those data.

EO APP (ANDROID & IOS)

Mobile applications are becoming a necessity nowadays for end user satisfaction and for reaching wider target audience with greater ease of access. NIC has developed EO Apps for iOS and Android mobile platforms. These Apps are developed as native applications, hence are more responsive, have better performance and enable instant notifications. These Apps follow the Flat Design and Material design principles for unified user experience on the Apple Sandbox and Android platforms respectively.

Security features of EO App

- The data exchanged between the mobile devices and the server is transferred through TLS channel only. This protects the application against eavesdropping and tampering with or forging the contents of

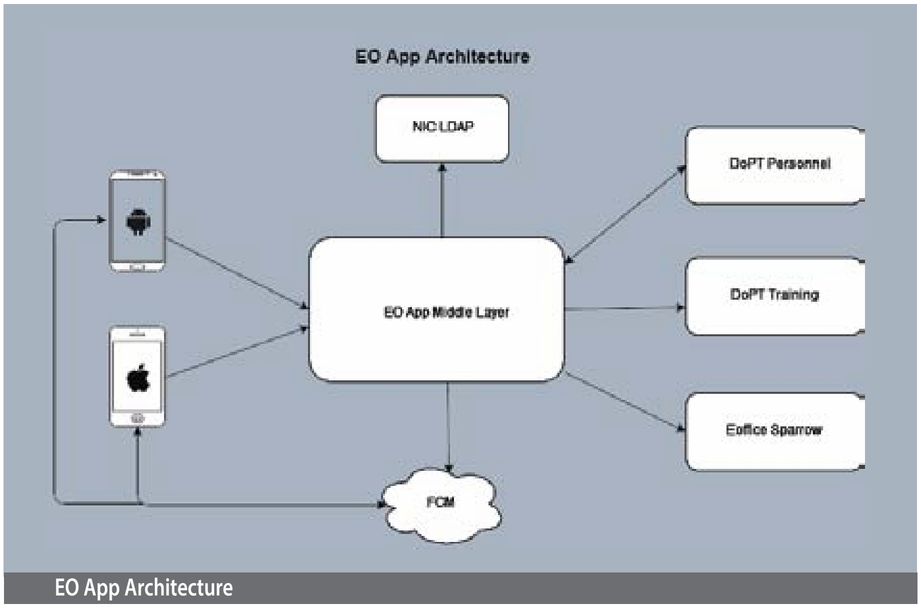
NIC officials and other dignitaries during the launch of EO App

the communication.

- These Apps are using TLS server certificate pinning mechanism to prevent man in the middle attacks.
- EO App doesn't store the password (LDAP password) of the user in the mobile device and instead JWT token is used for maintaining the session, which is valid for a specific amount of time. This increases the security of the App.
- The JWT Token, used for authentication, is stored in the mobile device in encrypted form only. This prevents the eavesdroppers or man in the middle attackers to misuse it.
- The EO iOS App conforms to the Apple's standard of "App Transport Security". It enforces applications to send network requests over a secure connection. If App Transport Security is enabled for an application, network requests are sent over HTTPS by default. Apple emphasizes its commitment to security and privacy by automatically enabling App Transport Security for applications built with XCode version 7 onwards.

Techniques & Technologies used in EO App

- **JSON Rest API:** The Restful APIs used in this App uses JSON as data exchange format. JSON (JavaScript Object Notation) is being used for data transfer between EO App and the middle layer. JSON is preferred to xml, as it is faster in parsing, lightweight and it has ability to represent complex data types in the form of objects (key-value pairs).
- **Data compression (gzip)**: To make data communication faster between EO App and middle layer the data is compressed in gzip format. It provides the effective way to save the consumption of network data and faster transfer of data. The data has been compressed to about 72% of the original size by this App after gzip enabling.

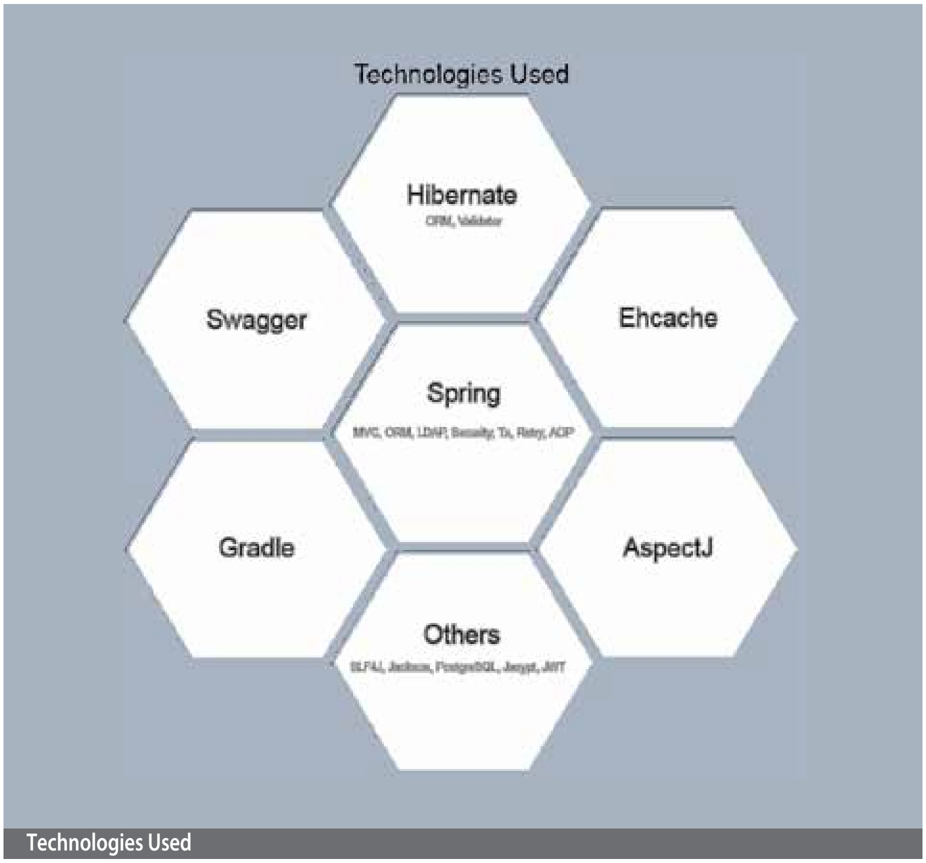


EO App Architecture

- **TLS certificate pinning**: EO Android App has used TLS server certificate pinning for additional security. It is one of the authentication process in which the client makes a connection to the server and the server responds with its SSL certificate. If the certificate is issued by a Certificate Authority that is trusted by the OS and is already pinned to the client, then the connection is allowed, otherwise the connection does not take place. This prevents Man-in-the-middle attacks, interception and other eavesdropping attacks. The Android App uses password protected bouncy castle key-store to trust and pin the certificate.

- **Push notifications (FCM)**: EO App uses Firebase Cloud Messaging (FCM) for push notifications. FCM is a new, improved version of the Google Cloud Messaging API and is known for being cross platform, so FCM makes a natural fit in the firebase suite of features designed for Android, iOS, and other mobile & web applications. It provides simpler client development and Firebase Notifications, a server less notifications solution with a web console that lets anyone send notifications to target specific audiences based on Firebase Analytic insights.

- EO App uses collapsible message, message may be replaced by a new message containing the same collapse key if it has yet to be delivered to the device. This App uses 2 collapse keys (one for


Technologies Used
Hibernate
ORM, Validator
Swagger
Ehcache
Spring
MVC, ORM, LDAP, Security, Tx, Retry, AOP
Gradle
AspectJ
Others
SLF4J, Jackson, PostgreSQL, Jasypt, JWT

Technologies Used

“What’s New” and another for EO Orders).

- EO App uses the topic messaging that allows to send a message to multiple devices that have opted in to a topic (like “What’s New” and EO Orders etc.) subscribed.
- EO App uses the data messages based notification, in which only client App is responsible for processing data messages. That is, the original message is not pushed from the FCM cloud to the device, instead a ping is sent. After receiving the ping, the mobile app polls the middle layer and gets new notifications and hence makes the app more secure.

The first version of EO Android App was released with Pull notification, whereby the mobile App pulls the server, at periodic interval, for any new notification. Since this process was not real time and battery draining, Push notification was used to replace the pull notification service. The push notification uses Firebase Cloud Messaging (FCM) in the latest version. Steps for enabling push notification:

- Registration of the project in Google. Google to acknowledge with providing sender ID and key.
- Whenever EO Mobile App is installed, a token will be shared to it by FCM.
- EO Mobile App will send the FCM token to the EO App Server.
- EO App Server will subscribe the newly registered tokens to a topic, in FCM.
- Whenever a new message is received from DoPT server, the EO App Server will construct a topic data message and send it to FCM.
- FCM will notify all the mobile Apps that are subscribed to that topic.

Features of EO Android App

- **Material design**: EO App has followed the material design. It resulted in the reduction of the overhead cost of designer and UX developers for multiple screen sized devices. As per material design, the App uses just two to three colours mostly to brand the entire App and a single prototype design can be used for web and mobile.
- **Network library (Volley)**: Volley library is used in this Android App for managing networking requests. It provides easier and faster networking requests handling by managing the processing and caching of network requests. It also saves development time by eliminating the need of writing the same network call/ cache code again and again. EO App sends multiple requests to the middle layer and these requests are being automatically scheduled by Volley. It provides transparent disk and memory caching using ET tags. It also allows a powerful cancellation request API, which allows cancellation of a single request or set blocks or scopes of requests to cancel.

Features of EO iOS App

- **Flat design**: EO iOS App has tried to follow the Apple flat design, wherever possible in the App in the given time frame. It resulted in the reduction of the overhead cost of designing for multiple screen sized devices like iPhone, iPad etc.
- **Security for jailbroken devices**: The EO iOS App checks if the App is being run on a jailbroken device or not while launching. If the device is jail broken, then the App will exit, otherwise it will work normally. This increases the security for the App, as it will not allow the App to proceed on any compromised devices for better security.

FUTURE ROADMAP

Currently, the ‘Push Notification’ service is tightly coupled with the EO App middle layer; in future, it will be decoupled and provided as a component or a web service, so that it can be reused by other NIC applications to avail ‘Push Notification’ service. The TLS pinning, currently implemented, is only Server certificate pinning; in future, the client side pinning also can be implemented for additional security.

For further information, please contact:
RACHNA SRIVASTAVA
Sr. Technical Director & HoD
eOffice Project Division
NIC, A-Block, CGO Complex, Lodhi Road
New Delhi-110 003
Email: rachna_sri@nic.in
Phone: 011-24364782