|  | **View** | **Talk stream** |
| --- | --- | --- |
|  |  | Voice Over:  Let me first take you through the automotive application before I begin to explain you how it was built using Oracle PaaS. John Dunbar is the regional sales manager for Acme Motors in San Francisco Bay area. He logs in to automotive application on his iPad using the same Oracle Sales Cloud credentials he uses in office. |
|  |  | Voice Over:  Once logged in, he is taken to the Home screen where he can see a list of actions he could perform based on his role as a manager.  Today, John wants to review how some of the key dealerships in his territories are doing. For doing that he can quickly browse the Territories option. |
|  |  | Voice Over:  Here the Automotive application is making a secure call to the Oracle Java Cloud and Sales Cloud for relevant information. The data transferred over the network is optimized for mobile consumption, encrypted, and relevant to the request. we will talk about one of the APIs when we see how Oracle PaaS is used to build this application |
|  |  | Voice Over:  Once John is in the Territories screen and can see a list of dealerships. Using thematic map to switch between different territories. |
|  |  | Voice Over:  The navigator on the left allows John to jump to various screens in the application. |
|  |  | Voice Over:  On Dealers screen, he gets to glance through the latest information on a particular dealer. All the key information about the dealer is concisely made available on this screen. He also gets to see the opportunities, revenue pipeline and past sales. Here the app calls the opportunity API to fetch the relevant information. |
|  |  | Voice Over:  On the Priority Assignment screen, he gets to see the latest priorities that are created by operations. John reviews the scope of a priority and based on his knowledge of the dealerships he choose to assign the priority to relevant dealerships. |
|  |  | Voice Over:  Not only John is interested in assigning priorities, but he also wants to ensure priorities are addressed by his Reps during their frequent visits to the dealerships.  A quick glance on the Urgency pie gives John an insight into what priorities he needs to follow up with his Reps. John calls his Reps to give importance to the ones that are urgent or approaching due date.  What’s cool here is that all of this was accessed from Oracle Sales Cloud using standard API calls running on Oracle Java Cloud and rendered in the iPad app. |
|  |  | Voice Over:  ‘Priorities by Sales Team’ graph gives him insight into the performance of his Sales Team during the current quarter. |
|  |  | Voice Over:  After John made the priority assignments, Lisa, who is a sales rep, can take benefit of Automotive app to stay productive while on the move.  She too can log in to automotive app on her iPad using the same Oracle Sales Cloud credentials she uses in office. |
|  |  | Voice Over:  Based on Lisa’s role as a Sales Rep, she sees a list of options that will assist her in staying productive.   |  | | --- | | As she is frequently visiting a number of accounts, having access to Dealer information is super helpful to her. | |  | | My Priorities screen gives Lisa access to all the priorities that have been assigned to her accounts. | |  | | The red badge on “My Priorities” icon located on the Home screen instantly shows her all the unread/newly assigned priorities that are on her list. | |  | |  | |
|  |  | Voice Over:  She can drill into any dealership and update her knowledge with the latest information. |
|  |  | Voice Over:  Now that you have seen the Automotive application, let me show you how you can use Oracle PaaS to build and deploy these applications quickly. To walk you through the steps, I will use one of the several APIs being used by the Automotive application. Let me start by logging on to my Oracle Cloud account. I can navigate to the relevant services from My Oracle Cloud Services dashboard.  This is a role-based dashboard that gives me a complete view of all my Oracle Cloud services and their key metrics in a single place  This ability to have a central view of all my subscriptions in one place is immensely useful and Oracle Cloud is uniquely positioned to offer this capability because we have a single integrated infrastructure for PaaS and SaaS, unlike the other providers. |
|  |  | Voice Over:  Let me get to my Automotive project in the Oracle Developer Cloud Service. As a developer, this is the center of my universe. From here I can do everything to run my projects. Track my tasks, collaborate with team members, and manage my development sprints and deployments. Oracle Developer Cloud Service supports a rich set of development tools that developers love, like GIT for source code repository, Hudson for continuous integration, Bugzilla for bug management and others.  As you can see, It supports multiple projects at the same time with appropriate access controls at various levels. |
|  |  | Voice Over:  Let me go to my Automotive project. From here I can use the key features like Tasks, Build, Review and Deploy to manage my development process for better quality code and developer productivity. |
|  |  | Voice Over:  Tasks allow me enter project requirements; do team assignments, set priorities and deadlines. The system automatically notifies the task details to the team members.  Let me create the task to fetch the dealer details for the specified product. |
|  |  | Voice Over:  For this task, I need to get to the code. Projects in Oracle Developer Cloud are attached to a git repository by default and can be accessed seamlessly using popular IDE’s such as JDeveloper and Eclipse. |
|  |  | Voice Over:  Before making changes to the code I want to clone it. I use the git extension in JDeveloper for that. |
|  |  | Voice Over:  I provide my Oracle Developer Cloud credentials along with the source git URL and select the branch with the source code I want to clone, |
|  |  | Voice Over:  As you can see, the source code gets automatically downloaded and the project loads up in JDeveloper.  Once I make changes to the source file, to complete the task assigned to me, I can document the changes I made as a part of comments section and commit the source code. |
|  |  | Voice Over:  At this point I am ready to push the changes into the Oracle Developer Cloud git repository and invite team members to review. |
|  |  | Voice Over:  I can create a new review, fill in the details like source/target branches, repository and description of the task and assign it to a reviewer. To show you how this works, let me assign myself as the reviewer. |
|  |  | Voice Over:  As you can see on the Review tab, I have a pending review. |
|  |  | Voice Over:  Here I can view the changes made in the target branch as compared to the source branch. Based on my review, I can Approve/Reject the code and add my comments. |
|  |  | Voice Over:  In this case I will approve the code and complete the review. |
|  |  | Voice Over:  Next I will start the build process by creating a build Job. |
|  |  | Voice Over:  Oracle Developer Cloud has wide range of build options to choose from like Maven and Ant. In this case I am using Ant. |
|  |  | Voice Over:  Once the Build Job is created, I can either schedule it immediately or for a future date. |
|  |  | Voice Over:  At the click of a button, the build artifacts are generated. I can check for the status in the build console.  Now I’m ready to deploy it to the Oracle Java Cloud. |
|  |  | Voice Over:  To deploy the application on Java cloud, I create a deployment profile and save it.  No other Cloud provides this kind of seamless development and deployment experience, with the ability to maintain a single code line and a single deployment process. |
|  |  | Voice Over:  As you can see from my Oracle Java Cloud console, the application is deployed and successfully.  This is how simple it was to develop and deploy the Automotive PaaS application using the Oracle Developer Cloaud and Java Cloud Services.  Hope this demonstrated to you how you can rapidly build your SaaS Extensions with Oracle PaaS. |