**Instabike Application**

# High-Level Design Document

**Version 1.0**

**31/07/2018**

**Table of Contents**

[**1.**](#_2et92p0) **Introduction 3**

[1.1](#_tyjcwt) Project Purpose 3

[**2.**](#_3dy6vkm) **System Architecture 4**

[2.1](#_4d34og8) System Overview 4

[**3.**](#_2s8eyo1) **Modules 6**

[3.1](#_3rdcrjn) Dealer Module (React Native) - Frontend 6

[3.2](#_26in1rg) Manufacturer Module (React.js) - Frontend 6

[3.3](#_lnxbz9) Financier Module (React.js) - Frontend 6

[3.4](#_35nkun2) Backend Module (Node.js) – Backend 6

[Backend structure 6](#_1ksv4uv)

[Models 6](#_44sinio)

[Services 8](#_z337ya)

## Introduction

CAMS (Computer Age Management Services) is a professionally managed, institutionally owned, SEBI regulated entity engaged as an IT enabled service solution partner to BFSI players in India. CAMS combines IT infrastructure, domain expertise and process maturity to serve both as a B2B and B2C solutions partner.

With 30+ years of experience and by leveraging its traditional strengths in customer service, transaction processing and volume record keeping, CAMS has the ability to offer Banks & NBFCs a suite of services covering document management, account set up, back-office processing and customer interaction management that help deliver superior service to their customers.

CAMS has envisaged the development of a B2C application that will bring a two-wheeler buyer, the automobile dealer and the auto financier on the same platform and enhancing the ease with which a customer can purchase a two-wheeler against a loan from a financier, benefitting all three stake-holders.

### Project Purpose

The purpose of the project is to make the lead created at the dealer end to seamlessly flow to the manufacturer and financier applications. The manufacturer gets high-level view on the leads generated across the different zones and the conversion ratio based on which he can access the performance of the dealership. Any lead opting for finance can be tracked in the financial application and status will get updated in the dealer application. The dealer application will be able to capture the lead from the time he/she walks in to the dealer showroom till the vehicle is registered.

## System Architecture

### 

### System Overview

The system consists of three applications namely: -

1. Dealer Application – Dealer application is mobile application designed specifically for Lenovo Tab 4 (8 inch). The Dealer application can be considered as a CRM application which has following key features: -
2. Lead creation
3. Lead status tracking
4. Lead Follow-up
5. Complete activity tracking of a lead
6. Provision to add multiple enquired vehicles for a lead
7. Inventory management (Not linked to DMS) along with test-ride vehicle count
8. Accessory Management
9. Booking of Test-ride – Booking, Canceling, Rescheduling
10. Booking of test-ride by uploading driving license
11. Managing the test-ride bookings
12. Vehicle details – Vehicle overview, Specifications, 360-degree view
13. Vehicle comparison – including competitor vehicle
14. Price break-down
15. Opting for Finance
16. Exchange bike value calculation
17. EMI Calculation
18. Option to add list of financiers by dealer and to choose one preferred financier
19. Financier representative can upload documents related to vehicle finance and do the OSV (Original Seen and Verified)
20. Manufacturer Application – Manufacturer application is a web-responsive application that can be viewed in desktop/laptop/mobile. The Manufacturer application has the following key features: -
21. Lead count across different categories
22. Top performing dealers in the current financial year
23. Sales Effectiveness across the dealer category
24. Sales Target Completion – 13-month detail
25. Vehicle details – Vehicle overview, Specifications, 360-degree view
26. Sales Performance graph for bikes and scooters
27. Sales Conversion analysis
28. Invoice analysis
29. Lost lead analysis
30. Dealer performance on MTD and YTD sales conversion – Zone-wise, State-wise, City-wise, Dealer-Category wise
31. Choosing preferred list of financiers
32. Financier Application - Financier application is a web-responsive application that can be viewed in desktop/laptop/mobile. The Financier application has the following key features: -
33. Lead count across different categories
34. Sales conversion based on dealership category
35. Complete activity tracking of a lead
36. Provision to add multiple enquired vehicles for a lead
37. Inventory management (Not linked to DMS) along with test-ride vehicle count
38. Accessory Management
39. Booking of Test-ride – Booking, Cancelling, Re-scheduling
40. Managing the test-ride bookings
41. Vehicle details – Vehicle overview, Specifications, 360-degree view
42. Vehicle comparison – including competitor vehicle
43. Price break-down
44. Opting for Finance
45. Exchange bike value calculation

## Modules

The different modules in Instabike application are: -

1. Dealer Module (React Native) - Frontend
2. Manufacturer Module (React.js) - Frontend
3. Financier Module (React.js) - Frontend
4. Backend Module (Node.js) – Backend

### Dealer Module (React Native) - Frontend

The Dealer Module is developed with following basic tool : [React](https://reactjs.org), [React-native](https://facebook.github.io/react-native/), [React- Navigation](https://reactnavigation.org), [React-Redux](https://medium.com/@imranhishaam/react-native-with-redux-for-beginners-6281959a2899), [Babel](https://babeljs.io) etc.This module has an organized file structure and the root directory is ‘app’ folder.

* app/index.js is the Application entry point.

Now let us have a detailed review of other directories in the app folder.

**Basic Structure:**

The primary folder sections of this modules are listed below,

* Layouts
* Components
* Redux
* Helpers
* Utils

**Layouts:**

This section consists of the screens of the applications. They are listed below with their corresponding purpose.

|  |  |  |
| --- | --- | --- |
| **File name** | **Purpose** | **Location** |
| Login | The screen which allows the dealer to login. | instabike-dealer/app/layouts/  Login |
| Onboarding | The files which has the onboarding steps for the dealer. | instabike-dealer/app/layouts/  Onboarding |
| ChooseAccessories | The screen which shows the list of accessories to choose. | instabike-dealer/app/layouts/  ChooseAccessories |
| ChooseFinancier | The screen which shows the list of financiers to choose. | instabike-dealer/app/layouts/  ChooseFinancier |
| CompareVehicles | The screen which adds various vehicles to compare. | instabike-dealer/app/layouts/  CompareVehicles |
| CreateNewLead | The files which has the onboarding steps to create new lead. | instabike-dealer/app/layouts/  CreateNewLead |
| Dashboard | The file which has the list of side menu screen files. | instabike-dealer/app/layouts/  Dashboard |
| DealershipDetails | The screens which gets the dealership details during onboarding process. | instabike-dealer/app/layouts/  DealershipDetails |
| EditTarget | The screens which allows to edit the targets of the team members. | instabike-dealer/app/layouts/  EditTarget |
| FilteredProductsScreen | The Screen which shows the list of vehicles based on filters applied during onboarding of new lead. | instabike-dealer/app/layouts/  FilteredProductsScreen |
| FinanceLeadOnboarding | The files which has the onboarding steps to create new finance lead. | instabike-dealer/app/layouts/  FinanceLeadOnboarding |
| FinancierListingScreen | The screen which has the list of financiers to choose for finance option to proceed. | instabike-dealer/app/layouts/  FinancierListingScreen |
| HomeDashboard | The screen which shows the MTD of lead details and target achieved. | instabike-dealer/app/layouts  /HomeDashboard |
| LeadDashboard | The screen which shows the leads for the last six months. | instabike-dealer/app/layouts/  LeadDashboard |
| LeadFollowUp | The screen which shows the list of leads for which the follow-up is scheduled. | instabike-dealer/app/layouts/  LeadFollowUp |
| LeadHistory | The screen which shows the activities of the lead. | instabike-dealer/app/layouts/  LeadHistory |
| MyDealership | The screen which shows the dealership details to edit. | instabike-dealer/app/layouts/  MyDealership |
| MyProducts | The screen which shows the list of vehicles available. | instabike-dealer/app/layouts  /MyProducts |
| MyTeam | The screen which shows the list of team members. | instabike-dealer/app/layouts/  MyTeam |
| NewLeadsOverview | The screen which shows the MTD of leads. | instabike-dealer/app/layouts/  NewLeadsOverview |
| PriceDetails | The screen which shows the price breakdown of the corresponding vehicle. | instabike-dealer/app/layouts/  PriceDetails |
| ProductDetail | The screen which has details description, features and view of the corresponding vehicle. | instabike-dealer/app/layouts/  ProductDetail |
| SearchLead | The screen which allow to search for the particular lead with name or mobile number. | instabike-dealer/app/layouts/  SearchLead |
| Splashscreen | The file contain the launch image of the application. | instabike-dealer/app/layouts/  Splashscreen |
| TeamMember | The screen which allows to create a new team member. | instabike-dealer/app/layouts/  TeamMember |
| TestRide | The screen which shows the list of rides available with category(eg. ongoing, scheduled, completed etc..). | instabike-dealer/app/layouts/  TestRide |
| TestRideTimings | The screen which allows to schedule the test rides. | instabike-dealer/app/layouts/  TestRideTimings |
| UpdateInventory | The screen which allows to update the stock details. | instabike-dealer/app/layouts/  UpdateInventory |

**Components:**

A component is a very basic element in react-native we can divide the large application into many small Components. This makes development fast and maintains the code very clear to understand.Some of the components used in this module are listed below.

|  |  |  |
| --- | --- | --- |
| **File name** | **Purpose** | **Location** |
| Button | The file which renders a button with required customisation. | instabike-dealer/app/  components/button |
| Cards | The file which renders a lead cards with customised information. | instabike-dealer/app/  components/cards |
| Charts | The file which renders a charts for the pictorial representation of the targets achieved. | instabike-dealer/app/  components/charts |
| Dropdown | The file which renders a lead dropdown with custom structure. | instabike-dealer/app/  components/dropdown |
| ExpandableTile | The file which renders a lead tile with custom structure. | instabike-dealer/app/  components/expandableTile |
| SideNav | The file which renders a lead side menu with available options. | instabike-dealer/app/  components/sideNav |
| Slider | The file which renders a view with animation. | instabike-dealer/app/  components/slider |
| Stepper | The file which display progress through a sequence of logical and numbered steps. | instabike-dealer/app/  components/Stepper |
| UserInput | The file which renders a lead Text Input with custom structure. | instabike-dealer/app/  components/userInput |

**Redux:**

“**Redux** is a predictable state container for JavaScript apps. It helps you write applications that behave consistently, run in different environments.” this means the entire data flow of the app is handle within a single container while persisting previous state.

Redux can be broken down into few sections while building the application which are listed below.

* Actions
* Reducers
* Store
* Middleware

**Actions:**

“Actions are payloads of information that send data from your application to your store. They are the *only* source of information for the store.” this means if any state change necessary the change required will be dispatched through the actions.

**Reducers:**

“Actions describe the fact that *something happened*, but don’t specify how the application’s state changes in response. This is the job of reducers.” when an action is dispatched for state change its the reducers duty to make the necessary changes to the state and return the new state of the application.

**Store:**

with help of reducers a store can be created which holds the entire state of the application it is recommended to use a single store for the entire application than having multiple stores which will violate the use of redux which only has a single store.

**Middleware:**

It provides a third-party extension point between dispatching an action, and the moment it reaches the reducer. People use Redux middleware for logging, crash reporting, talking to an asynchronous API, routing, and more.

**Helpers:**

create a js file full of many reusable functions and then import it in components and call it from there.

|  |  |  |
| --- | --- | --- |
| **File name** | **Purpose** | **Location** |
| AsyncStorage.js | It is a simple, unencrypted, asynchronous, persistent, key-value storage system that is global to the app. | instabike-dealer/app/helpers/  AsyncStorage.js |
| Platform.js | This file detects the platform in which the app is running. | instabike-dealer/app/helpers/  Platform.js |
| RestApiClient.js | This file is used to dynamically build the api url during run time. | instabike-dealer/app/helpers/  RestApiClient.js |

**Utils:**

Utils are the common functionalities that can be called and used across the applications. The list of Utils written as part is as follows: -

* Constants
* toaster
* Validation

|  |  |  |
| --- | --- | --- |
| F**ile name** | **Purpose** | **Location** |
| constants.js | This file contain the immutable values and objects which can be accessed globally. | instabike-dealer/app/utils/  constants.js |
| toaster.js | This file helps to present a popup info for user interface. | instabike-dealer/app/utils/  toaster.js |
| validations.js | This file is used to maintain the validation function which can be accessed globally. | instabike-dealer/app/utils/  validations.js |

* + 1. **Manufacturer Module (React.js) - Frontend**

The Manufacturer module boilerplate is developed with following tools: [React](https://reactjs.org/), [Redux](https://redux.js.org/), [React-router](https://reacttraining.com/react-router/core/api/Route/strict-bool), [Webpack](https://webpack.js.org/), [Babel](https://babeljs.io/) and [Redux-Saga](https://www.npmjs.com/package/redux-saga). This module has an organized file structure and the root directory is ‘app’ folder.For the project to build, these files must exist with exact filenames:

1.app/index.html is the page template

2.app/index.jsx is the JavaScript entry point.

Take a look at the application by clicking the [link](http://104.211.102.59:80). Now let us have a detailed review of other directories in the app folder.

**1. instabike-manufacturer/app/actions**

|  |  |  |
| --- | --- | --- |
| **S.no** | **File Name** | **Purpose** |
| 1 | BaseAction.js | This file acts like a interface. |
| 2 | Dashboard.js | This file has a list of actions relevant to dashboard entity. |
| 3 | Dealer.js | This file has a list of actions relevant to dealer entity. |
| 4 | Filter.js | This file has a list of actions for filter related functions. |
| 5 | FinancierDetail.js | This file has a list of actions relevant to financier entity. |
| 6 | FinancierList.js | This file has a list of actions related to financer CRUD. |
| 7 | Login.js | This file has a list of login actions. |
| 8 | ProductSales.js | This file has a list of actions relevant to product sales entity. |
| 9 | Profile.js | This file has a list of actions relevant to profile entity. |
| 10 | Vehicle.js | This file has a list of actions related to vehcile entity. |
| 11 | UserDetails.js | This file has a list of actions that are used for managing User Info. |
| 12 | ResetPassword.js | This file has actions used for user password management. |
| 13 | UserStatus.js | This file has a list of actions used to change the statuses of users. |
| 14 | VehicleInfo.js | This file has a list of actions that are used for managing Vehicle Info. |

**2.instabike-manufacturer/app/components**

|  |  |  |
| --- | --- | --- |
| **S.no** | **Filename** | **Purpose** |
| 1 | /AsyncImage/index.js | This file creates an image component. |
| 2 | /Card/index.js | This file creates a card component. |
| 3 | /EditableSpan/index.js | This file creates a form component. |
| 4 | /ErrorWrapper/index.js | This file serves as a wrapper for error component. |
| 5 | /Footer/index.jsx | This file returns the footer component. |
| 6 | /MenuBar/index.js | This file returns the menu component. |
| 7 | /Modal/index.jsx | This file renders a wrapper for modal. |
| 8 | /PageHeader/index.js | This file returns a customised header component. |
| 9 | /SideBar/index.jsx | This file returns a customised sidebar component. |
| 10 | /Table/Responsive/index.js | This file returns a customised responsive table wrapper. |
| 11 | /TextFieldValidator/index.jsx | This component displays error text. |
| 12 | /VehicleDetail/index.jsx | This file renders a vehicle detail page. |
| 13 | /VehicleDisplay/index.jsx | This component renders a responsive paginated table. |
| 14 | Loader.jsx | It returns a loader component. |

**3.instabike-manufacturer/app/constants**

This folder contains files in which each action constant is declared.

**4.instabike-manufacturer/app/containers**

|  |  |  |
| --- | --- | --- |
| **S.no** | **Filename** | **Purpose** |
| 1 | /Auth/index.js | This file renders login page or reset password screen based on condition. |
| 2 | /App/index.js | This is the parent page of the application. |
| 3 | /Dashboard/index.js | This page provides the analytics of dealer sales performance. |
| 4 | /Dealer/index.js | This page has functionalities for editing dealer information. |
| 5 | /Filter/index.js | It renders the filter page. |
| 6 | /FinancierDetail/index.js | This page displays the financier information. |
| 7 | /FinancierList/index.js | This page displays the list of financiers. |
| 8 | /HomePage/index.jsx | This component renders the child pages based on hierarchy. |
| 9 | /Login/index.js | This file renders the login page. |
| 10 | /NotFound/index.js | This file renders the error page. |
| 12 | /ProductDetails/index.jsx | This page shows the complete details of the vehicle. |
| 13 | /ProductList/index.js | It renders the complete list of products. |
| 14 | /ProductSales/index.js | This page shows the analytics of vehicle sales. |
| 15 | /Profile/index.js | This page is the profile management of the user. |
| 16 | /ResetPassword/index.js | Password resetting page. |
| 17 | /Routes/index.jsx | This page renders the routes based on hierarchy. |
| 18 | /VerifyEmail/index.js | Email verification page. |

**5.instabike-manufacturer/app/globals**

This folder has list of files wherein constants, error interceptors and local storage functions are written.

**6.instabike-manufacturer/app/reducers**

This folder has individual reducers for each page/module to maintain the application state.

**7.instabike-manufacturer/app/routes**

This routes folder has a file which defines route based on hierarchy.

**8.instabike-manufacturer/app/saga**

This files in this folder are required to handle the side effects in our redux app nice and simple. It achieves this by leveraging an ES6 feature called [Generators](https://developer.mozilla.org/en/docs/Web/JavaScript/Reference/Statements/function*), allowing us to write asynchronous code that looks synchronous, and is very easy to test.

**9.instabike-manufacturer/app/service**

The service files has the api endpoints which need to be hit when an action is triggered.

**10.instabike-manufacturer/app/store**

With help of reducers and redux saga, a store can be created which holds the entire state of the application. It is recommended to use a single store for the entire application than having multiple stores which will violate the use of redux which only has a single store.

**11.instabike-manufacturer/app/utils**

|  |  |  |
| --- | --- | --- |
| **S.no** | **Filename** | **Purpose** |
| 1 | financialYear.js | For price formatting throughout the application. |
| 2 | date.js | For all date formatting operations |
| 3 | email.js | For email validation operation. |
| 4 | url.js | For framing url endpoints dynamically. |
| 5 | image.js | For getting image urls of different resolutions. |
| 6 | numbers.js | For getting some mathematical results. |

* + 1. **Financier Module (React.js) - Frontend**

The Financier module boilerplate is developed with following tools: [React](https://reactjs.org/), [Redux](https://redux.js.org/), [React-router](https://reacttraining.com/react-router/core/api/Route/strict-bool), [Webpack](https://webpack.js.org/), [Babel](https://babeljs.io/) and [Redux-Saga](https://www.npmjs.com/package/redux-saga). This module has an organized file structure and the root directory is ‘app’ folder.For the project to build, these files must exist with exact filenames:

1.app/index.html is the page template

2.app/index.jsx is the JavaScript entry point.

Take a look at the application by clicking the [link](http://104.211.102.59:8080). Now let us have a detailed review of other directories in the app folder.

**1. instabike-financier/app/actions**

|  |  |  |
| --- | --- | --- |
| **S.no** | **File Name** | **Purpose** |
| 1 | appActions.js | This file has an action which will be dispatched on application initializing or reloading. |
| 2 | baseAction.js | This file acts like a interface. |
| 3 | Dealer.js | This file has a list of actions relevant to dealer entity. |
| 4 | globalActions.js | This file has a list of actions relevant to entire application. |
| 5 | Incentives.js | This file has a list of actions relevant to incentive entity. |
| 6 | leadAction.js | This file has a list of actions relevant to lead entity. |
| 7 | Performance.js | This file has a list of actions related to performance entity. |
| 8 | Profile.js | This file has a list of actions relevant to profile entity. |
| 9 | Target.js | This file has a list of actions relevant to target entity. |
| 10 | Team.js | This file has a list of actions related to team entity. |
| 11 | userLoginAction.js | This file has a list of actions that are used for User Authentication |
| 12 | userResetPasswordAction.js | This file has actions used for user password management. |
| 13 | userStatusAction.js | This file has a list of actions used to change the statuses of users. |

**2.instabike-financier/app/components**

|  |  |  |
| --- | --- | --- |
| **S.no** | **Filename** | **Purpose** |
| 1 | /Animated/index.jsx | This file contains multiple stateless functions primarily used for animation. |
| 2 | /buttons/ButtonWithLoader.js | This file contains a stateless function that will render a button. |
| 3 | /ChartWrapper/index.js | This file creates a chart component. |
| 4 | /ConfirmDialog/confirmAlert.js | This file renders a confirmation dialog box. |
| 5 | /DateRange/index.js | This file renders a date picker component. |
| 6 | /ErrorWrapper/index.js | This file serves as a wrapper for error component. |
| 7 | /InfiniteScrollHoc/index.js | This file has a component that serves as a wrapper for scrollable content. |
| 8 | /LoginWrapper/index.jsx | This file returns a wrapper for login and reset password screens. |
| 9 | /MonthRange/index.js | This file returns a customised month picker component. |
| 10 | /PageFooter/index.jsx | This file returns the footer component. |
| 11 | /PreviewFile/index.js | This file renders the file previewer component. |
| 12 | /Routes/index.jsx | This files renders a router component. |
| 13 | /SideBar/index.jsx | This file renders a side menu. |
| 14 | /Table/Responsive/index.js | This component renders a responsive paginated table. |
| 15 | /TextFieldValidator/index.js | This component displays error text. |
| 16 | DealershipCard.js | It returns a dealership card based on inputs. |
| 17 | DealershipSalesCard.js | It returns a customised dealership sales card. |
| 18 | Loader.jsx | The loader component is returned. |
| 19 | SalesExecutiveCard.js | It returns a customised sales executive card. |
| 20 | TeamCard.js | It returns a team member card. |

**3.instabike-financier/app/constants**

This folder contains files in which each action constant is declared.

**4.instabike-financier/app/containers**

|  |  |  |
| --- | --- | --- |
| **S.no** | **Filename** | **Purpose** |
| 1 | /Analytics/index.js | This is the analytics page of the instabike-financier application.It shows multiple analytics in the form of charts and graphs. |
| 2 | /App/index.js | This is the parent page of the application. |
| 3 | /Dashboard/index.js | This page has team member and sales executive management functionalities. |
| 4 | /EditTarget/index.js | This page has functionalities for editing targets for dealers. |
| 5 | /Header/index.jsx | This renders the customised header component |
| 6 | /Home/index.jsx | This is the home page of the financier application. |
| 7 | /Incentives/index.js | This file has the incentive management section. |
| 8 | /Lead/index.js | This file shows the leads of different categories and also the detailed view of each lead. |
| 9 | /Login/index.js | This file renders the login page. |
| 10 | /MyProfile/index.jsx | This file is the profile page of the logged in user.Profile management is available in this page. |
| 11 | /ResetEmail/index.js | Email verification page. |
| 12 | /ResetPassword/index.jsx | Password resetting page. |
| 13 | /Sidenav/index.js | Side menu is rendered based on the roles. |

**5.instabike-financier/app/globals**

This folder has list of files wherein constants, error interceptors and local storage functions are written.

**6.instabike-financier/app/reducers**

This folder has individual reducers for each page/module to maintain the application state.

**7.instabike-financier/app/routes**

This routes folder has a file which defines route based on hierarchy.

**8.instabike-financier/app/saga**

This files in this folder are required to handle the side effects in our redux app nice and simple. It achieves this by leveraging an ES6 feature called [Generators](https://developer.mozilla.org/en/docs/Web/JavaScript/Reference/Statements/function*), allowing us to write asynchronous code that looks synchronous, and is very easy to test.

**9.instabike-financier/app/service**

The service files has the api endpoints which need to be hit when an action is triggered.

**10.instabike-financier/app/store**

With help of reducers and redux saga, a store can be created which holds the entire state of the application. It is recommended to use a single store for the entire application than having multiple stores which will violate the use of redux which only has a single store.

**11.instabike-financier/app/utils**

|  |  |  |
| --- | --- | --- |
| **S.no** | **Filename** | **Purpose** |
| 1 | common.js | For price formatting throughout the application. |
| 2 | date.js | For all date formatting operations |
| 3 | email.js | For email validation operation. |
| 4 | url.js | For framing url endpoints dynamically. |

### Backend Module (Node.js) – Backend

All the frontend components from the different modules call the backend APIs which are written for all the three front-end modules. All the APIs developed follow REST standard and the complete list of APIs can be accesses through swagger from the [URL:-](about:blank)

http://104.211.102.59/explorer/

#### Backend structure

The backend (node.js) is basically divided into three main categories which will be namely: -

1. Models
2. Services
3. Utils

#### Models

Models are defined at database table level. The following list shows the complete list of models in the application developed so far: -

**Models**

|  |  |  |
| --- | --- | --- |
| **Model Name** | **Purpose** | **To access the list of APIs under this model** |
| Dealer | To handle all operations related to the dealer | <http://104.211.102.59/explorer/> (Under Dealer) |
| DealerAccessory | To handle all operations related to the accessories at dealer level | <http://104.211.102.59/explorer/> (Under DealerAccessory) |
| DealerCategory | To handle all operations related to the dealer categories | <http://104.211.102.59/explorer/> (Under DealerCategory) |
| DealerFinancier | To handle all operations related to financiers associated with dealers | <http://104.211.102.59/explorer/> (Under DealerFinancier |
| DealerInventory | To handle all operations related to the inventory of vehicles maintained at dealer level (No integration with DMS) | <http://104.211.102.59/explorer/> (Under DealerInventory) |
| DealerOffer | To handle all operations related to the offers given by the dealer | <http://104.211.102.59/explorer/> (Under DealerOffer) |
| DealerTarget | To handle all operations related to the target set at dealership/team/sales executive level | <http://104.211.102.59/explorer/> (Under DealerTarget) |
| DealerVehicle | To handle all operations related to the vehicles handled at dealer level | <http://104.211.102.59/explorer/> (Under DealerVehicle) |
| Zone | To handle all operations related to the Zones defined at manufacturer level | <http://104.211.102.59/explorer/> (Under Zones) |
| State | To handle all operations related to the States defined at manufacturer level | <http://104.211.102.59/explorer/> (Under States) |
| City | To handle all operations related to the Cities defined at manufacturer level | <http://104.211.102.59/explorer/> (Under Cities) |
| Document | To handle all operations related to document uploading | <http://104.211.102.59/explorer/> (Under Document) |
| ExchangeVehicle | To handle all operations related to exchange/trade-in vehicle | <http://104.211.102.59/explorer/> (Under ExchangeVehicle) |
| ExchangeVehicleLookup | To handle all operations related to database lookup for fetching the value for the exchange vehicle based on the model, year of manufacturer and kilometer driven | <http://104.211.102.59/explorer/> (Under ExchangeVehicleLookup) |
| Financier | To handle all operations related to financier | <http://104.211.102.59/explorer/> (Under Financier) |
| FinancierDealer | To handle all operations related to the association of financier and dealer | <http://104.211.102.59/explorer/> (Under FinancierDealer) |
| FinancierLead | To handle all operations related to financial lead (Financial lead is created when a lead at dealer end opts for financial option) | <http://104.211.102.59/explorer/> (Under FinancierLead) |
| FinancierTarget | To handle all operations related to the target set for financier sales executive and team | <http://104.211.102.59/explorer/> (Under FinancierTarget) |
| FinancierTeam | To handle all operations related to financial teams | <http://104.211.102.59/explorer/> (Under FinancierTeam) |
| Lead | To handle all operations related to lead created at dealer end | <http://104.211.102.59/explorer/> (Under Lead) |
| LeadActivity | To handle all operations related to activities tracked for a lead (Activity Log in lead detail) | <http://104.211.102.59/explorer/> (Under LeadActivity) |
| LeadDetail | To handle all operations related to lead detail (like vehicles enquired) | <http://104.211.102.59/explorer/> (Under LeadDetail) |
| LostReason | To handle all operations related to reasons for which a dealer lead may be lost | <http://104.211.102.59/explorer/> (Under LostReason) |
| Manufacturer | To handle all operations related to manufacturer | <http://104.211.102.59/explorer/> (Under Manufacturer) |
| ProformaInvoice | To handle all operations related to Performa Invoice | <http://104.211.102.59/explorer/> (Under ProformaInvoice) |
| TestRideVehicle | To handle all operations related to test ride vehicles | <http://104.211.102.59/explorer/> (Under TestRideVehicle) |
| Users | To handle all operations related to all the users in the system (like dealer roles, lead, financier roles, manufacturer roles etc) | <http://104.211.102.59/explorer/> (Under Users) |
| Vehicle | To handle all operations related to vehicle (like vehicle description, 360 degree etc) | <http://104.211.102.59/explorer/> (Under Vehicle) |
| VehicleFeature | To handle all operations related to vehicle features | <http://104.211.102.59/explorer/> (Under VehicleFeature) |
| VehiclePrice | To handle all operations related to price of the vehicle | <http://104.211.102.59/explorer/> (Under VehiclePrice) |

#### 

#### Services

The list of various services written as part of backend and their functionality is as follows: -

|  |  |  |
| --- | --- | --- |
| **Service Name** | **Purpose** | **To access the service in bitbucket** |
| base\_service.js | Common functionalities like login/logout, user creation process and extension of loop-back base method | Instabike-api/server/services/ base\_service.js |
| customer\_service.js | Sending OTP for customer (for test-ride booking) and for sending user login credentials | Instabike-api/server/services/ customer\_service.js |
| dealer\_inventory\_service.js | Maintenance of dealer inventory of vehicles with variant, colors, offers, test-ride vehicle count etc | Instabike-api/server/services/ dealer\_inventory\_service.js |
| dealer\_service.js | All CRUD operation related to dealer roles | Instabike-api/server/services/ dealer\_service.js |
| dealer\_target\_service.js | Operation related to setting target for different roles in dealer | Instabike-api/server/services/ dealer\_target\_service.js |
| document\_service.js | For all the document/photo upload service across dealer and finance applications | Instabike-api/server/services/ document\_service.js |
| financier\_lead\_service.js | CRUD operation for financial leads | Instabike-api/server/services/ financier\_lead\_service.js |
| financier\_service.js | All CRUD operation related to financier roles | Instabike-api/server/services/ financier\_service.js |
| financier\_target\_service.js | Operation related to setting target for different roles in financier | Instabike-api/server/services/ financier\_target\_service.js |
| financier\_team\_service.js | CRUD operations for financier team | Instabike-api/server/services/ financier\_team\_service.js |
| lead\_activity\_service.js | For tracking the lead activity (common for dealer lead and financier lead) | Instabike-api/server/services/ lead\_activity\_service.js |
| lead\_detail\_service.js | For tracking the enquired vehicles of a lead and its corresponding test-ride | Instabike-api/server/services/ lead\_detail\_service.js |
| lead\_service.js | CRUD operation for dealer leads | Instabike-api/server/services/ lead\_service.js |
| manufacturer\_service.js | All operation related to selection of financier and application of filters at zone-level, state-level, city-level | Instabike-api/server/services/ manufacturer\_service.js |
| proforma\_invoice\_service.js | All operations related to proforma invoice shown during price-breakdown | Instabike-api/server/services/ proforma\_invoice\_service.js |
| vehicle\_service.js | Operations related to showing vehicle specs, 360 degree, compare etc | Instabike-api/server/services/ vehicle\_service.js |

#### Utils

Utils are the common functionalities that can be called and used across the applications. The list of Utils written as part of backend and their functionality is as follows: -

|  |  |  |
| --- | --- | --- |
| **Util Name** | **Purpose** | **To access the service in bitbucket** |
| audit-utils.js | For tracking the audit of any operation | Instabike-api/server/utils/base\_service.js |
| date\_utils.js | For all date operations | Instabike-api/server/utils/date\_utils.js |
| emi\_utils.js | For EMI calculation | Instabike-api/server/utils/emi\_utils.js |
| external\_services\_providers.js | For integration with external service provider like SMS, Mail etc | Instabike-api/server/utils/external\_services\_providers.js |
| filter\_utils.js | For filter options (in manuafcturer, financier modules) | Instabike-api/server/utils/filter\_utils.js |
| image\_processor.js | For uploading of files and images | Instabike-api/server/utils/image\_processor.js |
| password\_utils.js | For hashing the passwords stored in DB (For security) | Instabike-api/server/utils/password\_utils.js |
| string\_utils.js | For all operations related to String | Instabike-api/server/utils/string\_utils.js |
| xlsx\_utils.js | For all operations related to importer | Instabike-api/server/utils/xlsx\_utils.js |