Oracle Banking Digital Experience

User Interface Build Guide
Release 18.1.0.0.0

Part No. E92727-01

January 2018

ORACLE®

User Interface Build Guide

January 2018

Oracle Financial Services Software Limited

Oracle Park

Off Western Express Highway

Goregaon (East)

Mumbai, Maharashtra 400 063

India

Worldwide Inquiries:

Phone: +91 22 6718 3000

Fax:+91 22 6718 3001
www.oracle.com/financialservices/

Table of Contents

1. Preface

1.1 Intended Audience

This document is intended for the following audience:

- Customers
- Partners

1.2 Documentation Accessibility

For information about Oracle's commitment to accessibility, visit the Oracle Accessibility Program website at http://www.oracle.com/pls/topic/lookup?ctx=accandid=docacc.

1.3 Access to Oracle Support

Oracle customers have access to electronic support through My Oracle Support. For information, visit

http://www.oracle.com/pls/topic/lookup?ctx=accandid=info or visit

http://www.oracle.com/pls/topic/lookup?ctx=accandid=trs if you are hearing impaired.

1.4 Structure

This manual is organized into the following categories:

Preface gives information on the intended audience. It also describes the overall structure of the User Manual.

The subsequent chapter covers the following:

- Task performed in the User Interface Build
- Steps to run the User Interface Build

1.5 Related Information Sources

For more information on Oracle Banking Digital Experience Release 18.1.0.0.0, refer to the following documents:

- Oracle Banking Digital Experience Licensing Guide
- Oracle Banking Digital Experience Installation Manuals

2. User Interface Build

The current GUI build is based on Grunt.

Grunt is a JavaScript Task Runner - an automation tool for performing repetitive tasks like minification, compilation, unit testing, linting etc.

The tasks performed during a typical GUI build are:

- Pre Build checks (For Some development rules)

- ESLint for the JS files.

- SCSS compilation to CSS

- CSS optimization

- HTML minification

- JS minification

- r.js Optimization to pack all the dependencies of a component into single file.

- Updating the files with configuration specified in the properties.json file.

- Generate integrity for all component files.

- Cache Busting for resources.

- Generate resource bundle for English language

Running UI Build:

Follow steps below to run UI Build:

- First make sure that Node js is installed on the machine and grunt is available in global scope.

- Run npm install command inside _build folder.

- Now run build.sh to run the build.

The properties.json file under _build folder is a configuration file having parameters to be set during build time:

1. resource_base_path: The base path from which the JS and HTML resources are to be fetched. It can be a relative path pointing to the same domain the page is running or fully qualified path to server. This can be configured to implement Domain Sharding. Default value of it set as **"/"**

2. image_base_path: The base path from which the image resources are to be fetched. It can also be a relative path pointing to the same domain the page is running on or a fully qualified path to different server on which images are hosted. Default value of it set as **"/images"**.

3. brand_base_path: This property is same as image_base_path used for those images which are brand dependent.

4. application_base_url: The application base resource path to which all the ajax calls are directed to. Default value of it set as "/digx".

5. application_base_url: This property hold the value of REST Application version. Default value of it set as "v1".

6. default_entity: This property hold the value of default entity of the application. This value used when entity not available to use. All request used this value

7. oam_base_url: This is the URL of OAM server for validating username and password.

8. obdx_base_path: To fund wallet using own bank account, need to redirect to bank portal.

9. secure_page: Set the protected page pathname, for example, "/pages/home.html".

10. public_page: Set the public page pathname, for example, "/index.html".

11. rtl_languages: In this property user can define the list of languages which supports RTL(Right To Left). This property supports array of languages. Default value of it is **['ar']**

12. localCurrency: This property stores the common currency used for implementation. This common currency is the default currency used in OBDX Application. All the origination request are using this property.

13. fb_sdk_url: This property stores the facebook SDK API URL. It is being used in OBDX Facebook component of social media module.

14. fb_api_key: This property stores the API key for the Facebook application. This API key is used for initializing and authenticating Facebook API.

15. linkedin_sdk_url: This property stores the Linkedin SDK API URL. It is being used in OBDX Linkedin component of social media module.

16. linkedin_api_key: This property stores the API key for the Linkedin app. This API key is used for initializing and authenticating Linkedin API.

17. google_map_sdk: This property is a unique key of google map to be integrated in OBDX application
User need to enable the used API for respective key in on developer.google.com after creating the key from same.

18. google_map_url: This URL is used to launch Google Maps with the location details passed as query parameters. Default value of it set as **http://maps.google.com/maps**

Home