

CXS Widget Collection 2.X

Certification Exam Study Guide



About this Document

This document provides references to help the trainees get prepared for the certification exam.

Version History

Version	Date	Description	Author
0.1	02-10-18	Initial	Varun Goel

Legal Notice

All documents are intended for Backbase and partners only, and should not be disclosed to any third party. No part of this publication may be reproduced electronically, by photocopy or print, without written permission from Backbase.

© Backbase, 2017. All rights reserved.

Table of Contents

Introduction	4
Module 1 - Extending Widgets	4
Backbase Retail Collection	4
bb-cli	4
Setting up your environment	5
Customizing existing items	5
Creating extensions for existing widgets	5
Module 2 - Developing Widgets	6
Understanding Design Goals	6
Creating custom widgets	6
Module 3 - Developing CXS Items	7
Start File	7
Datasources	7
Navigation Widget	7
Content Widget	7
Structured Content	7
Universal Collection	7
Create a Container	7
Targeting	8

Introduction

The purpose of this document is to provide you with a Study Guide towards your Certification Exam. This Study Guide contains all the references to the topics covered in the classroom training.

Module 1 - Extending Widgets

Classroom exercises for this module can be accessed [here](#).

The primary focus of this module comprises the following sections:

- Understanding Design Goals.
- Backbase Retail Collection.
- bb-cli.
- Setting up your environment.
- Creating extensions for existing widgets.
- Customizing existing Items.

Backbase Retail Collection

Most of the basic banking functionalities are provided out of the box by Backbase Widget Collection 2.x. Documentation on the collections:

- Widget Collection Retail Banking
 - [Functional Description](#)
 - [API](#)
- [Frontend Building Blocks API](#)

bb-cli

bb-cli is a command line tool provided and supported by Backbase to ease front-end development for the Backbase Platform. Check the [documentation](#) to learn more about the available commands.

To be able to learn about how to use generators to generate widget / extension etc., refer to this [documentation](#).

Setting up your environment

This [guide](#) will help you to setup your development environment.

Once you are ready with a development environment, you can also [configure your environment with bbconfig](#).

> *Tip: If you want to know more about bbconfig, refer to the [documentation](#).*

Note: The credentials to the Bitbucket were provided during the classroom training. If you face any issues in accessing the links, please send an email to academy@backbase.com

Customizing existing items

The [bb-customize](#) tool provided as a part of bb-cli is useful in order to be able to customize various available items e.g. theme, base URI for a service, etc.

Creating extensions for existing widgets

Extensions is a very important concept when reusing already implemented widgets. The documentation - [How to extend a widget](#) is useful in getting insight on this topic i.e. concepts like generating an extension, adding behavior using hooks / helpers, etc.

> *Refer: The major concepts covered in the classroom exercises included the following exercises.*

- *Customizing the Contact Manager extension*
- *Create an extension for the Contact Manager widget*
- *Making use of UI component with input group*
- *Style your widget through theme*
- *Filter the contacts list by implementing hooks*
- *Allow filtering by end-user through custom view helpers*
- *Translate the labels*

*More information about each of the concepts can be accessed on **community documentation**.*

Module 2 - Developing Widgets

Classroom exercises for this module can be accessed [here](#).

Understanding Design Goals

[Widget Blueprint](#) documentation can be used to gain knowledge on the design goals in Backbase Widget Architecture.

Optionally, you may also consider going through the detailed sections like Architecture Overview etc. to get detailed knowledge on the overall architecture.

Creating custom widgets

When there is a need to generate a new widget from scratch, following documentations are very useful:

- [How to Develop a new Widget](#)
- Adding a Custom Property
 - [How to add properties to a Widget](#)
 - [Control preferences by using viewHint](#)
- [Using bb-convert with RAML to generate Data Modules](#)
- [Events Handling - Publisher / Subscriber](#)

> Refer: The major concepts covered in the classroom exercises included the following exercises.

- *Create a custom ATM List widget using mock data*
- *Replace mock data with direct integration service*
- *Show bank details using inter-widget communication*
- *Creating preferences for a custom widget*

Module 3 - Developing CXS Items

Start File

This template defines the widget content. It is a file with the .html extension or with the .hbs extension in case you want to use Handlebars expressions. The convention for this file is that it is called index.html or index.hbs and located in the root of the item folder.

The following documentation provides more information about the [concept](#).

Datasources

There are various types of datasources available that can be used in CXS. More information about the concept can be accessed [here](#).

Navigation Widget

We generate one footer navigation in the classroom exercises. More information about navigation widget can be accessed on the community documentation.

Content Widget

ICE (In-Context Editing) Templates are used in Content Widgets to allow business managers to edit the content directly in CXP Manager. The following [documentation](#) provides more information on Content Widgets e.g. How does ICE work, How to customize ICE Template etc.

To be able to create your ICE Template, you will also need to know about Content Services i.e. CMIS object model, services etc. The documentation for Content Services can be accessed [here](#).

Structured Content

Structured Content is useful when you want to create reusable content and reuse it within different templates on various pages / portals. The documentation for Structured Content is available [here](#). This documentation provides more information on how to create a custom type, template etc.

Universal Collection

The containers banking collection contains a set of page containers and associated templates that can be used when designing portal pages (e.g. tab container, lightbox container, etc.) The list of available containers in the collection can be accessed [here](#).

Create a Container

If you want to develop your own container, the following [documentation](#) is useful.

Targeting

Creating client-side collector is possible to get user behavior data. More information about how to create a client-side collection can be accessed on the [link](#).