

UI5 User Guide

**How to develop responsive data-
centric client web applications**

Carsten Heinrigs

UI5 User Guide: How to develop responsive data-centric client web applications

Carsten Heinrigs

Copy Editor: John T. H. Dobson

Cover Designer: Rainer Roland

1. Edition

Copyright © 2018 Carsten Heinrigs

All rights reserved.

No part of this publication may be reproduced, stored in a retrieval system, or transmitted, in any form, or by any means, without the prior consent of the publisher.

The author/publisher has taken care in the preparation of this book, but makes no expressed or implied warranty of any kind and assumes no responsibility for errors or omissions. No liability is assumed for damages that may result from the use of information or code listings contained herein.

ISBN 978-3-9819591-1-6 (EPub)

ISBN 978-3-9819591-0-9 (Paperback)

ISBN 978-3-9819591-2-3 (PDF)

Table of Contents

Preface	xvii
1. Why this book?	xviii
2. Who this book is for	xviii
3. Structure of the book	xviii
4. Online resources for the book	xix
5. Notes on the e-book version	xix
About the Author	xxi
1. Introduction	1
1.1. Why UI5?	1
1.2. Getting started	2
1.3. Notes on UI5	3
1.3.1. Extended objects: common functionality	4
1.3.2. Dependency handling	5
1.3.3. OData	6
1.4. JSON and types	6
1.5. Some words about JavaScript	7
1.5.1. Synchronous Requests	10
1.6. Custom code	11
1.7. Conventions of example listings in the book	11
1.8. Development Environment	12
1.8.1. Testing	13
2. Building a basic responsive Web Application	15
2.1. Bootstrap the Core	16
2.2. Application manifest	17
2.3. Create a Component	18
2.4. Router	21
2.5. First views	23
2.6. Internationalization	24
2.7. Adding a LanguageSwitcher	26
2.8. Logging	31
2.9. More Routes and Pages	33
2.10. Adding Home and Back Buttons	36
2.11. Adding tiles to the entry page	38
2.11.1. Use a BaseController	40
2.11.2. Adding an entry controller	41
2.12. Adding a Footer	43
I. Basics of UI5 development	45
3. Views and Pages	49
3.1. Page layouts	49
3.1.1. Basic responsive Page	50
3.1.2. Fiori DynamicPage	52
3.1.3. Fiori SemanticPage	56
3.1.4. Multi-section pages: ObjectPageLayout	59
3.1.5. Administrative applications: ToolPage	63

3.2. A page just to present a Message	68
3.3. Summary	68
4. Navigation and Routing	69
4.1. Navigating without routes	69
4.2. Navigating with routes and targets	71
4.2.1. Pattern with optional parameter	73
4.2.2. Pattern with mandatory parameter	73
4.2.3. Pattern with multiple parameters	74
4.2.4. Pattern with additional parameters	74
4.2.5. Understanding target control parameters	75
4.2.6. General annotations to targets	78
4.3. Ways to navigate	78
4.4. Routing History	79
4.5. Summary	79
5. User Permissions	81
5.1. Requesting the current user profile data	81
5.2. Properties to implement user permissions	86
5.3. Summary	86
6. Events and Messages	87
6.1. Events	88
6.1.1. User Interface Events	88
6.1.2. Publish/Subscribe Events	89
6.1.3. Routing events	91
6.2. Messages to the user	91
6.2.1. Quick Notification: MessageToast	92
6.2.2. Ask the user: MessageBox	92
6.2.3. Present a list of messages to the user: MessageStrip	97
6.2.4. Navigate messages with details: MessagePopover	97
6.3. Value state messages	101
6.4. Summary	101
7. Models	103
7.1. Where do we use models?	103
7.2. Which model implementation to use?	103
7.3. Binding modes	104
7.3.1. OneWay or TwoWay?	104
7.4. Loading data	105
7.5. Setting and getting model data	106
7.6. Setting and getting models	107
7.7. Binding context and path	108
7.7.1. Complex binding syntax	108
7.7.2. Formatting values for display	109
7.7.3. Using a factory function	111
7.8. Simple model example	112
7.9. Scope and design of models	113
7.10. In client memory vs. constant reloading	114
7.11. Limits to Models	114
7.12. Summary	115

8. Lists and Tables	117
8.1. Common features and behavior	117
8.1.1. ListBase	117
8.1.2. ListItemBase	118
8.2. A List example	118
8.3. Controller functions handling events	120
8.4. A Table example	120
8.5. Sorting	123
8.5.1. Grouping	124
8.6. Filtering and Searching	124
8.6.1. Facet Filter	126
8.7. Summary	129
9. Forms and Input Validation	131
9.1. Form	131
9.2. SimpleForm	132
9.3. InputBase	133
9.4. Selected Form Controls	133
9.4.1. Label	133
9.4.2. Input	133
9.4.3. MaskInput	135
9.4.4. TextArea	135
9.4.5. ComboBox and MultiComboBox	136
9.4.6. Date/TimePicker	137
9.4.7. CheckBox and Switch	138
9.4.8. Select	139
9.4.9. RadioButton	139
9.5. Validation UI5 Style	140
9.6. Summary	141
II. Providing and using metadata for domain entities	143
10. Entity specifications: Mapping	147
10.1. Describe an entity	147
10.2. Describe Requests	150
10.3. Access the mappings	152
10.4. Summary	156
11. Request Handling	157
11.1. Request URL returning JSON	157
11.2. Running requests configured in the mapping	159
11.2.1. Process request parameters	159
11.2.2. Encode the parameter values	160
11.2.3. Prepare environments	161
11.2.4. Send and process the request	161
11.3. Testing the new functionality	162
11.3.1. Setup for testing	163
11.3.2. Specifying our expectations	165
11.4. Summary	170
12. Custom data objects	171
12.1. Provide common functions for domain objects	172

12.1.1. Collection	172
12.1.2. Entity	177
12.2. Adding data to the collections	179
12.3. Unit testing the entity collection and item objects	181
12.4. Handle referenced data	184
12.5. Further processing of incoming data	194
12.6. Adding logic to the entity object	195
12.7. Summary	198
13. Validation	199
13.1. Extending the mapping	200
13.1.1. Validation terms and related validation functions	201
13.2. Custom functions for validation	205
13.3. Summary	211
14. Form Control Generation	213
14.1. Common properties	216
14.2. Construct Input	218
14.3. Elements with option lists	219
14.4. Date and Time	221
14.5. Boolean elements	224
14.6. Add form element to a Form or SimpleForm	224
14.7. Summary	226
III. Building an Example Application	227
15. Beginning application development	231
15.1. Manifest	232
15.2. Common issues	234
15.2.1. Non-disruptive message for the receiving view	234
15.2.2. Request failure events	235
15.3. First view: A List of Orders	237
15.3.1. Preparing the item template	239
15.3.2. Construct the list and query orders data	242
15.3.3. Adding a page sub-header toolbar	244
15.3.4. Query Form	246
15.4. Summary	252
16. Edit Order page	253
16.1. Order details navigation	253
16.2. Order detail views: common fragments and controller	256
16.3. Order overview	258
16.3.1. Common Model	263
16.3.2. Indicate pending changes	264
16.3.3. Ship to billing address?	265
16.4. Edit Order lines	267
16.4.1. Changing the line quantity	270
16.4.2. Removing an item	270
16.4.3. Adding an item	271
16.5. Save and Cancel, Back and Home	274
16.5.1. Handle pending changes	277
16.6. Summary	280

17. Edit Address view	281
17.1. Search address dialog	286
17.2. Validating and saving the address	290
17.3. Reverting changes and resetting the view	292
17.4. Summary	293
18. Write user interface tests	295
18.1. One Page Acceptance testing	295
18.2. Matchers and Actions	300
18.3. Summary	302
19. Automate development tasks with Grunt	303
19.1. Installing Grunt and Extensions	303
19.2. Basic Grunt Setup	304
19.3. Linting	306
19.4. Building the distribution package	307
19.5. Documenting the code and generating HTML	310
19.5.1. Controllers	310
19.5.2. Custom Namespaces	310
19.5.3. Classes	311
19.5.4. Interfaces	311
19.5.5. Generating the documentation	312
19.6. Summary	313
A. More interesting controls	315
Index	317
Glossary	321

List of Figures

2.1. Basic structure of a UI5 Application	15
2.2. Browser console: Access Component	21
2.3. Entry page with custom header	36
2.4. SplitApp Header	36
2.5. Help view with navigation buttons	38
3.1. Sketch of sap.m.Page	50
3.2. Sketch of sap.f.DynamicPage	54
3.3. Sketch of sap.f.SemanticPage	57
3.4. Sketch of sap.uxap.ObjectPageLayout	60
3.5. Sketch of sap.tnt.ToolPage	64
9.1. Screenshot of UI5 Inspector ResponsiveGridLayout properties	132
11.1. Request Flow	157
12.1. Scheme of MVC Pattern	172
12.2. Browser console: listing available functions of the oum.do.products namespace	176
12.3. Browser console: listing available functions of the oum.do.Order object	179
12.4. Domain Data Handler Objects	187
14.1. Overview of important form field controls	214

Code Listings

2.1. sap-ui-bootstrap	16
2.2. Descriptor Application Manifest	17
2.3. Create a Component	18
2.4. Initialize the Component	20
2.5. Routing section of the manifest	21
2.6. Initial custom Router	22
2.7. XML view with sap.m.App	23
2.8. Basic JS View returning Page	23
2.9. Loading the i18n Resource	25
2.10. Page with title	25
2.11. Custom configuration	26
2.12. Initialize custom namespace	27
2.13. Custom namespace for configuration issues	28
2.14. LanguageSwitcher fragment	29
2.15. Add LanguageSwitcher fragment to Page	30
2.16. Attach handler to the change event of the language Select control	31
2.17. Namespace oui5lib.logger	32
2.18. Configuring target for non-existing routes	33
2.19. No such route Page	33
2.20. XML view with SplitApp	34
2.21. SplitApp target entries	34
2.22. Router function to navigate to a named target	35
2.23. HelpButton fragment	36
2.24. Router function to navigate back	37
2.25. Setup custom utility namespace	37
2.26. BackButton fragment	37
2.27. JavaScript view creating Page with custom header	38
2.28. Add Tiles to Page	39
2.29. A custom BaseController	40
2.30. The entry controller extends the BaseController	41
2.31. XML Toolbar Fragment showing basic application information	43
3.1. JSView example with controller	49
3.2. Define landmark roles and labels	51
3.3. Construct a sap.m.Page	51
3.4. Construct a sap.f.DynamicPageTitle	54
3.5. Controller function to toggle footer visibility	55
3.6. Construct a sap.f.DynamicPageHeader	56
3.7. Construct a basic sap.f.DynamicPage	56
3.8. Construct a basic sap.f.semantic.SemanticPage	58
3.9. Construct a sap.uxap.ObjectPageHeader	60
3.10. Construct a sap.uxap.ObjectPageDynamicHeaderTitle	61
3.11. Construct a sap.uxap.ObjectPageSection	62
3.12. Construct a sap.uxap.ObjectPageLayout	63
3.13. Construct a sap.tnt.ToolHeader	65

3.14. ToolPage controller function: toggle side navigation	66
3.15. Construct a sap.tnt.SideNavigation	67
3.16. Construct a sap.tnt.ToolPage	68
4.1. Load and add view to the App pages aggregation	69
4.2. Load and add view to the SplitApp detailPages aggregation	70
4.3. XML view with sap.m.App using navigate event	70
4.4. App controller with navigate event handler function	71
4.5. Controller implementing pattern-matched handler	72
5.1. Namespace oui5lib.currentuser	82
5.2. Check view permission	84
5.3. Protect objects: deepFreeze function	85
6.1. Construct a sap.m.ComboBox	89
6.2. ComboBox event handler functions	89
6.3. Subscribe to an event	90
6.4. Publish through the Core EventBus	91
6.5. Publish through the Component EventBus	91
6.6. Function to show a MessageToast	92
6.7. Use MessageBox	92
6.8. Utility function to get I18n resource property	94
6.9. Function to show an error message	94
6.10. Function to ask the user to confirm discarding unsaved changes	95
6.11. Function to ask the user to confirm deleting a record	96
6.12. Function to show a list of messages: MessageStrip	97
6.13. Construct a sap.m.MessagePopover	98
6.14. Construct Dialog for the MessageView	100
7.1. Change default binding mode of Model	104
7.2. Use the JSONModel to load data	105
7.3. Set data to a model	106
7.4. Get model data values by path and context	106
7.5. Set model data values by path and context	107
7.6. Using type Float to format displayed value	110
7.7. Use path/formatter notation	110
7.8. Use parts/formatter notation	111
7.9. Construct a sap.m.StandardListItem	111
7.10. Select control for the sort options	112
7.11. Checkbox control for the sort order	113
7.12. Example controller function getting model properties	113
8.1. Construct an ObjectListItem	118
8.2. Construct a sap.m.List	119
8.3. How to get the item data from a ListBase event	120
8.4. Controller function to get ObjectListItem properties	120
8.5. Construct a sap.m.Table	121
8.6. Construct a sap.m.ColumnListItem	122
8.7. Binding items with a Sorter	123
8.8. How to sort items	123
8.9. Binding model with a filter	125
8.10. Construct a sap.m.SearchField	125

8.11. Controller function to filter list items	125
8.12. Construct a sap.m.FacetFilter	127
8.13. Controller functions for the FacetFilter	128
9.1. Construct a sap.ui.layout.form.Form	131
9.2. Add FormElement to FormContainer	132
9.3. Construct a sap.ui.layout.form.SimpleForm	132
9.4. Construct a sap.m.Label	133
9.5. Construct an sap.m.Input	134
9.6. Construct a sap.m.MaskInput	135
9.7. Construct a sap.m.TextArea	135
9.8. Construct a sap.m.ComboBox	136
9.9. Binding data to an aggregation	136
9.10. Construct a sap.m.DatePicker	137
9.11. Construct a sap.m.DateRangeSelection	138
9.12. Construct a sap.m.CheckBox	138
9.13. Construct a sap.m.Switch	138
9.14. Construct a sap.m.Select	139
9.15. Construct a sap.m.RadioButtonGroup	139
9.16. Basic custom type: Email	140
10.1. General outline of a mapping	147
10.2. Example order	148
10.3. Example mapping: specify order requests	151
10.4. Namespace oui5lib.mapping	152
10.5. Process entity attribute and request parameter specifications	153
10.6. Set defaults for request configurations	154
10.7. Load and process entity mapping	154
11.1. Namespace oui5lib.request	157
11.2. Process request parameters	159
11.3. Get the request URL depending upon the configured environment	161
11.4. Function to send a request defined in the mapping	162
11.5. Jasmine Standalone SpecRunner.html	164
11.6. Describe oui5lib.request expectations	166
11.7. Jasmine specification of the mapping request Loader	168
11.8. Namespace oum.do.Loader: loading domain data	169
12.1. Utility function to extend an object	175
12.2. A basic orders collection object	175
12.3. A statuses collection object	176
12.4. A basic order entity object	178
12.5. Test the loading of orders	180
12.6. Loader function to handle the successful response	180
12.7. Specifications for orders collection object	181
12.8. Specifications for the order entity object	182
12.9. Additional Order Specifications	184
12.10. Function to add address names to orders	186
12.11. RefsHandler function to register incoming entities	188
12.12. Process incoming order data	194
12.13. Prepare and return the order lines	195

12.14. Get an order line by the product ID	196
12.15. Update the quantity of an order line and recalculate totals	197
12.16. Remove an order line by the product ID and recalculate order total	197
12.17. Add an order line with the given product ID	198
13.1. Use UI5 type to validate a value	200
13.2. Namespace oui5lib.validation	205
13.3. Function to get the value of the entity attribute	207
13.4. Specify expectations for the function to validate data	208
13.5. Validate data against the entity attribute specifications	210
14.1. Set common control properties	216
14.2. Namespace oui5lib.ui	217
14.3. FormController record changed status	218
14.4. FormController: Construct Input	218
14.5. FormController: Construct ComboBox	221
14.6. FormController: Construct DatePicker	223
14.7. FormController: Construct Label	224
14.8. Add Control to Form or SimpleForm	225
15.1. Fragment returning a Bar with Back and Home navigation	237
15.2. Initial orders view	238
15.3. Initial orders controller	238
15.4. Function to get date/time pattern	240
15.5. Query Orders	242
15.6. Controller subscribing to 'loading ready' events	243
15.7. Set Model to Status List	247
16.1. Master view: navigation to order details	253
16.2. Order details navigation controller	254
16.3. Controller function to navigate to the order detail routes	256
16.4. Fragment returning a Bar with a Save and Cancel button	257
16.5. Initial order view	257
16.6. Common controller for order detail controllers to extend	258
16.7. Initial order controller	259
16.8. Form showing basic order information	262
16.9. Set order model for the form	263
16.10. Order items table	268
16.11. SelectProductDialog fragment	272
16.12. Open message box with validation errors	275
16.13. Reset order view	276
16.14. Controller function to save an order	277
16.15. HomeButton fragment checking for unsaved changes	278
16.16. OrderBaseController function to handle unsaved changes	278
17.1. Initial order address view	281
17.2. Address form constructed with the FormController	283
17.3. Order address controller	284
17.4. Handle validation errors	290
17.5. Reset value states of SimpleForm or Form	293
18.1. Orders page	295
19.1. Install package grunt and add it as development dependency	303

19.2. Configure the eslint task in the Gruntfile	307
--	-----