

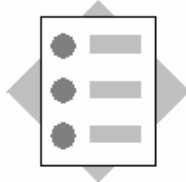
UI Elements

Exercise



Chapter: UI Elements

Theme: Dictionary, Table



At the end of this Exercise, you are able to:

- Define a dictionary structure and reuse it from a Web Dynpro application.
- Define standard table functionality.

1 Development Objectives

Defining Dictionary Structures

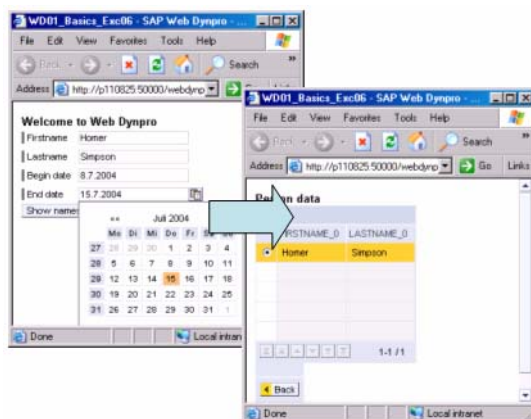
In the Web Dynpro perspective, you can comfortably create simple data types and structures, immediately deploy them to the SAP Web Application Server, and then use them in your Web Dynpro application straight away without having to switch to the Java Dictionary even once.

Structures created in the Java Dictionary can be reused in different Web Dynpro contexts.

Tables

Table UI elements allow authors to arrange data, preformatted text, and so on into rows and columns of cells.

2 Result



In this exercise, you will define a table and insert some data into it.

Optional part I:

You will learn, how to display details of a selected row.

Optional part II:

You will learn, how to delete an entry of a table.

Optional part III:

You will learn, how to sort the table content.

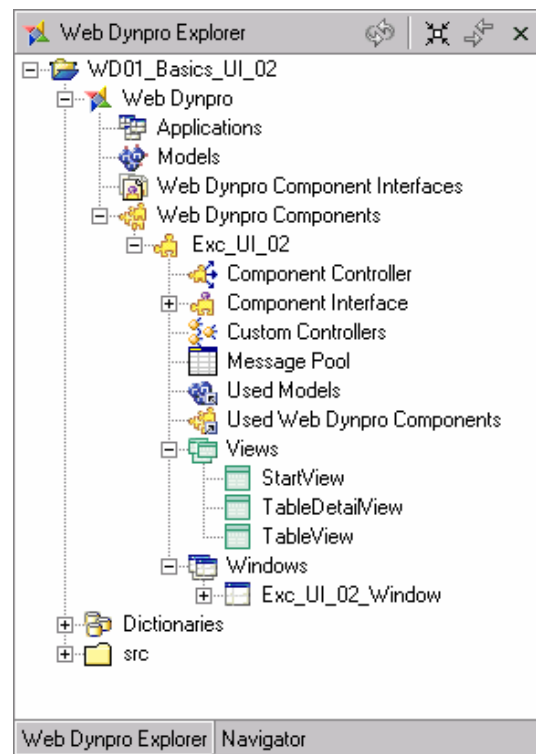
3 Prerequisites

You have launched the SAP NetWeaver Developer Studio.

You have selected the Web Dynpro perspective.

You have opened the *WD01_Basics_UI_02* project.

The structure of this project is currently displayed in the *Web Dynpro Explorer*.



For your convenience, you can start developing with a predefined view.

Expand the node *Exc_UI_02* and add the new functionality to the existing view *TableView*.

The graphic on the left shows the predefined project structure of this exercise with the predefined views:

StartView

Completely predefined.

TableView

Here you have to insert a table displaying person information.

TableDetailView

This is only needed for the optional exercise.

4 Overview: Developing

4-1 Define data transport from *StartView* to *TableView*.

4-1-1 Define four value attributes for the context of view *StartView*: *ctx_firstName*, *ctx_lastName*, *ctx_dateBegin* and *ctx_dateEnd*. Bind the input field UI elements to the context value attributes.

4-1-2 Define four value attributes for the context of the component controller: *cmpCtx_firstName*, *cmpCtx_lastName*, *cmpCtx_dateBegin* and *cmpCtx_dateEnd*.

4-1-3 Define the mapping between the context of the view *StartView* and the context of the component controller. Map the value attributes between the two contexts.
In addition, define the mapping between the context of the view *TableView* and the context of the component controller without mapping any context attributes.

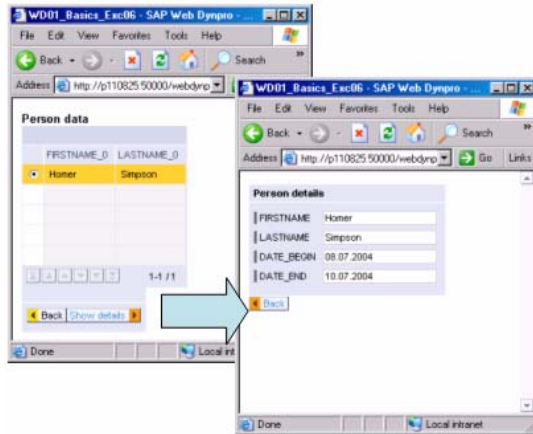
- 4-2 In the Layout of view *TableView*, define a table UI element. Bind this UI element to the views context.
- 4-2-1 Define the table UI element.
- 4-2-2 Define a Dictionary structure type, having the Name *Person* and the components *firstname*, *lastname* (both of type string, not null), *dateBegin* and *dateEnd* (both of type date).
- 4-2-3 Create the context for view *TableView* with structure binding to the structure *Person*.
- 4-2-4 Define the table binding between the table UI element and the context:
The elements *firstname* and *lastname* of the value node *Person* shall be bound to columns of the table UI element.
- 4-2-5 Insert the values of the input fields of *StartView* into the context of the view *TableView*, so they are displayed by the table UI element.
In order to realize this task, you have to implement the handler method *onPlugfromStartView(...)*:
- Define the local variables *firstname*, *lastname* (both type String), *date_begin* and *date_end* (both type Date) and get the corresponding values from the component controller context.
 - Define a local instance of the context element *Person* (type *IPersonElement*).
 - Set the elements of this instance to the values of the local variables.
 - Add this local instance to the context node *Person*.
- 4-3 Create the Web Dynpro application *WD01_Basics_UI_02*.

5 Overview: Building, Deploying, and Running

Deploy and run the Web Dynpro application.

6 Optional: Additional table features.

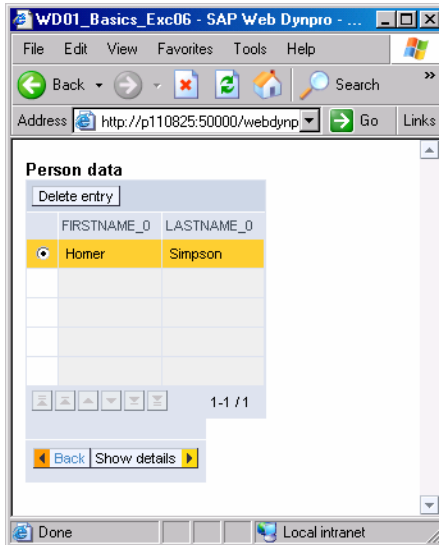
6-1 Show Details of a selected table row in a separate view.



In this exercise, you add an additional view *TableDetailView* to the component. When you select a table row and choose the Details button, the *TableDetailView* appears and the details of the selected person are displayed.

- 6-1-1 Define data transport from *TableView* to *TableDetailView*.
 - 6-1-1-1 Define the component context value node *Person* for the component controller.
 - 6-1-1-2 Define the view context value node *Person* for the view *TableDetailView*.
 - 6-1-1-3 Define the context mapping between the view context of *TableView* and the component context. Define the mapping between the value nodes of the two contexts.
 - 6-1-1-4 Define the context mapping between the context of the view *TableDetailView* and the context of the component controller. Define the mapping between the value nodes of the two contexts.
- 6-1-2 Add input fields for all context elements to the detail view *TableDetailView*. Apply the template *Form* to define input fields and labels.
- 6-1-3 Define the navigation between the views *TableView* and *TableDetailView*.
- 6-1-4 Create the Web Dynpro application *WD01_Basics_UI_02*.
- 6-1-5 Build, deploy and run the application.

6-2 Delete a selected table row.



In this exercise, you will add a button *Delete* to the toolbar of the table UI element.

If you select a table row and choose the button *Delete*, the table row will be deleted.

- 6-2-1 Define a toolbar for the table in the view *TableView*. Add a *toolbarButton* to the toolbar.
- 6-2-2 Define an action for the toolbar button and bind the *toolbarButton* to this action.
- 6-2-3 Implement the event handler bound to the action *Delete* in order to delete a person's entry:
 - The Method *wdContext.nodePerson().getLeadSelection()* can be used to find out the index of the selected element.
 - The Method *wdContext.nodePerson().getElementAt(index)* can be used to get back the reference on the element with index *index*.
 - The Method *wdContext.nodePerson().removeElement(object)* can be used to delete a given element *object* from the list of persons.
- 6-2-4 Create the Web Dynpro application *WD01_Basics_UI_02*
- 6-2-5 Build, deploy and run the application

6-3 Calculated attributes

Table data from the screenshot:

FIRSTNAME	LASTNAME	DATE_BEGIN	DATE_END	DURATION
Homer	Simpson	01.12.2004	01.12.2004	1
Lisa	Simpson	07.05.2003	06.05.2003	2
Fred	Clever	30.11.2004	02.12.2004	3
Jeff	Smart	10.02.2003	13.02.2003	4
Roland	Munster	09.07.2004	09.07.2004	2

In this exercise, you add an additional column *Duration* to the table. There is no input field for this column. However, the duration is the difference between the start date and the end date and can thus be calculated from the input of the corresponding fields.

- 6-3-1 Update the structure *Person* in the local dictionary. Add an additional attribute *DURATION* (type long).
- 6-3-2 Update the context of the component controller.
 - 6-3-2-1 Add the attribute *DURATION* to the context value node *Person*.
 - 6-3-2-2 Set the *calculated* property of this attribute *true*.
 - 6-3-2-3 Implement the generated method *getPersonDURATION(...)*. Calculate the attribute *DURATION* as the difference between the attributes *DATE_END* and *DATE_BEGIN*.
- 6-3-3 Update the context of the view *TableView*. Add the additional attribute *DURATION* from context *Person* and map it to the component context.
- 6-3-4 Update table *Person* on *TableView*. Add the additional attributes *DATE_BEGIN*, *DATE_END*, and *DURATION* from structure *Person*.
- 6-3-5 Create the Web Dynpro application *WD01_Basics_UI_02*
- 6-3-6 Build, deploy and run the application