

A screenshot of a computer

Description automatically generatedToday we are going to discuss a key concept in RESTful application programming model known as action.

In the context of RESTful application programming model.

An action is a non-standard operation that is a part of business logic.

It is used to make changes to the data of business object instance.

Unlike the standard operations when using action, it is not a standard update operation that's get

called, but the action with the predefined update implementation.

This allows developers to define custom operations that can modify the entity instance.

An action in RESTful application programming model needs to be defined in behavior definition and must

be implemented in behavior implementation class.

Let's take a simple example.

We need a button on your screen that can modify the field value of selected instance.

As you can see in this example, here we have a button set status by clicking on which the booking status

value will get modified from blank to x.

To achieve this functionality, we need to make changes in CDs behavior definition.

Behavior, projection, and in the annotations of projection view or in metadata extension.

Plus, we need to write a code in behavior implementation to modify the content in business object instance.

Now let's implement this functionality.

Go to edit tool in eclipse.

Here.

First we need to get into behavior definition.

And add action.

Definition.

We are using keyword action to define non-standard modify operation.

After keyword action, you need to specify the action name.

The output parameter for action is defined with the keyword result.

The result cardinality for action determines the multiplicity of output.

It indicates whether the action produces zero to 1 or 0 to many, or one to many output instances.

Here.

Result one produces output instance up to one data set.

When you return a result entity with self in UI service, the UI always stays on the same page where

the action is executed.

Now let's click on this yellow mark.

And add a method.

Here to save your time.

I have already written a code that I'll be using here.

It is very simple and straightforward code.

So let's first activate behavior definition.

And then we need to activate behavior implementation.

Let's set a breakpoint here.

I will explain you the operation that we are performing here.

But before that, we need to complete few more steps.

Now let's go to the projection behavior.

Here like a standard operations.

We need to define.

Non-standard operation action using keyword use action.

Action defined in behavior definition must be included in projection behavior definition if you wants

to expose it for the OData service.

Now let's activate it.

In the next step, we want to have that action button to be get displayed on the UI screen.

For that purpose, we need to get into metadata extension.

Based on the selection of the action button, we want to perform activity on booking status.

That is nothing but a field cancelled.

So here we need to add a few more annotations to link this field to the non-standard modify operation

action.

For action.

Use the data field type to expose the action to the client.

Data.

Action annotation.

Reference the technical name of the action.

And this is the label given to the action button.

So these annotations forms the linkage between.

Field cancelled, which is nothing but the booking status and the action button set status.

Now let's activate this one.

So as of now we have defined action in behavior definition.

We have implemented the operation against the action button in behavior implementation.

We have exposed the action in behavior projection and to display action button on screen we have added

annotations in metadata extension.

So now let's preview the Fury application.

So as you can see, the set status button appeared here.

Okay.

As of now, this said status button is disabled, but once we select any instance, that button gets

enabled.

So as per our functionality, once we click on these button set status.

It should set booking status against this instance as x.

Okay so now let's hit a set status button.

Since we have set a breakpoint control reach to this statement where we have set the breakpoint.

Now let's execute this statement.

So using this statement here, we are modifying the canceled field of a selected instance.

By passing the value to that field as x, along with the key fields of instance.

Let's break down this statement to understand it step by step.

The email statement modify entities of along with the keyword update fields is used to update specific

field of an instance.

In our case, this field is cancelled.

After keyword modifier entities of you need to specify root entity name in local mode.

Helps to exclude the authorization checks after the keyword entity.

You need to specify entity on which you wants to operate under the root entity, since in our case,

our business object has only one entity.

So that's why the root entity name and the entity name is same.

Then here after keyword update fields, you need to specify the fields that you want to update.

Here the value operator is used to create the internal table.

Which contains.

Key fields and fields cancel.

Here using for iteration, we are looping at derived internal table keys, which contains.

Key field information that we have passed on UI screen.

Then here we are populating value against the field.

Cancelled.

And populating the key fields.

So as a result of execution of this statement for the selected instance on the screen, the field cancelled

will get marked with X.

This statement over here is used to read change data for action result.

Here we are populating the parameter action result.

Read entities of root entity name in local mode, then specify the entity name on which he wants to

operate.

Now here we are reading all the fields corresponding to these key fields.

This is again a derived internal table which contains.

Key field information that we have passed through selection screen.

As a result, this internal table contains.

All the fields of entity.

Along with the modified field information.

Now that information we need to pass to the result parameter of action.

But that information needs to be passed in particular format.

As you can see here.

The result structure contains key field information and one component percent param.

This component has all the fields that are there in entity.

That's why we are using here value operator to create an internal table and pass it to the result parameter

of action.

Using for iteration.

We are looping at the content in this internal table that we have read against the key field of selected

instance.

And then here we are simply formatting it.

So as you can see here we are passing the key fields.

And then all the fields to this parameter.

So using this statement we are simply filling the required information in action.

Result.

Okay, so now let's do F8.

And let's go to Fury application.

As you can see here, the booking status against this instance against this record is marked X.

Let's remove the breakpoint.

And this time let's select.

Two records.

And heat set status.

So you can see for both the records the booking status is set to X.

So with this example I hope you understood the working of action.

Thank you for watching this video.

Have a nice day ahead.