

GETTING STARTED IN BSP

Summary

The code sample describes the step by step activities to be performed while creating and executing a simple BSP. It is suitable for budding BSP developers who needs a start up.

Created on: 22 May 200

Author Bio



Prabaharan Gnanasekaran is working for Wipro Technologies for the past 18 months as an ABAP Consultant.

Table of Contents

Create a Table of Contents in Microsoft Word. To change, right-click on the TOC and select Update Field.

Table of Contents	2
Code Explanation	3
Creating the BSP Application (Tcode: SE80)	4
Creating the pages for BSP Application	5
Coding for Layout of Enter Page :	9
Coding written under Event handler for Enter Page:	12
Coding written inside the Layout of Empid.htm page:	13
Disclaimer and Liability Notice	17

Code Explanation

The code explains the steps to create a simple BSP Application.

The BSP Application has three pages.

enter.htm

empid.htm

all.htm

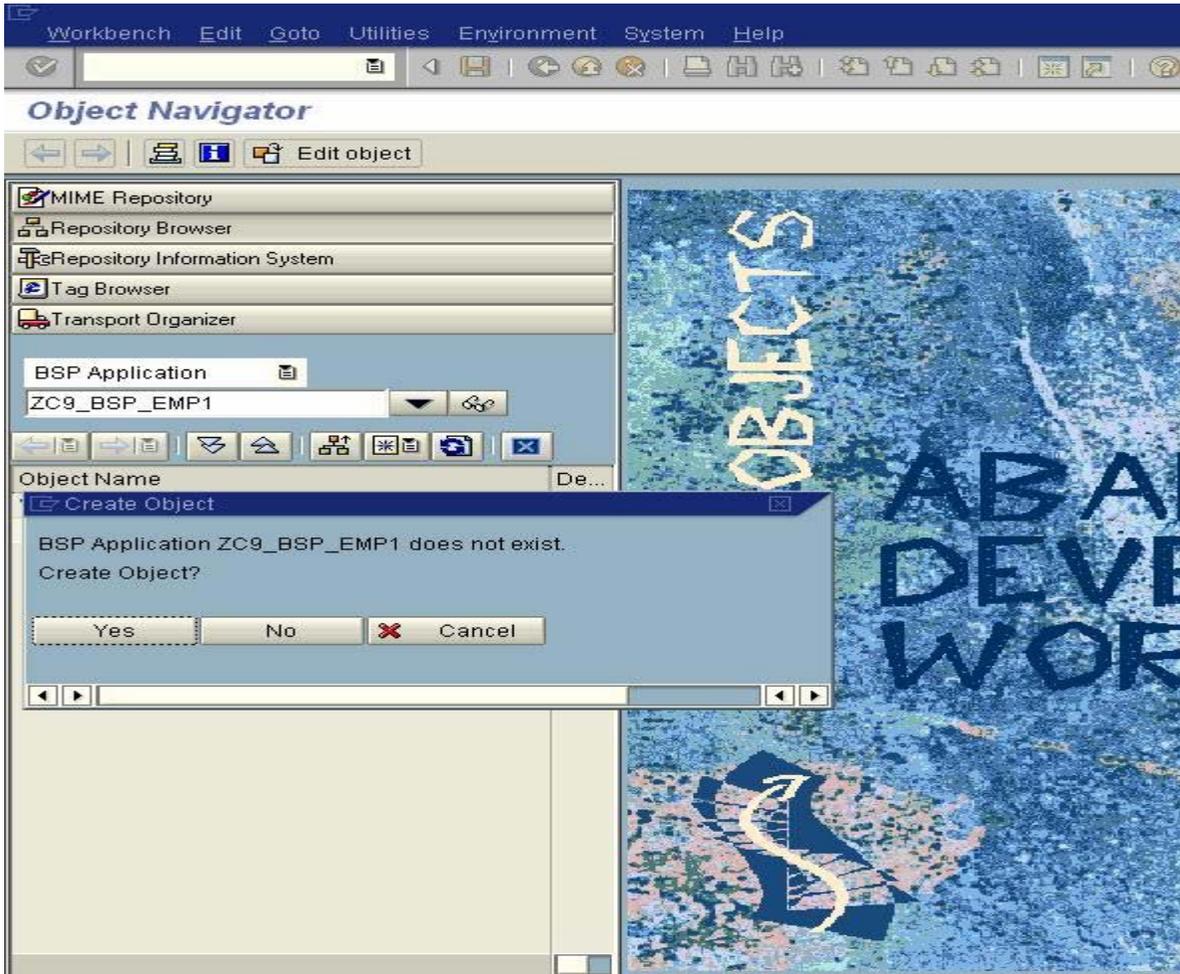
a) Each page holds the layout area where the code for page layout tab (how the page should look like) is written.

b) The Pages also have the event handler tab section under which the button click event of the pages are handled.

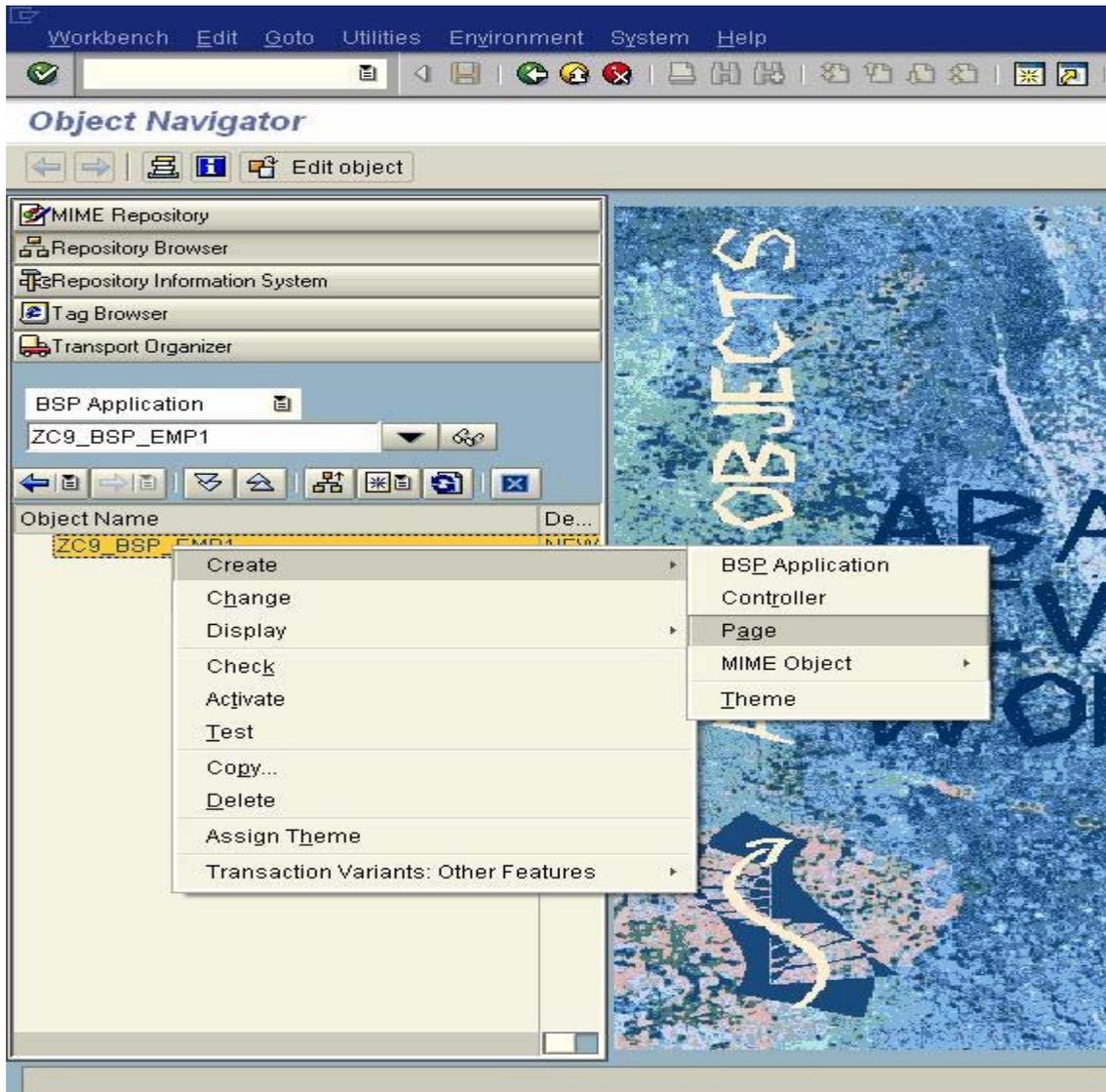
c) The page attributes tab gives a list of all data parameters used within the page and those that are transferred to the successive pages.

d) The code below holds sufficient inline comments which will be useful for better understanding. The screen shots will aid the reader to easily follow the coding.

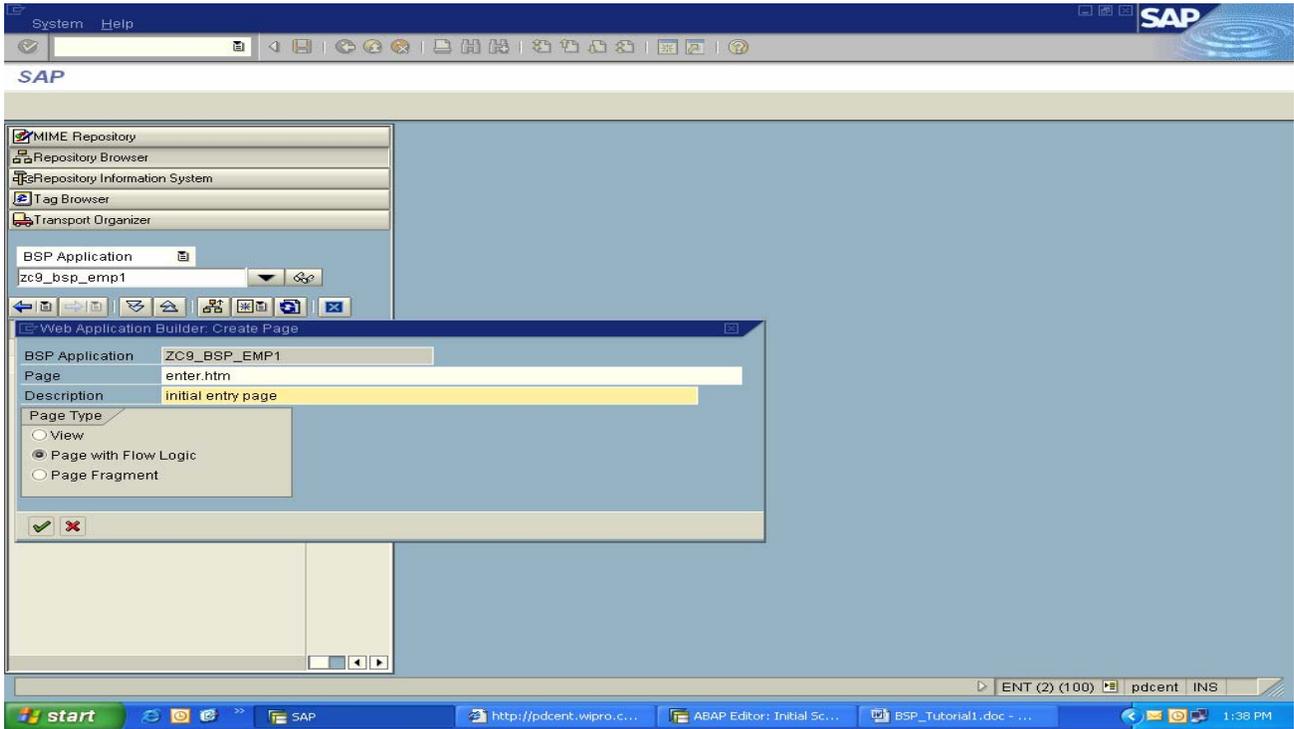
Creating the BSP Application (Tcode: SE80)



Creating the pages for BSP Application



Page should be created as page with flow logic



The screen shot showing the Layout,Event handler,Page Attributes Tabs of a page



Specifying the data parameters (used in the page) in page attributes

SAP

BSP Page Edit Goto Utilities Environment System Help

Web Application Builder: Edit Page ZC9_BSP_EMP

MIME Repository
Repository Browser
Repository Information System
Tag Browser
Transport Organizer

BSP Application
ZC9_BSP_EMP

Object Name
ZC9_BSP_EMP
Pages with Flow Logic
allemp.htm
empid.htm
enter.htm

Page enter.htm Active

Properties Layout Event Handler Page Attributes Type Definitions Preview

Attribute	Auto	Typin...	Associated Type	Description
employeeid	<input type="checkbox"/>	TYPE	STRING	
	<input type="checkbox"/>			

SAP

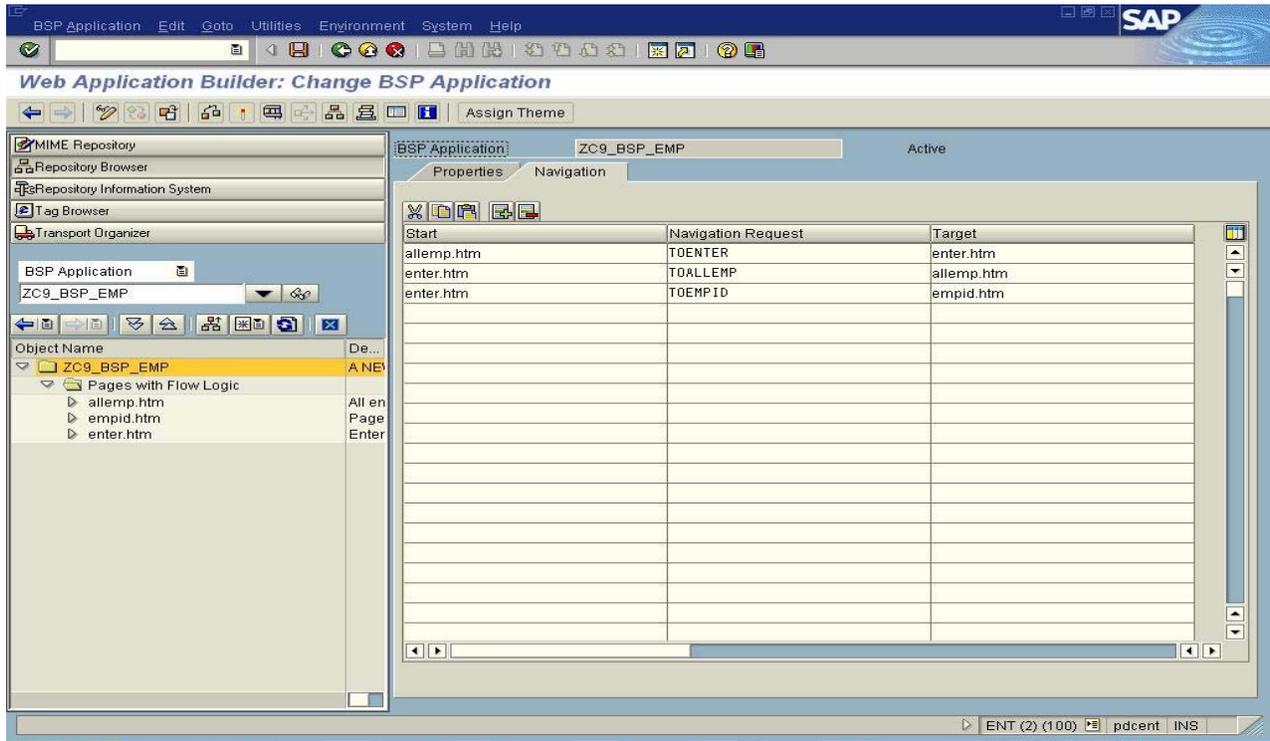
Web Application Builder: Edit Page ZC9_BSP_EMP

Page: allemp.htm

Attribute	Auto	Typin...	Associated Type	Description
_emp	<input type="checkbox"/>	TYPE	ZC9_EMPLOYEE	TEST TABLE BSP
_emp1	<input type="checkbox"/>	TYPE	ZC9_EMPLOYEE	TEST TABLE BSP
wa_emp	<input type="checkbox"/>	TYPE	ZC9_EMPLOYEE	TEST TABLE BSP

ENT (2) (100) pdcent INS

Select the main BSP Application and Fill the Navigation Request.



Coding for Layout of Enter Page :

The coding lines in below code starting with tag <! Indicates the Comment lines which is written for easy understanding.

*-----

<! The below code is a page directive which specifies the language used in the page >

<%@ page language="abap" %>

<html>

<head>

<! HTML Coding to add the Heading , an image from local desktop>

<h1> Employee Details </h1>

<body BGCOLOR="#B5E1D2" >

<h2> Welcome to our Employee Details Display </h2>

<p>

<! The interactive Input and Event triggering elements of the Page are written inside a form tag >

<form method="post">

<! Label for Text Box >

Employee-id

<!--Inserting a Text box input field (Type,Name,Value) -->

<input type=text name="employeeid" value="" >

<p>

<!--Inserting two buton field -->

<input type=submit name="onInputProcessing(select)"

value="select employee record">

<! In above code when the button is clicked the event routine written under the Event Handler
tab is called and executed >

```
<p>
```

```
<input type=submit name="onInputProcessing(all)"  
value=" list all employees ">
```

```
</form>
```

<! the below is written in event handler area of page>

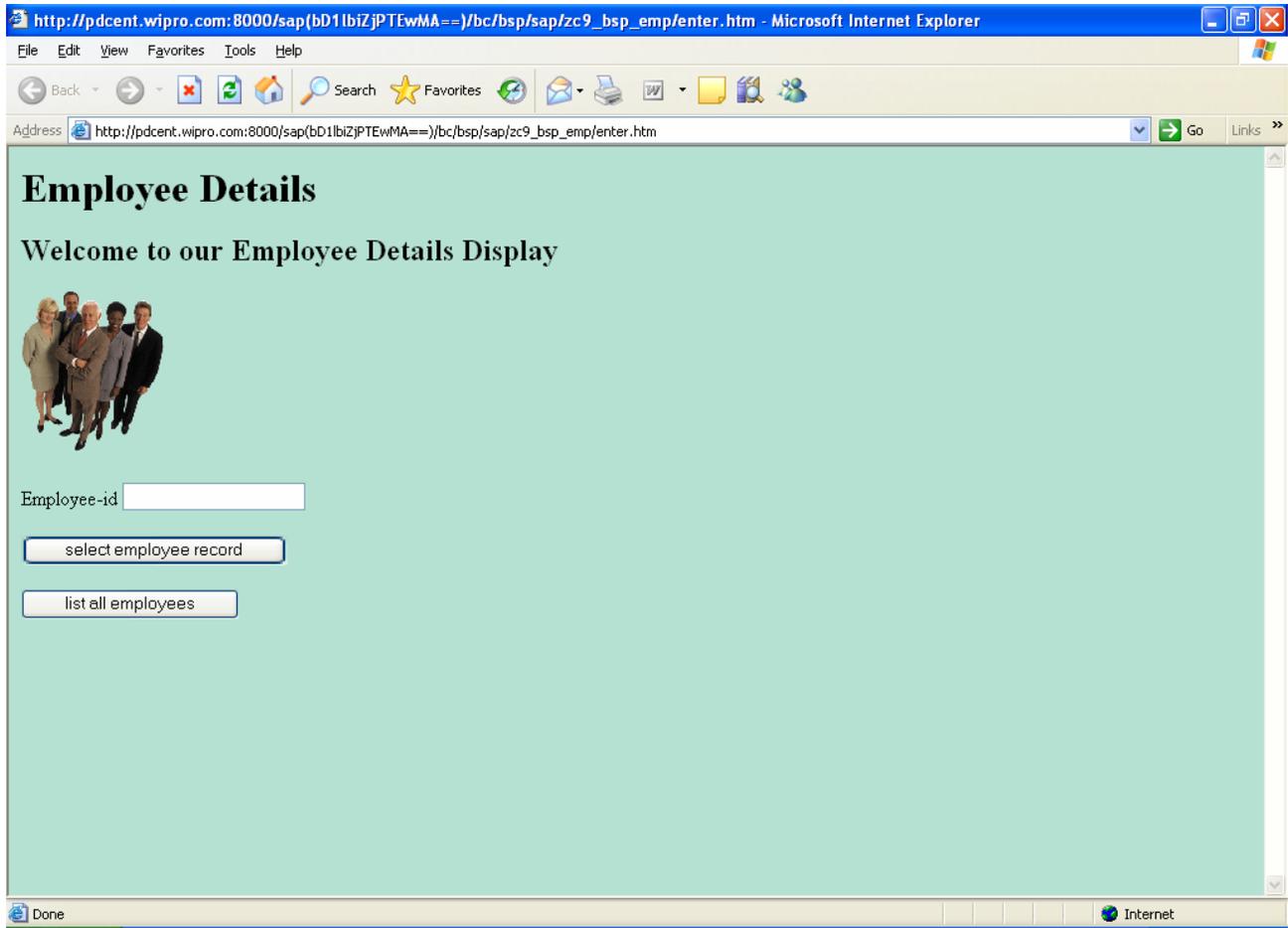
<!If in case in the target page-attrib there is automatic parameter employeeid then its value is passed as >

```
<!--navigation->set_parameter( 'employeeid' )-->
```

```
</body>
```

```
</html>
```

The layout coding for enter.htm page gives this output



Coding written under Event handler for Enter Page:

case event_id.

*Event_id holds the name of the event triggered in the page

when 'select'.

navigation->set_parameter('employeeid').

*The above command sets the value of the parameter employeeid so that it can be accessed *in successive pages.

navigation->next_page('TOEMPID').

*Here The next page which the browser should navigate to is specified. TOEMPID is the *navigation request which specifies the next page. *Refer to the screen shot above for navigation request.

when 'all'.

navigation->next_page('TOALLEMP').

when 'others'.

endcase.

Coding written inside the Layout of Empid.htm page:

The standard ABAP Coding to be written inside the Layout page is written between a <% and %> tag and it is automatically colored blue on saving the page.

```
<%@page language="abap"%>
<%@extension name="htmlb" prefix="htmlb"%>

<html>

  <body BGCOLOR="#B5E1D2">

    <h2> Employee Detail </h2>

    <! ABAP code to select the employee id from the table.>

    <% data: v_employeeid type zc9_employee-emp_id. %>
    <% v_employeeid = employeeid. %>

    <h3> selected Employee-ID : <%=employeeid %> </h3>
```

```

<% clear wa_emp.
select single * from zc9_employee into wa_emp where emp_id = employeeid. %>
<% if sy-subrc <> 0 or wa_emp-emp_name is initial . %>
<h3> Employee Record not found </h3>

<%else . %>

<! Constructing a table to display the retrieved record >
<table border=1>
  <tr>

    <td><b>Employee ID</b></td>
    <td><b>Employee Name</b></td>

  </tr>

  <tr>

    <td> <%= wa_emp-emp_id %> </td>

    <td> <%= wa_emp-emp_name %> </td>

  </tr>
<% endif. %>
</table>
</body>

</html>

```

Coding written under the layout of allemp.htm page:

```
<%@ page language="abap" %>
```

```
<html>
```

```
<body BGCOLOR="#B5E1D2">
```

```
<h2> List of Employees </h2>
```

```
<table border=1>
```

```
<tr>
```

```
<td><b>Employee ID</b></td>
```

```
<td><b>Employee Name</b></td>
```

```
</tr>
```

```
<% data: wa_emp type zc9_employee.
```

```
data: i_emp type table of zc9_employee.
```

```
select * from zc9_employee into table i_emp.
```

```
loop at i_emp into wa_emp. %>
```

```
<tr>
```

```
<td> <%= wa_emp-emp_id %> </td>
```

```
<td> <%= wa_emp-emp_name %> </td>
```

```
</tr>
```

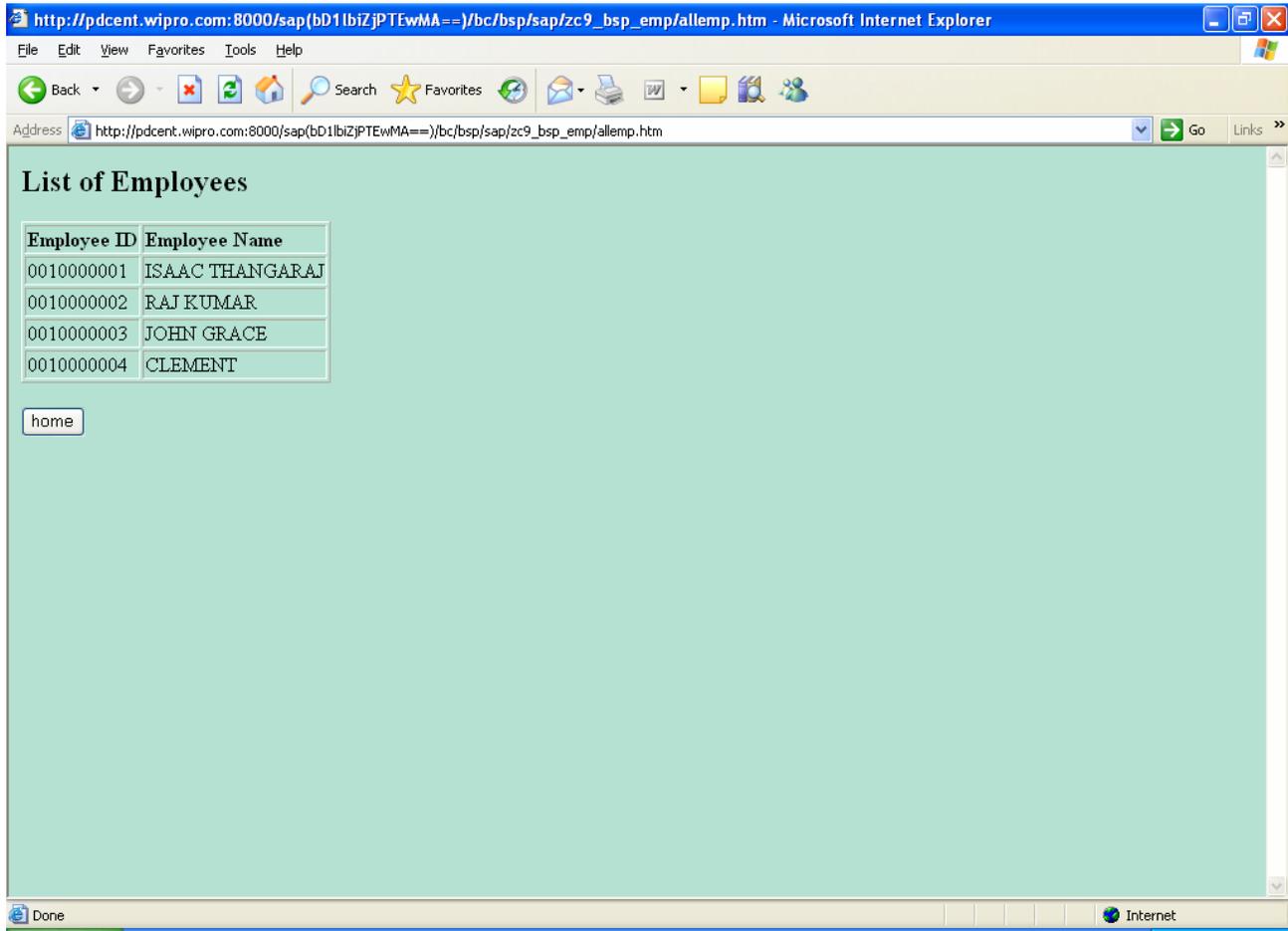
```
<% endloop. %>

</table>
  <form method="post">

  <p>
<! A button to navigate to home page ( enter page ) >
  <input type=submit name="onInputProcessing(back)" value="home">
  <p>

  </form>
</body>
</html>
```

The output of allemp.htm page listing all the employee details – called on clicking the list all employees button in enter.htm page



Disclaimer and Liability Notice

This document may discuss sample coding or other information that does not include SAP official interfaces and therefore is not supported by SAP. Changes made based on this information are not supported and can be overwritten during an upgrade.

SAP will not be held liable for any damages caused by using or misusing the information, code or methods suggested in this document, and anyone using these methods does so at his/her own risk.

SAP offers no guarantees and assumes no responsibility or liability of any type with respect to the content of this technical article or code sample, including any liability resulting from incompatibility between the content within this document and the materials and services offered by SAP. You agree that you will not hold, or seek to hold, SAP responsible or liable with respect to the content of this document.