

# ABAP101 – Tic Tac Toe – 1st ABAP Open Source Championship

## Presentation

Hello everybody! Welcome to the first ABAP Championship. I'm Fabio Pagoti from ABAP101.com and in this video I'm going to show you how to join the championship and as a consequence, let you enjoy a great opportunity to learn ABAP at the same time you can have some fun.



## About ABAP101 blog

Before start, let me introduce you blog ABAP101.com. The blog was created by Flavio Furlan in 2007 and our aim is to keep being a reference in the Brazilian SAP Community, helping ABAPers not just to be better developers, but to help them to be better professionals. One of our main pillars is open source projects. We understand that they can be a valuable starting point for those who would like to start to work with ABAP. The 1<sup>st</sup> ABAP101 championship is based on one of our Open Source projects hosted in SAP Code Exchange, the ABAP101 Tic Tac Toe project.



ABAP101 is written in Portuguese, but you can read our blog in your native language using tools like Google translator. Also, I invite you right now to subscribe our newsletter and to follow us in the media, where you can stay tuned in our projects and of course, keep learning ABAP with us.

### ABAP in the Media

- <http://abap101.com>
- <http://www.facebook.com/abap101>
- <http://www.twitter.com/abap101>
- ABAP101 Newsletter

## **The Project**

The challenge behind the championship is to develop a custom SAP object which will be used by ABAP101 Tic Tac Toe project. This is a project which has a very simple requirement: be a tic tac toe game. However, it was completely done using Web Dynpro 4 ABAP, ABAP Objects, MVC, and other cool stuff. I can guarantee you that the game is quite surprising at the first sight. The great thing is that you don't need to know all these technologies. In fact, you will only need a very basic understanding of ABAP Objects to participate in the championship.

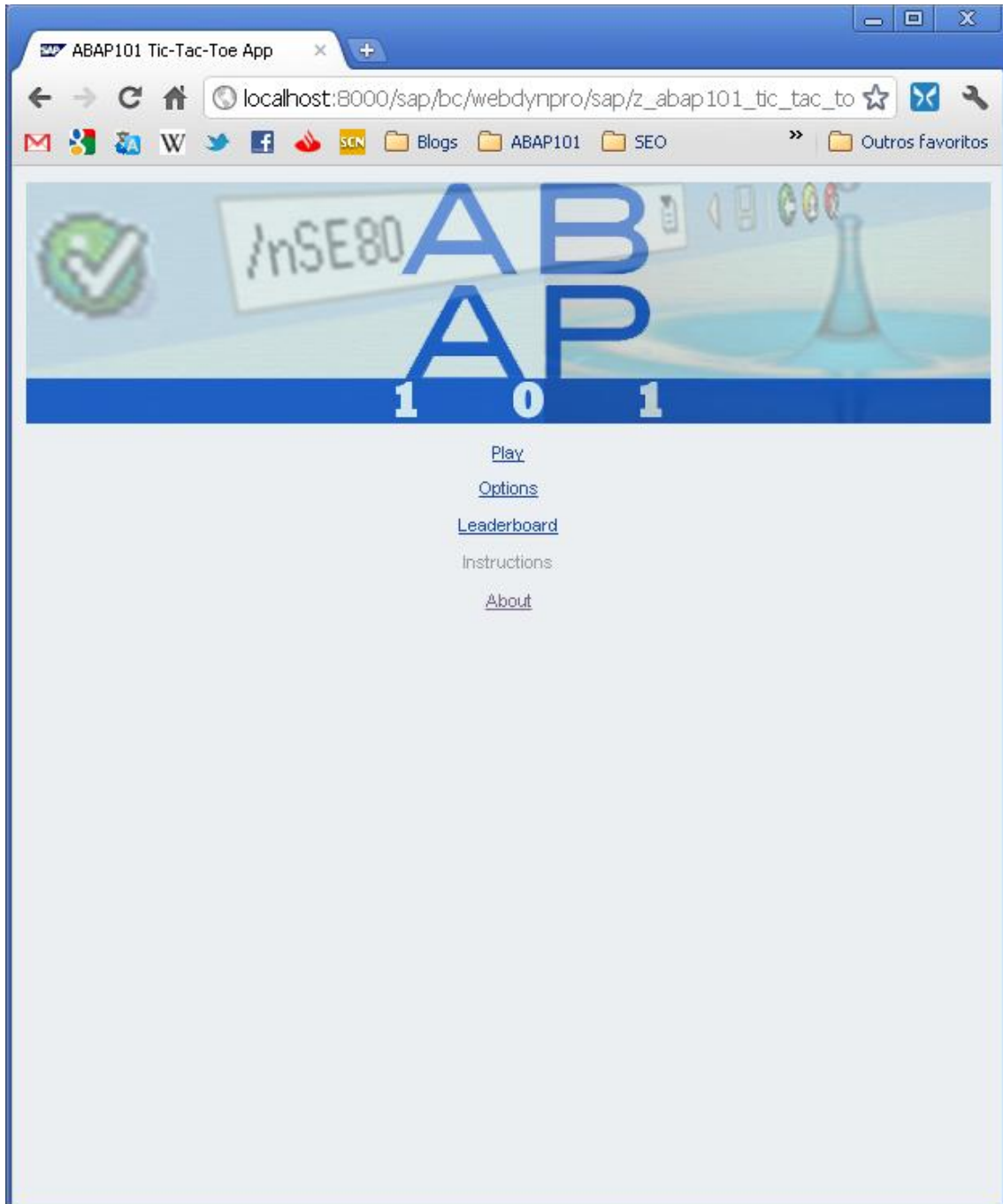
## **Installing the Project**

To install the project in your ABAP Trial version, visit our project home page in code exchange. There'll find all necessary instructions to install it.

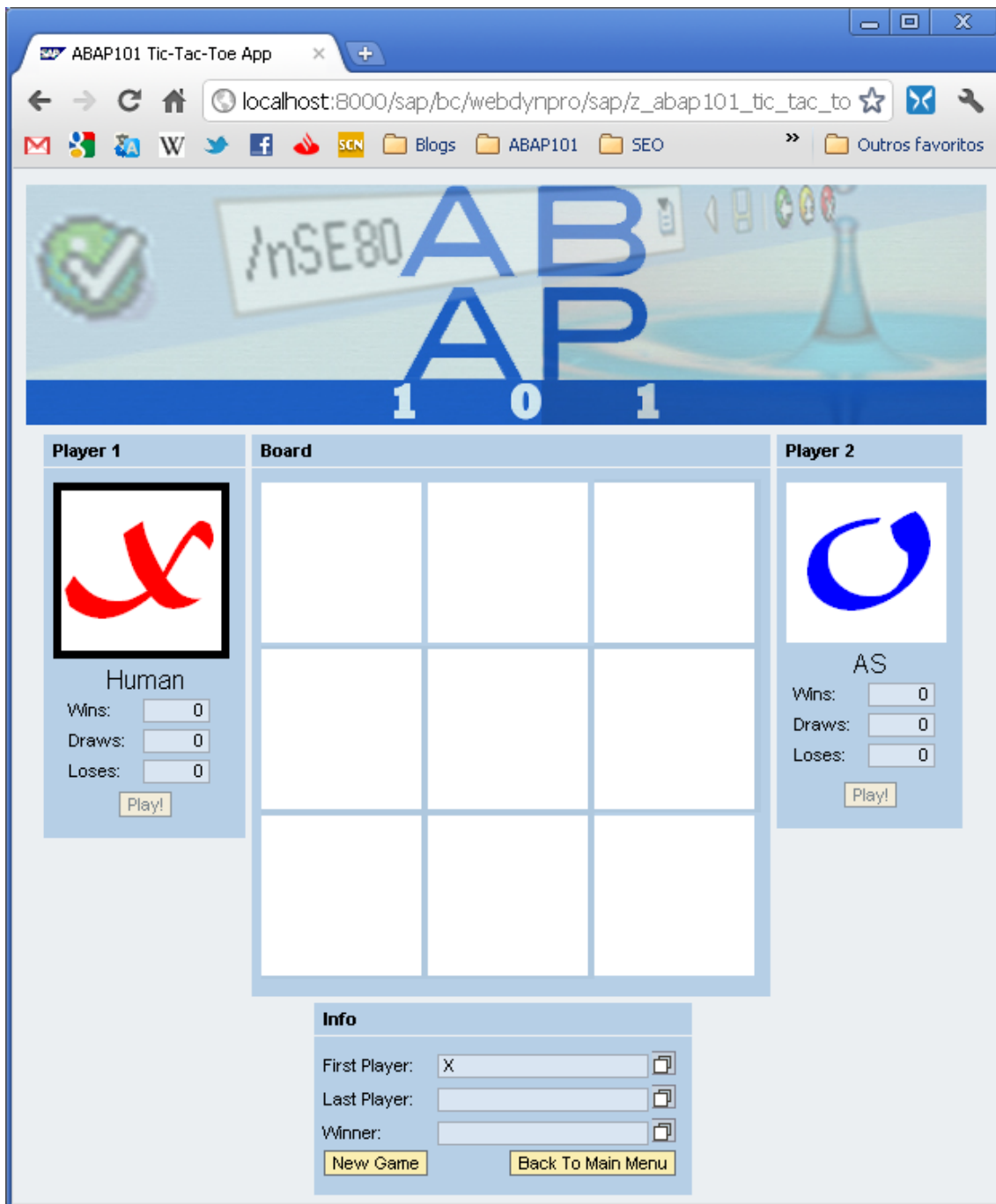
## How to play

After completely installing the project and before launching the challenge, let me show you how to play the game. This demonstration is important because it will help to understand what will be necessary to be done.

Run the WebDynpro Application located inside project's WebDynpro Component. The game menu will be shown.



To start playing, click at option "Play". The game board will be displayed.



The screen has three different and important pieces of information.

The main one, located in the center of the screen, is the game board. Internally, its squares are numbered from 1 to 9.

The second, are the two boxes located in each side of the screen. They show information about game players, like player's name, player's type, and their score.

Below the screen, there's the third and last box, where you can check game status.

The player responsible for the next turn has a black border in its avatar.

Below each player avatar, you can check the type of the player. There are two types of player, a human player and a non-human player. By default, the first player is a Human and the second player is a non-human player. This means that the ABAP Application Server is responsible for making the choices during the game related with the next play.
























To play a turn using a human player, just click in the desired blank square and it will be updated with the avatar corresponding to the respective player. To play with a non-human player, just click on “play” button located inside player’s information box. It will only be enabled when the player is the responsible for the current turn.

When the game ends, players’ score are updated, a message is displayed in the upper part of the screen and no more turns are possible. You can start a new game clicking in “new game” button.

ABAP101 Tic-Tac-Toe App

localhost:8000/sap/bc/webdynpro/sap/z\_abap101\_tic\_tac\_toe\_a

Game over

<b>Player 1</b>	<b>Board</b>	<b>Player 2</b>									
 Human Wins: <input type="text" value="4"/> Draws: <input type="text" value="1"/> Loses: <input type="text" value="2"/> <input type="button" value="Play!"/>	<table border="1"> <tr> <td></td> <td></td> <td></td> </tr> <tr> <td></td> <td></td> <td></td> </tr> <tr> <td></td> <td></td> <td></td> </tr> </table>										 AS Wins: <input type="text" value="2"/> Draws: <input type="text" value="1"/> Loses: <input type="text" value="4"/> <input type="button" value="Play!"/>
											
											
											
<p><b>Info</b></p> First Player: <input type="text" value="X"/> <input type="button" value="Copy"/> Last Player: <input type="text" value="X"/> <input type="button" value="Copy"/> Winner: <input type="text" value="X"/> <input type="button" value="Copy"/> <input type="button" value="New Game"/> <input type="button" value="Back To Main Menu"/>											

## Setting Game Options

You can define some game options, and here is where the challenge begins. In the game options , you can define player's settings, like their names and their types (I mean, human or non-human). If a player is defined as non human, you must define a global class responsible for its artificial intelligence. By default, the class "Lazy Player" ZCLABAP101\_TTT\_PLAYER\_LAZY is defined - I will talk a little bit about this class later. For now, it's important to know that the class defined here must follow some prerequisites. You can check if your class follows some of them using the "test class" button.

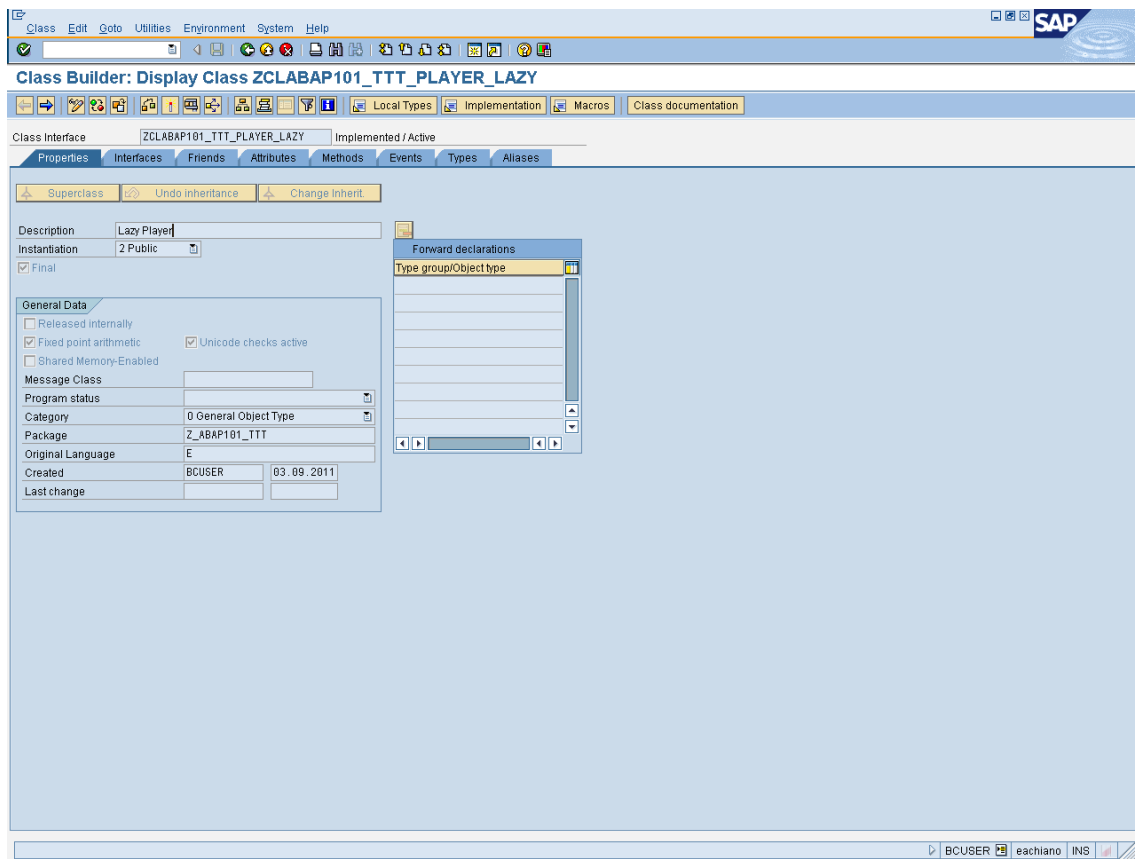
## **The challenge**

Your participation in the championship will be done through a custom global class created by you. You will have to build an artificial intelligence class for a tic tac toe player to be as smarter as possible. Given a game board, this class will have to decide what the best play is in order to win the game. You can build more than one class with different strategies if you want. So, you can have more possibilities to win the championship.

## How to build your custom class

The project already includes a custom global class which you can use as a model, I am talking about Lazy player class which is set as default inside game options. I have opened this class in Class Builder to show you what is mandatory in your class in order able it to be used in ABAP101 Tic Tac Toe game.

Basically, your class must have its instantiation public, implement a fixed interface and have a small piece of code inside its constructor. You define the instantiation in “Properties” tab in Class Builder.



The interface mandatory to be implemented is ZIFABAP101\_TTT\_PLAYER\_AS. You must include it in “Interfaces” tab. As soon you do so, attribute “my\_value” and method “DECIDE\_NEXT\_PLAY” will be created in your class. “My\_value” attribute will keep the information if your class is “X” or “O” during the game.



Method Edit Goto Utilities Environment System Help

SAP

Class Builder: Class ZCLABAP101\_TTT\_PLAYER\_LAZY Display

Pattern Pretty Printer Signature Public Section Protected Section Private Section

Parameter	Type spec.	Description
IM_BOARD_CONFIGURATION	TYPE REF TO ZABAP101TTT_BOARD	ABAP101 - Tic Tac Toe - Board
VALUE(RE_BOARD_POSITION)	TYPE ZABAP101TTT_BOARD_VALUE	ABAP101 - Tic Tac Toe - Board Value

Method ZIFABAP101\_TTT\_PLAYER\_AS-DECIDE\_NEXT\_PLAY Active

```

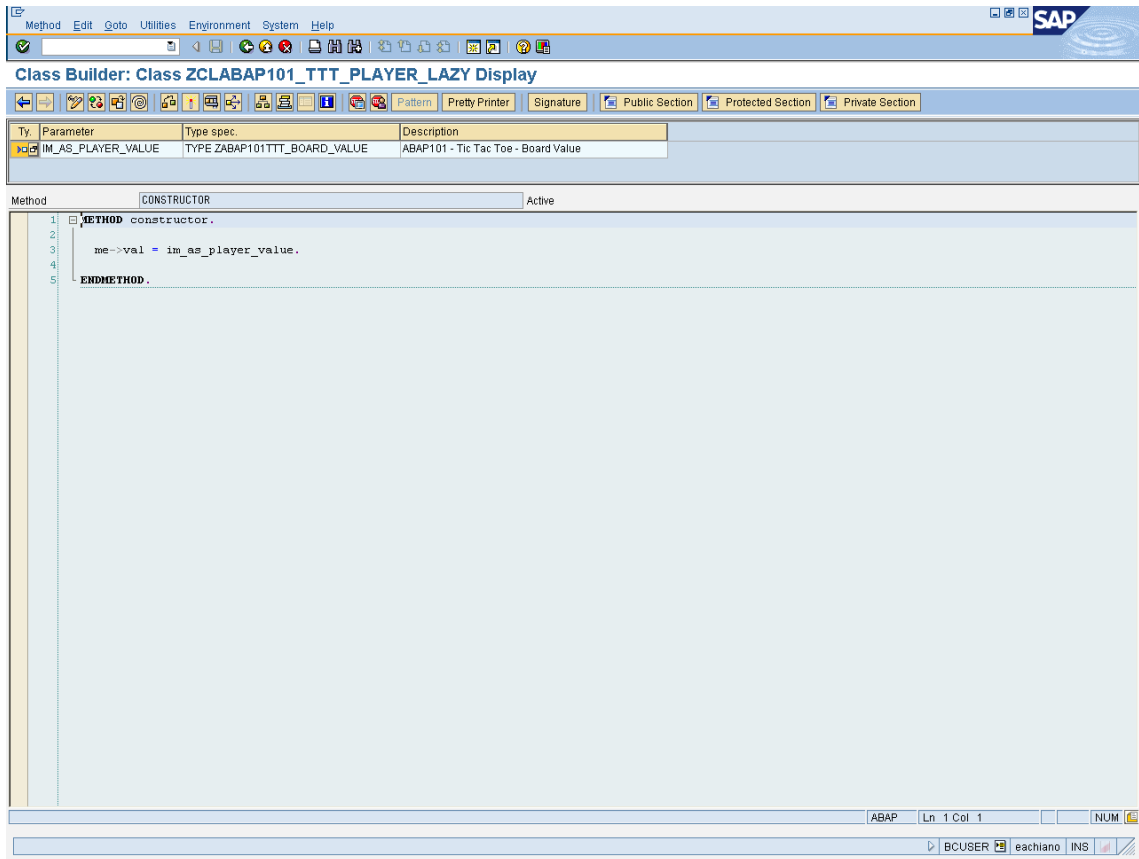
1 | METHOD zifabap101_ttt_player_as_decide_next_play.
2 |   DATA: l_i_board TYPE zabap101ttt_board_table,
3 |         wa_board TYPE zabap101ttt_board_struct.
4 |
5 |   CALL METHOD im_board_configuration->get_board
6 |     RECEIVING
7 |       re_board = l_i_board.
8 |
9 |   LOOP AT l_i_board INTO wa_board.
10 |     IF wa_board-value = space.
11 |       re_board_position = wa_board-pos.
12 |       RETURN.
13 |     ENDIF.
14 |   ENDLOOP.
15 |
16 | ENDMETHOD.

```

Scope IMETHOD zifabap101\_ttt\_player\_as-decide\_next\_play ABAP Ln 1 Col 1 NUM

BCUSER eachiano INS

Last but not least, you must implement a constructor for your class. This constructor must have its formal parameters exactly as the constructor of Lazy Class. Formal parameters names must be followed exactly as your constructor will be called dynamically during the game.



Inside constructor, you'll have to add the first line of code from Lazy constructor:

```
me->ZIFABAP101_TTT_PLAYER_AS~val = im_as_player_value.
```

This code just stores a "X" or "O" in "my\_value" attribute as soon a object is created. You can have more code inside the constructor if you want.

As soon your class is finished, you can use it the game.

## **Now, it's your turn!**

Now that you have all the information needed to build your class, don't lose time. Study the project, its classes and implement the best logic. Of course, have some fun with it too!

## **Stay tuned!**

Stay tuned In [ABAP101.com](http://ABAP101.com), in Tic Tac Toe project in SAP Code Exchange and in my blog at SDN – there you can read more about the championship, its rules, prizes and look for some hints. Good luck! Thanks for watching.