

SoapUI *Pro*(ReadyAPI)

Data Sources

Data Source	Description
Data Connection	Gets data from a database by using a JDBC connection. It is similar to the JDBC data source, but uses predefined JDBC database connections .
Data Generator	Generates data by using the built-in data generator engine.
Excel	Gets data from the Excel tables.
File	Obtains content of the specified file line-by-line.
JDBC	Gets data from a database by using a JDBC connection. Unlike the Data Connection data source, it uses a connection specified in the data source.
Grid	Obtains data specified in a grid.
Groovy	Generates data by using a Groovy script.
JSON	Parses the desired data from the JSON content with JSONPath expressions.
XML	Parses the desired data from the XML content with XPath expressions.
Directory	Iterates through the specified directory and gets file content.

DDT –Data Source as Excel

DataSource-XLS

DataSource: Excel

Properties

CountryName
ISOCode

Configuration

File: C:/SoapUITests/Countries.xlsx

Worksheet: Sheet1

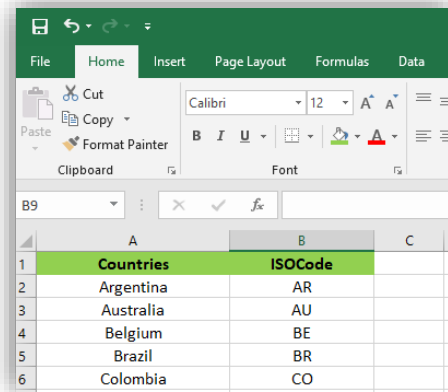
Start at Cell: A2

Ignore Empty: ☐ Select to skip empty rows

Evaluate formulas: ☐ Select if the file contains formulas that should be evaluated

[Learn about data sources](#)

CountryName	ISOCode
Argentina	AR
Australia	AU
Belgium	BE
Brazil	BR
Colombia	CO
Argentina	AR
Australia	AU
Belgium	BE
Brazil	BR
Colombia	CO



The screenshot shows the Microsoft Excel interface with the 'Home' tab selected. The ribbon includes options for File, Home, Insert, Page Layout, Formulas, and Data. The 'Clipboard' group shows Cut, Copy, and Paste options. The 'Font' group shows Calibri font, size 12, and various formatting options. The active cell is B9. The data is organized in a table with two columns: 'Countries' and 'ISOCode'.

	A	B	C
1	Countries	ISOCode	
2	Argentina	AR	
3	Australia	AU	
4	Belgium	BE	
5	Brazil	BR	
6	Colombia	CO	

DDT –Data Source as Text File

DataSource-TXT

Properties

CountryName
ISOCODE

Configuration

DataSource: File

File: C:/SoapUITests/Countries.txt

Separator: ,

Charset: Cp1252

Trim: ☒ Trim values

Quoted Values: ☐ Values are in quotes

[Learn about data sources](#)

CountryName	ISOCODE
Argentina	AR
Australia	AU
Belgium	BE
Brazil	BR
Colombia	CO
Argentina	AR
Australia	AU

Countries.txt - Notepad

File Edit Format View Help

Argentina,AR
Australia,AU
Belgium,BE
Brazil,BR
Colombia,CO

DDT –Data Generator

DataSource-DataGeneration

DataSource: Data Generator

Properties

CountryName

Configuration

Number of rows: 10

Property type: Country

Duplicate each value: ☐ 1 time(s)

Sample:

Barbados
Cayman Islands
Palestinian Territory, Occupied

Set Default Value

[Learn about data sources](#)

CountryName
Djibouti
Taiwan, Province of China
Bulgaria
Serbia and Montenegro
Zimbabwe
Christmas Island
Japan
Mayotte
France
Holy See (Vatican City State)

DDT –Data Source as Grid

DataSource-Grid

Properties

- CountryName
- ISOCode

Configuration

DataSource: Grid

CountryName	ISOCode
<input checked="" type="checkbox"/> Argentina	AR
<input checked="" type="checkbox"/> Australia	AU
<input checked="" type="checkbox"/> Belgium	BE
<input checked="" type="checkbox"/> Brazil	BR
<input checked="" type="checkbox"/>	

[Learn about data sources](#)

Delete rows Insert rows Clear Cells Fill Down Fill Right Select All Select None ☒ Recognize grid content as plain text

CountryName	ISOCode
Argentina	AR
Australia	AU
Belgium	BE
Brazil	BR

DDT –Data Source as JDBC

Data Source - JDBC1

Properties: COUNTRIES_COUNTRY_NAME, COUNTRIES_COUNTRY_ID

Configuration

Use prepared properties to create dynamic queries

+ Add Property...

Driver: oracle.jdbc.driver.OracleDriver

Connection String: jdbc:oracle:thin:hr/hr@localhost:1521/pdborcl

Password: ..

Test Connection: [Test]

SQL Query: `Select COUNTRIES.COUNTRY_NAME as
From COUNTRIES`

Fetch Size: 1

Stored Procedure: ☐ Select if this is a stored procedure

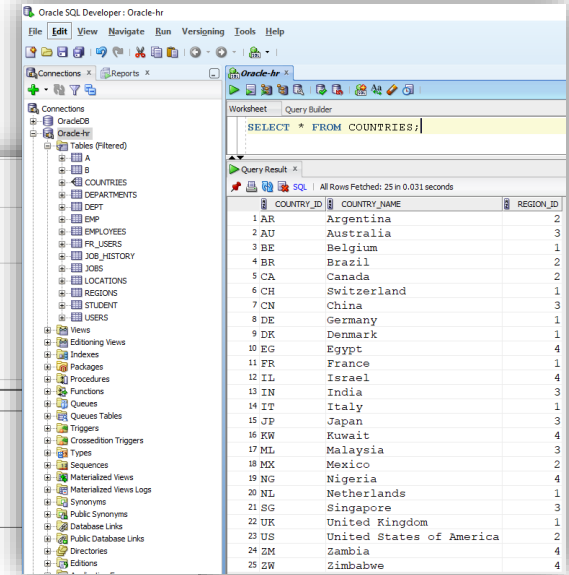
Use a wizard

Learn about data sources

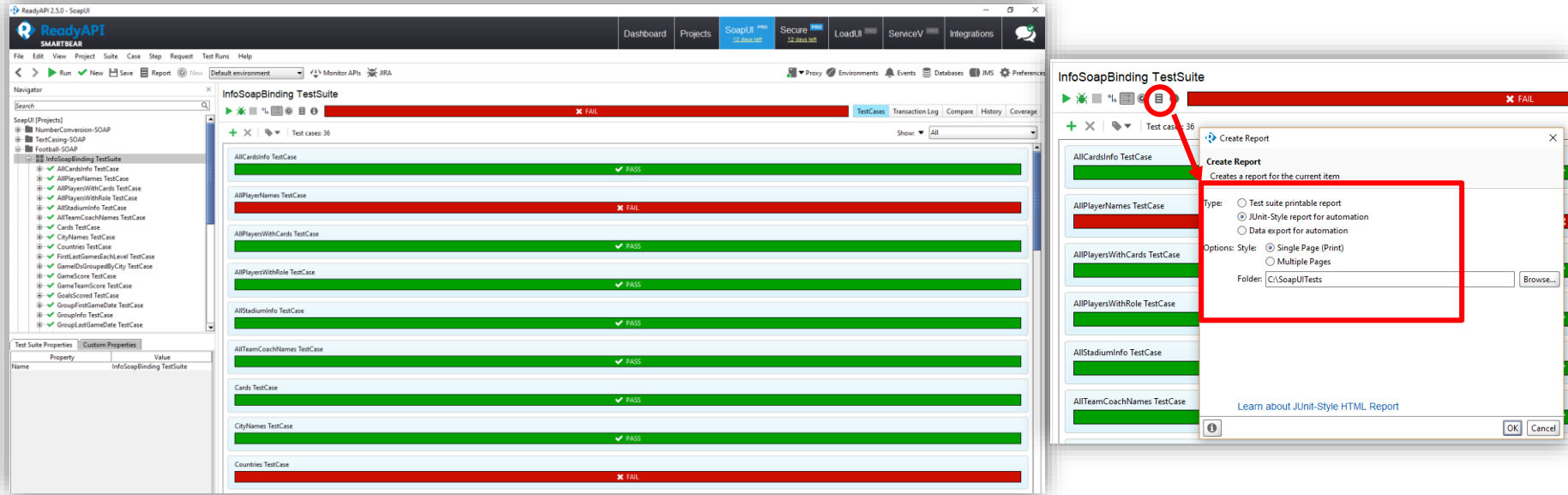
COUNTRIES_COUNTRY_NAME	COUNTRIES_COUNTRY_ID
Argentina	AR
Australia	AU
Belgium	BE
Brazil	BR
Canada	CA
Switzerland	CH
China	CN

- **JDBC placed Driver location:**
- **ReadyAPI:**
C:\Program Files\SmartBear\ReadyAPI-2.5.0\bin\ext

Driver : oracle.jdbc.driver.OracleDriver
Connection String: jdbc:oracle:thin:hr/hr@localhost:1521/pdborcl



Run Tests from GUI & Generate Reports



Run Tests through TestRunner

The screenshot illustrates the steps to run tests through the TestRunner in ReadyAPI 2.5.0 - SoapUI. The main interface shows the 'InfoSoapBinding TestSuite' selected in the Navigator. The context menu is open, and 'Launch TestRunner' is highlighted. The 'Launch TestRunner' dialog is shown with the 'Basic' tab selected. The 'Test Case' dropdown is set to '<all>'. The 'TestRunner Path' is set to 'C:\Program Files\SmartBear\ReadyAPI-2.5.0\bin'. The 'Add Settings' section is empty. The 'Project Password' is 'user-settings.xml Password:'. The 'Ignore Errors' section has three checkboxes: 'Ignore errors that occur during the test run' (unchecked), 'Save the project when the test run is over' (unchecked), and 'Use the same process as ReadyAPI' (unchecked). The 'Save After' section is empty. The 'Run in-process' section is empty. The 'Get Command Line' and 'Launch' buttons are highlighted with red boxes. The 'ReadyAPI TestRunner' window is shown with the command line: `"C:\Program Files\SmartBear\ReadyAPI-2.5.0\bin\testrunner.bat" "-sInfoSoapBinding TestSuite" -r -a -j -f$(WORKSPACE) -RJUnit-Style HTML Report -FXML -EDefault environment" E:\Football-sapui-project.xml`. The 'ReadyAPI TestRunner' window also shows the command: `cmd.exe /C testrunner.bat "-slInfoSoapBinding TestSuite" -r -a -j -f$(WORKSPACE)`.

ReadyAPI 2.5.0 - SoapUI

File Edit View Project Suite Case Step Request Test Runs Help

Default environment Monitor APIs JIRA

Navigator

Search

SoapUI [Projects]

- NumberConversion-SOAP
- TextCasing-SOAP
- Football-SOAP
 - InfoSoapBinding TestSuite
 - AllCardsInfo TestCase
 - AllPlayerNames TestCase
 - AllPlayersWithCards TestCase
 - AllPlayersWithRole TestCase
 - AllStadiumInfo TestCase
 - AllTeamCoachNames TestCase
 - Cards TestCase
 - CityNames TestCase
 - Countries TestCase
 - FirstLastGamesEachLevel TestCase
 - GameIdsGroupedByCity TestCase
 - GameScore TestCase
 - GameTeamScore TestCase
 - GoalsScored TestCase
 - GroupFirstGameData TestCase
 - GroupInfo TestCase
 - GroupLastGameData TestCase

Test Suite Properties Custom Properties

Property	Value
Name	InfoSoapBinding TestSuite

Launch TestRunner

Specify arguments for launching SoapUI TestRunner

Basic Overrides Reports Properties Tags Custom Args

Test Suite: InfoSoapBinding TestSuite

Test Case: <all>

Enable UI: <all>

TestRunner Path: C:\Program Files\SmartBear\ReadyAPI-2.5.0\bin

Save Project: test

Add Settings: apui-settings.xml file

Project Password: user-settings.xml Password:

Ignore Errors:
☐ Ignore errors that occur during the test run
☐ Save the project when the test run is over
☐ Use the same process as ReadyAPI

Save After:

Run in-process:

Get Command Line Launch Close

ReadyAPI TestRunner

"C:\Program Files\SmartBear\ReadyAPI-2.5.0\bin\testrunner.bat" "-sInfoSoapBinding TestSuite" -r -a -j -f\$(WORKSPACE) -RJUnit-Style HTML Report -FXML -EDefault environment" E:\Football-sapui-project.xml

Copy to Clipboard Cancel

ReadyAPI TestRunner

Running ReadyAPI TestRunner for [Football-SOAP]
directory: C:\Program Files\SmartBear\ReadyAPI-2.5.0\bin
command: cmd.exe /C testrunner.bat "-slInfoSoapBinding TestSuite" -r -a -j -f\$(WORKSPACE)

Cancel Close

Run Tests from Command Line & Generate Reports

- The following command runs all functional tests from the specified project:

```
cd C:\Program Files\SmartBear\ReadyAPI-2.5.0\bin  
testrunner.bat "E:\Football-soapui-project.xml"
```

- The following command runs the Test Case in your project and creates HTML files in a JUnit-style format:

```
testrunner.bat -FPDF -R"JUnit-Style HTML Report" -c "TestCase Name" "E:\Football-soapui-project.xml"
```

- Report URL:

file:/C:/Program%20Files/SmartBear/ReadyAPI-2.5.0/bin/InfoSoapBinding-TestSuite/Copy-of-Countries-TestCase/index.html

Data Driven Tests - Assignment

- WSDL:
- <http://dneonline.com/calculator.asmx?WSDL>

