

# Quantum: How to pass dynamic data from csv file?

## Step-by-step guide

1. Create a csv file containing the data, for example see the csv file source below that contains two pairs of Contact Names with Contact Phones:

```
contactName,phoneNumber  
  
Perfecto Quantum,1234  
  
Quantum Perfecto,4567
```

2. Place the file under `src/main/resources/data/`
3. Create a feature file that will take the csv columns as input and point to the data file as shown in the example below:

```
@dynamicData  
Feature: Add Contact  
  
Scenario Outline: Dynamic Data  
  Given I launch Contacts application  
  Then I Add Contact with name "<contactName>" and number "<phoneNumber>"  
Examples: {'datafile' : 'src/main/resources/data/contactNames.csv'}
```

4. Add an implementation for the step that will use the csv data, for example:

```

@When("I Add Contact with name \"([^\"]*)\" and number \"([^\"]*)\"$")
public void I_add_contact_with_name(String contactName, String phoneNumber) throws Throwable {
    DeviceUtils.switchToContext("NATIVE");

    if (!isAndroid()) {
        System.out.println("this is an iOS");
        getDriver().findElement("contacts.page").click();
    }

    //Add contact
    click("add.contact.btn");
    // in some Android devices, after clicking on add contact, a popup appears asking which account
to save the contact under
    //we will look for this and if it appears select the device option
    if (isAndroid()) {

        if(isText("Save contacts to",20)) {
            getDriver().findElement("save.phone").click();
        }

    }
    DeviceUtils.waitForPresentTextVisual("Phone", 20);
    //Insert name and number
    getDriver().findElement("add.name").sendKeys(contactName);
    Thread.sleep(2000);
    getDriver().findElement("phone.field").click();
    Thread.sleep(2000);
    getDriver().findElement("add.phone").sendKeys(phoneNumber);
    DeviceUtils.waitForPresentTextVisual(phoneNumber, 20);
    //Save contact

    getDriver().findElement("save.contact").click();

    if (isAndroid()) {
        DeviceUtils.waitForPresentTextVisual("connected via", 20);
        //return to main screen
        getDriver().findElement("back").click();
        DeviceUtils.waitForPresentTextVisual("contacts groups more", 20);
    }

}

```

5. Run the test by adding the scenario's tag to your TestNG xml configuration file.

## Related articles

- [Quantum Release Notes 1.20](#)
- [How to enable new architecture for Appium web and hybrid testing on iOS in Quantum](#)
- [How to enable new architecture for Appium web and hybrid testing on Android in Quantum](#)
- [Quantum: How to pass dynamic data from csv file?](#)
- [CI Jenkins integration and Quantum](#)