



Running Capybara Tests in Phresco

Version 3.0

October 2013

This document purports the execution steps involved in Capybara tests.

Specifications contained herein are subject to change and these changes will be reported in subsequent release notes or new editions.

© Copyright Photon Infotech Pvt Ltd 2013.

All rights reserved.

All Photon product names are either trademarks or registered trademarks of Photon. Other company and product names mentioned herein may be trademarks of their respective owners.

TABLE OF CONTENTS

1	About This Guide.....	3
2	Functional script structure	3
3	Develop scripts in Capybara.....	3
4	Pre-requisites for running test in Phresco	4
5	Executing test in Phresco	4

1 About This Guide

The purpose of this guide is to help test engineers execute test scripts in Phresco, developed using Capybara with implementation in Ruby.

2 Functional script structure

The structure of the scripts should be as below

```
Parent Dir
/features/ -
Contains feature file where acceptance criteria for scenarios are defined

-/features/step_definitions/
Contains Ruby methods for different scenarios as .rb file

-/features/support/
Contains Ruby implementation code for verifying scenarios as .rb file

/resources/
Contains the configurations
```

3 Develop scripts in Capybara

Define acceptance criteria in features file, for all scenarios of your application. Develop Ruby code to verify the acceptance criteria's.

4 Pre-requisites for running test in Phresco

- Install Ruby 1.9.3 and environment path has to be set
- Install latest version of Ruby Development kit
- Run the command “gem install cucumber” in terminal to install cucumber
- Run the command “ gem install capybara” in terminal to install capybara web libraries
- Run the command “gem install rspec” in terminal to install libraries for readable assertions
- Run the command “gem install selenium-webdriver” in terminal to install selenium webdriver API

5 Executing test in Phresco

1. Place the script in path /workspace/projects/project_name/test/functional
2. Execute the test as “Capybara” against functional framework