

UI Testing with SWTBot

Stefan Seelmann
mail@stefan-seelmann.de

- Stefan Seelmann
- Freelancer
 - Software Development with Java
 - LDAP, Identity- and Access-Management
- Open Source
 - Apache Directory Studio



What is SWTBot?

- SWTBot is an open-source UI/functional tool for testing SWT and Eclipse based applications
- Founded by Ketan Padegaonkar
- Since December 2008 at Eclipse (Incubation)
- SWTBot API
 - Access the UI widgets
 - Make assertions
- JUnit tests – Writing Java Code
- No recording tool!

- Central classes:
 - SWTBot
 - SWTWorkbenchBot
- SWTBot methods: [widget type][condition]
 - `button(String)`
 - `textWithLabel(String)`
 - `viewByTitle(String)`
- Wrappers around SWT Widgets
 - SWTBotButton – `isEnabled()`, `click()`
 - SWTBotText – `setText()`
 - SWTBotView – `close()`, `bot()`

Demo: Setup Environment

- Update Site
 - <http://download.eclipse.org/technology/swtbot/galileo/dev-build/update-site>
- Development environment
 - UI Plug-in
- Target environment
 - Finder Plug-ins
 - SWT/JFace (Text, Combo, Button, ...)
 - Eclipse Workbench (Perspective, View, Editor, ...)

Demo: Setup a Test Plug-in

- Create a Plug-in Project
- Add dependencies
 - `org.eclipse.ui`
 - `org.eclipse.core.runtime`
 - `org.eclipse.swtbot.swt.finder`
 - `org.eclipse.swtbot.eclipse.finder`
 - `org.eclipse.swtbot.junit4_x`
 - `org.junit4`
 - `org.hamcrest`

Demo: Create a Test

- Create a JUnit Test Case
- Create an instance of SWT(Workbench)Bot
 - `SWTWorkbenchBot bot = new SWTWorkbenchBot()`
- Click around

```
bot.menu("File").menu("New").menu("Java Project").click();  
bot.textWithLabel("Project name:").setText("My Project");  
bot.button("Finish").click();
```
- Run the Test
 - As “SWTBot Test“

SWTBot Advanced

- Matcher to find UI widgets
 - Custom Matcher
- Threading
 - Access to UI widgets only in UI thread
 - SWTBot runs in a non-UI thread!
 - SWTBot takes care of it
 - `UIThreadRunnable.syncExec()`
- Screenshots
 - On failure when using `SWTBotJUnit4ClassRunner`
 - `SWTBot.captureScreenshot(String)`

Pitfalls and Recommendations

- Long-running operations
 - `sleep()`
 - [Wait-Conditions](#) (write to file)
 - Usually handled by SWTBot
- Boiler-Plate Code, DRY
 - [Page Object pattern](#), Helper Bots, DSL
- [Reset](#)
 - Reset perspective
 - Close dialogs
 - Initialize test data

Pitfalls and Recommendations

- Native dialogs (i.e. file dialog)

- Provide an input field



- Error dialogs

- `ErrorDialog.AUTOMATED_MODE = false;`

- I18n

- Use always English
- Use Ids: `SWTBot.textWithId()`

- Menu and Context Menu issues

- Tree (with decorations)

- Performance
 - Eclipse startup
 - Not really TDD
- Blocks the computer
 - Hands off!
 - Use Virtual Machine
- Integration in Build-System / CI
- Only one committer
 - But community is growing

- Works
- Easy to use
 - JUnit and Java
- Brings quality
 - Automated tests through all layers
 - [End-to-end tests](#)
- Free and Open Source

- Support more SWT widgets
 - Spinner, ProgressBar
- Support more UI toolkits
 - Eclipse Forms, GEF, RAP, eSWT
- DSLs

- SWTBot
 - <http://www.eclipse.org/swtbot/>
- Newsgroup
 - <news://news.eclipse.org/eclipse.swtbot>
- Page Object pattern
 - <http://code.google.com/p/webdriver/wiki/PageObjects>
- Apache Directory Studio:
 - <http://directory.apache.org/studio/>