Photon: Building an Electron-Shiny app using a simple RStudio addin

Developed by Columbus Collaboratory
Presented by Abbas Rizvi
Motivation

Problem:
• Many ways to deploy Shiny apps
• Some companies/organizations do not prefer cloud solutions

Our solution:
• Deploy Shiny App as standalone desktop application or executable
• Use Electron framework
Implementation

• Made changes to R portable
  • [https://sourceforge.net/projects/rportable/](https://sourceforge.net/projects/rportable/)

• Packaged R alongside Node.js and Chromium
  • [http://github.com/ColumbusCollaboratory/electron-quick-start](http://github.com/ColumbusCollaboratory/electron-quick-start)

• Created photon (R Package)
  • pulls “ColumbusCollaboratory/electron-quick-start” repository into local relative path
  • Built in RStudio addin with miniUI
  • Can build on Windows, macOS and Linux*
  • Conveniently build Electron portable comprising Shiny App
  • Shiny App can have package dependencies from CRAN, Bioconductor and GitHub

* Work in progress
Installation and Launch

Step 1: `remotes::install_github("ColumbusCollaboratory/photon")`

Step 2: `library(photon)`

Step 3: Click Tools

Step 4: Click Addins > Browse Addins
## Installation and Launch

**Step 5:** Click on photon in Addin box

<table>
<thead>
<tr>
<th>Package</th>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>colourpicker</td>
<td>Plot Colour Helper</td>
<td>Interactively pick colours to use in your plot.</td>
</tr>
<tr>
<td>covr</td>
<td>Calculate package test coverage</td>
<td>Calculates the package test coverage and opens a report, using <code>covr::report()</code></td>
</tr>
<tr>
<td>devtools</td>
<td>Report test coverage for a package</td>
<td>Calculate and report the test coverage for the current package, using <code>devtools::test_coverage()</code>.</td>
</tr>
<tr>
<td>devtools</td>
<td>Report test coverage for a file</td>
<td>Calculate and report test coverage for the current test file, using <code>devtools::test_coverage_file()</code>.</td>
</tr>
<tr>
<td>devtools</td>
<td>Run a test file</td>
<td>Run the current test file, using <code>devtools::test_file()</code></td>
</tr>
<tr>
<td>photon</td>
<td>photon</td>
<td>Launches Photon Electron App Builder</td>
</tr>
<tr>
<td>reprex</td>
<td>Render reprex...</td>
<td>Run <code>reprex::reprex()</code> to prepare a reproducible example for sharing.</td>
</tr>
<tr>
<td>reprex</td>
<td>Reprex selection</td>
<td>Prepare reprex from current selection</td>
</tr>
<tr>
<td>supplierAD</td>
<td>supplierAD</td>
<td>Launches Cardinal Health Supplier Anomaly Detection application</td>
</tr>
<tr>
<td>tfruns</td>
<td>Training Run</td>
<td>Execute a training run with the current source document</td>
</tr>
<tr>
<td>tfruns</td>
<td>View Latest Run</td>
<td>View the most recent training run</td>
</tr>
<tr>
<td>tfruns</td>
<td>View Run History</td>
<td>View all training runs</td>
</tr>
</tbody>
</table>

**Step 6:** Click execute

Step 6 invokes the function `photon::photon_rstudioaddin()` which launches the miniUI
Photon miniUI display
Step 1: Fill in text boxes (add packages that your Shiny app needs that are not in the electron-quick-start R portable; comma separated, no spaces)
Step 1: Fill in text boxes (add packages that your Shiny app needs that are not in the electron-quick-start R portable; comma separated, no spaces)

Step 2: Select directory with app.R in it
Step 1: Fill in text boxes (add packages that your Shiny app needs that are not in the electron-quick-start R portable; comma separated, no spaces)

Step 2: Select directory with app.R in it. Click Select when done
Step 1: Fill in text boxes (add packages that your Shiny app needs that are not in the electron-quick-start R portable; comma separated, no spaces)

Step 2: Select directory with app.R in it. Click Select when done

Step 3: Click Create job
Electron app build process

• The relative R instance will install R packages
• Electron packager is invoked
• Standalone application built

> electron-quick-start@1.0.0 package-mac /Users/arizvi/Desktop/photon-shiny-addin/electron-quick-start
> electron-packager --overwrite --platform=darwin --arch=x64 --out=ElectronShinyAppMac

Packaging app for platform darwin x64 using electron v5.0.2
Wrote new app to ElectronShinyAppMac/electron-quick-start-darwin-x64
What’s next?

• Address current limitations – extending to GitHub packages and Linux
• Decrease bulkiness (storage) of application builds
• Create an application manager tab that allows editing existing applications
Acknowledgements

Columbus Collaboratory Photon Developers
Pete Gordon
Slava Nikitin
Katie Sasso-Schafer
QUESTIONS?

THANK YOU!
QUESTIONS?

THANK YOU!