



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/>
- Packaged R alongside Node.js and Chromium
 - <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*

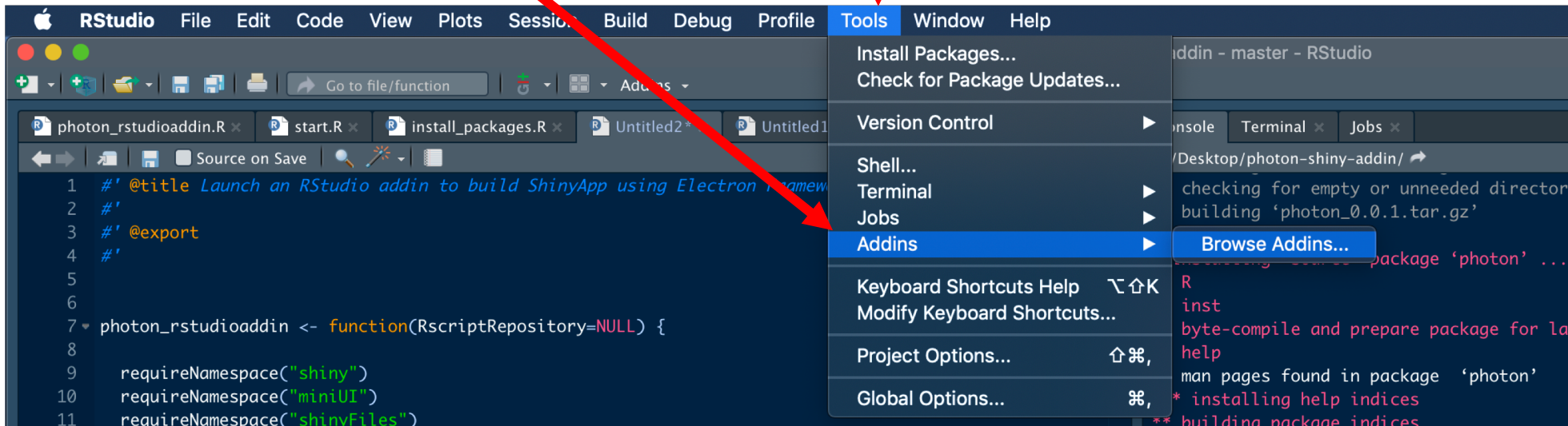
Installation and Launch

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

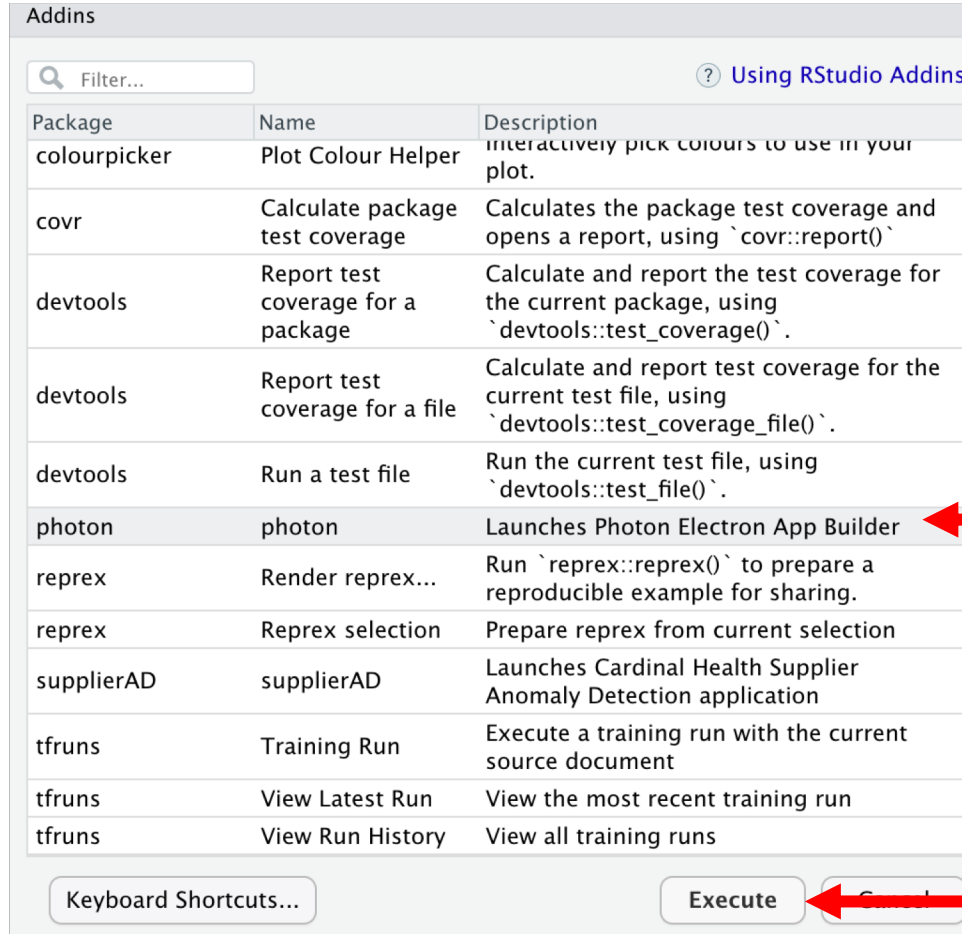
Step 2: `library(photon)`

Step 3: Click Tools

Step 4: Click Addins > Browse Addins



Installation and Launch



The screenshot shows the RStudio Addins panel. At the top, there is a search bar labeled "Filter..." and a link "? Using RStudio Addins". Below is a table with three columns: "Package", "Name", and "Description". The "photon" package is highlighted in grey. At the bottom of the panel, there are three buttons: "Keyboard Shortcuts...", "Execute", and "Cancel". A red arrow points from the text "Step 5: Click on photon in Addin box" to the "photon" row. Another red arrow points from the text "Step 6: Click execute" to the "Execute" button.

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

Step 5: Click on photon in Addin box

Step 6: Click execute

Step 6 invokes the function `photon::photon_rstudioaddin()` which launches the miniUI

Photon miniUI display

Photon Shiny App Builder

Cancel Use photon to build your shiny app Done

Choose your Shiny App directory

Select directory

Selected Rscript
Creation date: 2019-06-21

Additional arguments to Rscript

Rscript repository path: launch & log location


Job description
Runs a model to predict survival outcomes

CRAN packages
mgcv,matrixStats

GitHub packages
thomasp85/patchwork

Bioconductor packages
SummarizedExperimnt,VariantAnnotation

Create job

 Build standalone Shiny App for first time

Photon miniUI display

Photon Shiny App Builder

Cancel Use photon to build your shiny app Done

Choose your Shiny App directory

Select directory

Selected Rscript
Creation date: 2019-06-21

Additional arguments to Rscript

Rscript repository path: launch & log location

Job description
Runs a model to predict survival outcomes

CRAN packages
mgcv,matrixStats

GitHub packages
thomasp85/patchwork

Bioconductor packages
SummarizedExperiemnt,VariantAnnotation

Create job

Build standalone Shiny App for first time

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)

Photon miniUI display

The screenshot shows the Photon Shiny App Builder interface. At the top, there is a title bar "Photon Shiny App Builder" and a subtitle "Use photon to build your shiny app". Below the subtitle are "Cancel" and "Done" buttons. The main content area is titled "Choose your Shiny App directory" and contains a "Select directory" button. Below this, there are several sections for configuring the app:

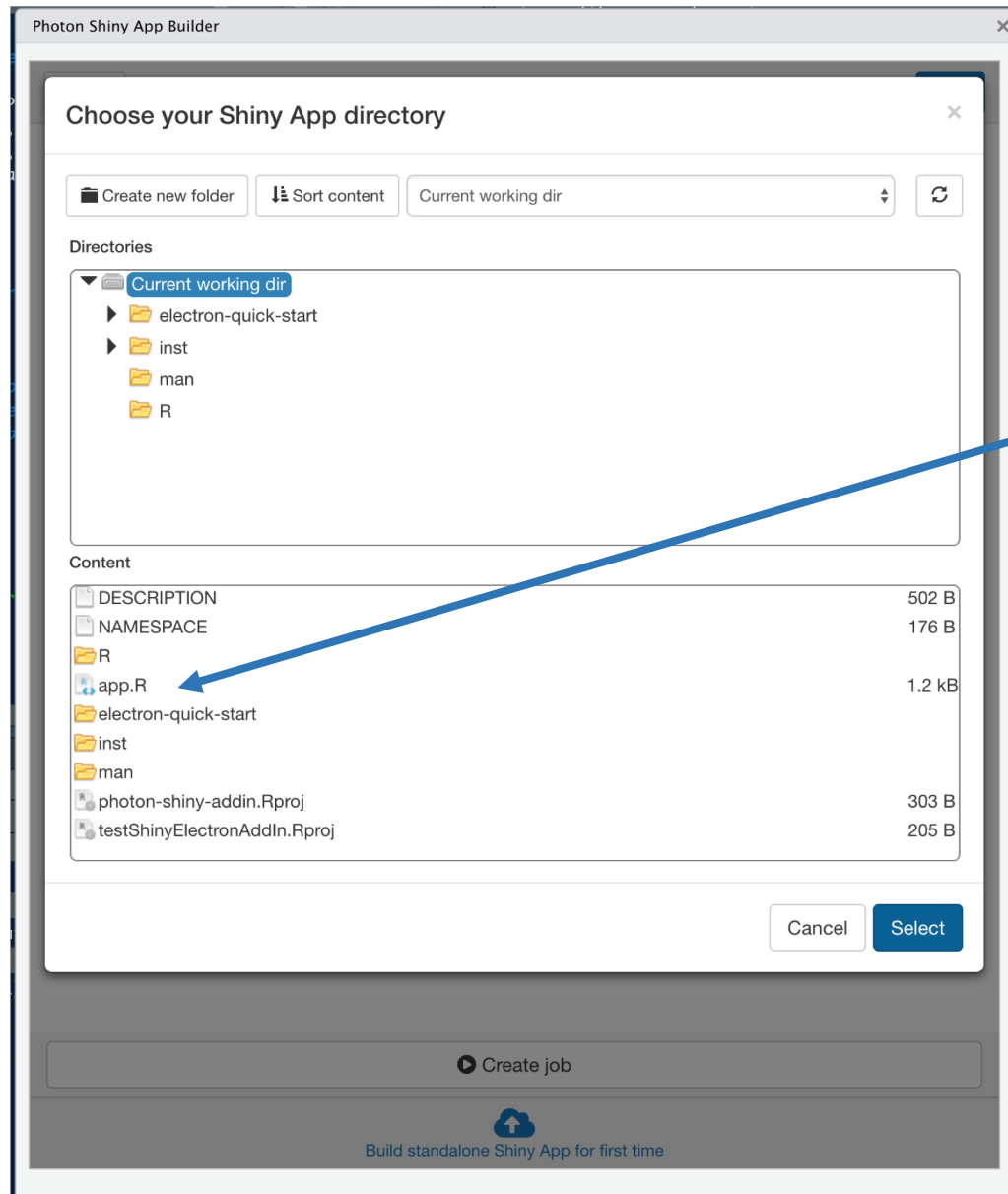
- Selected Rscript**: "Creation date:" with a text box containing "2019-06-21".
- Additional arguments to Rscript**: An empty text box.
- Rscript repository path: launch & log location**: An empty text box.
- Job description**: A text box containing "Runs a model to predict survival outcomes".
- CRAN packages**: A text box containing "mgcv,matrixStats".
- GitHub packages**: A text box containing "thomasp85/patchwork".
- Bioconductor packages**: A text box containing "SummarizedExperiemnt,VariantAnnotation".

At the bottom of the form, there is a "Create job" button. Below the form, there is a footer with a cloud icon and the text "Build standalone Shiny App for first time".

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

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

Photon miniUI display

The screenshot shows the 'Photon Shiny App Builder' window with the title 'Use photon to build your shiny app'. The interface includes a 'Cancel' button on the left and a 'Done' button on the right. The main content area is titled 'Choose your Shiny App directory' and contains several input fields:

- Select directory:** A button to choose the directory.
- Selected Rscript:** A text box containing the path `/Users/arizvi/Desktop/photon-shiny-adv`.
- Job description:** A text box containing the text 'Runs a model to predict survival outcomes'.
- Creation date:** A text box containing the date '2019-06-21'.
- CRAN packages:** A text box containing the packages 'mgcv,matrixStats'.
- Additional arguments to Rscript:** An empty text box.
- GitHub packages:** A text box containing the package 'thomasp85/patchwork'.
- Rscript repository path: launch & log location:** An empty text box.
- Bioconductor packages:** A text box containing the packages 'SummarizedExperiemnt,VariantAnnotation'.

At the bottom of the window, there is a 'Create job' button with a play icon, which is highlighted by a green rectangular box. Below this button, there is a link that says 'Build standalone Shiny App for first time'.

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!