RStudio Connect: User Guide

Version 1.5.10-6

Abstract

This guide will help a user configure and deploy content to RStudio Connect.

Contents

1 Introduction

RStudio Connect connects you and the work you do in R with others as never before. Only RStudio Connect provides:

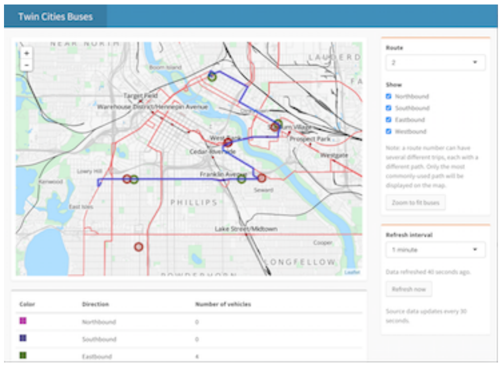
- "One button" deployment for any Shiny application, R Markdown document, or any static plot or graph to a single environment.
- The ability to manage and limit access to the work you've shared with others - and easily see the work they've shared with you.
- "Hands free" scheduling of updates to your documents and automatic email distribution.

For more information on running RStudio Connect in your organization please visit https://www.rstudio.com/products/connect/.

1.1 Shiny

Shiny is a web application framework for R that can help turn your analyses into interactive web applications. Get started by following the Shiny tutorial then visit the gallery for inspiration to use in your own dynamic visualizations.

RStudio Connect hosts your Shiny applications, allowing colleagues to interact with your data.



Route: 2

Show
- ☑ Northbound
- ☑ Southbound
- ☑ Eastbound
- ☑ Westbound

Note: a route number can have several different trips, each with a different path. Only the most commonly-used path will be displayed on the map.

Refresh interval: 1 minute

Data refreshed 40 seconds ago.

Source data updates every 30 seconds.

Color	Direction	Number of vehicles
🟪	Northbound	0
🟦	Southbound	0
🟩	Eastbound	4

Figure 1:

1.2 Plumber (Beta)

Note: Plumber APIs are currently in Beta.

Plumber is an R package that makes it easy to create API endpoints from annotated R code. For example:

```
## hello-world/plumber.R

#* @get /
helloEndpoint <- function() {
  "Hello, World!"
}
```

RStudio Connect hosts your APIs, controls access to your endpoints, and starts new instances when needed to handle high load.

1.2.1 Swagger Documentation

RStudio Connect makes it easy to share your API documentation using swagger. First, you will need Plumber 0.4.0 at least. See the Plumber website for instructions to install the latest version.

If you do not define a `GET /` handler, RStudio Connect will provide one that redirects to `GET /__swagger__`, a special link in Plumber that redirects to Swagger UI. To disable this, define a handler for `GET /`.

1.3 R Markdown

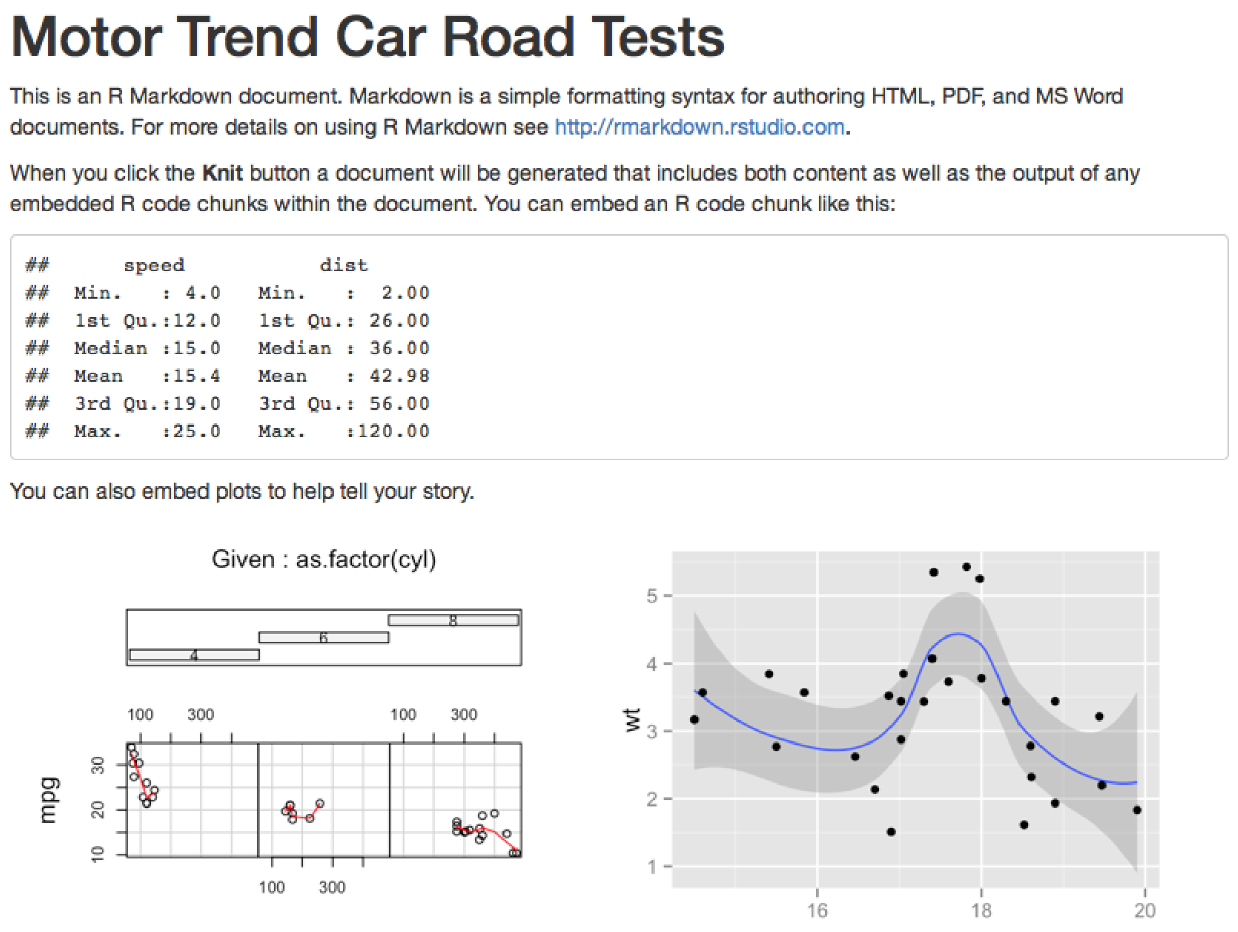
R Markdown is an authoring format that enables easy creation of dynamic documents, presentations, and reports from R. It combines the core markdown syntax with embedded R code chunks.

RStudio Connect helps you publicize and distribute your R Markdown documents. Content is published to the server, where colleagues can view your work. Documents can be periodically regenerated with the results distributed via email.

1.4 Plots and Graphs

R has very powerful plotting and graphing capabilities. With packages like ggplot2, it is easy to produce complex, multi-layered graphics.

RStudio Connect helps you share your plots and graphs by hosting that content.



Motor Trend Car Road Tests

This is an R Markdown document. Markdown is a simple formatting syntax for authoring HTML, PDF, and MS Word documents. For more details on using R Markdown see http://rmarkdown.rstudio.com.

When you click the **Knit** button a document will be generated that includes both content as well as the output of any embedded R code chunks within the document. You can embed an R code chunk like this:

```
##      speed           dist       
##  Min.   : 4.0   Min.   :  2.00  
##  1st Qu.:12.0   1st Qu.: 26.00  
##  Median :15.0   Median : 36.00  
##  Mean   :15.4   Mean   : 42.98  
##  3rd Qu.:19.0   3rd Qu.: 56.00  
##  Max.   :25.0   Max.   :120.00  
```

You can also embed plots to help tell your story.

Figure 2:

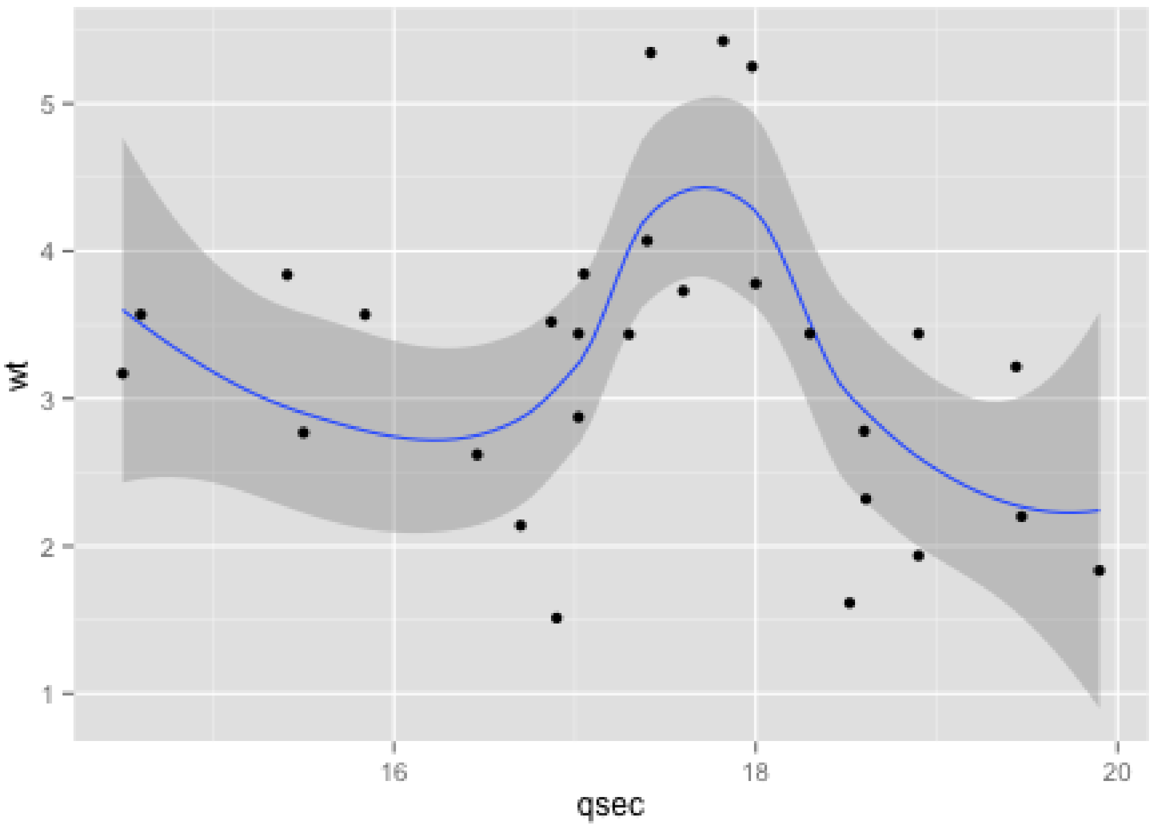


Figure 3:

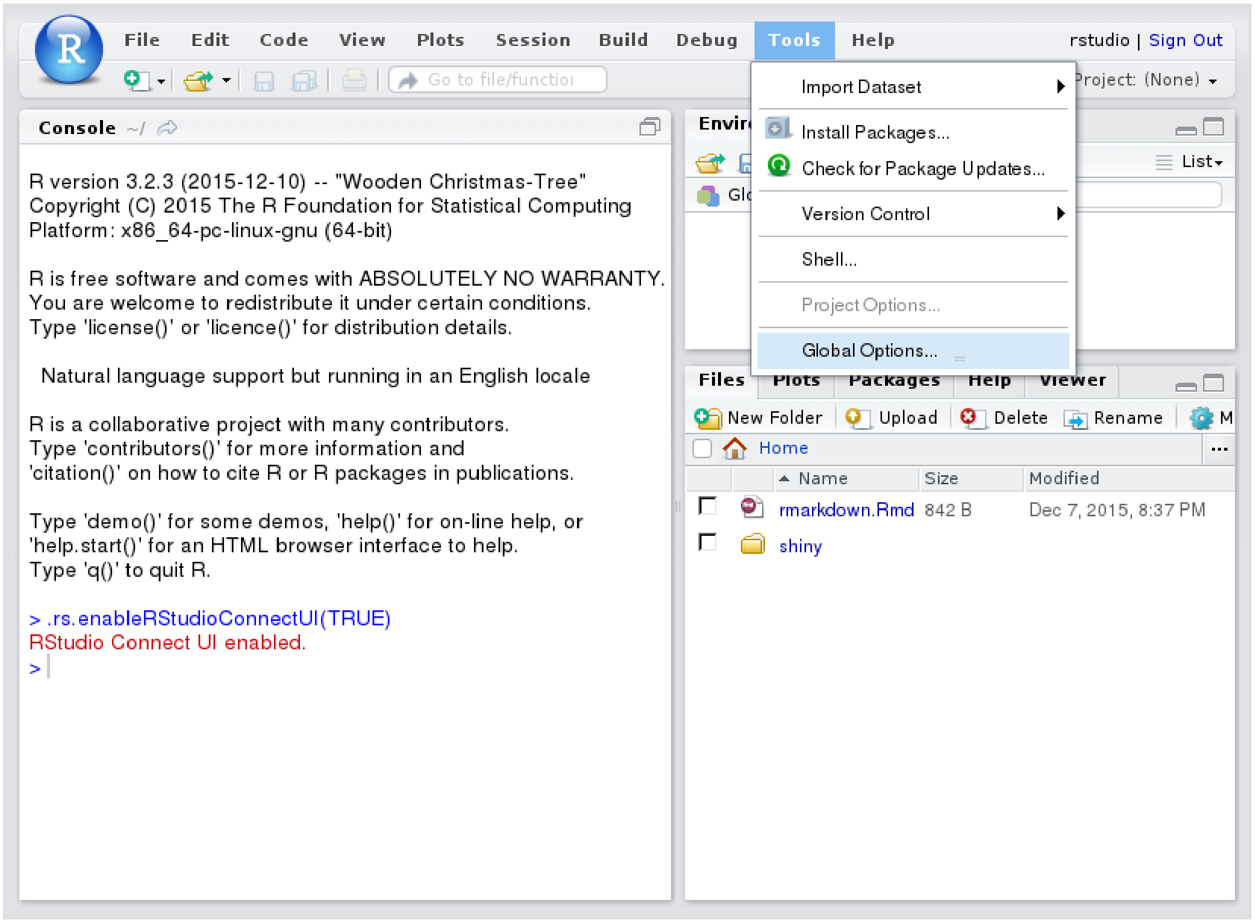
2 Connecting

If you have already connected your RStudio IDE, see Section 3 to learn how to publish content to RStudio Connect.

2.1 Update the IDE

At the time of this writing, you will need to install at least version 1.0.44 of the RStudio IDE to interact with Connect. You can confirm this by opening the IDE and clicking “Help” > “About RStudio” and checking the version number at the top of that pane. If you are not running at least version 1.0.44, you should download the latest release.

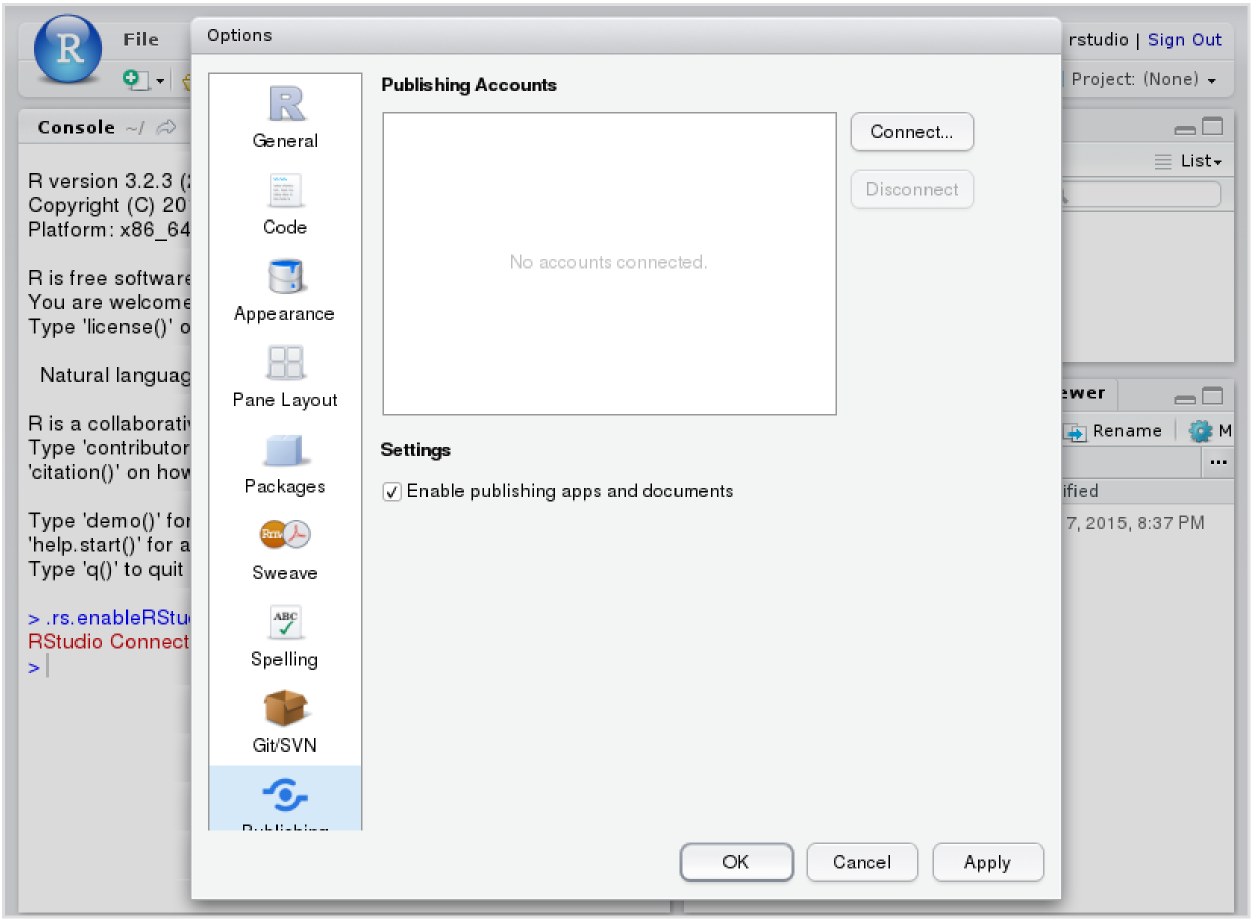
2.2 Connecting Your Account



```
R version 3.2.3 (2015-12-10) -- "Wooden Christmas-Tree"
Copyright (C) 2015 The R Foundation for Statistical Computing
Platform: x86_64-pc-linux-gnu (64-bit)

R is free software and comes with ABSOLUTELY NO WARRANTY.
You are welcome to redistribute it under certain conditions.
Type 'license()' or 'licence()' for distribution details.

  Natural language support but running in an English locale

R is a collaborative project with many contributors.
Type 'contributors()' for more information and
'citation()' on how to cite R or R packages in publications.

Type 'demo()' for some demos, 'help()' for on-line help, or
'help.start()' for an HTML browser interface to help.
Type 'q()' to quit R.

> .rs.enableRStudioConnectUI(TRUE)
RStudio Connect UI enabled.
>
```

Name	Size	Modified
rmarkdown.Rmd	842 B	Dec 7, 2015, 8:37 PM
shiny		

You’ll need to connect your IDE to Connect and authorize the IDE to use your Connect account. To add a publishing account to RStudio Connect, click **Tools** then **Global Options. . .** in the file menu.



Options

Publishing Accounts

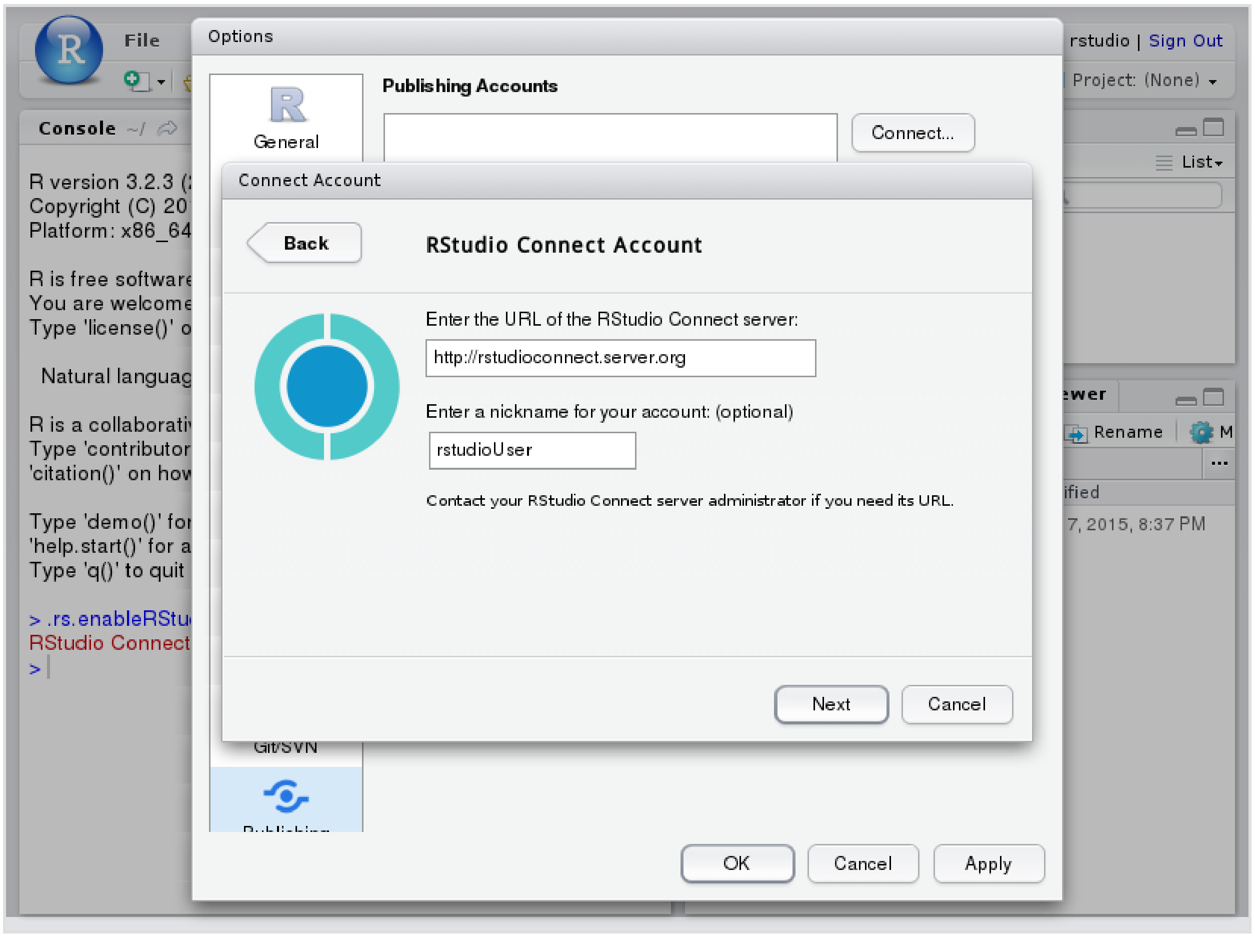
No accounts connected.

Settings

☑ Enable publishing apps and documents

```
R version 3.2.3 (
Copyright (C) 20
Platform: x86_64

R is free software
You are welcome
Type 'license()' o

  Natural languag

R is a collaborati
Type 'contributor
'citation()' on how

Type 'demo()' for
'help.start()' for a
Type 'q()' to quit

> .rs.enableRStu
RStudio Connect
>
```

This will open up the options menu, at which point you can click the **Publishing** icon on the sidebar, and then the **Connect. . .** button to create a new account.



RStudio Connect Account

Enter the URL of the RStudio Connect server: http://rstudioconnect.server.org

Enter a nickname for your account: (optional) rstudioUser

Contact your RStudio Connect server administrator if you need its URL.

Select that you want to configure an RStudio Connect account then enter the address of your server. If you are using an RStudio-provided URL such as “https://beta.rstudioconnect.com”, enter it here. Otherwise, your Connect administrator can provide you with the address of the RStudio Connect server.

The optional nickname is helpful when managing multiple Connect accounts.

Clicking **Next** will open up a login window for RStudio Connect.

Log into Connect and authorize the IDE to deploy on your behalf.

You have successfully configured the RStudio IDE and are ready to publish content to RStudio Connect! See Section 3.

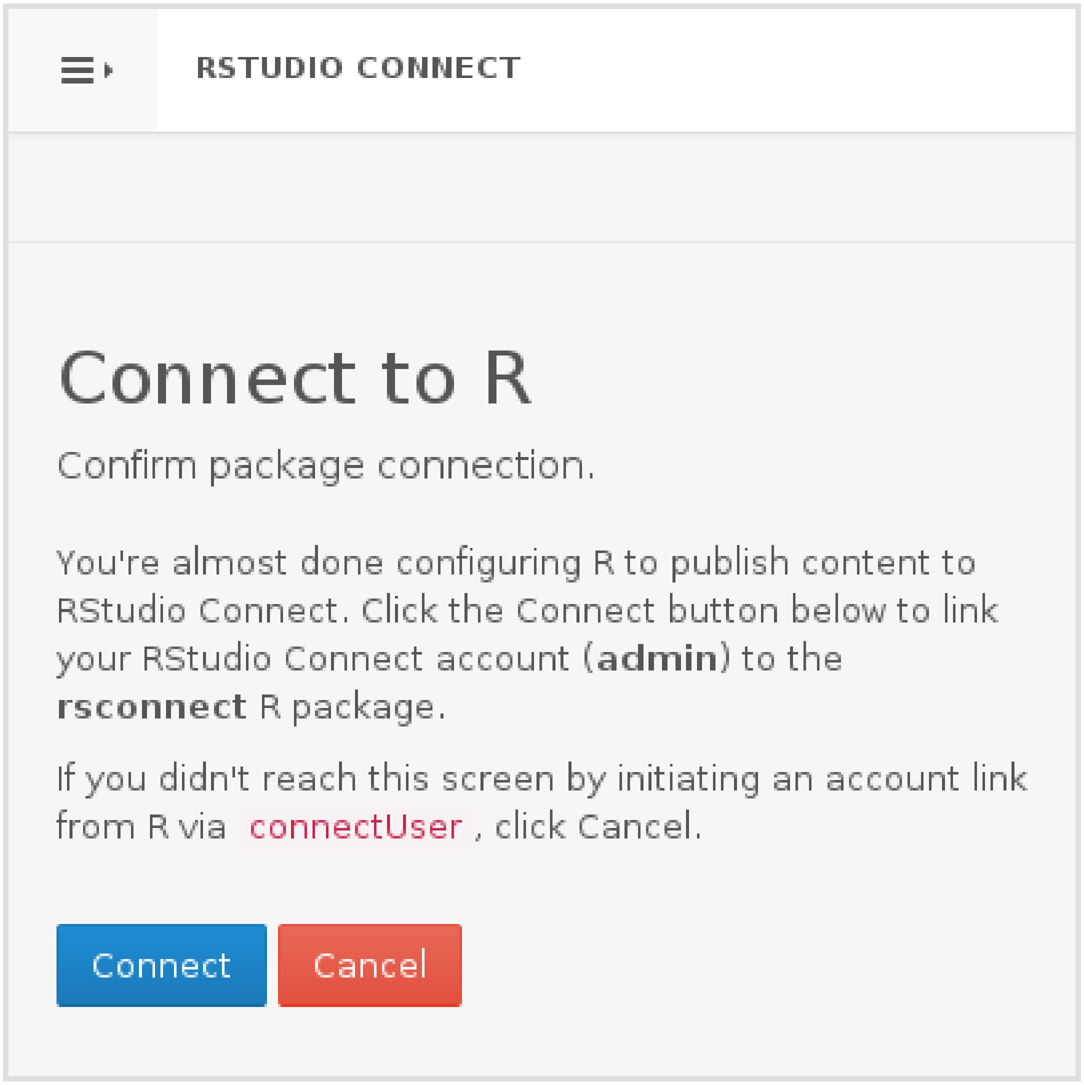
3 Publishing

Be sure to configure your Connect account before attempting to publish with the RStudio IDE. See Section 2 for information on configuring your Connect account

3.1 Publishing Instructions for All Content Types

RStudio Connect accepts Shiny applications, R Markdown documents, as well as plots and graphs. The blue **publishing icon** in the RStudio IDE indicates that a particular piece of content can be published.

You can find the blue **publishing icon** at the following locations:


RSTUDIO CONNECT

Connect to R

Confirm package connection.

You're almost done configuring R to publish content to RStudio Connect. Click the Connect button below to link your RStudio Connect account (**admin**) to the **rsconnect** R package.

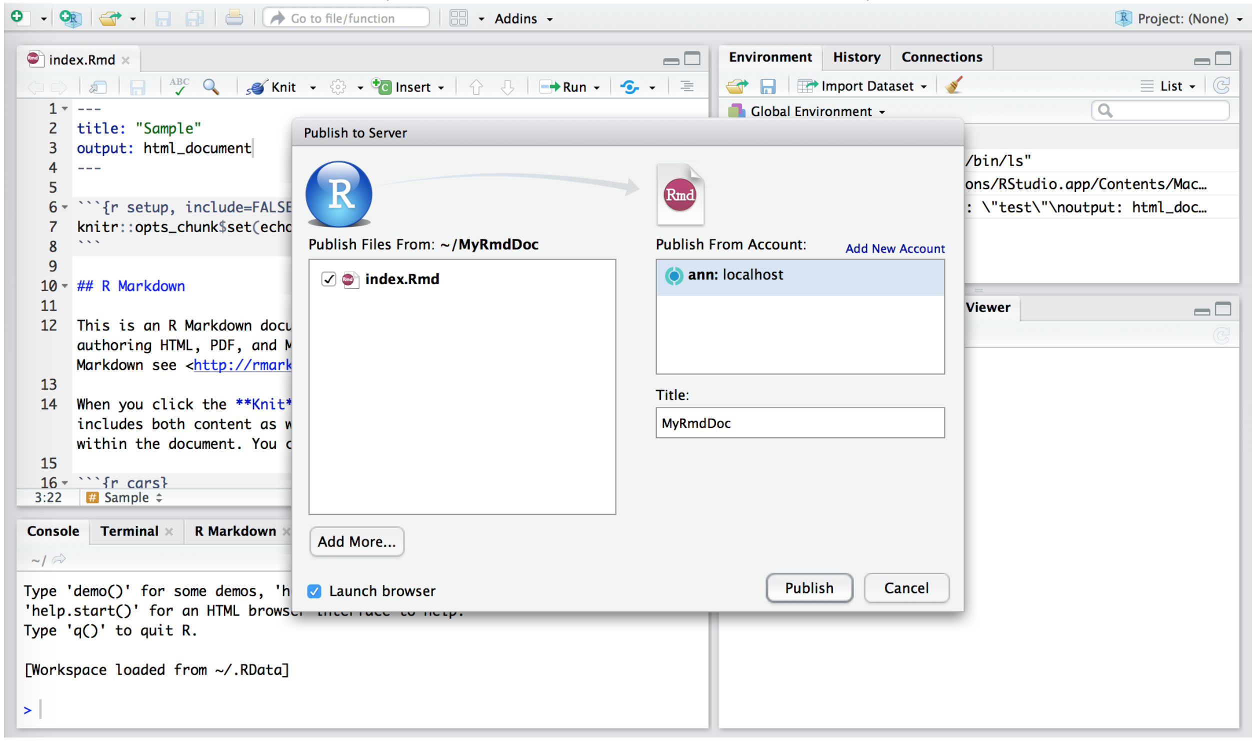
If you didn't reach this screen by initiating an account link from R via `connectUser`, click Cancel.

Figure 4:

Figure 5:

- The upper right of the file editor
- The document viewer when viewing a document
- The embedded viewer when running a Shiny application
- The plots pane

Click on this icon to open a publishing dialog where you can name your content and select additional files to include in the deployment. By default, RStudio will try to infer the data files and scripts are used in your application. This window lets you refine those file selections.



~~~
---
title: "Sample"
output: html_document
---

```{r setup, include=FALSE
knitr::opts_chunk$set(echo
~~~

R Markdown

This is an R Markdown docu
authoring HTML, PDF, and M
Markdown see <http://rmark

When you click the **Knit*
includes both content as w
within the document. You c

```{r cars}
```

```
/bin/ls"
ons/RStudio.app/Contents/Mac…
: \"test\"\noutput: html_doc…
```

```
~/
Type 'demo()' for some demos, 'h
'help.start()' for an HTML browser interface to help.
Type 'q()' to quit R.

[Workspace loaded from ~/.RData]

>
```

Publish Files From: ~/MyRmdDoc

☑ index.Rmd

Publish From Account: ann: localhost

Title: MyRmdDoc

☑ Launch browser

Most of the time, RStudio is able to determine automatically which files are needed to render your document on Connect. However, there are situations in which it will miss a file (for instance, if it isn't referenced directly in your document). The **Add More...** button lets you add files to the bundle that will be sent to Connect so that they will be available on the server when your document is rendered. You can also use the resource_files field in your document's YAML header to add additional files.

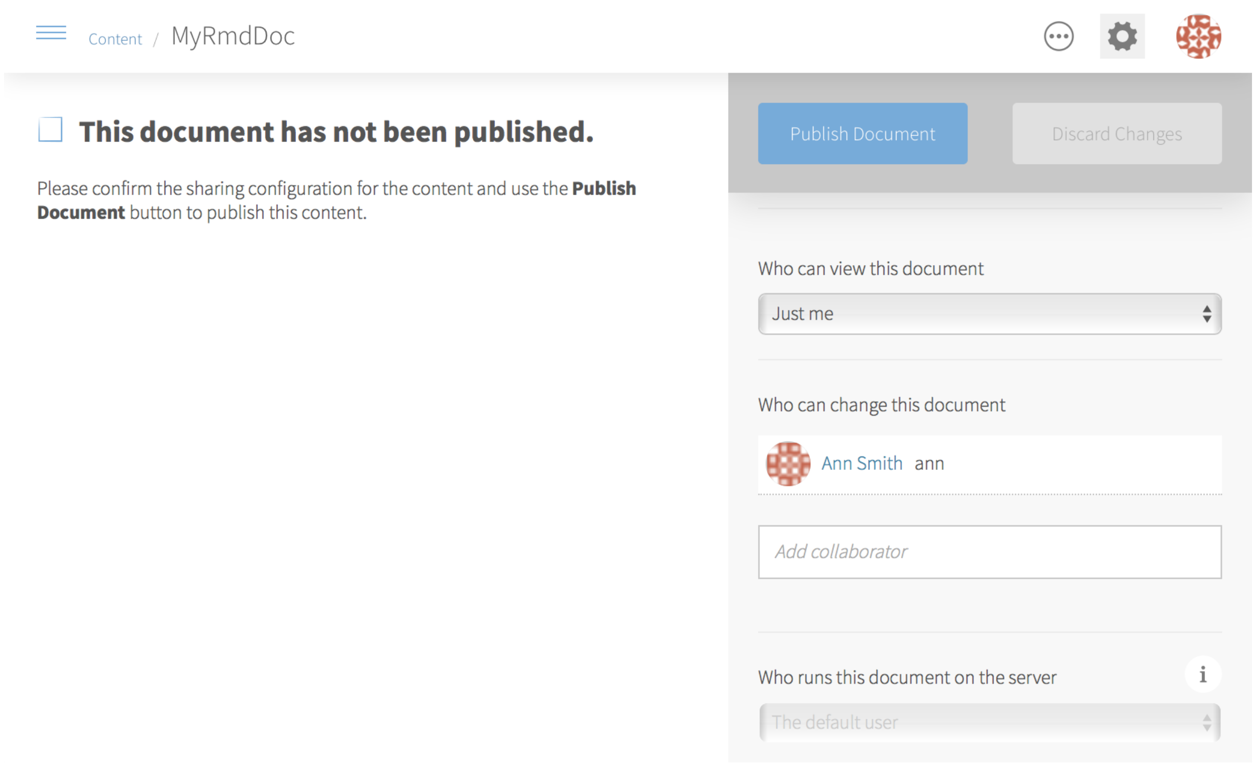
Deployed data files must be in the same directory as your Shiny application or R Markdown document. Files stored elsewhere on your computer (`C:\Users\me\mydata.csv`) will not be available on the remote server.

Click **Publish** after verifying your settings.

Your first deployment may take a few minutes, as Connect attempts to recreate the R library you use locally – referenced packages are downloaded and installed. These packages are cached on the server; subsequent deployments will be faster.

Not all of your IDE environment can be replicated on the server. Different operating systems or versions of R can occasionally make applications behave differently. Package installation failures may require the installation of additional system libraries.

When the deployment completes, you'll be taken to the application's settings page in RStudio Connect.



This document has not been published.

Please confirm the sharing configuration for the content and use the **Publish Document** button to publish this content.

Who can view this document: Just me

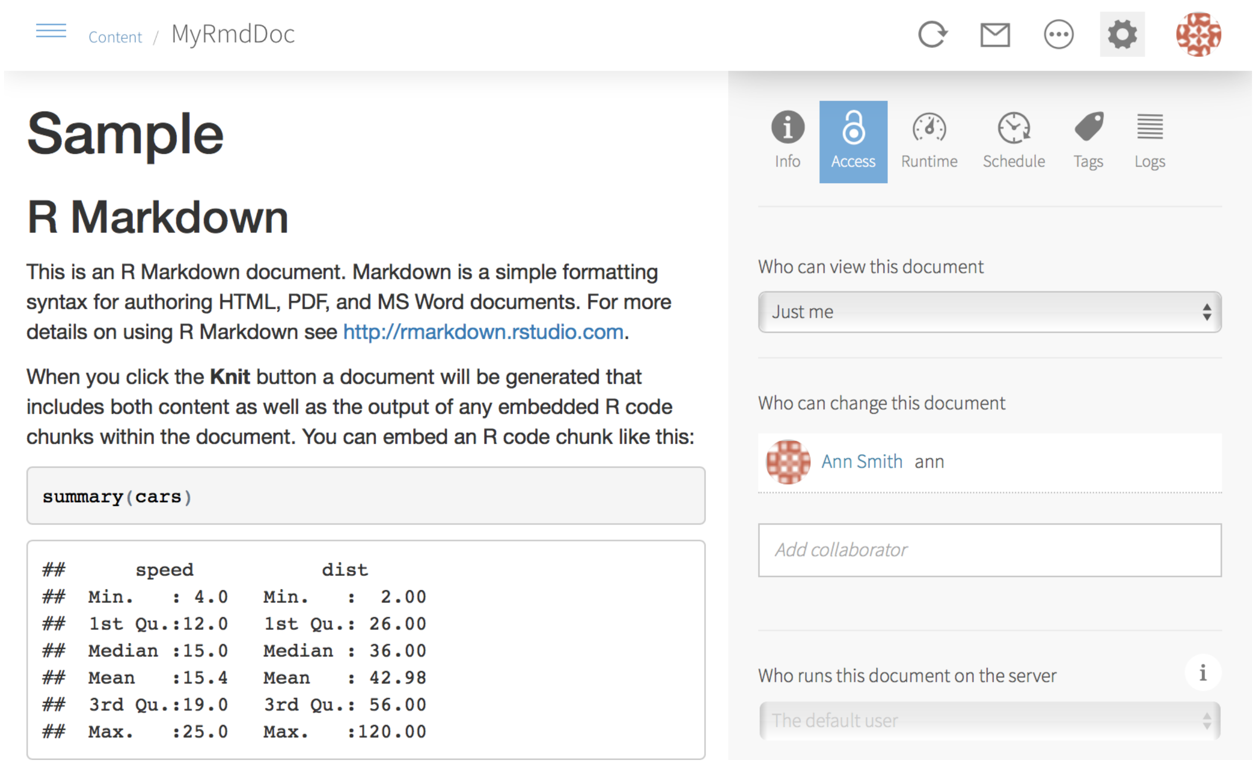
Who can change this document: Ann Smith ann

Add collaborator

Who runs this document on the server: The default user

This page lets you verify the sharing and visibility of your content. Click **Publish** to confirm these settings.

Content cannot be viewed until it is published.



Sample

R Markdown

This is an R Markdown document. Markdown is a simple formatting syntax for authoring HTML, PDF, and MS Word documents. For more details on using R Markdown see http://rmarkdown.rstudio.com.

When you click the **Knit** button a document will be generated that includes both content as well as the output of any embedded R code chunks within the document. You can embed an R code chunk like this:

```
summary(cars)
```

```
##      speed           dist       
##  Min.   : 4.0   Min.   :  2.00  
##  1st Qu.:12.0   1st Qu.: 26.00  
##  Median :15.0   Median : 36.00  
##  Mean   :15.4   Mean   : 42.98  
##  3rd Qu.:19.0   3rd Qu.: 56.00  
##  Max.   :25.0   Max.   :120.00  
```

Who can view this document: Just me

Who can change this document: Ann Smith ann

Add collaborator

Who runs this document on the server: The default user

You should now see your deployed content – a rendered version of the document or a live instance of your Shiny application. The content is displayed within the context of RStudio Connect, and you are able to further configure settings for your content.

3.1.1 Publishing Plumber APIs

Plumber APIs have the following known restrictions:

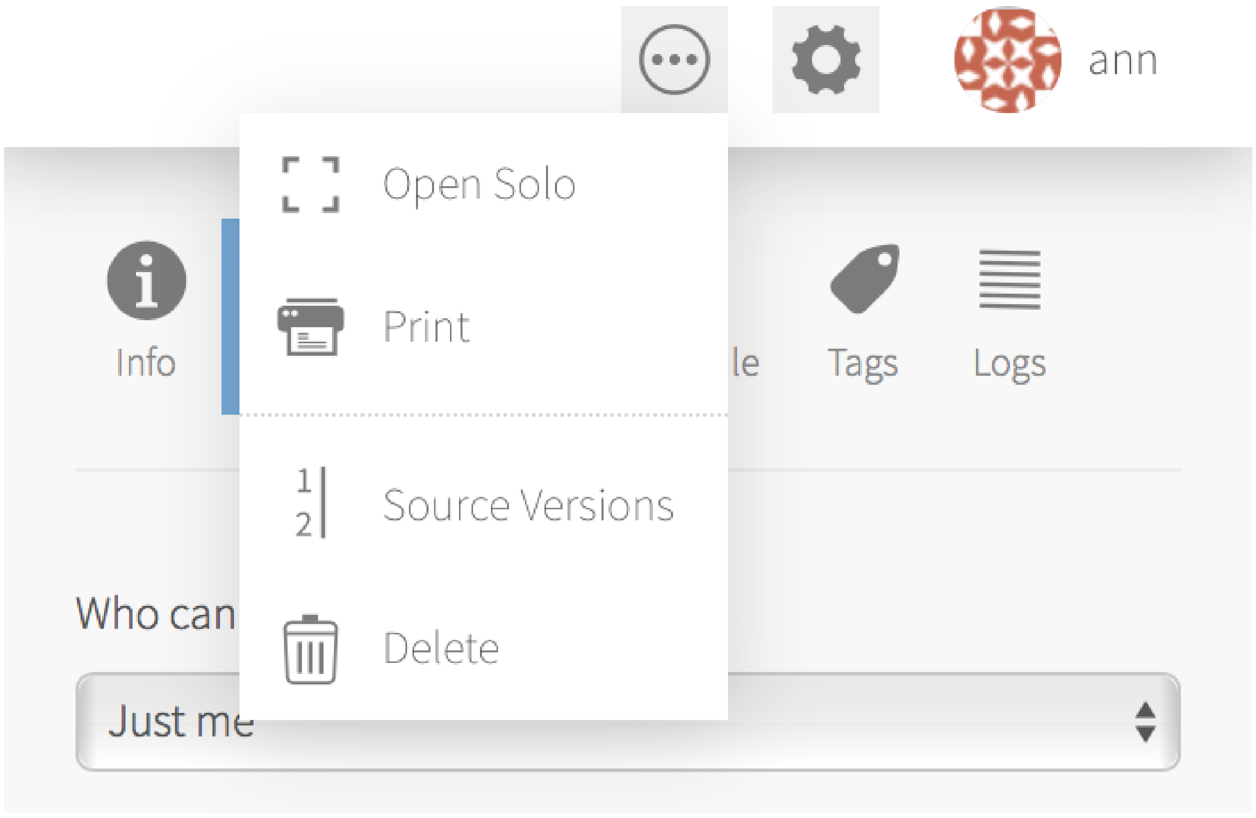
- Push-button publishing from the IDE is not supported
- Server-side latency is not tracked

To get started with publishing Plumber API endpoints, create a directory with a `plumber.R` file defining your endpoints. From the console, execute the following, replacing `<project-dir>` with your project's directory:

```
> rsconnect::deployAPI(api = '<project-dir>')
```

Once live, the content view will show you the results of your `@get /` endpoint. If `@get /` is not defined, you will see a "Swagger UI" presentation of your API. Swagger is an API documentation format; Plumber can automatically generate a `swagger.json` definition for your API by inspecting your endpoints. The "Swagger UI" page offers an interactive portal into your API that allows you to test your endpoints directly from the browser.

To make a call to your new API yourself, you're going to need the target URL. You can find it by looking at the bottom of the `Access` tab in your API's settings, or by clicking the `...` menu in the upper-right of the content view and select "Open Solo".



The location should look like the following:

```
https://example.com/content/42/
```

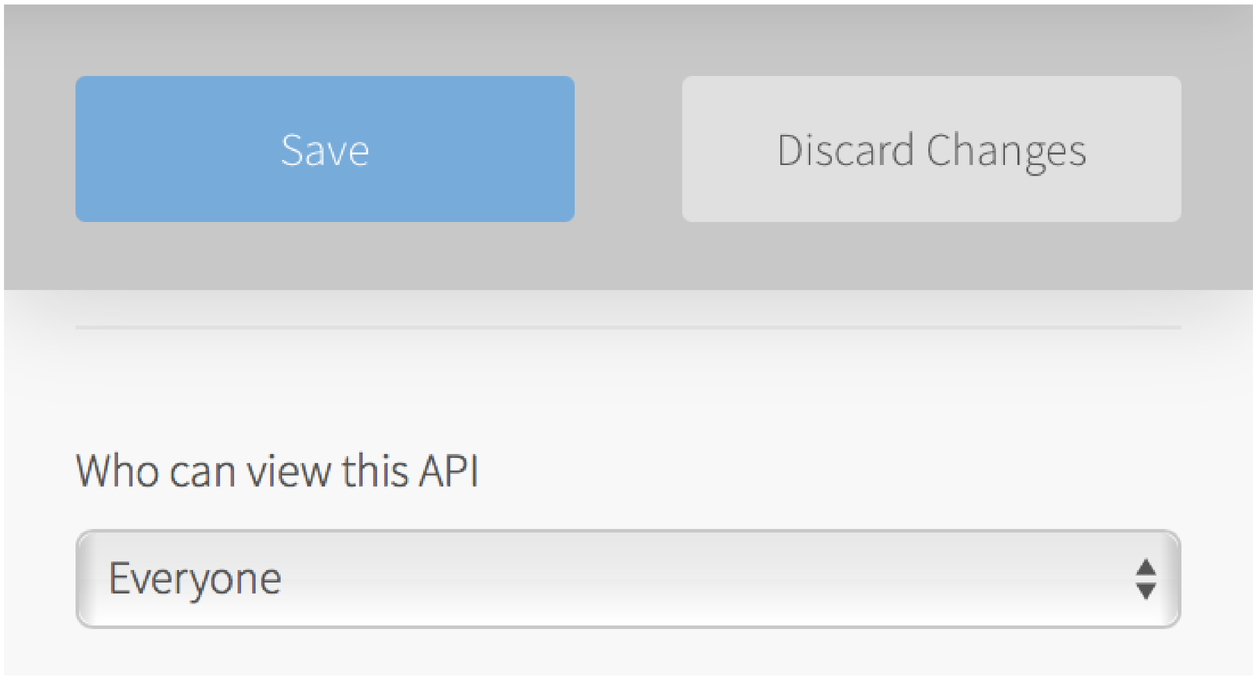
All calls can be made relative to this path. For example, if you want to make a request to `/volcano`, the location in the above example would be `https://example.com/content/42/volcano`, like so:

```
curl -XPOST https://example.com/content/42/volcano --data-binary '{ "line_plot": true }'
```

If your API restricts access to a particular set of users, then Connect will require that incoming requests authenticate the user making the request. The best approach for authenticating a request targeting an API is to use API Keys 4.

Alternatively, if your server is configured to use proxy authentication, you should ask your IT Administrator about ways to make API calls through that proxy.

If you want to perform access control in your Plumber API itself, or if you want to allow anyone to access your API, open the application settings, then the "Access" tab, and set "Who can view this API" to "Everyone".



Who can view this API: Everyone

A simple way to control access might be like the following:

```
#* @filter shared-secret-auth
keyAuth <- function(res, secret = NULL) {
    if(is.null(secret) || secret != 'mySecret') {
        res$status <- 401 # 401 UNAUTHORIZED
        list(error = "Authentication Failed")
    } else {
        plumber::forward()
    }
}
```

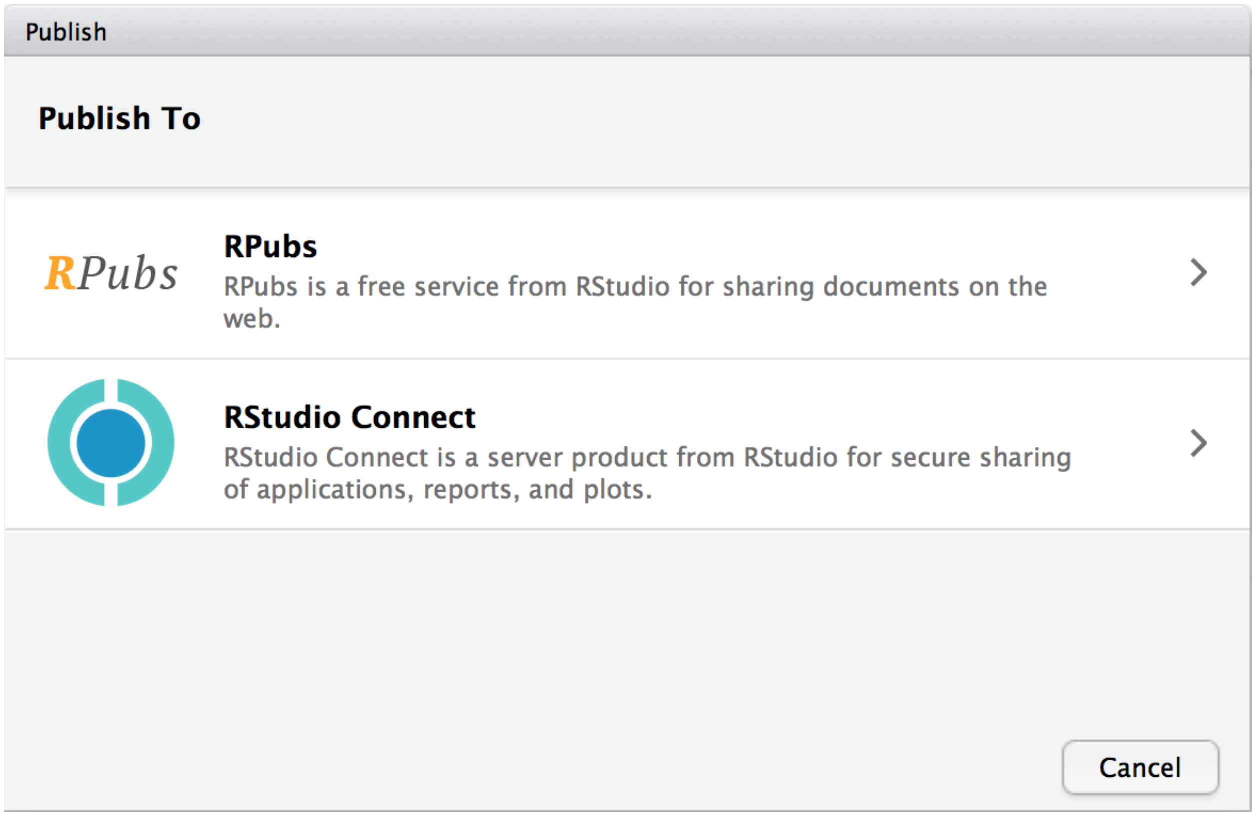
See the Plumber documentation for more information on `@filter` methods, and how they can be skipped in your route with `@preempt`.

Security is hard. The example above might be good enough for some purposes, but is unsuitable in cases where you need multiple keys you can invalidate arbitrarily. In those cases, API Keys 4 would be preferable. Ask your IT Administrator for guidance if you need help choosing a suitable authentication scheme.

3.2 Publishing Options for Documents

When publishing documents to RStudio Connect, you may encounter other deployment options depending on your content. These options are discussed below.

3.2.1 Publishing Destination



RPubs is a free service from RStudio for sharing documents on the web.

RStudio Connect is a server product from RStudio for secure sharing of applications, reports, and plots.

RPubs is a service for easily sharing R Markdown documents. RPubs is not related to Connect, and you should always choose "RStudio Connect" if you wish to publish your content to Connect.

RPubs documents are (1) always public, (2) always self-contained, and (3) and cannot contain any Shiny content. You will see the choice to publish to RPubs if your document is self-contained and does not require Shiny. Some organizations want to prohibit publishing to RPubs to reduce the chance that sensitive data will be accidentally made public; publishing to RPubs (and shinyapps.io) can be disabled if desired using an RStudio Server option.

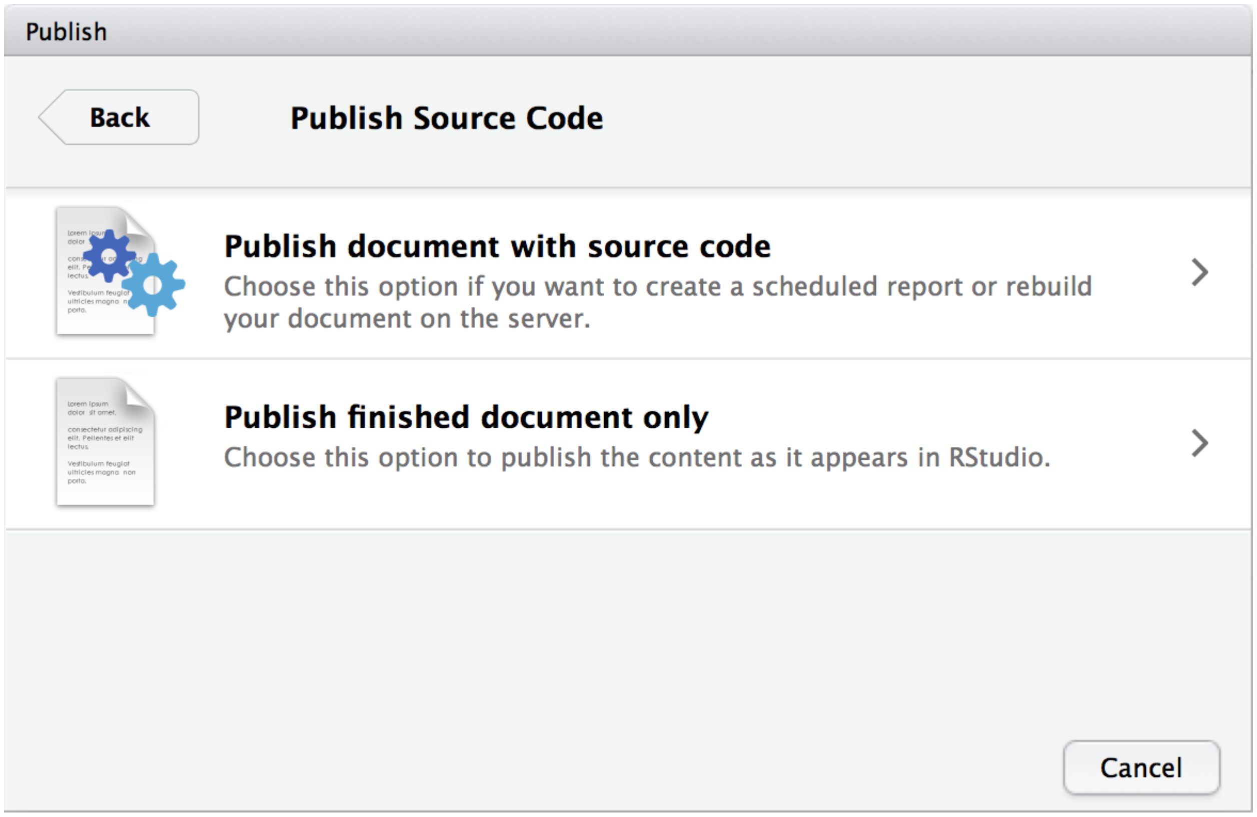
3.2.2 Publish Source Code

You will see these options when publishing from the document viewer.

Publishing the document with source code means that your R Markdown file (`.Rmd`) will be deployed to RStudio Connect. This file is rendered (usually to HTML) on the server.

Publishing only the finished document means that the HTML file you rendered locally is deployed to RStudio Connect.

We recommend publishing your documents with source code, as it allows you to re-render the document with RStudio Connect (on a weekly schedule, for example). If the document cannot be rendered by RStudio



Publish

Back

Publish Source Code

Publish document with source code

Choose this option if you want to create a scheduled report or rebuild your document on the server.

Publish finished document only

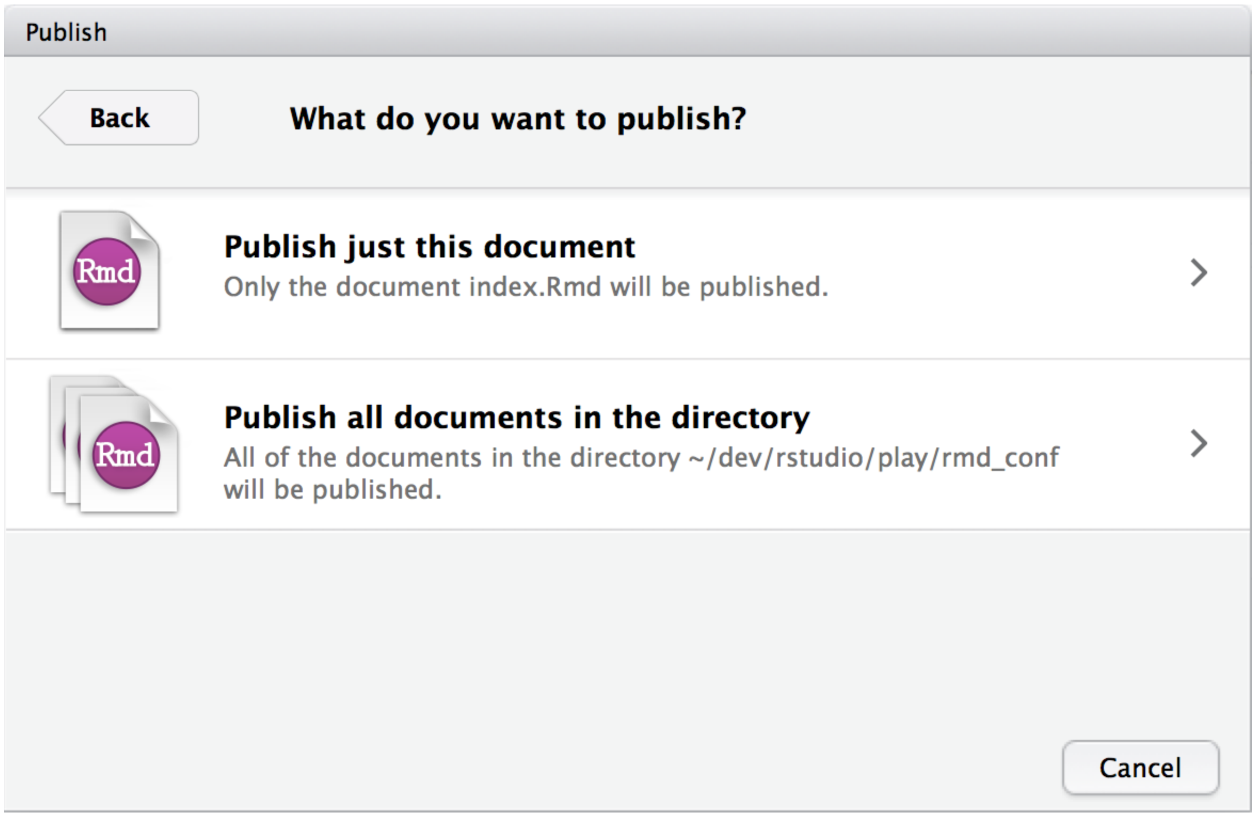
Choose this option to publish the content as it appears in RStudio.

Cancel

Figure 6: publish source code?

Connect because of files or data sources that are unavailable on the server, choose "Publish finished document only" so others can view your work.

3.2.3 Document Selection


Publish
Back
What do you want to publish?
Publish just this document
Only the document index.Rmd will be published.
Publish all documents in the directory
All of the documents in the directory ~/dev/rstudio/play/rmd_conf will be published.
Cancel

You will see these options when publishing an R Markdown document from a directory that contains more than one R Markdown document. It is possible to link together multiple R Markdown documents to make a multi-page document, so this is your chance to indicate that you've done this, and to publish all the documents at once. In most cases however you'll want to publish just the current document.

4 API Keys

API Keys allow you to programmatically access content on RStudio Connect. They are a substitute for logging in to RStudio Connect to access content.

API Keys are not associated with any content, and can be used to access any content that the owning user could access while logged in. The best practice is to create a new API Key for each external tool that needs access to content hosted on RStudio Connect, and to give each Key a name that helps you identify the tool that uses it. When the tool no longer needs access, or if you cannot trust the Key at any time, you can quickly revoke access by deleting the Key.

To access content with an API Key you must provide a HTTP header whose key is `Authentication` and value is `Key THE_API_KEY`.

All requests to content must be made to the target URL of the published content. You can find the target URL by clicking the ... menu in the upper-right of the content view and select "Open Solo".

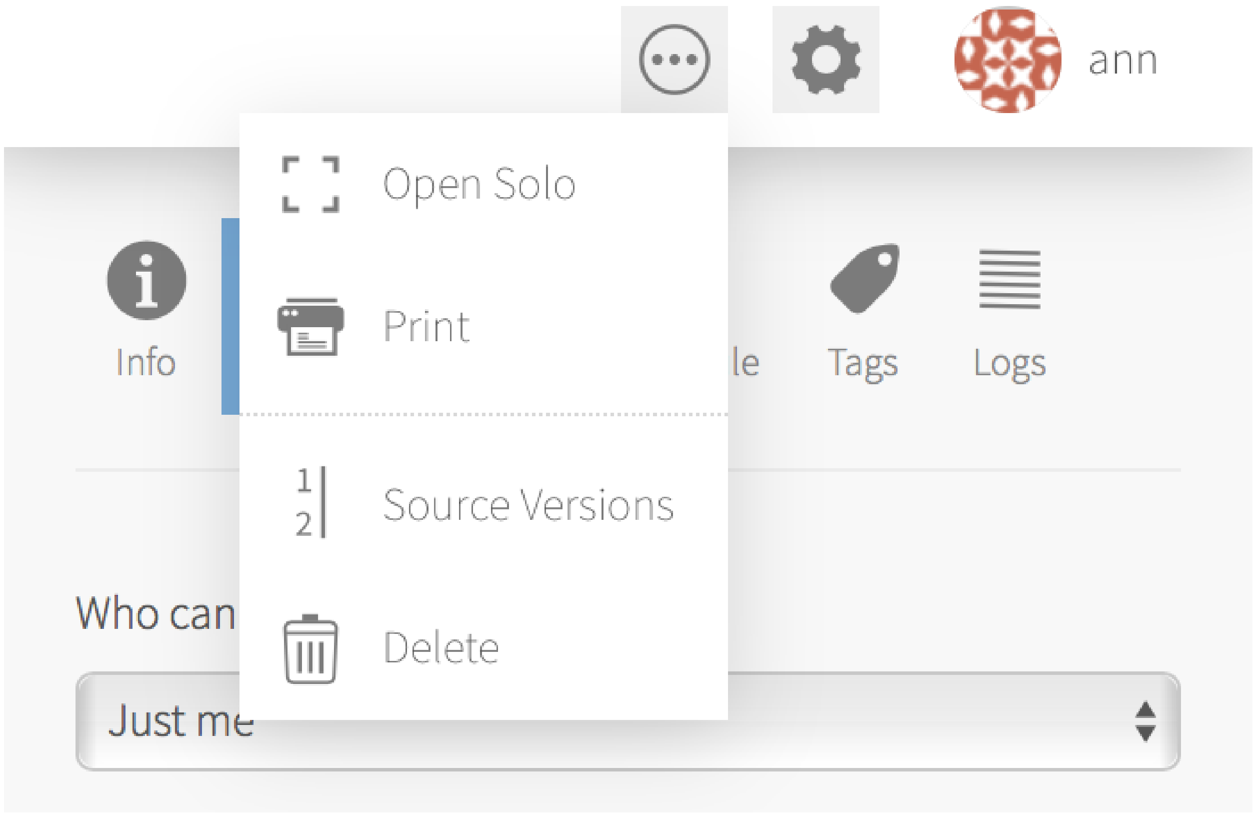


Figure 7: A screenshot of the menu with “Open Solo” displayed.

4.1 Examples

4.1.1 Creating and Deleting an API Key

To create an API key, click on the circular picture in the top-right portion of the screen. The picture may have your username next to it if you are viewing RStudio Connect on a large screen. Click the "API Keys" item in the menu that appears, and you should be taken to the API Keys page.

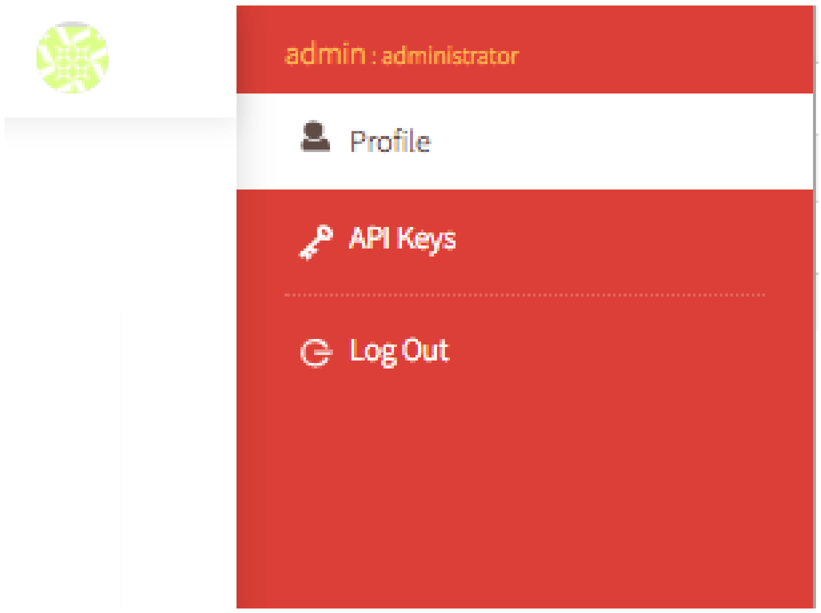
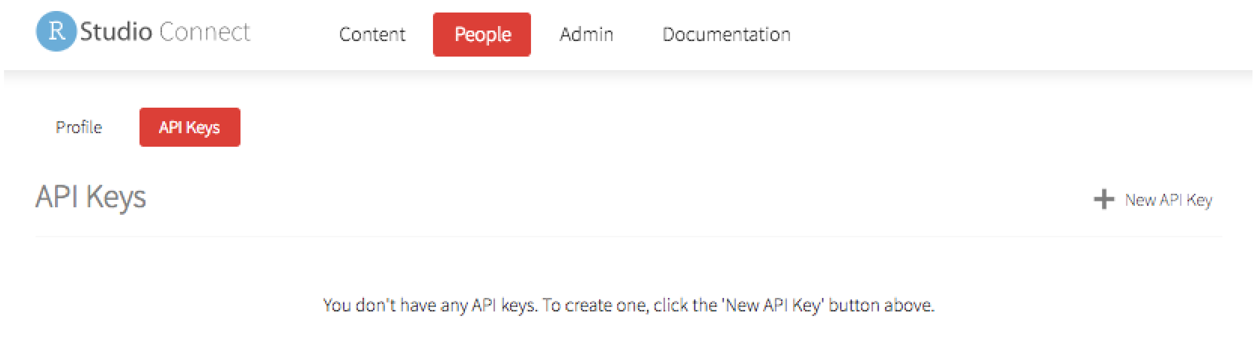


Figure 8: A screenshot of the user menu with "Profile, API Keys, Logout" items



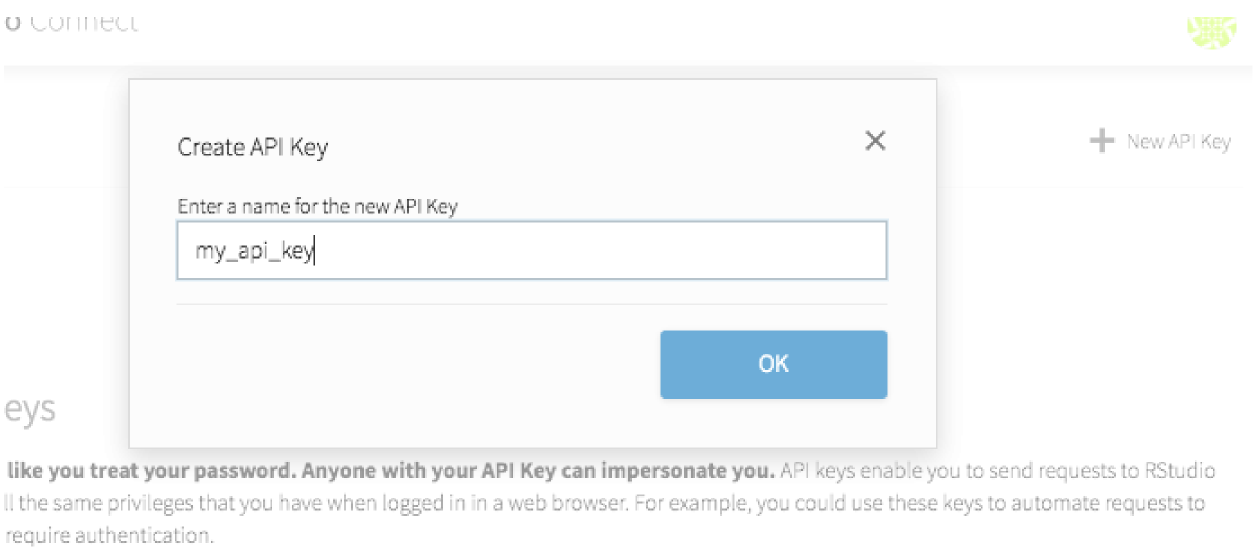
You don't have any API keys. To create one, click the 'New API Key' button above.

Figure 9: A screenshot of the API Keys page showing no API keys created

On the API Keys page, click "New API Key". You will be prompted to name your API key.

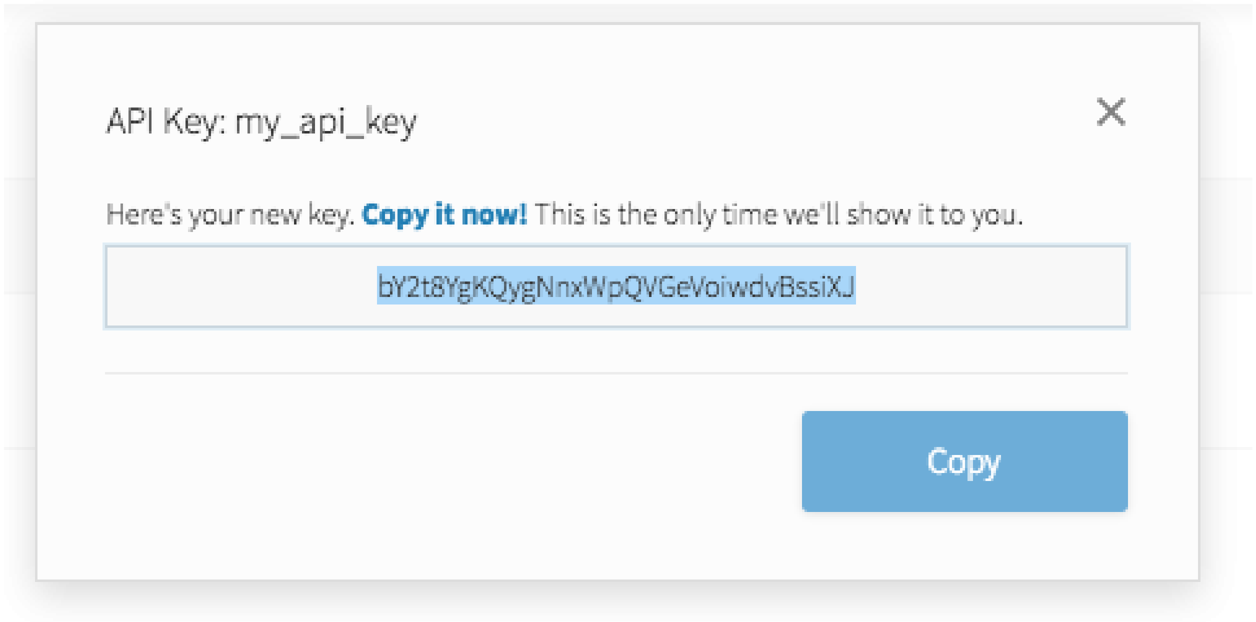
After creating an API key, you will be shown the key and given an opportunity to put it in a safe location. Once you close the dialog, you will NOT be shown the API key again. This helps to keep your user account safe.

If you have lost an API key, or if you simply don't need to use the API key anymore, remove the API key by clicking the trash bin icon on the far right column of the API key list.



like you treat your password. Anyone with your API Key can impersonate you. API keys enable you to send requests to RStudio
ll the same privileges that you have when logged in in a web browser. For example, you could use these keys to automate requests to
require authentication.

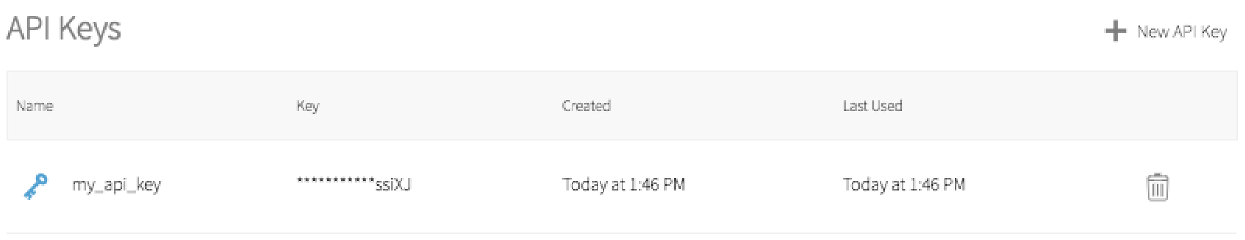
Figure 10: A screenshot of a dialog titled “Create API Key” with the name “my_api_key” entered



Here's your new key. **Copy it now!** This is the only time we'll show it to you.

bY2t8YgKQygNnxWpQVGeVoiwdvBssiXJ

Figure 11: A screenshot showing a created API key



Name	Key	Created	Last Used
my_api_key	***********ssiXJ	Today at 1:46 PM	Today at 1:46 PM

Figure 12: A screenshot of the API key list showing “my_api_key” and a trash bin icon

When you click the trash bin icon, a dialog will ask you to confirm that you want to delete the API key. Successfully deleting the API key will show a green status message at the top of the screen.

4.1.2 Static Content

Suppose you have published a plot whose target URL is `http://example.com/content/24/target.html`

You can download the content with an API Key via:

```
curl -o output.html -H 'Authorization: Key YOUR_API_KEY_HERE' \
    'http://example.com/content/24/target.html'
```

4.1.3 Plumber

Suppose you have published the following Plumber application whose target URL is `http://example.com/content/42`

```
## plumber.R

#* @get /mean
normalMean <- function(samples=10){
  data <- rnorm(samples)
  mean(data)
}
```

The following calls the function at the `/mean` endpoint with an API Key:

```
curl -H 'Authorization: Key YOUR_API_KEY_HERE' \
    'http://example.com/content/24/mean?samples=5'
```