PRO TIPS

Creating a Custom Dashboard with ‘`connectwidgets`’ and RStudio Connect

Custom landing page with `connectwidgets`

Authored By Ashley Henry - Technical Writer

When you want to share deployed content as a cohesive project

- How do you make sure your audience finds what they need on Posit Connect, *formerly RStudio Connect*, without having to page through the dashboard, remember the right search terms, or bookmark every content item you share?
- How do you share related deployments as a cohesive project?

You can use one of Connect’s content curation tools - `connectwidgets` - used to build a customized landing page to share with your audience.

What is `connectwidgets`

`connectwidgets` is an Posit-maintained R package that can be used to query a Connect server for a subset of your existing content items, then organize them within `htmlwidgets` in an R Markdown document or Shiny application.

This guide walks you through:

- installing `connectwidgets`
- updating the `connectwidgets` template and deploying to Connect
- adding custom CSS
- configuring a vanity URL for your `connectwidgets` dashboard
- configuring `RootRedirect` so your audience is automatically redirected to your `connectwidgets` dashboard

Requirements

- Have RStudio Connect v1.9.0+ or Posit Connect
- You will need to have administrative privileges to access and edit the Connect’s configuration to customize the following:
    - `RootRedirect` URL

Prework

API Keys

If you’ve previously added API Keys to your `.Renviron` file as environment variables, please continue to the next section (`connectwidgets`).

API Keys let us make authenticated requests from the RStudio IDE to Connect.

We are going to add our API credentials to a `.Renviron` file, making them available to R as environment variables.

- Navigate to Connect and create and copy your API Key.
- Return to the RStudio IDE.
- Use an `.Renviron` file to set the `CONNECT_SERVER` and `CONNECT_API_KEY` environment variables:
    - In the Console, type `usethis::edit_r_environ()` and press **Enter** to open your .Renviron for editing.
    - Add the following:

```
CONNECT_SERVER=<https://your-server-address.com/>
CONNECT_API_KEY=<paste key value>
```

- Save and close the file.
- Restart R:
    - Mac: **Command** + **Shift** + **F10**
    - PC/Linux: **Ctrl** + **Shift** + **F10**

`connectwidgets`

Install `connectwdigets`

You can either install from CRAN or install from GitHub:

- **Install from CRAN:**
    - Install `connectwidegets` and load the library:

```
install.packages("connectwidgets")
library(connectwidgets)
```

- **Install from GitHub:**

```
# This is the development version
# install.packages("remotes")
remotes::install_github("rstudio/connectwidgets")
library(connectwidgets)
```

The template

`connectwidgets` offers a built-in template that is easy to use and edit to get you started.

What does the template do?

The template supplies intro code chunks that:

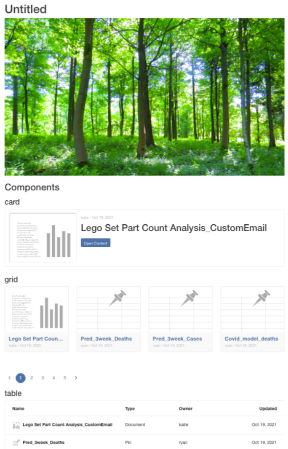
- Load `dplyr`, which will be used to create a subset of content from the server that you want to display.
- Establish a connection to the server by using the environment variables `CONNECT_SERVER` and `CONNECT_API_KEY`.
- Pulls down all of the content that you have access to on the Connect server.
- Creates a `sample_content` variable that slices a set number of content items.

Additionally, you can configure a theme, add custom CSS, and select the components that you wish to display.

To use the template:

- From the RStudio IDE, open a new R Markdown file: **File** > **New File** > **R Markdown...**.
- In the left pane, click **From Template**, select **connectwidgets (HTML)** and click **OK**.
- **Knit** the template.

Here is an example of the knitted output:



Untitled

Components

card

katie • Oct 19, 2021

Lego Set Part Count Analysis_CustomEmail

grid

Lego Set Part Coun...

katie • Oct 19, 2021

Pred_3week_Deaths

ryan • Oct 19, 2021

Pred_3week_Cases

ryan • Oct 19, 2021

Covid_model_deaths

ryan • Oct 19, 2021

table

Name	Type	Owner	Updated
Lego Set Part Count Analysis_CustomEmail	Document	katie	Oct 19, 2021
Pred_3week_Deaths	Pin	ryan	Oct 19, 2021

Figure 1: Knitted output of `connectwidgets` template

Why aren’t the thumbnails of the content displayed?

Unfortunately, the thumbnails for the content will not display in development because:

- That request cannot be made cross-origin. However, the actual content thumbnails will be pulled once the page is deployed to Connect.
- You do not have permission to view the images.
- An image has not been set.

In the knitted output, you will notice that your data is grouped by *Component*.

`connectwidgets` has different types of components, shown above, to display information about your filtered content data:

- card
- grid
- table
- search & filter *(not shown in image above)*

For a full description of these components, please see the `connectwidgets` documentation.

Edit the template

Now, we have gotten to the point where we are ready to edit the template, configure theming, and add customizations.

Don’t forget to Knit to apply your changes to the template.

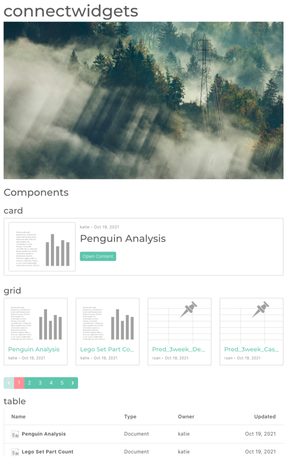
- Feel free to update the `title` of the template documentation.
- Set the theme:
  - You can set your theme by supplying a Bootswatch theme, such as `minty`, in the YAML header of the template:

```yaml
---
title: "connectwidgets" #for example
output:
  html_document:
    theme:
      bootswatch: minty
---
```

PRO TIPS

Creating a Custom Dashboard with ‘`connectwidgets`’ and RStudio Connect

Here is a sample output of the `minty` *theme:*



connectwidgets

Components

card

katie • Oct 19, 2021

Penguin Analysis

grid

Penguin Analysis
katie • Oct 19, 2021

Lego Set Part Co...
katie • Oct 19, 2021

Pred_3week_De...
ryan • Oct 19, 2021

Pred_3week_Cas...
ryan • Oct 19, 2021

table

Name	Type	Owner	Updated
Penguin Analysis	Document	katie	Oct 19, 2021
Lego Set Part Count	Document	katie	Oct 19, 2021

Figure 2: Knitted output of minty theme

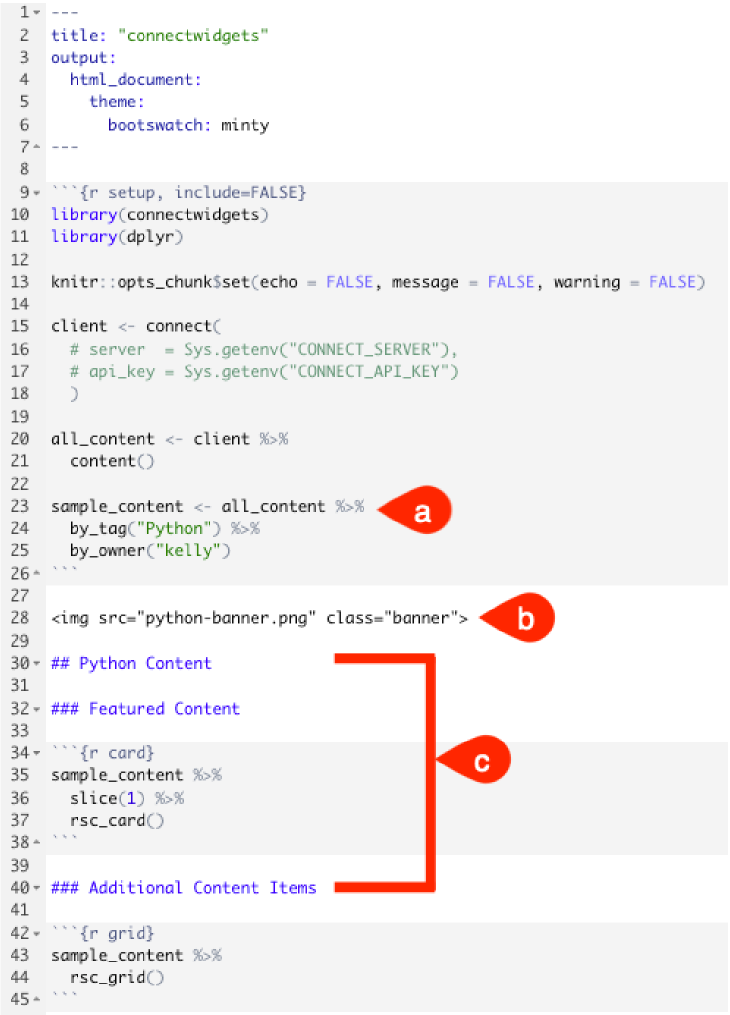
Customize and choose your displayed content:

a. Select the content to use/display by adding two helper functions to select content: `by_tag` and `by_owner`:

- Assuming that you don’t want to display a random set of content, for this example, we are going to filter by content tagged as “Python” and owned by user “kelly”.
- Delete the `table` and `search & filter` component entries from your template. Now, only the `card` and `grid` components are displayed.

b. You may also replace the stock image with an image of your own.

c. Additionally, customize the headers for each component.



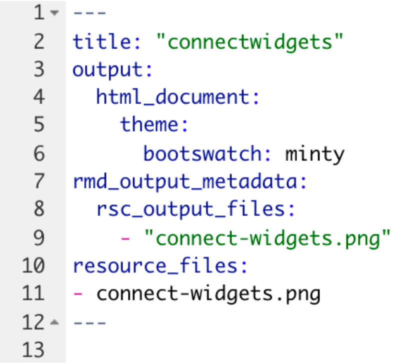
~~~
---
title: "connectwidgets"
output:
  html_document:
    theme:
      bootswatch: minty
---

```{r setup, include=FALSE}
library(connectwidgets)
library(dplyr)

knitr::opts_chunk$set(echo = FALSE, message = FALSE, warning = FALSE)

client <- connect(
  # server  = Sys.getenv("CONNECT_SERVER"),
  # api_key = Sys.getenv("CONNECT_API_KEY")
  )

all_content <- client %>%
  content()

sample_content <- all_content %>%
  by_tag("Python") %>%
  by_owner("kelly")
```

<img src="python-banner.png" class="banner">

## Python Content

### Featured Content

```{r card}
sample_content %>%
  slice(1) %>%
  rsc_card()
```

### Additional Content Items

```{r grid}
sample_content %>%
  rsc_grid()
```
~~~

Figure 3: Example of the template file

PRO TIPS

Creating a Custom Dashboard with ‘`connectwidgets`’ and RStudio Connect

- If you replaced the stock image with an image of your own that isn’t web-based, then add the following to the YAML header:
  - Add `rmd_output_metadata` and `resource_files`:



```
---
title: "connectwidgets"
output:
  html_document:
    theme:
      bootswatch: minty
rmd_output_metadata:
  rsc_output_files:
    - "connect-widgets.png"
resource_files:
- connect-widgets.png
---

```

Replace the section of the template below with the sample:

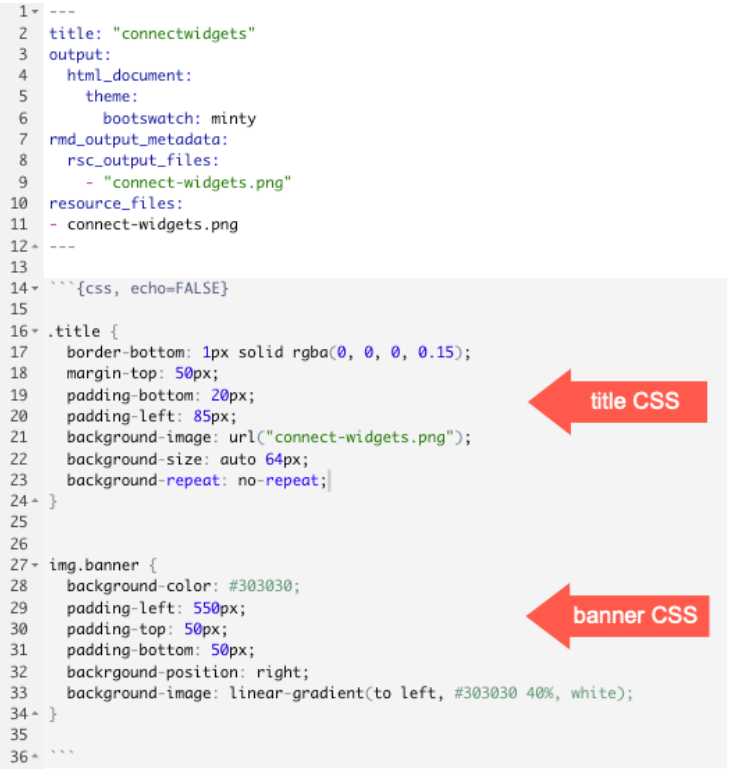
```
library(connectwidgets)
library(dplyr)

knitr::opts_chunk$set(echo = FALSE, message = FALSE,
                      warning = FALSE)

client <- connect(
# server  = Sys.getenv("CONNECT_SERVER"),
# api_key = Sys.getenv("CONNECT_API_KEY")
)

all_content <- client %>%
content()

sample_content <- all_content %>%
  by_tag("") %>%
  by_owner("")
```

Make sure you set `by_tag("")` and `by_owner("")` with properties from your own Connect server.

Custom CSS

Now, we can take customization a step further and add CSS:

- This CSS:
  - Adds a solid border to the bottom of the title.
  - Adds an image to the left of the title.
  - Styles the banner image and applies a background color, gradient, padding, and position.



~~~
---
title: "connectwidgets"
output:
  html_document:
    theme:
      bootswatch: minty
rmd_output_metadata:
  rsc_output_files:
    - "connect-widgets.png"
resource_files:
- connect-widgets.png
---

```{css, echo=FALSE}

.title {
  border-bottom: 1px solid rgba(0, 0, 0, 0.15);
  margin-top: 50px;
  padding-bottom: 20px;
  padding-left: 85px;
  background-image: url("connect-widgets.png");
  background-size: auto 64px;
  background-repeat: no-repeat;
}


img.banner {
  background-color: #303030;
  padding-left: 550px;
  padding-top: 50px;
  padding-bottom: 50px;
  backrgound-position: right;
  background-image: linear-gradient(to left, #303030 40%, white);
}

```
~~~

PRO TIPS

Creating a Custom Dashboard with ‘`connectwidgets`’ and RStudio Connect

Add the following CSS section to the template:

```
{css, echo=FALSE}

.title {
margin-top: 50px;
padding-bottom: 20px;
padding-left: 85px;
background-image: url("connect-widgets.png");
background-size: auto 64px;
background-repeat: no-repeat;
background-image: left-align;
}


img.banner {
background-color: #303030;
padding-left: 550px;
padding-top: 50px;
padding-bottom: 50px;
backrgound-position: right;
background-image: linear-gradient
  (to left, #303030 40%, white);
}
```

Here is an example of the rendered output including all of our customizations:

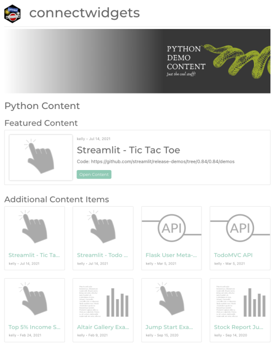


Figure 4: Customized connectwidgets dashboard

Lastly, let’s compare the default template vs. the template that has a theme and our custom CSS changes applied:

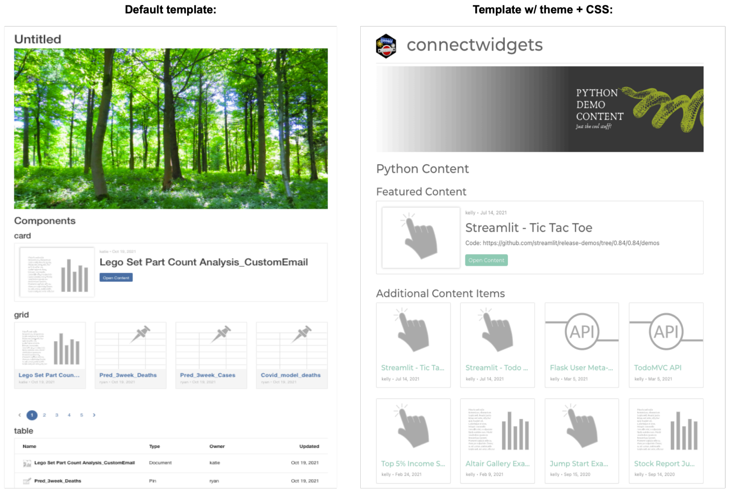


Figure 5: Default dashboard vs. customized dashboard

Publish your content to Connect

- Once you are happy with your `connectwidgets` customizations in the RStudio IDE, publish your content to Connect.

Let’s recap!

- You’ve installed `connectwidgets`
- You’ve selected the components that you wish to display
- Your template is customized
- You’ve successfully published your content to Connect

A custom dashboard

Instead of the default Connect dashboard, imagine that you want your users to see a *pretty* curated dashboard that is built with your `connectwidgets` project/app after logging into Connect.

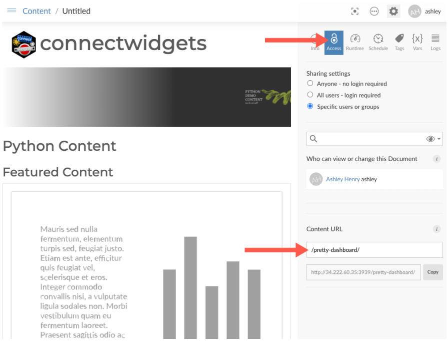
Setting the vanity URL path of your content with `RootRedirect` allows you to do that! E.g.: `RootRedirect ="/pretty-dashboard"`

This is a great way to showcase your pieces of related content and share them as a cohesive project.

To learn more about `RootRedirect` before continuing, please see the `RootRedirect` section in the Admin Guide.

Vanity URL

- Navigate to your Connect server’s dashboard and open your published `connectwidgets` content.
- Select the **Access** panel, set a unique **Content URL** and save your changes.



Sharing settings
○ Anyone - no login required
○ All users - login required
● Specific users or groups

Who can view or change this Document
Ashley Henry ashley

Content URL
/pretty-dashboard/
http://34.222.60.35:3939/pretty-dashboard/

Figure 6: Connect Access panel

To learn more, see the Custom Content URL section in the User Guide.

`RootRedirect`

Requires Role: Administrator

This section has procedures that require you to have administrative privileges to edit the Connect configuration. If you have not been granted these privileges, please coordinate with someone in your organization that is an administrator.

- Navigate to the `/etc/rstudio-connect/rstudio-connect.gcfg` file.
- To set `RootRedirect`, update the `etc/rstudio-connect/rstudio-connect.gcfg` file using the example below:

```
[Server] RootRedirect =“/pretty-dashboard”
```

The `/pretty-dashboard` value is the same path that you used in the previous section for your Content URL.

- Save your changes and restart your server:

```
systemctl restart rstudio-connect
```

Now, when anyone navigates to your Connect Server’s URL, they are redirected to the `RootRedirect` path which points to your `connectwidgets` content that is hosted at the vanity URL path.

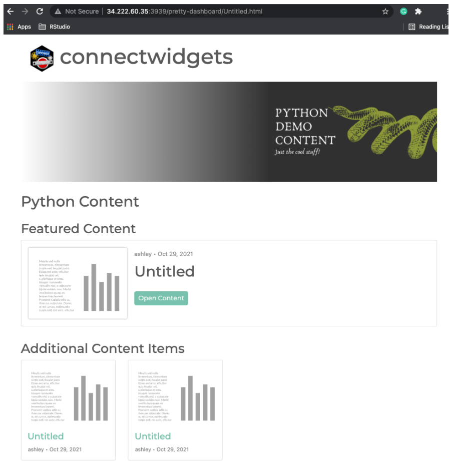


Figure 7: connectwidgets dashboard

`DashboardPath`

Wait! You may be wondering, how do I navigate to the Connect dashboard since the `RootRedirect` brings you to your `connectwidgets` landing page?

You can configure `DashboardPath` which is the URL path to where your Connect’s dashboard is hosted, allowing you to continue to access the dashboard.

- Navigate to the `/etc/rstudio-connect/rstudio-connect.gcfg` file.
- To set `DashboardPath`, update the `etc/rstudio-connect/rstudio-connect.gcfg` file using the example below:

```
[Server] DashboardPath =“/connect/dashboard”
```

- Save your changes and restart your server:

```
systemctl restart rstudio-connect
```

- Now, open a browser and navigate to your Connect server’s IP using the `DashboardPath` that you configured above: http://your-connect-server-address:3939/connect/dashboard/

Where do I go from here?

At this point, you know what `connectwidgets` is, whether it will be useful for your workflow, and how to implement it. What’s next?

As you try `connectwidgets` on your own, here is a list of resources that may be helpful as you implement `connectwidgets`:

PRO TIPS

Creating a Custom Dashboard with ‘`connectwidgets`’ and RStudio Connect

- The `connectwidgets` site has content that walks you through installation, implementation, and provides code examples for you to copy/paste into your own project: https://rstudio.github.io/connectwidgets/
- Posit’s Kelly O’Briant introduces `connectwidgets` in the RStudio Connect 1.9.0 - Content Curation Tools blog announcement here: https://blog.rstudio.com/2021/07/29/rstudio-connect-1-9-0/
- CRAN hosts a comprehensive package document about `connectwidgets`: https://cran.r-project.org/web/packages/connectwidgets/connectwidgets.pdf
- The Configuration section of the Connect Admin Guide Appendix covers `RootRedirect` and `DashboardPath`: https://docs.rstudio.com/connect/admin/appendix/configuration/#Server
- The Connect User Guide covers Custom Content URL: https://docs.rstudio.com/connect/user/content-settings/#custom-urld

Any issues? Let us know here: https://github.com/rstudio/connectwidgets/issues/