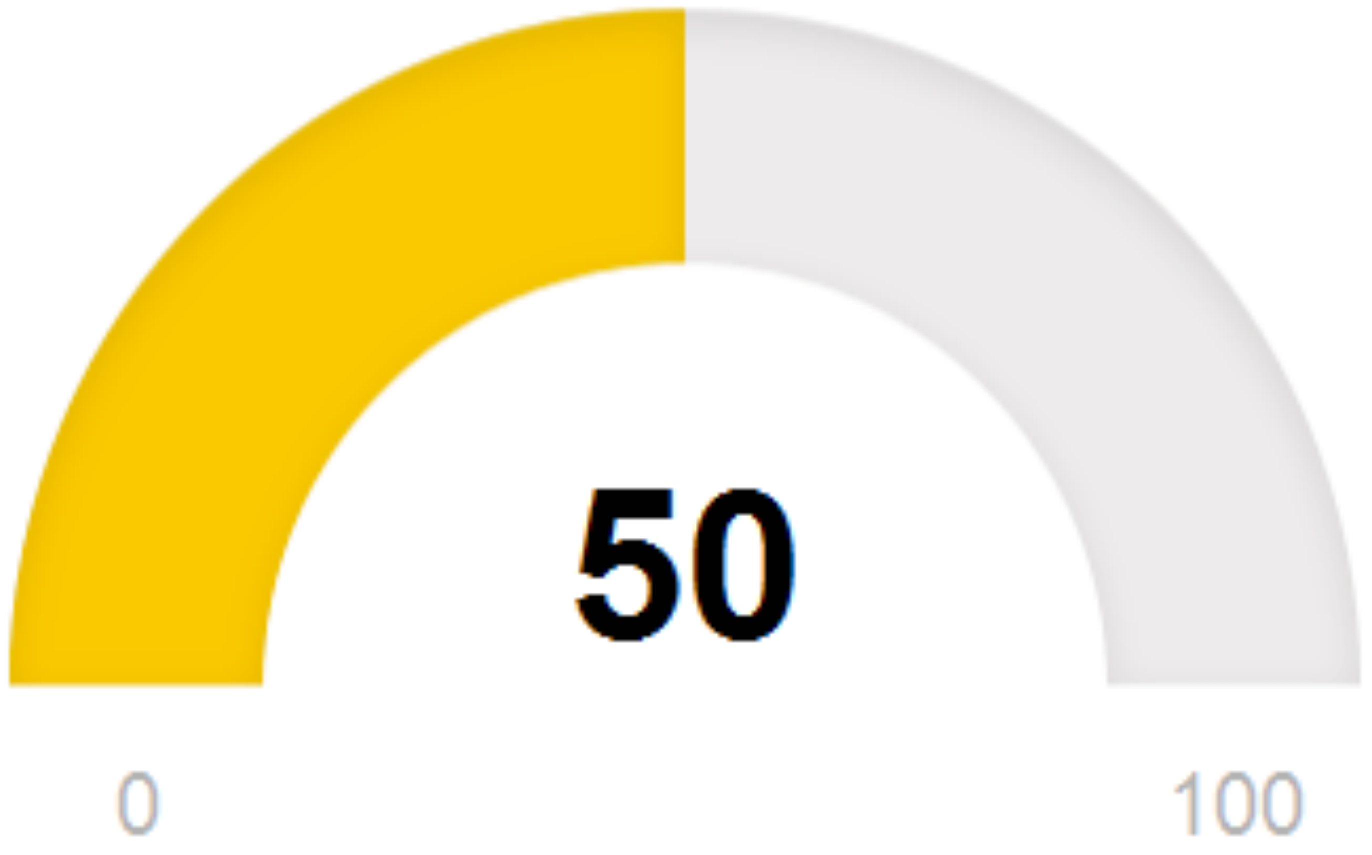
RStudio

250 Northern Ave, Boston, MA 02210
Phone: 844-448-1212

Email: info@rstudio.com
Web: http://www.rstudio.com

Building Dashboards

Using reactiveFileReader, shinydashboard, and htmlTemplate



Nathan Stephens

Director of Solutions Engineering
January 2016
Email: nathan@rstudio.com

What is a dashboard?

A dashboard is an app that:

- Is **always available**
- Displays **key information**
  (e.g. statistics, insights, data, etc.)
- Typically **refreshes automatically** or on a schedule
- May or may not be **interactive**

Dashboards are ubiquitous!

Topics

1. How to automatically update your dashboard with new data

    reactiveFileReader function

    reactivePoll function

2. How to build a great dashboard UI

    shinydashboard package

    NEW **htmlTemplate** function

Example Use Case

An app that monitors server load

Server Load

Log10 Load
1.00
0.10
0.01
Jan 21
Jan 23
Jan 25
Jan 27

Create a live data source

serverLoad.sh

```
#!/bin/bash

MYDIR=/home/nathan/ShinyApps/sol-eng-public/serverLoad/data
FILENAME=serverLoad.txt

LOAD=`uptime | sed 's/.*load average: //' | awk -F\, '{print $1}'`
DATE=`date +%Y-%m-%d:%H:%M:%S`
echo "$DATE,$LOAD" >> $MYDIR/$FILENAME
```

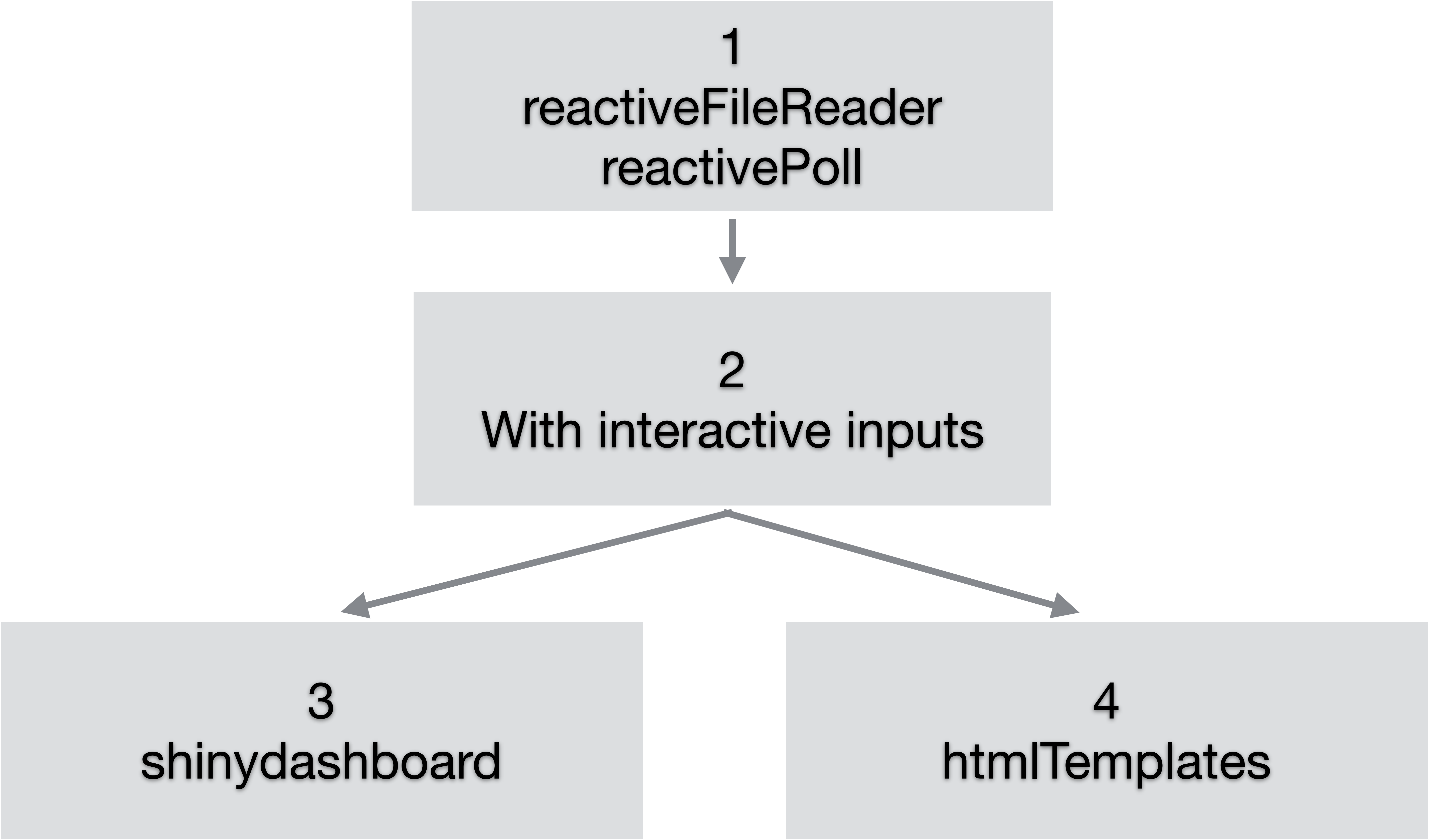
Automate data source

crontab

```
MYDIR=/home/nathan/ShinyApps/serverLoad/data
MYFILE=severLoad.sh
MYLOG=serverLoad.log

* * * * * $MYDIR/$MYFILE >> $MYDIR/$MYLOG 2>&1
* * * * * sleep 5; $MYDIR/$MYFILE >> $MYDIR/$MYLOG 2>&1
* * * * * sleep 10; $MYDIR/$MYFILE >> $MYDIR/$MYLOG 2>&1
                          .
                          .
                          .
* * * * * sleep 55; $MYDIR/$MYFILE >> $MYDIR/$MYLOG 2>&1
```

Server Load App Versions



1
reactiveFileReader
reactivePoll

2
With interactive inputs

3
shinydashboard

4
htmlTemplates

reactiveFileReader

```
fileReaderData <- reactiveFileReader(
  intervalMillis = 500,
  session = session,
  filePath = infile,
  readFunc = read_csv,
  col_names = c('dte', 'Load')
)
```

Server Load App
Version 1

reactiveFileReader

reactivePoll

reactivePoll

```
checkFunc <- function(){
  res <- dbSendQuery(con, "select max(id) from serverLoad")
  dbFetch(res)
}

valueFunc <- function(path){
  res <- dbSendQuery(con, "select * from serverLoad")
  dbFetch(res)
}

pollData <- reactivePoll(
  intervalMillis = 500,
  session = session,
  checkFunc = checkFunc,
  valueFunc = valueFunc
  )
```

Server Load App
Version 2

with interactive inputs

shinyDashboard

https://rstudio.github.io/shinydashboard/

- An R package designed specifically to help you create dashboards with Shiny
- Based on bootstrap
- Has a specific structure
- Some elements can be customized (e.g skins, CSS, colors, etc.)

Server Load App
Version 3

shinydashboard

htmlTemplate Recap

http://shiny.rstudio.com/articles/templates.html

- Integrate Shiny into existing HTML
- Generate complete webpages or components
- Include Javascript or CSS files
- Use templates in packages

Format the UI with htmlTemplate

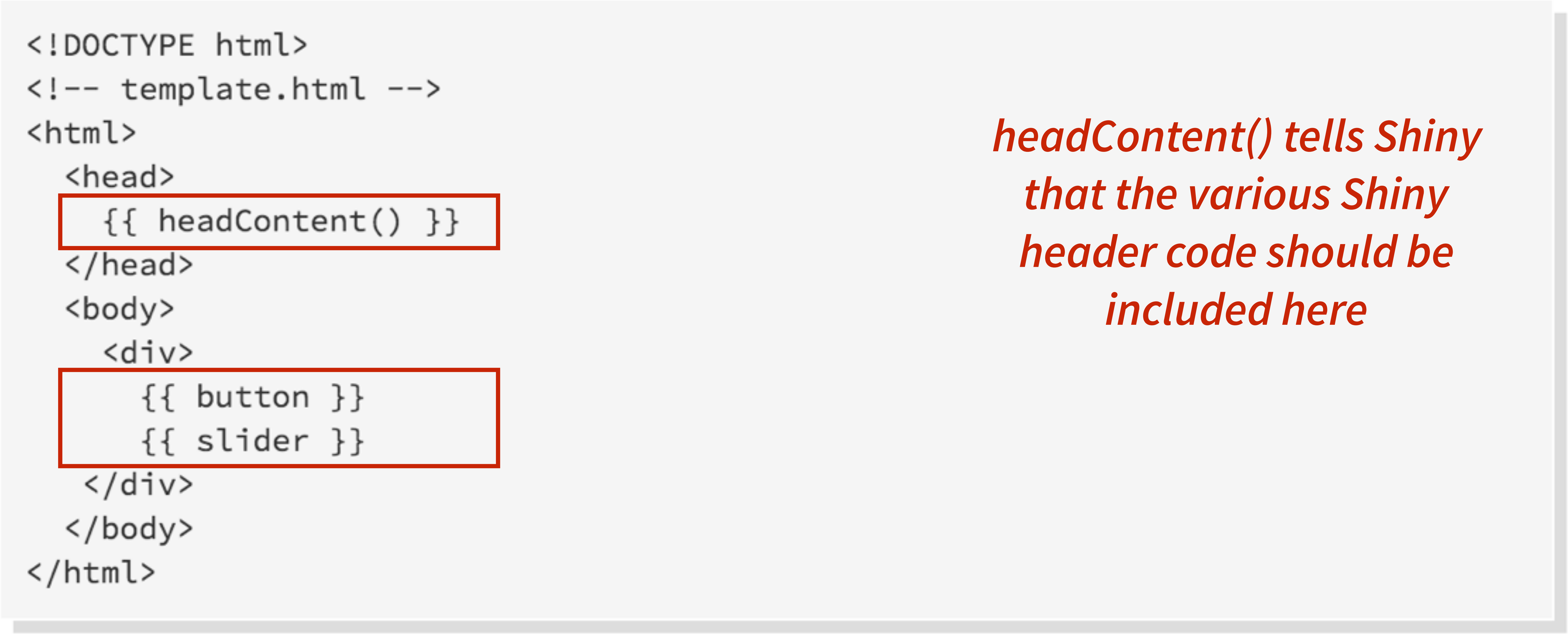
shiny UI

```
ui <- shinyUI(
    actionButton("action", "Action"),
    sliderInput("x" , "X", 1, 100, 50)
)
```

htmlTemplate UI

```
ui <- htmlTemplate(
  button = actionButton("action", "Action"),
  slider = sliderInput("x", "X", 1, 100, 50)
)
```

Insert the Shiny Elements Into HTML



```
<!DOCTYPE html>
<!-- template.html -->
<html>
  <head>
    {{ headContent() }}
  </head>
  <body>
    <div>
      {{ button }}
      {{ slider }}
     </div>
  </body>
</html>
```

headContent() tells Shiny that the various Shiny header code should be included here

Server Load App
Version 4

Education htmlTemplate
Finance htmlTemplate

Conclusion

Use **reactiveFileReader** and **reactivePoll** functions to automatically update your app with new data.

Use the **shinydashboard** package to format your UI with a specific structure.

NEW Use the **htmlTemplate** function to embed your shiny dashboard elements into your organization's HTML template.