Detect mobile devices on a shiny app

Why I wanted to do so

My wife is working on a shiny app... and she needed to show a plotly graph to desktop users and a ggplot2 graph to mobile users.

Check the user agent, he said

My first suggestion was for her to check out the user agent... she did it by using `session$request$HTTP_USER_AGENT` and locally, it worked just fine. The fun part was when she deployed the code to her shiny server...

She wouldn't get a user agent at all. Of course, I forgot to pass the header on nginx... but, even after I did (and verified on the shiny server traffic logs that it was being received by shiny) the value was still empty. And the reason was... it's a Pro feature! I understand they need to make money out of their beautiful software, but this sounds... odd to me.

How do we do it?

Aaanyway, then thanks to Dean Attali's suggestion, we started looking into how to actually do that via Javascript. After all, a shiny app is just a web app.

After a couple failed attempts with shinyjs (it requested to install V8 and it was heavy heavy heavy) I discovered input bindings. They're just a clean way for you to implement your own shiny elements, and trigger whatever code you want on Javascript.

They're not super complicated to set up... but just to save you a couple minutes of your life, I'll share my solution :)

The javascript code

I created a `www/js/mobile.js` file with this content:

```
var isMobileBinding = new Shiny.InputBinding();
$.extend(isMobileBinding, {
  find: function(scope) {
    return $(scope).find(".mobile-element");
    callback();
  },
  getValue: function(el) {
    return /((iPhone)|(iPod)|(iPad)|(Android)|(BlackBerry))/.test(navigator.
  },
  setValue: function(el, value) {
  },
  subscribe: function(el, callback) {
  },
  unsubscribe: function(el) {
  }
});

Shiny.inputBindings.register(isMobileBinding);
```

This is kind of cheating... as I know there will be no changes (if you're visiting on a mobile device, that's not going to change on this session) I'm triggering the `callback()` (what generates R to update the value) as soon as the element is initialized... and `getValue` just does a very rudimentary regex check on the user agent.

The R code

All you need to do is:

1. Define the `mobileDetect` function
2. Add it to the `ui` element (when you're interested about it)
3. Use it on the server :) now you can tell if the user is on a mobile device or not

Before your `ui` element, define this method:

```
mobileDetect <- function(inputId, value = 0) {
  tagList(
    singleton(tags$head(tags$script(src = "js/mobile.js"))),
    tags$input(id = inputId,
               class = "mobile-element",
               type = "hidden")
  )
}
```

That just loads the javascript file and creates an html hidden input. Then, as you're defining your `ui`, you need to use it specifying what name you'll assign to it like this:

```
shinyUI(fluidPage(
  titlePanel("Are you on a mobile device?"),

  mobileDetect('isMobile'),
  textOutput('isItMobile')
))
```

And then... it's going to be magically available on the server! you can just reference the element as if it was a built-in shiny one:

```
shinyServer(function(input, output) {
  output$isItMobile <- renderText({
    ifelse(input$isMobile, "You are on a mobile device", "You are not on a mo
```

```
    })
})
```

I put together a tiny example project. You can see its code here or view it running on shinyapps.io (if I have enough hours left :P)