1 Autoloading

Finally it is possible to ask Bokeh to return a `<script>` tag that will replace itself with a Bokeh plot, wherever happens to be located. The script will also check for ***BokehJS*** and load it, if necessary, so it is possible to embed a plot by placing this script tag alone in your document.

There are two cases:

1. Server Data
2. Static Data

1.1 server data

The simplest case is to use the Bokeh server to persist your plot and data. Additionally, the Bokeh server affords the opportunity of animated plots or updating plots with streaming data. The `autoload_server()` function accepts a plot object and a Bokeh server Session object. It returns a `<script>` tag that will load both your plot and data from the Bokeh server.

As a concrete example, here is some simple code using `autoload_server()` with a default session:

```
from bokeh.plotting import figure, push
from bokeh.embed import autoload_server
from bokeh.session import Session
from bokeh.document import Document

# alternative to these lines, bokeh.io.output_server(...)
document = Document()
session = Session()
session.use_doc('population_reveal')
session.load_document(document)

plot = figure()
plot.circle([1,2], [3,4])

document.add(push)
push(session, document)

script = autoload_server(plot, session)
```

The resulting <script> tag that you can use to embed the plot inside a document looks like:

```
<script
    src="http://localhost:5006/bokeh/autoload.js/f64f7959-017d-4d1b-924e-899a61
    id="f64f7959-017d-4d1b-924e-899a61fed42b"
    async="true"
    data-bokeh-data="server"
    data-bokeh-modelid="82ef36f7-9d58-47c8-9b0d-201947febb00"
    data-bokeh-root-url="http://localhost:5006/"
    data-bokeh-docid="2b4c75a2-8311-4b4d-b014-370b430d6469"
    data-bokeh-docapikey="8c4e34e5-04f9-4c1c-b92f-fb1ec0d52cae"
    data-bokeh-loglevel="info"
></script>
```

Note To execute the code above, a Bokeh server must be running.

1.2 static data

If you do not need or want to use the Bokeh server, then the you can use the autoload_static() function. This function takes the plot object you want to display together with a resources specification and path to load a script from. It will return a self-contained <script> tag, together with some JavaScript code that contains the data for your plot. This code should be saved to the script path you provided. The <script> tag will load this separate script to realize your plot.

Here is how you might use autoload_static() with a simple plot:

```
from bokeh.resources import CDN
from bokeh.plotting import figure
from bokeh.embed import autoload_static

plot = figure()
plot.circle([1,2], [3,4])

js, tag = autoload_static(plot, CDN, "some/path")
```

The resulting <script> tag looks like:

```
<script
    src="some/path"
    id="c5339dfd-a354-4e09-bba4-466f58a574f1"
    async="true"
    data-bokeh-data="static"
    data-bokeh-modelid="7b226555-8e16-4c29-ba2a-df2d308588dc"
    data-bokeh-modeltype="Plot"
    data-bokeh-loglevel="info"
></script>
```

The resulting JavaScript code should be saved to a file that can be reached on the server at `some/path`, from the document that has the plot embedded.
Note:In both cases the <`script`> tag loads a <`div`> in place, so it must be placed under <`head`>.