Invoke scripts for Jupyter notebooks

The hub tool invoke script is located in the tool's middleware/ subdirectory. When you first create a tool, the basic invoke script provided must be edited to work with Jupyter notebook tools.

This writeup shows you how to create Jupyter tools with three different appearances: notebook, App, and Tool mode.

invoke_app and start_jupyter

To deploy a Jupyter notebook as a tool on your hub, you call the invoke_app executable, which in turn calls start_jupyter. Each have their own arguments:

arguments for start_jupyter

- **-h, --help**  show this help message and exit.
- **-d**  Show debug (verbose) output.
- **-t**  Run as a Tool with no notebook controls.
- **-c**  Copy instead of link notebook files.
- **-A**  Run in AppMode.
- **-T dir**  Search for notebook starting in dir.
- **--themes**  Enable notebook themes

arguments for invoke_app

- **-t** Tool name
- **-C** command to execute
- **-r** Rappture version to use (normally specify none for notebook tools)
- **-w** headless
- **-u** environment package (repeat as necessary)

invoke_app: starting point

The basic invoke script for Jupyter notebooks looks like this:

```
/usr/bin/invoke_app "$@" -t TOOLNAME \
    -C "start_jupyter -T @tool APP.ipynb" \
    -r none \
    -w headless \
```
InVOKE SCRIPTS FOR JUPYTER NOTEBOOKS

-\texttt{u anaconda-X}

Invoking a Jupyter tool this way gives a notebook with all its code cells displayed to the user.

Where:

\begin{itemize}
  \item TOOLNAME is the short name of the tool
  \item APP is the name of the main notebook that runs the tool
  \item anaconda-X is the current anaconda installation
\end{itemize}

\textbf{start\_jupyter arguments}

Control the way the notebook appears when run as a tool, using the arguments passed to the start\_jupyter executable.

You can run a Jupyter tool in three ways:

\begin{itemize}
  \item notebook mode, in which all code cells are displayed to the user (shown above)
  \item app mode, in which code cells are initially hidden but can be displayed
  \item tool mode, in which code cells are hidden and cannot be displayed
\end{itemize}

\textbf{for App Mode}

For a notebook tool that hides its code cells and shows only the UI and markdown elements on initial run, add the -A argument in the start\_jupyter call:

\begin{verbatim}
/usr/bin/invoke_app "$@" -t TOOLNAME \
  -C "start_jupyter -A -T @tool APP.ipynb" \
  -u anaconda-X \
    -w headless \
  -r none
\end{verbatim}

The tool user can toggle the tool's "Edit App" button to show the underlying code cells, making this a great teaching/demo option.

\textbf{NOTE} that this differs from the invoke\_app -A argument.

\textbf{for Tool Mode}

To permanently hide code cells from the user in App Mode, specify the -A and -t arguments in

\begin{verbatim}
/usr/bin/invoke_app "$@" -t TOOLNAME \
  -C "start_jupyter -A -T @tool APP.ipynb" \
  -u anaconda-X \
    -w headless \
  -r none
\end{verbatim}
the start_jupyter call:

```
/usr/bin/invoke_app "$@" -t TOOLNAME \
   -C "start_jupyter -A -t -T @tool APP.ipynb" \
   -u anaconda-X \
       -w headless \
   -r none
```

The Edit App button will not be displayed to the tool user.

**NOTE** that this differs from the `invoke_app -t` argument.

**errors**

**specify no rappture**

```
/usr/bin/invoke_app "$@" -t TOOLNAME \
   -C "start_jupyter -T @tool APP.ipynb" \
   -u anaconda-X \
       -w headless \
   -r none
```

**Error:**

Running the tool's invoke script from a workspace, returns:

"could not find a rappture installation: RAPPTURE_PATH=,"

**Fix:**

Be sure to supply the "-r none" argument in the `invoke_app` call, as above. No quotation marks are needed.