Voilà

What is Voilà?

Voilà allows you to convert a Jupyter Notebook into an interactive dashboard that allows you to share your work with others. It is secure and customizable, giving you control over what your readers experience. Voilà can be used as a standalone application, or as a Jupyter server extension. This page describes how to do each. Before you begin, make sure that you follow the steps in Installing Voilà.

Installation

Voilà can be installed with the mamba or conda package manager

```
mamba install -c conda-forge voila
```

or from PyPi:

```
pip install voila
```

Using Voilà

Voilà can be used as a standalone application, or as a Jupyter server extension. This page describes how to do each. Before you begin, make sure that you follow the steps in Installing Voilà. As a standalone application

Voilà can be used to run, convert, and serve a Jupyter notebook as a standalone app. This can be done via the command-line, with the following pattern:

```
voila <path-to-notebook> <options>
```

For example, to render the bqplot example notebook as a standalone app, run
Voilà displays a message when your notebook-based application is live. By default, Voilà runs at localhost:8866.

To serve a directory of Jupyter Notebooks, navigate to the directory you’d like to serve, then simply run Voilà:

```
$ git clone https://github.com/voila-dashboards/voila
$ cd voila
$ voila notebooks/bqplot.ipynb
```

The page served by Voilà will now contain a list of any notebooks in the directory. By clicking on one, you will trigger Voilà’s conversion process. A new Jupyter kernel will be created for each notebook you click.

As a Jupyter extension

You can also use Voilà from within a Jupyter server (e.g., after running `jupyter lab` or `jupyter notebook`).

Voilà can also be used as a notebook server extension, both with the notebook server or with the jupyter_server.

To use Voilà within a pre-existing Jupyter server, first start the server, then go to the following URL:

```
<url-of-my-server>/voila
```

For example, if you typed `jupyter lab` and it was running at `http://localhost:8888/lab`, then Voilà would be accessed at `http://localhost:8888/voila`.

In this case, Voilà will serve the directory in which the Jupyter server was started.

For more information to Voilà check out the official documentation!

Examples

You can find some examples of Voilà dashboards in Voila Gallery TU Wien.
Voilà Templates offered by TU Wien

TU Wien offers two very useful Templates for Voilà:

- `material-tuwien`
- `vuetify-default-tuwien`

Material TU Wien Template
This template allows you to create basic standalone applications from your notebooks.

Classic **material-tuwien template** can be installed using pip:

```
pip install voila-material-tuwien
```

To use this template just pass **--template=material-tuwien** to the Voilà command line.

Let’s say you have a notebook named `my_notebook.ipynb` in your working directory. You would serve it with voila as follows:

```
voila my_notebook.ipynb --template=material-tuwien
```

**Material Template** also allows you to serve all notebooks in your working directory at the same time. To do so, just pass following command to Voilà command line:

```
voila . --template=material-tuwien
```

Tree structure of your working directory is shown, but only Python notebooks are served and can be opened.

**Vuetify Default Template**
Problem which **Material Template** was facing is responsivity. This can be easily solved using **Voila Vuetify Template**. You can take a look at one responsive notebook which we deployed [here](#).

Voila Vuetify Template can also be installed using pip:

```
pip install voila-vuetify-template-tuwien
```

To render use this template specify `--template=vuetify-default-tuwien` in Voila command line.

To open all notebooks in your working folder use:

Voila Vuetify Template can also be installed using pip:

```
voila . --template=vuetify-default-tuwien
```

To open a specific notebook from your working directory specify notebook's name instead of `.`.

Note that this template renders only Vuetify notebooks in which at least one figure has mount-id set to `content-main`.
If notebook has no figures with required mount-id: `content-main` a blank page is rendered with a message: Placeholder for widget with mount id `content-main`.

**ipyvetify for making a notebook responsive:**

Any jupyter notebook can be turned into Vuetify notebook. At the end of your notebook you just have to signalize to Vuetify template which figures should be rendered using `ipyvetify`. The following example is taken from [Histogram-Sliders.ipynb](https://github.com/mpetojevic/bqplots-voila-vuetify)
import ipyvuetify as v
v.Tabs(_metadata={'mount_id': 'content-main'}, children=[
    v.Tab(children=['BQ-Histogram Plots']),
    v.Tab(children=['Gaussian plot']),
    v.TabItem(children=[
        v.Layout(row=True, wrap=True, align_center=True, children=[
            v.Flex(xs12=True, lg6=True, xl4=True, children=[
                fig, slider
            ]),
            v.Flex(xs12=True, lg6=True, xl4=True, children=[
                figHist2, sliderHist2
            ]),
            v.Flex(xs12=True, xl4=True, children=[
                fig2
            ])
        ]),
        v.TabItem(children=[
            v.Container(children=[final_layout])
        ])
    ])
])

Note that fig, slider, figHist2, sliderHist2, fig2, final_layout are all figures created in the notebook.

**Deploy with Gitlab**

Git is a software for distributed Version management of files (especially of source code). With TUGitLab, TU.it provides an application for managing repositories based on Git. TU members with an active TUaccount can use this service when working on GitLab projects. In TUGitLab, institute administrators activated in the Online Account Management can manage GitLab groups, assign project permissions, and assign external project partners as additional GitLab users.

**Create a new Gitlab Group**

This step is only required if you do not already have a group set up.

In GitLab, you use groups to manage one or more related projects at the same time. You can use groups to manage permissions for your projects. If someone has access to the group, they get access to all the projects in the group.

You can also view all of the issues and merge requests for the projects in the group, and view analytics that show the group’s activity. You can use groups to communicate with all of the members of the group at once.

To create a group:

1. On the top bar, either:
   - Select Main menu > Groups > View all groups, and on the right, select New group.
   - To the left of the search box, select the plus sign and then New group.
2. Select Create group.
3. Enter a name for the group in Group name. For a list of words that cannot be used as group names, see reserved names.
4. Enter a path for the group in Group URL, which is used for the namespace.
5. Choose the visibility level.
6. Personalize your GitLab experience by answering the following questions:
   - What is your role?
   - Who will be using this group?
   - What will you use this group for?
7. Invite GitLab members or other users to join the group.
   - Please also add the dataLAB members Florian Jäger and Marijana Petojevic as owners, so that they can activate your deployment.

**Activate your Deployment by contacting dataLAB members**

Lastly your deployment must be manually activated by TU.it dataLAB members.

To activate your automatic deployment contact Florian Jäger (florian.jaeger@tuwien.ac.at) and Marijana Petojevic (marijana.petojevic@tuwien.ac.at) and include the following information:

- Your Institute or Organization of TU
- Repository link
You can contact us using following template: Access to Docker Registry for Voila Template - Request

Please also check if Florian Jäger and Marijana Petojevic have owner permissions in your Gitlab group.

After the activation your notebooks should now be publicly available under <project slug>.jupyter.hpc.tuwien.ac.at