A Template for Rapid Development of Interactive Computing Tools

Rob Campbell, Rajesh Kalyanam, Carol Song and Lan Zhao
Research Computing
Purdue University
West Lafayette, USA

ABSTRACT

A minimal set of reusable software packages and methodologies is described which have enabled the efficient creation of multiple, self-contained, data-centric web applications ("tools"). The tools themselves are dedicated to the analysis, sharing, and visualization of data produced by researchers from diverse domains. The development template described is intended to support a range of development contexts, developer proficiencies, and hosting environments.

Specifically, we organized the tools' codebases within a simplified Model–View–Controller pattern and used Anaconda, Python, Jupyter, Docker, and Voilà [1] to develop and deploy the tools. The basis of the tool code is Python running as a Jupyter notebook. However, we forgo traditional, cell-based notebook organization and use Jupyter only to enable the use of ipywidgets for user interface controls. Further, this development context allowed us to leverage the use of libraries such as Pandas for data access, Ipyleaflet for geospatial visualization, and Matplotlib and Plotly for plots, charts, and graphs.

A portion of the tools produced were published as stand-alone, containerized, independently hosted [2] web applications. We relied on Docker and Voilà to create containerized applications which include their own web servers. Others were published as tools embedded within a HUBzero [3] based science gateway. In this context, we relied on the gateway's Jupyter server and its ability to render the tool using Jupyter's "Appmode" extension. Thus far six tools have been developed using this template with subsequent tools only taking approximately one third as long as the initial effort employed in developing the template.

The development template emerged in order to easily enhance code created by researchers and supplement it with a user interface (UI) in order to expose a tool allowing others to explore the researchers' data through a variety of rich UI elements that provide data-specific filtering, query, and visualization capabilities. One requirement was that the researchers themselves, not dedicated web developers, should be able to enhance and maintain the tool code. The template then evolved as we used it as the basis for the development of other tools and then as the basis for student interns to quickly build data-centric applications.

We describe the steps we use to create these types of tools as well as the pitfalls we encountered as we assembled the template over time.

Keywords—*gateways; tools; development; Jupyter; Voilà*

REFERENCES

[1] QuantStack. 2019. And voilà! …from Jupyter notebooks to standalone applications and dashboards. https://blog.jupyter.org/and-voil%C3%A0-f6a2c08a4a93.

[2] Smith, PM, E Gough, A Younts, B Werts, TJ Hacker, N Neumeister, & JH Wisecaver. 2020. The "Geddes" Composable Platform-An Evolution of Community Clusters for a Composable World. IEEE/ACM International Workshop on Interoperability of Supercomputing and Cloud Technologies (SuperCompCloud). 2020: 33-38

[3] McLennan, M. and Kennell, R., 2010. HUBzero: a platform for dissemination and collaboration in computational science and engineering. Computing in Science & Engineering, 12(2), pp.48-