SAS® GLOBAL FORUM 2021

1135-2021

Writing SASⓇGlobal Forum papers in Markdown

Allan Bowe

ABSTRACT

Is your SAS scattered across SAS folders, local directories and server file systems? Are your deliverables delayed by the need to build tools? Does your deployment process involve manual steps?

The SASjs framework enables code consistency across teams and projects, de-risks the use of shared tools and dependencies, and facilitates continuous deployment to SAS environments. The framework can be applied to SAS® Viya® jobs, SAS® 9 Stored Processes, and even regular SAS programs on a file system. Join this session and learn how to create (scaffold) a SASjs project, add a job, add a macro dependency, add a program (include) dependency, add job init and term files, deploy the jobs to SAS, and run them as part of a flow. I will share the secret of how to deploy to SAS Viya without a client and secret. And, the entire demo will be performed from a local text editor (VSCode).

INTRODUCTION

If you are a SAS Developer with two or more projects under your belt, you'll be familiar with the fact that SAS affords incredible flexibility when it comes to development and deployment. Every company has different standards, and even you yourself have probably developed some coding habits that are not shared by your colleagues.

As it happens, this same situation applies also to JS (JavaScript). JS is a very flexible, loose language, and can easily be mis-used to create spaghetti projects that are hard to maintain and difficult to debug. In response, the industry has created a number of frameworks that 'wrap around' JS to provide structure and consistency in the way projects are built - such as React, Angular, and Vue. This is great for application owners, as it makes it much easier to on-board new developers, and it reduces the total cost of ownership, as well as maintaining development velocity as each application becomes increasingly complex.

The goal of SASjs is to provide a structure, set of opinions, and a suite of tools so that developers can spend more time on the things that matter - ie, rapid delivery of business value, to any SAS platform, in a repeatable and scalable fashion.

In this paper we walk through the process of setting up a project, deploying it, executing it and documenting it.

THE SASJS FRAMEWORK

The SASjs Framework provides a set of tools to abstract away the common complexities of a typical SAS deployment. Each of these tools can be used individually, or together as part of a cohesive and integrated approach to managing the SAS application lifecycle - ie project setup, development, compilation, deployment, execution, testing, documentation, and more.

The core components are:

sasjs/adapter - handles SASLogon authentication and communication between frontend and backend sasjs/cli - a Command Line Interface for automating common tasks sasjs/core - a library with over 120 SAS macros geared towards application developers Other components include:

sasjs/vscode-extension - SAS code execution, syntax highlighting and linting in the VS Code IDE sasjs/lint - the linter is used in both the VS Code extension and the SASjs CLI numerous seed apps ready to kick start a SAS project There are even a series of ready-made apps, built using the framework.

GETTING STARTED There are some pre-requisites to install before continuing:

NPM. This provides the run-time for the CLI. Instructions for the portable version are here. GIT. Not strictly necessary but highly recommended, and git-bash provides unix-like commands in windows. VS Code. You could use other IDEs, however this is the one primarily supported by SASjs.

For most of the rest of this paper we will be submitting terminal commands. If you are working in VS Code you may wish to change your default terminal to git-bash - to do this, open Terminal (Terminal/New Terminal), choose "select default profile" and select 'git bash' from the list of options (see image on the right).

To check you have everything installed correctly, try running:

```
npm -v
```

This should return a version number. If not, go back and check your installation of NodeJS (did you add the correct folder to your PATH if running the portable version? If in VS Code, did you try re-opening the editor?)

If you have a version number, you are ready to install the SASjs CLI. Simply run:

```
npm i -g @sasjs/cli
```

SUB SECTION

Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum.

CONCLUSION

Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum.

ACKNOWLEDGMENTS

Thanks to everyone for reading this!

RECOMMENDED READING

All the book!

CONTACT INFORMATION

Your comments and questions are valued and encouraged. Contact the author at:

Allan Bowe Email: allan.bowe@analytium.co.uk

REFERENCES