Trailhead Module

Continuous Integration Using Salesforce DX

https://trailhead.salesforce.com/modules/sfdx_travis_ci

sfdx-travisci

For a fully guided walk through of setting up and configuring this sample, see the Continuous Integration Using Salesforce DXTrailhead module.

This repository shows one way you can successfully setup Salesforce DX with Travis CI. We make a few assumptions in this README:

- You know how to get your Github repository setup with Travis CI. (Here's their Getting Started guide.)
- You've installed the Travis CLI.
- You have properly setup JWT-Based Authorization Flow (i.e. headless). I recommend using these steps for generating your Self-Signed SSL Certificate.

If any any of these assumptions aren't true, the following steps won't work.

Getting Started

1. Make sure you have the Salesforce CLI installed. Check by running `sfdx force --help` and confirm you see the command output. If you don't have it installed you can download and install it from here.
2. Confirm you can perform a JWT-based auth: `sfdx force:auth:jwt:grant --clientid <your_consumer_key> --jwtkeyfile server.key --username <your_username> --setdefaultdevhubusername`

   Note: For more info on setting up JWT-based auth see Authorize an Org Using the JWT-Based Flow in the Salesforce DX Developer Guide.

3. Fork this repo into your github account using the fork link at the top of the page.

4. Clone your forked repo locally: `git clone https://github.com/<git_username>/sfdx-travisci.git`

5. From you JWT-Based connected app on Salesforce, retrieve the generated `Consumer Key`.
6. Set your `Consumer Key` and `Username` using the Travis CLI. Note that this username is the username that you use to access your Dev Hub.

   ```
   travis env set CONSUMERKEY <your_consumer_key> travis env set USERNAME <your_username>
   ```

7. Add your `server.key` that you generated previously to the folder called `assets`.
8. Open the `.travis.yml` file and remove the first line that starts with `openssl ...` and save the file.
9. From the root folder of your local project, encrypt your `server.key` value:

   ```
   travis encrypt-file assets/server.key assets/server.key.enc --add
   ```

10. IMPORTANT! Remove your `server.key`: `rm assets/server.key`, you should never store keys or certificates in a public place.

And you should be ready to go! Now when you commit and push a change, your change will kick off a Travis CI build.

Enjoy!

Contributing to the Repository

If you find any issues or opportunities for improving this repository, fix them! Feel free to contribute to this project by forking this repository and make changes to the content. Once you've made your changes, share them back with the community by sending a pull request. Please see How to send pull requests for more information about contributing to Github projects.

Reporting Issues

If you find any issues with this demo that you can't fix, feel free to report them in the issues section of this repository.

Test