

# BOS Loader

In this article you'll learn how to develop, test, and deploy BOS components using CLI tools. You can use this workflow to tap into the collaboration, pull-request, and other GitHub benefits while still deploying components to the BOS.

[BOS Component Loader](#) serves a local directory of component files as a JSON payload properly formatted to be plugged into a BOSredirectMap . When paired with a viewer configured to call out to this loader, it enables local component development.

## Development flow

1. (Optional)
2. Download and install [bos CLI](#)
3. .
4. info
5. You needbos
6. CLI if you have component code on the BOS already that you want to use or if you want to manage component deploys locally instead of the GitHub actions CI/CD.
7. To get component code saved on the BOS, usebos
8. to download the source code. Otherwise, create asrc
9. folder.
10. Create a component within that src folder likesrc.jsx
11. .
12. tip
13. It's common practice to use.
14. delimited component names for namespacing. You can handle this with folders for better files organization.
15. For example,AppName.Component
16. →AppName/Component.jsx
17. .
18. Download and install [BOS Component Loader](#)
19. (bos-loader
20. ).
21. Runbos-loader --path src
22. (or run fromsrc
23. folder)
24. Open<https://near.org/flags>
25. , and set the loader URL tohttp://127.0.0.1:3030
26. .
27. Open<https://near.org/widget/>
28. (case sensitive)
29. info
30. If you're testing ontestnet
31. , use your testnet account and open<https://test.near.org>
32. instead.
33. Runbos-loader --path src
34. locally, set loader URL in<https://test.near.org/flags>
35. and open<https://test.near.org/widget/>
36. to view your component locally.
37. tip
38. You can work on multiple components at once by embedding them in a wrapper component.
39. Make changes to the component's code.
40. info
41. You must refresh the browser's web page to see the changes.
42. When you're done, use theX
43. on the banner to stop loading locally.

## Component deployment

At this point, your new component is ready to be deployed. To deploy, you can use either of the following two paths:

- Usebos
- CLI to deploy from command line:
- bos deploy
- Set up a [GitHub actions](#)
- deployment workflow. Check [this document](#)
- for instructions.

You should now be able to see your component in discovery. Happy Hacking [Edit this page](#) Last updated on Sep 18, 2023 by Vlad Frolov Was this page helpful? Yes No

[Previous Frontend & Multiple Contracts](#) [Next Smart Contract Interaction](#)