# Forms

Edge Delivery Services supports web forms. The Adobe portion is the mechanism that ingests submitted data into a Microsoft Excel or Google Sheet document.

For more information, see https://www.aem.live/developer/forms.

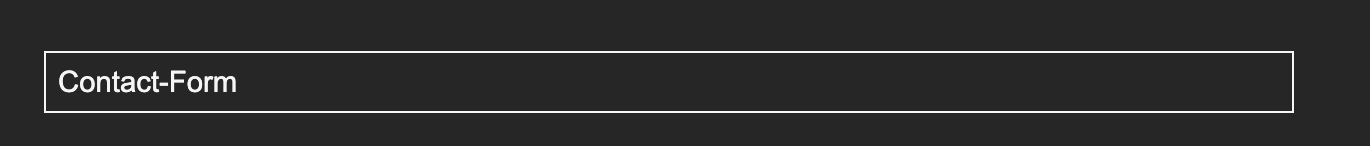
## Author the Form Block

In this step, you will author the Contact-Form Block in the /drafts Folder.

In Sharepoint, open your /drafts Folder of your site. Create a new Word Document named “contact-form.docx”. Open the document.

Create a table with 1 column and 1 rows.

In the cell, enter Contact-Form.



Preview this file. You should not see a form but that is because the code does not exist yet.

## Harvest the Forms Block Code

In this step, you will harvest the code for the Contact Form Block from another project.

Open your ensembledotcom project in Visual Studio Code.

In the blocks folder, create a folder named contact-form. In the contact-form Folder, create 2 new files; contact-form.css and contact-form.js.

Go to <https://github.com/EnsembleSC/EnsembleDotComAfter/tree/main/blocks/contact-form>.

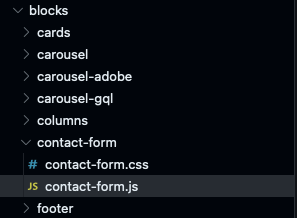
From Github, copy the contents of contact-form.css and contact-form.js into their respective copies in your ensembledotcom codebase.

There is a copy button in the Github UI for your convenience.



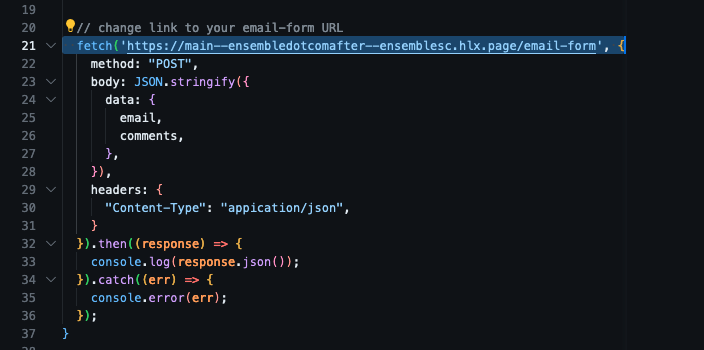
## Update the Data Intake File Path

In this step, you will update the code so that the submitted data goes into your intake file.



In Visual Studio Code, open the /blocks/contact-form/contact-form.js file.

Go to row 21. Look for <https://main--ensembledotcomafter--ensemblesc.hlx.page/email-form>



Update ensemblesc to your Github Username.

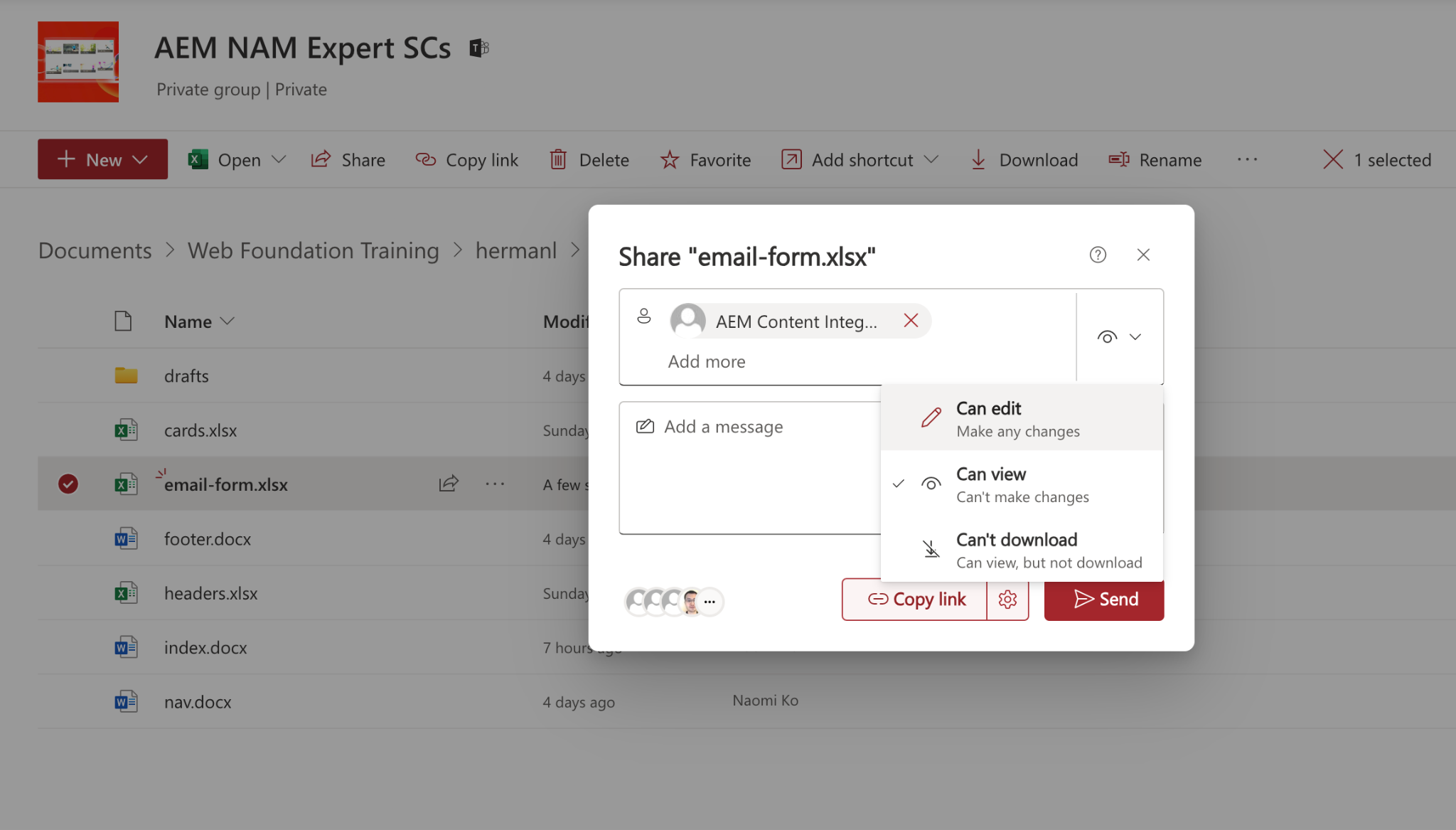
e.g. [https://main--ensembledotcom-](about:blank)<GITHUB USERNAME>.hlx.page/email-form

If you are on Helix 5, “hlx” should be changed to “aem”.

## Give Helix Bot Edit Permission

In this step, you will give the Helix Bot permission to write data into the contact form intake file.

In Sharepoint, select the email-form.xlsx. Share the file with [helix@adobe.com](mailto:helix@adobe.com) with the Can edit option.

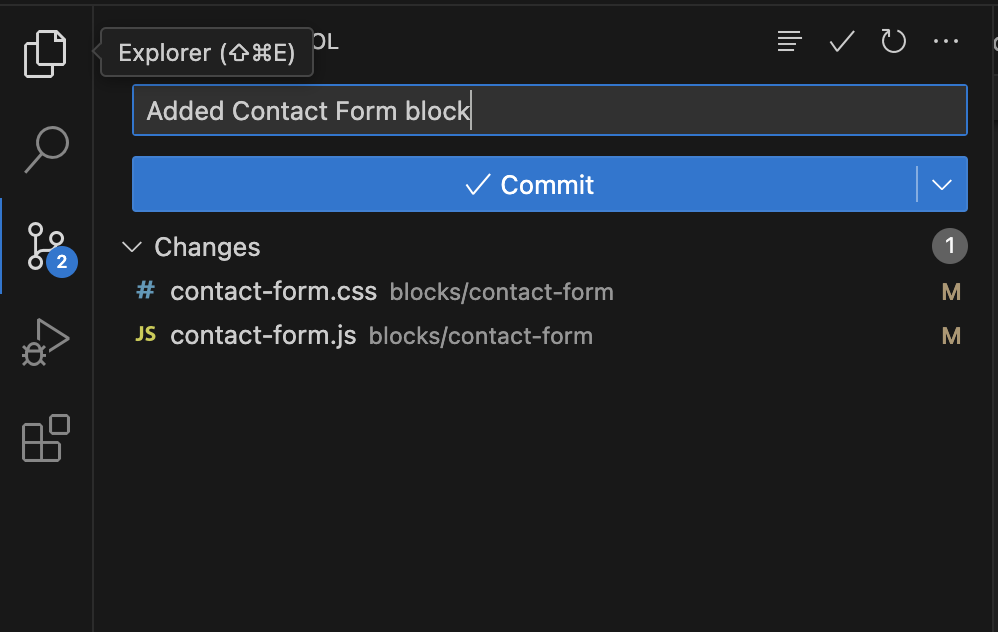


## Commit

In this step, you will push the code into Github. We will skip local testing to avoid troubleshooting CORS issues.

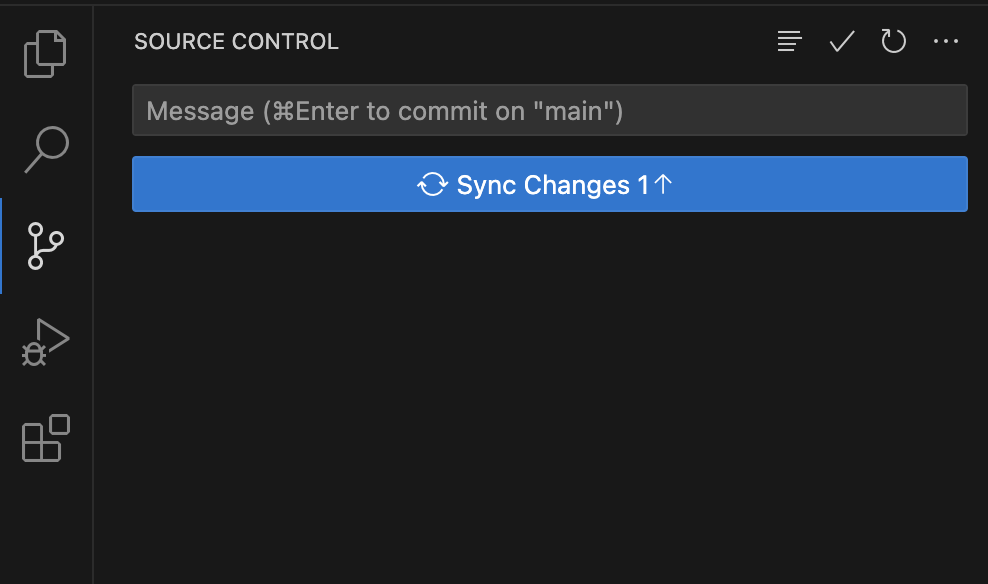
In Visual Studio Code, commit the change to the Github repo. Select the Source Control button in the left rail.

The next set of steps may differ depending on how your settings are.



Enter a git commit message such as “Added Contact Form Block” into the field. (You cannot commit without providing a message)

Select the Commit button.



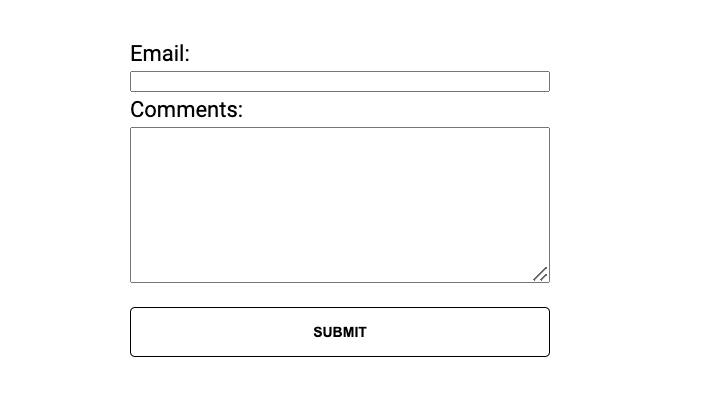
Select the Sync Changes button.

## Test the Contact-Form Block Code.

In this step, you will test the Contact Form Block.

In Sharepoint, preview /drafts/contact-forms.docx to open the page.

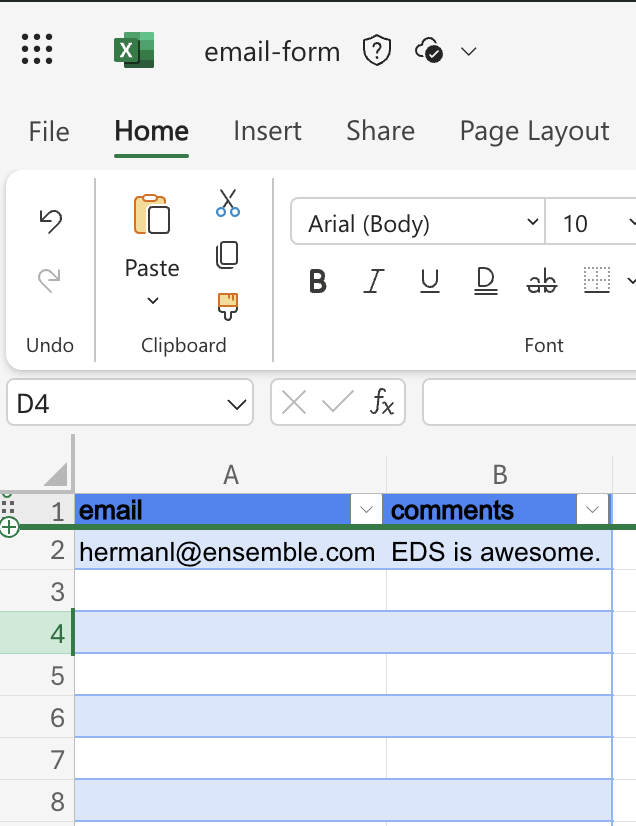
If the URL contains “hlx.live”, skip this step. If the URL contains “aem.live”, change aem to hlx.



Enter a valid email and a comment message. Press Submit.



In Sharepoint, open email-form.xlsx in your ensemble folder. The data that you submitted should be in one of the rows.



## Add the Contact Form content to index.docx

Now that the code works and the content is ready, you can add it to the index.docx.

Since the Contact Form Content is already available in /drafts/contact-form-gql.docx, you can copy it from there into the index.docx. file. Note, this content can be added to any page.

Don’t forget to publish the new content.

## 