

ReadMe for Student Registration demo

This ReadMe will explain how to use this Student Registration demo content.

This demo will involve MORF, Power automate, Sharepoint, PDF Document Generation Service and Adobe Acrobat Sign.

Upon submission from Morf form, it will insert the data into a sharepoint list and merge this data to a word template(stored in a sharepoint folder) using Document Generation API service. These are all processed and automated in Power Automate.

After hitting submit, the end user is then forwarded directly to the signing page in Adobe Acrobat Sign. Then a second signer will counter sign after via opening e-sign request email.

To see a full preview of this demo form, visit

<https://preview.getmorf.io/?form=/demos/education/studentregistration/student%20registration%20json.json#/>

Get Started

1. Acquire a 'Site Key' by going to MORF editor <https://editor.getmorf.io/#/> and clicking Request site key. (Name, email, Organization name and domain info are required)
2. Download the CSV file here
https://raw.githubusercontent.com/aftialabs/morf-preview/main/demos/education/studentregistration/Student%20and%20Parent_Guardian%20List.csv
3. In your Sharepoint Site, create a Sharepoint list by importing this CSV
4. Download the Word file template here
[https://github.com/aftialabs/morf-preview/raw/main/demos/education/studentregistration/Registration%20Form\(tagged\).docx](https://github.com/aftialabs/morf-preview/raw/main/demos/education/studentregistration/Registration%20Form(tagged).docx)
5. Create a Sharepoint folder and add the Registration form word file here

Power Automate setup

Create an automated flow in your power automate and rename any of the steps below to your liking.

Once the flow has been saved, an HTTP POST URL will be generated from the first step and this URL will need to get copy and pasted into the **submit** field in the Student Registration JSON.

Submitting data to the HTTP POST URL will trigger this flow and do the processing/automation of the json data payload. It then respond back to the POST API origin with the Signing URL. Morf will then forward the page from the form straight to the signing experience in Adobe Acrobat Sign.

1. Select 'When an HTTP request is received' on the first step
2. Select 'Initialize Variable' on the second step and enter the following below
 - a. Name: data
 - b. Type: Object
 - c. Value: `json(triggerFormDataURL('data'))`
3. Select 'Initialize Variable' on the third step and enter the following below
 - a. Name: GoalsList
 - b. Type: Array
4. Select 'Parse JSON' on the 4th step
 - a. Content: data
 - b. Schema:
<https://github.com/aftialabs/morf-preview/blob/renies-demo-content-addition/demos/education/studentregistration/schema.json>
5. Select 'Apply to each'
 - a. Select an output from prev steps: goals (under Dynamic Content > Parse JSON)
 - b. Add an action and select 'Append to array variable'
 - i. Name: GOalsList
 - ii. Value: goal (under Dynamic Content > Parse JSON)
6. Select 'Initialize Variable'
 - a. Name: agreementId
 - b. Type: String
7. Add an action and enter Sharepoint then select 'Create item'. (Each item in the list, enter the function code under Expression)
 - a. Site Address: [Pick the Sharepoint Site where your list resides in]
 - b. List Name: [Pick the Sharepoint List you created by importing the CSV file from Get Started, step 3]
 - c. Register Status: Not Registered
 - d. Student First Name: `variables('data')['student']['fname']`
 - e. Student Middle Name: `variables('data')['student']['mname']`
 - f. Student Last Name: `variables('data')['student']['lname']`
 - g. Gender: `variables('data')['student']['gender']`
 - h. Date of Birth: `variables('data')['student']['DOB']`
 - i. Grade: `variables('data')['student']['Grade']`
 - j. Student Cell Phone number: `variables('data')['student']['cellphone']`
 - k. Student email address: `variables('data')['student']['email']`

- l. Current School name: variables('data')?['student']?['currentSchool']
 - m. Future Goals: if(empty(variables('GoalsList')), 'No input', string(variables('GoalsList')))
 - n. Parent/Guardian First name: variables('data')?['pg']?['fname']
 - o. Parent/Guardian Last name: variables('data')?['pg']?['lname']
 - p. Parent/Guardian email: variables('data')?['pg']?['email']
 - q. Parent/Guardian phone: variables('data')?['pg']?['phone']
 - r. Relationship to the Student: variables('data')?['pg']?['relationshipToTheStudent']
 - s. Parent/Guardian 2 First name: variables('data')?['pg2']?['fname']
 - t. Parent/Guardian 2 Last name: variables('data')?['pg2']?['lname']
 - u. Parent/Guardian 2 email: variables('data')?['pg2']?['email']
 - v. Parent/Guardian 2 phone: variables('data')?['pg2']?['phone']
 - w. Relationship to the Student 2: variables('data')?['pg2']?['relationshipToTheStudent']
 - x. Street Address: variables('data')?['streetAddress']
 - y. City: variables('data')?['city']
 - z. Province: variables('data')?['province']
 - aa. Postal Code: variables('data')?['postalCode']
 - bb. How did you hear about us?: variables('data')?['howDidYouHearAboutUs']
8. Select 'Get file content' (Sharepoint) for the 8th step
- a. Site Address: [Select the Sharepoint Site the word template resides in from Get Started, step 5]
 - b. File Identifier: [Select the 'Registration Form(tagged).docx']
9. Select 'Generate document from Word template' (Adobe PDF Services)
- a. Template File Name: Registration Form(tagged).docx
 - b. Merge Data: string(variables('data'))
 - c. Template File Content: File Content (This is under Dynamic content > Get file content)
10. Select 'Upload a document and get a document ID' (Adobe Sign)
- a. File Name: Output File Name (This is under Dynamic content > Generate document from Word template)
 - b. File Content: Output File Content (This is under Dynamic content > Generate document from Word template)
11. Select 'Condition'
- a. Add: variables('data')?['pg2']?['email'] 'is equal to' null
 - b. If Yes
 - i. Select 'Create an agreement from an uploaded document and send for signature' (Adobe Sign)
 - 1. Agreement Name: WE.LEARN - Student Registration Form agreement for @{variables('data')?['student']?['fname']}

@{variables('data')}?['student']?['mname']}]

@{variables('data')}?['student']?['lname']}]

2. Document ID: Document ID (Select this under Dynamic Content > Upload a document and get a document ID)

3. Signature Type: ESIGN

4. For the first recipient, select the 'Switch to input array' icon on the right side and input:

```
{  
  "email": @{variables('data')}?['pg']?['email'],  
  "securityOption/authenticationMethod": "NONE"  
}
```

5. For the second recipient, input any demo account you would like to use as the second signer.

- ii. Select 'Set Variable'

1. Name: agreementId
2. Value: Agreement ID (Select this under Dynamic content > Create an agreement from an uploaded doc and send out for signature)

- c. If No

- i. Select 'Create an agreement from an uploaded document and send for signature' (Adobe Sign)

1. Agreement Name: WE.LEARN - Student Registration Form agreement for @{variables('data')}?['student']?['fname']]
@{variables('data')}?['student']?['mname']]
@{variables('data')}?['student']?['lname']]

2. Document ID: Document ID (Select this under Dynamic Content > Upload a document and get a document ID)

3. Signature Type: ESIGN

4. For the first recipient, select the 'Switch to input array' icon on the right side and input:

```
{  
  "email": @{variables('data')}?['pg']?['email'],  
  "securityOption/authenticationMethod": "NONE"  
}
```

- ii. Select 'Set Variable'

1. Name: agreementId
2. Value: Agreement ID (Select this under Dynamic content > Create an agreement from an uploaded doc and send out for signature)

12. Select 'Do until'

- a. Add:
 - outputs('Retrieve_the_Signing_URL')?['body/signingUrlSetInfos']?[0]?['signingUrls']?[0]?['esignUrl'] 'is not equal to' null
 - b. Select 'Delay'
 - i. Count: 2
 - ii. Unit: Second
 - c. Select 'Scope' and rename this step to Try
 - i. Select 'Retrieve the Signing URL' (Adobe Sign)
 1. Agreement ID: agreementId (Select this under Dynamic content > Variables)
 - d. Select 'Scope' and rename this step to Catch
 - i. Apply any catch method you prefer
13. Select 'Initialize variable'
- a. Name: url
 - b. Type: String
 - c. Value:
 - outputs('Retrieve_the_Signing_URL')?['body/signingUrlSetInfos']?[0]?['signingUrls']?[0]?['esignUrl']
14. Select 'Response' (HTTP)
- a. Status Code: 200
 - b. Headers: Content-Type | application/json
 - c. Body:


```
{
  "successUrl": @{variables('url')}
}
```

Using JSON code to generate the complete MORF form

1. Go to <https://github.com/aftialabs/morf-preview/blob/main/demos/education/studentregistration/student%20registration%20json.json>
2. Copy the JSON code and paste it into the editor in <https://editor.getmorf.io/#/>
3. You will see the full form on the right side
4. Paste the HTTP POST URL from 'Power Automate section, step 1' into the Submit field (Replace the current one)
5. Paste the Site Key from 'Get Started, step 1' into the sitekey field (Replace the current one)
6. Click on Preview at the top and a new tab will open
7. Fill out the form and you can hit Submit to test

Submit

1. Submitting the form will give a Spinning loading icon/message
2. If everything is setup correctly, the user will be brought straight to Adobe Acrobat Sign to complete the first recipient's signature