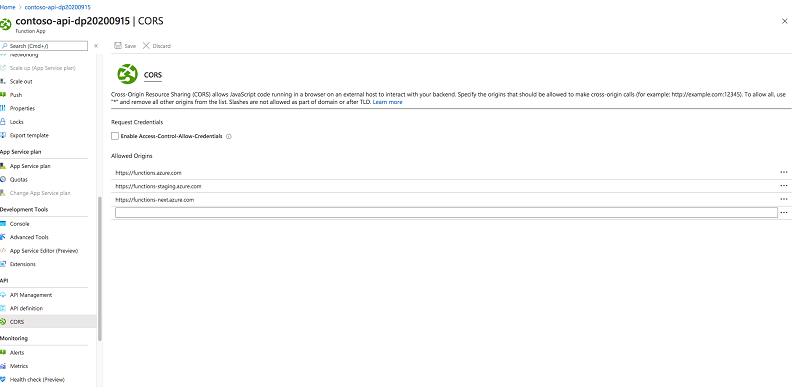
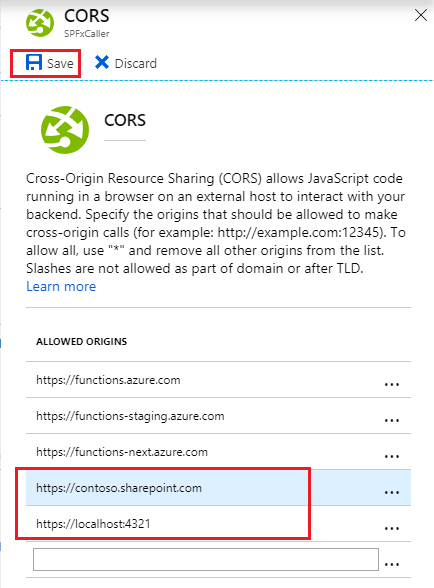
**SPFx#6 - Call Azure Function from SPFx WebPart**

In this article we will see how to call an Azure Function from SPFx WebPart. We assume that you have already created an Azure Function.

**Step 1-** Enable **CORS** on Azure Function. The Azure functions are hosted in MS Azure and they run in a **different domain** than our SharePoint site where our SharePoint Framework (SPFx) web part is hosted. By default**, cross-domain calls** are not allowed from SharePoint. To overcome this, we will have to enable **CORS (Cross-Origin Resource Sharing)** in Azure function. In the Function App, switch to the **Platform features** tab. From the **API** group, select the **CORS** link:



**Step 2-** In the list of allowed origins, add the URL of your M365/ SharePoint tenant, for example, https://contoso.sharepoint.com. Also you can add SharePoint local workbench URL for testing.



**Step 3:** Create SPFx web part and call the Azure function inside it.

Import http module

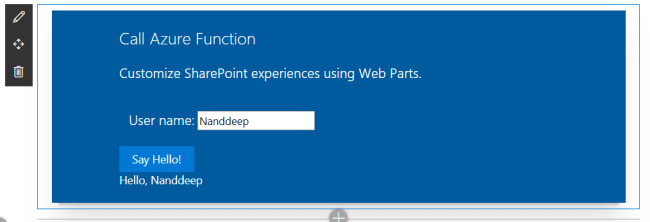
**import** { HttpClient, SPHttpClient, HttpClientConfiguration, HttpClientResponse, ODataVersion, IHttpClientConfiguration, IHttpClientOptions, ISPHttpClientOptions } from '@microsoft/sp-http';

1. **protected** functionUrl: string = "https://spfxcaller.azurewebsites.net/api/HttpTrigger";
2. **protected** callAzureFunction(): **void** {
3. **const** requestHeaders: Headers = **new** Headers();
4. requestHeaders.append("Content-type", "text/plain");
5. requestHeaders.append("Cache-Control", "no-cache");
7. **var** siteUrl: string = **this**.context.pageContext.web.absoluteUrl;
8. **var** userName: string = (<HTMLInputElement>document.getElementById("txtUserName")).value;
9. console.log(`SiteUrl: '${siteUrl}', UserName: '${userName}'`);
10. **const** postOptions: IHttpClientOptions = {
11. headers: requestHeaders,
12. body: `{ name: '${userName}' }`
13. };
14. let responseText: string = "";
15. let resultMsg: HTMLElement = document.getElementById("responseContainer");
16. **this**.context.httpClient.post(**this**.functionUrl, HttpClient.configurations.v1, postOptions).then((response: HttpClientResponse) => {
17. response.json().then((responseJSON: IData) => {
18. //responseText = JSON.stringify(responseJSON);
19. **if** (response.ok) {
20. resultMsg.style.color = "white";
21. } **else** {
22. resultMsg.style.color = "red";
23. }
25. resultMsg.innerText = responseJSON.name;
26. })
27. .**catch** ((response: any) => {
28. let errMsg: string = `WARNING - error when calling URL ${**this**.functionUrl}. Error = ${response.message}`;
29. resultMsg.style.color = "red";
30. console.log(errMsg);
31. resultMsg.innerText = errMsg;
32. });
33. });

**Step 4:** Render method

document.getElementById("btnCallAzureFunction").onclick = **this**.callAzureFunction.bind(**this**);

## Web Part:



Ref: <https://www.c-sharpcorner.com/article/sharepoint-framework-call-azure-function/>