

# Create Embedded Web Experiences with Tabs for Microsoft Teams

팀즈용 탭을 위한 임베디드 웹 환경 만들기



# Resources

- 본 과정의 상세 내용은 아래 모듈에서 확인하실 수 있습니다.
- <https://docs.microsoft.com/ko-kr/learn/modules/embedded-web-experiences/>

# 과정 개요

- 팀즈 앱
  - 구조
- 퍼스널 탭
- 채널 탭
  - 구성페이지
- 탭 인증
  - OAuth 인증 팝업

# 진행 개요

- 내용 강의
  - 3:30 ~ 4:10
- HOL1
  - 4:15 ~ 4:45
- HOL2
  - 4:45 ~ 5:00
- HOL3
  - ????????????

Learning

# Agenda

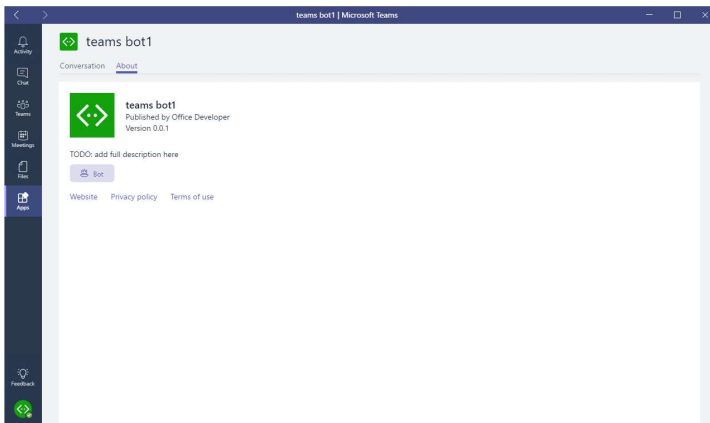
팀즈 확장성 살펴보기

팀즈 탭

팀즈를 위한 사용자지정 탭 만들기



# 팀즈의 확장성



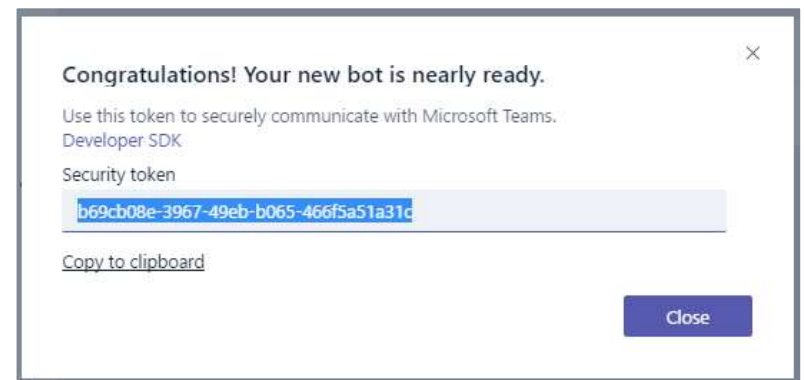
## Microsoft Teams App

Tabs

Bots

Messaging extensions

Webhooks & connectors



## Webhooks & connectors

Receive & reply to messages

# 팀즈 앱이란?

가장 의미 있는 컨텍스트 또는 "범위"에서 사용자가 사용할 수 있는 서비스 패키지

## Services

- Tabs
- Bots
- Messaging extensions
- Webhooks & connectors

## Scopes

- Personal & App bar
- Team & Channel
- Group Chat

## Package & deployment

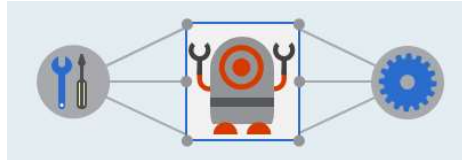
- Upload one Team
- Tenant App Catalog
- Office Store

# 팀즈앱을 만드는 방법



## Develop components

HTML  
TypeScript / JavaScript  
.NET



## Integrate with Teams

Microsoft Teams JavaScript SDK  
Bot Builder SDK - Node.js & C#



## Package and deploy

Host web pages and services  
Create and package manifest  
Upload/publish package



Learning

 Microsoft



# Microsoft Teams Tabs

# 팀즈의 탭이란?

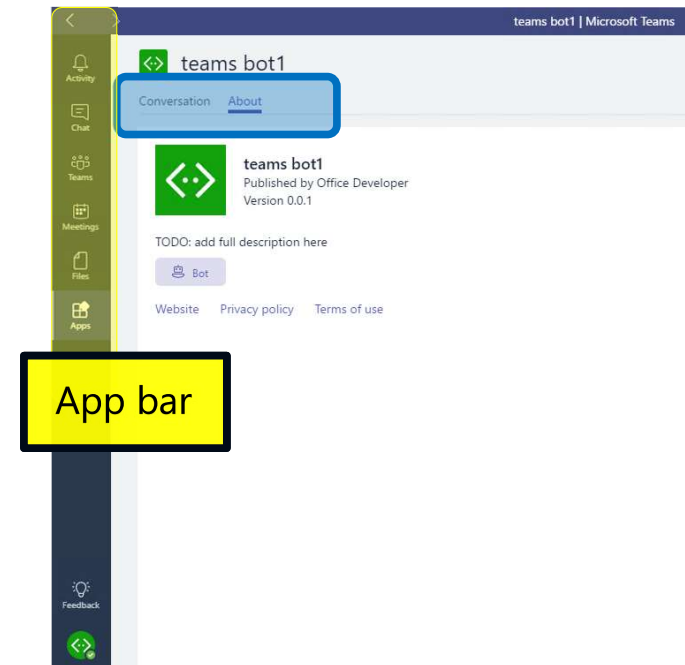
- 개발자가 팀즈 안에서 풍부한 대화형 웹 콘텐츠를 표시할 수 있게 해줍니다.
- 사용자 지정 탭을 만드는 선택사항:
  - 기존의 웹앱 환경을 활용
  - 처음부터 탭을 만듦
- 탭은 응용프로그램의 공급자가 호스트하는 웹응용프로그램
- 탭은 팀즈클라이언트안에서 <iframe>안에 웹응용프로그램을 표시

# 탭은 동일한 콘텐츠를 탐색할 때 웹응용프로그램과는 다르게 동작합니다.

- 탭은 <iframe> 안에 웹 콘텐츠를 표시
  - 팀즈 안에서만 동작하도록 제한 가능
- 인증이 다릅니다
  - 인증 팝업 사용
- 크로스 도메인 탐색이 다릅니다
  - 탭안에 표시될 모든 도메인이 사전정의되어야 합니다.
- 웹응용프로그램 스타일은 팀즈 클라이언트의 기본값, 어두운, 고대비 모드를 적용가능
- 팀즈와 페이지간의 통신은 JS Client SDK 를 사용합니다
- 페이지를 표시하기 위해 microsoftTeams.initialize() 메소드를 호출해야 합니다.

# Tab Scope: 탭을 어디에 위치시킬 것인가?

- Team
  - 팀내의 채널에 추가
  - 채널내의 모든 팀멤버와 공유
  - 이전 이름 : configurable tabs
- Group Chat
  - Team scope 과 비슷하지만 그룹챗안에 사용
  - 2명 이상의 대화에서 활용가능
- Personal
  - 개인사용자에게 제공
  - 앱바에서 접근가능
  - 이전이름 : static tabs



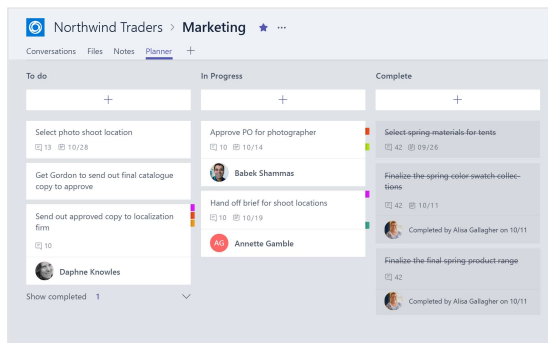
Learning

 Microsoft



# 사용자 지정 탭 개발하기

# 팀즈 탭 개발 개요

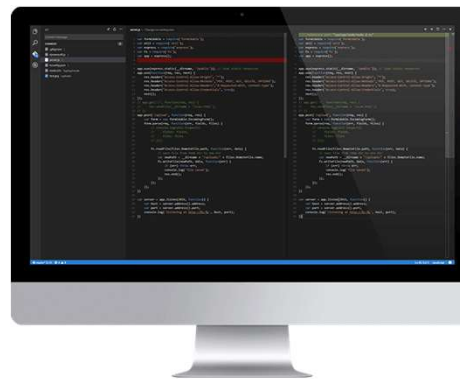


Design a great tab

Select relevant app functionality

Scope and focus the user experience

Integrate and streamline access

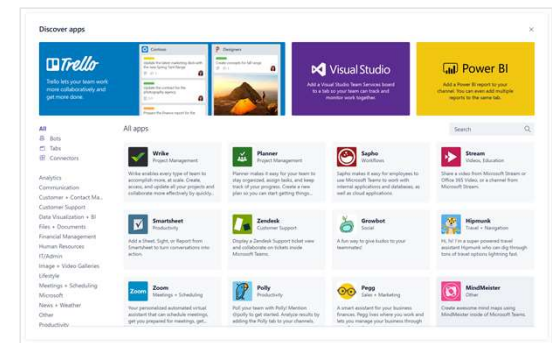


Develop your tab

HTML/JavaScript/TypeScript

REST services

Deep Linking and context



Deploy your tab

Upload to Team

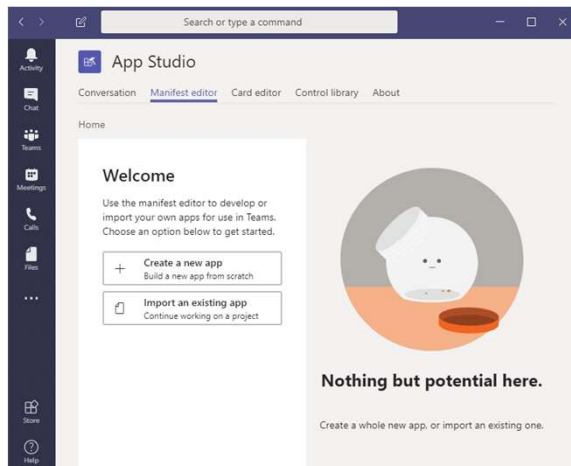
Upload to Tenant App Gallery

Submit to Office Store

# 사용자지정 팀즈 탭: key points

- 탭은 공급자나 개발자가 호스팅하는 웹페이지
- 탭안의 데이터가 팀즈에 저장되지 않음
- 최소한의 구성정보만 팀즈에 저장됨
- React, Angular, Vue.js 등의 모던 웹프레임워크를 사용하여 구현하는 것을 권장
  - 웹페이지가 아닌 웹응용프로그램을 사용자에게 제공

# 개발자 도구 옵션



## App Studio

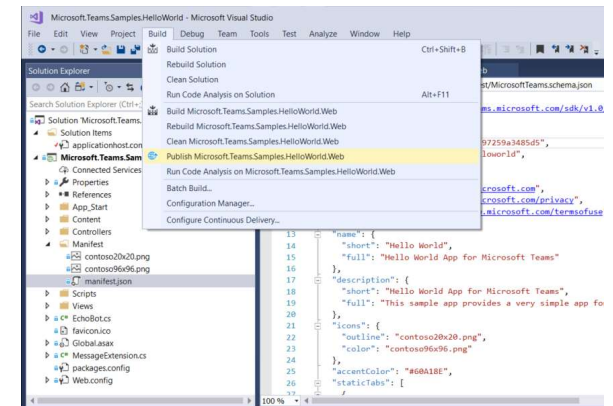
Manifest editor

Preview card rendering

```
? What is your solution name? LearnTeamsTabs
? Where do you want to place the files? Use the current folder
? Title of your Microsoft Teams App project? Learn Teams Tabs
? Your (company) name? (max 32 characters) Contoso
? Which manifest version would you like to use? devPreview
? Enter your Microsoft Partner Id, if you have one? (Leave blank to skip)
? What features do you want to add to your project? (Press <space> to skip) A Tab
? The URL where you will host this solution? https://learnteamstabs.azurewebsites.net
? Would you like to include Test framework and initial tests? No
? Would you like to use Azure Applications Insights for telemetry? No
? Default Tab name? (max 16 characters) LearnTeamsTabs
? Do you want to create a configurable or static tab? Static
```

## Microsoft Teams Yeoman Generator

Ideal for Node.js based projects



## Visual Studio

Ideal for .NET based projects



# 탭내에서 컨텍스트 구현하기

- 상황에 따라 현재 사용자의, 팀의, 회사의 정보가 필요할 수 있습니다.
- 컨텍스트를 얻을 수 있는 두가지 방법
  - URL placeholder values
  - Use Microsoft Teams JavaScript SDK

# 컨텍스트에 접근 - URL placeholder values

- Microsoft Teams replaces placeholders in the content URL when loading web apps as a tab in IFRAMES
- Some values set when configuring a tab or generating deep link
- <https://www.contoso.com/myapp?user={loginHint}&tenant={tid}&theme={theme}>

Properties	Description
{entityId}	The id you supplied on the tab's config page
{subEntityId}	The id you supplied when generating a deep link for specific item within the tab
{loginHint}	Value suitable as login hint for Azure AD
{locale}	Lowercase lang-locale
{theme}	default   dark   contrast
{userPrincipalName}	The user identifier of the current user in the current tenant in email format
{userObjectId}	The Azure AD object ID of the current user in the current tenant
{tid}	GUID id of the current Azure Tenant ID
{groupId}	GUID id of the current Office 365 Group ID

# 컨텍스트에 접근 - Microsoft Teams JavaScript client SDK

- Use the Microsoft Teams JavaScript client SDK to obtain context with Microsoft Teams client
- `microsoftTeams.getContext( function(teamsContext) { ... } );`

Noteworthy Properties	Description
<code>teamId</code>	The team ID in the format 19:[id]@thread.skype
<code>teamName</code>	Name of the current team
<code>channelId</code>	The channel ID in the format 19:[id]@thread.skype
<code>channelName</code>	Name of the current channel
<code>chatId</code>	The chat ID in the format 19:[id]@thread.skype
<code>locale</code>	Lowercase lang-locale
<code>theme</code>	default   dark   contrast
<code>entityId &amp; subEntityId</code>	The entity (& sub entity) id you set up on your config page
<code>userPrincipalName</code>	The user identifier in email format
<code>tid</code>	GUID identifying the current Tenant ID
<code>groupId</code>	GUID identifying the current Office 365 Group ID
<code>tenantSKU</code>	SKU of the current tenant (for example: EDU)

## Learning

# Personal Tabs – Manifest

## staticTabs

`contentUrl` is hosted in IFRAME in Teams

`websiteUrl` is used when the user selects "Open in browser"

## validDomains

valid domains from which the extension expects to load any content



```
"staticTabs": [  
  {  
    "entityId": "candidatesTab",  
    "name": "Candidates",  
    "contentUrl":  
      "https://.../Tabs/file.html",  
    "websiteUrl":  
      "https://.../Tabs/file.html?p=1",  
    "scopes": [  
      "personal"  
    ]  
  }  
],  
"validDomains": [  
  "token.botframework.com"  
]
```

# DEMO

Create a custom Microsoft Teams personal tab

Learning

# Agenda

퍼스널 탭과 채널 탭 비교

채널 탭 기능

구성 페이지

제거 페이지



# 퍼스널 탭과 채널 탭 비교

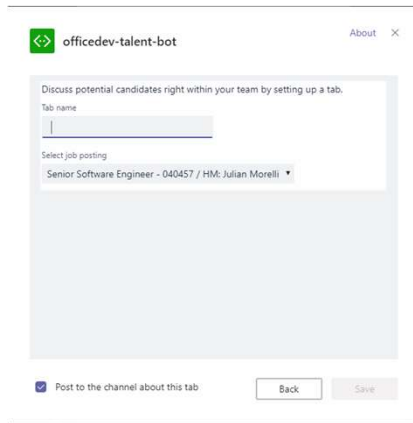
## Personal tabs

- Previously referred to as static tabs
- Scope = personal
- Support individual user
- Not intended to be shared with others in a team
- Displays a web page set in the manifest

## Channel tabs

- Previously referred to as configurable tabs
- Added to a channel
  - Scope = team
- Added as part of a chat between two or more users
  - Scope = groupchat
- Supports all members of a team / channel
- Displays a web page (content page) set in the configuration page

# 탭 구성과 콘텐츠



## Tab Configuration

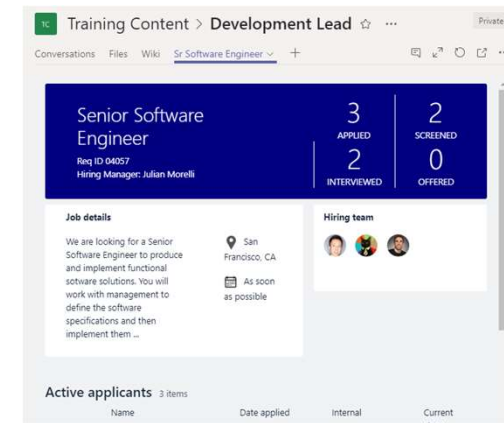
Configured in manifest

Displayed when Tab added to Channel

Displays a web page specified in the app manifest

Collect information

Call `setSettings()` specifying `ContentUrl` and `Entity Id`



## Tab Content

Rendered in IFRAME

Content URL specified by configuration page or manifest

Inspect context for `EntityId/SubEntityId`

Retrieve state based on `Entity/SubEntity/User`



# 채널 탭 매니페스터 구문

```
{  
  ...  
  "configurableTabs": [  
    {  
      "configurationUrl": "https://{HOSTNAME}/configMathTab/config.html",  
      "canUpdateConfiguration": true,  
      "scopes": ["team"],  
      "sharePointPreviewImage": "https://{HOSTNAME}/assets/configMathTab-preview.png",  
      "supportedSharePointHosts": ["sharePointFullPage", "sharePointWebPart" ]  
    }  
  ]  
  ...  
}
```

# 채널 탭 기능 : 구성 페이지

- Displayed after adding a tab to a channel
- Enables user to configure tab before added to a team
- Enable / disable ability to change settings after tab is added to a channel
- Example uses:
  - Specify mapping provider
  - Select how data should be rendered, such as a list or grid
  - Set the `contentURL` and `websiteURL` of a tab

# 채널 탭 기능 : 구성 페이지

Maps

M

Host a map as a tab. Give your tab a name, select Bing Maps or Google Maps, and click save.  
[Learn more](#)

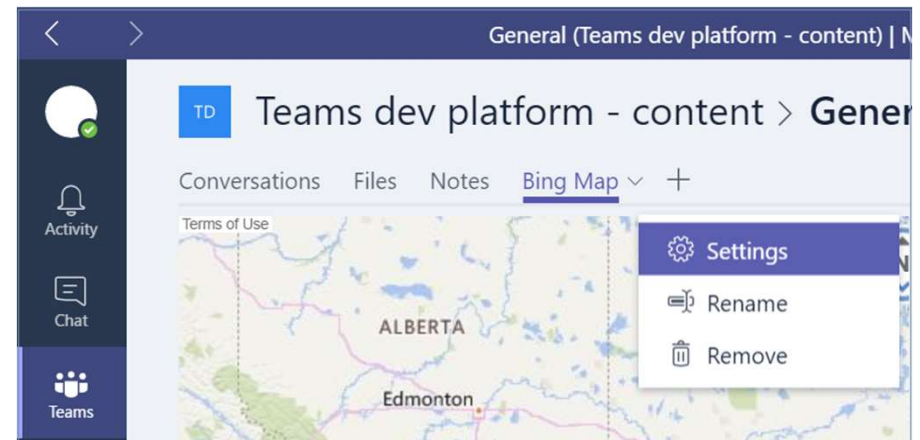
☒ Bing Maps

☐ Google Maps

☒ Post to the channel about this tab

Back

Save



# 구성 페이지 – 구현 및 요구사항

- Like a tab, just a web app loaded within an IFRAME
- Implement the user interface
- Notify Microsoft Teams once settings configured
- Consider: where & how will app save its settings?
- Requirements:
  - Must load within an IFRAME
  - Include Microsoft Teams JavaScript SDK & handle the `registerOnSaveHandler()` event, notifying Microsoft Teams of success

# 구성 페이지에서 설정저장

```
microsoftTeams.settings.registerOnSaveHandler(  
  (saveEvent: microsoftTeams.settings.SaveEvent) => {  
    const host = "https://" + window.location.host;  
  
    microsoftTeams.settings.setSettings({  
      contentUrl: host + "/myTab/?data=",  
      suggestedDisplayName: "My Tab",  
      removeUrl: host + "/myTab/remove.html",  
      entityId: this.state.value  
    });  
  
    saveEvent.notifySuccess();  
  }  
);
```

# 채널 탭 기능 : 제거 페이지

- Displayed when a tab is removed from a channel
- Optional & not required
- Useful when a tab provisioned resources that should be cleaned up when no longer in use

# DEMO

Create a custom Microsoft Teams channel or group tab

Learning

# Agenda

인증 및 팀즈 탭

OAuth 2.0 암시적 부여 흐름

인증 팝업 페이지 흐름

자동 인증 및 SSO

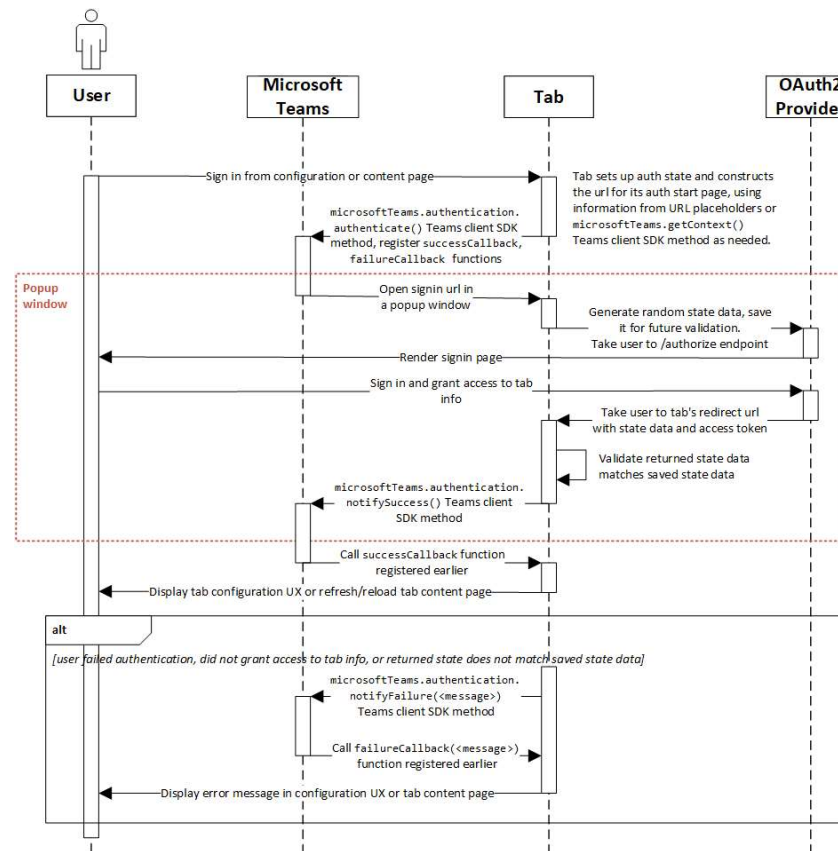




# 인증 및 팀즈 탭

- 탭과 그 안위 웹 응용프로그램은 필요한 경우 인증을 구현할 수 있습니다.
- 탭은 OAuth 2.0 를 이용한 암시적 인증흐름을 사용할 수 있습니다.
- 팀즈가 컨텍스트 힌트를 제공하지만, 그것이 인증에 바로 사용되지는 않습니다.
- 인증흐름은 사용자의 액션에 의해 시작되어야 합니다
  - 인증을 위한 팝업을 자동으로 열지 마십시오- likely to trigger browser popup blockers & confuse the user
  - 인증을 시작하기 위한 버튼을 설정페이지나 콘텐츠 페이지에 위치시키십시오

# OAuth 2.0 암시적 부여 흐름과 팀즈 탭



# 인증 팝업 페이지 흐름

- 팀즈에게 인증프로세스를 시작하도록 합니다

```
microsoftTeams.authentication.authenticate({options})
```

- 팝업이 열리고, 인증 프로바이더정보에 등록된 리다이렉트 URL로 이동합니다.
- 리다이렉트된 URL은 인증이 성공했는지 실패했는지 응답을 줍니다.
- 팀즈에 인증 성공/실패를 알립니다
- Predefined authentication failure reasons:
  - **CancelledByUser**: popup closed before completing the authentication flow
  - **FailedToOpenWindow**: popup couldn't be opened

# 인증 팝업 페이지 요구사항

- Primary goal of page: redirect user to your configured identity provider for user to sign in
- Redirection can also be done via:
  - Server-side: HTTP 302
  - Client-side: `window.location.assign(authEndpoint)`
- Consider using the tab's context to provide a login hint to the identity provider  
`msTeamsContext.userName`
- Consider including the `state` property option
  - Ensure malicious user is not involved in the authentication flow
  - Used to prevent CSRF attacks

# 인증 흐름 예제: 사용자 작업에서 팝업 트리거

```
private async getAccessToken(promptConsent: boolean = false): Promise<string> {  
    return new Promise<string>((resolve, reject) => {  
        microsoftTeams.authentication.authenticate({  
            url: window.location.origin + "/auth-start.html",  
            width: 600, height: 535,  
            successCallback: (accessToken: string) => {  
                resolve(accessToken);  
            },  
            failureCallback: (reason) => {  
                reject(reason);  
            }  
        });  
    });  
}
```

# 인증 흐름 예제: auth-start.html

```
let config = {
  clientId: "<<AzureAD AppID/ClientID",
  redirectUri: window.location.origin + "/auth-end.html",
  cacheLocation: "localStorage"
};
// override URL to AzureAD auth endpoint to include extra query params
config.displayCall = function (urlNavigate) {
  if (urlNavigate) {
    urlNavigate += "?scope=open+profile";
    window.location.replace(urlNavigate);
  }
}
let authContext = new AuthenticationContext(config);
authContext.login();
```

# 인증 흐름 예제: auth-end.html

```
// <omitted>configure ADAL.js & obtain AuthenticationContext</omitted>
if (authContext.isCallback(window.location.hash)) {
    authContext.handleWindowCallback(window.location.hash);
    if (window.opener) {
        if (authContext.getCachedUser()) {
            authContext.acquireToken("https://graph.microsoft.com", function (error, token) {
                if (token) { microsoftTeams.authentication.notifySuccess(token); }
                else if (error) { microsoftTeams.microsoftTeams.notifyFailure(error); }
                else { microsoftTeams.authentication.notifyFailure("UnexpectedFailure"); }
            });
        } else { microsoftTeams.authentication.notifyFailure(authContext.getLoginError()); }
    }
}
```

# 자동인증 과 SSO

- Azure AD 에서의 자동인증은 간소화된 형태의 SSO 입니다
  - Goal: 앱을 사용하는 동안 사용자가 로그인 자격 증명을 입력해야 하는 횟수를 최소화
- 클라이언트측의 인증을 유지하고 토큰을 자동으로 얻기 위해 ADAL.js 를 사용합니다.
- ADAL.js OAuth2 암시적 인증흐름을 위한 숨은 iframe을 사용합니다.
  - Specifies no never prompt user (using `prompt=none`)
- 사용자 상호작용이 필요하면, Azure AD 에서는 ADAL.js에서 처리할 수 있는 에러를 반환합니다.
  - In these scenarios, your app would initiate an interactive login



# DEMO

Implement authentication in a custom tab

# Thank You

