

Security and Authentication for Custom Applications

Lauren Boyd & George Owen

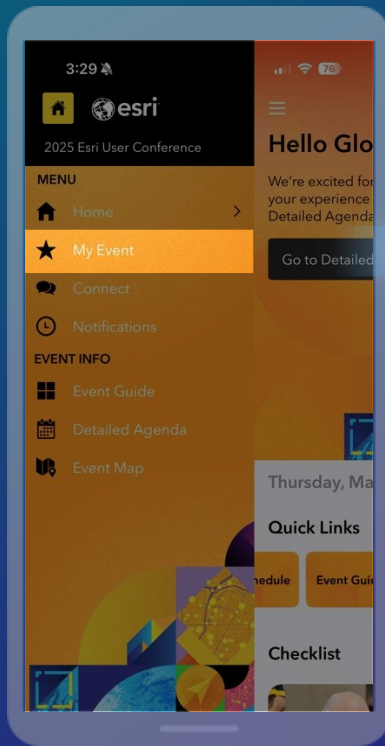
```
Map map = new Map(BasemapStyle.ArcGISNavigation);  
  
map.InitialViewpoint = new Viewpoint(40.72,  
-73.995,  
72000);  
  
mapView.Map = map;
```

```
class MainActivity : AppCompatActivity() {  
    override fun onCreate(savedInstanceState: Bundle?) {  
        super.onCreate(savedInstanceState)  
        setContentView(  
            SampleAppTheme {  
  
                val map = ArcGISMap(NavigationStyle)  
  
                map.InitialViewpoint = Viewpoint(  
                    40.72,  
                    -73.995,  
                    72000.0
```

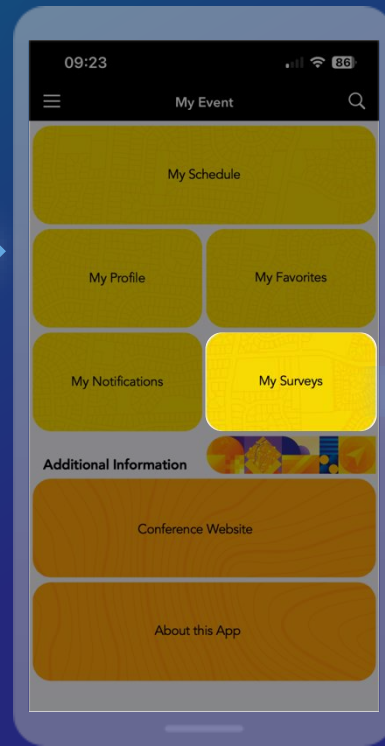


Please share your feedback in the app

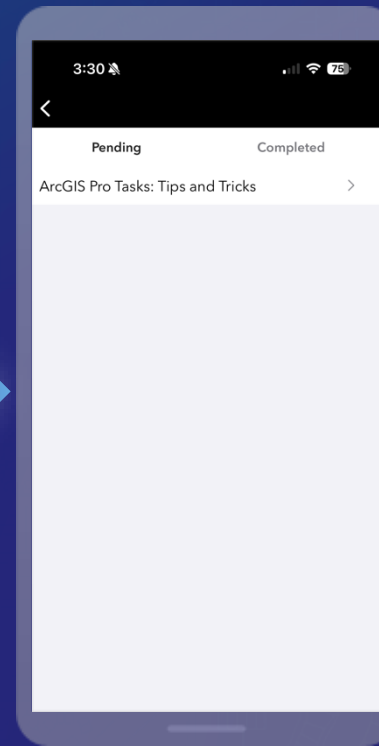
Select "My Event" in the event app main menu



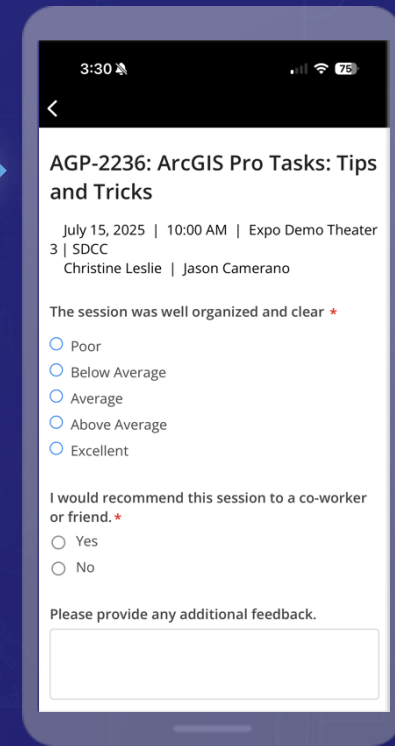
Select "My Surveys"



Select the desired survey



Complete the survey



Agenda

- Security & Auth fundamentals
- API key authentication
- User authentication
- App authentication
- Comparison
- Questions

Security fundamentals

George Owen

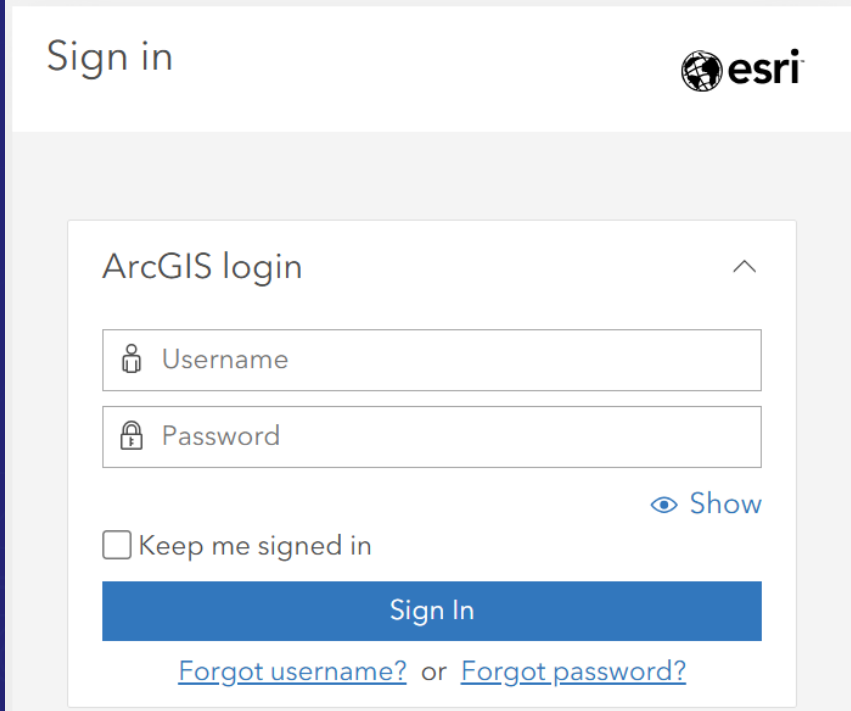


Security matters

- Digital security is paramount in the modern age
- Especially important for app developers
- Consider data privacy, user privacy in apps
- Apps that use ArcGIS need to be secure

Security in ArcGIS

- Sign-in required for all ArcGIS products (ArcGIS Pro, ArcGIS Online, Map Viewer, etc)
- ArcGIS portals are FedRAMP compliant, TRUSTe Certified Privacy Seal, many other certifications
- Organization admins:
 - Manage org security
 - Manage data sharing
 - Manage roles and privileges
- trust.arcgis.com
- [More on portal security](#)



The screenshot shows the ArcGIS login page. At the top left, it says "Sign in" and at the top right is the Esri logo. Below this is a white box titled "ArcGIS login" with an upward arrow icon. Inside the box, there are two input fields: "Username" with a person icon and "Password" with a lock icon. To the right of the password field is a "Show" link with an eye icon. Below the input fields is a checkbox labeled "Keep me signed in". At the bottom of the box is a blue "Sign In" button. Below the button are two links: "Forgot username?" and "Forgot password?".

Your app's security requirements

Authentication methods depend on your use case. Consider:

- Sharing level
 - Public
 - Organization
 - Private to owner
- Sensitivity of your data
- ArcGIS services and resources used

Authentication?

- **Authentication:** The process of verifying the identity of a user or application
- Secure ArcGIS resources require authentication to access

```
curl https://geoenrich.arcgis.com/arcgis/rest/services/World/geoenrichmentserver/Geoenrichment/enrich?  
-d "dataCollections=Health" \  
-d "f=json" \  

```



```
{"error":{"message":"Token required.", "code":499, "details":[]}}
```

Token-based authentication

- **Access token:** Authorization string that provides access to ArcGIS content
- Associated with an account
- Required in API requests to secure resources

```
curl https://geoenrich.arcgis.com/arcgis/rest/services/World/geoenrichmentserver/Geoenrichment/enrich?
-d "dataCollections=Health" \
-d "f=json" \
-d "token=<ACCESS_TOKEN>"
```



```
{
  "results": [
    {
      "paramName": "GeoEnrichmentResult",
      "dataType": "GeoEnrichmentResult",
      "value": {
        "version": "2.0",
        "FeatureSet": [
          {
            "displayName": "",
            "fieldAliases": {...},
            "spatialReference": {...},
            "fields": [...],
            "features": [
```

Secure ArcGIS resources

Resources that require an access token:

- ArcGIS Location Services
- Items and data services with a sharing level of **Private, Group, or Organization**
- ArcGIS Portal (management tasks)
- Subscriber content in ArcGIS Living Atlas
- ... and more



Privileges

- Access tokens have **privileges** associated with them
- Determines the resources a token is authorized to access
- Assigned using developer credentials
 - For user auth: Assigned using the signed-in user's account

Privileges

Grant privileges to allow your access token to perform specific operations.

🔍 Search privileges

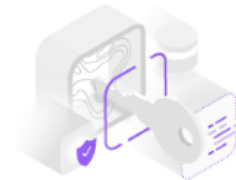
- > **Location services**
Enabled: 0/15
- > **Spatial Analysis**
Enabled: 0/1
- > **General privileges**
Enabled: 0/20 Personal
- > **Admin privileges**
Enabled: 0/18 Personal

How do I get access tokens?

1. Sign up for an account
2. Choose a **type of authentication**
3. Create developer credentials
4. Implement authentication in your application

- [Get started here](#)

 Guide



Security and authentication

Learn how to access secure content and services with API keys, OAuth 2.0, and user authentication.

Supported ArcGIS accounts



Account type	Requirements
ArcGIS Location Platform	Free
ArcGIS Online	Creator user type or higher
ArcGIS Enterprise	Creator user type or higher

Types of authentication



API key authentication

- Embed long-lived API key access tokens in your app
- Token linked to the developer's account
- Tokens embedded in application



User authentication

- Require app users to sign in with an ArcGIS account
- Tokens linked to the accounts of signed-in users
- Tokens created on-demand



App authentication

- Request access tokens on your server and send to client app
- Tokens linked to developer's account
- Tokens created on-demand

Developer credentials

- Item in your ArcGIS portal
- Manage authentication settings
- Required for all types of authentication



Developer credentials →

Create API key and OAuth 2.0 credentials to build custom applications.

Overview

Summary
API key credentials for use in a public application, created with ArcGIS Location Platform.

Status
API key credentials - Standard

Owner
GO George Owen

Sharing
Owner

Item updated
Dec 29, 2025

- Long beach safety sketch
- My OAuth 2.0 credentials (for app authentication)
- My OAuth 2.0 credentials (Standard)
- My API key credentials (for a public app)
- Navigation w/ Metro stations
- basemaps for sessions
- basemaps key
- user auth maplibre testing
- personal app auth

API key authentication

Lauren Boyd



What is API key authentication?

- Long-lived access token (up to a year)
- Embedded directly into an application
- Easy to create and use
- Set privileges to access ArcGIS service

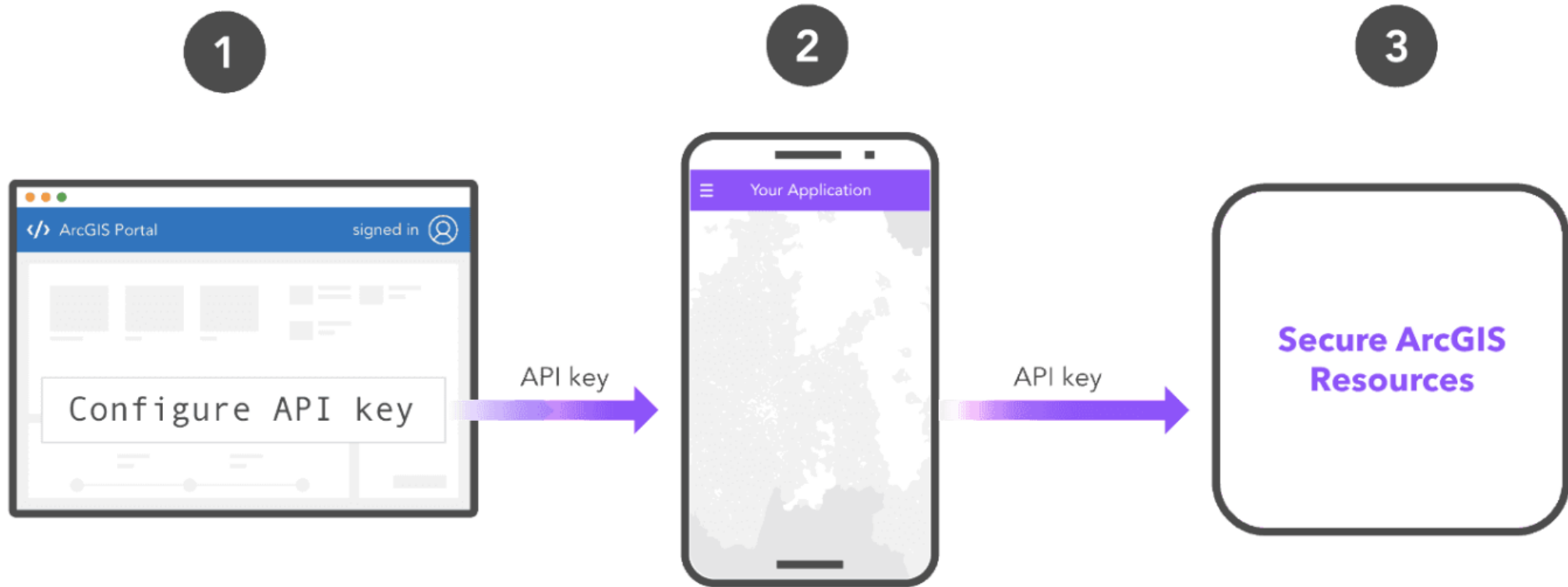
```
const apiKey = 'YOUR_API_KEY';  
// Create a new basemap from the basemap  
// styles service using an API key  
const basemap = new Basemap({  
  style: new BasemapStyle({  
    apiKey: apiKey,  
    id: 'arcgis/human-geography',  
    language: 'es',  
  }),  
});
```

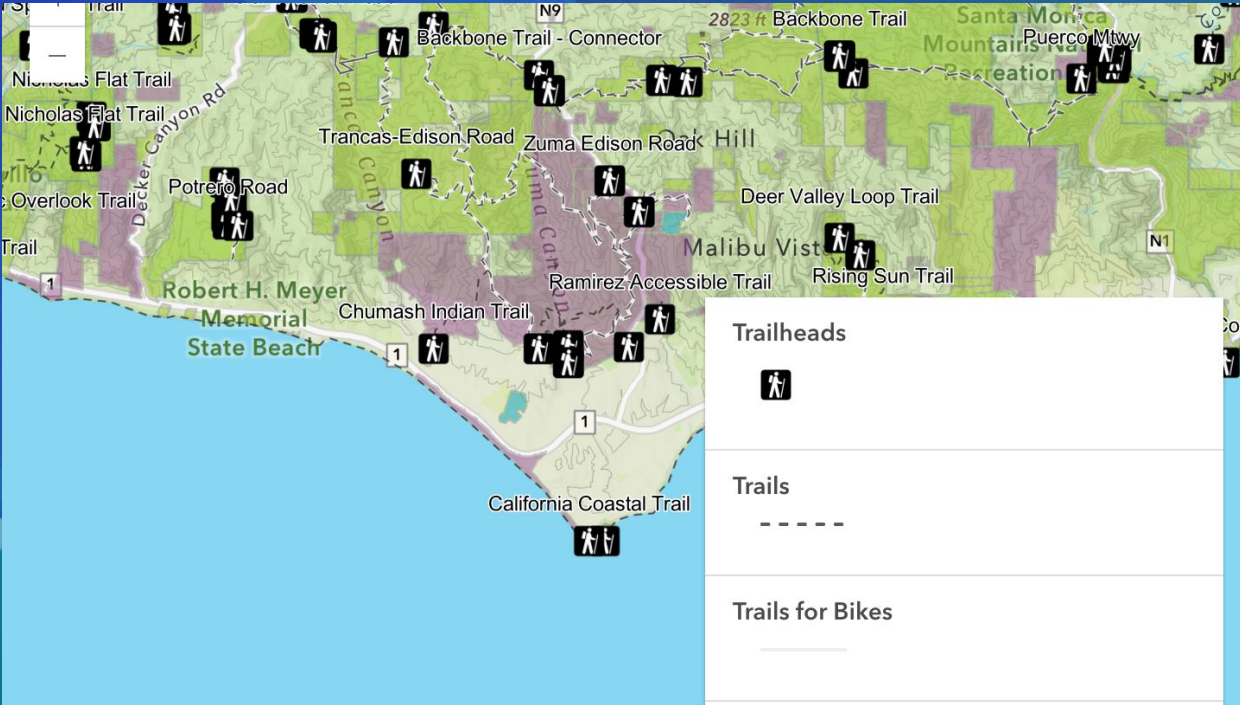
sample

API key authentication: Account prerequisites

Account type	Requirement
ArcGIS Location Platform	None
ArcGIS Online	<ul style="list-style-type: none">• Creator user type or higher• Custom role with developer privileges
ArcGIS Enterprise (starting at 11.4)	<ul style="list-style-type: none">• Creator user type or higher• Custom role with developer privileges

Workflow





Demo: Create and use an API key

Lauren Boyd

API key authentication: Best practices

- Create short-lived API keys
- Create separate keys for each app
- Monitor usage of all keys
- Rotate API keys frequently and programmatically
- Limit the privileges of keys in public apps - anyone can scrape your key!
- Do not access sensitive data in public apps
- Set referrer URLs

User authentication


George Owen



What is user authentication?

- **Require users of your app to sign in with an ArcGIS account**
- Generates a unique access token each time a user signs in
- Usage costs of session are billed to the signed-in account

George's custom app wants to access your ArcGIS Online account information [?](#)





Sign in 

ArcGIS login [^](#)

Keep me signed in

[Forgot username?](#) or [Forgot password?](#)

Your ArcGIS organization's URL [v](#)

[Privacy](#)

User authentication: Account requirements

Account type	Requirement
ArcGIS Location Platform	n/a
ArcGIS Online	Creator user type or higher
ArcGIS Enterprise	Creator user type or higher

Advantages of user authentication

- Perfect for apps within an organization
- Strong integration with ArcGIS Maps SDKs
- Access services on behalf of the signed-in user: items, data, etc
- Advanced usage monitoring capabilities
- Full control of token lifespan, refresh tokens
- Full compliance with OAuth 2.0 security protocol



OAuth 2.0 authorization

- OAuth 2.0: Industry standard authorization protocol
- Framework for how apps should request access to resources
- ArcGIS supports these flows:
 - **Authorization code flow with PKCE ****
 - Authorization code flow
 - Implicit flow (deprecated)
 - Client credentials -> App authentication

- Documentation

How to implement user authentication

1. Create OAuth credentials
2. Add the client ID to your app
3. Authenticate with IdentityManager / AuthenticationManager from ArcGIS Maps SDKs
4. Make authenticated requests

```
const [OAuthInfo, esriId] = await $arcgis.import([
  "@arcgis/core/identity/OAuthInfo.js",
  "@arcgis/core/identity/IdentityManager.js",
]);
var oAuthInfo = new OAuthInfo({
  appId: "YOUR-CLIENT-ID",
});
esriId.registerOAuthInfos([oAuthInfo]);
```

User Authentication Dev Credentials wants to access your ArcGIS Online account information ?

Sign in 

ArcGIS login ^

[Show](#)

Keep me signed in

Sign In

Cancel

[Forgot username?](#) or [Forgot password?](#)

Your ArcGIS organization's URL v



[Privacy](#)

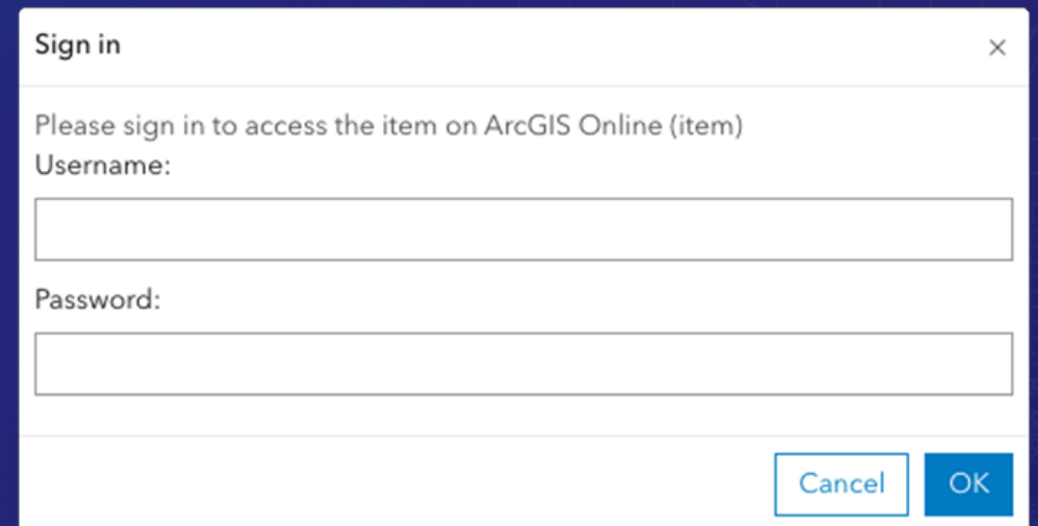
User Authentication

Lauren Boyd

Automatic User Auth with ArcGIS Maps SDKs

- Automatic prompt when loading secure resource in Maps SDK applications
- Type of user authentication that uses the generateToken endpoint to create a token

Automatic user authentication is only recommended in development and testing environments. Implement OAuth in production applications.



The image shows a 'Sign in' dialog box with a close button (X) in the top right corner. The text inside the dialog reads: 'Please sign in to access the item on ArcGIS Online (item)'. Below this text are two input fields: 'Username:' and 'Password:'. At the bottom right of the dialog are two buttons: 'Cancel' and 'OK'.

User authentication: Best practices

- The most secure authentication method if your use case allows it
- Use an ArcGIS Maps SDK or API to implement
- If implementing manually, use OAuth 2.0 authorization code grant with PKCE
- Keep expiration time of access tokens very short
- Cache refresh tokens to avoid repeated sign-ins

App Authentication

Lauren Boyd



What is App authentication?

- Temporary access tokens based on a set of OAuth credentials.
- Tokens valid for 2 weeks unless revoked
- Access tokens are associated with your account
- Request access tokens programmatically using an OAuth 2.0 flow
- Uses a **Client ID** and **Client Secret**
- Create OAuth credentials item to control privileges
- Only for use in console scripts or apps w/ a server-side component

App authentication: Account requirements

Account type	Requirement
ArcGIS Location Platform	n/a
ArcGIS Online	<ul style="list-style-type: none">• Creator user type or higher• Custom role with developer privileges
ArcGIS Enterprise (starting at 11.4)	<ul style="list-style-type: none">• Creator user type or higher• Custom role with developer privileges

App authentication: Best practices

- Treat your `client_secret` like your password
 - Limit the privileges of tokens in public apps
 - Harden server-side component for your use case
 - Use caution when accessing data in public apps
-
- Short-lived API keys + rotation is a viable alternative in some cases

Auth Comparison

George Owen



What type of auth should I use?

Ask yourself:

- Who is your target audience?
 - If your users have ArcGIS accounts, pick **User authentication**
- What ArcGIS product do you have?
- Do you have sensitive data? Other security requirements?

Authentication Recommendations

Account	Recommendation
ArcGIS Location Platform	<ul style="list-style-type: none">- Location services only: API key authentication- Private items or data: App authentication
ArcGIS Online	<ul style="list-style-type: none">- User authentication- For public apps: App authentication
ArcGIS Enterprise	<ul style="list-style-type: none">- User authentication- For public apps: App authentication

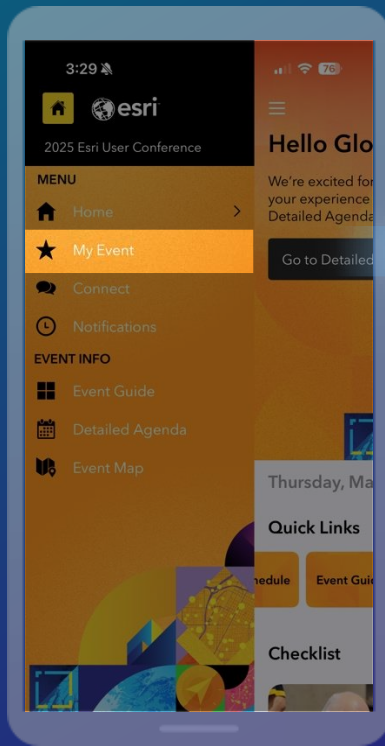
Resources

- <https://github.com/EsriDevEvents/security-and-authentication-for-custom-applications-2026>
- [Get started here](#)
- trust.arcgis.com
- [More on portal security](#)

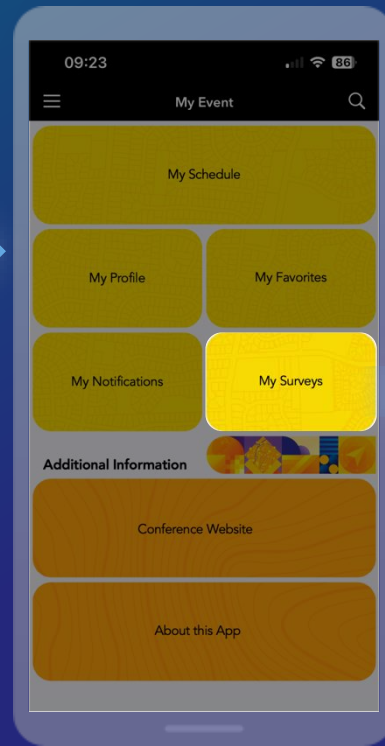


Please share your feedback in the app

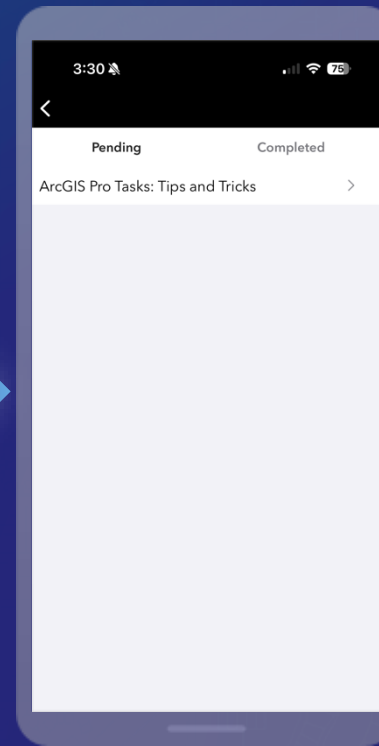
Select "My Event" in the event app main menu



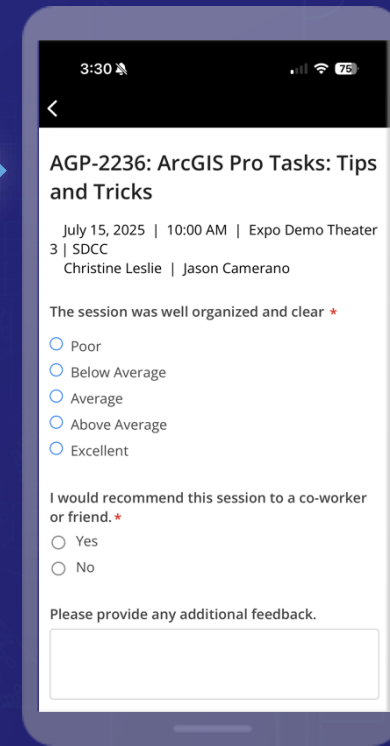
Select "My Surveys"



Select the desired survey



Complete the survey



Connect with us on social

Join the conversation using [#EsriDevTech2026](#)



x.com/EsriDevs



x.com/EsriDevEvents



youtube.com/@EsriDevs



links.esri.com/DevVideos



github.com/Esri



github.com/EsriDevEvents



links.esri.com/EsriDevCommunity



Questions





```
map = new Map(BasemapStyle.ArcGISNavigation);  
map.initialViewpoint = new Viewpoint(40.72,  
-73.995,  
72000);  
mapView.Map = map;
```

```
struct ContentView: View {  
    @State private var map = {  
        let map = Map(  
            basemapStyle: .arcGISNavigation  
        )  
  
        map.initialViewpoint = Viewpoint(  
            latitude: 40.72,  
            longitude: -73.995,  
            scale: 72_000  
        )  
  
        return map  
    }()  
  
    var body: some View {  
        MapView(map: map)  
    }  
}
```



esri[®]

**THE
SCIENCE
OF
WHERE**[®]