Esri International Developer Summit

Palm Springs, CA

Security and ArcGIS Web Development

Jeremy Bartley, Keyur Shah

Agenda

- Classic Logins (`generateToken`)
- OAuth2 User Logins
  - Registered Apps
  - Implicit Grants (1-Step Flow)
  - Authorization Grants (2-Step Flow)
  - Enterprise Logins (SAML)
  - Single Sign On
  - SDK Support (Identity Manager)
- OAuth2 App Logins
  - Workflow
  - Proxy Use Case
- Conclusion

Agenda

- **Classic Logins (`generateToken`)**
- OAuth2 User Logins
  - Registered Apps
  - Implicit Grants (1-Step Flow)
  - Authorization Grants (2-Step Flow)
  - Enterprise Logins (SAML)
  - Single Sign On
  - SDK Support (Identity Manager)
- OAuth2 App Logins
  - Workflow
  - Proxy Use Case
- Conclusion

Classic Logins (`generateToken`)

- Make a `generateToken` API call from the app
- Send username and password over https
- Returns a short lived token in response

- Use the Identity Manager Control in the client SDKs instead of invoking the API directly

`generateToken` Drawbacks

- App has access to user's password
- App is responsible for the login dialog
  - Identity Manager mitigates this
- Does not support enterprise logins
- Does not support Single Sign On
- No platform support for app usage tracking
- Cannot be listed in the Marketplace

Agenda

- Classic Logins (`generateToken`)
- **OAuth2 User Logins**
  - Registered Apps
  - Implicit Grant (1-Step Flow)
  - Authorization Grant (2-Step Flow)
  - Enterprise Logins (SAML)
  - Single Sign On
  - SDK Support (Identity Manager)
- OAuth2 App Logins
  - Workflow
  - Proxy Use Case
- Conclusion

Registered Apps

- Must register app with the portal for OAuth support
- Apps can be registered from the organization portal site or the developers site
- An App ID (`client_id`) and App Secret (`client_secret`) are generated on registration
- Benefits
  - App ID is encoded in all OAuth tokens
  - App usage tracking supported by the platform
  - Can be listed in the Marketplace
  - Can set access controls on users in Marketplace apps

User Login – Implicit Grant

- Login workflow is completed in 1 Step
- Used by browser-based apps
  - JavaScript, Flex, Silverlight

User Login – Implicit Grant (1 Step Flow)

- `/authorize?response_type=token&...`
  - Returns `access_token` on successful authorization
  - Completes the login flow

- Demo App

User Login – Authorization Grant

- Login workflow is completed in 2 Steps
- Used by mobile, desktop and server-side web apps

User Login – Authorization Grant (2-Step Flow)

- `/authorize?response_type=code&...`
  - Returns authorization code on successful authorization
- `/token?grant_type=authorization_code&code=...`
  - Returns `access_token` and `refresh_token`
  - Completes the login flow
- Refreshing tokens
  - `/token?grant_type=refresh_token&refresh_token=...`
    - Returns a new `access_token`
- Demo App

Enterprise Logins

- Login to ArcGIS Online using your enterprise login (Active Directory, LDAP, …)
- Uses the SAML standard
- Setting up Enterprise Logins – Doc

- Nothing changes for the App Developer
  - Use standard OAuth workflow (redirect user to `/authorize` URL as usual)
  - Portal detects enterprise login if configured for the organization
  - Redirects user to their enterprise provider
  - Enterprise redirects user to portal upon login
  - Portal generates token and sends it to the app

Single Sign On

- OAuth enables server-side login
- If user already has an active portal session
  - User is not required to enter credentials again
  - Presented with an approval screen to grant access to the app
- Again, nothing changes for the app developer

Client SDK Support (Identity Manager)

- All Client SDKs support an Identity Manager
- Simplifies login workflow for the app developer
- Handles all credential management for application lifecycle

Agenda

- Classic Logins (`generateToken`)
- OAuth2 User Logins
  - Registered Apps
  - Implicit Grants (1-Step Flow)
  - Authorization Grants (2-Step Flow)
  - Enterprise Logins (SAML)
  - Single Sign On
  - SDK Support (Identity Manager)
- **OAuth2 App Logins**
  - **Workflow**
  - **Proxy Use Case**
- Conclusion

App Login – Use Cases

- App uses secured portal resources
  - Allows anonymous user access
  - Users are not known to the ArcGIS Portal
- Back-office routines

App Login Workflow

- `/token?`
  `grant_type=client_credentials&`
  `client_id=...&`
  `client_secret=...`

  - Returns an app `access_token` on successful validation
  - Completes the login flow

App Logins – Proxy Use Case

- App uses secure resources
  - Allows anonymous user access
  - Users are not known to the ArcGIS Portal
- Steps
  - Register App
  - Configure proxy with app credentials
  - Proxy uses app credentials to get app token from the portal
  - Front-end app calls into the proxy
  - Proxy uses app token to call into secured portal resources
  - Proxy returns results to the app

Agenda

- Classic Logins (`generateToken`)
- OAuth2 User Logins
  - Registered Apps
  - Implicit Grants (1-Step Flow)
  - Authorization Grants (2-Step Flow)
  - Enterprise Logins (SAML)
  - Single Sign On
  - SDK Support (Identity Manager)
- OAuth2 App Logins
  - Workflow
  - Proxy Use Case
- **Conclusion**

Conclusion

- Use OAuth2 for user logins
- Benefits include usage tracking, enterprise logins, etc.
- Identity Manager simplifies login workflow in client SDKs
- Use app tokens to access secured resources in certain use cases


esri
Understanding our world.