



Aero App Data API

Aero App for Android

Publication Date: September 4, 2026

Table of Contents

1 Introduction.....	4
2 Getting Started	4
2.1 Gradle Dependencies.....	4
2.2 Authentication and Data Access	4
2.3 Make your first API call	5
2.3.1 Retrieve active cycle information	5
2.3.2 Retrieve airport information.....	6
3 Functions	10
3.1 checkAeroAppPermission	11
3.2 requestAeroAppPermission	11
3.3 getActiveCycle.....	12
3.4 getAeroAppVectorMapInfo	13
3.5 getAeroAppVectorTile	14
3.6 getAirportCharts	16
3.7 getAirports	17
3.8 getAreaPlanning.....	19
3.9 getAreaPlanningList	20
3.10 getARR	21
3.11 getArtcc.....	22
3.12 getChartSupplement.....	23
3.13 getChartSupplementsList.....	24
3.14 getGiantReport	25
3.15 getMtr	26
3.16 getNav aids	27
3.17 getParachuteJumpingArea.....	28

3.18 getSUA	29
3.19 getVO	30
3.20 getWaypoints	31
4 Classes	32
4.1 AeroAppAirport	34
4.2 AeroAppAirportCommunication	37
4.3 AeroAppAirportRemark	39
4.4 AeroAppARR	40
4.5 AeroAppARTCC	42
4.6 AeroAppBoundingBox	44
4.7 AeroAppMTR	45
4.8 AeroAppMTRRemark	48
4.9 AeroAppNavaid	49
4.10 AeroAppPJA	51
4.11 AeroAppResult	54
4.12 AeroAppRunway	55
4.13 AeroAppRunwayEnd	57
4.14 AeroAppSUA	59
4.15 AeroAppVO	61
4.16 AeroAppWaypoint	63
4.17 AeroAppAirportAQP	65
4.18 AeroAppCycle	66
4.19 AeroAppMapInfo	67
4.20 AeroAppPDFDocument	68
4.21 AirportChartData	69
4.22 AirportData	70
5 Enums	71

5.1 AeroAppVectorTileLayer	71
5.2 AirportChartType	73
5.3 ColorMode	74

1 Introduction

This document describes the Data API, which provides access to aeronautical data through a library for Android applications. The library communicates with a ContentProvider supplied by Aero App to retrieve data from the local Aero App database.

2 Getting Started

The section describes how to implement the Data API in an Android application, prepare Aero App data, grant application access, and make an API request.

2.1 Gradle Dependencies

Add the Data API dependency to your application's Gradle build file.

Kolin

```
dependencies {  
    implementation 'org.jetbrains.kotlinx:kotlinx-serialization-json:1.6.2'  
    implementation (files('libs/api-<version>.aar'))  
}
```

2.2 Authentication and Data Access

The Data API does not require authentication. It reads aeronautical data from the local Aero App database, when permission has been granted.

Before using the API, the required aeronautical data must be downloaded through Aero App using one of the supported authentication methods for the selected data source:

- **Amazon Web Service:** Supports Aero User Database, GEOAxis, or MDM authentication.
- **Aero Data Server:** Requires access to a local server.

After the required aeronautical data has been downloaded, the Android application must be granted permission to access Aero App data.

Use [checkAeroAppPermission](#) to verify whether the application has permission to access Aero App data:

Kotlin

```
var hasAeroAppPermission = AeroAppDataAPI(this).checkAeroAppPermission(this)
return hasAeroAppPermission
```

If [checkAeroAppPermission](#) returns `false`, use [requestAeroAppPermission](#) to request access.

Once the required data has been downloaded through Aero App and permission has been granted, no additional authentication is required to use the API.

2.3 Make your first API call

After adding the Aero App API library to your Android application and obtaining permission to access Aero App data, you can begin retrieving data from Aero App.

The following examples demonstrate how to retrieve the active cycle with [getActiveCycle\(\)](#) and search for airport data with [getAirports\(\)](#). Each method returns an [AeroAppResult](#) containing the requested data or an error if the data cannot be retrieved.

NOTE: The JSON examples in this section illustrate the structure of the returned data. API methods return Kotlin objects rather than JSON responses.

2.3.1 Retrieve active cycle information

Use [getActiveCycle\(\)](#) to retrieve the active data cycle from Aero App, including the cycle name, start data, and end date.

Example usage

The following example calls [getActiveCycle\(\)](#), checks for an error, and accesses the returned cycle data.

Kotlin

```
val cycleResult = AeroAppDataAPI(this).getActiveCycle(this).let {
    cycleResult.error?.let { throw it }
    val cycles = cycleResult.data.orEmpty()
}
```

In this example, `cycles` is a `List<AeroAppCycle>` containing the returned active cycle data.

Example data

The following JSON illustrates the structure of an [AeroAppCycle](#) object returned in the data list.

```
{
  "name": "2606",
  "start": "2026-06-11",
  "end": "2026-07-09"
}
```

2.3.2 Retrieve airport information

Use [getAirports\(\)](#) to retrieve airport data that matches the specified search terms and filters.

Example usage

The following example calls [getAirports\(\)](#) to retrieve DAFIF airport data for KBLV with the minimum runway length of 4,000', checks for an error, and accesses the returned airport data.

Kotlin

```
val airportResult = getAirports(
    searchTerms = arrayOf("KBLV"),
    partialMatches = false,
    inDAFIF = true,
    minRunwayLength = 4000
)

airportResult.error?.let { throw it }

val airports = airportResult.data.orEmpty()
```

In this example, `airports` is a `List<AirportData>` containing the airports that match the specified search criteria.

Example data

The following JSON illustrates the structure of an [AirportData](#) object returned in the data list.

```
{
  "wacInnr": "0359-00182",
  "info": {
    "icao": "KBLV",
    "name": "Scott Afb Midamerica St Louis",
    "location": "Belleville, Illinois",
    "country": "United States",
    "countryCode": "US",
    "region": "CONUS",
    "elevation": 459.0,
    "latitude": 38.54517778,
    "longitude": -89.83521111,
    "magneticVariance": -2.2511122300000004,
    "hasRotatingBeacon": true,
    "hasArrestingGear": false,
    "hasAQP": true,
    "inDAFIF": true,
    "communications": [
      {
        "callSign": "126 ARW COMD POST",
        "commType": "ACP",
        "frequencyList": [
          "138.55",
          "277.7"
        ]
      }
    ]
  }
}
```

```
    ]  
  }  
  ...  
],  
"runways": [  
  {  
    "identifier": "14L/32R",  
    "length": 10000,  
    "width": 150,  
    "surfaceCode": "CON",  
    "conditionCode": "G",  
    "runwayEndHi": {  
      "latitude": 38.53626111,  
      "longitude": -89.80945556,  
      "identifier": "32R",  
      "magHdg": 318.0,  
      "truHdg": 316.6,  
      "tdze": 441.8  
    }  
  }  
  ...  
],  
"remarks": [  
  {  
    "type": "CAUTION",  
    "rmk": "Dense civ air tfc all quad, all alt..."  
  }  
]
```

```
    ...  
  
  ],  
  
  "aqp": {  
  
    "description": "LOCATED 19 MILES EAST OF ST LOUIS, MO.",  
  
    "overview": "<ImageBitmap>",  
  
    "falconView": "<ImageBitmap>"  
  
  }  
}
```

3 Functions

<u>checkAeroAppPermission</u>	Checks whether Aero App data access permission is granted.
<u>requestAeroAppPermission</u>	Launches the Aero App data access permission request.
<u>getActiveCycle</u>	Retrieves the active data cycle.
<u>getAeroAppVectorMapInfo</u>	Retrieves vector map layer metadata.
<u>getAeroAppVectorTile</u>	Retrieves a vector tile for the specified map layer.
<u>getAirportCharts</u>	Retrieves an airport chart.
<u>getAirports</u>	Retrieves airport information.
<u>getAreaPlanning</u>	Retrieves Area Planning file URIs.
<u>getAreaPlanningList</u>	Retrieves the list of available Area Planning documents.
<u>getARR</u>	Retrieves Aerial Refueling Route (ARR) information.
<u>getArtcc</u>	Retrieves Air Route Traffic Control Center (ARTCC) information.
<u>getChartSupplement</u>	Retrieves Chart Supplement file URIs.
<u>getChartSupplementsList</u>	Retrieves the list of available Chart Supplement documents.
<u>getGiantReport</u>	Retrieves Giant Report file URIs for specified airports.
<u>getMtr</u>	Retrieves Military Training Route (MTR) information.
<u>getNavAids</u>	Retrieves navigation aid (NavAid) information.
<u>getParachuteJumpingArea</u>	Retrieves Parachute Jumping Area (PJA) information.
<u>getSUA</u>	Retrieves Special Use Airspace (SUA) information.
<u>getVO</u>	Retrieves Vertical Obstruction (VO) information.
<u>getWaypoints</u>	Retrieves Waypoint information.

3.1 checkAeroAppPermission

Checks whether the calling Android application has permission to access Aero App data through the Aero App ContentProvider. Returns `true` if permission is granted; otherwise, returns `false`.

Signature

```
fun checkAeroAppPermission(context: Context): Boolean
```

Added in 1.2306 Aero App for Android

Parameters

Parameter	Type	Required/Optional	Description
<code>context</code>	Context	Required	The application context.

3.2 requestAeroAppPermission

Launches the permission request to access Aero App data through the Aero App ContentProvider.

Signature

```
fun requestAeroAppPermission(activity: Activity)
```

Added in 1.2306 Aero App for Android

Parameters

Parameter	Type	Required/Optional	Description
<code>activity</code>	Activity	Required	The activity that launches the permission request.

3.3 getActiveCycle

Retrieves the active data cycle in Aero App, including the cycle name, effective date, and expiration date.

Signature
<pre>fun getActiveCycle(context: Context): AeroAppResult<AeroAppCycle></pre>

Added in 1.2402 Aero App for Android

Parameters

Parameter	Type	Required/Optional	Description
<code>context</code>	Context	Required	The application context.

Returns

An [AeroAppResult](#) containing the active [AeroAppCycle](#) object, or an error if the cycle cannot be retrieved.

3.4 getAeroAppVectorMapInfo

Retrieves metadata for the specified [AeroAppVectorTileLayer](#), including the minimum and maximum zoom levels and, when available, the geographic bounds of the vector map layer.

Signature

```
fun getAeroAppVectorMapInfo(  
    context: Context = this.context,  
    layer: AeroAppVectorTileLayer  
): AeroAppResult<AeroAppMapInfo>
```

Added in 1.2606 Aero App for Android

Parameters

Parameter	Type	Required/Optional	Description
<code>context</code>	Context	Optional	The application context.
<code>layer</code>	AeroAppVectorTileLayer	Required	The vector map layer for which tile is requested.

Returns

An [AeroAppResult](#) containing the requested [AeroAppMapInfo](#) object for the specified vector map layer, or an error if the metadata cannot be retrieved.

3.5 getAeroAppVectorTile

Retrieves the vector tile for the specified [AeroAppVectorTileLayer](#) at the requested zoom level and tile coordinates.

Signature

```
fun getAeroAppVectorTile(  
    context: Context = this.context,  
    layer: AeroAppVectorTileLayer,  
    zoom: Int,  
    x: Int,  
    y: Int,  
    flipY: Boolean,  
) : AeroAppResult<ByteArray>
```

Added in 1.2606 Aero App for Android

Parameters

Parameter	Type	Required/Optional	Description
Context	Context	Optional	The application context.
layer	AeroAppVectorTileLayer	Required	The vector map layer containing the requested tile.
zoom	Int	Required	The zoom level for the requested vector tile.
x	Int	Required	The X coordinate of the requested vector tile.
y	Int	Required	The Y coordinate of the requested vector tile.
flipY	Boolean	Required	If true , flips the Y coordinate for XYZ tile coordinate

compatibility. If <code>false</code> , uses the Y coordinate as provided.

Returns

An [AeroAppResult](#) containing the requested vector tile as `ByteData`, which holds the vector tile's byte array, or an error if the vector tile cannot be retrieved.

3.6 getAirportCharts

Retrieves the airport chart for the specified airport based on the selected [AirportChartType](#) and [ColorMode](#).

Signature

```
fun getAirportCharts(  
    wacInnr: String,  
    context: Context,  
    chartType: AirportChartType,  
    colorMode: ColorMode = ColorMode.Light  
) : AeroAppResult<AirportChartData>
```

Modified in 1.2402 Aero App for Android

Parameters

Parameter	Type	Required/Optional	Description
wacInnr	String	Required	The wacInnr identifier of the airport for which to retrieve the chart.
context	Context	Required	The application context.
chartType	AirportChartType	Required	The type of airport chart to retrieve.
colorMode	ColorMode	Optional	The color mode of the returned. Defaults to ColorMode.Light .

Returns

An [AeroAppResult](#) containing the requested [AirportChartData](#), including the chart name and image, or an error if the airport chart cannot be retrieved.

3.7 getAirports

Retrieves the airport information that matches the specified search terms and filters. The airport information includes the airport general information and, when available, the airport AQP data.

Signature

```
fun getAirports(  
    searchTerms: Array<String>,  
    partialMatches: Boolean,  
    inDAFIF: Boolean = true,  
    minRunwayLength: Int = 0,  
    info: Boolean = true,  
    aqpDescripton: Boolean = true,  
    aqpOverview: Boolean = true,  
    aqpFalconView: Boolean = true  
) : AeroAppResult<AirportData>
```

Updated in 1.2606 Aero App for Android

Parameters

Parameter	Type	Required/Optional	Description
<code>searchTerms</code>	<code>Array<String></code>	Required	One or more search terms used to find matching airports.
<code>partialMatches</code>	<code>Boolean</code>	Required	If <code>true</code> , enables partial matching and returns airports whose identifiers contain the provided search term. If <code>false</code> , only information for the airport whose identifier exactly matches the search term will be returned.

<code>inDAFIF</code>	Boolean	Optional	If <code>true</code> , limits results to airports included in the DAFIF database. Defaults to <code>true</code> .
<code>minRunwayLength</code>	Int	Optional	The minimum runway length required for an airport to be included in the results. Defaults to <code>0</code> .
<code>info</code>	Boolean	Optional	If <code>true</code> , includes AeroAppAirport data for each matching airport. Defaults to <code>true</code> .
<code>aqpDescripton</code>	Boolean	Optional	If <code>true</code> , includes AQP description for each matching airport when AQP data is available. Defaults to <code>true</code> .
<code>aqpOverview</code>	Boolean	Optional	If <code>true</code> , includes AQP Overview image for each matching airport when available. Defaults to <code>true</code> .
<code>aqpFalconView</code>	Boolean	Optional	If <code>true</code> , includes AQP FalconView image for each matching airport when available. Defaults to <code>true</code> .

Returns

An [AeroAppResult](#) containing the requested [AirportData](#) objects, or an error if the airport data cannot be retrieved.

3.8 getAreaPlanning

Retrieves the file URIs for the specified Area Planning documents.

Signature

```
fun getAreaPlanning(  
    names: List<String>,  
    context: Context  
) : AeroAppResult<Uri>
```

Added in 1.2402 Aero App for Android

Parameters

Parameter	Type	Required/Optional	Description
<code>names</code>	List<String>	Required	The Area Planning document names used to retrieve the corresponding file URIs.
<code>context</code>	Context	Required	The application context.

Returns

An [AeroAppResult](#) containing the URI for the requested Area Planning documents, or an error if the file URIs cannot be retrieved.

3.9 getAreaPlanningList

Retrieves the list of available Area Planning PDF documents, including each document's name and, when available, its description.

Signature

```
fun getAreaPlanningList(  
    context: Context  
) : AeroAppResult<AeroAppPDFDocument>
```

Added in 1.2402 Aero App for Android

Parameters

Parameter	Type	Required/Optional	Description
<code>context</code>	Context	Required	The application context.

Returns

An [AeroAppResult](#) containing the available [AeroAppPDFDocument](#) objects, or an error if the document list cannot be retrieved.

3.10 getARR

Retrieves the Aerial Refueling Route (ARR) information that match the specified search terms and filters. The ARR information includes the ARR identifier, direction, refueling track type, frequencies, TACAN channels, altitudes, scheduling unit, and the assigned ATC.

Signature

```
fun getARR(  
    context: Context,  
    searchTerms: Array<String>,  
    partialMatches: Boolean,  
    inDAFIF: Boolean = false,  
) : AeroAppResult<AeroAppARR>
```

Added in 1.2606 Aero App for Android

Parameters

Parameter	Type	Required/Optional	Description
<code>context</code>	Context	Required	The application activity.
<code>searchTerms</code>	Array<String>	Required	One or more search terms used to find matching ARR.
<code>partialMatches</code>	Boolean	Required	If <code>true</code> , enables partial matching and returns ARRs whose identifiers contain the provided search term. If <code>false</code> , only information for the ARR whose identifier exactly matches the search term will be returned.
<code>inDAFIF</code>	Boolean = false	Optional	If <code>true</code> , limits results to airports included in the DAFIF database. Defaults to <code>false</code> .

Returns

An [AeroAppResult](#) containing the requested [AeroAppARR](#) objects, or an error if the Aerial Refueling Route data cannot be retrieved.

3.11 getArtcc

Retrieves the Air Route Traffic Control Center (ARTCC) data for the specified identifier including center identification, controlling agency, communication, airport classification, and altitude limits.

Signature

```
fun getArtcc(  
    context: Context,  
    id: String  
): AeroAppResult<AeroAppARTCC>
```

Added in 1.2606 Aero App for Android

Parameters

Parameter	Type	Required/Optional	Description
<code>context</code>	Context	Required	The application context.
<code>id</code>	String	Required	The unique identifier of the ARTCC.

Returns

An [AeroAppResult](#) containing the requested [AeroAppARTCC](#) objects, or an error if the ARTCC data cannot be retrieved.

3.12 getChartSupplement

Retrieves the file URIs for the specified Chart Supplement documents.

Signature
<pre>fun getChartSupplement(names: List<String>, context: Context) : AeroAppResult<Uri></pre>

Added in 1.2402 Aero App for Android

Parameters

Parameter	Type	Required/Optional	Description
<code>names</code>	List<String>	Required	The Chart Supplement document names used to retrieve the corresponding file URIs.
<code>context</code>	Context	Required	The application context.

Returns

The [AeroAppResult](#) containing the URI for the requested Chart Supplement documents, or an error if the file URIs cannot be retrieved.

3.13 getChartSupplementsList

Retrieves the list of available Chart Supplement, including each document's name and, when available, its description.

Signature

```
fun getChartSupplementsList(  
    context: Context  
) : AeroAppResult<AeroAppPDFDocument>
```

Modified in 1.2606 Aero App for Android

Parameters

Parameter	Type	Required/Optional	Description
<code>context</code>	Context	Required	The application context.

Returns

An [AeroAppResult](#) containing the available [AeroAppPDFDocument](#) objects, or an error if the document list cannot be retrieved.

3.14 getGiantReport

Retrieves the file URIs for Giant Reports associated with the specified ICAO identifiers.

Signature

```
fun getGiantReport(  
    icaos: List<String>,  
    context: Context  
) : AeroAppResult<Uri>
```

Added in 1.2606 Aero App for Android

Parameters

Parameter	Type	Required/Optional	Description
<code>icaos</code>	List<String>	Required	The list of ICAOs.
<code>context</code>	Context	Required	The application context.

Returns

An [AeroAppResult](#) containing the URI for the requested Giant Reports, or an error if the file URIs cannot be retrieved.

3.15 getMtr

Retrieves the Military Training Route (MTR) information for the specified identifier. The MTR information includes the route type, route identifiers, coordinates, altitude information, geographic bounds, route length, segment information, activity information, effective time, and remarks.

Signature

```
fun getMtr(  
    context: Context,  
    id: String  
): AeroAppResult<AeroAppMTR>
```

Added in 1.2606 Aero App for Android

Parameters

Parameter	Type	Required/Optional	Description
<code>context</code>	Context	Required	The application context.
<code>id</code>	String	Required	The unique identifier of the MTR.

Returns

An [AeroAppResult](#) containing the requested [AeroAppMTR](#) objects, or an error if the MTR data cannot be retrieved.

3.16 getNavAids

Retrieves the navigational aid (NavAid) information for the specified search terms and filters. The NavAid information includes identification, location, elevation, coordinates, magnetic variation, frequency, and TACAN channel information.

associated with the specified navigational aid (NavAid).

Signature

```
fun getNavAids(  
    searchTerms: Array<String>,  
    partialMatches: Boolean,  
    inDAFIF: Boolean = true,  
) : AeroAppResult<AeroAppNavAid>
```

Modified in 1.2606 Aero App for Android

Parameters

Parameter	Type	Required/Optional	Description
<code>searchTerms</code>	<code>Array<String></code>	Required	One or more search terms used to find matching NavAids.
<code>partialMatches</code>	<code>Boolean</code>	Required	If <code>true</code> , enables partial matching and returns NavAids whose identifiers contain the provided search term. If <code>false</code> , only information for the NavAid whose identifier exactly matches the search term will be returned.
<code>inDAFIF</code>	<code>Boolean</code>	Optional	If <code>true</code> , limits results to NavAids included in the DAFIF database. Defaults to <code>true</code> .

Returns

An [AeroAppResult](#) containing the requested [AeroAppNavAid](#) objects, or an error if the NavAid data cannot be retrieved.

3.17 getParachuteJumpingArea

Retrieves the Parachute Jumping Area (PJA) information for the specified identifier and country. The PJA information includes identification, region, name, type, province, horizontal datum, operating times, hours, altitude, jump status, remarks, and notes

Signature

```
fun getParachuteJumpingArea(  
    context: Context,  
    id: String,  
    country: String  
): AeroAppResult<AeroAppPJA>
```

Added in 1.2606 Aero App for Android

Parameters

Parameter	Type	Required/Optional	Description
<code>context</code>	Context	Required	The application context.
<code>id</code>	String	Required	The unique identifier of the PJA.
<code>country</code>	String	Required	The GENC2 country code associated with the PJA.

Returns

An [AeroAppResult](#) containing the requested [AeroAppPJA](#) objects, or an error if the PJA data cannot be retrieved.

3.18 getSUA

Retrieves the Special Use Airspace (SUA) information for the specified identifier and filter. The SUA information includes name, type, sector, controlling agency, frequencies, communication name, altitude limits, effective times, and remarks.

Signature

```
fun getSUA(  
    id: String,  
    partialMatches: Boolean = false  
): AeroAppResult<AeroAppSUA>
```

Added in 1.2606 Aero App for Android

Parameters

Parameter	Type	Required/Optional	Description
<code>id</code>	String	Required	The unique identifier of the SUA.
<code>partialMatches</code>	Boolean	Optional	If <code>true</code> , enables partial matching and returns SUAs whose identifiers contain the provided search term. If <code>false</code> , only information for the SUA whose identifier exactly matches the search term will be returned. Defaults to <code>false</code> .

Returns

An [AeroAppResult](#) containing the requested [AeroAppSUA](#) objects, or an error if the SUA data cannot be retrieved.

3.19 getVO

Retrieves the Vertical Obstruction (VO) information for the specified UUID. The VO information includes identification, country code, obstruction type, coordinates, height, elevation, obstruction count, and accuracy information.

Signature

```
fun getVO(  
    uuid: String  
): AeroAppResult<AeroAppVO>
```

Added in 1.2606 Aero App for Android

Parameters

Parameter	Type	Required/Optional	Description
<code>uuid</code>	String	Required	The unique identifier of the VO.

Returns

An [AeroAppResult](#) containing the requested [AeroAppVO](#) objects, or an error if the VO data cannot be retrieved.

3.20 getWaypoints

Retrieves the Waypoint information for the specified search terms and filters. The waypoint information includes identification, name, location, coordinates, and magnetic variation.

Signature

```
fun getWaypoints(  
    searchTerms: Array<String>,  
    partialMatches: Boolean,  
    inDAFIF: Boolean = true,  
) : AeroAppResult<AeroAppWaypoint>
```

Modified in 1.2606 Aero App for Android

Parameters

Parameter	Type	Required/Optional	Description
<code>searchTerms</code>	<code>Array<String></code>	Required	One or more search terms used to find matching Waypoints.
<code>partialMatches</code>	<code>Boolean</code>	Required	If <code>true</code> , enables partial matching and returns Waypoints whose identifiers contain the provided search term. If <code>false</code> , only information for the Waypoint whose identifier exactly matches the search term will be returned.
<code>inDAFIF</code>	<code>Boolean</code>	Optional	If <code>true</code> , limits results to Waypoints included in the DAFIF database. Defaults to <code>true</code> .

Returns

An [AeroAppResult](#) containing the requested [AeroAppWaypoint](#) objects, or an error if the Waypoint data cannot be retrieved.

4 Classes

<u>AeroAppAirport</u>	Represents an airport and its associated information.
<u>AeroAppAirportCommunication</u>	Represents airport communication information.
<u>AeroAppAirportRemark</u>	Represents an airport remark.
<u>AeroAppARR</u>	Represents an Aerial Refueling Route (ARR) and its associated information.
<u>AeroAppARTCC</u>	Represents an Air Route Traffic Control Center (ARTCC) and its associated information.
<u>AeroAppBoundingBox</u>	Represents the geographic bounds of an area.
<u>AeroAppMTR</u>	Represents a Military Training Route (MTR) and its associated information.
<u>AeroAppMTRRemark</u>	Represents the Military Training Route's remark.
<u>AeroAppNavaid</u>	Represents a navigational aid (NavAid) and its associated information.
<u>AeroAppPJA</u>	Represents a Parachute Jumping Area (PJA) and its associated information.
<u>AeroAppResult</u>	Represents the result of an Aero App data retrieval operation.
<u>AeroAppRunway</u>	Represents an airport runway.
<u>AeroAppRunwayEnd</u>	Represents an airport runway end.
<u>AeroAppSUA</u>	Represents a Special Use Airspace (SUA) and its associated information.
<u>AeroAppVO</u>	Represents a Vertical Obstruction (VO) and its associated information.
<u>AeroAppWaypoint</u>	Represents a Waypoint and its associated information.
<u>AeroAppAirportAQP</u>	Represents an airport's AQP information.

AeroAppCycle	Represents the data cycle information.
AeroAppMapInfo	Represents the vector map metadata.
AeroAppPDFDocument	Represents a PDF document information.
AirportChartData	Represents airport chart data information.
AirportData	Represents the airport data returned by the API.

4.1 AeroAppAirport

Represents information associated with the specified airport.

Class
<pre> @Parcelize @kotlinx.serialization.Serializable data class AeroAppAirport(val icao: String, val name: String, val location: String?, var country: String, var countryCode: String, val region: String?, val elevation: Float, val latitude: Double, val longitude: Double, val magneticVariance: Double, val hasRotatingBeacon: Boolean, val hasArrestingGear: Boolean, val hasAQP: Boolean, val inDAFIF: Boolean, val communications: List<AeroAppAirportCommunication>, val runways: List<AeroAppRunway>, val remarks: List<AeroAppAirportRemark>,) : Parcelable </pre>

Updated in 1.2606 Aero App for Android

Properties

Property	Type	Nullable	Description
<code>icao</code>	String	No	Specifies the ICAO code associated with the airport.
<code>name</code>	String	No	Specifies the name of the airport.
<code>location</code>	String	Yes	Specifies the city and state of the airport.
<code>country</code>	String	No	Specifies the country where the airport is located.
<code>countryCode</code>	String	No	Specifies the GENC2 country code associated with the airport.
<code>region</code>	String	Yes	Specifies the geographic region where the airport is located.
<code>elevation</code>	Float	No	Specifies the airport elevation in feet above mean sea level.
<code>latitude</code>	Double	No	Specifies the airport latitude in decimal degrees.
<code>longitude</code>	Double	No	Specifies the airport longitude in decimal degrees.
<code>magneticVariation</code>	Double	No	Specifies the magnetic variation at the airport in degrees.
<code>hasRotatingBeacon</code>	Boolean	No	Indicates whether the airport has a rotating beacon.
<code>hasArrestingGear</code>	Boolean	No	Indicates whether the airport has arresting gear.
<code>hasAQP</code>	Boolean	No	Indicates whether the airport has AQP.
<code>inDAFIF</code>	Boolean	No	Indicates whether the airport is included in DAFIF data.

communications	List< AeroAppAirportCommunication >	No	Contains the communication facilities and frequencies associated with the airport.
runways	List< AeroAppRunway >	No	Contains the runways associated with the airport.
remarks	List< AeroAppAirportRemark >	No	Contains additional remarks associated with the airport.

4.2 AeroAppAirportCommunication

Represents the communication facility information associated with an airport.

Class
<pre> @Parcelize @kotlinx.serialization.Serializable data class AeroAppAirportCommunication(val callSign: String, val commType: String, val multi: String?, val frequencyList: List<String>, val oprHrs: String?, val sector: String?, val remark: String?) : Parcelable </pre>

Updated in 1.2606 Aero App for Android

Properties

Property	Type	Nullable	Description
<code>callSign</code>	String	No	Specifies the call sign associated with the communication facility.
<code>commType</code>	String	No	Specifies the type of communication facility.
<code>multi</code>	String	Yes	Specifies whether multiple communication facilities are associated with the airport.
<code>frequencyList</code>	List<String>	No	Contains the frequencies assigned to the communication facility.
<code>oprHrs</code>	String	Yes	Specifies the operating hours of the communication facility.

<code>sector</code>	String	Yes	Specifies the sector associated with the communication facility.
<code>remark</code>	String	Yes	Contains additional remarks about the communication facility.

4.3 AeroAppAirportRemark

Represents remark information associated with an airport.

Class
<pre>@Parcelize @kotlinx.serialization.Serializable data class AeroAppAirportRemark(val type: String, val rmk: String) : Parcelable</pre>

Updated in 1.2606 Aero App for Android

Properties

Property	Type	Nullable	Description
<code>type</code>	String	No	Specifies the type of category of the airport remark.
<code>rmk</code>	String	No	Contains the text of the airport remark.

4.4 AeroAppARR

Represents information associated with an Air Refueling Route (ARR).

Class
<pre> @Parcelize @kotlinx.serialization.Serializable data class AeroAppARR(var airRefIdent: String?, var direc: String?, var refuelTrkType: String?, var primUhfFreq: Double?, var backupUhfFreq: Double?, var receiverTacanChan: Int?, var tankerTacanChan: Int?, var ref1Alt1: String?, var ref1Alt2: String?, var schedUnit: String?, var assignAtc: String?) : Parcelable </pre>

Added in 1.2606 Aero App for Android

Properties

Property	Type	Nullable	Description
<code>airRefIdent</code>	String	Yes	Specifies the identifier of the ARR.
<code>direc</code>	String	Yes	Specifies the cardinal direction of the ARR.
<code>refuelTrkType</code>	String	Yes	Specifies the track type of the ARR.

<code>primUhfFreq</code>	Double	Yes	Specifies the primary ultra-high frequency used for the ARR.
<code>backupUhfFreq</code>	Double	Yes	Specifies the backup ultra-high frequency used for the ARR.
<code>receiverTacanChan</code>	Int	Yes	Specifies the TACAN channel assigned to the receiver.
<code>tankerTacanChan</code>	Int	Yes	Specifies the TACAN channel assigned to the tanker.
<code>ref1Alt1</code>	String	Yes	Specifies the first reference altitude associated with the ARR.
<code>ref1Alt2</code>	String	Yes	Specifies the second reference altitude associated with the ARR.
<code>schedUnit</code>	String	Yes	Specifies the scheduling unit associated with the ARR.
<code>assignAtc</code>	String	Yes	Specifies the ATC assignment associated with the ARR.

4.5 AeroAppARTCC

Represents information associated with an Air Route Traffic Control Center (ARTCC).

Class
<pre>@Parcelize @Serializable data class AeroAppARTCC(val ident: String, val name: String, val icao: String, val conAgency: String, val commName: String, val classAirspaceVariable: String, val level: String, val upper: String, val lower: String): Parcelable</pre>

Added in 1.2606 Aero App for Android

Properties

Property	Type	Nullable	Description
<code>ident</code>	String	No	Specifies identifier of the ARTCC.
<code>name</code>	String	No	Specifies name of the ARTCC.
<code>icao</code>	String	No	Specifies the ICAO code associated with the ARTCC.
<code>conAgency</code>	String	No	Specifies controlling agency responsible for the ARTCC or associated airspace.

<code>commName</code>	String	No	Specifies name associated with the ARTCC communication facility.
<code>classAirspaceVariable</code>	String	No	Specifies airspace classification or variable associated with the ARTCC.
<code>level</code>	String	No	Specifies airspace level associated with the ARTCC.
<code>upper</code>	String	No	Specifies upper limit of the airspace associated with the ARTCC.
<code>lower</code>	String	No	Specifies lower limit of the airspace associated with the ARTCC.

4.6 AeroAppBoundingBox

Represents the geographic bounding box of a map area.

Class
<pre>@Serializable @Parcelize data class AeroAppBoundingBox(val minLat: Double, val minLon: Double, val maxLat: Double, val maxLon: Double) : Parcelable</pre>

Added in 1.2606 Aero App for Android

Properties

Property	Type	Nullable	Description
<code>minLat</code>	Double	No	Specifies southernmost minimum latitude of the bounding box.
<code>minLon</code>	Double	No	Specifies westernmost minimum longitude of the bounding box.
<code>maxLat</code>	Double	No	Specifies northernmost maximum latitude of the bounding box.
<code>maxLon</code>	Double	No	Specifies easternmost maximum longitude of the bounding box.

4.7 AeroAppMTR

Represents the information associated with a Military Training Route (MTR).

Class
<pre>@Parcelize @Serializable data class AeroAppMTR(val ident: String, val mtrRouteType: String, val mtrIdent: String, val identifiers: List<String>, val crossAlt: List<String>, val coords: List<Double>, val minWgsLat: Double?, val minWgsLon: Double?, val maxWgsLat: Double?, val maxWgsLon: Double?, val length: Int?, val midpoints: List<Double>, val segmentLengths: List<Int>, val origAct: String?, val schedAct: String?, val effectTime: String?, val remarks: List<AeroAppMTRRemark>): Parcelable</pre>

Added in 1.2606 Aero App for Android

Properties

Property	Type	Nullable	Description
<code>ident</code>	String	No	Identifies the MTR using its route type and identifier number.
<code>mtrRouteType</code>	String	No	Specifies the MTR type based on the applicable flight rules.
<code>mtrIdent</code>	String	No	Identifies the MTR by its route number.
<code>identifiers</code>	List<String>	No	Identifies the individual MTR segments.
<code>crossAlt</code>	List<String>	No	Specifies the altitude at which the pilot should cross the MTR corridor.
<code>coords</code>	List<Double>	No	Contains the geographic coordinates of the MTR.
<code>minWgsLat</code>	Double	Yes	Specifies the minimum WGS 84 latitude of the MTR.
<code>minWgsLon</code>	Double	Yes	Specifies the minimum WGS 84 longitude of the MTR.
<code>maxWgsLat</code>	Double	Yes	Specifies the maximum WGS 84 latitude of the MTR.
<code>maxWgsLon</code>	Double	Yes	Specifies the maximum WGS 84 longitude of the MTR.
<code>length</code>	Int	Yes	Specifies the total MTR distance in statute miles.
<code>midpoints</code>	List<Double>	No	Contains the geographic centerline point of each MTR segment.

<code>segmentLengths</code>	List<Int>	No	Specifies the distance of each MTR segment in statute miles.
<code>origAct</code>	String	Yes	Contains contact information for the activity responsible for the MTR.
<code>schedAct</code>	String	Yes	Contains contact information for the activity responsible for scheduling the MTR.
<code>effectTime</code>	String	Yes	Specifies the effective operating hours for the MTR.
<code>remarks</code>	List< AeroAppMTRRemark >	No	Contains additional remarks about the MTR.

4.8 AeroAppMTRRemark

Represents remark information associated with a Military Training Route (MTR).

Class
<pre>@Parcelize @Serializable data class AeroAppMTRRemark(val rmkSeq: Int, val rmk: String): Parcelable</pre>

Properties

Property	Type	Nullable	Description
<code>rmkSeq</code>	Int	No	Specifies the sequence number of the remark.
<code>rmk</code>	String	No	Contains the remark text.

4.9 AeroAppNavAid

Represents information associated with an aviation navigation aid (NavAid).

Class
<pre> @Parcelize @kotlinx.serialization.Serializable data class AeroAppNavAid(val identifier: String, val name: String, val elev: Int? = null, val location: String? = null, val state: String? = null, val country: String? = null, val latitude: Double, val longitude: Double, val magVar: Double, val freq: String? = null, val tacanChan: String? = null) : Parcelable </pre>

Updated in 1.2606 Aero App for Android

Properties

Property	Type	Nullable	Description
<code>identifier</code>	String	No	Identifies the NavAid.
<code>name</code>	String	No	Specifies the name of the NavAid.
<code>elev</code>	Int	Yes	Specifies the elevation of the NavAid in feet above mean sea level.

location	String	Yes	Specifies the state and country of the NavAid.
state	String	Yes	Specifies the state where the NavAid is located.
country	String	Yes	Specifies the country where the NavAid is located.
latitude	Double	No	Specifies the latitude of the NavAid in decimal degrees.
longitude	Double	No	Specifies the longitude of the NavAid in decimal degrees.
magVar	Double	No	Specifies the magnetic variation at the NavAid in degrees. Positive values indicate east variation; negative values indicate west variation.
freq	String	Yes	Specifies the NavAid frequency.
tacanChan	String	Yes	Specifies the TACAN channel assigned to the NavAid.

4.10 AeroAppPJA

Represents information associated with Parachute Jumping Area (PJA).

Class
<pre>@Parcelize @Serializable data class AeroAppPJA(val country: String, val pjaIdent: String, val icaoRgn: String, val pjaName: String, val pjaTypeCode: Int, val pjaType: String, val province: String, val lclHorizDatum: String, val wgsDatum: String, val oprTimesCode: String, val oprTimes: List<String>, val hours: String, val alt: String, val paraJumpInwork: String, val paraJumpClosed: String, val oprTimesRmk: String, val note: String): Parcelable</pre>

Added in 1.2606 Aero App for Android

Properties

Property	Type	Nullable	Description
<code>country</code>	String	No	Specifies the country where the PJA is located.
<code>pjaIdent</code>	String	No	Identifies the PJA.
<code>icaoRgn</code>	String	No	Specifies the ICAO region code associated with the PJA.
<code>pjaName</code>	String	No	Specifies the name of the PJA.
<code>pjaTypeCode</code>	Int	No	Specifies the numeric code for the PJA type.
<code>pjaType</code>	String	No	Specifies the type of PJA.
<code>province</code>	String	No	Specifies the province code where the PJA is located.
<code>lclHorizDatum</code>	String	No	Specifies the local horizontal datum used for the PJA.
<code>wgsDatum</code>	String	No	Specifies the WGS datum used for the PJA.
<code>oprTimesCode</code>	String	No	Specifies the numeric code for the PJA operating times.
<code>oprTimes</code>	List<String>	No	Contains the operating times for the PJA.
<code>hours</code>	String	No	Specifies the operating hours for the PJA.
<code>alt</code>	String	No	Specifies the altitude associated with the PJA.
<code>paraJumpInwork</code>	String	No	Indicates whether parachute jumping operations are in progress.
<code>paraJumpClosed</code>	String	No	Indicates whether parachute jumping operations are closed.

<code>oprTimesRmk</code>	String	No	Contains additional remarks about the PJA operating times.
<code>note</code>	String	No	Contains additional information about the PJA.

4.11 AeroAppResult

Represents the result of an API operation, including any error and returned data.

A successful operation returns the requested data in `data`. If the operation fails, `error` contains the error that occurred. The data property can be `null` when the operation does not return data.

Class
<pre>class AeroAppResult<T : Parcelable>(val error: Throwable?, val data: List<T>?)</pre>

Modified in 1.2402 Aero App for Android

Properties

Property	Type	Nullable	Description
<code>error</code>	Throwable	Yes	Contains the error that occurred during the API operation.
<code>data</code>	List<T>	Yes	Contains the data returned by the API operation.

Note:

- `T` specifies the type of object returned in the `data` list.
- Nullable properties may be `null` when the corresponding information is unavailable.

4.12 AeroAppRunway

Represents information associated with an airport runway.

Class
<pre>@Parcelize @kotlinx.serialization.Serializable data class AeroAppRunway(val identifier: String, val length: Int, val width: Int, val surfaceCode: String, val conditionCode: String, val runwayEndHi: AeroAppRunwayEnd, val runwayEndLo: AeroAppRunwayEnd, val oprStatus: String?, val pcn: String?, val lcn: String?, val maxDemoSpan: String?) : Parcelable</pre>

Added in 1.2306 Aero App for Android

Properties

Property	Type	Nullable	Description
<code>identifier</code>	String	No	Identifies the runway.
<code>length</code>	Int	No	Specifies the runway length in feet.
<code>width</code>	Int	No	Specifies the runway width in feet.

surfaceCode	String	No	Specifies the code identifying the runway surface type.
conditionCode	String	No	Specifies the code identifying the runway surface condition.
runwayEndHi	AeroAppRunwayEnd	No	Contains information for the higher-numbered runway end.
runwayEndLo	AeroAppRunwayEnd	No	Contains information for the lower-numbered runway end.
oprStatus	String	Yes	Specifies the operational status of the runway.
pcn	String	Yes	Specifies the pavement classification number of the runway.
lcn	String	Yes	Specifies the load classification number of the runway.
maxDemoSpan	String	Yes	Specifies the maximum demonstrated aircraft wingspan category for the runway.

4.13 AeroAppRunwayEnd

Represents information associated with a runway end.

Class
<pre>@Parcelize @kotlinx.serialization.Serializable data class AeroAppRunwayEnd(val latitude: Double, val longitude: Double, val identifier: String, var magHdg: Float?, var truHdg: Float?, var tdze: Float?, var overrunLength: Float?) : Parcelable</pre>

Added in 1.2306 Aero App for Android

Properties

Property	Type	Nullable	Description
<code>latitude</code>	Double	No	Specifies the latitude of the runway end in decimal degrees.
<code>longitude</code>	Double	No	Specifies the longitude of the runway end in decimal degrees.
<code>identifier</code>	String	No	Identifies the runway end.
<code>magHdg</code>	Float	Yes	Specifies the magnetic heading of the runway end.
<code>truHdg</code>	Float	Yes	Specifies the true heading of the runway end.

<code>tdze</code>	Float	Yes	Specifies the touchdown zone elevation in feet MSL.
<code>overrunLength</code>	Float	Yes	Specifies the length of the runway overrun area in feet.

4.14 AeroAppSUA

Represents operation information associated with a Special Use Airspace (SUA) area.

Class

```
@Parcelize
@kotlinx.serialization.Serializable
class AeroAppSUA(
    val name: String? = null,
    val type: String? = null,
    val sector: String? = null,
    val conAgency: String? = null,
    val frequency1: String? = null,
    val frequency2: String? = null,
    val commName: String? = null,
    val level: String? = null,
    val minAltitude: String? = null,
    val maxAltitude: String? = null,
    val effTimes: String? = null,
    val remarks: String? = null
) : Parcelable
```

Added in 1.2606 Aero App for Android

Properties

Property	Type	Nullable	Description
<code>name</code>	String	Yes	Specifies the name of the SUA area.
<code>type</code>	String	Yes	Specifies the type of the SUA.

sector	String	Yes	Specifies the sector associated with the SUA.
conAgency	String	Yes	Specifies the controlling agency for the SUA.
frequency1	String	Yes	Specifies the primary communication frequency for the SUA.
frequency2	String	Yes	Specifies the secondary communication frequency for the SUA.
commName	String	Yes	Specifies the name of the communication facility associated with the SUA.
level	String	Yes	Specifies the airspace level associated with the SUA.
minAltitude	String	Yes	Specifies the minimum altitude of the SUA.
maxAltitude	String	Yes	Specifies the maximum altitude of the SUA.
effTimes	String	Yes	Specifies when the SUA is effective.
remarks	String	Yes	Contains additional information or notes about the SUA.

4.15 AeroAppVO

Represents information associated with a Vertical Obstruction (VO).

Class
<pre>@Parcelize @kotlinx.serialization.Serializable class AeroAppVO(val identifier: String, val countryCode: String? = null, val type: Int? = null, val typeName: String?, val latitude: Double, val longitude: Double, val agl: Int? = null, val msl: Int? = null, val count: Int? = null, val aglAccuracy: Int? = null, val mslAccuracy: Int? = null, val horizontalAccuracy: Int? = null,) : Parcelable</pre>

Added in 1.2606 Aero App for Android

Properties

Property	Type	Nullable	Description
<code>identifier</code>	String	No	Identifies the VO.
<code>countryCode</code>	String	Yes	Specifies the GENC2 country code where the VO is located.

type	Int	Yes	Specifies the numeric code for the VO type.
typeName	String	Yes	Specifies the type of VO.
latitude	Double	No	Specifies the latitude of the VO in decimal degrees.
longitude	Double	No	Specifies the longitude of the VO in decimal degrees.
agl	Int	Yes	Specifies the height of the VO above ground level.
msl	Int	Yes	Specifies the elevation of the VO above mean sea level.
count	Int	Yes	Specifies the number of VOs.
aglAccuracy	Int	Yes	Specifies the accuracy of the above ground level measurement of the VO.
mslAccuracy	Int	Yes	Specifies the mean sea level measurement of the VO.
horizontalAccuracy	Int	Yes	Specifies the horizontal positional accuracy of the VO.

4.16 AeroAppWaypoint

Represents information associated with a navigation Waypoint.

Class
<pre>@Parcelize @kotlinx.serialization.Serializable data class AeroAppWaypoint(val identifier: String, val name: String, val location: String? = null, val state: String? = null, val country: String? = null, val latitude: Double, val longitude: Double, val magVar: Double) : Parcelable</pre>

Updated in 1.2606 Aero App for Android

Properties

Property	Type	Nullable	Description
<code>identifier</code>	String	No	Identifies the waypoint.
<code>name</code>	String	No	Specifies the name of the waypoint.
<code>location</code>	String	Yes	Specifies the state and country of the waypoint.
<code>state</code>	String	Yes	Specifies the state where the waypoint is located.

country	String	Yes	Specifies the country where the waypoint is located.
latitude	Double	No	Specifies the latitude of the waypoint in decimal degrees.
longitude	Double	No	Specifies the longitude of the waypoint in decimal degrees.
magVar	Double	No	Specifies the magnetic variation at the waypoint in degrees. Positive values indicate east variation; negative values indicate west variation.

4.17 AeroAppAirportAQP

Represents the Advanced Qualification Program (AQP) information associated with an airport.

Class
<pre>@Parcelize data class AeroAppAirportAQP(val description: String, val overview: ImageBitmap?, val falconView: ImageBitmap?) : Parcelable</pre>

Added in 1.2402 Aero App for Android

Properties

Property	Type	Nullable	Description
<code>description</code>	String	No	Contains the description of the AQP.
<code>overview</code>	ImageBitmap	Yes	Contains an overview image of the AQP, if available.
<code>falconView</code>	ImageBitmap	Yes	Contains a FalconView image associated with the AQP, if available.

4.18 AeroAppCycle

Represents information associated with the available cycles.

Class
<pre>@Parcelize data class AeroAppCycle(val name: String, val start: LocalDate, val end: LocalDate) : Parcelable</pre>

Updated in 1.2606 Aero App for Android

Properties

Property	Type	Nullable	Description
<code>name</code>	String	No	Specifies the name of the data cycle.
<code>start</code>	LocalDate	No	Specifies the date on which the cycle becomes effective.
<code>end</code>	LocalDate	No	Specifies the date on which the cycle expires.

4.19 AeroAppMapInfo

Represents information associated with the vector map.

Class
<pre>@Serializable @Parcelize data class AeroAppMapInfo(val minZoom: Int, val maxZoom: Int, val bounds: AeroAppBoundingBox? = null) : Parcelable</pre>

Added in 1.2606 Aero App for Android

Properties

Property	Type	Nullable	Description
<code>minZoom</code>	Int	No	Specifies the minimum zoom level supported by the map.
<code>maxZoom</code>	Int	No	Specifies the maximum zoom level supported by the map.
<code>bounds</code>	AeroAppBoundingBox	Yes	Specifies geographic bounding box of a map area.

4.20 AeroAppPDFDocument

Represents information associated with a PDF document.

Class
<pre>@Parcelize data class AeroAppPDFDocument (val name: String, val description: String?) : Parcelable</pre>

Updated in 1.2606 Aero App for Android

Properties

Property	Type	Nullable	Description
<code>name</code>	String	No	Specifies the name of the PDF document.
<code>description</code>	String	Yes	Contains a description of the PDF document.

4.21 AirportChartData

Represents information associated with an airport chart.

Class
<pre>@Parcelize data class AirportChartData(val name: String, val bitmap: ImageBitmap) : Parcelable</pre>

Modified in 1.2402 Aero App for Android

Properties

Property	Type	Nullable	Description
<code>name</code>	String	No	Specifies the name of the airport chart.
<code>bitmap</code>	ImageBitmap	No	Contains the airport chart image.

4.22 AirportData

Represents airport information and the associated Advanced Qualification Program (AQP) data.

Class
<pre>@Parcelize data class AirportData(val wacInnr: String, val info: AeroAppAirport, val aqp: AeroAppAirportAQP?) : Parcelable</pre>

Modified in 1.2402 Aero App for Android

Properties

Property	Type	Nullable	Description
<code>wacInnr</code>	String	No	Identifies the chart data associated with the airport.
<code>info</code>	AeroAppAirport	No	Contains information associated with the airport.
<code>aqp</code>	AeroAppAirportAQP	Yes	Contains AQP information associated with the airport, if available.

5 Enums

AeroAppVectorTileLayer	Specifies the type of airport chart.
AirportChartType	Specifies the vector map layer.
ColorMode	Specifies the color mode for an airport chart.

5.1 AeroAppVectorTileLayer

Specifies the available map layers.

Enum
<pre>enum class AeroAppVectorTileLayer { Airports, ARR, ARTCC, InternationalBoundaries, MTR, NavAids, ParachuteJumpingAreas, SUA, VO, Waypoints }</pre>

Added in 1.2606 Aero App for Android

Enum Fields

Enum	Description
Airports	Represents the Airport map layer.
ARR	Represents the Aerial Refueling Route map layer.
ARTCC	Represents the Air Route Traffic Control Center map layer.
InternationalBoundaries	Represents the International Boundaries map layer.
MTR	Represents the Military Training Route map layer.
NavAids	Represents the Navigation Aid map layer.
ParachuteJumpingAreas	Represents the Parachute Jumping Area map layer.
SUA	Represents the Special Use Airspace map layer.
VO	Represents the Vertical Obstruction map layer.
Waypoints	Represents the Waypoint map layer.

5.2 AirportChartType

Specifies the type of chart available for an airport.

Enum
<pre>enum class AirportChartType : Parcelable { APD, IAP, ARR, DEP, MIN }</pre>

Modified in 1.2402 Aero App for Android

Enum Fields

Enum	Description
APD	Represents an Airport Diagram.
IAP	Represents an Instrument Approach Procedure.
ARR	Represents an Arrival chart.
DEP	Represents a Departure chart.
MIN	Represents a Minimums chart.

5.3 ColorMode

Specifies the color mode used by the application.

Enum
<pre>enum class ColorMode : Parcelable { Light, Dark }</pre>

Added in 1.2402 Aero App for Android

Enum Fields

Enum	Description
<code>Light</code>	Represents the light color mode.
<code>Dark</code>	Represents the dark color mode.