Verizon 5G Mobile Edge Compute (MEC)

Edge Discovery Service (EDS) Client Software Development Kit (SDK) - for iOS Devices

Table of Contents

Revision History

Version	Date	Description
1.0	Nov, 2022	Initial release

1. Introduction

Verizon 5G Edge Services is a multi-access edge computing (MEC) platform that enables cloud servers to run closer to optimal endpoints to reduce latency and accelerate local processing. With the platform, developers can extend their cloud environments to include Verizon edge zones and consume APIs to simplify device management and connectivity to optimal edge endpoints.

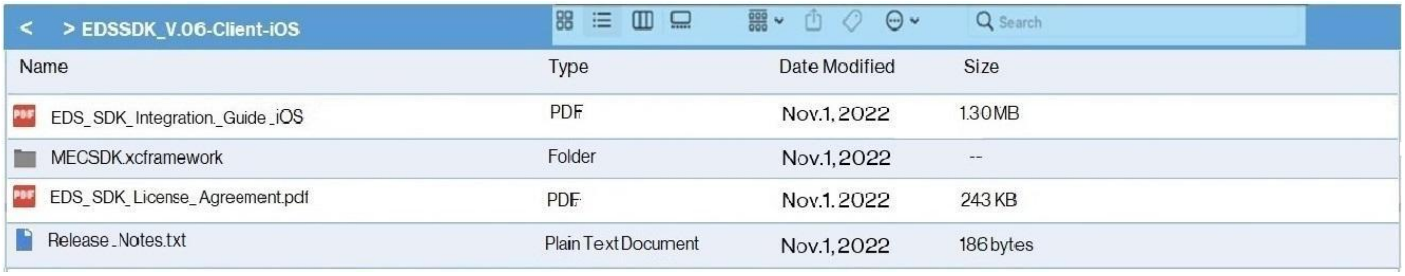
This document elaborates on an SDK that developers can use to simplify edge discovery from an iOS mobile client application.

1.1 Purpose

Verizon's Edge Discovery Service APIs enable developers to discover 5G edge platforms and register endpoints. This document explains how developers can use an SDK, instead of the REST interface, to discover optimal endpoints for iOS mobile client applications.

This interface specification guide describes the integration and use of Verizon's iOS Edge Discovery Service (EDS) Client SDK, including detailed application integration instructions and API specifications.

1.2 Package Contents



< > EDSSDK_V.06-Client-iOS

Name	Type	Date Modified	Size
EDS_SDK_Integration_Guide_iOS	PDF	Nov.1, 2022	1.30MB
MECSDK.xcframework	Folder	Nov.1, 2022	--
EDS_SDK_License_Agreement.pdf	PDF	Nov.1. 2022	243 KB
Release_Notes.txt	Plain Text Document	Nov.1, 2022	186 bytes

Figure 1 The EDS SDK Client Integration-iOS Package

The package includes:

- EDS_SDK_Integration_Guide_iOS.pdf
- MECSDK.xcframework
- License_Agreement.pdf
- Release_Notes.txt

1.3 System Requirements

1.3.1 Hardware

- iPhone 8 and later
- iPad 8th Generation and later

1.3.2 Software

- iOS Versions 13 and beyond

2. Prerequisites

5G Edge-hosted services must be configured in the Verizon 5G Edge MEC platform. (See Verizon 5G Edge Platform Documentation). The developer must register the endpoints.

3. SDK Usage

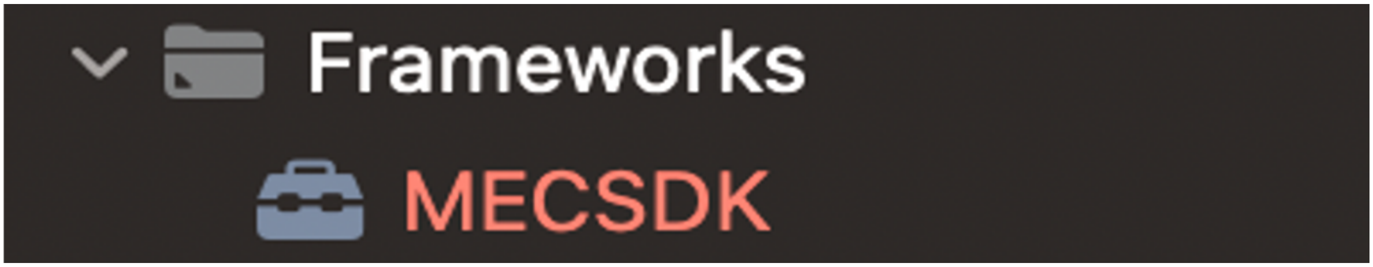
As the Find Edge Application Service Endpoints section explains, a developer can use a limited token and call the aforementioned REST API. The SDK can be used if the developer wants to avoid coding the REST interface.

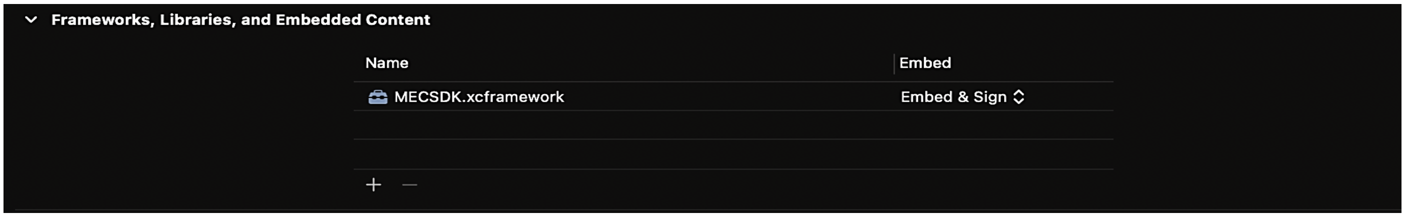
The SDK enables developers to use the same parameters the REST API supports—and bypass using limited access tokens —providing that they complete application onboarding first. This process is discussed in Onboarding an Application on the MEC Portal (chapter 5). This use case is discussed in greater detail in the SDK Usage Option 1 chapter.

Additionally, a developer can invoke the SDK using a limited access token and avoid onboarding the application in that use case. This use case is described in SDK Usage Option 2.

3.1 Importing the SDK into the Application

This section explains the use of **MECSDK.xcframework** in a Swift application. This is the SDK in the form of a framework. Import the framework inside the **Frameworks, Libraries and Embedded Contents** section of the package (ensure the **Embed & Sign** option is selected).

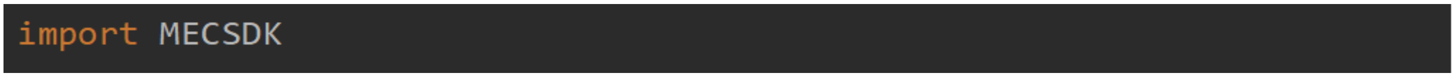




Name	Embed
MECSDK.xcframework	Embed & Sign

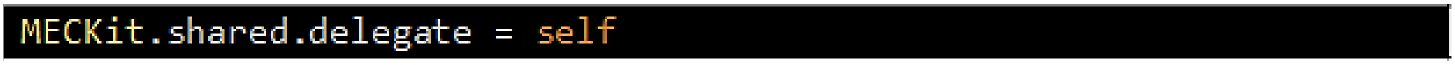
3.2 Adding the Code

The import command must be in the view controller where the developer intends to use the SDK interface:



```
import MECSDK
```

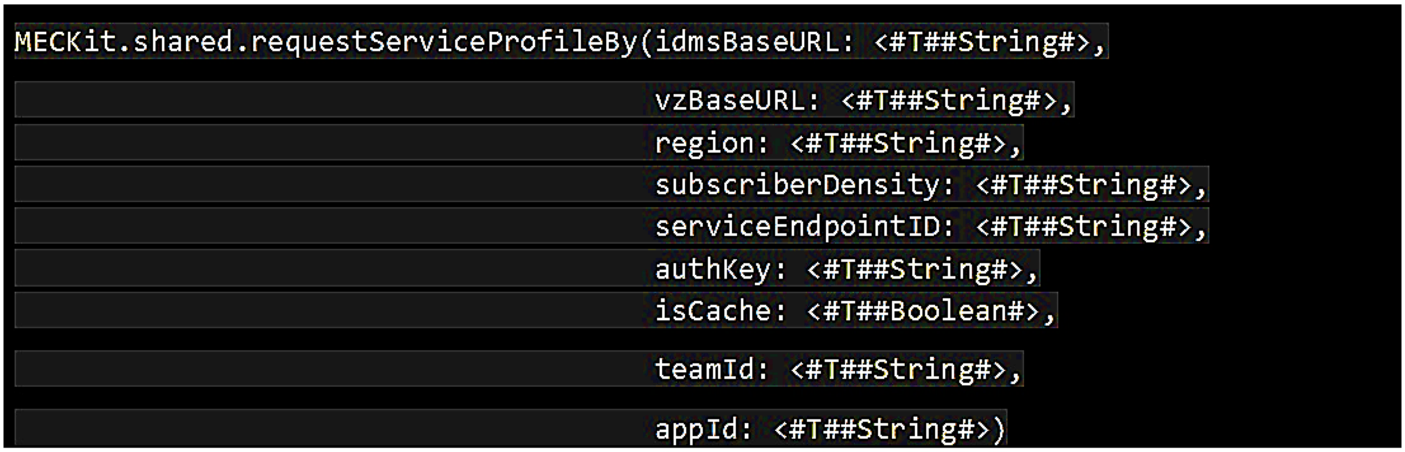
In the **ViewDidLoad** function, set the **delegate** operation:



```
MECKit.shared.delegate = self
```

4. The Edge Discovery API

To obtain optimal endpoints, the developer must invoke this API:



```
MECKit.shared.requestServiceProfileBy(idmsBaseURL: <#T##String#>,
                                      vzBaseURL: <#T##String#>,
                                      region: <#T##String#>,
                                      subscriberDensity: <#T##String#>,
                                      serviceEndpointID: <#T##String#>,
                                      authKey: <#T##String#>,
                                      isCache: <#T##Boolean#>,
                                      teamId: <#T##String#>,
                                      appId: <#T##String#>)
```

Depending on the use case, the developer must input the parameters.

These parameters are common to both SDK Use Cases

Variable	Description	Required?	Type
region	Developers aren't expected to pass this parameter, based on their mobile phones network location. The optimal endpoint will be returned. However, if a developer wants an edge discovery search in a specific region, the value can be passed. The developer must know the valid region in advance. The region API allows developers to find Verizon-supported regions. Additionally, the edge discovery API documentation lists the supported regions.	No	String
subscriberDensity	The minimum number of 4G/5G subscribers per square kilometer.	No	String
serviceEndpointID	This is the value returned after the endpoint is registered using the Register Edge Application Service Endpoints API (on the EDS portal)	Yes	String
isCache	Toggle input to use cache on the **SDK. True** to use Cache and **False** to avoid using the cache. If cached, the data is cached for up to 10 minutes	No	Boolean

The other parameters are use case-specific and explained below.

4.1 SDK Use Cases

4.1.1 SDK Usage Option 1

A developer can use these parameters to bypass having to acquire a limited access token. The prerequisite is that the application must be onboarded on the MEC portal as discussed in the Onboarding an Application on the MEC Portal section.

Variable	Description	Required?	Type
idmsBaseURL	This must be: https://ws.idms.myvzw.com/	Yes	String`
teamId	An Apple developer account generated Team ID. This is discussed in Finding a Team ID on the Apple Developer Portal. The teamId is used when calling Apple's attestation API.	Yes	String
appId	The applicationID value is viewed against the Key-set on the MEC portal. This is discussed in Finding an applicationID on the MEC Portal .	Yes	String

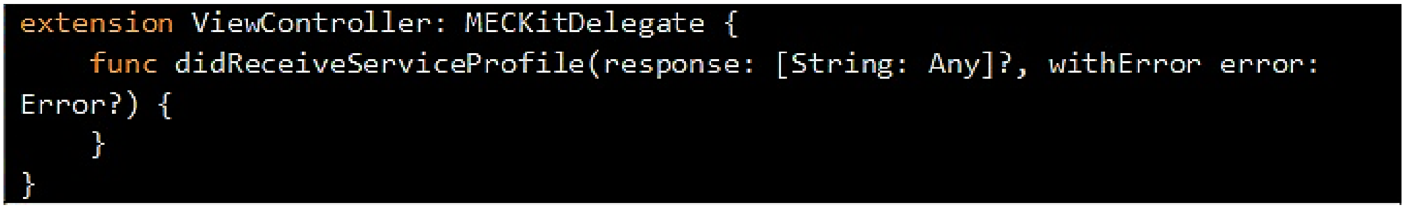
4.1.2 SDK Usage Option 2

This option can be used as a fallback to Option 1 in the event the platform doesn't support Apple's attestation.

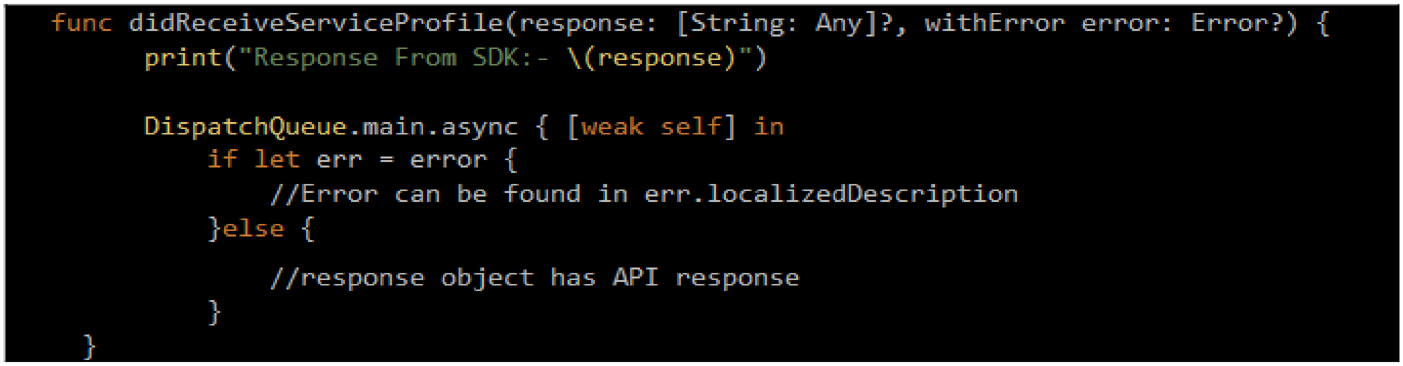
Variable	Description	Required?	Type
vzBaseURL	This must be: https://5gedge.verizon.com/api/ (See Verizon Edge Discovery Service API)	Yes	String`
authKey	This is the restricted token. It can be obtained from the MEC portal as discussed in Obtaining a Restricted Token.	Yes	String

4.2 Managing the Response

Once the API is called, the response can be managed as described in this example.

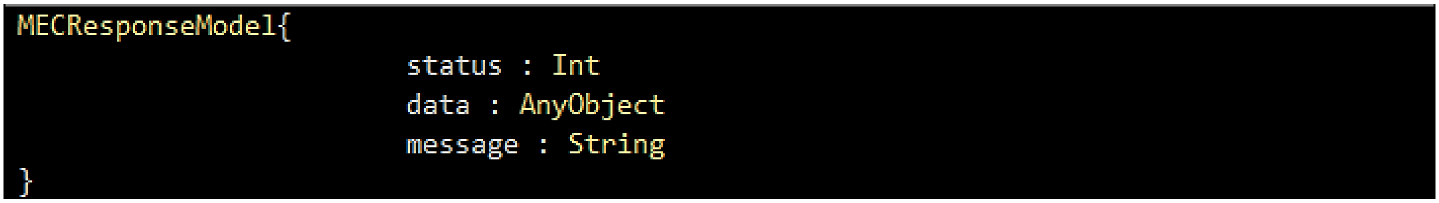


```
extension ViewController: MECKitDelegate {
    func didReceiveServiceProfile(response: [String: Any]?, withError error:
Error?) {
    }
}
```



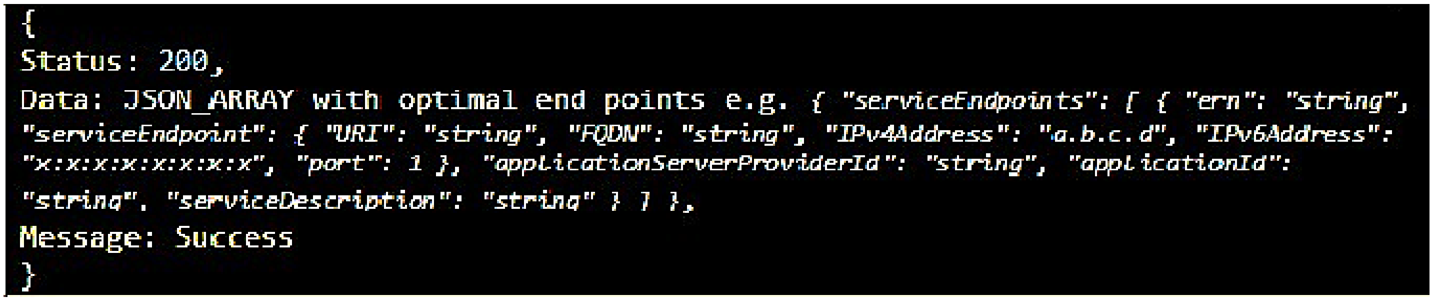
```
  func didReceiveServiceProfile(response: [String: Any]?, withError error: Error?) {
        print("Response From SDK:- \(response)")

        DispatchQueue.main.async { [weak self] in
            if let err = error {
                //Error can be found in err.localizedDescription
            }else {

                //response object has API response
            }
    }
```

The response object model:



```
MECResponseModel{
                        status : Int
                        data : AnyObject
                        message : String
}
```

The status field indicates the error code as defined in the Return Codes section. If the call is a success, the message value is *Success* with a status value of *200* and the data object will have JSON having one or more endpoints. (E.g. Success example).

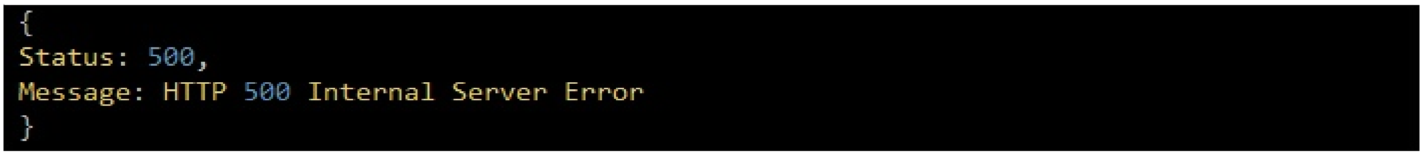


```
{
Status: 200,
Data: JSON_ARRAY with optimal end points e.g. { "serviceEndpoints": [ { "ern": "string",
"serviceEndpoint": { "URI": "string", "FQDN": "string", "IPv4Address": "a.b.c.d", "IPv6Address":
"x:x:x:x:x:x:x:x", "port": 1 }, "applicationServerProviderId": "string", "applicationId":
"string", "serviceDescription": "string" } ] },
Message: Success
}
```

In case of success, the Data field contains a JSON object and keys within that are detailed in the Find Edge Application Service Endpoints section on the EDS portal and defined in the table here.

Variable	Description
ERN	**Edge Resource Name**-The identification of an Edge Resource to use as a connection
URI	**Universal Resource Identifier**-The identification of a resource.
FQDN	**Fully Qualified Domain Name**-The URL including the HTTP or HTTPS designation. In EDS, the service registry, APIs, and the customer calling the API must set FQDN for the edge app server endpoint.
IPv4Address	The 32-bit IPv4 address of the service endpoint.
IPv6Address	The 256-bit IPv6 address of the service endpoint.
port	The HTTP port is used for the service endpoint.
applicationServerProviderId	ID for the Cloud Service Provider (CSP) hosting the application. AWS is the only value accepted.
applicationId	Name of the application host using the service endpoint.
serviceDescription	Details about the service parameters supported by the service endpoint (i.e. latency).

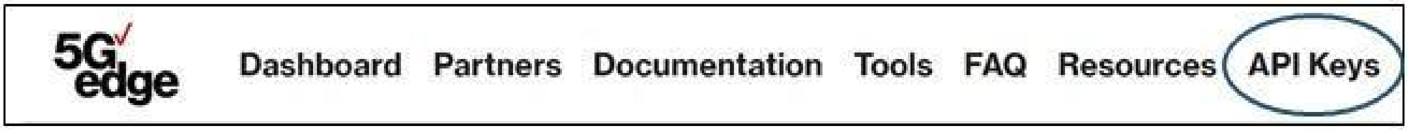
An error example:



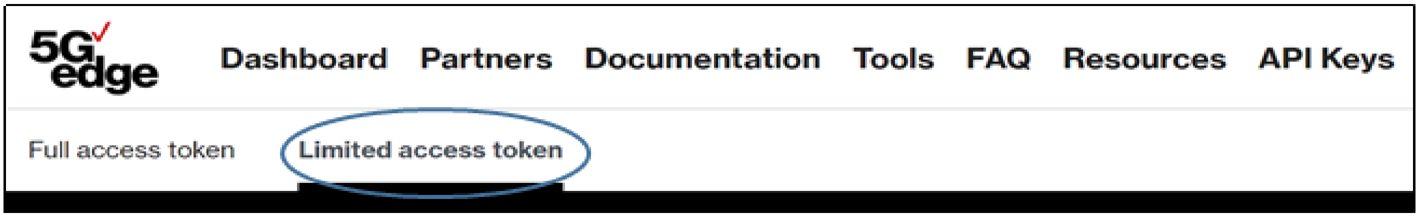
```
{
Status: 500,
Message: HTTP 500 Internal Server Error
}
```

5. Onboarding an Application on the MEC Portal

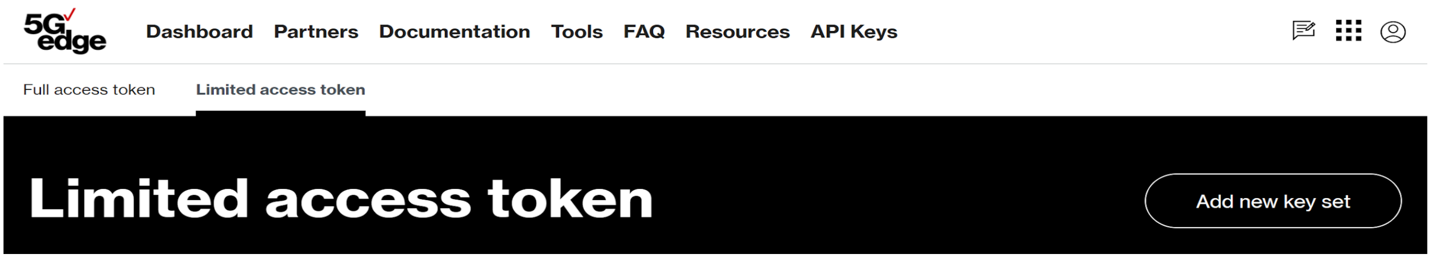
1. login to the MEC portal
2. Click on the **API Keys** section



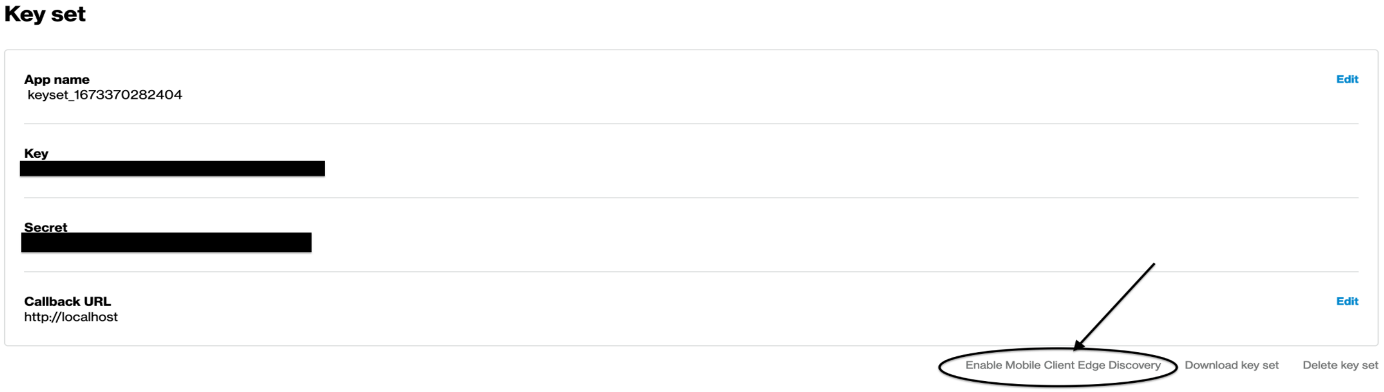
3. Click on the **Limited access token** tab



4. Add a new key set (if it wasn't added)



5. Click on the link ***"Enable Mobile Client Edge Discovery"*** at the bottom of the Limited access token you would want to enable.


Key set

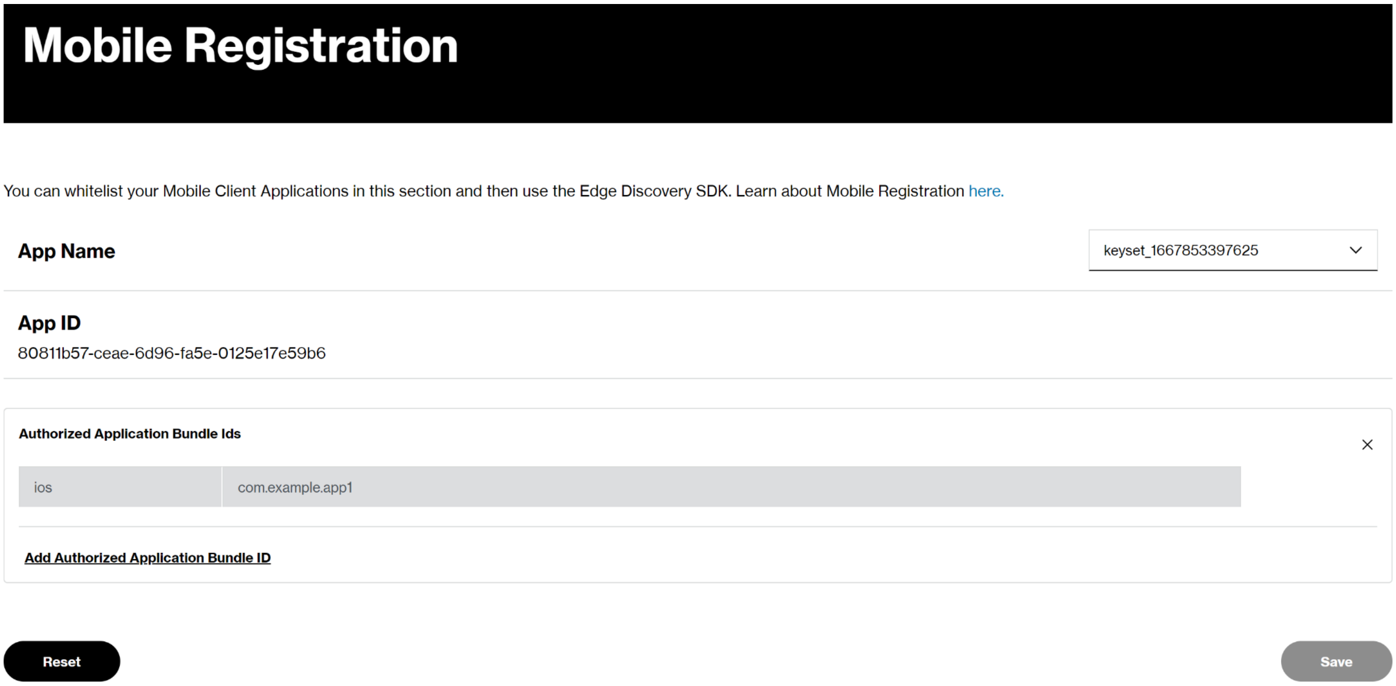
App name
keyset_1673370282404

Key
[redacted]

Secret
[redacted]

Callback URL
http://localhost

6. The mobile client link leads to a section for authorizing applications performing edge discovery via the SDK. Click the ***"Add Authorized Application ID"*** link and add the application's bundle identifier. Additional IDs can be added for multiple applications.



Mobile Registration

You can whitelist your Mobile Client Applications in this section and then use the Edge Discovery SDK. Learn about Mobile Registration here.

App Name: keyset_1667853397625

App ID: 80811b57-ceae-6d96-fa5e-0125e17e59b6

Authorized Application Bundle Ids

ios	com.example.app1

6. Finding a Team ID on the Apple Developer Portal

1. Log in to the Apple Developer Portal
2. Click on the **Account** section
3. Click on the **Membership** section
4. Under the **Membership Information** heading are the **Team Name** and **Team ID** fields

7. Finding an ApplicationID on the MEC Portal

In the Onboarding an Application on the MEC Portal, section, in **Step 6**, the **App ID** value as an **Application ID** value is available.

8. Obtaining a Restricted Token

In Onboarding an Application on the MEC Portal, **Step 5**, the ***My test token*** label is located. Under that heading is the restricted token that is valid for an hour is available. A new token can be computed using the provisioned **Key** and **Secret** as explained under the **Limited Access** heading in the Getting Started section of the **Verizon Edge Developer Portal**.

9. Return Codes

Status code 200 is used for successful calls and respective data from those calls is available in the **data object** area. The other status codes designate failed calls. The reason for those failures is captured in the **message** response parameter.

```
{
Status: 200,
Data: JSON_ARRAY,
Message: Success
}
```

Status	Category
400	HTTP 400 Bad Request
401	HTTP 401 Unauthorized
404	Not_Found
500	HTTP 500 Internal Server Error
200	Success

Return code examples:

Status	Data
200	{ "serviceEndpoints": [{ "ern": "string", "serviceEndpoint": { "URI": "string", "FQDN": "string", "IPv4Address": "a.b.c.d", "IPv6Address": "x:x:x:x:x:x:x:x", "port": 1 }, "applicationServerProviderId": "string", "applicationId": "string", "serviceDescription": "string" }] }

400	{"message":"appId is required"}
	{"message":"serviceEndPoindId is required"}
	{"message":"clientIp is required"}
	{"message":"teamId is required"}
	{"message":"keyId is required"}
	{"message":"appAttestationObject is required"}
	{"message":"packageId is required"}
	{"message":"packageId not found in client details"}
	{"message":"Invalid vzBase URL"}
	{"message":"authKey is required"}

403	{"message":"appId not found: random-test-key-id-12345"}
	{"message":"keyId not found: team-id-123"}
	{"message":"*App attestation Failed*" }

500	{"message":"unable to decode base64 string: invalid characters encountered in base64 data"} {"message":"App Attestation service is not supported"}

FAQs

1. **Which devices are supported?**

   The SDK supports iOS devices running iOS 13 or later.

2. **Which SIMs and networks are supported?**

   Verizon Edge Discovery is available on any device connected to the Verizon network, including 5G and 4G/LTE devices. SIM roaming on the Verizon network and Verizon MVNO / Wholesale carriers are supported.

3. **How do we resolve issues/errors / open questions?**

   To submit an issue for investigation, visit customer support or call 1-800-473-0466. The submission should include:

   - A detailed description and reproduction steps, including resulting errorCode, error message, and any exception information
   - Information on device make, model, and OS version

Terms and Definitions

Term	Definition
5G:	Fifth Generation wireless standard
5G Edge	Verizon MEC computing platform
accessToken	Runtime MEC API OAUTH access token based on MEC API key and secret
appId	ID associated with Verizon MEC keyset
API:	Application Programming Interface
EDS:	Edge Discovery Service
ERN	Edge Resource Number
FQND	Fully Qualified Domain Name
IDMS:	Identity Management Server
LTE:	Long Term Evolution (4G wireless standard providing increased network capacity and speed for IoT devices)
MDP:	MEC Developer Portal
MEC:	Multi-access Edge Compute
MVNO	Mobile Virtual Network Operator
UE:	User Equipment (e.g. Mobile Phone)
UE-ID	A unique identifier for specific User Equipment. (Primarily an IP address, for this document).
UI:	User Interface
VZ:	Verizon