

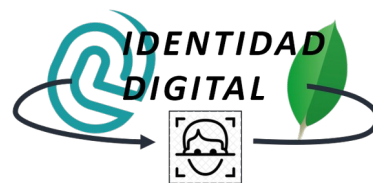
Descripción de servicio getInitString

Fecha de publicación: 30/Marzo/2020

Vigencia: 31/Diciembre/2020

Última revisión: 30/Marzo/2020

Generado por: Víctor Daniel Hernández Berrios



Versión: [1.0.0.0]

Fecha Publicación: 30 de marzo de
2020

Documento Confidencial.

Todos los derechos reservados. El contenido de este documento es propiedad de Identidad Digital, confidencial y con copyright.

Cualquier reproducción completa o en parte está estrictamente prohibida sin el permiso por escrito de Identidad Digital.

Este documento está sujeto a cambios. Cualquier comentario, corrección o pregunta deberá ser dirigida al autor.

Tabla de contenido

Método <i>getInitString</i>.....	2
Detalles del servicio	2
Ejemplo de petición	2
Ejemplo de respuesta	2
<i>Implementacion en android</i>.....	3
<i>Implementacion en iOS</i>.....	4

Este método permite obtener el contenido del archivo ***appLicense.txt*** ubicado en la ruta especificada por **KRLND_ZOOM_WORK_DIRECTORY_PATH**

URL (desarrollo)	http://url-del-servicio/
Tipo de servicio	REST
Método	GET
Endpoint	/getInitString

A efecto de probar el `getInitString` del servicio puede ejecutar el comando CURL:

```
curl -location -request GET 'https:url-del-servicio/getInitString'
```

com.myApp1.example,com.myApp2.example,com.myApp3.example

```

1  {
2    "data": {
3      "licenseText": "appId = \com.                ,com.                ,com.                ,
                     com.                ,com.                ,com.                ,
                     com.                ,com.                ,com.                ,
                     com.                \\\nexpiryDate = 2020-02-29\\nkey =
                     0030450220386f0f10f273aa00f3f663a3bda959bd25afb77345cb9f5ee44ec066a9f9c6820221008ae82b16f970ca10fbc86d3362fa565763a
                     f85123ee4b04f67c0c7135a3a115"
4    },
5    "meta": {
6      "ok": true,
7      "status": 200
8    }
9  }

```

Implementacion en android

- 1.- Se hace el request hacia getInitString
- 2.- Inicializar zoom mediante ZoomSDK.initializeWithLicense pasando los parametros.
 - 1.- Context
 - 2.- El string obtenido del request
 - 3.- El DeviceLicenseKeyIdentifier de su aplicación.

```
@Override
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    activityMainBinding = DataBindingUtil.setContentView(this,
R.layout.activity_main);
    marketingCloudConfigInit();
    // Set the encryption key to be used for ZoOm Server FaceMaps.
    ZoomSDK.setFaceMapEncryptionKey(publicKey);

    Request requestString = new Request.Builder()
        .url(zoomServerBaseUrl+"/getInitString")
        .addHeader("Content-type", "application/json")
        .build();

    new OkHttpClient().newCall(requestString).enqueue(new Callback() {

        @Override
        public void onFailure(Call call, IOException e) {
            e.printStackTrace();
        }

        @Override
        public void onResponse(Call call, Response response) throws IOException {
            String responseString = response.body().string();
            try {
                JSONObject responseJSON = new JSONObject(responseString);
                responseJSON = new
                JSONObject(responseJSON.get("data").toString());
                String licenseText = responseJSON.get("licenseText").toString();
                initZoom(licenseText);
            } catch (Exception e){
                Log.e("Zoom","Excepción");
            }
        }
    });
}

public void initZoom(String licenseText){
    ZoomSDK.initializeWithLicense(
        this,
        licenseText,
        licenseKey,
        new ZoomSDK.InitializeCallback() {

            @Override
            public void onCompletion(final boolean successful) {
```

```
Log.e("Zoom","Activacion completada: "+successful);  
if(succesful) {  
  
    }  
}  
}  
);  
}
```

Implementacion en iOS

- 1.- Se hace el request get hacia getInitString
- 2.- Inicializar zoom mediante Zoom.sdk.initialize pasando los parametros.
 - 1.- licenseText (El string obtenido del request).
 - 2.- licenseKeyIdentifier El licenseKeyIdentifier de su aplicación.
 - 3.- callback de acción

```
override func viewDidLoad() {  
    super.viewDidLoad()  
    utils = SampleAppUtilities(vc: self)  
  
    let url = URL(string: "\(ZoomGlobalState.ZoomServerBaseURL)/getInitString");  
    var request = URLRequest(url: url!)  
    request.httpMethod = "GET"  
    request.addValue("application/json", forHTTPHeaderField: "Content-Type")  
    let task = URLSession.shared.dataTask(with: request) { (data: Data?, response:  
URLResponse?, error: Error?) in  
        if error != nil  
        {  
            print("error=\(error)")  
            return  
        }  
        do {  
            let json = try JSONSerialization.jsonObject(with: data!, options: .mutableContainers)  
as? NSDictionary  
            if let parseJSON = json {  
                let dataInf = parseJSON["data"] as? NSDictionary  
                if let data = dataInf {  
                    let licenseText = data["licenseText"] as? String  
                    print("licenseText: \(licenseText)")  
                    self.initZoom(licenseText: licenseText ?? "")  
                }  
            }  
        }  
    }
```

```
    } catch {  
        print(error)  
    }  
}  
task.resume()  
}  
  
func initZoom(licenseText: String){  
    if(licenseText != ""){  
        Zoom.sdk.initialize(licenseText: licenseText,  
                             licenseKeyIdentifier: ZoomGlobalState.DeviceLicenseKeyIdentifier,  
                             completion: { initializationSuccessful in  
                                 print("Activación completada con éxito")  
                             })  
    }  
}
```