JavaScript SDK Guide

From now on, the ComponentsManager.js file will no longer be loaded locally, and the use of ADO Tech’s official CDN is recommended for better version management, improved performance, and automatic updates.

The import is replaced:

```
<script type="text/javascript" src="Assets/scanovate_card_capture/script.js"></script>
<script type="text/javascript" src="Assets/ComponentsManager.js"></script>
```

with:

```
<script type="text/javascript" src="https://cdn-js.ado-tech.com/latest/ComponentsManager.js"></script>
```

Using latest will ensure that the most recent available version is always used, currently covering versions 1.0 through 3.0.0.1 Playground

IMPORTANT: ADO must be provided with a list of the domains from which the CDN will be consumed so they can be added to the allowlist and the service can be used.

Integrating ADO Technologies' JavaScript SDK into your web application enables you to leverage advanced identity verification features, such as Liveness Detection and Document Capture. This guide provides a structured approach to seamlessly incorporate these functionalities, enhancing the security and user experience of your platform.

Overview

The ADO Technologies JavaScript SDK offers a comprehensive suite of tools designed for real-time identity verification. By integrating this SDK, you can authenticate users by capturing their facial features and identification documents directly within your web application. This process is streamlined and user-friendly, ensuring a high level of accuracy in identity verification.

Requirements

Before starting the integration, ensure you have:

- Access to ADO Technologies' JavaScript SDK url.
- The API key and project name provided by ADO Technologies.

- A clear understanding of the specific features (e.g., Liveness Detection, Document Capture) you wish to implement.

Integration Steps

1. **Include SDK and Assets**: Incorporate the JavaScript SDK and related assets into your web project. This involves linking to the SDK's script files and CSS for styling.
2. **Configure SDK Parameters**: Set up the necessary parameters for the SDK, including the base URL, project name, API key, and product ID. These parameters are crucial for initializing the SDK and ensuring it functions correctly within your application.
3. **Implement User Interface**: Design and implement the user interface through which users will interact with the identity verification features. This includes input fields for configuration parameters and buttons to initiate the capture process.
4. **Capture Process**: Utilize the SDK's functions to capture facial images or documents based on the user's selection. This process should be intuitive, with clear instructions provided to the user.
5. **Handle Responses**: Implement logic to handle the SDK's responses, including success and error callbacks. Display the results appropriately within your application, ensuring users are informed of the outcome.
6. **Testing and Validation**: Thoroughly test the integration to ensure the identity verification process works as expected. Pay special attention to user experience, ensuring the process is smooth and intuitive.

Parameters

To initialize the ADO Technologies JavaScript SDK for identity verification within your web application, you'll need to configure several key parameters. These parameters are essential for tailoring the SDK's functionality to your specific needs and ensuring the verification process operates correctly. Below is an explanation of each parameter required for initialization:

1. **UrlBase**: The base URL of the ADO Technologies service. This URL is the entry point for all SDK requests and should be provided by ADO Technologies. It determines where the SDK sends its verification requests.
2. **ProjectName**: The name of your project as registered with ADO Technologies. This parameter helps the service identify which client is making the request, ensuring that the verification process is correctly attributed and logged.
3. **ApiKey**: A unique key provided by ADO Technologies that authenticates your application's requests. The API key is crucial for securing communication between your application and the ADO Technologies service, preventing unauthorized access.
4. **ProductId**: An identifier for the specific product or service you're using from ADO Technologies. This could relate to different types of verification services offered, such as Liveness Detection or Document Capture.

5. **functionCapture**: Determines the type of capture process to be initiated. This parameter allows you to specify whether you're performing Liveness Detection, Document Capture, or other supported verification processes. The options are typically represented as numerical values or specific strings defined by the SDK.
6. **IsFrontSide**: A boolean parameter indicating whether the document capture (if applicable) should focus on the front side of the identification document. This is relevant for services that require document images as part of the verification process.
7. **UidDevice**: A unique identifier for the device being used to perform the verification. This can be useful for logging, analytics, and ensuring that verification attempts are uniquely associated with a specific device.
8. **Token**: An optional parameter that may be required for additional authentication or session management purposes. If your verification process involves multiple steps or requires maintaining a session state, this token can be used to manage that state across requests.
9. **ProcessId**: An identifier for the specific verification process instance. This can be used to track the progress of a verification attempt or to retrieve results after the process has been completed (How to generate the process Id).

These parameters are typically set by assigning values to the corresponding input fields or variables within your web application's frontend code. Once configured, these parameters are passed to the SDK's initialization function, which prepares the SDK for the capture and verification process based on the provided configuration.

It's important to handle these parameters securely, especially those that could be sensitive, such as the `ApiKey` and `Token`. Ensure that your application's frontend and backend architecture support secure transmission and storage of these values.

Example Implementation

Below is an example HTML structure demonstrating how to set up the SDK in your web application. This example includes the SDK and asset links, configuration inputs, and the capture initiation button.

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0, user-scalable=0, minimal-ui">
    <title>Demo ADO Components</title>
    <script type="text/javascript" src="https://cdn-js.ado-tech.com/latest/ComponentsManager.js"></script>
```

```
    <link rel="stylesheet"
href="Assets/scanovate_card_capture/assets/main.css">
    <link rel="stylesheet"
href="Assets/scanovate_card_capture/assets/loader.css">
</head>
<body>
    <!-- Configuration and Capture UI omitted for brevity -->

    <script>
        function InitCapture() {
            // Capture initialization logic and callbacks
        }
    </script>
</body>
</html>
```

This structure is a starting point for integrating the SDK. Customize the configuration and UI according to your application's needs and the specific features you plan to use.

By following this guide, you can effectively integrate ADO Technologies' JavaScript SDK into your web application, enabling robust identity verification functionalities that enhance the security and user experience of your platform.

Revision #7
Created 22 March 2024 22:33:29 by Admin
Updated 24 March 2026 14:38:29 by roger de avila