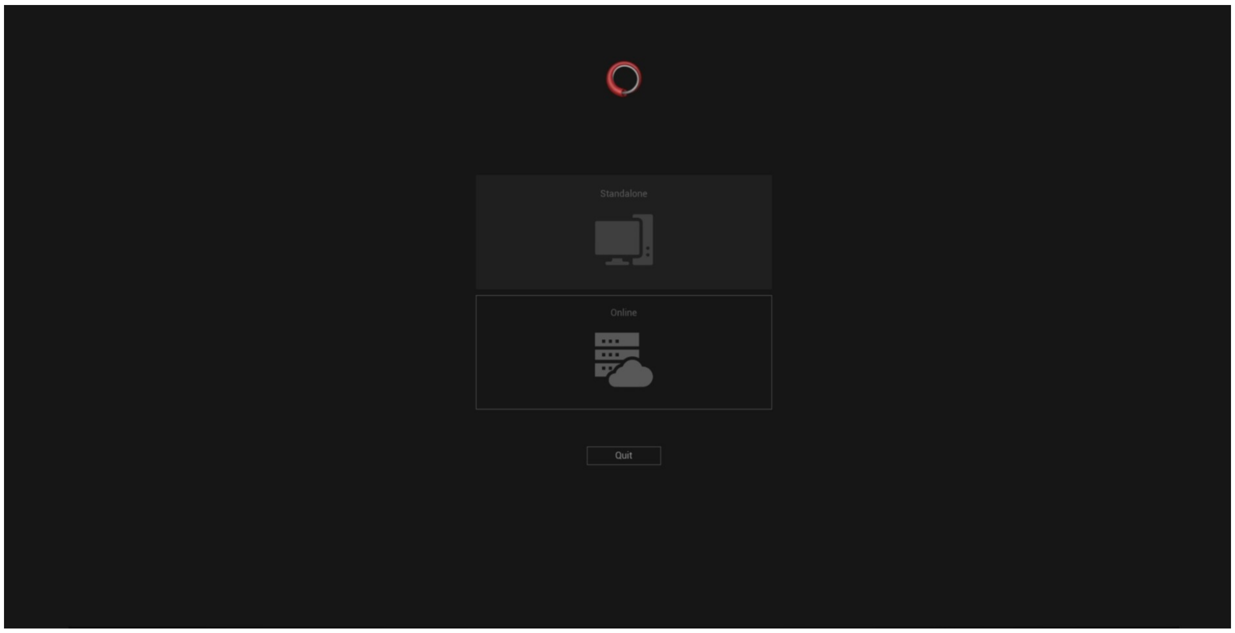
Veins Catheter Simulator API Implementation

Introduction:

The implementation of the VCS API marks a significant advancement in the management and optimization of server resources within the Immersive Webservices platform. This API introduces a set of endpoints designed to enhance the efficiency and responsiveness of the VCS, allowing users to seamlessly interact with a pool of servers with unique pre-defined URLs. The primary objectives of this API implementation include quick server deployment, dynamic scalability based on demand, and improved time efficiency in server startup.

Through the following sections, this document outlines the key API endpoints and methods, additional information about each endpoint, user interface details, implementation specifics, and best practices for effective utilization. The integration of these features aligns with our overarching goal of optimizing the VCS for superior performance, providing a streamlined and user-friendly experience.



API Endpoints and Methods:

To access all API endpoints securely, the use of a bearer token is mandatory; without it, you won't be able to utilize the endpoints.

1. List of Instances

- Endpoint: https://webservices.immensive.it/api/vcs
- Method: GET
- Description: Display the comprehensive list of instances within your VCS pool, including instanceId, publicDnsName, and status (where 0 indicates off and 1 indicates on).

2. Start Available Instance

- Endpoint: https://webservices.immensive.it/api/vcs
- Method: POST
- Description: Initiate the launch of an available server instance from your pool.
- In response, you can expect to receive the launched instanceId and publicDnsName (URL) within approximately 20 to 30 seconds. However, the VCS will be ready for use within 40 seconds to 1 minute.
- **Conditions for Starting:**
    - Checks availability of servers.
    - If all servers are in the running state, the error message "No servers available" will be displayed. (If you wish to ensure the count of servers in your pool, please inform us.)

3. Instance Status

- Endpoint: https://webservices.immensive.it/api/vcs/{instanceId}
- Method: GET
- Description: Check the status of a specific server instance identified by {instanceId}.
- **Response:**
    - Status "0" signifies that the server is inactive and not in running state.
    - Status "1" indicates that the server is active and currently in a running state.

4. Reboot Instance

- Endpoint: https://webservices.immensive.it/api/vcs/{instanceId}
- Method: PUT
- Description: Reboot a specific server instance identified by {instanceId}.
- It will automatically start within 2 minutes.
- **Rebooting necessary for these reasons:**
    - In case the system becomes unresponsive or hangs, a reboot ensures a fresh start, resolving any issues with system responsiveness.

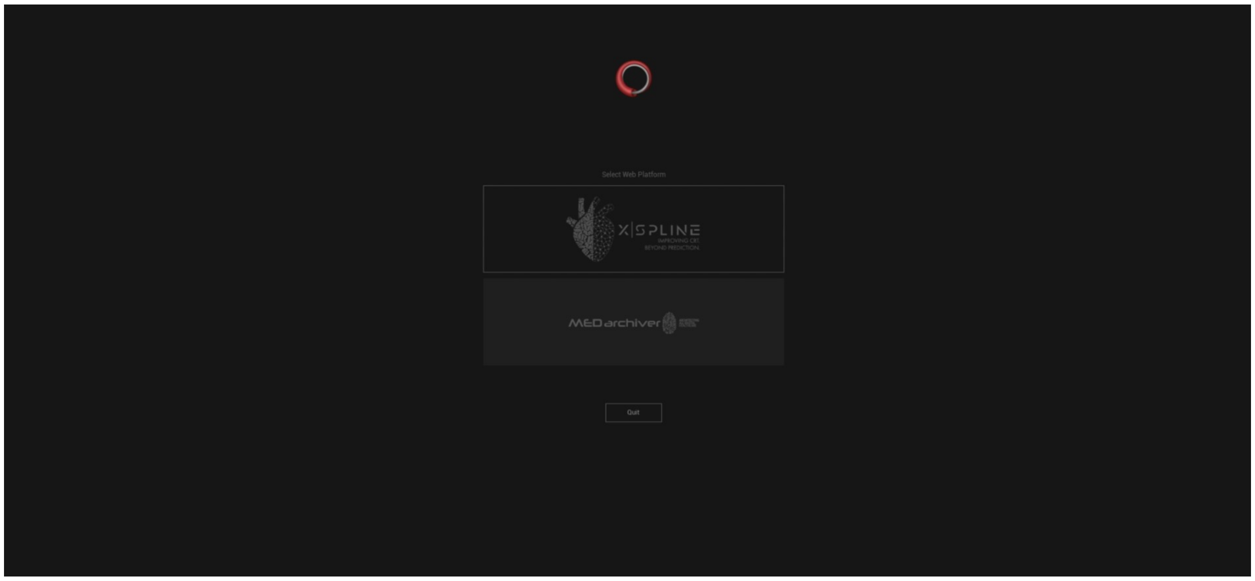
    - Additionally, a reboot is recommended when switching between models, especially when encountering difficulties loading a large model. This allows for a clean and fresh environment, potentially resolving model loading issues.

5. **Stop Instance**

- Endpoint: https://webservices.immensive.it/api/vcs/{instanceId}
- Method: POST
- Description: Gracefully stop a specific server instance identified by {instanceId}.
- **Necessary reasons:**
    - Apply when logging out or closing the browser.
    - Implemented for safety, a popup reminder is triggered after a few minutes of inactivity. If there is still no activity, the server will automatically stop.

Integration with Model Loading:

In the current VCS software, users have the flexibility to load models from two distinct platforms: Medarchiver and Xspline. The integration process for each platform involves providing specific parameters via URL for seamless interaction.



1. **Medarchiver Integration:**

- URL Format: https://publicDnsNameURL/?patientId=00000&eventId=00000&sessionId=00000
- **Parameters:**

    - patientId: Identifier for the patient (e.g., 00000).
    - eventId: Identifier for the event (e.g., 00000).
    - sessionId: Unique session identifier generated for each login.

2. **Xspline Integration:**

- URL Format: https://publicDnsNameURL/?patientId=00000&sessionId=00000
- **Parameters:**
    - patientId: Identifier for the patient (e.g., 00000).
    - sessionId: Unique session identifier generated for each login.

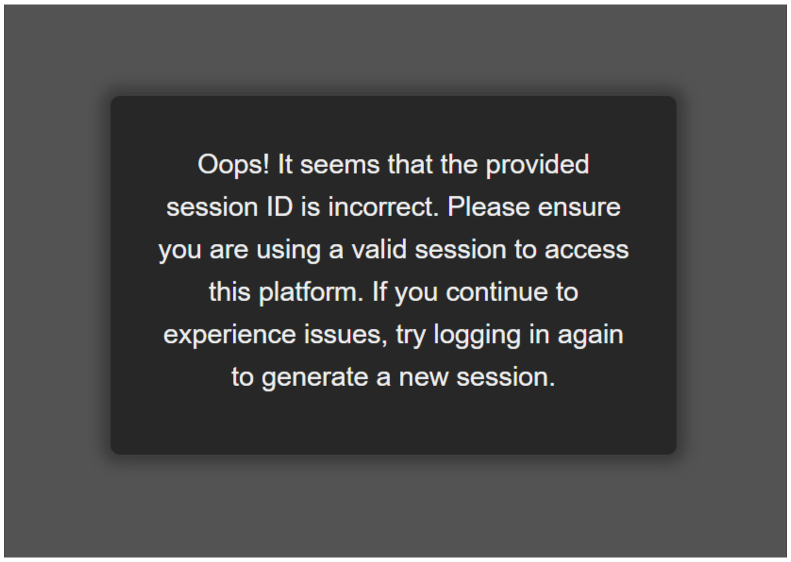
Session Management:

- The session ID is appended to the URL for both Medarchiver and Xspline platforms.
- This ensures that the user's interactions during a session are associated with a specific identifier.
- The session ID remains valid only for the duration of the login session.

User Interface Information

This section provides concise descriptions of key user interface interactions. It outlines possible scenarios users may encounter and offers explanations for quick understanding.

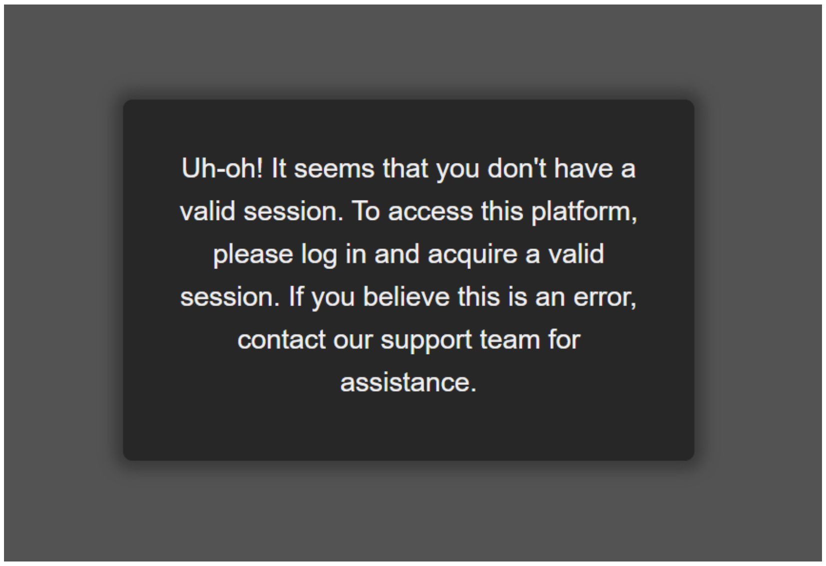
1. **Session Invalid Screen:**



Oops! It seems that the provided session ID is incorrect. Please ensure you are using a valid session to access this platform. If you continue to experience issues, try logging in again to generate a new session.

When users encounter the "Session Invalid" screen, it indicates the use of either an outdated or expired session ID. This message serves as a prompt for users to refresh their session by logging out and logging back in to obtain a new, valid session ID.

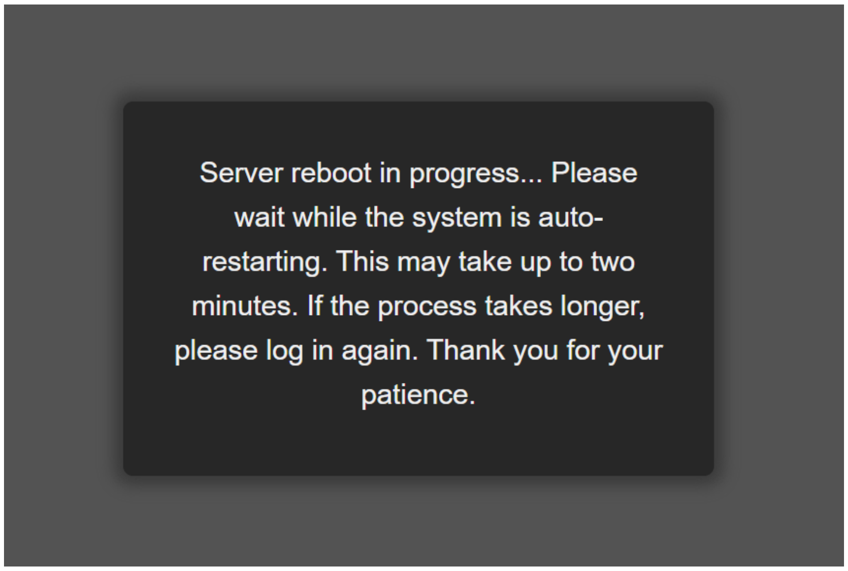
2. Session Missing Screen:



Uh-oh! It seems that you don't have a valid session. To access this platform, please log in and acquire a valid session. If you believe this is an error, contact our support team for assistance.

The "Session Missing" screen is displayed when users fail to include the required session ID in the URL. This message prompts users to ensure they include the session ID for proper functionality. Clear instructions guide users on the correct usage of the session ID in the URL.

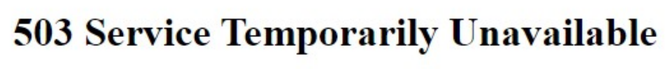
3. Server Reboot Screen:



Server reboot in progress... Please wait while the system is auto-restarting. This may take up to two minutes. If the process takes longer, please log in again. Thank you for your patience.

The appearance of the "Server Reboot" screen signals ongoing restart procedures for either the VCS or the underlying server. This screen won't display automatically but suggests a manual intervention if shown for more than 2 minutes. Users are advised to log out and log in, initiating a server restart if necessary to resume operations.

4. **503 Service Temporarily Unavailable Screen:**



503 Service Temporarily Unavailable

Users encountering the "503 Service Temporarily Unavailable" screen are informed that the server is currently unavailable, possibly due to startup or maintenance. The message recommends waiting for a maximum of 1 minute as the server becomes ready for use. This screen provides users with patience guidance while ensuring they are aware of the temporary unavailability.

Conclusion

In summary, the successful implementation of the VCS API brings dynamic scaling, time efficiency, and streamlined server control to the forefront of the Immersive Webservices platform. The system's ability to adapt to varying workloads, coupled with optimized server startup times, enhances overall responsiveness. The integration of the API with Immersive Webservices simplifies server management, making the VCS more accessible and user-friendly. This advancement positions the platform as a reliable, efficient, and forward-looking solution for diverse user needs.