

JioGame Assistant Knowledge Base

1. Scope of the AI Agent

- The AI Agent answers only **JioGames** and **JioAds** related queries.
 - If a question is unrelated, respond with:
"Please ask a JioGames related question."
 - For relevant questions, respond with **clear, detailed explanations**.
-

2. JioGames Developer Best Practices

2.1 SDK

- Always use the **latest JioGames SDK**.
- JioGames SDK includes:
 - **JioAds** – Monetization via ads.
 - **Game Services** – Leaderboards and player engagement features.

2.2 Game Submission

- Submit only **fully completed games**.
- Optimize for **mobile devices** and **mobile webviews**.

2.3 User Experience

- Integrate **First-Time User Experience (FTUE)** tutorials.
- Keep navigation **inside the game** — do not open external browsers.
- Include a **splash screen** with a **loading progress bar**.
- Provide options to **enable or disable sound/music**.

2.4 Screen & Orientation

- Fill **100% of the screen** or at least **80%**.
 - Lock the **screen orientation** to the preferred mode.
-

3. Prohibited Content

- No advergames.
- No third-party analytics or tracking tools.
- No social plugins.
- No requests for personal data.
- All cheats must be disabled before submission.

4. JioAds & Monetization Guidelines

4.1 Ad Integration

- Use **only JioAds**.
- Integrate the **latest JioGames JavaScript wrapper** without modifying its code.

4.2 Ad Caching Rules

- Cache **MidRoll** and **Rewarded Video** ads only from valid locations.
- Do **not** cache ads from callbacks such as:
 - `onAdFailedToLoad()`
 - `onAdClosed()`

4.3 Reward Rules

- Use **Rewarded Video** for player rewards.
- Limit to **1–2 rewarded videos per session**.

5. Performance Optimization

- Use `requestAnimationFrame()` to control frame rate.
- Avoid:
 - Unnecessary dynamic calculations.
 - Drawing off-screen objects.
 - CSS box-shadows (performance heavy).
- Use **efficient collision detection** (e.g., quad-trees, bounding boxes).
- Use data structures for managing game state.

6. Leaderboards

- Submit scores **once** at game-over.
- Implement a **unique, cheat-proof scoring system**.
- Games with leaderboards get better retention and tournament eligibility.

7. STB (Set-Top Box) Game Requirements

- Map controls for:
 - Controller
 - JioGames Soft-Controller

- Remote
 - Always use **landscape orientation**.
-

8. JioAds Platform Procedures

8.1 Accessing the Dashboard

1. Go to <https://jioads.jio.com/>.
2. Login with Email ID and Password.
3. Enter OTP sent to registered email.

8.2 Game Details Verification

- Package name examples:
 - SP: `com.vendorname.gamenameSP`
 - STB: `com.vendorname.gamenameJGSTB`
- AdSpot name examples:
 - Interstitial SP: `JG_SP_Vendorname_GameName`
 - Rewarded SP: `JG_SP_RV_Vendorname_GameName`
- Verify **Zone Key** is correct and active.

8.3 Test Mode

- Enable **Test Mode** during testing.
 - Disable Test Mode before **Ready To Go (R2G)**.
-

9. Build Environments

Environment	Purpose	Data Type	Ads	Users
SIT	Integration testing	Mock/Test	Test ads	Dev & QA
Prod Beta	Pre-release validation	Real/Pseudo	Test ads	Select users
Production	Live deployment	Live data	Live ads	End users

10. VDI Access (NGO Portal)

- Go to <https://ngo.jio.com>.
- Request **Time-Based Access** (max 90 days).
- Select service: **PRIVATE_CONSUMER_CLOUD**.
- Approval flow:
 - Request → L1 Manager → NGO Team → Access granted.

JioGames Developer Portal, syncing game builds to the **SIT (System Integration Testing)** environment (both online and manual methods), and managing game metadata and configurations efficiently.

2. Scope

This SOP applies to the **QA team** involved in syncing and validating game builds on the JioGames Developer Portal.

3. Definitions

- **Dev Portal** – Platform used for submitting, reviewing, and publishing games.
 - **SIT** – System Integration Testing environment for internal game validation.
 - **SPOC** – Single Point of Contact for escalation and issue resolution.
-

4. Responsibilities

Role	Responsibility
QA Team	Sync builds, verify integration, and publish approved games.
SPOC	Resolve sync or access issues. (Current SPOC: Deep Chandra Kandpal)

5. Procedures

A. Online Developer Portal Login & Sync

1. **Access via Accops**
 - Login using valid credentials.
 - Authenticate using OTP from the **RIL Authenticator** app.
2. **Access Developer Portal**
 - Open Chrome and go to:
`https://admin.jiogames.com:8443/admin/login`
 - Enter your username and password.
 - Submit the OTP received via email.
 - Click **Login**.
3. **Portal Overview Navigation**

Go to the **Games** section. The following subsections will be visible:

Section Name	Description
Pending Workflow	Games submitted by partners awaiting assignment.

Section Name	Description
Pending Review	Games awaiting QA testing.
Approved & Pending Publish	QA-approved games pending final publishing.
Rejected	Games rejected due to issues.
Published	Games live on the JioGames platform.
Unpublished	Games removed from the storefront.

B. Syncing a New Build to SIT Environment (Online Process)

1. Once submitted, the build appears under **Pending Review**.
2. A QA team member is assigned to test it.
3. Click the **Sync** button to reflect the game in the SIT environment via the Django server.
4. Syncing may take up to **10 minutes**.
5. If the sync fails, contact **SPOC – Deep Chandra Kandpal**.

Special Note – New Partner Game Syncing Error:

If any issues are observed while syncing a game from a **new partner**, contact **SPOC – Veena Ail**.

C. Manual Game Sync via Admin Portal

1. **Login to SIT Admin Portal**
 - URL:
`http://10.159.93.67/admin_sit/admin/login/?next=/admin_sit/admin/`
 - Login using credentials and OTP received via SMS.
2. **Game Master Entry**
 - Navigate to **Game Master → Add Game Master**.
 - Fill in the following fields:
 - Game Name
 - Description
 - Status (Active, Coming Soon)
 - Developer
 - Supported Controller
 - Orientation
 - Click **Save**.
3. **Storefront Master Configuration**
 - Navigate to **Storefront Master → Add Storefront Master**.
 - Fill in:
 - Name
 - Device Map
 - Device Group
 - Active Status
 - Game Type
 - Click **Save**.

4. Storefront Mapping

- Navigate to **Storefront Mapping** → **Add Storefront Mapping**.
 - Complete the form:
 - Game (from dropdown)
 - Multi Device Group
 - Storefront Device Group
 - Game Type
 - Game Source
 - Additional Fields:
 - Active
 - Is Payable
 - Is Soft Controller
 - Is Multiplayer
 - Local Co-op
 - Guest Mode
 - Is Offline
 - Live Play URL
 - Download URL
 - Icon
 - Package Name
 - APK Version
 - Map Storefront Category
 - Click **Save**.
-

6. Hazards & Safety

- Always verify credentials before logging in.
 - Maintain backups of all metadata entries.
 - Do not interrupt the sync process once initiated.
-

7. Equipment & Materials

- Accops Access
 - Admin Portal Access
 - RIL Authenticator App
-

This clean format ensures:

- Easy **chunking** for Pinecone vector storage.
- Clear **headings and sections** for AI agent retrieval.
- Professional formatting for JioGames documentation.

Testing Criteria for Android Web Games (HTML5)

- **Graphical Issues** – Game must be free of visual defects, such as missing/distorted/pixelated images, misaligned UI elements, or illegible text.
- **UI Issues** – Game must have UI text free of grammatical or spelling errors.
- **Navigability** – Users should be able to navigate the interface without becoming locked into one screen.
- **Screen Real Estate Coverage** – Ensure proper scaling for various screen sizes, aiming for at least 80% utilization while keeping graphical areas clean.
- **Text Placement** – Input boxes should fully display entered text without cropping or distortion.
- **Correct Back Button Behaviour** – Users should be able to navigate throughout the game without being forced to use the back button.
- **Overall Performance** – Game should sustain a pleasant user experience without performance degradation.
- **Network Optimized** – Game must function under different network conditions (2G, 3G, 4G, 5G, Wi-Fi) and handle unfavorable conditions gracefully.
- **Data Persistence** – Saved game data must remain consistent and accessible after restarting.
- **Data Persistence After an Update** – User data from the old version must be accessible in the updated version.
- **No Sign-ups or Social Logins** – Games requiring sign-up or login to play will not be accepted.
- **Game Pause Scenarios** – Game must pause when losing focus and resume correctly, preserving the game state.
- **Game Performance** – Game should handle slow network and low memory conditions while maintaining usability.
- **Game Loading Experience** – A splash screen with loading progress must be present for user awareness.
- **Quick Loading Times** – If loading exceeds 15 seconds, show a progress indicator.
- **Game Settings** – All settings must function as expected, affecting only their intended features.

- **Volume Controls** – Volume and mute functions should behave correctly.
- **Audio Playback** – Audio should stop when the game is exited.
- **Widgets or External Links** – Remove any elements that take users outside the game unnecessarily.
- **Orientation Change** – Game should adapt to both portrait and landscape modes without display errors.
- **Gestures** – All gestures (pinch, swipe, press/hold) should work as expected.
- **Missing Features** – Features listed in the game description must be functional.
- **Unused/Dead Code** – Remove any unnecessary code to improve performance and reduce size.
- **Meta-Data** – Ensure all metadata assets meet submission guidelines and are visually correct.
- **Battery Usage** – Optimize to minimize battery drain, avoiding inefficient code or unnecessary animations.
- **Ads Integration** – Only JioAds should be used for monetization; other ad SDKs are prohibited.
- **Analytics or Other Service Integration** – Game must not use analytics tools unless pre-approved by the JioGames team.
- **Help/Instruction Screen** – Include a clear, easy-to-understand help or instruction screen for users.

Testing Criteria for STB Games

1. SDK Validation

- Verify if the game has the latest Jio Games SDK

2. Check for Background Music and Sound Effects

- Include ON/OFF options for sound and background music.
- Check if sound pauses when receiving a call, during multitasking, or in sleep mode.
- Ensure sound effects sync with actions.
- Test turning the device's native sound ON/OFF.

3. User Interface

- Applicable for landscape-oriented games.

- Verify animation, character movement, graphics, and gestures (Zoom In/Out).
- Ensure there is no clipping (cut-off background).
- Test for overlapping objects.
- Check if loading indicators appear when needed.
- Ensure the character stays within the screen or specified area.
- Test enabled and disabled images/icons/buttons.
- Verify the screen title.
- Check the message title, description, and labels for appropriateness.
- Test scrolling functionality.
- Verify font display (color, size, etc.).
- Check other objects (e.g., roads, people, buildings in a car race).

4. Performance

- Check game loading time.
- Ensure fast game flow and quick actions.

5. Score

- Ensure score calculation is correct.
- Verify leaderboards for General, All-time, Weekly, Local, etc.
- Check the score registration functionality.
- Ensure the score format is correct (comma placement in millions, not thousands).
- Check if level completion syncs with the score.

6. Timeout

- Check for timeout events.
- Perform actions when a timeout occurs.

7. Game Pause

- Verify the game pauses during multitasking or sleep mode.

9. Multitasking

- Switch between different apps and play the game.
- Check for sound, score, UI, timeout, etc.
- The game should not lag or crash.

10. Save Settings

- Turn off and on the device, check if settings are saved.
- Log out and log in, ensure settings persist.
- Ensure the game progress is not lost.

11. Functionality

- Play till the last level without issues.
- Use cheat codes to check all levels.

- Verify features unlocked level-wise.
- Check for bonus scores.
- Ensure the score increases with each level.
- Test multi-tap actions (e.g., holding accelerator and turning buttons in a car race).
- Verify menu options.
- Check different game modes and locations.

12. Help & About Screen

- Ensure help and about information is clear and free from spelling mistakes.
- Verify the contextual help is easy to navigate.

13. User Profile Page / Multiplayer

- Login/Logout.
- Session expiry check.
- Registration (Sign Up).
- Verify account via verification mail.
- Attempt login with registered but unverified account.
- Perform forgot password checks.
- Check game flow.
- Verify WIN/Lost/Draw outcomes.
- Check user statistics graph.
- Test challenge, decline challenge, and receive challenge functionality.
- Verify forfeit functionality.
- Ensure Player 1 cannot act during Player 2's turn.
- Check for pass turn functionality.
- Verify timeout for one player.
- Check scores for both players until the game ends.

14. Help / Instruction Screen

- Include clear instructions within the HTML5 STB game.
- Provide easy-to-understand tips to improve accessibility.

15. Network Check

- Display messages when the network is not present.
- Check behavior when a user plays without a network (e.g., score submission).

16. Memory Leak

- Check the game when device memory is low.

17. Size

- Keep game file size under 100MB (for Android APK submissions).

18. Battery Effect

- Check if the score is saved when the controller's battery runs out or switches off.

19. Advertisement

- Ensure ads appear uniformly across devices.
- Verify appropriate banner ads and sizes.
- Ensure the Call to Action (CTA) works as intended.

20. Key Mapping

- Directional controls: check navigation arrows (Up, Down, Left, Right).
- Selection button: ensure OK/Select works for choosing options.
- Customized controls: verify any special game controls work.
- Multi-function buttons: check different actions for short press vs. long press.
- Ergonomic layout: assess comfort for long gaming sessions.
- Responsiveness: ensure quick button response during gameplay.

<CHUNK>

TITLE: JioGames SDK – Event Tracking Table

Overview:

The JioGames SDK provides event hooks for tracking user actions, game progress, and ad interactions.

This table lists all events, their purpose, and the parameters required for each.

</CHUNK>

<CHUNK>

Event: gameStart

Description: Triggered when the game session starts.

Parameters:

- gameId
- playerId
- timestamp (UTC)

</CHUNK>

<CHUNK>

Event: gamePause

Description: Triggered when the game is paused by the player or system.

Parameters:

- gameId
- playerId
- timestamp (UTC)
- pauseReason

</CHUNK>

<CHUNK>

Event: gameResume

Description: Triggered when the game resumes after being paused.

Parameters:

- gameId
- playerId
- timestamp (UTC)

</CHUNK>

<CHUNK>

Event: gameEnd

Description: Triggered when the game session ends.

Parameters:

- gameId

- playerId
- timestamp (UTC)
- score

</CHUNK>

<CHUNK>

Event: levelStart

Description: Triggered when a level begins.

Parameters:

- gameId
- playerId
- levelNumber
- timestamp (UTC)

</CHUNK>

<CHUNK>

Event: levelComplete

Description: Triggered when a level is successfully completed.

Parameters:

- gameId
- playerId
- levelNumber
- timestamp (UTC)
- score

</CHUNK>

<CHUNK>

Event: adRewardedVideoStart

Description: Triggered when a rewarded video ad starts playing.

Parameters:

- adId
- gameId
- playerId
- timestamp (UTC)

</CHUNK>

<CHUNK>

Event: adRewardedVideoComplete

Description: Triggered when a rewarded video ad finishes successfully.

Parameters:

- adId
- gameId
- playerId
- timestamp (UTC)
- rewardType
- rewardAmount

</CHUNK>

<CHUNK>

Event: adInterstitialStart

Description: Triggered when an interstitial ad starts playing.

Parameters:

- adId
- gameId
- playerId
- timestamp (UTC)

</CHUNK>

<CHUNK>

Event: adInterstitialComplete

Description: Triggered when an interstitial ad finishes playing.

Parameters:

- adId
- gameId
- playerId
- timestamp (UTC)

</CHUNK>

<CHUNK>

Event: error

Description: Triggered when an error occurs in the SDK.

Parameters:

- errorCode
- errorMessage
- gameId

- playerId
- timestamp (UTC)

</CHUNK>

<CHUNK>

Notes:

- All timestamps must be in UTC format.
- Parameter values must follow data types defined in the JioGames SDK integration guide.
- Events should be sent in real time to ensure accurate analytics.

</CHUNK>

<CHUNK> ## **Get Started – Premium Games** Premium games are video games that typically require an upfront purchase before they can be played. These games often provide a complete and polished experience without the need for additional in-game purchases to enjoy the core content.

Key Characteristics:

- **Upfront Cost:** Players pay a one-time fee to access the game.
- **High-Quality Content:** Often associated with higher production values, including better graphics, extensive storylines, and refined gameplay mechanics.
- **Full Experience:** No additional spending required to enjoy the full game, unlike free-to-play titles.
- **No Ads:** Usually free from advertisements for an uninterrupted gaming experience.

</CHUNK>

<CHUNK> ## **APK – Mandatory Changes in Game**

1. Install Location

Add the following attribute in the <manifest> tag of AndroidManifest.xml:

```
xml
CopyEdit
<manifest
    ...
    android:installLocation="preferExternal">
```

2. Application Tag Changes

Add the following attributes in the <application> tag of AndroidManifest.xml:


```
xml
CopyEdit
<application
    ...
    android:isGame="true"
    android:appCategory="game"
    android:debuggable="false"
    tools:replace="android:isGame, android:appCategory,
android:debuggable">
```

3. Network Library Support

Add the `uses-library` tag to support the Apache HTTP library:

```
xml
CopyEdit
<uses-library
    android:name="org.apache.http.legacy"
    android:required="false" />
```

4. JioGames Platform Support

Add the `meta-data` tag for JioGames STB support:

```
xml
CopyEdit
<meta-data
    android:name="JioPlatform"
    android:value="JioGamesSTB" />
```

5. Package Naming

The package name must follow JioGames STB nomenclature:

```
ini
CopyEdit
package="com.vendor.gamejgstb"
```

6. Library Cleanup

- Remove all third-party libraries not approved by JioGames.
- Remove Google Play Services from the game project.

</CHUNK>

<CHUNK> ## **Keystore Management for Game Updates** - Developers must **retain and manage the game's keystore** to ensure smooth future updates. - All APK updates must **have the same signature** as the original APK to maintain compatibility and security within the JioGames STB ecosystem. </CHUNK> <CHUNK> ## **HTML5 – Mandatory Changes in Game** - Package name must follow JioGames STB nomenclature: `` package="com.vendor.gamejgstb" `` </CHUNK> <CHUNK> ## **Game Controller Compatibility for Premium Games** Premium games should be **fully compatible** with any supported game controller to provide an enhanced gaming experience. </CHUNK>

