JSON-Based Story Management Guide

Introduction

This guide explains how to manage your choose-your-own-adventure stories using JSON files. This approach gives you direct control over your story content without needing a visual editor.

Understanding the JSON Structure

The `stories.json` file contains all your story data in a structured format:

```json
{
  "stories": [
    {
      "id": "story-id",
      "title": "Story Title",
      "description": "Short description of the story",
      "theme": "theme-name",
      "font": "Font Name, fallback",
      "titleFont": "Title Font, fallback",
      "accentColor": "#color-code",
      "nodes": [
        {
          "id": "node-id",
          "content": [
            {
              "type": "message-type",
              "text": "Message text content"
            }
          ],
          "choices": [
            {
              "text": "Choice text",
              "nextNode": "next-node-id"
            }
          ],
          "statusIndicator": "Status text"
        }
      ]
    }
  ]
}
```

Key Components:

1. **Story Properties**:
2. `id` : Unique identifier for the story (used in URLs)
3. `title` : Display title of the story
4. `description` : Brief description shown on the home page
5. `theme` : Visual theme identifier (affects styling)
6. `font` : Main font for story text
7. `titleFont` : Font for titles
8. `accentColor` : Highlight color for the story
9. **Nodes**:
10. Each node represents a scene or decision point
11. `id` : Unique identifier for the node
12. `content` : Array of message objects
13. `choices` : Array of choice objects (if applicable)
14. `statusIndicator` : Text shown in the status area (e.g., "♥ 86 BPM")
15. **Messages**:
16. `type` : Message type (system, agent, suspect, narrator, etc.)
17. `text` : The actual message content
18. **Choices**:
19. `text` : The text shown for this choice
20. `nextNode` : ID of the node to go to when this choice is selected
21. **Paywall Nodes**:
22. `isPaywall` : Set to true for paywall nodes
23. `paywallTitle` : Title shown on the paywall
24. `paywallText` : Description text for the paywall
25. `paywallIndicator` : Status indicator for the paywall

Adding a New Story

To add a new story:

1. Create a new story object in the `stories` array

2. Give it a unique `id` (this will be used in the URL)
3. Add all required properties (title, description, theme, etc.)
4. Create at least a "start" node and a "paywall" node
5. Define the content, choices, and flow between nodes

Example: Adding a New Story

Here's an example of adding a new story:

```json
{
  "id": "space-odyssey",
  "title": "Space Odyssey",
  "description": "Navigate the perils of deep space as you command a mission to an uncharted planet.",
  "theme": "space-odyssey",
  "font": "Roboto Mono, monospace",
  "titleFont": "Orbitron, sans-serif",
  "accentColor": "#00b4d8",
  "nodes": [
    {
      "id": "start",
      "content": [
        {
          "type": "system",
          "text": "Mission Log: Day 342"
        },
        {
          "type": "narrator",
          "text": "The ship's alarms blare as you're jolted awake from cryosleep. Red warning lights flash across the command console."
        },
        {
          "type": "crew",
          "text": "Captain, we've detected an anomaly in our flight path. The navigation system is recommending an immediate course correction."
        }
      ],
      "choices": [
        {
          "text": "Follow the navigation system's recommendation",
          "nextNode": "paywall"
        },
        {
          "text": "Run a diagnostic before changing course",
          "nextNode": "paywall"
        }
      ],
      "statusIndicator": "OXYGEN: 98%"
    },
    {
      "id": "paywall",
      "isPaywall": true,
      "paywallTitle": "Continue the mission",
      "paywallText":
"Your next decision could determine the fate of your entire crew and the success of
the mission.",
      "paywallIndicator": "CRITICAL DECISION POINT"
    }
  ]
}
```

Updating Your Website

After editing the JSON file:

1. **If using GitHub**:
2. Upload the updated `stories.json` file to your repository
3. The website will automatically update
4. **If using direct hosting**:
5. Upload the updated `stories.json` file to your web server
6. Replace the existing file in the same location

Tips for JSON Editing

1. **Use a JSON Editor**: Tools like JSONLint or VS Code can help validate your JSON
2. **Keep a Backup**: Always save a copy of your JSON file before making changes
3. **Test Locally**: Open the website locally to test your changes before uploading
4. **Maintain Node Connections**: Ensure all `nextNode` references point to valid node IDs
5. **Consistent Formatting**: Keep your JSON formatting consistent for easier editing

Troubleshooting

- **Story Not Appearing**: Check that the story ID is unique and properly formatted
- **Broken Paths**: Verify that all `nextNode` values reference existing node IDs
- **JSON Errors**: Use a JSON validator to check for syntax errors
- **Styling Issues**: Ensure font names and color codes are properly formatted

---

This approach gives you complete control over your stories while keeping the management process straightforward. By directly editing the JSON file, you can quickly add new stories or modify existing ones without needing additional tools.