SpellChain – A Spring Boot Demo

Prof. Dr.-Ing. Christoph Neumann

Mattis Schulte
OTH-Amberg-Weiden, November 5, 2025

Introduction and Objective

SpellChain is a fun and educational word-building game for 2 to 4 players. Players take turns adding letters to form valid English words while trying to avoid creating invalid prefixes. The objective of the game is to build longer words, which earn more points. Players can choose to either work together or compete against each other to achieve this. *SpellChain* not only encourages strategic thinking but also helps enhance vocabulary skills in an enjoyable and potentially competitive setting. The game uses a subset from the English-language Wiktionary, providing a vast and challenging collection of words. *Please Note: This project has been migrated from a Python console/server implementation to a Java Spring Boot application with a browser-based interface. It supports online play via WebSockets (STOMP); you can play the live demo here: https://spellchain.mattisschulte.io/.*

Basic Rules and Scoring

The rules of *SpellChain* are designed to keep the game fair while also encouraging a competitive atmosphere. These rules consist of the following key elements:

a) **Taking Turns:** Players take turns in a predefined order. On each turn, a player adds either a single letter, a space, or a punctuation mark (*“-”*, *“’”*, *“/”*, *“.”*) to the existing sequence of characters.

b) **Valid Prefix:** After each addition, the sequence must remain a valid prefix of at least one word in the dictionary. For example, if the sequence is *“car”*, adding a *“t”* is valid because “cart” is a word.

c) **Scoring Points:** When a player completes a valid word, they earn points based on the word’s length. The scoring formula is:

$$\text{Points Earned} = \left\lfloor \frac{\text{Word Length} + 1}{2} \right\rfloor$$

For instance, completing the word *“javanese”* (eight letters) would earn the player four points.

d) **Preventing Word Reuse:** After a word has been successfully completed and points have been awarded to a player, it cannot be used again by any player during the game. This rule encourages players to be inventive and strategic in their word choices.

e) **Handling Invalid Sequences:** If a player adds a letter or punctuation mark that doesn't match any prefix in the dictionary, the sequence resets, a new round starts, and the turn passes to the next player. For example, if the sequence becomes *"zzz"*, which isn't the start of any English word, the sequence resets.

f) **Exiting the Game:** At any point, a player can leave the room using the UI, which ends an ongoing game and broadcasts a summary (see below).

After each move, the game will display the current sequence, all players' scores, and the current round to keep everyone updated on the game's progress.

Winning Conditions

The game does not enforce a predefined win condition, allowing players to play indefinitely or set custom rules such as a target score, maximum rounds, or a time limit. If a player exits during an ongoing game, the server ends the game and broadcasts a detailed summary, including each player's total score, the number of rounds, and the words they successfully completed, providing a clear overview of the game's outcome.

Implementation Highlights

SpellChain is now implemented using Java and Spring Boot, while preserving the original smooth and enjoyable game mechanics:

a) **SQLite-backed read-only dictionary:** Instead of an in-memory trie, the game uses a read-only SQLite database that contains the word list and definitions. The server opens it at startup and performs fast, normalized checks for words and prefixes; lookups are safe for multiple players.

b) **WebSockets (STOMP) with Spring Boot:** Real-time updates are delivered over STOMP on raw WebSockets. The server exposes message mappings for creating/joining rooms, starting games, adding characters, and exiting. Game state is broadcast to `/topic/rooms/{roomId}`, and per-user replies (e.g., errors, room creation) are sent to `/user/queue/reply`.

c) **Browser-Based Interface:** The console interface has been replaced by a lightweight web UI reminiscent of the previous terminal interface. Players can create or join rooms, start the game (host), add characters on their turn, and exit the room. The interface shows the live sequence, scores, round count, system messages, and definitions for completed words.