**ES8+ Features for Backend Development**

ECMAScript (ES) has continuously evolved, and ES8 (also known as ES2017) through ES13 (ES2022) introduced several features that have significantly improved backend development

**1. Async/Await (ES8/ES2017)**

Async and await simplify asynchronous code. It provides a cleaner, more readable way to write asynchronous functions and handle promises compared to the traditional .then() and .catch() methods.

2. **Object.entries() and Object.values() (ES8/ES2017)**

Both return an array but **Object.entries()** returns an array of key-value pairs from an object, while **Object.values()** returns an array of the object's values.

3. **String Padding (padStart() and padEnd()) (ES8/ES2017)**

These methods allow you to pad strings at the start or end with specified characters.

4. **Promise.finally() (ES9/ES2018)**

Adds a finally() method to promises, allowing you to execute code after a promise is settled (resolved or rejected), regardless of the outcome.

5. **Rest/Spread Properties for Objects (ES9/ES2018)**

Spread syntax allows you to spread properties from one object into another, and rest syntax allows you to gather the remaining properties into a new object.

6. **for await...of (ES10/ES2019)**

This syntax allows you to iterate over async iterable objects (like promises) in a for-loop-like manner.

7. **Optional Chaining (ES11/ES2020)**

Optional chaining (?.) allows you to safely access deeply nested object properties without having to manually check for null or undefined at every level.

8. **BigInt (ES11/ES2020)**

**BigInt** allows for representing and operating on large integers.

9. **Top-Level Await (ES13/ES2022)**

Top-level await allows you to use the await keyword at the top level of modules without wrapping it in an async function.