समानांतर उपयोग

यह लगता है कि आप "□□□□□□□□□□□□□"का उपयोग फ्रंटएंड विकास के संदर्भ में पूछ रहे हैं। हालांकि, "□□□□□□□□□□□□□"आमतौर पर एक □□□□.□□ पैकेज है जो कई कमांड या स्क्रिप्ट को एक साथ चलाने के लिए उपयोग किया जाता है, जो आमतौर पर बैकएंड या विकास वर्कफ्लो के संदर्भ में होता है, फ्रंटएंड कोड में सीधे नहीं। क्या आपने फ्रंटएंड प्रोजेक्ट सेटअप में इसका उपयोग करने के बारे में पूछा है (जैसे, स्क्रिप्ट चलाना), या आप कुछ और के बारे में सोच रहे हैं, जैसे फ्रंटएंड जावास्क्रिप्ट में समांतर ऑपरेशन (जैसे, प्रॉमिस, एसिंक/एवेट)।

`concurrently` का उपयोग फ्रंटएंड प्रोजेक्ट में

अगर आप एक फ्रंटएंड प्रोजेक्ट पर काम कर रहे हैं (जैसे, रिएक्ट, व्यू, या एंजुलर), तो `concurrently` विकास में कई प्रक्रियाओं को एक साथ चलाने के लिए आमतौर पर उपयोग किया जाता है—जैसे एक विकास सर्वर और एक बिल्ड वॉचर शुरू करना। यहाँ यह कैसे किया जा सकता है:

1. `concurrently` का इंस्टॉल करें पहले, इसे अपने प्रोजेक्ट में एक डेवलपमेंट डिपेंडेंसी के रूप में इंस्टॉल करें:

```
npm install --save-dev concurrently
```

2. `package.json` में स्क्रिप्ट जोड़ें अपने `package.json` में आप जो स्क्रिप्ट चलाना चाहते हैं, उन्हें एक साथ चलाने के लिए परिभाषित करें। उदाहरण के लिए, अगर आप एक फ्रंटएंड डेव सर्वर और एक □□□ प्रीप्रोसेसर जैसे सास चल रहे हैं:

```
{
  "scripts": {
    "start:frontend": "react-scripts start", //
    "watch:css": "sass --watch src/styles:dist/styles",
    "dev": "concurrently \"npm run start:frontend\" \"npm run watch:css\""
  }
}
```

- `start:frontend`: आपका फ्रंटएंड डेवलपमेंट सर्वर चलाता है (जैसे, रिएक्ट, वाइट, आदि)।
- `watch:css`: आपकी □□□ फाइलों को देखता है और उन्हें कॉम्पाइल करता है।
- `dev`: `concurrently` का उपयोग दोनों कमांड को एक साथ चलाने के लिए करता है।

3. समांतर स्क्रिप्ट चलाएं अपने टर्मिनल में, बस चलाएं:

```
npm run dev
```

यह दोनों फ्रंटएंड सर्वर और □□□ वॉचर को एक साथ शुरू करेगा। आप दोनों प्रक्रियाओं से आउटपुट देखेंगे और `concurrently` उन्हें एक साथ चलाए रखेगा।

4. विकल्पिक संरचना आप `concurrently` को विकल्पों जैसे के साथ कस्टमाइज़ कर सकते हैं: - `--kill-others`: अगर एक फेल हो जाता है, तो सभी प्रक्रियाओं को रोक देता है। - `--names`: प्रत्येक प्रक्रिया के लिए लेबल के लिए स्पष्ट आउटपुट।

`package.json` में विकल्पों के साथ उदाहरण:

```
"dev": "concurrently --names \"Frontend,CSS\" --kill-others \"npm run start:frontend\" \"npm run watch:css\""
```

आउटपुट इस प्रकार दिख सकता है:

```
[Frontend]              ...
[CSS] CSS           ...
```

फ्रंटएंड वर्कफ्लो में उपयोग के मामले मान लीजिए आप एक रिएक्ट ऐप बन रहे हैं जिसमें एक अलग □□□ सर्वर है: - फ्रंटएंड: `npm run start` (रिएक्ट डेव सर्वर पोर्ट 3000 पर)। - □□□: `npm run start:api` (जैसे, नोड.जेएस सर्वर पोर्ट 5000 पर)। `concurrently` के साथ, आप दोनों चल सकते हैं:

```
"dev": "concurrently \"npm run start\" \"npm run start:api\""
```

अगर आपने फ्रंटएंड में समांतर जावास्क्रिप्ट का मतलब किया

अगर आपने फ्रंटएंड कोड में समांतर ऑपरेशन को संभालने के बारे में पूछा था (जैसे, एक साथ कई एपीआई को फेटच करना), तो यहाँ `Promise.all` का उपयोग करके एक तेज़ उदाहरण है:

```javascript
async function fetchData() {
  const [userData, postsData] = await Promise.all([
    fetch('https://api.example.com/user').then(res => res.json()),
    fetch('https://api.example.com/posts').then(res => res.json())
  ]);
  console.log(userData, postsData);
}
fetchData();
```

यह एपीआई कॉल्स को समांतर चलाता है और दोनों को रिजॉल्व होने तक इंतजार करता है।

जरूरत पड़ने पर स्पष्ट करें!

मुझे बताएं अगर आपने "फ्रंटएंड में समांतर"से कुछ और का मतलब किया, जैसे एक विशेष लाइब्रेरी, फ्रेमवर्क, या उपयोग मामला, और मैं जवाब को और अधिक अनुकूलित करूँगा!