

Kannst du den folgenden Code bite überprüfen:

```
useEffect(() => {
  if (!ingredient) {
    return;
  }
  else {
    fetch(`https://www.themealdb.com/api/json/v1/1/filter.php?i=${ingredient}`)
      .then(response => response.json())
      .then(data => {
        setDishes(data.meals);
      })

    /* Testabfrage */
    const dishesTest = [];
    dishes?.map((dish) => (
      fetch(`https://www.themealdb.com/api/json/v1/1/lookup.php?i=${dish.idMeal}`)
        .then(response => response.json())
        .then(data => {
          dishesTest.push(data.meals);
        })
    ))
    setDishesComplete(dishesTest);
    console.log(dishesTest);
    console.log(dishesComplete);
    /* Testabfrage */
  }
}, [ingredient]);
```

ChatGPT

Dein Code sieht größtenteils korrekt aus, aber es gibt eine Asynchronitätsproblematik in deiner Testabfrage. Der fetch-Aufruf ist asynchron, daher wird setDishesComplete(dishesTest) und die console.log-Anweisungen möglicherweise ausgeführt, bevor die Anfragen abgeschlossen sind.

Hier ist eine überarbeitete Version des Codes mit einer asynchronen Funktion, um sicherzustellen, dass die Testabfrage erst abgeschlossen wird, bevor setDishesComplete aufgerufen wird:

```
useEffect(() => {
  if (!ingredient) {
    return;
  }
  else {
    fetch(`https://www.themealdb.com/api/json/v1/1/filter.php?i=${ingredient}`)
      .then(response => response.json())
      .then(data => {
        setDishes(data.meals);

        // Testabfrage
        const fetchDishDetails = async () => {
          const dishesTest = await Promise.all(
            data.meals.map(async dish => {
              const response = await fetch(`https://www.themealdb.com/api/json/v1/1/lookup.php?i=${dish.idMeal}`);
              const dishData = await response.json();
              return dishData.meals;
            })
          );
        };
      })
  }
}, [ingredient]);
```

```

);

setDishesComplete(dishesTest);
console.log(dishesTest);
console.log(dishesComplete);
};

fetchDishDetails();
// Ende der Testabfrage
});
}
}, [ingredient]);

```

In dieser überarbeiteten Version wird `Promise.all` verwendet, um die asynchronen Anfragen für jedes Gericht parallel auszuführen, und die `fetchDishDetails`-Funktion wird verwendet, um sicherzustellen, dass die Testabfrage abgeschlossen ist, bevor `setDishesComplete` aufgerufen wird.

kannst du den ompletten Code bitte überprüfen:

```

import { useState, useEffect } from "react";

const Recipe = ({ingredient, onClickRecipe}) => {

  const [dishes, setDishes] = useState([]);
  const [dishesComplete, setDishesComplete] = useState([]);
  const dishesTest = [];

  useEffect(() => {
    if (!ingredient) {
      return;
    } else {
      fetch(`https://www.themealdb.com/api/json/v1/1/filter.php?i=${ingredient}`)
        .then(response => response.json())
        .then(data => {
          setDishes(data.meals);

          // Testabfrage
          const fetchDishDetails = async () => {
            const dishesTest = await Promise.all(
              data.meals.map(async dish => {
                const response = await
fetch(`https://www.themealdb.com/api/json/v1/1/lookup.php?i=${dish.idMeal}`);
                const dishData = await response.json();
                return dishData.meals;
              })
            );
            setDishesComplete(dishesTest);
            console.log(dishesTest);
            console.log(dishesComplete);
          };

          fetchDishDetails();

```

```

    // Ende der Testabfrage
  });
}
}, [ingredient]);

return (
  <div className="output-container">
    {dishesComplete?.length > 0 &&
      <h3>found {dishesComplete.length} recipes:</h3>
    }
    <div className="card-container">
      {dishesComplete?.map((dishComplete) => (
        <div key={dishComplete.dish[0].idMeal} className="card">
          <img alt="dish" src={dishComplete.dish[0].strMealThumb}/>
          <h5 className="card-title">{dishComplete.dish[0].strMeal}</h5>
          <h5 className="card-title">{dishComplete.dish[0].strArea}</h5>
          <h5 className="card-title">{dishComplete.dish[0].strCategory}</h5>
          <button onClick={() => onClickRecipe(dishComplete.dish[0].idMeal)} className="btn btn-primary">open
recipe</button>
        </div>
      )
    )}
  </div>
</div>
);
}

export default Recipe;

```

ChatGPT

Dein Code sieht größtenteils in Ordnung aus, aber es gibt einige Punkte, die verbessert werden können:

Fehler in der `.map`-Funktion:

In der `.map`-Funktion für `dishesComplete` scheint es einen Fehler zu geben. Du versuchst auf `dishComplete.dish[0]` zuzugreifen, aber `dishComplete` ist bereits das Array der Gerichte. Du solltest direkt auf `dishComplete[0]` zugreifen.

Unnötige Verwendung von `dishesTest`:

Die Variable `dishesTest` wird als Konstante deklariert und dann innerhalb des `fetchDishDetails`-Blocks überschrieben. Du kannst die äußere Variable `dishesTest` entfernen, da sie nicht mehr benötigt wird.

Styling-Klasse "output-container":

Die Klasse "output-container" wird in deinem JSX verwendet, aber es sieht so aus, als ob du diese Klasse nicht definiert hast. Stelle sicher, dass diese Klasse in deinem Stylesheet oder in deinem CSS definiert ist.

Hier ist eine überarbeitete Version deines Codes:

```
import { useState, useEffect } from "react";

const Recipe = ({ ingredient, onClickRecipe }) => {
  const [dishes, setDishes] = useState([]);
  const [dishesComplete, setDishesComplete] = useState([]);

  useEffect(() => {
    if (!ingredient) {
      return;
    } else {
      fetch(`https://www.themealdb.com/api/json/v1/1/filter.php?i=${ingredient}`)
        .then(response => response.json())
        .then(async data => {
          setDishes(data.meals);

          // Testabfrage
          const dishesTest = await Promise.all(
            data.meals.map(async dish => {
              const response = await fetch(`https://www.themealdb.com/api/json/v1/1/lookup.php?i=${dish.idMeal}`);
              const dishData = await response.json();
              return dishData.meals[0];
            })
          );

          setDishesComplete(dishesTest);
          console.log(dishesTest);
          console.log(dishesComplete);
          // Ende der Testabfrage
        });
    }
  }, [ingredient]);

  return (
    <div className="output-container">
      {dishesComplete?.length > 0 && <h3>found {dishesComplete.length} recipes:</h3>}
      <div className="card-container">
        {dishesComplete?.map((dishComplete) => (
          <div key={dishComplete.idMeal} className="card">
            <img alt="dish" src={dishComplete.strMealThumb} />
            <h5 className="card-title">{dishComplete.strMeal}</h5>
            <h5 className="card-title">{dishComplete.strArea}</h5>
            <h5 className="card-title">{dishComplete.strCategory}</h5>
            <button onClick={() => onClickRecipe(dishComplete.idMeal)} className="btn btn-primary">open
recipe</button>
          </div>
        ))}
      </div>
    </div>
  );
}

export default Recipe;
```

Stelle sicher, dass du die notwendigen Styles für die Klassen "output-container" und "card" in deinem CSS- oder Stylesheet-Datei definiert hast.