

```

import { useEffect, useRef, useState } from 'react';

const isObjectEqual = (objA, objB) => {
  return JSON.stringify(objA) === JSON.stringify(objB);
};

const useFetch = (url, options) => {
  const [result, setResult] = useState(null);
  const [loading, setLoading] = useState(false);
  const [shouldLoad, setShouldLoad] = useState(false);
  const urlRef = useRef(url);
  const optionsRef = useRef(options);

  useEffect(() => {
    let changed = false;

    if (!isObjectEqual(url, urlRef.current)) {
      urlRef.current = url;
      changed = true;
    }

    if (!isObjectEqual(options, optionsRef.current)) {
      optionsRef.current = options;
      changed = true;
    }

    if (changed) {
      setShouldLoad((s) => !s);
    }
  }, [url, options]);

  useEffect(() => {
    let wait = false;
    const controller = new AbortController();
    const signal = controller.signal;

    setLoading(true);

    const fetchData = async () => {
      await new Promise((r) => setTimeout(r, 1000));

      try {
        const response = await fetch(urlRef.current, { signal, ...optionsRef.current });
        const jsonResult = await response.json();

        if (!wait) {
          setResult(jsonResult);
          setLoading(false);
        }
      } catch (e) {
        if (!wait) {
          setLoading(false);
        }
        console.log('MY ERROR:', e.message);
      }
    };

    fetchData();

    return () => {
      wait = true;
      controller.abort();
    };
  }, [shouldLoad]);

  return [result, loading];
};

export const Home = () => {
  const [postId, setPostId] = useState('');
  const [result, loading] = useFetch('https://jsonplaceholder.typicode.com/posts/' + postId, {
    headers: {
      abc: '1' + postId,
    }
  });

```

```

    },
  });

  if (loading) {
    return <p>Loading...</p>;
  }

  const handleClick = (id) => {
    setPostId(id);
  };

  if (!loading && result) {
    // 1234
    return (
      <div>
        {result?.length > 0 ? (
          result.map((p) => (
            <div key={`post-${p.id}`} onClick={() => handleClick(p.id)}>
              <p>{p.title}</p>
            </div>
          ))
        ) : (
          <div onClick={() => handleClick('')}>
            <p>{result.title}</p>
          </div>
        )}
      </div>
    );
  }

  return <h1>0i</h1>;
};

```