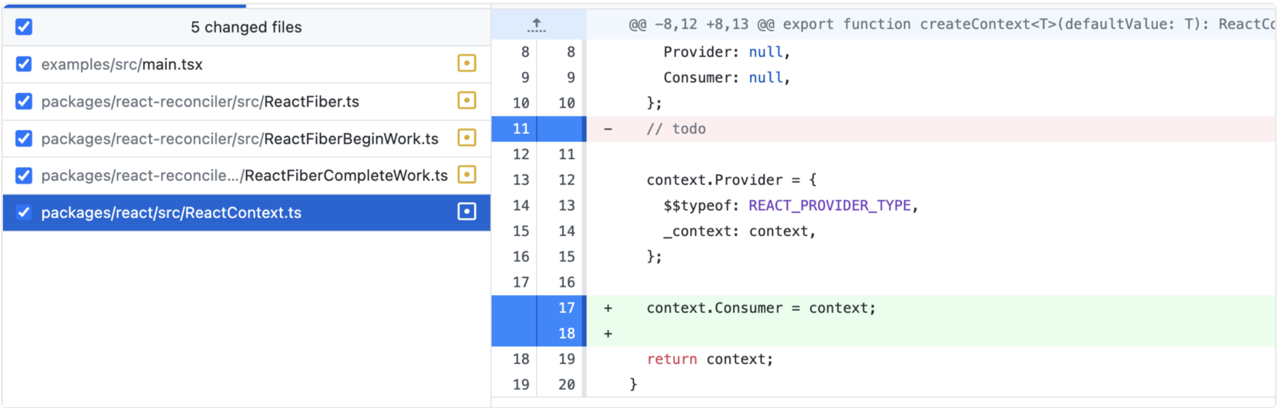
15-5 实现 Consumer

Consumer 组件

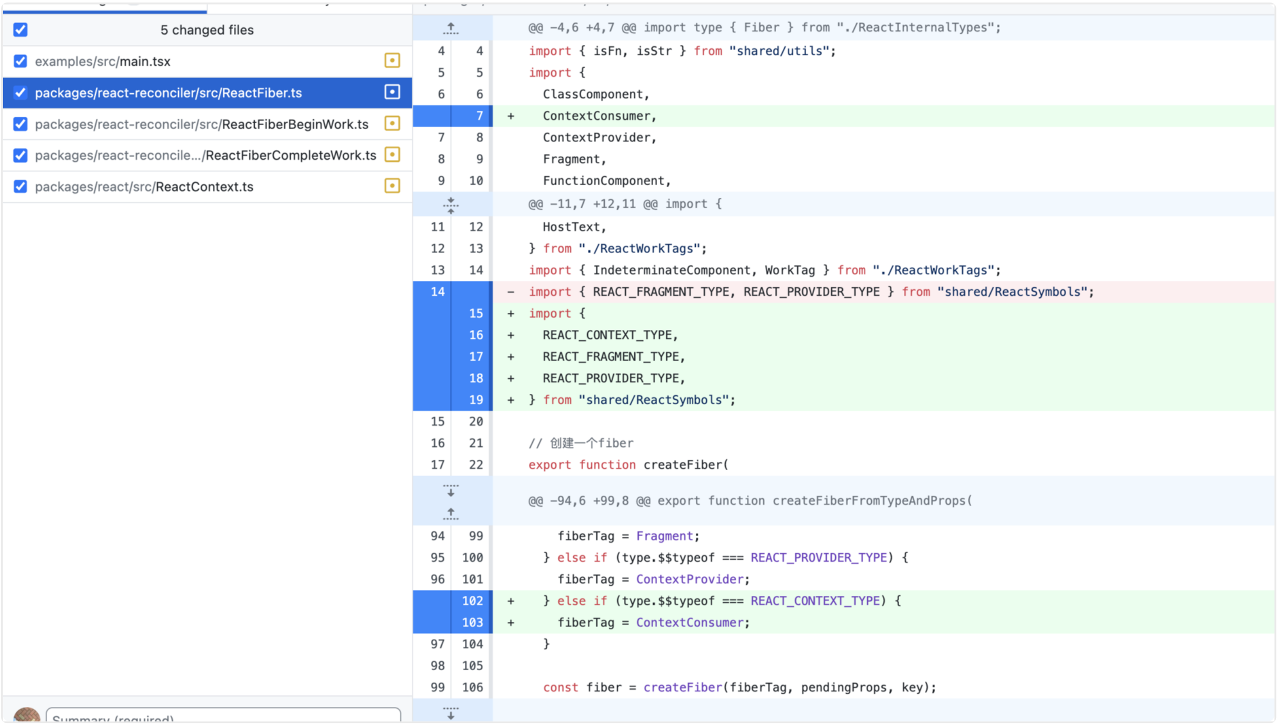


5 changed files
examples/src/main.tsx
packages/react-reconciler/src/ReactFiber.ts
packages/react-reconciler/src/ReactFiberBeginWork.ts
packages/react-reconcile.../ReactFiberCompleteWork.ts
packages/react/src/ReactContext.ts

```
@@ -8,12 +8,13 @@ export function createContext<T>(defaultValue: T): ReactC
        Provider: null,
        Consumer: null,
      };
-     // todo

      context.Provider = {
        $$typeof: REACT_PROVIDER_TYPE,
        _context: context,
      };

+     context.Consumer = context;
+
      return context;
    }
```



☑ 5 changed files
☑ examples/src/main.tsx
☑ packages/react-reconciler/src/ReactFiber.ts
☑ packages/react-reconciler/src/ReactFiberBeginWork.ts
☑ packages/react-reconcile.../ReactFiberCompleteWork.ts
☑ packages/react/src/ReactContext.ts

Summary (required)

```
@@ -4,6 +4,7 @@ import type { Fiber } from "./ReactInternalTypes";
 import { isFn, isStr } from "shared/utils";
 import {
   ClassComponent,
+  ContextConsumer,
   ContextProvider,
   Fragment,
   FunctionComponent,
@@ -11,7 +12,11 @@ import {
   HostText,
 } from "./ReactWorkTags";
 import { IndeterminateComponent, WorkTag } from "./ReactWorkTags";
-import { REACT_FRAGMENT_TYPE, REACT_PROVIDER_TYPE } from "shared/ReactSymbols";
+import {
+  REACT_CONTEXT_TYPE,
+  REACT_FRAGMENT_TYPE,
+  REACT_PROVIDER_TYPE,
+} from "shared/ReactSymbols";

 // 创建一个fiber
 export function createFiber(
@@ -94,6 +99,8 @@ export function createFiberFromTypeAndProps(
     fiberTag = Fragment;
   } else if (type.$$typeof === REACT_PROVIDER_TYPE) {
     fiberTag = ContextProvider;
+  } else if (type.$$typeof === REACT_CONTEXT_TYPE) {
+    fiberTag = ContextConsumer;
   }

   const fiber = createFiber(fiberTag, pendingProps, key);
```

render 阶段

beginWork

TypeScript

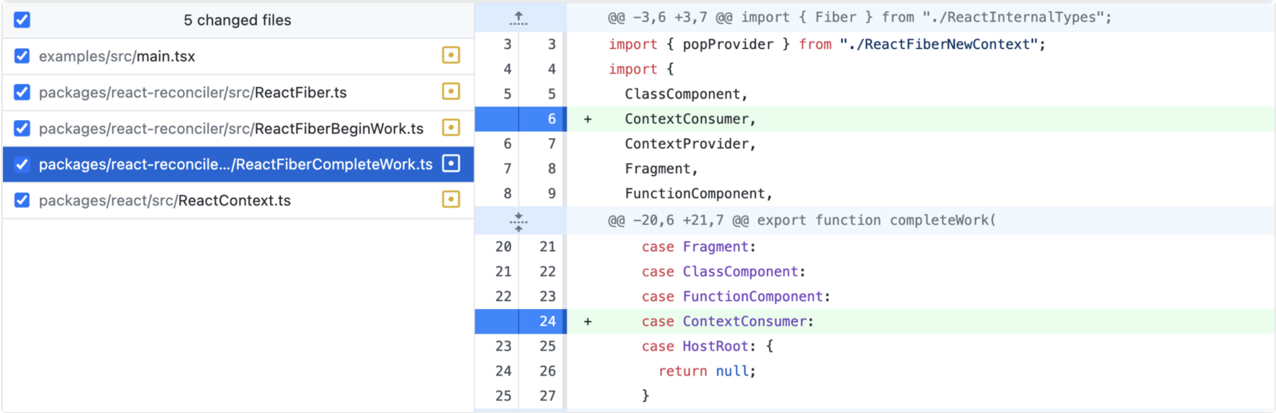
```typescript
function updateContextConsumer(current: Fiber | null, workInProgress:
  const context = workInProgress.type;
  const newValue = readContext(context);

  const render = workInProgress.pendingProps.children;

  const newChildren = render(newValue);
  reconcileChildren(current, workInProgress, newChildren);

  return workInProgress.child;
}
```

completeWork



☑ 5 changed files

☑ examples/src/main.tsx

☑ packages/react-reconciler/src/ReactFiber.ts

☑ packages/react-reconciler/src/ReactFiberBeginWork.ts

☑ packages/react-reconcile.../ReactFiberCompleteWork.ts

☑ packages/react/src/ReactContext.ts

```
@@ -3,6 +3,7 @@ import { Fiber } from "./ReactInternalTypes";
  import { popProvider } from "./ReactFiberNewContext";
  import {
    ClassComponent,
+   ContextConsumer,
    ContextProvider,
    Fragment,
    FunctionComponent,
@@ -20,6 +21,7 @@ export function completeWork(
      case Fragment:
      case ClassComponent:
      case FunctionComponent:
+     case ContextConsumer:
      case HostRoot: {
        return null;
      }
```