



Menu

On this page

HTTP 네트워킹

Bedrock에서 네트워크 통신을 하는 방법을 소개해요.

Fetch API 사용하기

Bedrock에서는 React Native처럼 [Fetch API](#)를 사용해서 네트워크 통신을 할 수 있어요. Fetch API는 비동기 네트워크 요청을 간단히 구현할 수 있는 표준 웹 API예요.

다음은 "할 일 목록"을 가져오는 API를 사용해 "할 일"이 완료됐을 때 취소선을 표시하는 예제예요.

pages/index.tsx

Todo.tsx

tsx

```
import { BedrockRoute } from "react-native-bedrock";
import { useCallback, useState } from "react";
import { Button, ScrollView } from "react-native";
import { Todo, TodoItem } from "../Todo";

export const Route = BedrockRoute("/", {
  component: Index,
});

function Index() {
  const [todos, setTodos] = useState<TodoItem[]>([]);
  const handlePress = useCallback(async () => {
    /**
     * JSONPlaceholder API에서 할 일 데이터를 가져와요.
     * @link https://jsonplaceholder.typicode.com/
     */
    const result = await fetch("https://jsonplaceholder.typicode.com/todos");
    const json = await result.json(); // 응답 데이터를 JSON으로 변환해요.
    setTodos(json); // 가져온 데이터를 상태로 저장해요.
  }, []);
```

```
return (  
  <>  
  <Button title="할 일 목록 확인하기" onPress={handlePress} />  
  <ScrollView>  
    {todos.map((todo) => {  
      return (  
        <Todo  
          key={todo.id}  
          id={todo.id}  
          title={todo.title}  
          completed={todo.completed}  
        />  
      );  
    })}  
  </ScrollView>  
</>  
);  
}
```

예제 영상을 보면 버튼을 클릭하면 네트워크 요청이 발생하고, 화면에 할 일 목록이 표시돼요. 네트워크 요청이 발생할 때 네트워크 인스펙터에서 요청과 응답을 확인할 수 있어요.

0:00 / 0:17

- 네트워크 요청 확인 방법은 [디버깅하기 문서](#)를 참고하면 자세히 알 수 있어요.

다른 라이브러리 사용하기

React Native는 [XMLHttpRequest API](#)를 지원해요. 따라서, 이 API를 사용하는 써드파티 네트워크 라이브러리도 사용할 수 있어요.

자세한 내용은 [React Native 공식 문서](#)를 참고하세요.

Previous page

[화면 이동하기](#)

Next page

[권한이 필요한 함수 앱에 설정하기](#)