

cms-client-ui > src > pages > \_tests\_ > groupEnrollment

ClientFileConfigWizard.tsx ClientList.tsx Group.tsx Group.test.tsx

```
1 import { LocalDate } from "@js-joda/core";
2 import { fireEvent, render, waitFor } from "@testing-library/react";
3 import * as React from "react";
4
5 import { getGroupBySearch } from "../../api/GroupAPI";
6 import { GroupSearch } from "../Group";
7
8 jest.mock( moduleName: "../../api/GroupAPI");
9
10 describe( name: "groups page", fn: () => {
11   beforeEach( fn: async () => {
12     (
13       getGroupBySearch as jest.MockedFunction<typeof getGroupBySearch>
14     ).mockResolvedValue( value: {
15       totalPages: 1,
16       totalElements: 2,
17       size: 2,
18       content: [
19         {
20           clientCode: "interopCode",
21           partBDCode: "D",
22           coverageTypeCode: "EGWP",
23           groupHierarchy: {
24             groupLevel1: "ES123",
25             groupLevel2: "ORARE",
26             groupLevel3: "3006",
27             groupLevel4: "THERA COLLEGE",
28           },
29           groupEnrollment: {
30             callback for describe() > callback for beforeEach() > content > groupEnrollment
```

File Edit View Navigate Code Refactor Build Run Tools Git Window Help cms-client-ui - Group.test.tsx

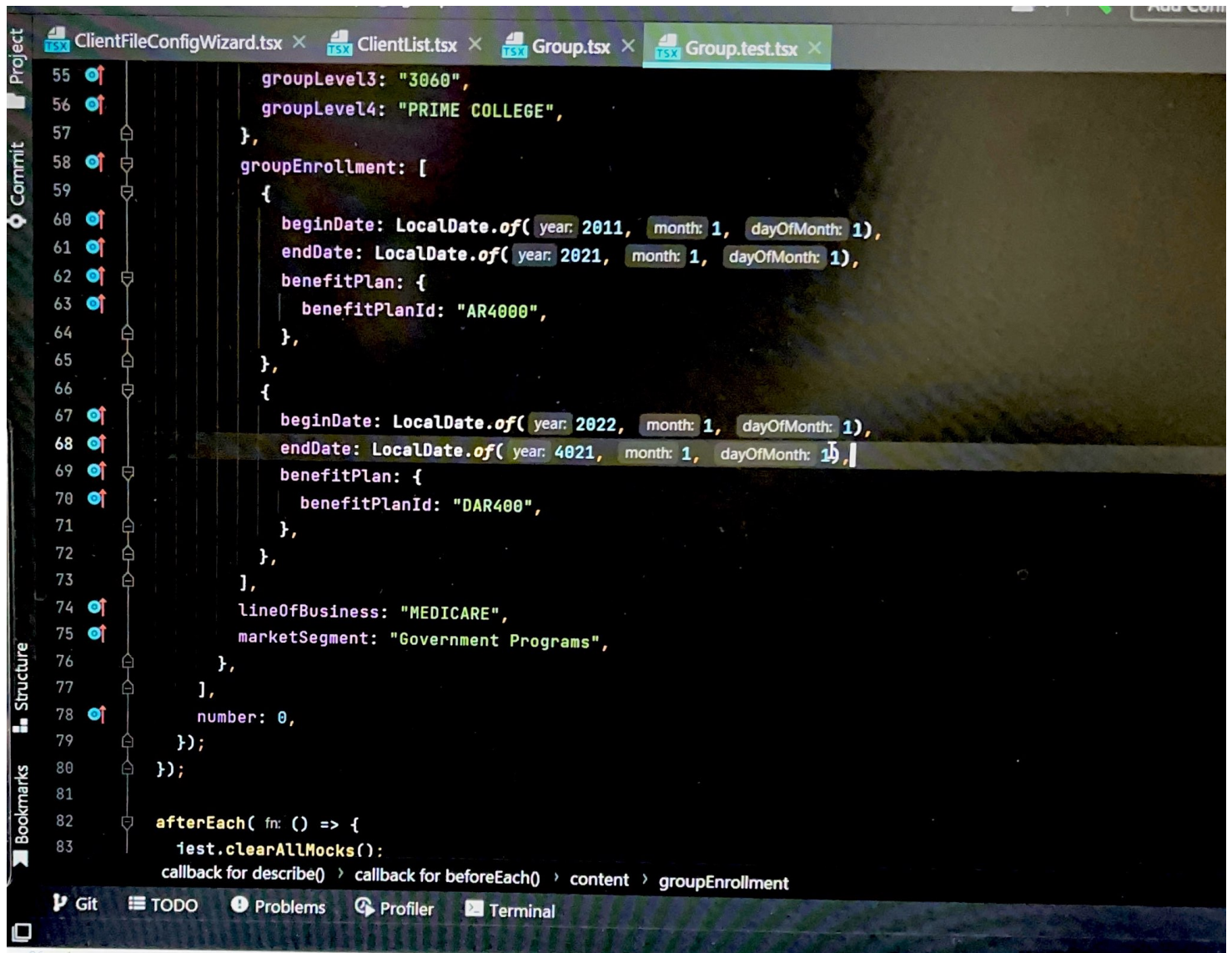
cms-client-ui > src > pages > \_tests\_ > groupEnrollment Add Config

ClientFileConfigWizard.tsx ClientList.tsx Group.tsx Group.test.tsx

```
28 },
29 groupEnrollment: [
30   {
31     beginDate: LocalDate.of( year: 2011, month: 1, dayOfMonth: 1),
32     endDate: LocalDate.of( year: 2021, month: 1, dayOfMonth: 1),
33     benefitPlan: {
34       benefitPlanId: "DAR4000",
35     },
36   },
37   {
38     beginDate: LocalDate.of( year: 2022, month: 1, dayOfMonth: 1),
39     endDate: LocalDate.of( year: 4022, month: 1, dayOfMonth: 1),
40     benefitPlan: {
41       benefitPlanId: "DAR4000",
42     },
43   },
44 ],
45 lineOfBusiness: "MEDICARE",
46 marketSegment: "Government Programs",
47 },
48 {
49   clientCode: "interopCode",
50   partBDCODE: "B",
51   coverageTypeCode: "INDIVIDUAL",
52   groupHierarchy: {
53     groupLevel1: "EARS123",
54     groupLevel2: "AREOR",
55     groupLevel3: "3060",
56     groupLevel4: "PRIME COLLEGE".
57   }
58 },
59 ],
60 },
61 }
```

callback for describe() > callback for beforeEach() > content > groupEnrollment

Git TODO Problems Profiler Terminal





The screenshot shows an IDE window titled "cms-client-ui - Group.test.tsx". The breadcrumb navigation at the top indicates the file path: "cms-client-ui > src > pages > \_tests\_ > groupEnrollment". The editor displays the content of "Group.test.tsx", with line numbers 82 through 110 visible on the left. The code is a Jest test file for a component named "GroupSearch". It includes an "afterEach" block to clear mocks and three "it" blocks for different test scenarios: "should render group Tab", "should Perform Search", and "should Clear Search". The "should Perform Search" test uses "fireEvent" to simulate user interactions and "waitFor" to ensure an asynchronous callback is called. The IDE interface includes a menu bar at the top, a sidebar on the left with "Project", "Commit", "Structure", and "Bookmarks" views, and a breadcrumb at the bottom: "callback for describe() > callback for beforeEach() > content > groupEnrollment".

```
82   afterEach( fn: () => {
83     jest.clearAllMocks();
84   });
85
86   it( name: "should render group Tab", fn: async () => {
87     const component = render(<GroupSearch />);
88
89     expect(component.getByLabelText( args: "Line of Business")).toBeInTheDocument();
90     expect(component.getByLabelText( args: "Group ID")).toBeInTheDocument();
91     expect(component.getByLabelText( args: "Group Name")).toBeInTheDocument();
92     expect(component.getByLabelText( args: "Client")).toBeInTheDocument();
93     expect(component.getByLabelText( args: "Health Plan")).toBeInTheDocument();
94     expect(component.getByLabelText( args: "Benefit Plan Code")).toBeInTheDocument();
95   });
96
97   it( name: "should Perform Search", fn: async () => {
98     const component = render(<GroupSearch />);
99     const SearchByClient = component.getByLabelText( args: "client");
100    fireEvent.change(SearchByClient, options: { target: { value: "interopCode" } });
101    fireEvent.click(component.getByText( args: "SEARCH"));
102
103    await waitFor( callback: () => {
104      expect(getGroupBySearch).toHaveBeenCalledWith(expect.anything());
105    });
106  });
107
108   it( name: "should Clear Search", fn: async () => {
109     const component = render(<GroupSearch />);
110
```

callback for describe() > callback for beforeEach() > content > groupEnrollment

File Edit View Navigate Code Refactor Build Run Tools Git Window Help cms-client-ui - Group.test.tsx

cms-client-ui > src > pages > \_tests\_ > groupEnrollment

ClientFileConfigWizard.tsx ClientList.tsx Group.tsx Group.test.tsx

94 expect(component.getByLabelText( args: "Benefit Plan Code")).toBeInTheDocument();  
95 });  
96  
97 it( name: "should Perform Search", fn: async () => {  
98   const component = render(<GroupSearch />);  
99   const SearchByClient = component.getByLabelText( args: "client");  
100   fireEvent.change(SearchByClient, options: { target: { value: "interopCode" } });  
101   fireEvent.click(component.getByText( args: "SEARCH"));  
102  
103   await waitFor( callback () => {  
104     expect(getGroupBySearch).toHaveBeenCalled();  
105   });  
106 });  
107  
108 it( name: "should Clear Search", fn: async () => {  
109   const component = render(<GroupSearch />);  
110  
111   fireEvent.click(component.getByText( args: "Clear search"));  
112   expect(  
113     component.getByText( args: "Get started by searching for a Group")  
114   ).toBeInTheDocument();  
115 });  
116 });  
117

callback for describe() > callback for beforeEach() > content > groupEnrollment