Community Call

Using app only model with Microsoft Graph APIs for Planner

Bio.tsx

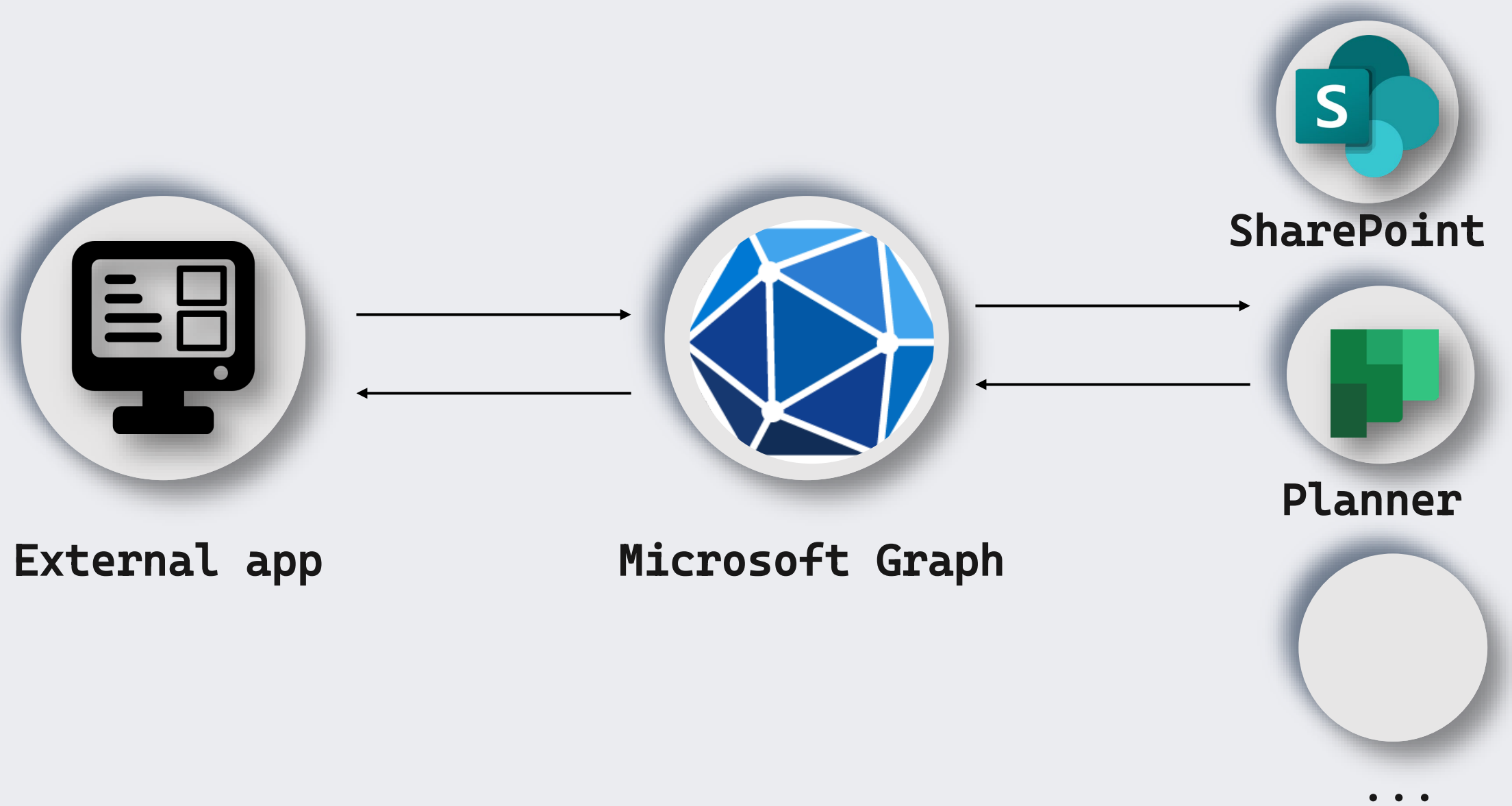
```
return(
      <div>
            <img src="  "/>
            <Name>Anoop Tatti</Name>
            <Work>MVP, Developer, Content+Cloud, UK</Work>
            <Profile link="aka.ms/anoopt" />
            <Blog link="https://anoopt.medium.com" />
            <Twitter username="anooptells" />
            <GitHub username="anoopt" />
      </div>
);
```

Microsoft Graph

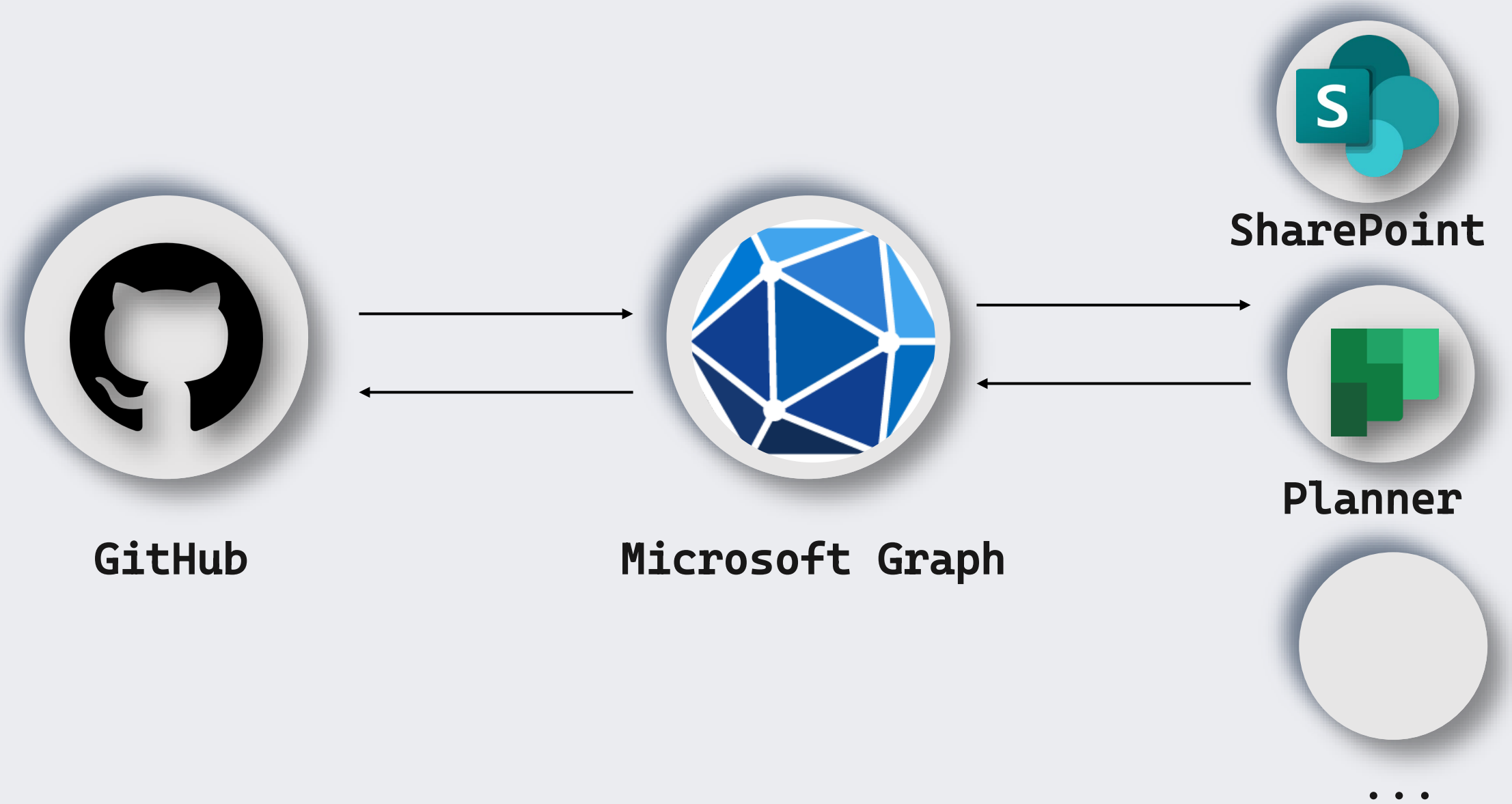
Microsoft Graph is the gateway to data and intelligence in Microsoft 365.

It provides a unified programmability model that you can use to access the tremendous amount of data in Microsoft 365, Windows, and Enterprise Mobility + Security.

Interacting using Microsoft Graph



GitHub + Microsoft Graph



GitHub + Microsoft Graph + Planner

GitHub

- GitHub workflow
- Runs on pull request
- GitHub Action
- Passes data to Graph

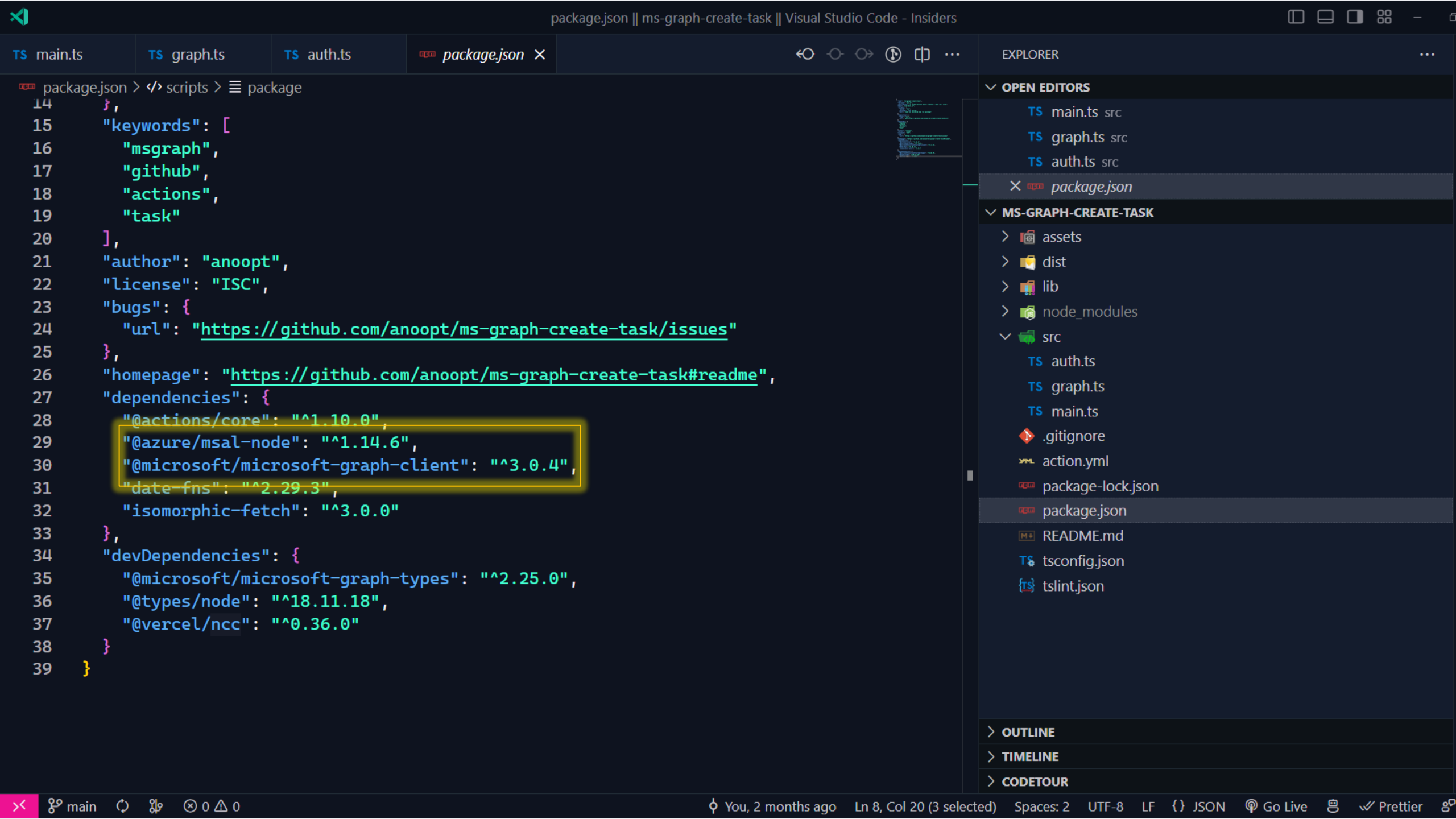
Microsoft Graph

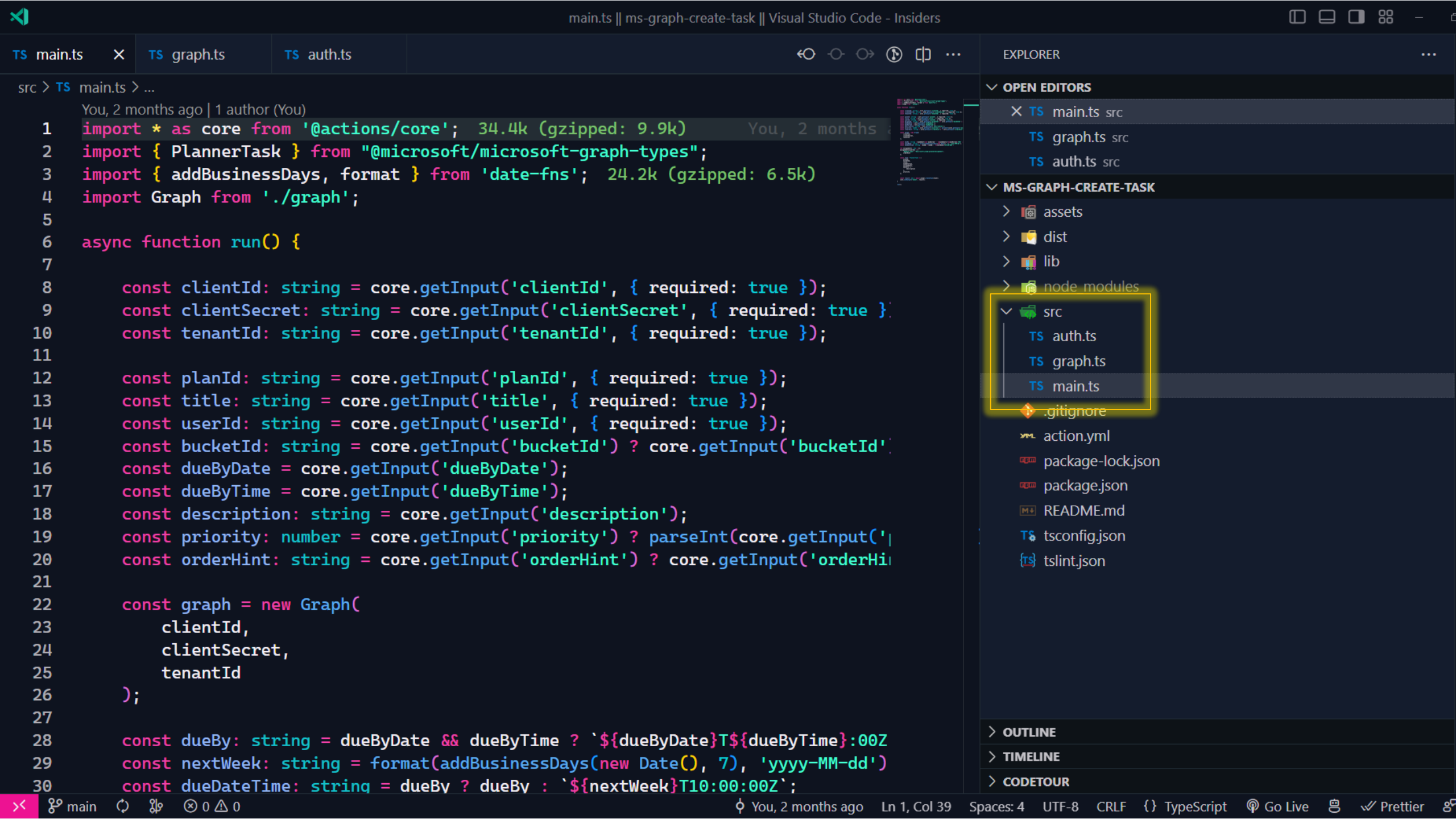
- Verifies auth
- Uses **app only permissions**
- Creates a planner task

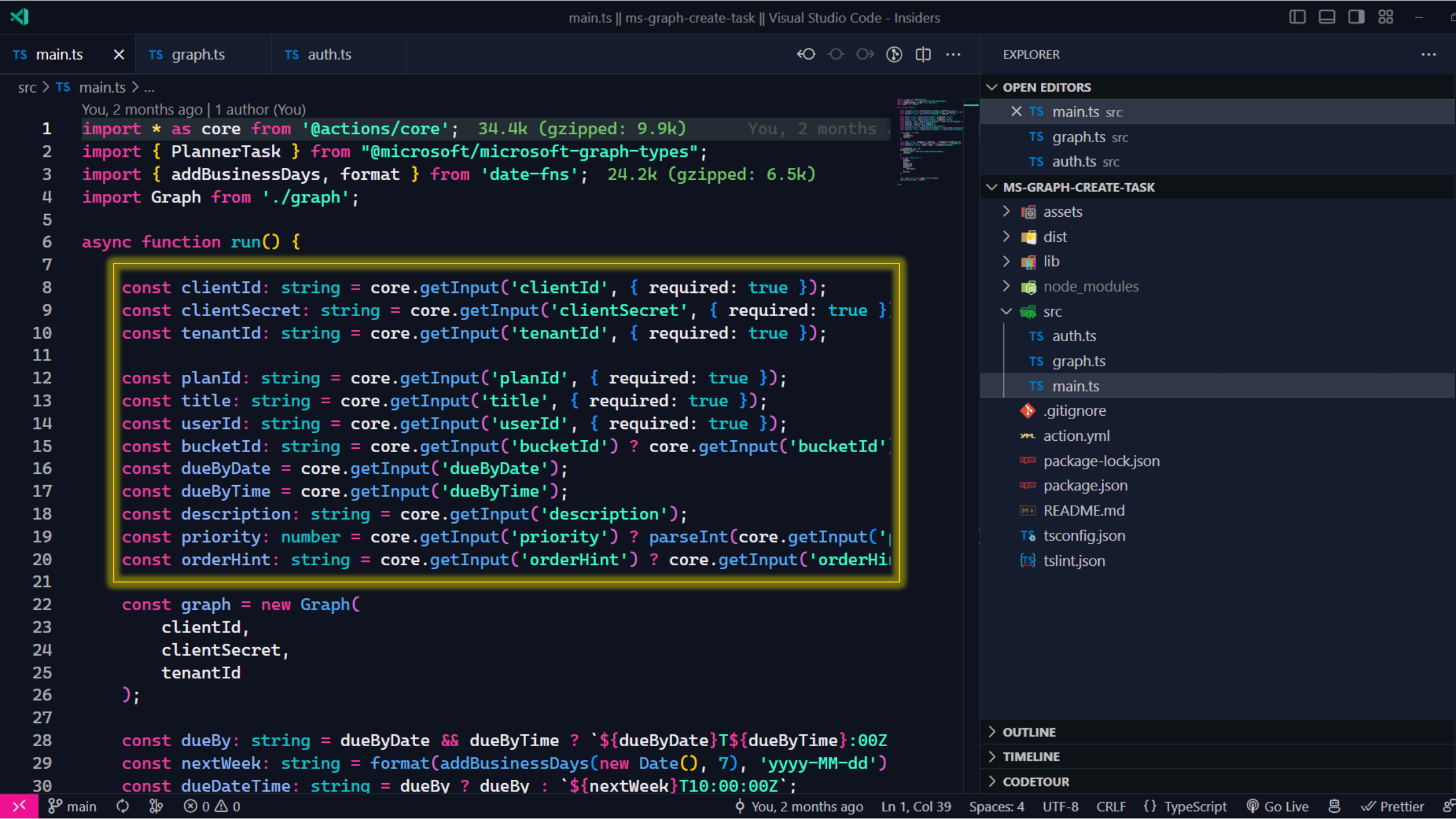
Planner

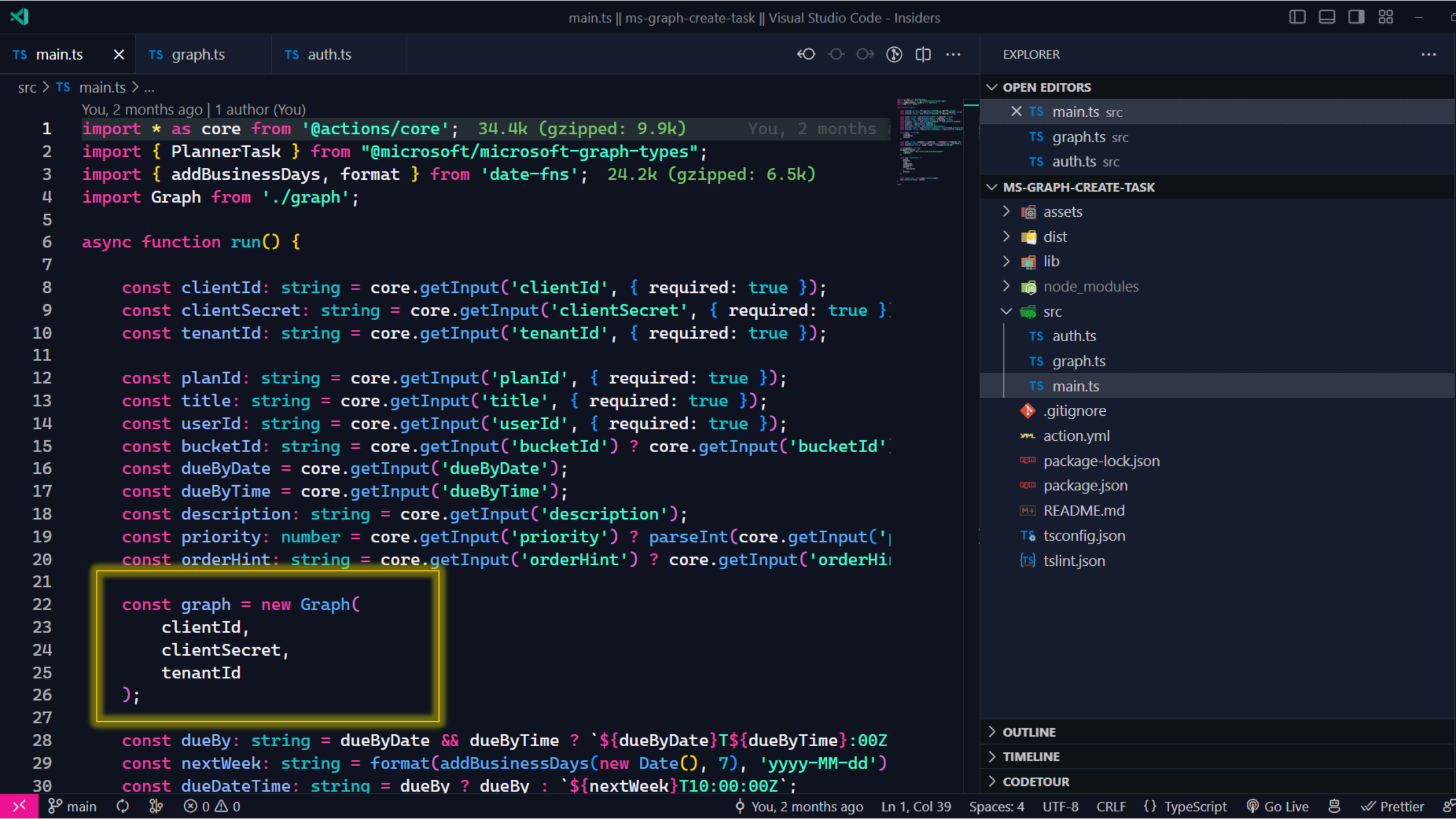
- User interaction with task

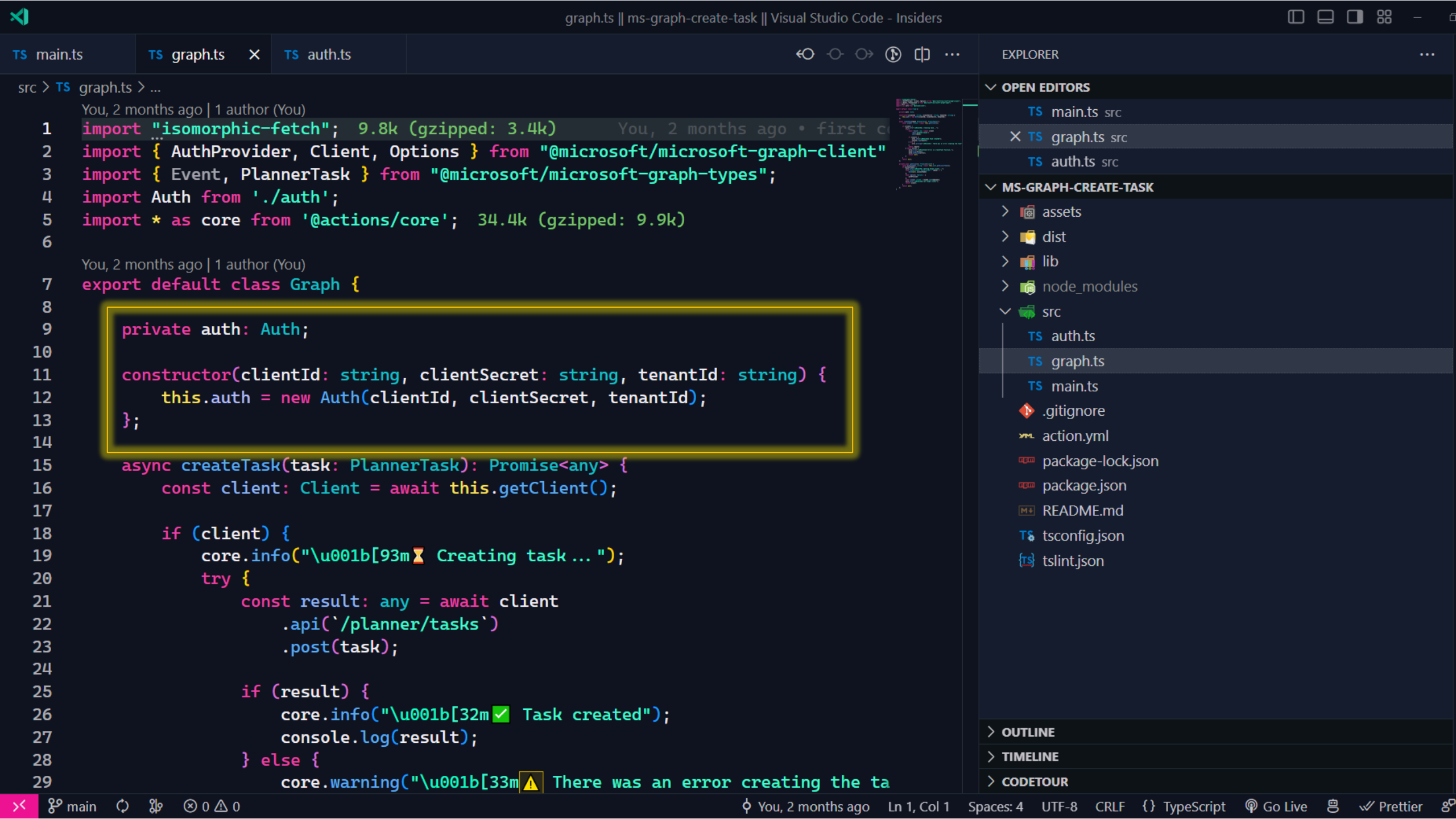
Demo

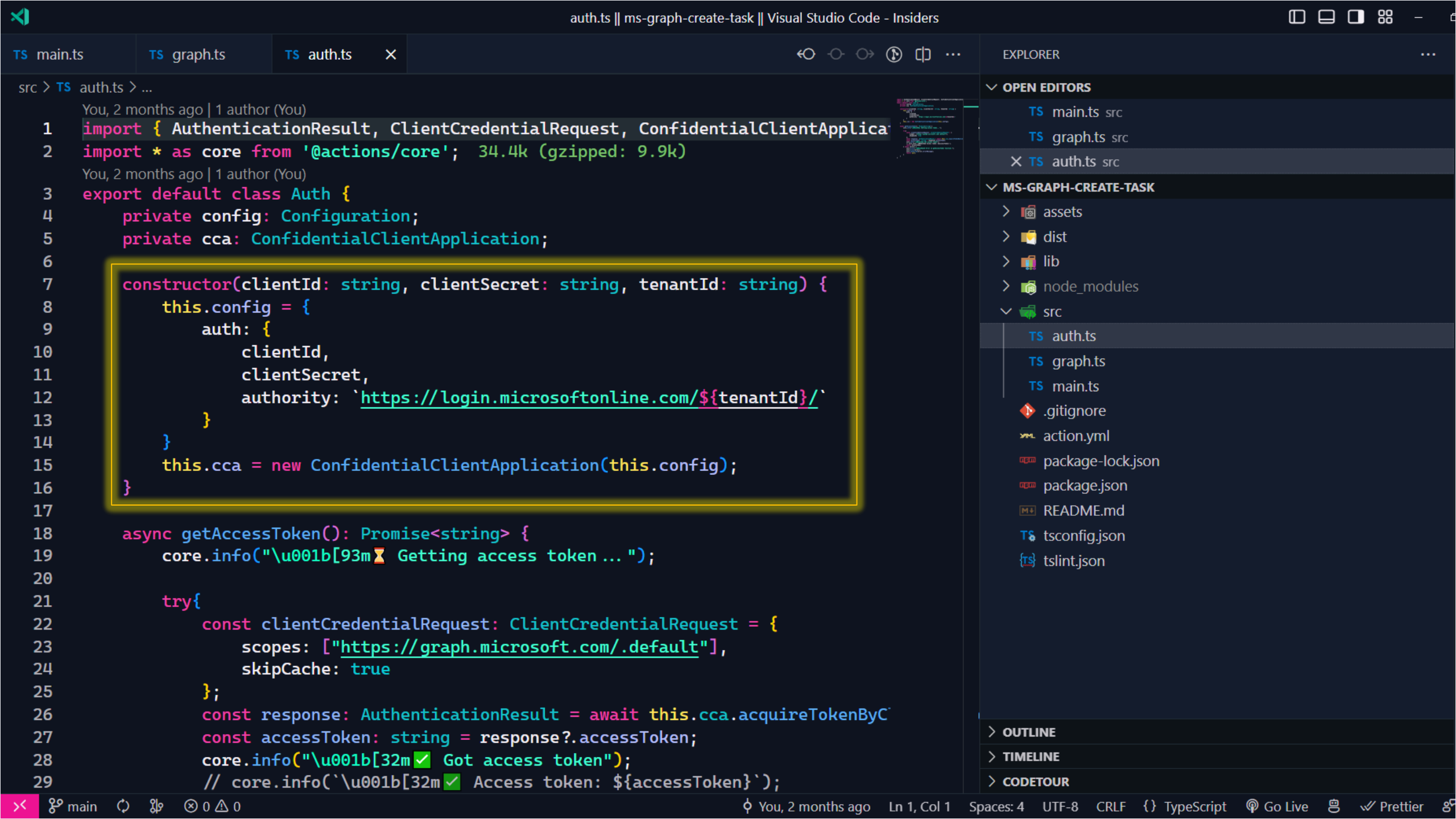

package.json || ms-graph-create-task || Visual Studio Code - Insiders
main.ts
graph.ts
auth.ts
package.json
package.json > scripts > package
"keywords": [
"msgraph",
"github",
"actions",
"task"
],
"author": "anoopt",
"license": "ISC",
"bugs": {
"url": "https://github.com/anoopt/ms-graph-create-task/issues"
},
"homepage": "https://github.com/anoopt/ms-graph-create-task#readme",
"dependencies": {
"@actions/core": "^1.10.0",
"@azure/msal-node": "^1.14.6",
"@microsoft/microsoft-graph-client": "^3.0.4",
"date-fns": "^2.29.3",
"isomorphic-fetch": "^3.0.0"
},
"devDependencies": {
"@microsoft/microsoft-graph-types": "^2.25.0",
"@types/node": "^18.11.18",
"@vercel/ncc": "^0.36.0"
}
}
EXPLORER
OPEN EDITORS
main.ts src
graph.ts src
auth.ts src
package.json
MS-GRAPH-CREATE-TASK
assets
dist
lib
node_modules
src
auth.ts
graph.ts
main.ts
.gitignore
action.yml
package-lock.json
package.json
README.md
tsconfig.json
tslint.json
OUTLINE
TIMELINE
CODETOUR
main
0 0
You, 2 months ago
Ln 8, Col 20 (3 selected)
Spaces: 2
UTF-8
LF
JSON
Go Live
Prettier

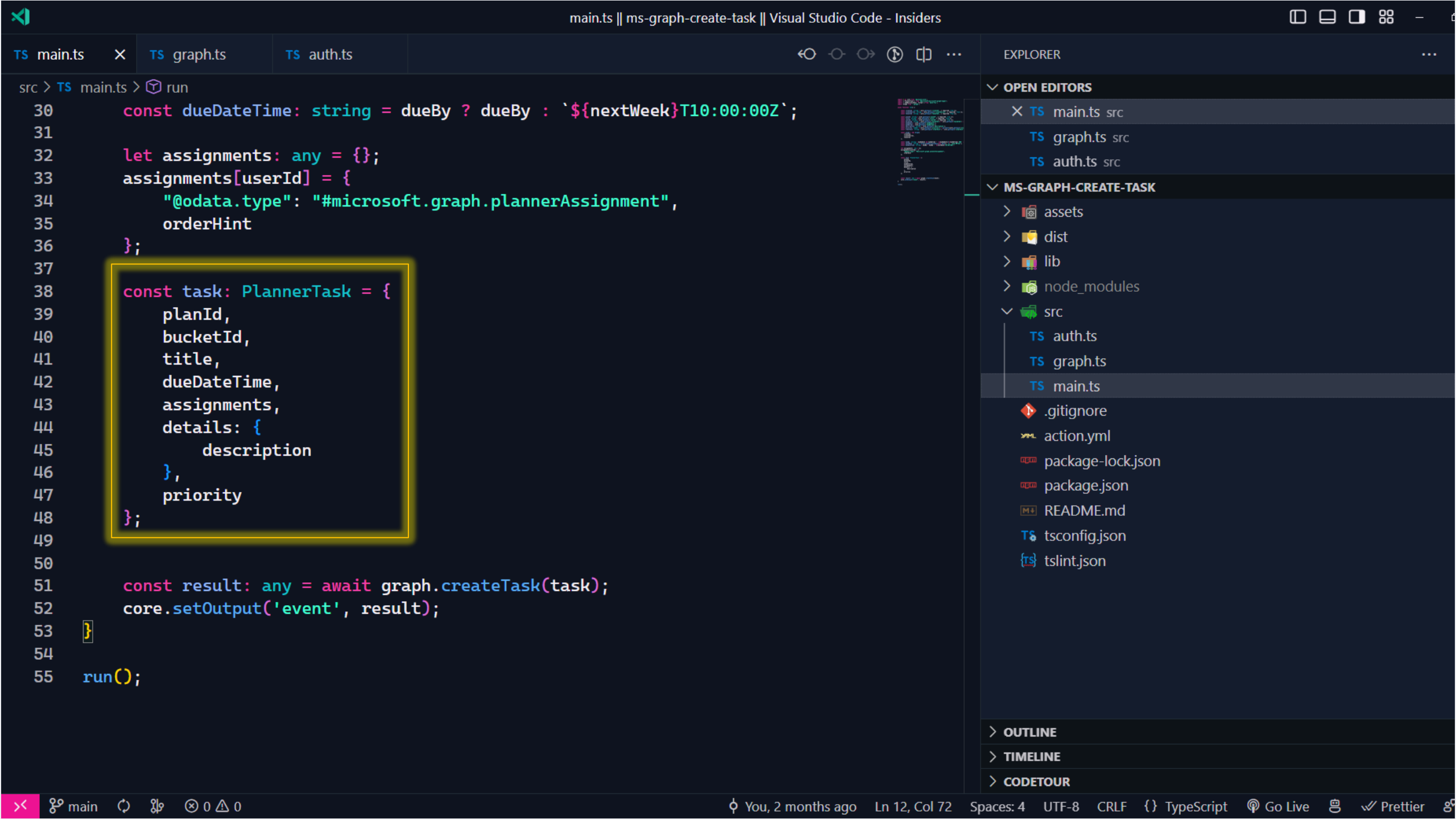

main.ts || ms-graph-create-task || Visual Studio Code - Insiders
main.ts
graph.ts
auth.ts
src > main.ts > ...
You, 2 months ago | 1 author (You)
import * as core from '@actions/core'; 34.4k (gzipped: 9.9k) You, 2 months
import { PlannerTask } from "@microsoft/microsoft-graph-types";
import { addBusinessDays, format } from 'date-fns'; 24.2k (gzipped: 6.5k)
import Graph from './graph';
async function run() {
const clientId: string = core.getInput('clientId', { required: true });
const clientSecret: string = core.getInput('clientSecret', { required: true }
const tenantId: string = core.getInput('tenantId', { required: true });
const planId: string = core.getInput('planId', { required: true });
const title: string = core.getInput('title', { required: true });
const userId: string = core.getInput('userId', { required: true });
const bucketId: string = core.getInput('bucketId') ? core.getInput('bucketId'
const dueByDate = core.getInput('dueByDate');
const dueByTime = core.getInput('dueByTime');
const description: string = core.getInput('description');
const priority: number = core.getInput('priority') ? parseInt(core.getInput('
const orderHint: string = core.getInput('orderHint') ? core.getInput('orderHi
const graph = new Graph(
clientId,
clientSecret,
tenantId
);
const dueBy: string = dueByDate && dueByTime ? `${dueByDate}T${dueByTime}:00Z
const nextWeek: string = format(addBusinessDays(new Date(), 7), 'yyyy-MM-dd')
const dueDateTime: string = dueBy ? dueBy : `${nextWeek}T10:00:00Z`;
EXPLORER
OPEN EDITORS
main.ts src
graph.ts src
auth.ts src
MS-GRAPH-CREATE-TASK
assets
dist
lib
node_modules
src
auth.ts
graph.ts
main.ts
.gitignore
action.yml
package-lock.json
package.json
README.md
tsconfig.json
tslint.json
OUTLINE
TIMELINE
CODETOUR
main
0 0
You, 2 months ago
Ln 1, Col 39
Spaces: 4
UTF-8
CRLF
TypeScript
Go Live
Prettier


main.ts || ms-graph-create-task || Visual Studio Code - Insiders
main.ts
graph.ts
auth.ts
src > main.ts > ...
You, 2 months ago | 1 author (You)
import * as core from '@actions/core'; 34.4k (gzipped: 9.9k)
import { PlannerTask } from "@microsoft/microsoft-graph-types";
import { addBusinessDays, format } from 'date-fns'; 24.2k (gzipped: 6.5k)
import Graph from './graph';
async function run() {
const clientId: string = core.getInput('clientId', { required: true });
const clientSecret: string = core.getInput('clientSecret', { required: true }
const tenantId: string = core.getInput('tenantId', { required: true });
const planId: string = core.getInput('planId', { required: true });
const title: string = core.getInput('title', { required: true });
const userId: string = core.getInput('userId', { required: true });
const bucketId: string = core.getInput('bucketId') ? core.getInput('bucketId'
const dueByDate = core.getInput('dueByDate');
const dueByTime = core.getInput('dueByTime');
const description: string = core.getInput('description');
const priority: number = core.getInput('priority') ? parseInt(core.getInput('
const orderHint: string = core.getInput('orderHint') ? core.getInput('orderHi
const graph = new Graph(
clientId,
clientSecret,
tenantId
);
const dueBy: string = dueByDate && dueByTime ? `${dueByDate}T${dueByTime}:00Z
const nextWeek: string = format(addBusinessDays(new Date(), 7), 'yyyy-MM-dd')
const dueDateTime: string = dueBy ? dueBy : `${nextWeek}T10:00:00Z`;
EXPLORER
OPEN EDITORS
main.ts src
graph.ts src
auth.ts src
MS-GRAPH-CREATE-TASK
assets
dist
lib
node_modules
src
auth.ts
graph.ts
main.ts
.gitignore
action.yml
package-lock.json
package.json
README.md
tsconfig.json
tslint.json
OUTLINE
TIMELINE
CODETOUR
main
0 0
You, 2 months ago
Ln 1, Col 39
Spaces: 4
UTF-8
CRLF
TypeScript
Go Live
Prettier


main.ts || ms-graph-create-task || Visual Studio Code - Insiders
main.ts
graph.ts
auth.ts
src > main.ts > ...
You, 2 months ago | 1 author (You)
import * as core from '@actions/core'; 34.4k (gzipped: 9.9k) You, 2 months
import { PlannerTask } from "@microsoft/microsoft-graph-types";
import { addBusinessDays, format } from 'date-fns'; 24.2k (gzipped: 6.5k)
import Graph from './graph';
async function run() {
const clientId: string = core.getInput('clientId', { required: true });
const clientSecret: string = core.getInput('clientSecret', { required: true }
const tenantId: string = core.getInput('tenantId', { required: true });
const planId: string = core.getInput('planId', { required: true });
const title: string = core.getInput('title', { required: true });
const userId: string = core.getInput('userId', { required: true });
const bucketId: string = core.getInput('bucketId') ? core.getInput('bucketId'
const dueByDate = core.getInput('dueByDate');
const dueByTime = core.getInput('dueByTime');
const description: string = core.getInput('description');
const priority: number = core.getInput('priority') ? parseInt(core.getInput('
const orderHint: string = core.getInput('orderHint') ? core.getInput('orderHi
const graph = new Graph(
clientId,
clientSecret,
tenantId
);
const dueBy: string = dueByDate && dueByTime ? `${dueByDate}T${dueByTime}:00Z
const nextWeek: string = format(addBusinessDays(new Date(), 7), 'yyyy-MM-dd')
const dueDateTime: string = dueBy ? dueBy : `${nextWeek}T10:00:00Z`;
EXPLORER
OPEN EDITORS
main.ts src
graph.ts src
auth.ts src
MS-GRAPH-CREATE-TASK
assets
dist
lib
node_modules
src
auth.ts
graph.ts
main.ts
.gitignore
action.yml
package-lock.json
package.json
README.md
tsconfig.json
tslint.json
OUTLINE
TIMELINE
CODETOUR
main
0 0
You, 2 months ago
Ln 1, Col 39
Spaces: 4
UTF-8
CRLF
TypeScript
Go Live
Prettier


graph.ts || ms-graph-create-task || Visual Studio Code - Insiders
main.ts
graph.ts
auth.ts
src > graph.ts > ...
You, 2 months ago | 1 author (You)
import "isomorphic-fetch"; 9.8k (gzipped: 3.4k) You, 2 months ago • first c
import { AuthProvider, Client, Options } from "@microsoft/microsoft-graph-client"
import { Event, PlannerTask } from "@microsoft/microsoft-graph-types";
import Auth from './auth';
import * as core from '@actions/core'; 34.4k (gzipped: 9.9k)
You, 2 months ago | 1 author (You)
export default class Graph {
private auth: Auth;
constructor(clientId: string, clientSecret: string, tenantId: string) {
this.auth = new Auth(clientId, clientSecret, tenantId);
};
async createTask(task: PlannerTask): Promise<any> {
const client: Client = await this.getClient();
if (client) {
core.info("\u001b[93m⌛ Creating task ... ");
try {
const result: any = await client
.api(`/planner/tasks`)
.post(task);
if (result) {
core.info("\u001b[32m✅ Task created");
console.log(result);
} else {
core.warning("\u001b[33m⚠ There was an error creating the ta
EXPLORER
OPEN EDITORS
main.ts src
graph.ts src
auth.ts src
MS-GRAPH-CREATE-TASK
assets
dist
lib
node_modules
src
auth.ts
graph.ts
main.ts
.gitignore
action.yml
package-lock.json
package.json
README.md
tsconfig.json
tslint.json
OUTLINE
TIMELINE
CODETOUR
main
0 0
You, 2 months ago
Ln 1, Col 1
Spaces: 4
UTF-8
CRLF
TypeScript
Go Live
Prettier

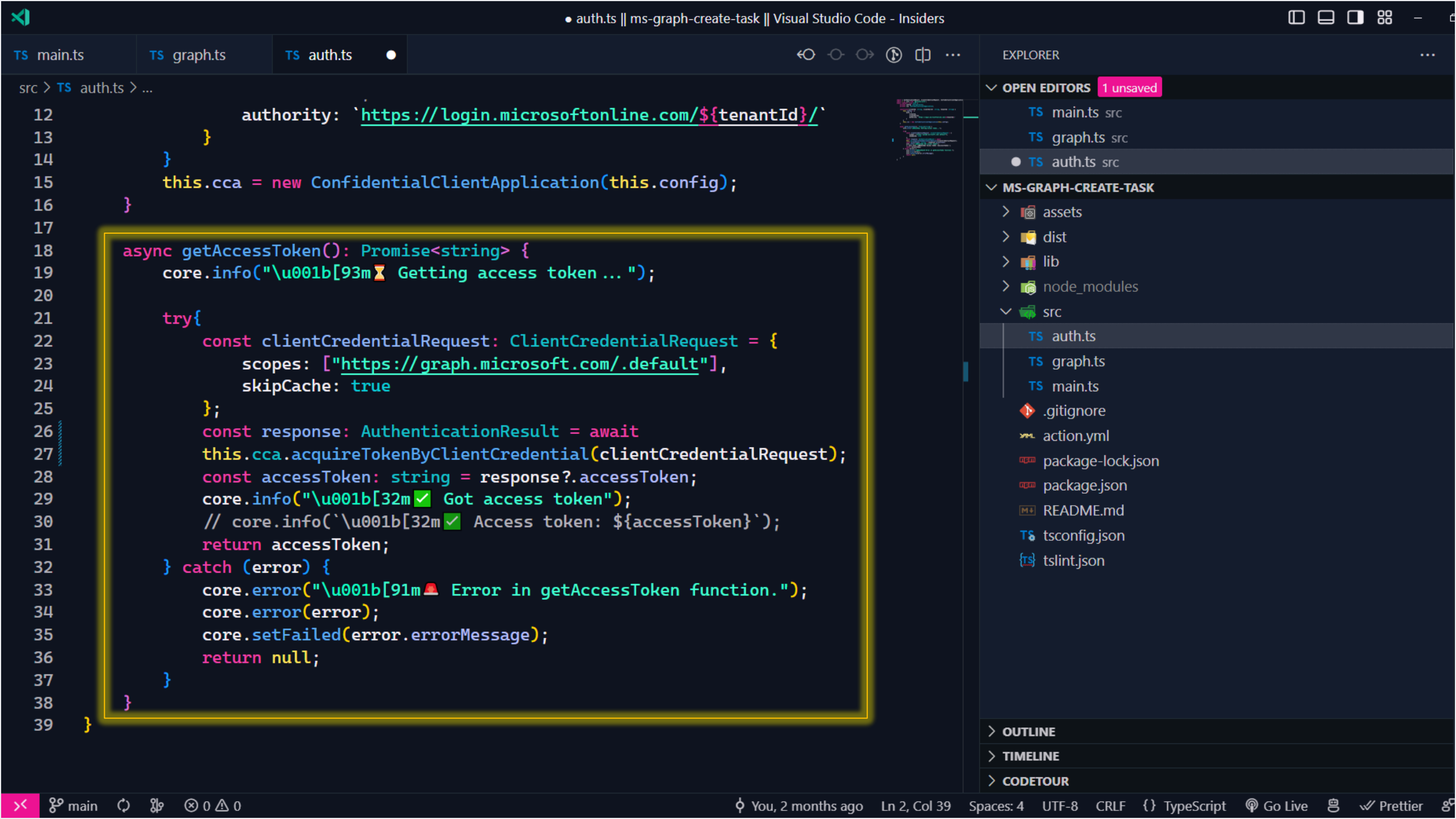

auth.ts || ms-graph-create-task || Visual Studio Code - Insiders
main.ts
graph.ts
auth.ts
src > auth.ts > ...
You, 2 months ago | 1 author (You)
import { AuthenticationResult, ClientCredentialRequest, ConfidentialClientApplica
import * as core from '@actions/core'; 34.4k (gzipped: 9.9k)
You, 2 months ago | 1 author (You)
export default class Auth {
private config: Configuration;
private cca: ConfidentialClientApplication;
constructor(clientId: string, clientSecret: string, tenantId: string) {
this.config = {
auth: {
clientId,
clientSecret,
authority: `https://login.microsoftonline.com/${tenantId}/`
}
}
this.cca = new ConfidentialClientApplication(this.config);
}
async getAccessToken(): Promise<string> {
core.info("\u001b[93m⌛ Getting access token...");
try{
const clientCredentialRequest: ClientCredentialRequest = {
scopes: ["https://graph.microsoft.com/.default"],
skipCache: true
};
const response: AuthenticationResult = await this.cca.acquireTokenByC
const accessToken: string = response?.accessToken;
core.info("\u001b[32m✅ Got access token");
// core.info(`\u001b[32m✅ Access token: ${accessToken}`);
EXPLORER
OPEN EDITORS
main.ts src
graph.ts src
auth.ts src
MS-GRAPH-CREATE-TASK
assets
dist
lib
node_modules
src
auth.ts
graph.ts
main.ts
.gitignore
action.yml
package-lock.json
package.json
README.md
tsconfig.json
tslint.json
OUTLINE
TIMELINE
CODETOUR
main
0 0
You, 2 months ago
Ln 1, Col 1
Spaces: 4
UTF-8
CRLF
TypeScript
Go Live
Prettier

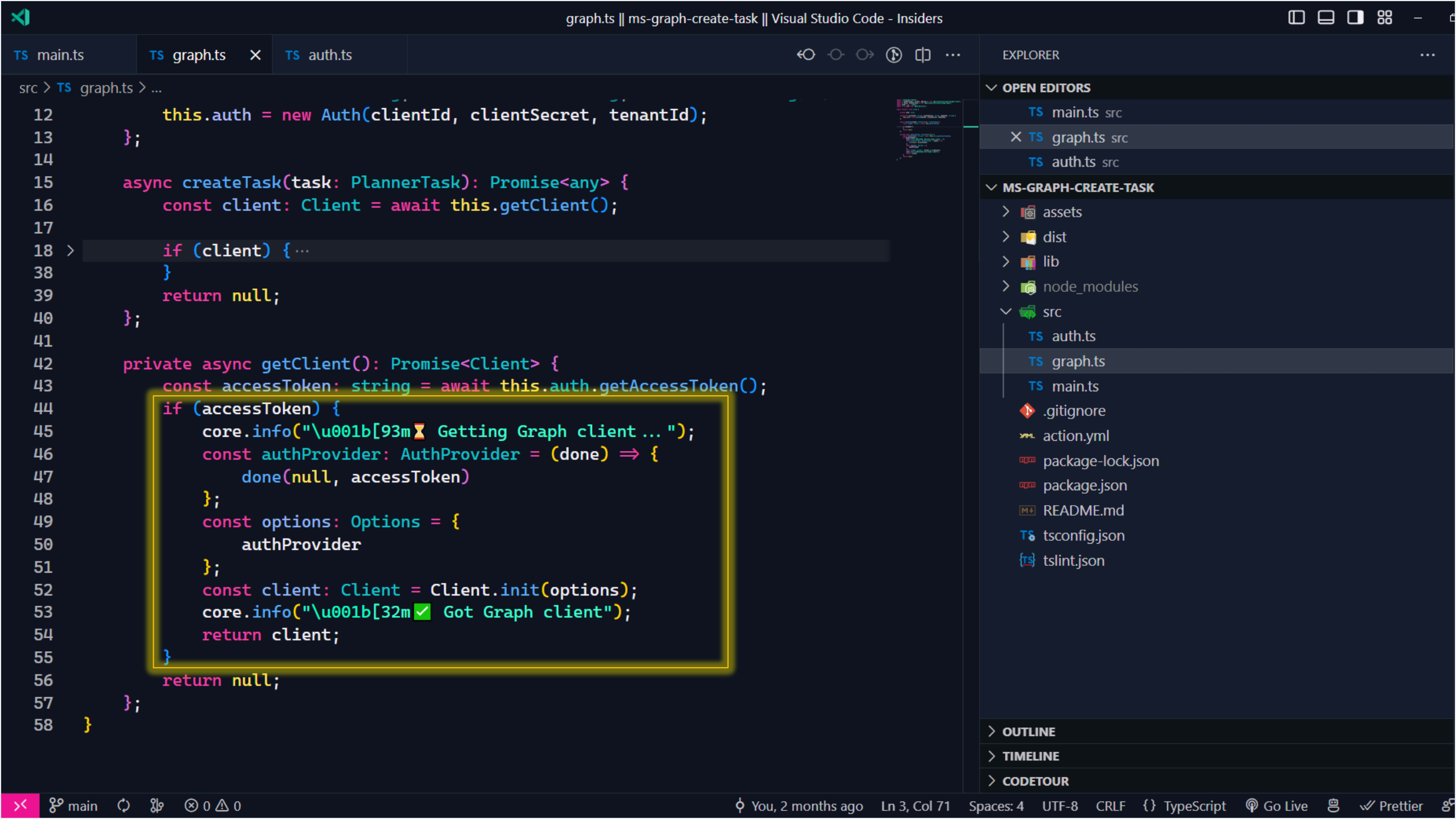

main.ts || ms-graph-create-task || Visual Studio Code - Insiders
main.ts
graph.ts
auth.ts
src > main.ts > run
30 const dueDateTime: string = dueBy ? dueBy : `${nextWeek}T10:00:00Z`;
31
32 let assignments: any = {};
33 assignments[userId] = {
34 "@odata.type": "#microsoft.graph.plannerAssignment",
35 orderHint
36 };
37
38 const task: PlannerTask = {
39 planId,
40 bucketId,
41 title,
42 dueDateTime,
43 assignments,
44 details: {
45 description
46 },
47 priority
48 };
49
50
51 const result: any = await graph.createTask(task);
52 core.setOutput('event', result);
53 }
54
55 run();
EXPLORER
OPEN EDITORS
main.ts src
graph.ts src
auth.ts src
MS-GRAPH-CREATE-TASK
assets
dist
lib
node_modules
src
auth.ts
graph.ts
main.ts
.gitignore
action.yml
package-lock.json
package.json
README.md
tsconfig.json
tslint.json
OUTLINE
TIMELINE
CODETOUR
main
0 0
You, 2 months ago
Ln 12, Col 72
Spaces: 4
UTF-8
CRLF
TypeScript
Go Live
Prettier

```
30      const dueDateTime: string = dueBy ? dueBy : `${nextWeek}T10:00:00Z`;
31
32      let assignments: any = {};
33      assignments[userId] = {
34          "@odata.type": "#microsoft.graph.plannerAssignment",
35          orderHint
36      };
37
38      const task: PlannerTask = {
39          planId,
40          bucketId,
41          title,
42          dueDateTime,
43          assignments,
44          details: {
45              description
46          },
47          priority
48      };
49
50
51      const result: any = await graph.createTask(task);
52      core.setOutput('event', result);
53  }
54
55  run();
```

```
12          this.auth = new Auth(clientId, clientSecret, tenantId);
13      };
14
15      async createTask(task: PlannerTask): Promise<any> {
16          const client: Client = await this.getClient();
17
18          if (client) {...
38          }
39          return null;
40      };
41
42      private async getClient(): Promise<Client> {
43          const accessToken: string = await this.auth.getAccessToken();
44          if (accessToken) {
45              core.info("\u001b[93m⌛ Getting Graph client...");
46              const authProvider: AuthProvider = (done) => {
47                  done(null, accessToken)
48              };
49              const options: Options = {
50                  authProvider
51              };
52              const client: Client = Client.init(options);
53              core.info("\u001b[32m✅ Got Graph client");
54              return client;
55          }
56          return null;
57      };
58  }
```

```
12          this.auth = new Auth(clientId, clientSecret, tenantId);
13      };
14
15      async createTask(task: PlannerTask): Promise<any> {
16          const client: Client = await this.getClient();
17
18          if (client) {···
38          }
39          return null;
40      };
41
42      private async getClient(): Promise<Client> {
43          const accessToken: string = await this.auth.getAccessToken();
44          if (accessToken) {
45              core.info("\u001b[93m⌛ Getting Graph client...");
46              const authProvider: AuthProvider = (done) => {
47                  done(null, accessToken)
48              };
49              const options: Options = {
50                  authProvider
51              };
52              const client: Client = Client.init(options);
53              core.info("\u001b[32m✅ Got Graph client");
54              return client;
55          }
56          return null;
57      };
58  }
```


● auth.ts || ms-graph-create-task || Visual Studio Code - Insiders
main.ts
graph.ts
auth.ts
src > auth.ts > ...
authority: `https://login.microsoftonline.com/${tenantId}/`
}
}
this.cca = new ConfidentialClientApplication(this.config);
}
async getAccessToken(): Promise<string> {
core.info("\u001b[93m⏳ Getting access token...");
try{
const clientCredentialRequest: ClientCredentialRequest = {
scopes: ["https://graph.microsoft.com/.default"],
skipCache: true
};
const response: AuthenticationResult = await
this.cca.acquireTokenByClientCredential(clientCredentialRequest);
const accessToken: string = response?.accessToken;
core.info("\u001b[32m✅ Got access token");
// core.info(`\u001b[32m✅ Access token: ${accessToken}`);
return accessToken;
} catch (error) {
core.error("\u001b[91m🚨 Error in getAccessToken function.");
core.error(error);
core.setFailed(error.errorMessage);
return null;
}
}
}
EXPLORER
OPEN EDITORS 1 unsaved
main.ts src
graph.ts src
auth.ts src
MS-GRAPH-CREATE-TASK
assets
dist
lib
node_modules
src
auth.ts
graph.ts
main.ts
.gitignore
action.yml
package-lock.json
package.json
README.md
tsconfig.json
tslint.json
OUTLINE
TIMELINE
CODETOUR
main
0 0
You, 2 months ago
Ln 2, Col 39
Spaces: 4
UTF-8
CRLF
TypeScript
Go Live
Prettier


graph.ts || ms-graph-create-task || Visual Studio Code - Insiders
main.ts
graph.ts
auth.ts
src > graph.ts > ...
this.auth = new Auth(clientId, clientSecret, tenantId);
};
async createTask(task: PlannerTask): Promise<any> {
const client: Client = await this.getClient();
if (client) {...
}
return null;
};
private async getClient(): Promise<Client> {
const accessToken: string = await this.auth.getAccessToken();
if (accessToken) {
core.info("\u001b[93m⌛ Getting Graph client ...");
const authProvider: AuthProvider = (done) => {
done(null, accessToken)
};
const options: Options = {
authProvider
};
const client: Client = Client.init(options);
core.info("\u001b[32m✅ Got Graph client");
return client;
}
return null;
};
}
EXPLORER
OPEN EDITORS
main.ts src
graph.ts src
auth.ts src
MS-GRAPH-CREATE-TASK
assets
dist
lib
node_modules
src
auth.ts
graph.ts
main.ts
.gitignore
action.yml
package-lock.json
package.json
README.md
tsconfig.json
tslint.json
OUTLINE
TIMELINE
CODETOUR
main
0 0
You, 2 months ago
Ln 3, Col 71
Spaces: 4
UTF-8
CRLF
TypeScript
Go Live
Prettier

```
14
15      async createTask(task: PlannerTask): Promise<any> {
16          const client: Client = await this.getClient();
17
18          if (client) {
19              core.info("\u001b[93m⏳ Creating task ... ");
20              try {
21                  const result: any = await client
22                      .api(`/planner/tasks`)
23                      .post(task);
24
25                  if (result) {
26                      core.info("\u001b[32m✅ Task created");
27                      console.log(result);
28                  } else {
29                      core.warning("\u001b[33m⚠️ There was an error creating the ta
30                  }
31                  return result;
32              } catch (error) {
33                  core.error("\u001b[91m🚨 Error in createTask function.");
34                  core.error(error);
35                  core.setFailed(error);
36                  return null;
37              }
38          }
39          return null;
40      };
41
42      private async getClient(): Promise<Client> {
43          const accessToken: string = await this.auth.getAccessToken();
44          if (accessToken) {
```

Summary

- Microsoft Graph Planner API now has app only permissions
  - Delegated permissions already existed
- Azure AD app reg
  - Tasks.Read.All
  - Tasks.ReadWrite.All
- GitHub scenario
  - A GitHub action to create task
  - Runs on pull request

Resources

- Docs - https://devblogs.microsoft.com/microsoft365dev/new-microsoft-planner-api-capabilities-now-available-in-microsoft-graph/
- Workflow - https://gist.github.com/anoopt/21a5ed9a251d7ae8eb991a310380a194
- Action - https://github.com/marketplace/actions/create-a-task-with-microsoft-graph-using-msal