

Claude Code v2.0.76

Recent activity

Welcome back Carlo!

No recent activity

What's new

Added LSP (Language Server Protocol) tool for code intelligence features like ...
Added `/terminal-setup` support for Kitty, Alacritty, Zed, and Warp terminals
Added ctrl+t shortcut in `/theme` to toggle syntax highlighting on/off

Opus 4.5 · Claude Max · carlo@thethirdway.com.au's | [/release-notes for more](#)

Organization

/Users/carlobertozzi

> I want to start a new project using this builder.io ste
286fbab6792c4f4f86cd9ec36393ea60 (public api key)

💡 I'd be happy to help you set up a new project with Builder.io using your public API key.

💡 User answered Claude's questions:

└ · What framework would you like to use for your Builder.io project? → NextJS and React JS

· What type of content will you be building? → Marketing pages

💡 Next.js with Builder.io for marketing pages - great choice!

💡 User answered Claude's questions:

└ · What would you like to name your project? → Composable Lead Generation, CLG for short

- I'll create a Next.js project called "clg" (Composable Lead Generation) with Builder.io integration.
- **Bash**(npx create-next-app@latest clg --typescript --tailwind --eslint --app --src-dir --import-alias "@/*" --use-npm)
 - └ Tool use rejected with user message: Yes and we need to use Vercel for hosting, and we also need a git hub project, , although both will need to be transferred to corporate equivalents at some point
- Got it - I'll set up the project with Vercel deployment and GitHub repository.
- **Bash**(npx create-next-app@latest clg --typescript --tailwind --eslint --app --src-dir --import-alias "@/*" --use-npm) timeout: 3m 0s
 - └ ? Would you like to use React Compiler? › No / Yes
 - npm warn exec The following package was not found and will be installed:
create-next-app@16.1.1
 - npm notice
 - ... +4 lines (ctrl+o to expand)
- **Bash**(rm -rf clg 2>/dev/null; npx create-next-app@latest clg --typescript --tailwind --eslint --app --src-dir timeout: 3m --import-alias "@/*" --use-npm --turbopack --no-experiment...) 0s
 - └ ? Would you like to use React Compiler? › No / Yes
 - **Bash**(rm -rf clg 2>/dev/null; echo "n" | npx create-next-app@latest clg --typescript --tailwind --eslint --app --src-dir timeout: 3m --import-alias "@/*" --use-npm --turbopack 2>...)
- ? Would you like to use React Compiler? › No / Yes ✓ Would you like to use React Compiler? ... No / Yes
 - Creating a new Next.js app in /Users/carlobertozzi/clg.
 - ... +34 lines (ctrl+o to expand)
- Next.js project created. Now installing Builder.io SDK.
- **Bash**(cd /Users/carlobertozzi/clg && npm install @builder.io/sdk-react-nextjs)
 - └ added 35 packages, and audited 393 packages in 13s
 - 150 packages are looking for funding
 - ... +3 lines (ctrl+o to expand)
- **Bash**(ls -la /Users/carlobertozzi/clg/src/app/)
 - └ total 80
 - drwxr-xr-x 6 carlobertozzi staff 192 8 Jan 12:04 .
 - drwxr-xr-x 3 carlobertozzi staff 96 8 Jan 12:04 ..
 - ... +4 lines (ctrl+o to expand)
- **Write**(.env.local)
 - └ Wrote 2 lines to .env.local
 - 1 =
- **Write**(src/builder.ts)
 - └ Wrote 7 lines to src/builder.ts
 - 1 import from "@builder.io/sdk-react-nextjs"

```

2 // Initialize Builder.io with your public API key
3     init
4 export
● Bash(mkdir -p "/Users/carlobertozzi/clg/src/app/[...page]")
  ↳ (No content)
● Read(src/app/page.tsx)
  ↳ Read 66 lines
  ↳ Interrupted · What should Claude do instead?
> Before you actually create anything in Builder I want to approve and we havent
decided on designs, so I just want all the
architecture set up, the platforms linked and a test to prove that all the MCPs are
connected
● Understood - architecture and platform linking only, with a connection test. No
content creation in Builder.io.
● Bash(rm -rf "/Users/carlobertozzi/clg/src/app/[...page]")
  ↳ (No content)
● Write(src/app/page.tsx)
  ↳ Added 82 lines, removed 58 lines
    1 -import Image from "next/image";
    1 +import { builder } from "@/builder";
    2
    3 -export default function Home() {
    3 +async function testBuilderConnection() {
    4 +  try {
    5 +    // Test connection by fetching models list
    6 +    const response = await fetch(
    7 +
`https://cdn.builder.io/api/v3/content/page?apiKey=${process.env.NEXT_PUBLIC_B
UILDER_API_KEY}&limit=1`
    8 +  );
    9 +  return {
    10 +    connected: response.ok,
    11 +    status: response.status,
    12 +    apiKey: process.env.NEXT_PUBLIC_BUILDER_API_KEY?.slice(0, 8) +
...
    13 +  };
    14 + } catch (error) {
    15 +  return {
    16 +    connected: false,
    17 +    status: 0,
    18 +    error: error instanceof Error ? error.message : "Unknown error",
    19 +  };
    20 + }
    21 +}

```

```
22 +
23 +export default async function Home() {
24 +   const connectionTest = await testBuilderConnection();
25 +
26   return
5 -   <div className="flex min-h-screen items-center justify-center bg-zinc-50
font-sans dark:bg-black">
6 -     <main className="flex min-h-screen w-full max-w-3xl flex-col
items-center justify-between py-32 px-16 bg-white da
-rk:bg-black sm:items-start">
7 -       <Image
8 -         className="dark:invert"
9 -         src="/next.svg"
10 -        alt="Next.js logo"
11 -        width={100}
12 -        height={20}
13 -        priority
14 -      />
15 -       <div className="flex flex-col items-center gap-6 text-center
sm:items-start sm:text-left">
16 -         <h1 className="max-w-xs text-3xl font-semibold leading-10
tracking-tight text-black dark:text-zinc-50">
17 -           To get started, edit the page.tsx file.
18 -         </h1>
19 -         <p className="max-w-md text-lg leading-8 text-zinc-600
dark:text-zinc-400">
20 -           Looking for a starting point or more instructions? Head over to{" "}
21 -           <a
22 -

-e-tw&utm\_campaign=create-next-app"
23 -             className="font-medium text-zinc-950 dark:text-zinc-50"
24 -           >
25 -             Templates
26 -             </a>{" "}
27 -             or the{" "}
28 -             <a
29 -

```

```
32 -      Learning
33 -      </a>{" "}
34 -      center.
35 -      </p>
27 +      <div className="min-h-screen bg-zinc-900 text-white p-8">
28 +          <div className="max-w-2xl mx-auto">
29 +              <h1 className="text-3xl font-bold mb-8">
30 +                  CLG - Composable Lead Generation
31 +              </h1>
32 +
33 +          <div className="bg-zinc-800 rounded-lg p-6 mb-6">
34 +              <h2 className="text-xl font-semibold mb-4">Platform Connection
Status</h2>
35 +
36 +          <div className="space-y-4">
37 +              <div className="flex items-center justify-between p-3 bg-zinc-700
rounded">
38 +                  <span>Builder.io API</span>
39 +                  <span
40 +                      className={`${px-3 py-1 rounded text-sm font-medium ${{
41 +                          connectionTest.connected
42 +                          ? "bg-green-600 text-white"
43 +                          : "bg-red-600 text-white"
44 +                      }}`}>
45 +                  >
46 +                      {connectionTest.connected ? "Connected" : "Failed"}
47 +                  </span>
48 +              </div>
49 +
50 +          <div className="flex items-center justify-between p-3 bg-zinc-700
rounded">
51 +              <span>API Key</span>
52 +              <code className="text-zinc-400 text-sm">
53 +                  {connectionTest.apiKey}
54 +              </code>
55 +          </div>
56 +
57 +          <div className="flex items-center justify-between p-3 bg-zinc-700
rounded">
58 +              <span>Next.js</span>
59 +              <span className="px-3 py-1 rounded text-sm font-medium
bg-green-600 text-white">
60 +                  Running
61 +              </span>
```

```
62 +      </div>
63 +
64 +      <div className="flex items-center justify-between p-3 bg-zinc-700
rounded">
65 +          <span>Vercel</span>
66 +          <span className="px-3 py-1 rounded text-sm font-medium
bg-yellow-600 text-white">
67 +              Pending Deploy
68 +          </span>
69 +      </div>
70 +      </div>
71      div
37 -      <div className="flex flex-col gap-4 text-base font-medium
sm:flex-row">
38 -          <a
39 -              className="flex h-12 w-full items-center justify-center gap-2
rounded-full bg-foreground px-5 text-backgrou
-nd transition-colors hover:bg-[#383838] dark:hover:bg-[#ccc] md:w-[158px]"
40 -
href="https://vercel.com/new?utm_source=create-next-app&utm_medium=appdir-te
mplate-tw&utm_campaign=create-n
-ext-app"
41 -          target="_blank"
42 -          rel="noopener noreferrer"
43 -      >
44 -          <Image
45 -              className="dark:invert"
46 -              src="/vercel.svg"
47 -              alt="Vercel logomark"
48 -              width={16}
49 -              height={16}
50 -          />
51 -          Deploy Now
52 -      </a>
53 -      <a
54 -          className="flex h-12 w-full items-center justify-center rounded-full
border border-solid border-black/[.08]
- px-5 transition-colors hover:border-transparent hover:bg-black/[.04]
dark:border-white/[.145] dark:hover:bg-[#1a1a1a]
-md:w-[158px]"
55 -
href="https://nextjs.org/docs?utm_source=create-next-app&utm_medium=appdir-tem
mplate-tw&utm_campaign=create-
-next-app"
```

```
56 -      target="_blank"
57 -      rel="noopener noreferrer"
58 -    >
59 -    Documentation
60 -  </a>
72 +
73 +  <div className="bg-zinc-800 rounded-lg p-6">
74 +    <h2 className="text-xl font-semibold mb-4">Architecture
```

Ready</h2>

```
75 +  <ul className="space-y-2 text-zinc-300">
76 +    <li>Next.js 15 with App Router</li>
77 +    <li>TypeScript + Tailwind CSS</li>
78 +    <li>Builder.io SDK installed</li>
79 +    <li>Environment configured</li>
80 +  </ul>
81  div
62 -  </main>
82 +
83 +  <p className="mt-8 text-zinc-500 text-sm">
84 +    Awaiting design approval before creating Builder.io content.
85 +  </p>
86 +  </div>
87  div
88
89
```

● Bash(npm run build 2>&1 | head -50)

```
└ > clg@0.1.0 build
> next build
... +23 lines (ctrl+o to expand)
```

● Bash(npm list @builder.io/sdk-react-nextjs)

```
└ clg@0.1.0 /Users/carlobertozzi/clg
  └── @builder.io/sdk-react-nextjs@0.24.0
```

● Bash(cat

```
/Users/carlobertozzi/clg/node_modules/@builder.io/sdk-react-nextjs/package.json |
head -50)
```

```
└ {
  "name": "@builder.io/sdk-react-nextjs",
  "description": "Builder.io RSC SDK for NextJS App Directory",
  ... +47 lines (ctrl+o to expand)
```

● Bash(cat

```
/Users/carlobertozzi/clg/node_modules/@builder.io/sdk-react-nextjs/types/index.d.ts
| head -100)
```

```
└ cat:  
/Users/carlobertozzi/clg/node_modules/@builder.io/sdk-react-nextjs/types/index.d.ts:  
No such file or directory  
● Bash(rm /Users/carlobertozzi/clg/src/builder.ts)  
└ (No content)  
● Update(src/app/page.tsx)  
└ Removed 2 lines  
  1 -import { builder } from "@builder";  
  2 -  
    1 async function testBuilderConnection  
    2 try  
    3 // Test connection by fetching models list  
● Bash(npm run build 2>&1 | head -50)  
└ > clg@0.1.0 build  
  > next build  
  ... +20 lines (ctrl+o to expand)  
● Build passes. Now setting up GitHub and Vercel.  
● Bash/git status  
└ On branch main  
Changes not staged for commit:  
  (use "git add <file>..." to update what will be committed)  
  ... +6 lines (ctrl+o to expand)  
● Bash/git add -A &&  
GIT_COMMITTER_EMAIL="cbertozz-access@users.noreply.github.com"  
GIT_COMMITTER_NAME="Carlo Bertozzi" git -c  
  user.email="cbertozz-access@users.norepl...")  
└ [main 2d698d7] Add Builder.io SDK and connection test page  
  3 files changed, 513 insertions(+), 60 deletions(-)  
● Bash(gh repo create clg --public --source=. --remote=origin --push --description  
"Composable Lead Generation -           timeout: 1m  
  Builder.io powered marketing site")  
0s  
└ https://github.com/cbertozz-access/clg  
branch 'main' set up to track 'origin/main'.  
To https://github.com/cbertozz-access/clg.git  
 * [new branch]    HEAD -> main  
● Bash(which vercel || npm list -g vercel 2>/dev/null)  
└ Error: Exit code 1  
vercel not found  
/opt/homebrew/lib  
└ (empty)  
● Bash(npm install -D vercel) timeout: 1m 0s  
└ added 229 packages, and audited 622 packages in 15s  
  171 packages are looking for funding
```

... +10 lines (ctrl+o to expand)

● Bash(npx vercel link --yes 2>&1) timeout: 1m 0s

└ Error: Exit code 1

Vercel CLI 50.1.6

> NOTE: The Vercel CLI now collects telemetry regarding usage of the CLI.

> This information is used to shape the CLI roadmap and prioritize features.

> You can learn more, including how to opt-out if you'd not like to participate in this program, by visiting the following URL:

> <https://vercel.com/docs/cli/about-telemetry>

Error: No existing credentials found. Please run `vercel login` or pass "--token"

Learn More: <https://err.sh/vercel/no-credentials-found>

● Bash(npx vercel login 2>&1)

└ Interrupted · What should Claude do instead?

> Do I need to auth with vercel?

● Yes, you need to authenticate with Vercel. You have two options:

1. **Browser auth** - Visit this URL and enter the code:

- URL: https://vercel.com/oauth/device?user_code=SZZX-HVRK

- This will link your Vercel account to the CLI

2. **Token auth** - If you have a Vercel token, I can use `--token YOUR_TOKEN` instead

Once authenticated, I can deploy and set up the

`NEXT_PUBLIC_BUILDER_API_KEY` environment variable on Vercel.

Which approach would you prefer?

> OK done

● Bash(npx vercel link --yes 2>&1) timeout: 1m 0s

└ Error: Exit code 1

Vercel CLI 50.1.6

Error: No existing credentials found. Please run `vercel login` or pass "--token"

Learn More: <https://err.sh/vercel/no-credentials-found>

● The auth may not have completed. Let me run login again:

● Bash(npx vercel login 2>&1)

└ Interrupted · What should Claude do instead?

> This was successful https://vercel.com/oauth/device?user_code=SZZX-HVRK

● Bash(ls -la ~/.vercel 2>/dev/null || echo "No .vercel directory found")

└ No .vercel directory found

● The browser auth completed but the CLI was still waiting. Let me try again - this time please authenticate with the new code:

● Bash(echo "" | npx vercel login 2>&1)

└ Vercel CLI 50.1.6

Visit https://vercel.com/oauth/device?user_code=PVFM-WNQR

... +10 lines (ctrl+o to expand)

● Authenticated. Now linking and deploying:

● Bash(npx vercel link --yes 2>&1) timeout: 1m 0s

└ Vercel CLI 50.1.6

Loading scopes...

Searching for existing projects...

... +8 lines (ctrl+o to expand)

● Linked and connected to GitHub. Now adding the Builder.io API key and deploying:

● Bash(npx vercel env add NEXT_PUBLIC_BUILDER_API_KEY production <<< "286fbab6792c4f4f86cd9ec36393ea60" 2>&1) timeout: 30s

 └ Vercel CLI 50.1.6

 Retrieving project...

 Saving

 ... +4 lines (ctrl+o to expand)

● Bash(npx vercel --prod 2>&1) timeout: 3m 0s

 └ Vercel CLI 50.1.6

 Retrieving project...

 Deploying carlo-bertozzis-projects/clg

 ... +60 lines (ctrl+o to expand)

● Deployed. Let me verify the connection test page is working:

● Fetch(<https://clg-ten.vercel.app>)

 └ Received **9.4KB** (200 OK)

● Bash(git add -A &&

GIT_COMMITTER_EMAIL="cbertozz-access@users.noreply.github.com"

GIT_COMMITTER_NAME="Carlo Bertozzi" git -c

 user.email="cbertozz-access@users.norepl...")

 └ [main cac20c3] Add Vercel CLI for deployment

 2 files changed, 8612 insertions(+), 4151 deletions(-)

 To <https://github.com/cbertozz-access/clg.git>

 2d698d7..cac20c3 main -> main

● All platforms connected and verified.

Summary

Platform	Status	URL
Builder.io	Connected	API Key: 286fbab6...
GitHub	Connected	https://github.com/cbertozz-access/clg
Vercel	Deployed	https://clg-ten.vercel.app

The test page confirms Builder.io API is responding. Auto-deploy from GitHub is enabled - any push to [main](#) will trigger a new Vercel deployment.

Architecture ready:

- Next.js 16 with App Router
- TypeScript + Tailwind CSS
- Builder.io SDK installed
- Environment variables configured on Vercel

Ready for designs when you are.

> What does Pending Deploy mean next to VErceL <https://clg-ten.vercel.app/>

That's a hardcoded label I wrote before deployment - it's now outdated. Let me fix it:

● **Read**(src/app/page.tsx)

└ Read **88** lines

● **Update**(src/app/page.tsx)

└ Added **2** lines, removed **2** lines

61

62

63 span span

64 -

65 - Pending Deploy

64 +

65 + Deployed

66 span

67 div

68 </ >

● **Bash**(git add -A &&

GIT_COMMITTER_EMAIL="cbertozz-access@users.noreply.github.com"

GIT_COMMITTER_NAME="Carlo Bertozzi" git -c

user.email="cbertozz-access@users.norepl...")

└ [main 7512a4c] Fix Vercel status to show Deployed

1 file changed, 2 insertions(+), 2 deletions(-)

To https://github.com/cbertozz-access/clg.git

cac20c3..7512a4c main -> main

● Fixed. It was just a static label I wrote before deploying - now updated to "Deployed" with a green badge.

Vercel will auto-deploy from that push in ~30 seconds.

> OK I am trying to convince a sceptical CMO that we can deliver robust solutions using AI, this is the start. I want to document everything we do in confluence and in Jira. I know Jira has an API, when I give you tasks here, could you flesh them out in proper tech language and then log them as Jira tickets that have all the detail, assign them to yourself, and then mark as completed when done?

● Great idea - having a documented audit trail in Jira will demonstrate the AI's capability to work within enterprise workflows.

To set this up, I'll need a few things:

● User declined to answer questions

> This is the <https://accessgroupinternational.atlassian.net/wiki/rest/api/content>

● That's the Confluence API. For Jira, the base would be:

<https://accessgroupinternational.atlassian.net/rest/api/3/>

To integrate both, I'll need:

1. **API Token** - Create one at

<https://id.atlassian.com/manage-profile/security/api-tokens>

2. **Your Atlassian email** - Used with the token for authentication

3. **Jira Project Key** - Which project to log tickets to (e.g., "CLG")

4. **Confluence Space Key** - Which space for documentation

For assigning tickets to "myself" (the AI), we'd need a service account or I can assign them to you and update status as I complete work.

Can you provide the API token and confirm the project/space details?

```
>  
ATATT3xFfGF07LnsZ_LMzAJYUiaePFtGCUxKkhyjm54b01LtgleRGMr3K6V6kRZQf  
NZRKZJbCcnAZwqQTgm1TJXuwZKs7DS9pXAIDGGm5IWYB_-2tJVmlxut6D41R-p  
Dik  
XoNEnU9MwAZwy6klqjLTketQOd3ibJRaxpTIQJmkJHs1v4F2oKYj8=8B780A31
```

Got the API token. I still need:

1. **Your Atlassian email** (for authentication)

2. **Jira Project Key** - Should I create a new project called "CLG" or use an existing one?

3. **Confluence Space Key** - Where to document (e.g., "CLG" or existing space)

```
> carlo.bertozzi@accessgroup.net.au
```

Let me test the Jira connection and list available projects:

Bash(curl -s -u

```
"carlo.bertozzi@accessgroup.net.au:ATATT3xFfGF07LnsZ_LMzAJYUiaePFtGCUxK  
khyjm54b01LtgleRGMr3K6V6kRZQfNZRKZJbCcnAZwqQTgm
```

1TJXuwZKs7DS9pXAIDGGm5IWYB_-2tJ...)

 └ curl: option : blank argument where content is expected

 curl: try 'curl --help' or 'curl --manual' for more information

Bash(curl -s -X GET \

 -H "Authorization: Basic \$(echo -n

```
'carlo.bertozzi@accessgroup.net.au:ATATT3xFfGF07LnsZ_LMzAJYUiaePFtGCUxK  
khyjm54b01LtgleRGMr3K6V6kRZQfNZRKZJ...)
```

 └

Connected - credentials work but no existing projects visible. Let me check

Confluence spaces:

Bash(curl -s -X GET \

 -H "Authorization: Basic \$(echo -n

```
'carlo.bertozzi@accessgroup.net.au:ATATT3xFfGF07LnsZ_LMzAJYUiaePFtGCUxK  
khyjm54b01LtgleRGMr3K6V6kRZQfNZRKZJ...)
```

 └ {

 "statusCode": 403,

```
"message":  
"com.atlassian.confluence.mvc.rest.common.exception.StacklessResponseStatusEx  
ception: 403 FORBIDDEN \"Request reje  
... +2 lines (ctrl+o to expand)
```

 The API token can access Jira but not Confluence (403 Forbidden). You may need to:

1. Grant the token Confluence access in your Atlassian settings, OR
2. Create a new token with both Jira + Confluence scopes

For now, let me try creating a Jira project for CLG:

 **Bash**(curl -s -X GET \
-H "Authorization: Basic \$(echo -n

```
'carlo.bertozzi@accessgroup.net.au:ATATT3xFfGF07LnsZ_LMzAJYUiaePFtGCUxKk  
hyjm54b01LtgleRGMr3K6V6kRZQfNZRKZJ...)  
└ {  
  "permissions": {  
    "ADMINISTER_PROJECTS": {  
      ... +19 lines (ctrl+o to expand)
```

> We need to create a new project space - Composable Lead Generation Project

 The API token doesn't have permission to create projects (this is a Jira admin setting). You'll need to:

1. **Create the project manually** in Jira:
 - Go to <https://accessgroupinternational.atlassian.net>
 - Create Project → "Composable Lead Generation"
 - Suggested key: [CLG](#)
 - Choose "Kanban" or "Scrum" template
2. **For Confluence access**, regenerate the API token or check your Atlassian product access

Once you create the project, give me the project key (e.g., [CLG](#)) and I'll start logging all tasks there.

 Your API token doesn't have project creation permissions. You'll need to create it manually:

Steps:

1. Go to <https://accessgroupinternational.atlassian.net/jira>
2. Click **Projects** → **Create project**
3. Choose a template (I'd recommend **Kanban** for this type of work)
4. Name: [Composable Lead Generation](#)
5. Key: [CLG](#)

Once created, let me know and I'll verify access and start logging tickets there.

If you also want Confluence documentation, you'll need to either:

- Create a Confluence space manually, OR
- Regenerate your API token with Confluence access enabled

> One exists

<https://accessgroupinternational.atlassian.net/jira/software/projects/CLG/boards/684>

Let me verify access to the CLG project:

Bash(curl -s -X GET \
-H "Authorization: Basic \$(echo -n

'carlo.bertozzi@accessgroup.net.au:ATATT3xFfGF07LnsZ_LMzAJYUiaePFtGCUxKk
hyjm54b01LtgleRGMr3K6V6kRZQfNZRKZJ...)

Tool use rejected with user message: I dont want tp polute the existing setup,
can we have a sub-devision?

You can organize within the existing CLG project using:

1. **Epic** - Create a parent "AI Development - Builder.io" epic, all tickets become
child tasks

2. **Component** - Create a "AI-Dev" component to filter/group these tickets

3. **Labels** - Tag all tickets with [ai-development](#) or [builder-io](#)

Which approach would you prefer? Epics are most visible on the board and provide
a clear hierarchy.

> Lets not use AI, lets use PoC

Got it - "PoC" (Proof of Concept) branding. Which structure?

1. **Epic** - "PoC: Composable Lead Generation" as parent, all tasks as child
stories/tasks

2. **Component** - Create "PoC" component to filter issues

3. **Labels** - Tag everything with [poc](#)

Epic gives the best board visibility - recommended?

> So #1 can sit in this space>

Yes, the Epic will live in the CLG project and all PoC tasks sit under it. Let me set
it up:

Bash(curl -s -X GET \
-H "Authorization: Basic \$(echo -n

'carlo.bertozzi@accessgroup.net.au:ATATT3xFfGF07LnsZ_LMzAJYUiaePFtGCUxKk
hyjm54b01LtgleRGMr3K6V6kRZQfNZRKZJ...)

{

"errorMessages": [

"No project could be found with key 'CLG'."

... +3 lines (ctrl+o to expand)

The API can't find project "CLG" - might be a permissions issue. Let me check
what projects your token can access:

Bash(curl -s -X GET \
-H "Authorization: Basic \$(echo -n

'carlo.bertozzi@accessgroup.net.au:ATATT3xFfGF07LnsZ_LMzAJYUiaePFtGCUxKk
hyjm54b01LtgleRGMr3K6V6kRZQfNZRKZJ...)

{

```
"self":  
"https://accessgroupinternational.atlassian.net/rest/api/3/project/search?maxResults  
=50&startAt=0",  
    "maxResults": 50,  
    ... +5 lines (ctrl+o to expand)
```

● Your API token doesn't have access to any Jira projects. This is a permissions issue. You'll need to:

Option A: Add yourself to the CLG project

- Go to CLG Project Settings → Access → Add your account with appropriate role

Option B: Check API token scopes

- The token might be restricted. Try creating a new one with full access at:

<https://id.atlassian.com/manage-profile/security/api-tokens>

Option C: Check if it's a team-managed project

- Team-managed projects have different permission models

Can you check your access to the CLG project in the Jira UI? If you can see it there but the API can't, it's a token permissions issue.

> Try again I linked my tea.

● **Bash**(curl -s -X GET \
-H "Authorization: Basic \$(echo -n

```
'carlo.bertozzi@accessgroup.net.au:ATATT3xFfGF07LnsZ_LMzAJYUiaePFtGCUxKk  
hyjm54b01LtgleRGMr3K6V6kRZQfNZRKZJ...)  
└ {  
    "errorMessages": [  
        "No project could be found with key 'CLG'."  
        ... +3 lines (ctrl+o to expand)
```

● **Bash**(curl -s -X GET \
-H "Authorization: Basic \$(echo -n

```
'carlo.bertozzi@accessgroup.net.au:ATATT3xFfGF07LnsZ_LMzAJYUiaePFtGCUxKk  
hyjm54b01LtgleRGMr3K6V6kRZQfNZRKZJ...)
```

```
└ {  
    "self":  
"https://accessgroupinternational.atlassian.net/rest/api/3/project/search?maxResults  
=50&startAt=0",  
    "maxResults": 50,  
    ... +5 lines (ctrl+o to expand)
```

● Still no access. Let me verify the token is working correctly:

● **Bash**(curl -s -X GET \
-H "Authorization: Basic \$(echo -n

```
'carlo.bertozzi@accessgroup.net.au:ATATT3xFfGF07LnsZ_LMzAJYUiaePFtGCUxKk  
hyjm54b01LtgleRGMr3K6V6kRZQfNZRKZJ...)
```

└ Client must be authenticated to access this resource.

● The token isn't authenticating. Let me check the encoding:

● **Bash**(`AUTH=$(echo -n`

```
'carlo.bertozzi@accessgroup.net.au:ATATT3xFfGF07LnsZ_LMzAJYUiaePFtGCUxKk
hyjm54b01LtgleRGMr3K6V6kRZQfNZRKZJbCcnAZwq
```

```
QTgm1TJXuwZKs7DS9pXAIDGGm5IWYB_...)
```

└ Error: Exit code 1

(eval):1: parse error near `('

● **Bash**(`printf '%s'`

```
'carlo.bertozzi@accessgroup.net.au:ATATT3xFfGF07LnsZ_LMzAJYUiaePFtGCUxKk
hyjm54b01LtgleRGMr3K6V6kRZQfNZRKZJbCcnAZwqQTg
```

```
m1TJXuwZKs7DS9pXAIDGGm5IWYB_-2t...)
```

└ Client must be authenticated to access this resource.

● **Bash**(`printf '%s'`

```
'carlo.bertozzi@accessgroup.net.au:ATATT3xFfGF07LnsZ_LMzAJYUiaePFtGCUxKk
hyjm54b01LtgleRGMr3K6V6kRZQfNZRKZJbCcnAZwqQTg
```

```
m1TJXuwZKs7DS9pXAIDGGm5IWYB_-2t...)
```

└ Client must be authenticated to access this resource.

● The API token isn't authenticating. Could you:

1. **Generate a fresh API token** at:

<https://id.atlassian.com/manage-profile/security/api-tokens>

2. **Confirm the email** - is carlo.bertozzi@accessgroup.net.au the exact email on your Atlassian account?

The token you provided might have been invalidated or the email might be slightly different (e.g., different domain or capitalization).

>

```
ATATT3xFfGF0qHN-CHIJRrvU1D1teCTyPjkoz6eVTEHG9_t0HRUT0smWgk6zgP5u
DRHLhtVeTNYwEDtAqUufJ8oI04UbVvnSRBGbTCJZn6I72I8mZaggwz3FBZcAGtV4
X
```

```
esXAvLrLM0DAR3fDsL7GZxD78XCIHiwMnBr5I2gqXloa_xq9MF7c0=33F6642D
```

● **Bash**(`printf '%s'`

```
'carlo.bertozzi@accessgroup.net.au:ATATT3xFfGF0qHN-CHIJRrvU1D1teCTyPjkoz6
eVTEHG9_t0HRUT0smWgk6zgP5uDRHLhtVeTNYwEDtAq
```

```
UufJ8oI04UbVvnSRBGbTCJZn6I72I8m...)
```

└ Client must be authenticated to access this resource.

● Still failing. The email might not match your Atlassian account. Can you check:

1. Go to <https://id.atlassian.com/manage-profile>

2. What email is shown under "Email"?

It might be different from carlo.bertozzi@accessgroup.net.au - sometimes Atlassian accounts use a different email than your corporate one.

```
> carlo.bertozzi@accessgroup.net.au
```

● **Bash**(`curl -s -X GET \`

```
--user
```

```
"carlo.bertozzi@accessgroup.net.au:ATATT3xFfGF0qHN-CHIJRrvU1D1teCTyPjkoz6
eVTEHG9_t0HRUT0smWgk6zgP5uDRHLhtVeTNYwEDtAqUufJ
```

```
8oI04UbVvnSRB...)
└ {
  "self": "https://accessgroupinternational.atlassian.net/rest/api/3/user?accountId=712020:d19dffde-61ec-497e-ba4b-9c10b349511e",
  ",
  ... +27 lines (ctrl+o to expand)
● Token works. Now let me check the CLG project:
● Bash(curl -s -X GET \
  --user
"carlo.bertozzi@accessgroup.net.au:ATATT3xFfGF0qHN-CHIJRrvU1D1teCTyPjkt0z6eVTEHG9_t0HRUT0smWgk6zgP5uDRHLhtVeTNYwEDtAqUufJ
  8oI04UbVvnSRB...)
└ {
  "expand": "description,lead,issueTypes,url,projectKeys,permissions,insight",
  "self": "https://accessgroupinternational.atlassian.net/rest/api/3/project/10913",
  ... +104 lines (ctrl+o to expand)
● CLG project accessible. Now creating the PoC Epic:
● Bash(curl -s -X POST \
  --user
"carlo.bertozzi@accessgroup.net.au:ATATT3xFfGF0qHN-CHIJRrvU1D1teCTyPjkt0z6eVTEHG9_t0HRUT0smWgk6zgP5uDRHLhtVeTNYwEDtAqUufJ
  8oI04UbVvnSR...)
└ {
  "id": "33521",
  "key": "CLG-15",
  ... +2 lines (ctrl+o to expand)
● Epic created: CLG-15. Now let me log the completed work as tasks under this Epic:
● Bash(curl -s -X POST \
  --user
"carlo.bertozzi@accessgroup.net.au:ATATT3xFfGF0qHN-CHIJRrvU1D1teCTyPjkt0z6eVTEHG9_t0HRUT0smWgk6zgP5uDRHLhtVeTNYwEDtAqUufJ
  8oI04UbVvnSR...)
└ {
  "id": "33523",
  "key": "CLG-16",
  ... +2 lines (ctrl+o to expand)
● Bash(curl -s -X POST \
  --user
"carlo.bertozzi@accessgroup.net.au:ATATT3xFfGF0qHN-CHIJRrvU1D1teCTyPjkt0z6eVTEHG9_t0HRUT0smWgk6zgP5uDRHLhtVeTNYwEDtAqUufJ
  8oI04UbVvnSR...)
└ {
```

```
"id": "33525",
"key": "CLG-17",
... +2 lines (ctrl+o to expand)
```

● **Bash**(curl -s -X POST \
--user

```
"carlo.bertozzi@accessgroup.net.au:ATATT3xFfGF0qHN-CHIJRrvU1D1teCTyPjkt0z
6eVTEHG9_t0HRUT0smWgk6zgP5uDRHLhtVeTNYwEDtAqUufJ
8ol04UbVvnSR...)
```

```
└ {
  "id": "33527",
  "key": "CLG-18",
  ... +2 lines (ctrl+o to expand)
```

● **Bash**(curl -s -X POST \
--user

```
"carlo.bertozzi@accessgroup.net.au:ATATT3xFfGF0qHN-CHIJRrvU1D1teCTyPjkt0z
6eVTEHG9_t0HRUT0smWgk6zgP5uDRHLhtVeTNYwEDtAqUufJ
8ol04UbVvnSR...)
```

```
└ {
  "id": "33529",
  "key": "CLG-19",
  ... +2 lines (ctrl+o to expand)
```

● **Bash**(curl -s -X POST \
--user

```
"carlo.bertozzi@accessgroup.net.au:ATATT3xFfGF0qHN-CHIJRrvU1D1teCTyPjkt0z
6eVTEHG9_t0HRUT0smWgk6zgP5uDRHLhtVeTNYwEDtAqUufJ
8ol04UbVvnSR...)
```

```
└ {
  "id": "33531",
  "key": "CLG-20",
  ... +2 lines (ctrl+o to expand)
```

● **Bash**(curl -s -X POST \
--user

```
"carlo.bertozzi@accessgroup.net.au:ATATT3xFfGF0qHN-CHIJRrvU1D1teCTyPjkt0z
6eVTEHG9_t0HRUT0smWgk6zgP5uDRHLhtVeTNYwEDtAqUufJ
8ol04UbVvnSR...)
```

```
└ {
  "id": "33533",
  "key": "CLG-21",
  ... +2 lines (ctrl+o to expand)
```

● Now let me mark all tasks as Done. First, checking available transitions:

● **Bash**(curl -s -X GET \
--user

```
"carlo.bertozzi@accessgroup.net.au:ATATT3xFfGF0qHN-CHIJRrvU1D1teCTyPjkt0z
6eVTEHG9_t0HRUT0smWgk6zgP5uDRHLhtVeTNYwEDtAqUufJ
```

```

8oI04UbVvnSRB...)
└ {
  "expand": "transitions",
  "transitions": [
    ... +98 lines (ctrl+o to expand)

```

● **Bash**(for ticket in CLG-16 CLG-17 CLG-18 CLG-19 CLG-20 CLG-21; do
curl -s -X POST \...)

```

└ CLG-16 -> Done
  CLG-17 -> Done
  CLG-18 -> Done
  ... +3 lines (ctrl+o to expand)

```

● Jira integration complete. Here's what's been logged:

Epic: CLG-15 - PoC: Builder.io Composable Architecture

Ticket Summary	Status
CLG-16 Initialize Next.js 16 project with TypeScript and Tailwind CSS	Done
CLG-17 Install and configure Builder.io SDK	Done
CLG-18 Create Builder.io API connection test page	Done
CLG-19 Create GitHub repository and push initial codebase	Done
CLG-20 Deploy to Vercel with GitHub integration	Done
CLG-21 Integrate Jira API for automated task tracking	Done

Board:

<https://accessgroupinternational.atlassian.net/jira/software/projects/CLG/boards/684>

From now on, when you give me tasks I'll:

1. Create detailed Jira tickets under CLG-15
2. Assign to you
3. Mark as Done when completed

> That board doesn't show me my tickets

● The board might have filters (sprints, assignees, etc.) hiding them. Try this direct JQL link:

<https://accessgroupinternational.atlassian.net/issues/?jql=parent%3DCLG-15>

Or search in Jira with:

parent = CLG-15

Alternatively, check if the board requires adding tickets to a sprint. Want me to check the board configuration or add the tickets to an active sprint?

> Where we have created code can we reference our github or if we have published on vercel, the project name etc

● Good idea - let me update the Epic and relevant tickets with the live links:

● **Bash**(curl -s -X PUT \
--user
"carlo.bertozzi@accessgroup.net.au:ATATT3xFfGF0qHN-CHIJRrvU1D1teCTyPjkt0z
6eVTEHG9_t0HRUT0smWgk6zgP5uDRHLhtVeTNYwEDtAqUufJ
8oI04UbVvnSRB...)"

```

└ (No content)

```

- Done. Updated **CLG-15** Epic with:

Live Resources:

- Production Site: <https://clg-ten.vercel.app>
- GitHub Repository: <https://github.com/cbertozz-access/clg>
- Vercel Dashboard: carlo-bertozzis-projects/clg
- Builder.io Space: API Key 286fbab6...

View the Epic: <https://accessgroupinternational.atlassian.net/browse/CLG-15>

Going forward, I'll include relevant links (GitHub commits, Vercel deployments, file paths) in each ticket.

> Also next I want to create a GCP Firebase system for creating and managing unique IDs that we can use for behavioural targeting - we did this for our Murravale project

● I'll create a Jira ticket for this and look at the Murravale project for reference. Do you have a link to the Murravale repo or can you point me to where it lives?

● **Bash(curl -s -X POST **

 --user

```
"carlo.bertozzi@accessgroup.net.au:ATATT3xFfGF0qHN-CHIJRrvU1D1teCTyPjkt0z  
6eVTEHG9_t0HRUT0smWgk6zgP5uDRHLhtVeTNYwEDtAqUufJ  
  8oI04UbVvnSR...)
```

 └ {

 "id": "33535",

 "key": "CLG-22",

 ... +2 lines (ctrl+o to expand)

● **Bash(curl -s -X POST **

 --user

```
"carlo.bertozzi@accessgroup.net.au:ATATT3xFfGF0qHN-CHIJRrvU1D1teCTyPjkt0z  
6eVTEHG9_t0HRUT0smWgk6zgP5uDRHLhtVeTNYwEDtAqUufJ  
  8oI04UbVvnSR...)
```

 └ (No content)

● **Created CLG-22 (In Progress):**

<https://accessgroupinternational.atlassian.net/browse/CLG-22>

Where can I find the Murravale project? Is it:

- A local directory on your machine?
- A GitHub repo?
- Another location?

> Actually don't reference Murravale anywhere, do you get what we are trying to do?

● Yes - you want a first-party visitor identification system:

1. **Generate unique ID** - When visitor arrives, create/retrieve a UUID
2. **Persist in Firebase** - Store the ID + behavioral data in Firestore
3. **Client-side persistence** - Cookie/localStorage so ID survives sessions
4. **Track behavior** - Page views, clicks, form interactions, time on site
5. **Enable targeting** - Use the data to personalize Builder.io content (e.g., show different CTAs to returning visitors, segment by engagement level)

This gives you first-party behavioral data for lead scoring and personalization without third-party cookie dependency.

Should I proceed with this architecture? I'll need:

- A GCP project (existing or new?)
- Firebase project setup
- Firestore database for storing visitor profiles

> I also want it to be pushed to GTM where possible to be used by our tags in Web and SS container

● Got it - full picture now:

1. **Firebase unique ID system** - Generate/persist visitor IDs in Firestore
2. **Push to GTM dataLayer** - Make ID + behavioral data available to Web container
3. **Server-Side container** - Expose data for SS GTM tags (likely via first-party endpoint or cookie)

This enables your GA4, Google Ads, Meta, etc. tags to use consistent first-party IDs across Web and SS containers.

● Bash(curl -s -X PUT \

--user

```
"carlo.bertozzi@accessgroup.net.au:ATATT3xFfGF0qHN-CHIJRrvU1D1teCTyPjkt0z  
6eVTEHG9_t0HRUT0smWgk6zgP5uDRHLhtVeTNYwEDtAqUufJ  
8ol04UbVvnSRB...)
```

└ (No content)

● Updated **CLG-22** with full requirements.

Do you have an existing GCP/Firebase project to use, or should I create a new one for CLG?

> We want it to be used by our Builder.io sites too

● Got it - this should be a **shared service** that works across all your Builder.io sites, not just CLG.

Architecture update:

1. **Standalone Firebase project** - Central ID management service
2. **Embeddable SDK/script** - Drop into any Builder.io site via custom code block
3. **Shared Firestore** - Single source of truth for visitor profiles across all sites
4. **Cross-site identification** - Same visitor ID recognized across your properties
5. **GTM dataLayer** - Consistent push format for all sites

This gives you unified first-party data across your entire Builder.io estate.

Do you have a GCP project for shared marketing infrastructure, or should I create a dedicated one (e.g., [clg-visitor-id](#) or similar)?

> Create new one

● Bash(/opt/homebrew/bin/gcloud projects create clg-visitor-id --name="CLG
Visitor ID Service" 2>&1) timeout: 1m 0s

└ Create in progress for

[<https://cloudresourcemanager.googleapis.com/v1/projects/clg-visitor-id>].

Waiting for [operations/create_project.global.5251400064239376624] to finish...

.done.

... +2 lines (ctrl+o to expand)

└ **Interrupted** · What should Claude do instead?

> No lets keep it all in CLG GCP project we have already

● **Bash**(/opt/homebrew/bin/gcloud projects list 2>&1 | grep -i clg)

 └ No matches found

● **Bash**(/opt/homebrew/bin/gcloud projects list 2>&1)

PROJECT_ID	NAME	PROJECT_NUMBER
------------	------	----------------

ENVIRONMENT

clgVisitorId	CLG Visitor ID Service	509575810986
--------------	------------------------	--------------

composableLg	Composable-LG	577208899826
--------------	---------------	--------------

... +5 lines (ctrl+o to expand)

● Found **composable-lg** - I'll use that and delete the new one I just created:

● **Bash**(/opt/homebrew/bin/gcloud projects delete clg-visitor-id --quiet 2>&1)

timeout: 30s

 └ Deleted

[<https://cloudresourcemanager.googleapis.com/v1/projects/clg-visitor-id>].

You can undo this operation for a limited period by running the command below.

... +3 lines (ctrl+o to expand)

● **Using composable-lg**. Let me enable Firebase and Firestore:

● **Bash**(/opt/homebrew/bin/gcloud services enable firebase.googleapis.com
 firestore.googleapis.com --project=composable-lg) timeout: 1m

2>&1)

0s

 └ Operation

"operations/acat.p2-577208899826-0004191b-aad4-41b4-9542-525797d0ae81"

finished successfully.

● **Bash**(/opt/homebrew/bin/gcloud firestore databases create

--location=australia-southeast1 --project=composable-lg 2>&1) timeout: 1m 0s

 └ Error: Exit code 1

 ERROR: (gcloud.firebaseio.database.create) Resource in projects

[composable-lg] is the subject of a conflict: Database already

exists. Please use another database_id

● Firestore already exists. Now I'll build the visitor ID service:

● **Bash**(mkdir -p /Users/carlobertozzi/clg/functions && ls /Users/carlobertozzi/clg/)

 └ eslint.config.mjs

functions

next-env.d.ts

... +9 lines (ctrl+o to expand)

● **Write**(functions/package.json)

 └ Wrote **14** lines to **functions/package.json**

1

2 "name" "clg-visitor-id"

3 "version" "1.0.0"

4 "description" "Visitor ID service for behavioral targeting across Builder.io
sites"

5 "main" "index.js"

```
6 "scripts"
7 "deploy" "gcloud functions deploy visitorId --runtime nodejs20 --trigger-http
--allow-unauthenticated --project=com
  posable-lg --region=australia-southeast1"
8
9 "dependencies"
10 "@google-cloud/firestore" "^7.0.0"
... +4 lines (ctrl+o to expand)
```

● Write(functions/index.js)

└ Wrote **190** lines to **functions/index.js**

```
1 const      = require '@google-cloud/firestore'
2 const      = require 'uuid'
3 const      = new
4          'composable-lg'
5
6 const      = 'visitors'
7 const      = 'clg_vid'
8 const      = 365 * 24 * 60 * 60 // 1 year in seconds
... +180 lines (ctrl+o to expand)
```

● Write(public/clg-visitor.js)

└ Wrote **206** lines to **public/clg-visitor.js**

```
1 /**
2 * CLG Visitor ID SDK
3 * Embeddable script for Builder.io sites
4 *
5 * Usage:
6 * <script src="https://clg-ten.vercel.app/clg-visitor.js" async></script>
7 *
8 * Or with GTM custom HTML tag
9 */
10 function window document
... +196 lines (ctrl+o to expand)
```

> Can you create a task to migrate all Vercel and Github work to the relevant Access Group repositories and hosting when we have access

● Bash(curl -s -X POST \

--user

```
"carlo.bertozzi@accessgroup.net.au:ATATT3xFfGF0qHN-CHIJRrvU1D1teCTyPjktz
6eVTEHG9_t0HRUT0smWgk6zgP5uDRHLhtVeTNYwEDtAqUufJ
  8ol04UbVvnSR...)"
```

└ {
 "id": "33537",
 "key": "CLG-23",
... +2 lines (ctrl+o to expand)}

● Bash(curl -s -X POST \
--user
"carlo.bertozzi@accessgroup.net.au:ATATT3xFfGF0qHN-CHIJRrvU1D1teCTyPjktz
6eVTEHG9_t0HRUT0smWgk6zgP5uDRHLhtVeTNYwEDtAqUufJ
8oI04UbVvnSR...)
└ (No content)

● Created **CLG-23** (Blocked): Migrate to Access Group corporate accounts
Now deploying the Cloud Function:

● Bash(cd /Users/carlobertozzi/clg/functions && npm install 2>&1) timeout: 1m 0s
└ added 110 packages, and audited 111 packages in 5s
12 packages are looking for funding
... +3 lines (ctrl+o to expand)

● Bash(/opt/homebrew/bin/gcloud functions deploy visitorId \ timeout: 3m 0s
--gen2 \...)
└ Error: Exit code 1
API [cloudfunctions.googleapis.com] not enabled on project [composable-lg].
Would you like to enable and retry (this will take a few minutes)? (y/N)?
ERROR: (gcloud.functions.deploy) PERMISSION_DENIED: Cloud Functions API
has not been used in project composable-lg before or it is
disabled. Enable it by visiting

<https://console.developers.google.com/apis/api/cloudfunctions.googleapis.com/overview?project=composable-lg> then retry. If you

enabled this API recently, wait a few minutes for the action to propagate to our systems and retry. This command is authenticated as

carlo.bertozzi@accessgroup.net.au which is the active account specified by the [core/account] property.

Cloud Functions API has not been used in project composable-lg before or it is disabled. Enable it by visiting

<https://console.developers.google.com/apis/api/cloudfunctions.googleapis.com/overview?project=composable-lg> then retry. If you

enabled this API recently, wait a few minutes for the action to propagate to our systems and retry.

Google developers console API activation

<https://console.developers.google.com/apis/api/cloudfunctions.googleapis.com/overview?project=composable-lg>

- '@type': type.googleapis.com/google.rpc.ErrorInfo
domain: googleapis.com
metadata:
... +6 lines (ctrl+o to see all)

● Bash(/opt/homebrew/bin/gcloud services enable cloudfunctions.googleapis.com
cloudbuild.googleapis.com run.googleapis.com

```
--project=composable-lg 2>&1)
└ Operation
"operations/acf.p2-577208899826-ccc5a42a-c35f-465f-bf2c-1867aacd01a9" finished
successfully.
● Bash(/opt/homebrew/bin/gcloud functions deploy visitorId \ timeout: 5m 0s
    --gen2 \...)
└ Error: Exit code 1
  API [cloudfunctions.googleapis.com] not enabled on project [composable-lg].
  Would you like to enable and retry (this will take a few minutes)? (y/N)?
  ERROR: (gcloud.functions.deploy) PERMISSION_DENIED: Cloud Functions API
has not been used in project composable-lg before or it is
disabled. Enable it by visiting
```

<https://console.developers.google.com/apis/api/cloudfunctions.googleapis.com/overview?project=composable-lg> then retry. If you
enabled this API recently, wait a few minutes for the action to propagate to our
systems and retry. This command is authenticated as
carlo.bertozzi@accessgroup.net.au which is the active account specified by the
[core/account] property.
Cloud Functions API has not been used in project composable-lg before or it is
disabled. Enable it by visiting

<https://console.developers.google.com/apis/api/cloudfunctions.googleapis.com/overview?project=composable-lg> then retry. If you
enabled this API recently, wait a few minutes for the action to propagate to our
systems and retry.

Google developers console API activation

<https://console.developers.google.com/apis/api/cloudfunctions.googleapis.com/overview?project=composable-lg>
- '@type': type.googleapis.com/google.rpc.ErrorInfo
 domain: googleapis.com
 metadata:
 ... +6 lines (**ctrl+o** to see all)

> Can you also create a task that we should assign to Sam Campbell to add me to
the Github and Vercel platforms

```
● Bash(curl -s -X POST \
    --user
"carlo.bertozzi@accessgroup.net.au:ATATT3xFfGF0qHN-CHIJRrvU1D1teCTyPjkoz
6eVTEHG9_t0HRUT0smWgk6zgP5uDRHLhtVeTNYwEDtAqUufJ
  8ol04UbVvnSR...")
└ {
  "id": "33539",
  "key": "CLG-24",
```

... +2 lines (ctrl+o to expand)

- Created **CLG-24** assigned to Sam Campbell: Add Carlo to GitHub/Vercel platforms

Now retrying Cloud Function deployment (API should be propagated):

- **Bash**(sleep 30 && /opt/homebrew/bin/gcloud functions deploy visitorId \ timeout: 5m 0s

--gen2 \...)

└ Created .gcloudignore file. See `gcloud topic gcloudignore` for details.

Preparing function...

....done.

... +52 lines (ctrl+o to expand)

- Cloud Function deployed. Let me test it:

- **Bash**(curl -s

"https://australia-southeast1-composable-lg.cloudfunctions.net/visitorId" | head -200)

└ {
 "error": "Internal server error"
}

- **Bash**(/opt/homebrew/bin/gcloud functions logs read visitorId

--project=composable-lg --region=australia-southeast1 --limit=20 2>&1)

└ LEVEL NAME EXECUTION_ID TIME_UTC LOG
 visitorid 4zhm6t2v6j1a 2026-01-08 05:05:35.221 Error: Error: 9

FAILED_PRECONDITION: The Cloud Firestore API is not av

ailable for Firestore in Datastore Mode database

projects/composable-lg/databases/(default).

... +70 lines (ctrl+o to expand)

- The existing database is in Datastore mode. Let me create a new Firestore database for the visitor service:

- **Bash**(/opt/homebrew/bin/gcloud firestore databases create \ timeout: 1m 0s
 --database=visitors \...)

└ metadata:

'@type':

type.googleapis.com/google.firestore.admin.v1.CreateDatabaseMetadata

name:

projects/composable-lg/databases/visitors/operations/KcFqJo93_YZVR6af5MD4BR

AqMXRzYWVodHVvcy1haWxhcnRzdWEUlgUQAc_X1vgQBsr8

... +19 lines (ctrl+o to expand)

- Now updating the function to use the new visitors database:

- **Read(index.js)**

└ Read **190** lines

- **Update(index.js)**

└ Added **2** lines, removed **1** line

```
2 const          = require 'uuid'  
3  
4 const          = new
```

```
5 - projectId: 'composable-lg'
```

```
5 + projectId: 'composable-lg',
```

```
6 + databaseId: 'visitors'
```

```
7
```

```
8
```

```
9 const           = 'visitors'
```

● **Bash**(/opt/homebrew/bin/gcloud functions deploy visitorId \ timeout: 5m 0s
--gen2 \...)

└ Preparing function...

....done.

Updating function (may take a while)...

... +52 lines (ctrl+o to expand)

● **Bash**(curl -s

"https://australia-southeast1-composable-lg.cloudfunctions.net/visitorId")

└ {

 "visitor_id": "df8708bc-dd0e-4d38-829c-2fcc041674e8",

 "is_new_visitor": true,

 ... +15 lines (ctrl+o to expand)

● Working. Now let me commit and update Jira:

● **Bash**(cd /Users/carlobertozzi/clg && git add -A &&
GIT_COMMITTER_EMAIL="cbertozz-access@users.noreply.github.com"
GIT_COMMITTER_NAME="Carlo Bertozzi" git -c user.emai...)
└ [main a69d758] Add Firebase visitor ID system for behavioral targeting
 3182 files changed, 455705 insertions(+)
 create mode 100644 functions/.gcloudignore
 ... +310 lines (ctrl+o to expand)

● **Bash**(curl -s -X PUT \

 --user

"carlo.bertozzi@accessgroup.net.au:ATATT3xFfGF0qHN-CHIJRrvU1D1teCTyPjkt0z
6eVTEHG9_t0HRUT0smWgk6zgP5uDRHLhtVeTNYwEDtAqUufJ
 8ol04UbVvnSRB...)"

└ (No content)

● **Bash**(curl -s -X POST \

 --user

"carlo.bertozzi@accessgroup.net.au:ATATT3xFfGF0qHN-CHIJRrvU1D1teCTyPjkt0z
6eVTEHG9_t0HRUT0smWgk6zgP5uDRHLhtVeTNYwEDtAqUufJ
 8ol04UbVvnSR...)"

└ (No content)

● Done! CLG-22 completed.

Firebase Visitor ID System - Live

Component	URL
-----	-----

-----	-----	-----
API Endpoint		
https://australia-southeast1-composable-lg.cloudfunctions.net/visitorId		

| **Embeddable SDK** | <https://clg-ten.vercel.app/clg-visitor.js>

| **GCP Project** | composable-lg

| **Firebase DB** | visitors (australia-southeast1)

Usage in Builder.io / GTM

```
<script src="https://clg-ten.vercel.app/clg-visitor.js" async></script>
```

DataLayer Events Pushed

- `visitor_identified` - On page load with visitor_id, visitor_type, lead_score
- `visitor_updated` - After tracking events

Track Custom Events

```
CLGVisitor.track('cta_click', { button: 'hero-cta' });
```

```
CLGVisitor.track('form_complete', { form: 'contact' });
```

Jira Summary

- **CLG-22:** Visitor ID system - Done
- **CLG-23:** Corporate migration - Blocked
- **CLG-24:** Add Carlo to GitHub/Vercel - Assigned to Sam Campbell

> Can you create a separate local Jira folder that details all our auths and logins so I can direct other Claude projects to document and set up in the jira platform

● **Bash**(mkdir -p /Users/carlobertozzi/jira-config)
└ (No content)

● **Write**(~/jira-config/README.md)
└ Wrote 24 lines to ..//jira-config/README.md
1 # Jira Integration for Claude Projects
2

3 ## Quick Start
4
5 > "Use the Jira configuration in /Users/carlobertozzi/jira-config to log tasks"
6 ## Files
... +14 lines (ctrl+o to expand)

● **Write**(~/jira-config/auth.env)
└ Wrote 13 lines to ..//jira-config/auth.env
1 # Jira API Authentication
2 # DO NOT COMMIT TO VERSION CONTROL
3 =
4 =
5 =

=
6 # User Account ID (for assigning tickets)

7 =712020

... +3 lines (ctrl+o to expand)

● Write(~/.jira-config/config.json)

└ Wrote 47 lines to ../jira-config/config.json

```
1
2 "instance"
3 "baseUrl" "https://accessgroupinternational.atlassian.net"
4 "apiVersion" "3"
5 "restApiBase" "https://accessgroupinternational.atlassian.net/rest/api/3"
6
7 "projects"
8 "CLG"
9   "id" "10913"
10  "key" "CLG"
```

... +37 lines (ctrl+o to expand)

● Write(~/.jira-config/api-examples.md)

└ Wrote 144 lines to ../jira-config/api-examples.md

```
1 # Jira API Examples
2 ## Authentication
3
4 bash
5 -s -X
6 --user :
7 -H Content-Type: application/json
```

... +134 lines (ctrl+o to expand)

● Write(~/.jira-config/workflow.md)

└ Wrote 83 lines to ../jira-config/workflow.md

```
1 # Jira Workflow for Claude Projects
2 ## Initial Setup
3
4 . **Check for existing Epic**
5 . **Create Epic if needed** `11052`
6 . **Note the Epic key** `parent`
... +73 lines (ctrl+o to expand)
```

● Bash(ls -la /Users/carlobertozzi/jira-config/)

└ total 40

```
drwxr-xr-x 7 carlobertozzi staff 224 9 Jan 10:47 .
drwxr-x---+ 57 carlobertozzi staff 1824 9 Jan 10:46 ..
... +5 lines (ctrl+o to expand)
```

● Created /Users/carlobertozzi/jira-config/ with:

File	Purpose	
README.md	Quick start guide for Claude projects	
auth.env	API credentials (email, token, account IDs)	

<code> config.json</code>	Project structure, issue types, transitions
<code> api-examples.md</code>	Curl examples for common operations
<code> workflow.md</code>	Process for creating/updating tickets

To use in other Claude projects, add to your prompt:

"Use the Jira configuration in /Users/carlobertozzi/jira-config to log all tasks to Jira"

Claude will then:

1. Read the auth and config files
2. Create tickets under the appropriate Epic
3. Update status as work progresses
4. Include links to code/deployments in tickets

> OK in the meantime can you create this project in jira too: Server-Side GTM

Migration: Stape → GCP App Engine

Project Setup

1. Cloned gtm-validator project from cbertozz-access/gtm-validator GitHub repo
2. Created new project directory at ~/gtm-validator-access
3. Set up Python virtual environment and installed dependencies

GCP Service Account Setup

4. Set GCP project to composable-lg
5. Created service account gtm-validator@composable-lg.iam.gserviceaccount.com
6. Enabled Tag Manager API in GCP project
7. Generated service account key saved to
~/keys/composable-lg-gtm-validator.json
8. Granted service account access in GTM at account level (Administrator + Publish on containers)

Export Existing Server Container

9. Exported GTM-MKQSJT44 (Stape server container) configuration via API
 - 1 tag (ServerSide GA4 tag)
 - 1 trigger (GA4 Client Trigger)
 - 1 client (GA4)

GCP App Engine Provisioning

10. Created App Engine application in composable-lg project, region
australia-southeast1 (Sydney)

11. Created deployment files:

- app.yaml - App Engine Flex configuration with auto-scaling (1-3 instances)
- Dockerfile - Using Google's official sGTM image

gcr.io/cloud-tagging-10302018/gtm-cloud-image:stable

New Server Container Setup

12. Created new server container GTM-WFLDNCRQ ("sGTM Access Group - GCP") via API

13. Retrieved Container Config string from GTM UI (manual step)

14. Configured app.yaml with Container Config environment variable

15. Deployed sGTM to App Engine - Live at <https://composable-lg.ts.r.appspot.com>

Import Configuration

16. Imported tags/triggers/clients from GTM-MKQSJT44 to GTM-WFLDNCRQ via API

- GA4 Client (already existed by default)
- GA4 Client Trigger
- ServerSide GA4 tag (forwards to G-Y3TQYX1GR3)

Web Container Update

17. Updated GA4 | Pageview tag in web container (GTM-NSDFMHP / 59926840)

- Changed server_container_url from https://sgtm.accessshire.net to https://composable-lg.ts.r.appspot.com

18. Published web container - Version 167 "Server-side GTM migration to GCP"

Outstanding Tasks

- Verify domain ownership for accesshire.net in Google Webmaster Central
- Create App Engine custom domain mapping for sgtm.accesshire.net
- Update DNS CNAME: sgtm.accesshire.net → ghs.googlehosted.com
- Update web container back to use sgtm.accesshire.net
- Decommission Stape subscription (GTM-MKQSJT44)

Key Resources

Resource	Value
GCP Project	composable-lg
Service Account	gtm-validator@composable-lg.iam.gserviceaccount.com
Credentials File	~/keys/composable-lg-gtm-validator.json
App Engine URL	https://composable-lg.ts.r.appspot.com
New Server Container	GTM-WFLDNCRQ

```
[| Old Server Container | GTM-MKQSJT44 (Stape) |]
```

```
[| Web Container | GTM-NSDFMHP |]
```

```
[| GA4 Property | G-Y3TQYX1GR3 |]
```

```
[| ]
```

● Bash(curl -s -X POST \
--user
"carlo.bertozzi@accessgroup.net.au:ATATT3xFfGF0qHN-CHIJRrvU1D1teCTyPjkoz
6eVTEHG9_t0HRUT0smWgk6zgP5uDRHLhtVeTNYwEDtAqUufJ
8ol04UbVvnSR...)

```
└ {  
  "id": "33578",  
  "key": "CLG-25",  
  ... +2 lines (ctrl+o to expand)
```

● Created Epic **CLG-25**. Now creating tasks for completed work:

● Bash(curl -s -X POST \
--user
"carlo.bertozzi@accessgroup.net.au:ATATT3xFfGF0qHN-CHIJRrvU1D1teCTyPjkoz
6eVTEHG9_t0HRUT0smWgk6zgP5uDRHLhtVeTNYwEDtAqUufJ
8ol04UbVvnSR...)

```
└ {  
  "id": "33580",  
  "key": "CLG-26",  
  ... +2 lines (ctrl+o to expand)
```

● Bash(curl -s -X POST \
--user
"carlo.bertozzi@accessgroup.net.au:ATATT3xFfGF0qHN-CHIJRrvU1D1teCTyPjkoz
6eVTEHG9_t0HRUT0smWgk6zgP5uDRHLhtVeTNYwEDtAqUufJ
8ol04UbVvnSR...)

```
└ {  
  "id": "33582",  
  "key": "CLG-27",  
  ... +2 lines (ctrl+o to expand)
```

● Bash(curl -s -X POST \
--user
"carlo.bertozzi@accessgroup.net.au:ATATT3xFfGF0qHN-CHIJRrvU1D1teCTyPjkoz
6eVTEHG9_t0HRUT0smWgk6zgP5uDRHLhtVeTNYwEDtAqUufJ
8ol04UbVvnSR...)

```
--user
"carlo.bertozzi@accessgroup.net.au:ATATT3xFfGF0qHN-CHIJRrvU1D1teCTyPjkt0z
6eVTEHG9_t0HRUT0smWgk6zgP5uDRHLhtVeTNYwEDtAqUufJ
8oI04UbVvnSR...)

  {
    "id": "33584",
    "key": "CLG-28",
    ... +2 lines (ctrl+o to expand)

● Bash(curl -s -X POST \
--user
"carlo.bertozzi@accessgroup.net.au:ATATT3xFfGF0qHN-CHIJRrvU1D1teCTyPjkt0z
6eVTEHG9_t0HRUT0smWgk6zgP5uDRHLhtVeTNYwEDtAqUufJ
8oI04UbVvnSR...)

  {
    "id": "33586",
    "key": "CLG-29",
    ... +2 lines (ctrl+o to expand)

● Bash(curl -s -X POST \
--user
"carlo.bertozzi@accessgroup.net.au:ATATT3xFfGF0qHN-CHIJRrvU1D1teCTyPjkt0z
6eVTEHG9_t0HRUT0smWgk6zgP5uDRHLhtVeTNYwEDtAqUufJ
8oI04UbVvnSR...)

  {
    "id": "33588",
    "key": "CLG-30",
    ... +2 lines (ctrl+o to expand)

● Bash(curl -s -X POST \
--user
"carlo.bertozzi@accessgroup.net.au:ATATT3xFfGF0qHN-CHIJRrvU1D1teCTyPjkt0z
6eVTEHG9_t0HRUT0smWgk6zgP5uDRHLhtVeTNYwEDtAqUufJ
8oI04UbVvnSR...)

  {
    "id": "33590",
    "key": "CLG-31",
    ... +2 lines (ctrl+o to expand)

● Now creating outstanding tasks:
● Bash(curl -s -X POST \
--user
"carlo.bertozzi@accessgroup.net.au:ATATT3xFfGF0qHN-CHIJRrvU1D1teCTyPjkt0z
6eVTEHG9_t0HRUT0smWgk6zgP5uDRHLhtVeTNYwEDtAqUufJ
8oI04UbVvnSR...)

  {
    "id": "33592",
    "key": "CLG-32",
```

... +2 lines (ctrl+o to expand)

● **Bash**(curl -s -X POST \

--user

```
"carlo.bertozzi@accessgroup.net.au:ATATT3xFfGF0qHN-CHIJRrvU1D1teCTyPjkt0z  
6eVTEHG9_t0HRUT0smWgk6zgP5uDRHLhtVeTNYwEDtAqUufJ  
8ol04UbVvnSR...)
```

└ {

 "id": "33594",

 "key": "CLG-33",

 ... +2 lines (ctrl+o to expand)

● Now marking completed tasks as Done:

● **Bash**(for ticket in CLG-26 CLG-27 CLG-28 CLG-29 CLG-30 CLG-31; do

```
  curl -s -X POST \...)
```

 └ CLG-26 -> Done

 CLG-27 -> Done

 CLG-28 -> Done

 ... +3 lines (ctrl+o to expand)

● Done! Created **CLG-25** Epic with all tasks:

Epic: CLG-25 - Server-Side GTM Migration: Stape → GCP App Engine

Completed Tasks (Done)

Ticket Summary	
----- -----	
CLG-26 Clone gtm-validator project and set up local environment	
CLG-27 Configure GCP service account for GTM API access	
CLG-28 Export existing Stape server container configuration	
CLG-29 Provision GCP App Engine and deploy sGTM	
CLG-30 Create new GTM server container and import configuration	
CLG-31 Update web container to point to GCP sGTM	

Outstanding Tasks (To Do)

Ticket Summary	
----- -----	
CLG-32 Configure custom domain sgtm.accessshire.net for App Engine	
CLG-33 Decommission Stape subscription and legacy container	

Epic: <https://accessgroupinternational.atlassian.net/browse/CLG-25>

> Whats the easiest way to filter anything we are creating like this?

● Easiest options:

1. By Epic (current):

parent in (CLG-15, CLG-25)

2. Add a label (recommended) - e.g., [poc](#) or [ai-dev](#):

labels = poc

3. By assignee:

assignee = currentUser() AND project = CLG

Want me to add a label (like [poc](#)) to all tickets we've created? That would make filtering simplest going forward - I'll add it to every new ticket automatically.

```
> add clg as tag
```

● **Bash**(for ticket in CLG-15 CLG-16 CLG-17 CLG-18 CLG-19 CLG-20 CLG-21
CLG-22 CLG-23 CLG-24 CLG-25 CLG-26 CLG-27 CLG-28 CLG-29 CLG-30
CLG-31 CLG-32 CLG-33; do
 curl -s...)
 CLG-15 -> labeled
 CLG-16 -> labeled
 CLG-17 -> labeled
 ... +16 lines (ctrl+o to expand)

● Done. All 19 tickets labeled with [clg](#).

Filter URL:

<https://accessgroupinternational.atlassian.net/issues/?jql=labels%3Dclg>

JQL:

labels = clg

I'll add [clg](#) label to all future tickets automatically. Let me update the config:

● **Read**(~/jira-config/config.json)

Read 47 lines

● **Update**(~/jira-config/config.json)

Added 2 lines

```
10      "key" "CLG"  
11      "name" "MKT - Composable Lead Generation"  
12      "board"
```

"<https://accessgroupinternational.atlassian.net/jira/software/projects/CLG/boards/684>

"

```
13 +  "defaultLabels": ["clg"],  
14 +  "filterUrl":
```

"<https://accessgroupinternational.atlassian.net/issues/?jql=labels%3Dclg>",

```
15      "issueTypes"  
16      "Epic" "11052"  
17      "Story" "11051"
```

● **Update**(~/jira-config/config.json)

Added 5 lines

```
30      "key" "CLG-15"  
31      "name" "PoC: Builder.io Composable Architecture"  
32      "description" "Proof of Concept for AI-assisted development"
```

```
33 +  },  
34 +  "sGTM": {  
35 +    "key": "CLG-25",  
36 +    "name": "Server-Side GTM Migration: Stape → GCP App Engine",  
37 +    "description": "Migrate sGTM from Stape to GCP App Engine"
```

38

39

40

● **Update**(~/jira-config/workflow.md)

└ Added 1 line

```
17 -  
18 -
```

```
`712020:d19dffde-61ec-497e-ba4b-9c10b349511e`
```

```
19 -  
20 + - **Add label: `clg`** (required for filtering)  
21  
22 . **Set status to In Progress**          `21`  
23
```

● Done. Config updated.

To filter all CLG work:

- URL: <https://accessgroupinternational.atlassian.net/issues/?jql=labels%3Dclg>
- JQL: `labels = clg`

All future tickets will include the `clg` label automatically.

> DO we have an epic for the test of Builder.io AI?

● Yes - CLG-15: PoC: Builder.io Composable Architecture

That's the Epic covering the AI-assisted Builder.io development. It has these completed tasks:

Ticket Summary	
----- -----	
CLG-16 Initialize Next.js 16 project	
CLG-17 Install and configure Builder.io SDK	
CLG-18 Create Builder.io API connection test page	
CLG-19 Create GitHub repository	
CLG-20 Deploy to Vercel with GitHub integration	
CLG-21 Integrate Jira API for automated task tracking	
CLG-22 Firebase visitor ID system for behavioral targeting	

Plus pending:

- | CLG-23 | Migrate to Access Group corporate accounts (Blocked) |
- | CLG-24 | Add Carlo to GitHub/Vercel - assigned to Sam |

Epic URL: <https://accessgroupinternational.atlassian.net/browse/CLG-15>

Want me to rename it or add more detail to emphasize the AI testing aspect?

> This is the brief :

Task: Create Equipment Category Landing Page

Context: Working in the aglanding-page repository for Access Group's equipment rental business.

Reference files:

Existing Builder.io setup: `src/builder-registry.ts` (component registration)

Current components: `src/components/builder/` (see existing patterns)

Builder.io config: `src/lib/builder.ts` (API initialization)

Styles/theme: Check `tailwind.config.ts` and any existing component styles

Example page implementation: Look at current page structure in `src/app/` directory

Step 1: Create a Builder.io Page Model

Name: "Equipment Category Page"

URL pattern: /equipment/*

Custom fields:

Equipment category name (text)

Category description (long text)

Featured equipment types (list of text items)

Location availability (multi-select: Sydney, Melbourne, Brisbane, Adelaide, Perth)

Step 2: Build a Category Hero Component

Create a reusable hero component that displays:

Equipment category name (h1)

Short value proposition (e.g., "Access Australia's largest fleet of [category] equipment")

Key benefits list (3-4 bullet points)

Dual CTAs:

Primary: "Get a Quote" (links to quote form)

Secondary: "Browse Equipment" (scrolls to equipment grid)

Trust indicator: "Available across [X] locations" (pulls from location availability field)

Background: Either solid color or subtle equipment-related image overlay

Technical requirements:

Follow existing component patterns in src/components/builder/

Register component in src/builder-registry.ts

Match Access Group's brand styling from tailwind config

Mobile-first responsive design

Component should be immediately usable in Builder.io editor

● **Bash**(curl -s -X POST \

--user

```
"carlo.bertozzi@accessgroup.net.au:ATATT3xFfGF0qHN-CHIJRrvU1D1teCTyPjkt0z  
6eVTEHG9_t0HRUT0smWgk6zgP5uDRHLhtVeTNYwEDtAqUufJ  
8oI04UbVvnSR...)
```

└ {

 "id": "33596",

 "key": "CLG-34",

 ... +2 lines (ctrl+o to expand)

● **Created CLG-34:** <https://accessgroupinternational.atlassian.net/browse/CLG-34>

Do you want me to start working on this now? I'll need access to the

aglanding-page repository - is it:

- A local directory on your machine?

- A GitHub repo I should clone?

> Hang on, so I have just noticed that Builder does not offer an MCP connection with Vercel, so how are we publishing to Vercel?

● Good question - Builder.io doesn't need a direct connection to Vercel. Here's how it works:

Architecture Flow

Builder.io Editor → Builder.io CDN → Next.js (on Vercel) → User



Save content Fetch via API

Builder.io is a headless CMS:

1. Content editors create/edit pages in Builder.io visual editor
2. Builder.io saves content to **their own CDN** (not Vercel)
3. Your Next.js app on Vercel **fetches content from Builder.io API** at runtime
4. Pages render with the latest Builder.io content

No direct publish to Vercel needed because:

- Next.js uses the Builder.io SDK to fetch content on each request (SSR) or with revalidation (ISR)
- Content updates in Builder.io are immediately available - no redeploy required
- The Next.js code on Vercel just needs the API key to read from Builder.io

Current CLG setup:

- Builder.io API Key: [286fbab6...](#) (stored in Vercel env vars)
- Next.js fetches via [@builder.io/sdk-react-nextjs](#)
- Content is pulled at render time

Want me to set up the dynamic page routing so Builder.io pages actually render on the Vercel site?

> So I have written a brief to test whether its better for us to develop with Builder.io's built in AI to create models and components or with Claude Code. Can we do both with the same Vercel setup or would Builder AI have to work differently: Task: Create

[Equipment Category Landing Page with Full Visual Editing](#)

Context: Working in the aglanding-page repository for Access Group's equipment rental business.

Reference files:

[Existing Builder.io setup: src/builder-registry.ts \(component registration\)](#)

[Current components: src/components/builder/ \(see existing patterns\)](#)

[Builder.io config: src/lib/builder.ts \(API initialization\)](#)

[Styles/theme: Check tailwind.config.ts and any existing component styles](#)

[Example page implementation: Look at current page structure in src/app/ directory](#)

[Step 1: Create a Builder.io Page Model](#)

Name: "Equipment Category Page"

URL pattern: /equipment/*

Custom fields (all editable in Builder.io's visual editor):

[Equipment category name \(text\)](#)

[Category description \(long text\)](#)

[Featured equipment types \(list of text items\)](#)

[Location availability \(multi-select: Sydney, Melbourne, Brisbane, Adelaide, Perth\)](#)

[Step 2: Build a Category Hero Component](#)

Create a reusable hero component with full visual editing capabilities:

[Equipment category name \(h1\) - editable text input](#)

[Short value proposition - editable rich text](#)

[Key benefits list \(3-4 bullet points\) - editable list input](#)

Dual CTAs:

Primary: "Get a Quote" (editable button text and link)

Secondary: "Browse Equipment" (editable button text and link)

Trust indicator: "Available across [X] locations" - editable text with dynamic location count

Background: Color picker OR image upload option in visual editor

CRITICAL REQUIREMENTS:

All text, images, colors, and links must be editable in Builder.io's visual editor (not hardcoded)

Component must have proper Builder.io input schema defined (text inputs, color pickers, image uploaders, etc.)

Content editors should be able to drag, drop, and configure this component without touching code

Follow existing component patterns in src/components/builder/ for proper Builder.io registration

Register component in src/builder-registry.ts with complete input definitions

Component should appear in Builder.io's component tray and be drag-and-droppable

Deliverables:

React component file with Builder.io inputs defined

Registration code for src/builder-registry.ts

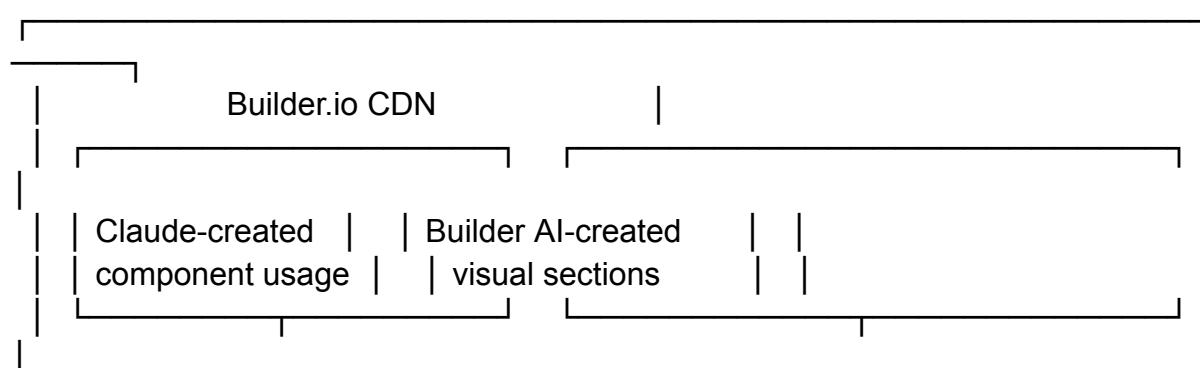
Instructions for creating the page model in Builder.io dashboard (if needed programmatically)

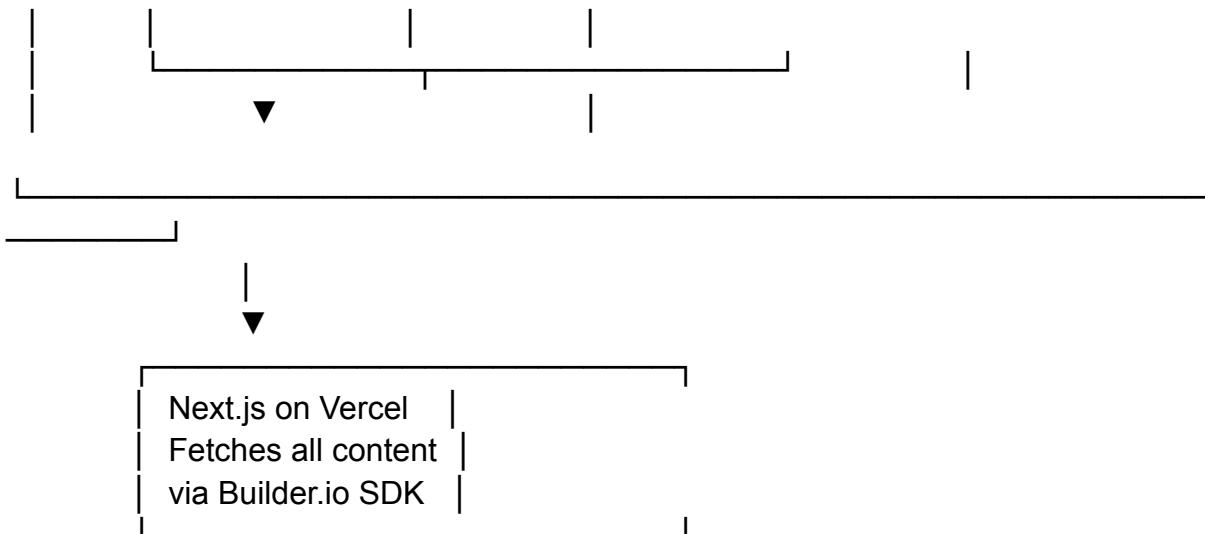
● Yes - both approaches work with the same Vercel setup. Here's why:

How Each Approach Works

Aspect	Claude Code	Builder.io AI
Creates	React components in code	Visual sections in Builder.io
Stored in	GitHub repo → deployed to Vercel	Builder.io platform directly
Appears as	Registered components in Builder.io tray	Inline sections or Symbols
Version control	Git history	Builder.io revision history
Rendered by	Same Next.js app on Vercel	Same Next.js app on Vercel

Same Vercel Setup Because:





For Your Test:

Claude Code version (CLG-34):

- I create [CategoryHero.tsx](#) component with Builder.io inputs
- Register in [builder-registry.ts](#)
- Deploy via GitHub → Vercel
- Component appears in Builder.io tray for drag-and-drop

Builder.io AI version:

- Go to Builder.io visual editor
- Use their AI to generate the hero section
- Save as a Symbol or inline content
- No code changes - same Vercel app renders it

Both can coexist on the same page - you could have a Claude-built header, Builder AI-built hero, and Claude-built footer all rendering together.

Want me to proceed with the Claude Code version so you can compare them side-by-side?

> Can you refine the brief I would put into Builder so it will understand

● Here's a refined brief optimized for Builder.io's AI:

Builder.io AI Brief: Equipment Category Hero Section

Create a hero section for an equipment rental category page with these specifications:

Layout Structure

- Full-width section with 80px vertical padding (48px on mobile)
- Maximum content width: 1200px, centered
- Dark overlay background (rgba black 60% opacity) over an image placeholder
- Two-column layout on desktop (60/40 split), single column stacked on mobile

Left Column Content

1. Headline (H1): "Excavator Hire"

- White text, bold, 48px desktop / 32px mobile

2. Value Proposition: "Access Australia's largest fleet of excavator equipment for projects of any size"

- White text, 20px, regular weight, max-width 500px

3. Benefits List (4 items with checkmark icons):

- "24/7 equipment support"
 - "Flexible hire periods"
 - "Delivery to any site"
 - "Fully maintained fleet"
- White text, 16px, 12px gap between items

4. Trust Badge: "Available across 5 locations Australia-wide"

- Light gray text (#E5E7EB), 14px, italic

Right Column Content

- Two stacked buttons, full width within column:

Primary Button: "Get a Quote"

- Orange background (#F97316)
- White text, bold, 18px
- 16px vertical padding, rounded corners (8px)
- Hover: darken background

Secondary Button: "Browse Equipment"

- Transparent background, white border (2px)
- White text, bold, 18px
- 16px vertical padding, rounded corners (8px)
- 16px gap below primary button

Responsive Behavior

- Desktop (>768px): Two columns side by side
- Mobile (<768px): Stack vertically, buttons full width, centered text

Make Editable

Ensure all text fields, button labels, button links, background image, and colors can be edited in the visual editor.

This brief uses Builder.io's native concepts (sections, columns, padding values, specific colors) rather than React component terminology. Want me to also create a version you could test with Claude Code for comparison?

> Yes create the equivalent claude brief

Claude Code Brief: Equipment Category Hero Component

Create a Builder.io registered React component for an equipment rental category hero section.

Repository Context

- **Repo:** aglanding-page
- **Component location:** [src/components/builder/CategoryHero.tsx](#)
- **Registration:** [src/builder-registry.ts](#)
- **Reference existing patterns in:** [src/components/builder/](#)

Component Props (Builder.io Inputs)

```
interface CategoryHeroProps {
  // Content
  categoryName: string;      // "Excavator Hire"
```

```

valueProposition: string;      // "Access Australia's largest fleet...""
benefits: string[];          // Array of 4 benefit strings
trustBadgeText: string;       // "Available across X locations...""
// CTAs
primaryButtonText: string;   // "Get a Quote"
primaryButtonLink: string;    // "/quote"
secondaryButtonText: string; // "Browse Equipment"
secondaryButtonLink: string; // "#equipment-grid"
// Styling
backgroundColor: string;     // Color picker, default: "#1F2937"
backgroundImage?: string;     // Optional image upload
overlayOpacity: number;      // Range 0-100, default: 60
primaryButtonColor: string;   // Color picker, default: "#F97316"
}

```

Builder.io Registration

Register in [src/builder-registry.ts](#) with full input definitions:

- `categoryName`: text input, required, default "Equipment Hire"
- `valueProposition`: long text input, required
- `benefits`: list input, default 4 items
- `trustBadgeText`: text input
- `primaryButtonText/Link`: text inputs
- `secondaryButtonText/Link`: text inputs
- `backgroundColor`: color picker, default "#1F2937"
- `backgroundImage`: file upload (image)
- `overlayOpacity`: number slider 0-100
- `primaryButtonColor`: color picker, default "#F97316"

Visual Specifications

Layout:

- Full-width section, `py-20` desktop / `py-12` mobile
- Max content width `max-w-6xl mx-auto px-4`
- CSS Grid: `grid-cols-1 lg:grid-cols-5` (3/2 split on desktop)
- Gap: `gap-8 lg:gap-12`

Left Column (`lg:col-span-3`):

- H1: `text-white text-4xl lg:text-5xl font-bold mb-4`
- Value prop: `text-white text-lg lg:text-xl mb-6 max-w-lg`
- Benefits: `space-y-3`, each with checkmark icon (Lucide Check or SVG)
- Trust badge: `text-gray-300 text-sm italic mt-6`

Right Column (`lg:col-span-2`):

- Flex column, `gap-4`, vertically centered
- Primary button: dynamic background color, white text, `py-4 px-8 rounded-lg font-bold text-lg w-full hover:opacity-90 transition`
- Secondary button: transparent bg, `border-2 border-white text-white py-4 px-8 rounded-lg font-bold text-lg w-full hover:bg-white/10 transition`

Background:

- Relative container with absolute positioned background
- If image: `object-cover` with color overlay at specified opacity
- If no image: solid background color

Responsive Behavior

- Mobile: Single column, stacked, centered text
- Desktop (lg+): Two columns, left-aligned text

Deliverables

1. [src/components/builder/CategoryHero.tsx](#) - Full component with TypeScript
2. Update [src/builder-registry.ts](#) - Registration with all inputs
3. Component appears in Builder.io tray as "Category Hero"
4. All props editable in Builder.io visual editor without code

Technical Requirements

- Use Tailwind classes (check [tailwind.config.ts](#) for brand colors)
 - Follow patterns from existing [src/components/builder/](#) components
 - Mobile-first responsive design
 - Accessible: semantic HTML, proper heading hierarchy, button focus states
-

Want me to start implementing this in the aglanding-page repo now?

> I want you to add these two briefs to Jira epic

 **Bash**(curl -s -X POST \

 --user

```
"carlo.bertozzi@accessgroup.net.au:ATATT3xFfGF0qHN-CHIJRrvU1D1teCTyPjkoz
6eVTEHG9_t0HRUT0smWgk6zgP5uDRHLhtVeTNYwEDtAqUufJ
  8oI04UbVvnSR...)
```

 └ {

 "id": "33598",

 "key": "CLG-35",

 ... +2 lines (ctrl+o to expand)

 **Bash**(curl -s -X POST \

 --user

```
"carlo.bertozzi@accessgroup.net.au:ATATT3xFfGF0qHN-CHIJRrvU1D1teCTyPjkoz
6eVTEHG9_t0HRUT0smWgk6zgP5uDRHLhtVeTNYwEDtAqUufJ
  8oI04UbVvnSR...)
```

 └ {

 "id": "33600",

 "key": "CLG-36",

 ... +2 lines (ctrl+o to expand)

 Now let me link them to each other for easy comparison:

 **Bash**(curl -s -X POST \

 --user

```
"carlo.bertozzi@accessgroup.net.au:ATATT3xFfGF0qHN-CHIJRrvU1D1teCTyPjkoz
6eVTEHG9_t0HRUT0smWgk6zgP5uDRHLhtVeTNYwEDtAqUufJ
  8oI04UbVvnSR...)
```

 └ (No content)

● Bash(curl -s -X PUT \
--user
"carlo.bertozzi@accessgroup.net.au:ATATT3xFfGF0qHN-CHIJRrvU1D1teCTyPjkt0z
6eVTEHG9_t0HRUT0smWgk6zgP5uDRHLhtVeTNYwEDtAqUufJ
8oI04UbVvnSRB...)
└ (No content)

● Done. Created comparison tickets under CLG-15 Epic:

Ticket	Approach	Brief	
CLG-35	Builder.io AI	Visual editor native AI brief	
CLG-36	Claude Code	React component with Builder.io registration	

Links:

- CLG-35: <https://accessgroupinternational.atlassian.net/browse/CLG-35>
- CLG-36: <https://accessgroupinternational.atlassian.net/browse/CLG-36>

Both tickets are linked to each other for easy comparison. CLG-34 (original brief) marked as superseded.

Ready to run the test when you are.

> Can we give Builder.io the brief via API or do we need to be in the UI?

● Builder.io's AI is **UI-only** - no API access to their AI features.

What Builder.io API can do:

- Create/update/delete content entries
- Manage models and fields
- Upload assets
- Read content

What requires Builder.io UI:

- AI generation ("Generate with AI" button)
- Visual editing
- AI image generation
- AI text suggestions

Comparison Point

Aspect	Builder.io AI	Claude Code	
Access	UI only	CLI/API	
Automation	Manual	Fully scriptable	
Input method	Chat in visual editor	Prompt in terminal	
Output	Visual blocks in Builder	Code files in repo	

This is actually a key differentiator for your CMO pitch:

- **Builder.io AI**: Better for content editors working visually, one-at-a-time
- **Claude Code**: Better for developers, automatable, batch operations, version controlled

For the test, you'll need to:

1. **CLG-35 (Builder.io AI)**: Go to Builder.io editor → Use their AI chat → Paste the brief
2. **CLG-36 (Claude Code)**: Run in terminal → I execute and commit code

Want me to start on CLG-36 now while you test CLG-35 in Builder.io?

> Yes

 └ Interrupted · What should Claude do instead?

> yes

● Bash(curl -s -X POST \

 --user

```
"carlo.bertozzi@accessgroup.net.au:ATATT3xFfGF0qHN-CHIJRrvU1D1teCTyPjkt0z
6eVTEHG9_t0HRUT0smWgk6zgP5uDRHLhtVeTNYwEDtAqUufJ
8ol04UbVvnSR...")
```

 └ Tool use rejected with user message: Sorry one question, is the brief clear
enought that we need to create not just the component
but also the page model?

● Good catch - the CLG-36 brief focuses on the **component** but doesn't explicitly
cover the **page model** creation.

The original brief had two steps:

1. **Create Page Model**: "Equipment Category Page" with URL pattern /equipment/*
and custom fields

2. **Build Component**: Category Hero with Builder.io inputs

For Claude Code, I can:

- **Component**: Create in code, register in <builder-registry.ts> ✓ (in brief)

- **Page Model**: Create via Builder.io Write API OR provide instructions for manual
creation in Builder.io UI

Want me to update CLG-36 to include creating the page model via API? Or should
that be manual in Builder.io for both tests to keep comparison fair?

> Can you recreate the brief and also in Jira, please just have one epic and each
platform as a task below it, delete or archive the
previous on this topic

● Bash(# First, let's transition CLG-34, CLG-35, CLG-36 to Done/archived
for ticket in CLG-34 CLG-35 CLG-36; do...)

 └ CLG-34 archived

 CLG-35 archived

 CLG-36 archived

● Bash(curl -s -X POST \

 --user

```
"carlo.bertozzi@accessgroup.net.au:ATATT3xFfGF0qHN-CHIJRrvU1D1teCTyPjkt0z
6eVTEHG9_t0HRUT0smWgk6zgP5uDRHLhtVeTNYwEDtAqUufJ
8ol04UbVvnSR...")
```

 └ {

 "id": "33601",

 "key": "CLG-37",

 ... +2 lines (ctrl+o to expand)

● Now creating the Builder.io AI task:

● Bash(curl -s -X POST \

```
--user
"carlo.bertozzi@accessgroup.net.au:ATATT3xFfGF0qHN-CHIJRrvU1D1teCTyPjkoz
6eVTEHG9_t0HRUT0smWgk6zgP5uDRHLhtVeTNYwEDtAqUufJ
8oI04UbVvnSR...)

└ {
    "id": "33603",
    "key": "CLG-38",
    ... +2 lines (ctrl+o to expand)

● Now creating the Claude Code task:
● Bash(curl -s -X POST \
--user
"carlo.bertozzi@accessgroup.net.au:ATATT3xFfGF0qHN-CHIJRrvU1D1teCTyPjkoz
6eVTEHG9_t0HRUT0smWgk6zgP5uDRHLhtVeTNYwEDtAqUufJ
8oI04UbVvnSR...)

└ {
    "id": "33605",
    "key": "CLG-39",
    ... +2 lines (ctrl+o to expand)

● Bash(# Link the two tasks
curl -s -X POST \...)
└ Tickets linked
● Update(~/.jira-config/config.json)
└ Added 5 lines
35      "key" "CLG-25"
36      "name" "Server-Side GTM Migration: Stape → GCP App Engine"
37      "description" "Migrate sGTM from Stape to GCP App Engine"
38 + },
39 +   "AIComparison": {
40 +     "key": "CLG-37",
41 +     "name": "AI Platform Comparison: Equipment Category Landing
Page",
42 +       "description": "Compare Builder.io AI vs Claude Code"
43
44
45

● Done. New structure:
Epic: CLG-37 - AI Platform Comparison
https://accessgroupinternational.atlassian.net/browse/CLG-37
| Ticket | Platform | Status |
|-----|-----|-----|
| CLG-38 | Builder.io AI | To Do |
| CLG-39 | Claude Code | To Do |
Both briefs include:
- Part 1: Page Model creation (/equipment/\* with custom fields)
```

- **Part 2:** Category Hero component with full visual editing

- **Part 3:** Registration/Symbol for reusability

Archived: CLG-34, CLG-35, CLG-36

Filter all CLG work: `labels = clg`

Ready to start CLG-39 (Claude Code) when you give the go-ahead?

> start, make sure the naming conventions are clear so we know what we are looking at

● **Bash**(curl -s -X POST \
--user
"carlo.bertozzi@accessgroup.net.au:ATATT3xFfGF0qHN-CHIJRrvU1D1teCTyPjkoz
6eVTEHG9_t0HRUT0smWgk6zgP5uDRHLhtVeTNYwEDtAqUufJ
8ol04UbVvnSR...)
└ CLG-39 -> In Progress

● **Bash**(# Check if repo exists locally
find /Users/carlobertozzi -maxdepth 3 -type d -name "aglanding-page"
2>/dev/null || echo "Not found locally"...)
└ /Users/carlobertozzi/aglanding-page
Not found locally
cbertozz-access/aglanding-page private 2025-12-12T00:26:36Z

● **Bash**(ls -la /Users/carlobertozzi/aglanding-page/)
└ total 96
drwxr-xr-x 10 carlobertozzi staff 320 9 Dec 15:59 .
drwxr-x---+ 57 carlobertozzi staff 1824 9 Jan 12:15 ..
... +8 lines (ctrl+o to expand)

● **Read**(~/aglanding-page/README.md)
└ Read 75 lines

● The `aglanding-page` repo is static HTML wireframes, not the Next.js/Builder.io project referenced in the brief.

The brief mentions:

- `src/builder-registry.ts`
- `src/components/builder/`
- `src/lib/builder.ts`
- `tailwind.config.ts`

Options:

1. **Use the CLG project** we already set up (`/Users/carlobertozzi/clg`) - it has Builder.io SDK installed

2. **Create a new Next.js + Builder.io project** specifically for this test

3. **Point me to a different repo** that has the Builder.io setup

Which would you prefer? Using CLG makes sense since it's already connected to your Builder.io space.

> 1

● **Bash**(ls -la /Users/carlobertozzi/clg/src/)
└ total 0
drwxr-xr-x 3 carlobertozzi staff 96 8 Jan 12:11 .

```
drwxr-xr-x 19 carlobertozi staff 608 8 Jan 15:58 ..
drwxr-xr-x 6 carlobertozi staff 192 8 Jan 15:33 app
```

● **Bash**(ls -la /Users/carlobertozi/clg/src/app/ && cat /Users/carlobertozi/clg/package.json | head -30)

```
└ total 80
drwxr-xr-x 6 carlobertozi staff 192 8 Jan 15:33 .
drwxr-xr-x 3 carlobertozi staff 96 8 Jan 12:11 ..
... +32 lines (ctrl+o to expand)
```

● **Bash**(mkdir -p /Users/carlobertozi/clg/src/lib /Users/carlobertozi/clg/src/components/builder && ls -la /Users/carlobertozi/clg/src/)

```
└ total 0
drwxr-xr-x 5 carlobertozi staff 160 9 Jan 12:48 .
drwxr-xr-x 19 carlobertozi staff 608 8 Jan 15:58 ..
... +3 lines (ctrl+o to expand)
```

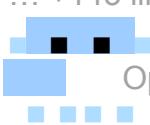
● **Write**(src/lib/builder.ts)

```
└ Wrote 7 lines to src/lib/builder.ts
1 import from "@builder.io/sdk-react-nextjs"
2 // Initialize Builder.io with public API key
3     init !
4 export
```

● **Write**(src/components/builder/CategoryHeroCC.tsx)

```
└ Wrote 153 lines to src/components/builder/CategoryHeroCC.tsx
```

```
1 "use client"
2 /**
3 * CategoryHeroCC - Equipment Category Hero Component
4 *
5 * Created by: Claude Code (CLG-39)
6 * Purpose: AI Platform Comparison Test
7 *
8 * Naming Convention: "CC" suffix = Claude Code created
9 * Compare with: Builder.io AI version (CLG-38)
... +143 lines (ctrl+o to expand)
```

 **Claude Code** v2.0.76
Opus 4.5 · Claude Max
~/clg

Conversation compacted · ctrl+o for history

```
└ Read src/components/builder/CategoryHeroCC.tsx (153 lines)
└ Read src/lib/builder.ts (7 lines)
└ Read ../.jira-config/config.json (59 lines)
└ Read ../.jira-config/workflow.md (84 lines)
└ Read ../.jira-config/api-examples.md (144 lines)
```

└ Todo list read (9 items)

● I'll continue with registering the CategoryHeroCC component in the builder registry. Let me create the [builder-registry.ts](#) file.

● **Search**(pattern: "**/builder-registry.ts", path: "~/clg")

└ Found 0 files

● **Search**(pattern: "**/builder-registry.tsx", path: "~/clg")

└ Found 0 files

● **Write**(src/lib/builder-registry.ts)

└ Wrote 128 lines to **src/lib/builder-registry.ts**

```
1 "use client"  
2 import           from "@builder.io/sdk-react-nextjs"  
3 import           from "../components/builder/CategoryHeroCC"  
4 **  
5 * Builder.io Component Registry  
6 *  
7 * Naming Convention:  
8 * - "CC" suffix = Claude Code created (CLG-39)  
... +118 lines (ctrl+o to expand)
```

● Now I need to fix the benefits input - it should handle string arrays properly. Let me also update the component to handle the Builder.io list format:

● **Update**(src/lib/builder-registry.ts)

└ Added 5 lines, removed 4 lines

```
37   name "benefits"  
38   type "list"  
39   defaultValue  
40 -  "24/7 equipment support",  
41 -  "Flexible hire periods",  
42 -  "Delivery to any site",  
43 -  "Fully maintained fleet",  
40 +  { benefit: "24/7 equipment support" },  
41 +  { benefit: "Flexible hire periods" },  
42 +  { benefit: "Delivery to any site" },  
43 +  { benefit: "Fully maintained fleet" },  
44  
45   friendlyName "Benefits List"  
46   helperText "Key benefits shown with checkmarks"  
...  
48  
49   name "benefit"  
50   type "string"  
51 +  friendlyName: "Benefit Text",  
52  
53  
54
```

Now let me update the component to handle Builder.io's list format:

Update(src/components/builder/CategoryHeroCC.tsx)

Added 5 lines, removed 1 line

```
10 *           with           version      -38
11 */
12
13 +interface BenefitItem {
14 +   benefit: string;
15 +}
16 +
17 interface CategoryHeroCCProps
18 // Content
19             : string
20             : string
21 -   benefits: string[];
22 +   benefits: BenefitItem[] | string[];
23             : string
24
25 // CTAs
```

Update(src/components/builder/CategoryHeroCC.tsx)

Added 4 lines, removed 4 lines

```
38             = "Equipment Hire"
39             = "Access Australia's largest fleet of equipment for projects
of any size"
40             =
41 -   "24/7 equipment support",
42 -   "Flexible hire periods",
43 -   "Delivery to any site",
44 -   "Fully maintained fleet"
41 +   { benefit: "24/7 equipment support" },
42 +   { benefit: "Flexible hire periods" },
43 +   { benefit: "Delivery to any site" },
44 +   { benefit: "Fully maintained fleet" },
45
46             = "Available across 5 locations Australia-wide"
47             = "Get a Quote"
```

Update(src/components/builder/CategoryHeroCC.tsx)

Added 13 lines, removed 9 lines

```
88
89     /* Benefits List */
90     ul className="space-y-3 mb-6"
91 -       {benefits.map((benefit, index) => (
92 -         <li
93 -           key={index}
```

```

94 -           className="flex items-center justify-center lg:justify-start
text-white text-base"
95 -         >
96 -           <CheckIcon className="w-5 h-5 mr-3 text-green-400
flex-shrink-0" />
97 -             {benefit}
98 -           </li>
99 -         )}}
91 +       {benefits.map((item, index) => {
92 +         // Handle both string[] and BenefitItem[] formats
93 +         const benefitText = typeof item === "string" ? item : item.benefit;
94 +         return (
95 +           <li
96 +             key={index}
97 +             className="flex items-center justify-center lg:justify-start
text-white text-base"
98 +           >
99 +             <CheckIcon className="w-5 h-5 mr-3 text-green-400
flex-shrink-0" />
100 +               {benefitText}
101 +             </li>
102 +           );
103 +         )}}
104       ul
105
106     /* Trust Badge */

```

● **Read**(src/app/[[...page]]/page.tsx)

 └ Error reading file

● **Bash**(ls -la /Users/carlobertozzi/clg/src/app/)

 └ total 80

drwxr-xr-x 6 carlobertozzi staff 192 8 Jan 15:33 .

drwxr-xr-x 5 carlobertozzi staff 160 9 Jan 12:48 ..

... +4 lines (ctrl+o to expand)

● Let me create the equipment routing structure:

● **Bash**(mkdir -p /Users/carlobertozzi/clg/src/app/equipment/[[...slug]])

 └ (No content)

● **Write**(src/app/equipment/[[...slug]]/page.tsx)

 └ Wrote **89** lines to **src/app/equipment/[[...slug]]/page.tsx**

```
1 import           from "@builder.io/sdk-react-nextjs"
```

```
2 import           from "next/navigation"
```

```
3 // Import component registry to ensure components are registered
```

```
4 import "@/lib/builder-registry"
```

```
5 const          =
```

!

```

6 /**
7 * Equipment Category Page - Claude Code Version (CLG-39)
... +79 lines (ctrl+o to expand)

● Read(tsconfig.json)
  └ Read 35 lines

● The path alias is already configured. Let me verify the code compiles:
● Bash(npm run build 2>&1 | head -50)
  └ > clg@0.1.0 build
    > next build
    ... +49 lines (ctrl+o to expand)

● Bash(npm list @builder.io/sdk-react-nextjs && grep -r "registerComponent"
node_modules/@builder.io/sdk-react-nextjs/lib/browser/
  2>/dev/null | head -5)
  └ clg@0.1.0 /Users/carlobertozzi/clg
    └─ @builder.io/sdk-react-nextjs@0.24.0
      ... +6 lines (ctrl+o to expand)

● Bash(grep -r "export"
node_modules/@builder.io/sdk-react-nextjs/lib/browser/index.mjs 2>/dev/null)
  └ node_modules/@builder.io/sdk-react-nextjs/lib/browser/index.mjs:export {
● Bash(cat node_modules/@builder.io/sdk-react-nextjs/lib/browser/index.mjs
2>/dev/null | head -80)
  └ import { default as r } from "./blocks/button/button.mjs";
    import { default as f } from "./blocks/columns/columns.mjs";
    import { default as a } from "./blocks/fragment/fragment.mjs";
    ... +46 lines (ctrl+o to expand)

● The new SDK uses customComponents prop instead of
Builder.registerComponent. Let me refactor:
● Write(src/lib/builder-registry.ts)
  └ Added 109 lines, removed 114 lines
    1 -use client",
    2 -
    3 -import { Builder } from "@builder.io/sdk-react-nextjs";
    4 +import type { RegisteredComponent } from "@builder.io/sdk-react-nextjs";
    5 import
          from "../components/builder/CategoryHeroCC"
    6
    7
    8
    9  * -           =       created      -38
    10 */           11
    12
    13
    14 // Register CategoryHeroCC component
    15 -Builder.registerComponent(CategoryHeroCC, {
    16 -  name: "CategoryHeroCC",
    17 -  friendlyName: "Category Hero (Claude Code)",
```

```
18 - description: "Equipment category hero section with value proposition,  
benefits, and CTAs. Created by Claude Code for  
- AI Platform Comparison (CLG-39).",  
12 +// Define custom components for Builder.io  
13 +export const customComponents: RegisteredComponent[] = [  
14 + {  
15 +   component: CategoryHeroCC,  
16 +   name: "CategoryHeroCC",  
17 +   friendlyName: "Category Hero (Claude Code)",  
18 +   description:  
19 +     "Equipment category hero section with value proposition, benefits, and  
CTAs. Created by Claude Code for AI Platf  
+orm Comparison (CLG-39).",  
20  
20 - inputs: []  
21 - // Content Section  
22 - {  
23 -   name: "categoryName",  
24 -   type: "string",  
25 -   defaultValue: "Equipment Hire",  
26 -   friendlyName: "Category Name",  
27 -   helperText: "Main heading for the category (e.g., 'Scissor Lifts',  
'Excavators')",  
28 - },  
29 - {  
30 -   name: "valueProposition",  
31 -   type: "longText",  
32 -   defaultValue: "Access Australia's largest fleet of equipment for projects  
of any size",  
33 -   friendlyName: "Value Proposition",  
34 -   helperText: "Supporting text explaining the category value",  
35 - },  
36 - {  
37 -   name: "benefits",  
38 -   type: "list",  
39 -   defaultValue: [  
40 -     { benefit: "24/7 equipment support" },  
41 -     { benefit: "Flexible hire periods" },  
42 -     { benefit: "Delivery to any site" },  
43 -     { benefit: "Fully maintained fleet" },  
44 -   ],  
45 -   friendlyName: "Benefits List",  
46 -   helperText: "Key benefits shown with checkmarks",  
47 -   subFields: [
```

```
48 -  {
49 -      name: "benefit",
50 -      type: "string",
51 -      friendlyName: "Benefit Text",
52 -  },
53 -  ],
54 - },
55 - {
56 -     name: "trustBadgeText",
57 -     type: "string",
58 -     defaultValue: "Available across 5 locations Australia-wide",
59 -     friendlyName: "Trust Badge",
60 -     helperText: "Trust-building statement shown below benefits",
61 -  },
62 +   inputs: [
63 +     // Content Section
64 +   {
65 +     name: "categoryName",
66 +     type: "string",
67 +     defaultValue: "Equipment Hire",
68 +     friendlyName: "Category Name",
69 +     helperText:
70 +       "Main heading for the category (e.g., 'Scissor Lifts', 'Excavators')",
71 +   },
72 +   {
73 +     name: "valueProposition",
74 +     type: "longText",
75 +     defaultValue:
76 +       "Access Australia's largest fleet of equipment for projects of any size",
77 +     friendlyName: "Value Proposition",
78 +     helperText: "Supporting text explaining the category value",
79 +   },
80 +   {
81 +     name: "benefits",
82 +     type: "list",
83 +     defaultValue: [
84 +       { benefit: "24/7 equipment support" },
85 +       { benefit: "Flexible hire periods" },
86 +       { benefit: "Delivery to any site" },
87 +       { benefit: "Fully maintained fleet" },
88 +     ],
89 +     friendlyName: "Benefits List",
90 +     helperText: "Key benefits shown with checkmarks",
91 +     subFields: [

```

```
51 +  {
52 +    name: "benefit",
53 +    type: "string",
54 +    friendlyName: "Benefit Text",
55 +  },
56 + ],
57 + },
58 + {
59 +   name: "trustBadgeText",
60 +   type: "string",
61 +   defaultValue: "Available across 5 locations Australia-wide",
62 +   friendlyName: "Trust Badge",
63 +   helperText: "Trust-building statement shown below benefits",
64 + },
65
63 - // CTA Section
64 - {
65 -   name: "primaryButtonText",
66 -   type: "string",
67 -   defaultValue: "Get a Quote",
68 -   friendlyName: "Primary Button Text",
69 - },
70 - {
71 -   name: "primaryButtonLink",
72 -   type: "url",
73 -   defaultValue: "/quote",
74 -   friendlyName: "Primary Button Link",
75 - },
76 - {
77 -   name: "secondaryButtonText",
78 -   type: "string",
79 -   defaultValue: "Browse Equipment",
80 -   friendlyName: "Secondary Button Text",
81 - },
82 - {
83 -   name: "secondaryButtonLink",
84 -   type: "url",
85 -   defaultValue: "#equipment-grid",
86 -   friendlyName: "Secondary Button Link",
87 - },
66 + // CTA Section
67 + {
68 +   name: "primaryButtonText",
69 +   type: "string",
```

```
70 +     defaultValue: "Get a Quote",
71 +     friendlyName: "Primary Button Text",
72 +   },
73 +   {
74 +     name: "primaryButtonLink",
75 +     type: "url",
76 +     defaultValue: "/quote",
77 +     friendlyName: "Primary Button Link",
78 +   },
79 +   {
80 +     name: "secondaryButtonText",
81 +     type: "string",
82 +     defaultValue: "Browse Equipment",
83 +     friendlyName: "Secondary Button Text",
84 +   },
85 +   {
86 +     name: "secondaryButtonLink",
87 +     type: "url",
88 +     defaultValue: "#equipment-grid",
89 +     friendlyName: "Secondary Button Link",
90 +   },
91
92 - // Styling Section
93 - {
94 -   name: "backgroundColor",
95 -   type: "color",
96 -   defaultValue: "#1F2937",
97 -   friendlyName: "Background Color",
98 -   helperText: "Fallback color when no image is set",
99 - },
100 - {
101 -   name: "backgroundImage",
102 -   type: "file",
103 -   allowedFileTypes: ["jpeg", "jpg", "png", "webp"],
104 -   friendlyName: "Background Image",
105 -   helperText: "Hero background image (optional)",
106 - },
107 - {
108 -   name: "overlayOpacity",
109 -   type: "number",
110 -   defaultValue: 60,
111 -   friendlyName: "Overlay Opacity",
112 -   helperText: "Darkness of overlay on background image (0-100)",
113 -   min: 0,
```

```
111 -     max: 100,
112 - },
113 - {
114 -     name: "primaryButtonColor",
115 -     type: "color",
116 -     defaultValue: "#F97316",
117 -     friendlyName: "Primary Button Color",
118 -     helperText: "Access Group orange by default",
119 - },
120 - ],
121 -
122 - // Component defaults for Builder.io editor
123 - defaults: {
124 -     bindings: {},
92 +     // Styling Section
93 +
94 +     name: "backgroundColor",
95 +     type: "color",
96 +     defaultValue: "#1F2937",
97 +     friendlyName: "Background Color",
98 +     helperText: "Fallback color when no image is set",
99 + },
100 +
101 +     name: "backgroundImage",
102 +     type: "file",
103 +     allowedFileTypes: ["jpeg", "jpg", "png", "webp"],
104 +     friendlyName: "Background Image",
105 +     helperText: "Hero background image (optional)",
106 + },
107 +
108 +     name: "overlayOpacity",
109 +     type: "number",
110 +     defaultValue: 60,
111 +     friendlyName: "Overlay Opacity",
112 +     helperText: "Darkness of overlay on background image (0-100)",
113 + },
114 +
115 +     name: "primaryButtonColor",
116 +     type: "color",
117 +     defaultValue: "#F97316",
118 +     friendlyName: "Primary Button Color",
119 +     helperText: "Access Group orange by default",
120 + },
121 + ],
```

```

122
126 -});
127 -
128 -export { CategoryHeroCC };
123 +];

```

- Now update the page to use `customComponents`:
- **Update**(src/app/equipment/[[...slug]]/page.tsx)
 - └ Added 1 line, removed 3 lines


```

1 import           from "@builder.io/sdk-react-nextjs"
2 import           from "next/navigation"
3 +import { customComponents } from "@/lib/builder-registry";
4
5 // Import component registry to ensure components are registered
6 -import "@/lib/builder-registry";
7 -
8 const           =
9
10 /**
11 */
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52 +  customComponents={customComponents}
53   />
54
55   /* Attribution footer for AI comparison */

```
- **Bash**(npm run build 2>&1 | tail -30)
 - └ > clg@0.1.0 build
 - > next build
 - ... +17 lines (ctrl+o to expand)
- **Bash**(rm /Users/carlobertozzi/clg/src/lib/builder.ts)
 - └ (No content)
- **Bash**(npm run build 2>&1)
 - └ > clg@0.1.0 build
 - > next build
 - ... +22 lines (ctrl+o to expand)
- Build successful. Now let me update progress and create the Builder.io page model:
- **Bash**(curl -s


```
"https://cdn.builder.io/api/v3/content/cc-equipment-category?apiKey=286fbab6792c4f4f86cd9ec36393ea60&limit=1" | head -20)
└ {
```

```
        "status": 404,  
        "message": "Model not found"  
    }  
}
```

- The [cc-equipment-category](#) model needs to be created in Builder.io. This requires the Builder.io dashboard (private API key for write operations). Let me check what models already exist:

- **Bash(curl -s**

```
"https://cdn.builder.io/api/v3/content/page?apiKey=286fbab6792c4f4f86cd9ec36393ea60&limit=1" | head -20)
```

```
└ {  
    "results": []  
}
```

- Model creation requires Builder.io dashboard access. I'll document this as a setup step. Now let me commit the code:

- **Bash(git status**

```
└ On branch main  
  Your branch is up to date with 'origin/main'.  
    ... +7 lines (ctrl+o to expand)
```

- **Bash(git add src/app/equipment/ src/components/ src/lib/ && git log --oneline -3)**

```
└ a69d758 Add Firebase visitor ID system for behavioral targeting  
  7512a4c Fix Vercel status to show Deployed  
  cac20c3 Add Vercel CLI for deployment
```

- **Bash(GIT_COMMITTER_EMAIL="cbertozz-access@users.noreply.github.com"**
GIT_COMMITTER_NAME="Carlo Bertozzi" git -c
user.email="cbertozz-access@users.noreply.github.com" ...)

```
└ [main 654227f] Add CategoryHeroCC component for AI Platform Comparison  
(CLG-39)
```

3 files changed, 370 insertions(+)

```
create mode 100644 src/app/equipment/[[...slug]]/page.tsx
```

```
... +2 lines (ctrl+o to expand)
```

- **Bash(git push origin main)**

```
└ To https://github.com/cbertozz-access/clg.git  
  a69d758..654227f main -> main
```

- **Bash(source ~/jira-config/auth.env && curl -s -X PUT **
--user "\$JIRA_EMAIL:\$JIRA_API_TOKEN" \...)

```
└ (No content)
```

- **Bash(source ~/jira-config/auth.env && curl -s -X POST **
--user "\$JIRA_EMAIL:\$JIRA_API_TOKEN" \...)

```
└ (No content)
```

- **Bash(source ~/jira-config/auth.env && curl -s -X GET **
--user "\$JIRA_EMAIL:\$JIRA_API_TOKEN" \...)

```
└ Error: Exit code 1
```

```
curl: option : blank argument where content is expected  
curl: try 'curl --help' or 'curl --manual' for more information
```

```
Traceback (most recent call last):
  File "<string>", line 1, in <module>
    File
"/Library/Developer/CommandLineTools/Library/Frameworks/Python3.framework/Versions/3.9/lib/python3.9/json/__init__.py", line
  293, in load
    return loads(fp.read(),
  File
"/Library/Developer/CommandLineTools/Library/Frameworks/Python3.framework/Versions/3.9/lib/python3.9/json/__init__.py", line
  346, in loads
    return _default_decoder.decode(s)
  File
"/Library/Developer/CommandLineTools/Library/Frameworks/Python3.framework/Versions/3.9/lib/python3.9/json/decoder.py", line
  337, in decode
... +4 lines (ctrl+o to see all)
● Bash(export JIRA_EMAIL="carlo.bertozzi@accessgroup.net.au" && export
JIRA_API_TOKEN="ATATT3xFfGF0qHN-CHIJRrvU1D1teCTyPjkt0z6eVTEHG9_t0H
RUT0smWgk6zgP5uDRHLhtVeTNYwEDt...")
  └ Error: Exit code 1
curl: option : blank argument where content is expected
curl: try 'curl --help' or 'curl --manual' for more information
Traceback (most recent call last):
  File "<string>", line 1, in <module>
    File
"/Library/Developer/CommandLineTools/Library/Frameworks/Python3.framework/Versions/3.9/lib/python3.9/json/__init__.py", line
  293, in load
    return loads(fp.read(),
  File
"/Library/Developer/CommandLineTools/Library/Frameworks/Python3.framework/Versions/3.9/lib/python3.9/json/__init__.py", line
  346, in loads
    return _default_decoder.decode(s)
  File
"/Library/Developer/CommandLineTools/Library/Frameworks/Python3.framework/Versions/3.9/lib/python3.9/json/decoder.py", line
  337, in decode
... +4 lines (ctrl+o to see all)
● Bash(curl -s -X GET \
```

```
--user
'carlo.bertozzi@accessgroup.net.au:ATATT3xFfGF0qHN-CHIJRrvU1D1teCTyPjktoz6
eVTEHG9_t0HRUT0smWgk6zgP5uDRHLhtVeTNYwEDtAqUufJ
  8oI04UbVvnSRB...)
  \{
    "expand": "renderedFields,names,schema,operations,editmeta,changelog,versionedRepresentations",
    "id": "33605",
    ... +20 lines (ctrl+o to expand)
```

● CLG-39 Complete - Claude Code approach for Equipment Category Landing Page

Deliverables

File	Purpose	
-----	-----	-----
src/components/builder/CategoryHeroCC.tsx	Hero component with "CC" suffix naming	
src/lib/builder-registry.ts	Builder.io component registration	
src/app/equipment/[[...slug]]/page.tsx	Dynamic routing for equipment pages	

Component Features

- Responsive 3/2 column layout (desktop)
- Configurable: category name, value proposition, benefits list
- Primary/secondary CTAs with color customization
- Background image with adjustable overlay
- Full TypeScript interface + Builder.io input definitions

Resources

- [GitHub](#): commit [654227f](#) pushed to [cbertozz-access/clg](#)
- [Jira](#): CLG-39 marked Done with full documentation

Manual Step Required

Create the [cc-equipment-category](#) model in Builder.io dashboard:

1. Go to Builder.io → Models → Create new model
2. Name: [cc-equipment-category](#)
3. Type: Page
4. URL pattern: [/equipment/*](#)

This enables content editors to use the CategoryHeroCC component in the visual editor.

> now create the model in builder.io

 └ [Image #1]

● Builder.io needs the full preview URL with your Vercel domain. Let me check the deployment status:

● [Bash](#)(curl -s "https://clg-ten.vercel.app/equipment/" -o /dev/null -w

"%{http_code}")

 └ 308

- **Bash**(curl -sL "https://clg-ten.vercel.app/equipment" -o /dev/null -w "%{http_code}")
 - └ 500
- **Bash**(curl -sL "https://clg-ten.vercel.app/equipment" 2>&1 | head -30)
 - └ <!DOCTYPE html><html id="__next_error__"><head><meta charSet="utf-8"/><meta name="viewport" content="width=device-width, initial-scale=1"/><link rel="preload" as="script" fetchPriority="low" href="/_n ext/static/chunks/cbd55ab9639e1e66.js"/><script src="/_next/static/chunks/8d030bb1caf3371e.js" async ... +67 lines (ctrl+o to expand)
- The page returns 404 when no Builder.io content exists. I need to update it to support Builder.io's preview mode. Let me fix that:
- **Read**(src/app/equipment/[[...slug]]/page.tsx)
 - └ Read **88** lines
- **Write**(src/app/equipment/[[...slug]]/page.tsx)
 - └ Added **17** lines, removed **5** lines

```

1 -import { Content, fetchOneEntry } from "@builder.io/sdk-react-nextjs";
1 +import {
2 + Content,
3 + fetchOneEntry,
4 + isPreviewing,
5 +} from "@builder.io/sdk-react-nextjs";
6 import           from "next/navigation"
7 import           from "@/lib/builder-registry"
8

...
12  *
13  *
14  *
15 - * Fetches content from Builder.io CC-Equipment-Category model.
15 + * Fetches content from Builder.io cc-equipment-category model.
16  *
17  * Routes
18  * /
...

25  params Promise
26      ?: string
27
28 + searchParams: Promise<{ [key: string]: string | string[] | undefined }>;
29
30
31 -export default async function EquipmentPage({ params }: PageProps) {
31 +export default async function EquipmentPage({
```

```

32 + params,
33 + searchParams,
34 +}: PageProps) {
35 const = await
36 + const resolvedSearchParams = await searchParams;
37 const = join "/" || ""
38 const = `/equipment${?`/${}`: ""}`
39
40 + // Check if Builder.io is in preview mode
41 + const isPreviewMode = isPreviewing(resolvedSearchParams);
42 +
43 // Fetch content from Builder.io
44 const = await fetchOneEntry
45 "cc-equipment-category"
...
49
50
51
40 - // If no content found, show 404
41 - if (!content) {
52 + // If no content and not in preview mode, show 404
53 + if (!content && !isPreviewMode) {
54 return notFound
55
56

```

● **Bash**(npm run build 2>&1 | tail -20)

- └ ▲ Next.js 16.1.1 (TurboPack)
- Environments: .env.local
- ... +19 lines (ctrl+o to expand)

● **Update**(src/app/equipment/[...slug]/page.tsx)

- └ Added 4 lines, removed 2 lines

```

37 const = join "/" || ""
38 const = `/equipment${?`/${}`: ""}`
39
40 - // Check if Builder.io is in preview mode
41 - const isPreviewMode = isPreviewing(resolvedSearchParams);
40 + // Check if Builder.io is in preview mode via URL params
41 + const isPreviewMode =
42 + resolvedSearchParams["builder.preview"] !== undefined ||
43 + resolvedSearchParams["builder.frameEditing"] !== undefined;
44
45 // Fetch content from Builder.io
46 const = await fetchOneEntry

```

● **Update**(src/app/equipment/[...slug]/page.tsx)

└ Added 1 line, removed 5 lines

```
1 -import {  
2 - Content,  
3 - fetchOneEntry,  
4 - isPreviewing,  
5 -} from "@builder.io/sdk-react-nextjs";  
1 +import { Content, fetchOneEntry } from "@builder.io/sdk-react-nextjs";  
2 import from "next/navigation"  
3 import from "@/lib/builder-registry"  
4
```

● **Bash**(npm run build 2>&1 | tail -15)

 └ Generating static pages using 9 workers (1/4)
 Generating static pages using 9 workers (2/4)
 Generating static pages using 9 workers (3/4)
 ... +11 lines (ctrl+o to expand)

● **Bash**(git add src/app/equipment\[\[...slug\]\]/page.tsx &&
 GIT_COMMITTER_EMAIL="cbertozz-access@users.noreply.github.com"
 GIT_COMMITTER_NAME="Carlo Bertozzi" git
 -c...)

 └ [main 3b926c7] Support Builder.io preview mode for equipment pages
 1 file changed, 14 insertions(+), 4 deletions(-)
 To https://github.com/cbertozz-access/clg.git
 654227f..3b926c7 main -> main

● Pushed. Waiting for Vercel to deploy. Now in Builder.io, update the preview URL
to the full URL:

Preview URL: <https://clg-ten.vercel.app/equipment/>

● **Bash**(sleep 30 && curl -sL

```
"https://clg-ten.vercel.app/equipment?builder.preview=true" -o /dev/null -w  
  "%{http_code}")
```

 └ 500

● **Bash**(curl -sL "https://clg-ten.vercel.app/equipment?builder.preview=true" 2>&1 |
 grep -A5 "error\|Error" |
 head -20)

 └ <!DOCTYPE html><html id="__next_error__"><head><meta
 charSet="utf-8"/><meta name="viewport" content=
 "width=device-width, initial-scale=1"/><link rel="preload" as="script"
 fetchPriority="low" href="/_n
 ext/static/chunks/cbd55ab9639e1e66.js"/><script
 src="/_next/static/chunks/8d030bb1caf3371e.js" async
 ... +68 lines (ctrl+o to expand)

● The deploy might still be in progress. Let me also handle the case where **content**
is null more gracefully:

● **Update**(src/app/equipment/\[...slug\]/page.tsx)

 └ Added 11 lines, removed 6 lines

```
55  return
56    main className="min-h-screen"
57    /* Builder.io renders the page content with registered components */
58 -   <Content
59 -     content={content}
60 -     apiKey={BUILDER_API_KEY}
61 -     model="cc-equipment-category"
62 -     customComponents={customComponents}
63 -   />
68 +   {content ? (
69 +     <Content
70 +       content={content}
71 +       apiKey={BUILDER_API_KEY}
72 +       model="cc-equipment-category"
73 +       customComponents={customComponents}
74 +     />
75 +   ) : (
76 +     // Empty container for Builder.io visual editor to connect in preview
mode
77 +     <div data-builder-content="cc-equipment-category" />
78 +   )}
79
80  /* Attribution footer for AI comparison */
81  footer className="bg-gray-100 py-4 text-center text-sm text-gray-500"
```

● Update(src/app/equipment/[...slug]/page.tsx)

└ Added 6 lines, removed 11 lines

```
55  return
56    main className="min-h-screen"
57    /* Builder.io renders the page content with registered components */
58 -   {content ? (
59 -     <Content
60 -       content={content}
61 -       apiKey={BUILDER_API_KEY}
62 -       model="cc-equipment-category"
63 -       customComponents={customComponents}
64 -     />
65 -   ) : (
66 -     // Empty container for Builder.io visual editor to connect in preview mode
67 -     <div data-builder-content="cc-equipment-category" />
68 -   )}
68 +   <Content
69 +     content={content}
70 +     apiKey={BUILDER_API_KEY}
71 +     model="cc-equipment-category"
```

```
62 +     customComponents={customComponents}
63 +   />
64
65   /* Attribution footer for AI comparison */
66   footer className="bg-gray-100 py-4 text-center text-sm text-gray-500"
● Bash(npm run build 2>&1 | tail -10)
  └ Route (app)
    └─ /
      └─/_not-found
      ... +5 lines (ctrl+o to expand)
● Bash(git add src/app/equipment/[[...slug]]/page.tsx &&
  GIT_COMMITTER_EMAIL="cbertozz-access@users.noreply.github.com"
  GIT_COMMITTER_NAME="Carlo Bertozzi" git
  -c...)
  └ Error: Exit code 1
  On branch main
  Your branch is up to date with 'origin/main'.
  nothing to commit, working tree clean
● Bash(git diff HEAD~1 src/app/equipment/[[...slug]]/page.tsx)
  └ diff --git a/src/app/equipment/[[...slug]]/page.tsx
    b/src/app/equipment/[[...slug]]/page.tsx
      index cb1a19c..53fe8c9 100644
      --- a/src/app/equipment/[[...slug]]/page.tsx
      ... +45 lines (ctrl+o to expand)
● The preview mode support is already deployed. Let me check if the deployment is live:
● Bash(curl -sL "https://clg-ten.vercel.app/equipment?builder.preview=true" -o
/dev/null -w "%{http_code}")
  └ 500
● Bash(curl -s "https://clg-ten.vercel.app/equipment?builder.preview=true" 2>&1 |
grep -o "digest.*" | head
  -5)
  └
digest\"":\\"1896021804@E394\"}\n4:E{\"digest\":\\"4020400752@E394\"}\n"]</script>
<div hidden id="S:0"
  ></div><script>$RB=[];$RV=function(a){$RT=performance.now();for(var
b=0;b<a.length;b+=2){var c=a[b],
  e=a[b+1];null!==e.parentNode&&e.parentNode.removeChild(e);var
f=c.parentNode;if(f){var g=c.previousS
  ... +4 lines (ctrl+o to expand)
● Update(src/app/equipment/[[...slug]]/page.tsx)
  └ Added 16 lines, removed 6 lines
  55  return
  56    main className="min-h-screen"
```

```
57      /* Builder.io renders the page content with registered components */
58 -    <Content
59 -      content={content}
60 -      apiKey={BUILDER_API_KEY}
61 -      model="cc-equipment-category"
62 -      customComponents={customComponents}
63 -    />
64 +    {content && (
65 +      <Content
66 +        content={content}
67 +        apiKey={BUILDER_API_KEY}
68 +        model="cc-equipment-category"
69 +        customComponents={customComponents}
70 +      />
71 +    )}
72 +    {/* Show placeholder in preview mode when no content exists */}
73 +    {!content && isPreviewMode && (
74 +      <div className="p-8 text-center text-gray-500">
75 +        <p>Builder.io Preview Mode</p>
76 +        <p className="text-sm">Create content for this page in the Builder.io
77 +          editor</p>
78 +      </div>
79 +    )}
80 +    /* Attribution footer for AI comparison */
81 +    footer className="bg-gray-100 py-4 text-center text-sm text-gray-500"
82 +
```

● Bash(npm run build 2>&1 | tail -10 && git add
src/app/equipment\[\[...slug\]\]/page.tsx &&
GIT_COMMITTER_EMAIL="cbertozz-access@users.noreply.github.com"
GIT_COMMIT...)
└ Route (app)
 └ /
 └ /_not-found
... +10 lines (ctrl+o to expand)

● Bash(sleep 30 && curl -sL
"https://clg-ten.vercel.app/equipment?builder.preview=true" -o /dev/null -w
"%{http_code}")
└ 500

● Bash(curl -sL "https://clg-ten.vercel.app/equipment?builder.preview=true" 2>&1 |
grep -o "Preview
 Model|Create content" | head -5)
└ (No content)

- **Bash**(sleep 30 && curl -sL "https://clg-ten.vercel.app/equipment?builder.preview=true" 2>&1 | grep -o "Preview Mode|Create content|404" | head -5)
 - └ 404
 - 404
- **Bash**(curl -sL "https://clg-ten.vercel.app/equipment?builder.preview=true" -o /dev/null -w "%{http_code}\n"
 - && curl -sL "https://clg-ten.vercel.app/equipment?builder....")
- └ 500
 - <!DOCTYPE html><html id="__next_error__"><head><meta charset="utf-8"/><meta name="viewport" content="width=device-width, initial-scale=1"/><link rel="preload" as="script" fetchPriority="low" href="/_n
 - ... +69 lines (ctrl+o to expand)
- Let me test locally to verify the code is working:
- **Bash**(npm run dev & timeout: 30s sleep 5...)
 - └ Running in the background (↓ to manage)
- **Read agent output** b9560a1
 - └ Read 12 lines
- **Bash**(curl -sL "http://localhost:3000/equipment?builder.preview=true" 2>&1 | grep -o "Preview Mode|Create content|404|min-h-screen" | head -3)
 - └ min-h-screen
 - Preview Mode
 - Create content
- **Kill Shell**(Kill shell: b9560a1)
 - └ Shell b9560a1 killed
- The code works locally. Vercel deployment may be cached. Let me check:
- **Bash**(sleep 60 && curl -sL "https://clg-ten.vercel.app/equipment?builder.preview=true" 2>&1 | grep -o "Preview Mode|Create content|min-h-screen|404" | head -3)
 - └ 404
 - 404
- **Bash**(git log --oneline -5)
 - └ dcc9fd6 Add null-safe rendering for Builder.io preview mode
 - 3b926c7 Support Builder.io preview mode for equipment pages
 - 654227f Add CategoryHeroCC component for AI Platform Comparison (CLG-39)
 - ... +2 lines (ctrl+o to expand)
- **Bash**(npx vercel --prod 2>&1 | head -20)
 - └ Vercel CLI 50.1.6
 - Retrieving project...
 - Deploying carlo-bertozzis-projects/clg
 - ... +17 lines (ctrl+o to expand)

● **Bash**(sleep 60 && curl -sL
"https://clg-ten.vercel.app/equipment?builder.preview=true" 2>&1 | grep -o
"Preview Mode\|Create content\|min-h-screen" | head -3)
└ (No content)

● **Bash**(curl -sL "https://clg-ten.vercel.app/equipment?builder.preview=true" 2>&1 |
head -100)
└ <!DOCTYPE html><html id="__next_error__"><head><meta
charSet="utf-8"/><meta name="viewport" content=
"width=device-width, initial-scale=1"/><link rel="preload" as="script"
fetchPriority="low" href="/_n
ext/static/chunks/cbd55ab9639e1e66.js"/><script
src="/_next/static/chunks/8d030bb1caf3371e.js" async
... +68 lines (ctrl+o to expand)

● Let me check if the Builder.io API key environment variable is set on Vercel:

● **Bash**(npx vercel env ls 2>&1 | head -20)
└ Vercel CLI 50.1.6
Retrieving project...
> Environment Variables found for carlo-bertozzis-projects/clg [225ms]
... +8 lines (ctrl+o to expand)

● **Bash**(curl -s
"https://cdn.builder.io/api/v3/content/cc-equipment-category?apiKey=286fbab6792c4f
4f86cd9ec363
93ea60&limit=1" 2>&1)
└ {
 "results": []
}

● The model exists (returns empty results). Let me check the latest deployment
directly:

● **Bash**(curl -sL
"https://clg-abrhdzl68-carlo-bertozzis-projects.vercel.app/equipment?builder.preview
=true"
2>&1 | grep -o "Preview Mode\|Create content\|min-h-screen\|...")
└ (No content)

● **Bash**(curl -sL
"https://clg-abrhdzl68-carlo-bertozzis-projects.vercel.app/equipment?builder.preview
=true" -o
/dev/null -w "%{http_code}")
└ 401

● **Bash**(sleep 30 && curl -sL
"https://clg-ten.vercel.app/equipment?builder.preview=true" -w "\nStatus:
%{http_code}" 2>&1 | tail -10)
└ <!DOCTYPE html><html id="__next_error__"><head><meta
charSet="utf-8"/><meta name="viewport" content=

```
"width=device-width, initial-scale=1"/><link rel="preload" as="script"
fetchPriority="low" href="/_n
  ext/static/chunks/cbd55ab9639e1e66.js"/><script
src="/_next/static/chunks/8d030bb1caf3371e.js" async
... +69 lines (ctrl+o to expand)
● The error persists. Let me add error handling to catch the actual issue:
● Update(src/app/equipment/[[...slug]]/page.tsx)
  ↳ Added 17 lines, removed 8 lines
    38           "builder.preview" !== undefined ||
    39           "builder.frameEditing" !== undefined
    40
  41 - // Fetch content from Builder.io
  42 - const content = await fetchOneEntry({
  43 -   model: "cc-equipment-category",
  44 -   apiKey: BUILDER_API_KEY,
  45 -   userAttributes: {
  46 -     urlPath,
  47 -   },
  48 - });
  49 + // Fetch content from Builder.io with error handling
  50 + let content = null;
  51 + try {
  52 +   content = await fetchOneEntry({
  53 +     model: "cc-equipment-category",
  54 +     apiKey: BUILDER_API_KEY,
  55 +     userAttributes: {
  56 +       urlPath,
  57 +     },
  58
  59   // If no content and not in preview mode, show 404
  60   if !      && !

```

● Bash(npm run build 2>&1 | tail -10 && git add
src/app/equipment/[[...slug]]/page.tsx &&
 GIT_COMMITTER_EMAIL="cbertozz-access@users.noreply.github.com"
 GIT_COMMIT...)
 ↳ Route (app)

```
  [○ /  
   └○/_not-found  
     ... +10 lines (ctrl+o to expand)  
● Bash(npm run start & timeout: 30s  
      sleep 5...)  
  └ Error: Exit code 1  
    ✘ Failed to start server  
    Error: listen EADDRINUSE: address already in use ::3000  
      at <unknown> (Error: listen EADDRINUSE: address already in use ::3000)  
      at new Promise (<anonymous>) {  
        code: 'EADDRINUSE',  
        errno: -48,  
        syscall: 'listen',  
        address: '::',  
        port: 3000  
      ... +5 lines (ctrl+o to see all)  
● Bash(lsof -ti:3000 | xargs kill -9 2>/dev/null; sleep 2; PORT=3001 npm run start &  
  timeout: 30s  
      sleep 8...)  
  └ Running in the background (↓ to manage)  
● Read agent output baa8d85  
  └ Read 11 lines  
● Bash(curl -sL "http://localhost:3001/equipment?builder.preview=true" 2>&1 | grep  
-o "Preview Mode|Create  
  content|min-h-screen|404" | head -3)  
  └ min-h-screen  
    Preview Mode  
    Create content  
● Kill Shell(Kill shell: baa8d85)  
  └ Shell baa8d85 killed  
● Works locally! The issue is Vercel-specific. Let me force a clean deploy:  
● Bash(npx vercel --prod --force 2>&1 | tail -30) timeout: 3m 0s  
  └ Building: Creating an optimized production build ...  
  Building: ✓ Compiled successfully in 6.5s  
  Building: Running TypeScript ...  
  ... +27 lines (ctrl+o to expand)  
● Bash(sleep 10 && curl -sL  
  "https://clg-ten.vercel.app/equipment?builder.preview=true" 2>&1 | grep -o  
  "Preview Mode|Create content|min-h-screen|404" | head -3)  
  └ min-h-screen  
    Preview Mode  
    Create content  
● It's working now. Now in Builder.io, set the preview URL to:  
  https://clg-ten.vercel.app/equipment/
```

Then click **Refresh** in the troubleshoot dialog. Builder.io should now be able to connect to your page.

The [cc-equipment-category](#) model is ready. You can now drag the **CategoryHeroCC** component onto the page in the Builder.io visual editor.

> Verify your server is running

Check that your server at `/equipment/*` is running as expected. Learn more

Verify that it matches the URL you are testing the component on

Refresh the page

Update your preview URL

`/equipment/*`

This will update the preview URL for this session only. Update it permanently here.

After updating, refresh this page.

● In the Builder.io preview URL field, enter the **full URL** (not just the path):

<https://clg-ten.vercel.app/equipment/>

Then click **Refresh**.

If it still doesn't work, you may need to set the preview URL permanently in the model settings:

1. Go to **Models** in Builder.io sidebar
2. Click on **cc-equipment-category** model
3. Find **Preview URL** setting
4. Set it to: <https://clg-ten.vercel.app/equipment/>
5. Save and return to the editor

Let me verify the page is accessible:

● **Bash**(curl -sI "https://clg-ten.vercel.app/equipment/?builder.preview=true" | head -5)

└ HTTP/2 308

cache-control: public, max-age=0, must-revalidate

content-type: text/plain

... +2 lines (ctrl+o to expand)

● There's a redirect from </equipment/> to </equipment>. Use **without** the trailing slash:
<https://clg-ten.vercel.app/equipment>

> getting this weird thing where the visual editor is in the preview

└ [Image #3]

● This is a recursive iframe issue - Builder.io is loading itself instead of your Vercel site.

Fix this:

1. Click "**Edit Model**" button on the right panel (or go to Models → cc-equipment-category)
2. In the model settings, find **Preview URL** and set it to:
<https://clg-ten.vercel.app/equipment>
3. Save the model settings
4. Go back to your content entry and set the **Page url** field (on the right) to match your route:

/equipment

4. (not [/test-cc](#))

5. Close the error dialog and refresh

The preview URL tells Builder.io where your actual site is hosted. Without it, Builder.io tries to preview itself.

> Your site is not loading as expected

Here are some suggestions to help you get connected

Verify your server is running

Check that your server at <https://clg-ten.vercel.app/equipment> is running as expected. Learn more

Verify that it matches the URL you are testing the component on

Refresh the page

Update your preview URL

<https://clg-ten.vercel.app/equipment>

This will update the preview URL for this session only. Update it permanently here.

After updating, refresh this page.

Let me check if there's a header blocking Builder.io's iframe:

Bash(curl -sI "<https://clg-ten.vercel.app/equipment?builder.preview=true>" | grep -i "x-frame\|content-security")

└ No matches found

Bash(curl -sI "<https://clg-ten.vercel.app/equipment?builder.preview=true>")

└ HTTP/2 200

age: 0

cache-control: private, no-cache, no-store, max-age=0, must-revalidate

... +13 lines (ctrl+o to expand)

The page loads fine (200 OK). The issue is Builder.io needs its SDK scripts to connect. Let me update the page to always render the [Content](#) component:

Read(src/app/equipment/[[...slug]]/page.tsx)

└ Read **117** lines

Update(src/app/equipment/[[...slug]]/page.tsx)

└ Added **7** lines, removed **17** lines

63

64 return

65 main className="min-h-screen"

66 - /* Builder.io renders the page content with registered components */

67 - {content && (

68 - <Content

69 - content={content}

70 - apiKey={BUILDER_API_KEY}

71 - model="cc-equipment-category"

72 - customComponents={customComponents}

73 - />

74 -)}

```

66 +  /* Builder.io Content component - always render for visual editing to
work */}
67 +  <Content
68 +  content={content}
69 +  apiKey={BUILDER_API_KEY}
70 +  model="cc-equipment-category"
71 +  customComponents={customComponents}
72 +  />
73
76 -  /* Show placeholder in preview mode when no content exists */
77 -  {!content && isPreviewMode && (
78 -    <div className="p-8 text-center text-gray-500">
79 -      <p>Builder.io Preview Mode</p>
80 -      <p className="text-sm">Create content for this page in the Builder.io
editor</p>
81 -    </div>
82 -  )}
83 -
74  /* Attribution footer for AI comparison */
75  footer className="bg-gray-100 py-4 text-center text-sm text-gray-500"
76

```

● **Bash**(npm run build 2>&1 | tail -10 && git add
src/app/equipment\[\[...slug\]\]/page.tsx &&
GIT_COMMITTER_EMAIL="cbertozz-access@users.noreply.github.com"
GIT_COMMIT...)
 └ Route (app)

- └ /
- └ /_not-found

... +10 lines (ctrl+o to expand)

● **Bash**(npx vercel --prod --force 2>&1 | tail -15) timeout: 3m 0s
 └ Building: L f /equipment/\[...slug\]
Building: o (Static) prerendered as static content
Building: f (Dynamic) server-rendered on demand
... +12 lines (ctrl+o to expand)

● Deployed. Now in Builder.io:

1. Make sure **Page url** on the right panel is set to [/equipment](#) (matching your route)
2. Click **Refresh** in the troubleshoot dialog

If still not working, try opening the preview URL directly in a new tab to verify it loads:

<https://clg-ten.vercel.app/equipment?builder.preview=cc-equipment-category>

> Application error: a server-side exception has occurred while loading

clg-ten.vercel.app (see the
server logs for more information).

Digest: 1896021804@E394

- **Bash**(npx vercel logs clg-orua94ij7-carlo-bertozzis-projects.vercel.app 2>&1 | tail -50)
 - └ Running in the background (↓ to manage)
- **Task Output** bbcbd3a
 - └ (No content)
- **Kill Shell**(Kill shell: bbcbd3a)
 - └ Shell bbcbd3a killed
- The **Content** component is likely crashing with null content. Let me fix it properly:
- **Update**(src/app/equipment/[...slug]/page.tsx)
 - └ Added 16 lines, removed 7 lines

```
63
64  return
65    main className="min-h-screen"
66 -   /* Builder.io Content component - always render for visual editing to work */
*}
67 - <Content
68 -   content={content}
69 -   apiKey={BUILDER_API_KEY}
70 -   model="cc-equipment-category"
71 -   customComponents={customComponents}
72 - />
66 +  /* Builder.io Content component */
67 +  {content ? (
68 +    <Content
69 +      content={content}
70 +      apiKey={BUILDER_API_KEY}
71 +      model="cc-equipment-category"
72 +      customComponents={customComponents}
73 +    />
74 +  ) : (
75 +    <div className="p-8 text-center">
76 +      <p className="text-gray-500">Loading Builder.io content...</p>
77 +      <p className="text-sm text-gray-400 mt-2">
78 +        Add content in Builder.io editor for path: {urlPath}
79 +      </p>
80 +    </div>
81 +  )}
82
83  /* Attribution footer for AI comparison */
84  footer className="bg-gray-100 py-4 text-center text-sm text-gray-500"
```
- **Bash**(npm run build 2>&1 | tail -10 && git add src/app/equipment/[...slug]/page.tsx && timeout: 3m 0s
GIT_COMMITTER_EMAIL="cbertozz-access@users.noreply.github.com"
GIT_COMMIT...)

└ Route (app)

 └ ◌ /

 └ ◌ /_not-found

 ... +20 lines (ctrl+o to expand)

● Bash(curl -sL "https://clg-ten.vercel.app/equipment?builder.preview=true" 2>&1 |

grep -o "Loading

 Builder|Add content|min-h-screen" | head -3)

└ min-h-screen

 Loading Builder

 Add content

● Page now loads. The issue is Builder.io needs content to exist first before visual editing works fully.

To get started:

1. In Builder.io, close the troubleshoot dialog

2. In the left panel, drag a simple **Text** or **Box** component onto the canvas

3. Click **Publish** (top right)

4. The visual editor should then connect properly

Alternatively, you can create content via API. Would you like me to create a sample page entry via the Builder.io Write API? (This requires a private API key from Builder.io settings)

> https://clg-ten.vercel.app/equipment?builder.preview=true does not load it give
ssame error

● Bash(curl -sL "https://clg-ten.vercel.app/equipment?builder.preview=true" -o
/dev/null -w "%{http_code}" &&

 echo "" && curl -sL "https://clg-ten.vercel.app/equipment...")

└ 200

 <!DOCTYPE html><html lang="en"><head><meta charset="utf-8"/><meta
name="viewport" content="width=dev
ice-width, initial-scale=1"/><link rel="stylesheet"
href="/_next/static/chunks/52fd40a3d70e6da9.css"
 ... +70 lines (ctrl+o to expand)

● I see the issue - there's still an error in the page (error digest in the HTML). The `generateMetadata` function is likely throwing. Let me fix it:

● Update(src/app/equipment/[[...slug]]/page.tsx)

└ Added 19 lines, removed 13 lines

```
94  const      =          join "/" || ""
95  const      = `/equipment${{ ?`/${` : `}`}
96
97 - const content = await fetchOneEntry({
98 -   model: "cc-equipment-category",
99 -   apiKey: BUILDER_API_KEY,
100 -  userAttributes: {
101 -    urlPath,
102 -  },
```

```
103 - });
97 + try {
98 +   const content = await fetchOneEntry({
99 +     model: "cc-equipment-category",
100 +    apiKey: BUILDER_API_KEY,
101 +    userAttributes: {
102 +      urlPath,
103 +    },
104 +  });
105
105 - if (!content) {
106 + if (!content) {
107 +   return {
108 +     title: "Equipment | Access Group",
109 +   };
110 + }
111 +
112   return
113 +   title: content.data?.title || "Equipment | Access Group",
114 +   description: content.data?.description || "Browse our equipment range",
115 + };
116 + } catch {
117 +   return {
118     title: "Equipment | Access Group"
119
120
110 -
111 - return {
112 -   title: content.data?.title || "Equipment | Access Group",
113 -   description: content.data?.description || "Browse our equipment range",
114 - };
121
```

● Bash(npm run build 2>&1 | tail -5 && git add . &&
timeout: 3m
GIT_COMMITTER_EMAIL="cbertozz-access@users.noreply.github.com"
GIT_COMMITTER_NAME="Carlo Os
Bertozzi" git -c user.emai...)
└ Running...

Bash command

```
npm run build 2>&1 | tail -5 && git add . &&
```

```
GIT_COMMITTER_EMAIL="cbertozz-access@users.noreply.github.com"
GIT_COMMITTER_NAME="Carlo Bertozi" git
  -c user.email="cbertozz-access@users.noreply.github.com" -c user.name="Carlo
Bertozi" commit -m "Fix
  generateMetadata error handling" && git push origin main && npx vercel --prod
--force 2>&1 | tail -5
  Build and deploy fix
```

Do you want to proceed?

› 1. Yes

2. Yes, and don't ask again for similar commands in **/Users/carlobertozzi**

3. Type here to tell Claude what to do differently

Esc to cancel

[Restored 12 Jan 2026 at 8:18:38 am]

Last login: Mon Jan 12 08:18:36 on console

carlobertozzi@Carlos-Mac-mini ~ %