

# Instructions to Migrate Buying Buddy Setup

You are migrating an existing **Buying Buddy** real-estate IDX installation in this HighLevel Ai Studio (React + Vite + TypeScript) project from the old JavaScript-plugin method to the official [@buyingbuddy/react](#) package. Work carefully and **do not change anything until I have approved your plan.**

**Important — how Buying Buddy widgets are styled (read this before you touch any CSS):** Buying Buddy widgets render their content inside a **shadow DOM**. Each widget attaches a shadow root and renders its UI there, with fully encapsulated styles. This means the widget's CSS **cannot** leak out into my site, and my site's CSS does **not** reach inside the widget. There is therefore **no styling conflict to prevent** between Buying Buddy and my site. Do **not** rename, remap, prefix, scope, or otherwise modify any of my site's CSS, design tokens, CSS variables, Tailwind config, or theme in an attempt to "avoid conflicts" — there are none, and doing so will break my site's colors and components. Your job is to convert widgets, not to defend my styles against them.

## Step 0 — Tell me my Vibe Preview domain and my header offsets

**0a. Vibe Preview domain.** Derive my Vibe Preview domain by extracting the numeric project ID from the `vibe.filesafe.space` image URLs in `index.html` (or any other file in this project) and outputting it as `preview-[PROJECT_ID].vibepreview.com` — the literal text `preview-` followed by that project ID, then `.vibepreview.com`, **without the leading `https://`**. This is the permanent address for previewing my unpublished site. It is **optional** — I only need to add it to my Buying Buddy account if I want to preview the migrated site before publishing. Tell me clearly that it is optional and that if I plan to publish straight away I can skip it. If you cannot determine the project ID from the image URLs, tell me, and I will copy the domain from the preview URL shown in my Ai Studio editor.

**0b. Widget Header Offset.** Check whether this site has a **sticky or fixed header** (a header bar that stays pinned to the top of the screen while scrolling — typically `position: fixed` or `position: sticky` on the header/nav element).

- If it does **not** have a sticky/fixed header, tell me no header offset is needed and skip to the confirmation below.
- If it **does**, measure the rendered height of that header (in pixels) at three viewport widths and give me the three numbers to enter in my Buying Buddy dashboard:
  - **Mobile** (viewport < 768px)
  - **Tablet** (viewport 768px – 991px)
  - **Desktop** (viewport ≥ 992px)
- Report them clearly, for example: "Header Offset — Mobile: 56, Tablet: 64, Desktop: 72". If the header height is the same across sizes, give the same number for each. These go in the **Widget Header Offset** panel in my Buying Buddy account (just below the Domain Name and SSL panel).

Then ask me to confirm I have completed these in my Buying Buddy account before you continue:

- **Required:** Set my Website Type to "HighLevel Ai Studio"
- **Optional (recommended for previewing before publish):** Added the Vibe Preview domain you derived above
- **(If a header offset was needed)** entered the three offset values above

**Do not proceed to Step 1 until I confirm the Website Type is set.** (The Vibe Preview domain is optional — only needed if I want to preview before publishing. The header offset can be

entered now or after the migration — it only affects how map widgets sit under your header, not whether widgets load.)

## Step 1 — Verify the current setup (inspect only, change nothing)

Confirm the project matches the expected pattern and report what you find:

1. There is an old Buying Buddy plugin block, normally in `index.html` (in the `<head>`). It looks similar to this (values will differ):

HTML

```
<!-- Buying Buddy plugin v5.02 for HighLevel; Authorized Domain: example.com -->

<!-- ** DO NOT DEFER ** This must load at the top of the page, e.g. in the page
head. -->

<script src="https://www.mbb2.com/version3/css/theme/acid/XXXXXXXX"></script>

<script>

    var MBB = {seo : "false", data:{ acid : "XXXXXXXX" } };

    function mbbMapLoaded(){ MBB.googleMaps = true; };

</script>

<script
src="https://maps.googleapis.com/maps/api/js?callback=mbbMapLoaded&libraries=place
s&key=AIza..."></script>

<script
src="https://d2w6u17ngtanmy.cloudfront.net/scripts/my-buying-buddy.5.0.js.gz"></sc
ript>

<!-- End Buying Buddy Plugin. Copyright 2003-2025 -->
```

2. **Extract my Activation Key** from the `var MBB = {... acid : "XXXXXXXX" ...}` line — the value of `acid` is my activation key. Show it to me.
3. Find **every** widget element used anywhere in the project. There are three forms to look for:
  - **Standard widgets:** `<bb-widget data-type="..." data-filter="...">` — for each, note the file, the `data-type` value, and the full `data-filter` value (if present).
  - **Legacy widgets:** `<div id="MBBv3_...">` — for each, note the file and the `id` value. These are an older form of the same Buying Buddy widgets and **will be converted** (see Step 5).
  - **Possible invalid/foreign widgets:** elements like `<div className="bb-widget" data-bb-widget="idx" data-bb-id="...">` — these use a different attribute structure (`data-bb-id`) and are **not** valid Buying Buddy widgets. Note the file for each. These will **not** be converted; you will list them for me to review manually (see Step 5).
4. Confirm whether a Disclaimer widget is present in the site footer (either as `<bb-widget data-type="Disclaimer">` or `<div id="MBBv3_Disclaimer">`). **If it is missing, you will need to add it in Step 5.**

5. Confirm whether the site has the **two foundation pages**:
  - A **Search Results** page — uses `<bb-widget data-type="SearchResults">` or `<div id="MBBv3_ListingResults">` (both forms map to the same `ListingResults` widget).
  - A **Property Details** page — uses `<bb-widget data-type="SearchDetails">` or `<div id="MBBv3_SearchDetails">`.

For each one you find, note its file path and route path. **Explicitly call out either foundation page that is missing — you will need to create it in Step 5.**

## Step 2 — Count and categorize

Report:

- The total number of files/pages that contain widgets, and the total number of widgets overall — counting both standard `<bb-widget>` elements and legacy `<div id="MBBv3_...">` elements (both will be converted).
- A separate list of any files containing **possible invalid/foreign widgets** (`data-bb-id` style) that you will **not** convert and are flagging for manual review.
- Which pages are the **foundation pages** — the Search Results page (uses `ListingResults`) and the Property Details page (uses `SearchDetails`) — versus ordinary **content pages** that contain one or more widgets mixed in with page content (for example, community/neighborhood pages with `MarketStats` and `InteractiveMap`).

## Step 3 — Summarize your plan and ask for my approval

Before changing anything, give me a clear summary of exactly what you will do (the steps in Step 5), including:

- That you will remove the entire old plugin block, **install** `@buyingbuddy/react`, **wrap** the app root in the provider, and **convert** all `<bb-widget>` elements to `<BuyingBuddyWidget>` components.
- The count of pages and widgets you will change.
- **Any foundation pages or the Disclaimer widget that you will need to CREATE because they are missing** (from your Step 1 findings), and where you will create them.

Also tell me clearly:

- **"I will not publish your site. After the migration you should preview it on your Vibe Preview domain — where widgets will now work because you authorized that domain — and publish it yourself only when you are happy."**
- **"If anything is wrong, you can revert by restoring the backup created in the next step."**

Then **stop and wait for my explicit approval** ("yes, proceed") before doing anything.

## Step 4 — Create a backup

After I approve, create a backup / restore point / version snapshot of the current site so I can roll back if needed, and tell me you have done so. If you are **not able** to create a backup yourself, stop and tell me to create one manually (or publish a restore point) before you continue — and wait for me to confirm.

## Step 5 — Execute the migration

Make these changes:

1. **Remove the old plugin block entirely** from `index.html` (and anywhere else it appears): all of the Buying Buddy comment markers, the `mbb2.com/.../acid/...` theme `<script>`, the `var MBB = {...}/mbbMapLoaded()` config `<script>`, and the `my-buying-buddy.5.0.js.gz` `<script>`.
  - The block also includes a **Google Maps API `<script>`** (`maps.googleapis.com/.../js?...key=...`). Remove it as part of the Buying Buddy block, **but flag this to me** — if I use that same Google Maps API key for any other map on my site, I need to know it was removed.
2. **Install** the npm package `@buyingbuddy/react`.
3. **Wrap the whole app** in the Buying Buddy provider at the app root (`src/main.tsx` or `src/App.tsx`), using the activation key you extracted in Step 1:

None

```
import { BuyingBuddyProvider } from "@buyingbuddy/react"

<BuyingBuddyProvider activationKey="YOUR_ACTIVATION_KEY">
  { /* the rest of the app */ }
</BuyingBuddyProvider>
```

4. **Convert the widgets** to the React component `<BuyingBuddyWidget>`, following these rules exactly.

**Standard widgets** (`<bb-widget ...>`):

- `data-type="X"` becomes `type="X"`.
- `data-filter="..."` becomes `filter="..."`. **Preserve the filter value byte-for-byte** — do not reformat, reorder, re-encode, "tidy", or change any character. The `+`-joined tokens, colons, commas, spaces, decimals, and negative numbers must all stay exactly as they were.

**Legacy widgets** (`<div id="MBBv3_...">`):

- The widget type is the value of `id="..."` with the `MBBv3_` prefix removed. For example, `<div id="MBBv3_ListingResults">` becomes `<BuyingBuddyWidget type="ListingResults" />`, and `<div id="MBBv3_Calculator">` becomes `<BuyingBuddyWidget type="Calculator" />`.
- If the element also has a `data-filter="..."` or `filter="..."` attribute, migrate its value to `filter="..."` — again, **byte-for-byte**, with no changes.

**For all converted widgets:**

- Add `import { BuyingBuddyWidget } from "@buyingbuddy/react"` to every file that now uses the component.

- **Leave all surrounding page content exactly as it is.** Only the widget elements change — the text, layout, and everything else on each page must stay untouched.

**Do NOT convert invalid/foreign widgets.** Elements like `<div className="bb-widget" data-bb-widget="idx" data-bb-id="...">` use a different attribute structure (`data-bb-id`) and are **not** Buying Buddy widgets. **Leave them exactly as they are**, and list the files that contain them in your final summary, marked for my manual review. Do not guess a mapping for them.

Example conversion:

HTML

`<!-- before -->`

```
<bb-widget data-type="MarketStats"
data-filter="reportType:totals+mls_id:md257+listing_status:active,coming-soon,under-contract,sold+sub_area:Fitler Square+prop_img:true+state:pa"></bb-widget>
```

None

`// after`

```
<BuyingBuddyWidget type="MarketStats"
filter="reportType:totals+mls_id:md257+listing_status:active,coming-soon,under-contract,sold+sub_area:Fitler Square+prop_img:true+state:pa" />
```

5. **Foundation pages.** Make sure the site has BOTH foundation pages, matching the install prompt's pattern. On each foundation page, the widget is the **only** Buying Buddy / listing component — do not hand-build any listing UI around it.
  - **Search Results page** (`<BuyingBuddyWidget type="ListingResults" />`):
    - **If it exists** (converted from the old `<bb-widget data-type="SearchResults">` or `<div id="MBBv3_ListingResults">`): make its page content conform to the pattern below.
    - **If it does not exist:** create a new route for it. Use the slug `/listing-results` unless you have clear evidence of a different slug already in use.
    - Render it inside the existing site layout (header, navigation, footer). Add a page heading — an `<h1>` such as "Search Results" — inside a normal constrained container. Immediately below the heading, place the widget inside a FULL-WIDTH container with standard responsive horizontal padding so it does not bleed to the screen edges (e.g. `<div className="w-full px-4 sm:px-6 lg:px-8">`).
  - **Property Details page** (`<BuyingBuddyWidget type="SearchDetails" />`):
    - **If it exists** (converted from the old `<bb-widget data-type="SearchDetails">` or `<div id="MBBv3_SearchDetails">`): make its page content conform to the pattern below.

- **If it does not exist:** create a new route for it. Use the slug `/listing-details` unless you have clear evidence of a different slug already in use.
  - Render it inside the existing site layout the same way, with an `<h1>` such as "Property Details" inside a normal constrained container, and the widget placed immediately below it within that SAME standard constrained container (NOT full-width — keep it constrained).
  - On all other (content) pages, convert each widget in place and keep the page's content around it.
6. **Disclaimer.** The MLS disclaimer must appear in the site footer so it shows on every page.
- **If a disclaimer widget exists** (converted from `<bb-widget data-type="Disclaimer">` or `<div id="MBBv3_Disclaimer">`): keep it in the footer.
  - **If it does not exist:** add `<BuyingBuddyWidget type="Disclaimer" />` to the site footer so it appears on every page.
7. **Verify the build and that I changed nothing else.** After making all changes, confirm the project still compiles/builds with no errors, and confirm you did **not** modify any global stylesheet, design tokens/CSS variables, Tailwind config, theme, or shared UI components. If your edits introduced any syntax or import errors, fix them before reporting. Do not report the migration as complete until the build succeeds and the only changes are widget conversions plus removal of the old plugin block.

## Step 6 — Report and hand back to me

**Do not publish the site.** When finished, give me a summary that includes:

- How many widgets you converted and in how many files (note how many were standard `<bb-widget>` and how many were legacy `<div id="MBBv3_...">`).
- Whether either foundation page (Search Results, Property Details) or the Disclaimer widget had to be CREATED because it was missing (vs. converted from an existing widget) — if so, where you placed them and what slugs/routes you used.
- Any files containing **invalid/foreign widgets** (`data-bb-id` style) that you left unchanged and are flagging for my manual review. If there were none, say so.
- Confirmation that the old plugin block (and the Google Maps loader) was removed, with the Google Maps note repeated.
- That the package is installed and the provider wraps the app root.
- Confirmation that the project builds with no errors.
- Confirmation that you did **not** change any of my site's styling, design tokens, Tailwind config, theme, or shared UI components.
- A reminder if a header offset was needed: **"If you haven't yet, enter the Widget Header Offset values I gave you (Step 0) in your Buying Buddy account so map widgets aren't hidden under your sticky header."**
- Clear next steps for me: **"Preview your site on your Vibe Preview domain — the widgets will now load because you authorized that domain. When everything looks right, publish your site. If anything is wrong, restore the backup from Step 4."**

## Rules for the whole job

- Do **not** change any routing, navigation, styling, or other code unrelated to this migration.
- Do **not** touch my site's styling or design system. Specifically: do **not** edit, rename, remap, prefix, or reorganize any CSS variables/design tokens (e.g. `--primary`,

--secondary), `src/index.css` (or any global stylesheet), `tailwind.config.*`, theme files, or shared layout/header/button/UI components. Buying Buddy widgets are shadow-DOM isolated, so there is never a reason to modify my styles "to avoid conflicts." The **only** style-related change you may make is removing the old plugin's own theme `<script>/<link>` as part of removing the old plugin block (Step 5).

- Do **not** alter any `data-filter` / `filter` values.
- Do **not** convert invalid/foreign (`data-bb-id`) widgets — leave them and list them for manual review.
- Do **not** publish the site — I will preview and publish myself.
- If you are unsure about anything, stop and ask me rather than guessing.