

The most complex app I've ever built,
I built in 3.5 days.

This is how I LIFT



THE OLD WAY

I'm looking at [[X DOCS LINK]]. What does this mean? [QUOTE SOMETHING DIRECTLY]

Example 01

EA i'm following this intro to react router tutorial:
<https://reactrouter.com/tutorials/address-book>

what does this mean?

URL segments, layouts, and data are more often than not coupled (tripled?) together
can see it in this app already:

URL SegmentComponentData/ <App> list of contacts
contacts/:contactId <Contact> individual contact

Because of this natural coupling, React Router has data conventions to get data in route components easily.

Great question! Let me visualize this for you because it's a really important concept.

The "Triples" Relationship - Think of it like a restaurant:

Imagine each URL in your app is like a **different section of a restaurant**, and they naturally need three things together:

Restaurant Section (URL)	→	Staff Needed (Component)	→	Menu/Info They Need (Data)
-----------------------------	---	-----------------------------	---	-------------------------------

Here's the pattern visualized:

/ (lobby)	→	<App>	→	[List of all tables]
/table/5	→	<TableView>	→	[Info about table 5]
/table/5/order/3	→	<OrderView>	→	[Details of order 3]

In your address book app:

URL Component Data

THE SHIFT

DOES MY MENTAL MODEL MATCH REALITY?

I've read [X] docs [[LINK TO DOCS]], my goal is [X], before we implement, can you test my mental model?

Here's what I think is happening: [YOUR UNDERSTANDING]. Is this correct? If not, explain why—avoid code examples, focus on concepts.

my goal for this morning is to implement this feature:

- * Get the AI to use a single tool, perhaps to get the weather
- * based on these docs: <https://ai-sdk.dev/docs/ai-sdk-core/tools-and-tools>

can you test my mental model first? after that i'd like you to plainly outline the need to take to implement this feature. i don't want you to show me the exp just help me work through the steps of a concept so i can slowly build at the first in an easily digestable manner.

Great to see you tackling tool calling! Let me test your mental model first, then we'll go down the implementation steps.

Testing Your Mental Model:

I notice you're thinking about this as a `/weather` command that users type - like a command in Slack or Discord. Let me ask you a few questions to clarify:

- 1. Who decides when to call the tool?** In your current mental model, you have the user explicitly typing `/weather`. But based on the Vercel AI SDK docs, who actually makes the decision about whether to call a tool - the user or the AI model itself?
- 2. What triggers tool execution?** You described a parser that extracts the tool name from user input. But think about this: if a user just asks "What's the weather like today", will the AI be able to call the weather tool? Or do they *have* to type `/weather`?
- 3. What's the AI's role?** In your flow, the chat app sends an intent directly to the AI. But what does the AI model (like Claude or GPT) do in this process? Is it just a passive responder, or does it actively participate in deciding when and how to use tools?

Take a moment to think about these questions. Your answers will help us figure out how to adjust your mental model before diving into implementation steps.

THE OUTPUT

You're right about [X] and [Y], but you're treating [Z] as critical when it's just an example. The actual critical piece is [W].

EA if my mental model is correct, we would receive the chat in `chat.tsx`, convert it in `chat.tsx` then pass it to `ai.ts`

[CONCEPT]

Good thinking, but let's trace through where the conversion actually needs to happen.

Think about it this way:

- `chat.tsx` uses `useChat()` which **creates and manages** UI messages
- `useChat()` automatically sends those messages to your server action
- Your `ai.ts` action **receives** those UI messages and needs to pass them to `streamText()`

The **mismatch happens on the server side** - you're receiving UI messages but `streamText()` needs Model messages.

Here's a key question: **Can you change what format `useChat()` sends messages in?**

No - `useChat()` is a library hook that always sends UI messages. That's its job.

So where does the conversion need to happen?

Look at your action in `ai.ts`:

```
typescript
const { messages } = await request.json() // UI messages arrive here
const result = streamText({
  model: google("gemini-2.5-flash"),
  messages // Wrong format being passed here!
})
```

The conversion needs to happen **between** receiving the messages and passing them to `streamText()`.

THE RESULTS

With this approach, I implemented:



React Router
Supabase DB
Drizzle ORM



Better Auth
Vercel AI SDK

I DIDN'T JUST BUILD
AN APP—
I LEARNED HOW TO
LEARN 10X FASTER.

KNOWLEDGE GAINED
COMPOUNDS FOREVER.

