The Practical Guide to Building LiveChat Apps: Plan out the app development process

Get to know the app building plan

Starting out is hard, and we know it. When you don't have a mentor or even simply a good guide to follow along, wandering from one resource to another and filtering the value of all incoming information is an engaging and energy-draining process.

So welcome to the Practical Guide to Building LiveChat Apps, where we will tackle building apps with the LiveChat Developer Program from A to Z — extensively, easily, but most importantly, efficiently.

In this video, we'll show you how to prepare for coding an app, and teach you how to plan out the general development process. It's an essential part of the whole app-building journey, but at the same time, not an easy one to do well. So, knowing all the good practices and tricks will help you to avoid falling into coding pitfalls along the way. And this is the knowledge we will help you to obtain in this video.

So, you already know that building an app is usually a multiple-step process. And it's very important to come up with an execution plan. We divided this course into steps easy for you to follow along. And here's what we're gonna do from start to finish:

1. First, we will configure the environment.
2. Second, we will fetch the repository with the sample app and add a functional method to it.
3. Then, we will deploy our app to one of the most popular hostings.
4. And, finally, we will prepare ourselves and our application for the app review.

And this is the plan for our journey for the entire period of this course. Additionally, throughout the episodes, here and there we'll put extra tips and good practices worth to keep in the back of your head for both prepping and during coding.

Technologies you can build with at LiveChat

So before we start, it's also good to note that our APIs allow you to build with many different technologies; pretty much the sky is the limit. So you can build in a language you're most comfortable with.

For instance, our training app is built with React. And to give you an image of the languages we use in our available SDKs, among others, there's JavaScript, Go, Python, Typescript, and node.js.

So, as long as we know that, first things first — we need to configure our environment.

Prepare the environment and fetch the training app repository

In this course, we'll be working in Visual Studio Code. To keep our code neat and clean, we use the Prettier extension. Also, you need to make sure you've installed npm and node on your machine. It's also a good practice to check the prerequisites of the tools you're personally using before you start working as well.

Once you have all that ready, let's move forward.

As mentioned before, the sample app we'll be working on in our videos is built with React, and you'll find your training code under the Tag Master LiveChat repository.
We put the link to the repository in the resources file as well. So our sample app has a simple frontend design and an outline of the authorization implemented already.
Before you move on to the next episode, make sure you fetch this repository to your machine so we can kick off smoothly with coding.

Tools useful for app development

Naturally, you should know there's also a variety of other resources you can use when building your app, which we used for our work as well.

To not list all of the internal tools separately, you will find all necessary links to what we will use during building, but also the tools that we already used to prepare our sample app in our resources file. That includes our docs, repositories, and others.

But to really perfect your application, we recommend using external tools as well. They can really elevate your workflow and productivity, especially the tools you'll use for creating your code architecture. We really can't express enough how a well thought out planning process pays off later; when you're in the depths of your application's code.

So, in our resources file, we put together a list of great apps, extensions, and sites that we use in our programming work on a daily basis. They should be an enormous help for you to really polish your application's vibe and feel. So make sure you check them out.

Good practices before you start your work

But besides the tools, you'll have to put some thought into the building process itself as well. In the end — you'll be the very head of this operation.

So, we talked to our dev team to list some good practices to smooth out your development process and make your life generally easier as a developer. And here's what they said:

Plan your code architecture ahead and start coding only when you've extended this topic. Indeed — you'll spend some more time on planning — but you won't have to rewrite your code over and over when you actually start working on it. It can be helpful to use tools like Swagger UI or Draw.io. They make it so much easier to link the connections between endpoints and help you to really think about how to wrap up your app's general flow.

The next thing is - write tests! It's rewarding to see your code running smoothly, of course! But your mood can be so easily changed when you can't find the bug that's making it all go wonky. You might even consider a development process called test-driven development — that is when you write tests even before you start coding. It might feel odd at first, but it will help you to remember your code design should be cohesive and reusable.

Our next piece of advice would be that when you're a beginner, remember that it's easier to deliver a couple of small features than one, but a huge one. With that, you'll also be able to set smaller and reachable goals for yourself. And it's always a great feeling to check something off your to-do list!

We also advise that if you get stuck, reach out to a community of sorts. Try not to spend more than half a day on debugging. With that, we really recommend joining our Discord for Developers, where you'll find helpful advice on working with the LiveChat APIs specifically.

And if you really want to take your craft up a notch, you can also have a look at the SOLID principle, and follow its fundamentals.

The last thing, but definitely not the least important one. Remember — the documentation is the developers' ultimate best friend.

Outro

So, with all that knowledge, we hope you're hyped for what's coming next. We're moving straight to the most challenging part of the process — coding!
In the next video, we will implement authorization to our training app and add a functional method to it using the 3.5 version of the Configuration API.
If you want to make sure you've prepped all the required resources for the training app, check the to-do list for the first episode in our resources file.

And, see you in the next one!