 Reference > Live Reloading

Live Reloading

When building your app it can be useful to view your changes in a web browser as you make them. FastHTML supports live reloading which means that it watches for any changes to your code and automatically refreshes the webpage in your browser.

To enable live reloading simply replace [FastHTML](#) in your app with [FastHTMLWithLiveReload](#).

```
from fasthtml.common import *  
app = FastHTMLWithLiveReload()
```

Then in your terminal run [uvicorn](#) with reloading enabled.

```
uvicorn main:app --reload
```

⚠ Gotchas - A reload is only triggered when you save your changes. - [FastHTMLWithLiveReload](#) should only be used during development. - If your app spans multiple directories you might need to use the [--reload-dir](#) flag to watch all files in each directory. See the [uvicorn docs](#) for more info. - The live reload script is only injected into the page when rendering [ft components](#).

Live reloading with [fast_app](#)

In development the [fast_app](#) function provides the same functionality. It instantiates the [FastHTMLWithLiveReload](#) class if you pass [live=True](#):


```
main.py  
  
from fasthtml.common import *  
  
app, rt = fast_app(live=True)  
  
serve()
```

①

②

- ① [fast_app\(\)](#) instantiates the [FastHTMLWithLiveReload](#) class.
- ② [serve\(\)](#) is a wrapper around a [uvicorn](#) call.

To run [main.py](#) in live reload mode, just do [python main.py](#). We recommend turning off live reload when deploying your app to production.

 Report an issue