

Tauri

- [Basics](#)
- [Vue Invoke](#)

Basics

Tauri is a framework for building desktop applications using the programming language [Rust](#). It uses Rust for the core of the application, but gives you the ability to define the user interface using JS/TS based frameworks like Vue or React.

Install and create a project

To install Tauri and create a new project, you need to have Rust installed. Afterwards run the following commands:

```
cargo install create-tauri-app --locked
cargo create-tauri-app
```

You can find alternative ways to install Tauri [in the Tauri documentation](#).

You will get asked the following questions:

- The name of your project directory
- An Identifier (e.g. org.example.my-app)
- Language of your Frontend (Rust, TypeScript / JavaScript, .NET)
- With JS/TS: Your package manager (pnpm, yarn, npm, bun)
- With Rust: Your UI Template (Vanilla, Yew, Leptos, Sycamore)
- With JS/TS: Your UI Template (Vanilla, Vue, Svelte, React, Solid, Angular, Preact)
- With .NET: Your UI Template (Blazor)

Start the development server

The development server will automatically build and run the application. Frontend changes will be updated using hot reload. Rust changes will cause the application to be build and restarted.

```
# Choose depending on your package manager choice

# npm
cd your-project
npm install
```

```
npm run tauri dev
```

```
# or accordingly
```

```
yarn tauri dev
```

```
pnpm tauri dev
```

```
deno task tauri dev
```

```
bun tauri dev
```

```
cargo tauri dev
```

Vue Invoke

When writing the user interface, you need to invoke the functions written in rust using the "invoke" function in vue.

Example code for a rust function:

```
#[tauri::command]
fn my_function() -> String {
    "Hello world!".to_string()
}

#[cfg_attr(mobile, tauri::mobile_entry_point)]
pub fn run() {
    tauri::Builder::default()
        .plugin(tauri_plugin_opener::init())
        .invoke_handler(tauri::generate_handler![my_function]) // Your function is registered
here
        .run(tauri::generate_context!())
        .expect("error while running tauri application");
}
```

Now to invoke the function, you need to import the helper function from '@tauri-apps/api/core'.

```
<script setup lang="ts">
import { invoke } from '@tauri-apps/api/core'

async function onClick() {
    const message = await invoke('my_function');
    console.log(message);
}
</script>
```

You add as first parameter the name of your function. The function returns a promise with the return value, in this case a string, so you need to await the response.