

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.

Revision #1

Created 2026-02-15 23:18:34 UTC by Matthias

Updated 2026-02-15 23:23:57 UTC by Matthias