

# Getting Started

## Install Rust and Wasm toolchain

Follow [these instructions](#) for setting up Rust.

To install Rust on Linux or MacOS, use the following command:

```
curl --proto '=https' --tlsv1.2 https://sh.rustup.rs -sSf | sh
```

source HOME/.cargo/env Then, add thewasm32-unknown-unknown toolchain. This toolchain is required because the contracts that we will build will be compiled to [Wasm](#) to run on the NEAR blockchain.

```
rustup target add wasm32-unknown-unknown
```

## Create a new project

The best way to create a new NEAR app connected with a frontend is through [create-near-app](#) . When initializing the project, be sure to include the--contract=rust flag to use the Rust SDK. Add--frontend=react to use react. Default is vanilla HTML.

npx create-near-app my-project --contract rust --frontend none --tests rust If you only wish to develop and deploy a Rust contract, the [status message example](#) is great to use as a template or through [cargo-generate](#) .

To initialize a new project with cargo-generate , run the following commands:

```
cargo install cargo-generate --features vendored-openssl
```

```
cargo generate --git https://github.com/near-examples/rust-status-message --name my-project cd my-project
```

If you would like to generate a new crate manually with cargo new --lib , make sure you include the following configuration in the generated Cargo.toml :

```
[dependencies] near-sdk = "4.0.0"
```

```
[lib] crate-type = ["cdylib"]
```

```
[profile.release] codegen-units = 1
```

## Tell rustc to optimize for small code size.

```
opt-level = "z" lto = true debug = false panic = "abort"
```

## Opt into extra safety checks on arithmetic operations <https://stackoverflow.com/a/64136471/249801>

overflow-checks = true [Edit this page](#) Last updated on Feb 1, 2023 by Damián Parrino Was this page helpful? Yes No

[Previous](#) [About Rust SDK](#) [Next](#) [near\\_bindgen](#)