

config

config (crates.io) is a library for building configuration structs from sources like env variables, yaml files and so on. You can combine it with [validator](#) to build a full configuration for your application.

Install

```
# Install config
cargo add config
cargo add serde --features derive

# Optional libraries I recommend
cargo add dotenv
cargo add validator --features derive
```

Example

In this example, I have created a struct, which holds `base_url` and `port`. `base_url` is mandatory while `port` has a default value. I can load it using a config file, or using environment variables.

I also included `dotenv`, so the environment variables are optionally read using an `.env` file.

```
use config::{Environment, File};
use dotenv::dotenv;
use serde::Deserialize;
use validator::Validate;

use crate::errors::AppError;

#[derive(Debug, Clone, Deserialize, Validate)]
pub struct Config {
    #[validate(length(min = 1))]
    pub base_url: String,
    #[validate(range(min = 1, max = 65535))]
    pub port: u16,
}
```

```
    pub port: u16,  
  }  
  
pub fn load() -> Result<Config, AppError> {  
  dotenv().ok();  
  
  let settings = config::Config::builder()  
    .set_default("port", 8080)?  
    .add_source(File::with_name("config").required(false))  
    .add_source(Environment::default())  
    .build()?;  
  
  let config: Config = settings.try_deserialize()?;  
  config  
    .validate()  
    .map_err(|err| AppError::ConfigValidation(err))?;  
  
  Ok(config)  
}
```

Revision #1

Created 2026-04-12 19:47:34 UTC by Matthias

Updated 2026-04-12 19:53:57 UTC by Matthias