

```
use async_trait::async_trait;
use loco_rs::{
    app::{AppContext, Hooks},
    boot::{create_app, BootResult, StartMode},
    controller::AppRoutes,
    task::Tasks,
    worker::Processor,
    Result,
};

use crate::controllers;

pub struct App;
#[async_trait]
impl Hooks for App {
    fn app_name() -> &'static str {
        env!("CARGO_CRATE_NAME")
    }

    fn app_version() -> String {
        format!(
            "{} ({})",
            env!("CARGO_PKG_VERSION"),
            option_env!("BUILD_SHA")
                .or(option_env!("GITHUB_SHA"))
                .unwrap_or("dev")
        )
    }

    async fn boot(mode: StartMode, environment: &str) -> Result<BootResult> {
        create_app::<Self>(mode, environment).await
    }

    fn routes() -> AppRoutes {
        AppRoutes::empty()
            .add_route(controllers::guide::routes())
            .prefix("/api")
            .add_route(controllers::home::routes())
            .add_route(controllers::home2::routes())
    }

    fn connect_workers<'a>(_p: &'a mut Processor, _ctx: &'a AppContext) {}

    fn register_tasks(_tasks: &mut Tasks) {}
}
```