```
    pub fn to_properties(&self) -> Vec<ConfigProperty> {
        vec![
            ConfigProperty::UpdateMap(
               self.icp_slam_config.updating_map),
            ConfigProperty::UseCollisions(
               self.planner_config.use_collisions),
            ConfigProperty::UseDebug(self.debug_config.enabled),
            ConfigProperty::ReactionTimeout(Duration::from_millis
               (
                self.planner_config.idle_timeout_ms,
            )),
            ConfigProperty::RobotParams(
                self.planner_config.robot.clone().into(),
            ),
            ConfigProperty::CollisionDistance(
                self.planner_config.collision_distance,
            ),
            ConfigProperty::PlannerParams((&self.planner_config)
               .into()),
            ConfigProperty::StuckParams(
               self.planner_config.stuck.clone()),
            ConfigProperty::MatchingParams(
               self.matching_config.clone()),
            #[cfg(feature = "localization")]
            ConfigProperty::SensorParams(
               self.devices_config.clone()),
            ConfigProperty::MetricsParams(
               self.planner_config.metrics.clone()),
            #[cfg(feature = "localization")]
            ConfigProperty::NoiseParams(
               self.estimation_config.noise_params()),
            #[cfg(not(feature = "localization"))]
            ConfigProperty::NoiseParams(Default::default()),
            #[cfg(feature = "gps")]
            ConfigProperty::MapParams((&self.world_map_config)
               .into()),
            #[cfg(feature = "localization")]
            ConfigProperty::LocalizationParams(
               self.estimation_config.params()),
            #[cfg(not(feature = "localization"))]
            ConfigProperty::LocalizationParams(Default::default()
               ),
        ]
    }
}

==> ./config/world_map.rs <==
use std::time::Duration;

use roboq_types::model::map::MapParams;

#[derive(Clone, serde::Serialize, serde::Deserialize, o2o::o2o)]
#[from(MapParams)]
#[into(MapParams)]
```