# Model Structure

Table of Contents

[Model Description 1](#_Toc151124585)

[Model.jl 1](#_Toc151124586)

[Initialization 1](#_Toc151124587)

[Movement 1](#_Toc151124588)

[Predation 1](#_Toc151124589)

[Energy Allocation 1](#_Toc151124590)

[Mortality 1](#_Toc151124591)

## model.jl

* Calls all other files to create functions and existing packages
  + Packages
    - PlanktonIndividuals
      * .Grids
      * .Architectures: device, Architecture, GPU, CPU, rng\_type, array\_type
    - KernelAbstractions
      * @kernel
      * @index
    - Random
    - CSV
    - DataFrames
    - StructArrays
    - JLD2
  + Files
    - *utilities.jl*
    - *create.jl*
    - *diagnostics.jl*
    - *simulation.jl*
    - *update.jl*
    - *output.jl*
    - *movement.jl*
    - *predation.jl*
    - *mortality.jl*
    - *energy.jl*
    - *timestep.jl*
* User sets working directory and loads in *trait, state,* and *grid* .csv files
* State variables are assigned and allocated appropriate datatype
* Output grid is created using *RectilinearGrid()* function that can be found in the PlanktonIndividuals package
  + Uses 3-D resolution and boundary coordinates from *grid.csv*
* Individuals are created using the *generate\_individuals()* function that can be found in the *create.jl* file.
* Model object is created using the state variables, grid, and mutable struct of *individuals*.
* Simulation parameters are stored as a mutable struct using the *simulation()* function
* Output parameters are stored in the *MarineOutputWriter* mutable struct
* Contains the *update!()* function that progresses the model run

## utilities.jl

* Contains mutable structs *plankton, individuals,* and *MarineModel* to store information