FB-OSMOSE Bridge Documentation

# **Table of Contents**

[**Table of Contents** 1](#_Toc511826726)

[**Source Code Location** 2](#_Toc511826727)

[**Code Repository** 2](#_Toc511826728)

[**Development Flow Diagram** 3](#_Toc511826729)

[**Application Flow Diagram** 3](#_Toc511826730)

[**Class Diagram** 3](#_Toc511826731)

[**Fb-osmose-bridge/src/main/java/com/github/jhpoelen/fbob** 4](#_Toc511826732)

[**CORSResponseFilter.java** 4](#_Toc511826733)

[**Config.java** 5](#_Toc511826734)

[**ConfigService.java** 6](#_Toc511826735)

[**ConfigServiceUtil.java** 8](#_Toc511826736)

[**ConfigServiceV2.java** 12](#_Toc511826737)

[**ConfigUtil.java** 13](#_Toc511826738)

[**Group.java** 32](#_Toc511826739)

[**GroupType.java** 33](#_Toc511826740)

[**GroupTypeConstants.java** 33](#_Toc511826741)

[**LtlBiomassUtil.java** 34](#_Toc511826742)

[**Main.java** 37](#_Toc511826743)

[**Msg.java** 39](#_Toc511826744)

[**OsmoseProperty.java** 39](#_Toc511826745)

[**Ping.java** 40](#_Toc511826746)

[**PropertyMapper.java** 41](#_Toc511826747)

[**PropertyMapping.java** 41](#_Toc511826748)

[**StreamFactory.java** 41](#_Toc511826749)

[**StreamFactorylmpl.java** 42](#_Toc511826750)

[**Taxon.java** 43](#_Toc511826751)

[**TraitFinder.java** 44](#_Toc511826752)

[**TraitFinderFishBaseAPI.java** 45](#_Toc511826753)

[**ValidateConfig.java** 51](#_Toc511826754)

[**ValidateConfigV2.java** 52](#_Toc511826755)

[**ValueFactory.java** 52](#_Toc511826756)

[**ValueFactoryCalculated.java** 52](#_Toc511826757)

[**ValueFactoryDefault.java** 57](#_Toc511826758)

[**ValueFactoryFishbaseAPI.java** 58](#_Toc511826759)

[**ValueFactoryFishbaseBase.java.** 60](#_Toc511826760)

[**ValueFactoryFishbaseCache.java** 61](#_Toc511826761)

[**ValueFactoryMappingDefault.java** 66](#_Toc511826762)

[**ValueFactoryMappingDefaultsForGroup.java** 67](#_Toc511826763)

[**ValueFactoryNA.java** 68](#_Toc511826764)

[**ValueFactoryProxy.java** 68](#_Toc511826765)

[**ValueFactorySexRationConstraints.java** 69](#_Toc511826766)

[**Fb-osmose-bridge/src/main/resources/com/github/jhpoelen/fbob** 69](#_Toc511826767)

[**Osmose\_config** 69](#_Toc511826768)

[**Fishbase-mapping-phytoplankton.csv** 69](#_Toc511826769)

[**Fisbase-mapping-zooplankton.csv** 69](#_Toc511826770)

[**Fishbase-mapping.csv** 70](#_Toc511826771)

[**Logging.properties** 70](#_Toc511826772)

# **Source Code Location**

Location: <https://github.com/jhpoelen/fb-osmose-bridge>

# **Code Repository**

Two Main Repository

fb-osmose-bridge/src/main

* Holds the main files and processes

fb-osmose-bridge/src/test

* Holds the output checking and resources

# **Development Flow Diagram**



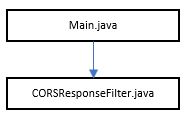
Every time a GitHub code is changed, it triggers a build in Travis-CI where it checks the code syntaxes and performs series of tests for the expected output.

# **Application Flow Diagram**

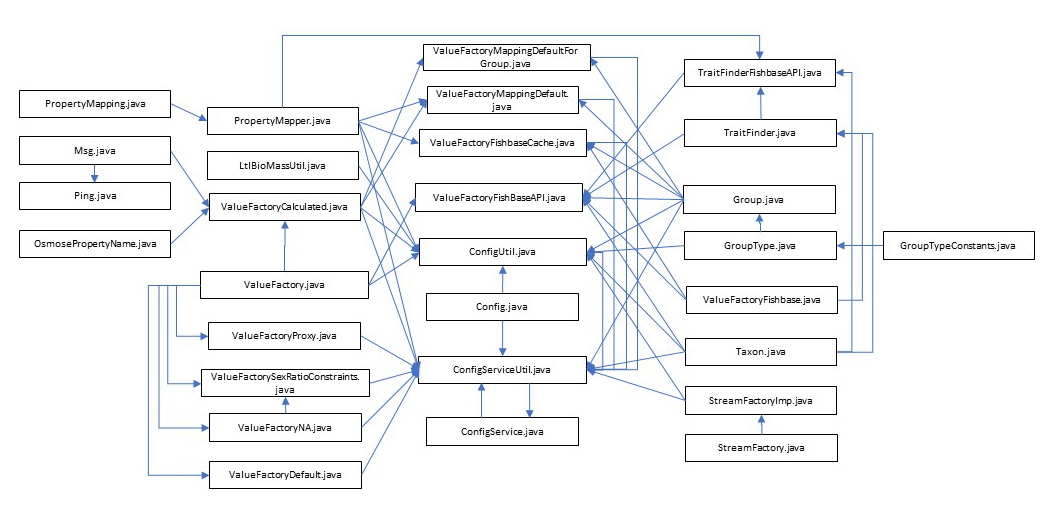


A user sets parameters and values on the FB-OSMOSE wizard then passes the values to the API where it is run on the Heroku platform then outputs the osmose configuration zip file.

# **Class Diagram**



Builds the application and setting needed for header configurations.



Data from FB-Osmose wizard will be processed. All the different functional groups, classes, species, maximum length, and habitat will be processed to form the osmose zip. Tables from FishBase and SeaLifeBase will be retrieved from the releases repository through ValueFactoryFishBaseCache.java. All data will then undergo processes and computations specific for the needed configuration for Osmose. Default values will also be set. ConfigUtil.java will then be called inside ConfigServiceUtil.java to generate the osmose configuration zip file that can be downloaded by the user.

## **Fb-osmose-bridge/src/main/java/com/github/jhpoelen/fbob**

### **CORSResponseFilter.java**



Declares the necessary import parameters for CORSResponseFilter.java. Adds Access-Control-Allow-Origin, Access-Control-Allow-Headers and Access-Control-Allow-Methods on the header for access permissions and restrictions.

### **Config.java**



Sets the value of timeStepsPerYear variable to 12 on Config class. Declares getter and setter for groups.

### **ConfigService.java**



Declares the necessary import parameters for ConfigService.java. Calls the ConfigServiceUtil.asStream method passing the list of groups and output of method ConfigServiceUtil.getValueFactory passing list of groups parameter.



Checks if a parameter “focalGroupName” was retrieved. If it has, it calls the method configArchive under ConfigServiveUtil class. If not, it creates an array list ltlGroupNames.



After that, both lists were combined to form the groups list. One is focal and one is background.

### **ConfigServiceUtil.java**



Declares the necessary import parameters for ConfigServiceUtil.java. Creates a log using java utility Logger. Sets “com.github.jhpoelen.fbob.osmose\_config” to be scanned and set the scanners instances for scanning metadata. Returns files with csv format.



Loops and checks the files on the repository, adds log information and writes the specified resource.



Method that checks the response and creates a response instance from the given parameter. Uses Content-disposition header to provide presentation of information in osmose\_config.zip.



Outputs an error message if there isn’t at least one focal or background group. Saves logs and outputs “configuration generating…”.



Accepts groupNames and group type parameter then returns groupName, type and taxon based on the given parameter.



Closes the ZIP output stream and returns the response inside the osmose\_config.zip.



Calls the class ValueFactoryFishbaseCache to get the values of the current release and patch on <https://github.com/jhpoelen/fishbase_archiver/releases>. Sets defaults values for phytoplankton and zooplankton with background group type. Calls ValueFactoryCalculated method passing the array list from the valueFactoryFishBaseCachePatch, valueFactoryCache and valueDefaults. Calls ValueFactory method to check if there is “NA” value from calling ValueFactorySexRatioConstraints method from the processed valueFactories array list.

### **ConfigServiceV2.java**



Not being used.

### **ConfigUtil.java**



Declares the necessary import parameters for ConfigUtil.java. Log messages for the class.



Sets default values for osm\_param\_output.csv



Gets the values in string format then returns values for a CSV column and writes a new line for rows with “\n”.



Calls method writeLine but passes true for Boolean parameter for creating a new line.



Checks the group count and creates reproduction.season.file.sp based on the count of group. Creates a list of paired string of months.



Creates a reproduction-seasonality-sp files based on the group size. Populates the files per group name getting the data per group name by looping and processing data based on the numberOfTimestepsPerYear variable and the month array list.



Checks if array values length is greater than zero then checks if variableSum if equals to zero. If yes, gets the quotient from 1.0 and numberOfTimestepsPerYear. If not, gets the values[timestep] array value divided by the sum of the values on the array list. Store in valueNormalized variable.



Method to format integer value to have 3 decimal places.

Method that returns reproduction-seasonality-sp + i variable based i integer parameter.



Method that accepts list of groups, factory and timeStepsPerYear variable then passing to methods generateFishingSeasonalityConfig and generateFishingSeasonalityTables.



Method that loops the group values then creates a file fishing-seasonality- plus group name inside fishing folder. Populates with data based on the formatted timeStepsPerYear variable.



Method that creates osm\_param-fishing.csv file. Populating the file by calling writeZerosFor method passing groups, mortality labels and outputstream. Also writes mortality.fishing.season.distrib.file.sp + group index and group names.



Method that accepts group names and returns file name with group name.



Method that accepts list of group names, string value for label and output stream variable and formats it with zero values.



Method that creates osm\_param-starvation.csv file with values based on the group size and adds 0.3 value.



Method that creates osm\_param-species.csv file that gets populated by values from the releases tables by calling writeParamLines method for formatting.



Method that creates osm\_param-ltl.csv that gets populated by values from the releases tables by calling writeParamLines method for formatting.



Method that formats the result by returning the list of groups, value from the tables and output stream.



Method that formats the result by returning the list of groups, label, and same two values from the releases table.



Method that formats the result and concatenates all the string values of the array list with labels and group name index.



Method that creates osm\_param-predation.csv that gets populated by values from the releases tables by calling writeParamLines and writeLine method for formatting.



Method that creates osm\_all-parameters csv that gets populated by values from the releases tables by calling writeLine method for formatting.



Method that creates osm\_param-output.csv that gets populated by values from the releases tables by calling writeLine method for formatting.



Method that creates osm\_all-natural-mortality.csv that gets populated by values from the releases tables by calling writeLine and writeParamLines method for formatting.



Method that creates osm\_all-natural-init-pop.csv that gets populated by values from the releases tables by calling writeParamLines method for formatting.



Method that calls generateFromTemplate that creates osm\_param-grid.csv and README.xlsx from static template.



Method that gets the static template from [fb-osmose-bridge](https://github.com/jhpoelen/fb-osmose-bridge/tree/a4dd22ab381bae17b92c4cb7f80b6fd3f93da76a)/[src](https://github.com/jhpoelen/fb-osmose-bridge/tree/a4dd22ab381bae17b92c4cb7f80b6fd3f93da76a/src)/[main](https://github.com/jhpoelen/fb-osmose-bridge/tree/a4dd22ab381bae17b92c4cb7f80b6fd3f93da76a/src/main)/[resources](https://github.com/jhpoelen/fb-osmose-bridge/tree/a4dd22ab381bae17b92c4cb7f80b6fd3f93da76a/src/main/resources)/[com](https://github.com/jhpoelen/fb-osmose-bridge/tree/a4dd22ab381bae17b92c4cb7f80b6fd3f93da76a/src/main/resources/com)/[github](https://github.com/jhpoelen/fb-osmose-bridge/tree/a4dd22ab381bae17b92c4cb7f80b6fd3f93da76a/src/main/resources/com/github)/[jhpoelen](https://github.com/jhpoelen/fb-osmose-bridge/tree/a4dd22ab381bae17b92c4cb7f80b6fd3f93da76a/src/main/resources/com/github/jhpoelen)/[fbob](https://github.com/jhpoelen/fb-osmose-bridge/tree/a4dd22ab381bae17b92c4cb7f80b6fd3f93da76a/src/main/resources/com/github/jhpoelen/fbob)/ osmose\_config repository.



Method that creates grid-mask.csv that gets its template to the osmose\_config repository. Calls generateMovementConfig and generateMovementMapTemplates method.



Method that gets its template from osmose\_config/maps/Amberjacks\_1.csv



Method that creates osm\_param-movement.csv file that gets populated by values from the releases tables by calling writeParamLines method for formatting.



Methods that are being called inside generateMovementConfig for adding string prefixes in populating the csv file created.



Method that initiates the creation of the config files filtered by group type.



Method that calls the creation of the csv files.



Method that creates functional\_groups.csv that is being populated with functional group name, functional group type, species name and species url.



Method that generates osm\_ltlbiomass.nc by calling method generateLtlBioMassNC from LtlBiomassUtil class.



Method that calls ValueFactoryProxy class passing valueFactories parameter.



Predefined constants Overlap and EcologicalRegion.



Method that creates predation-accesibility.csv. Checks the focal groups values based from the release tables checks whether the group type is for juvenile or adult.



Gets the stream values for background group and gets the group name. Calculates the overlap of the rows and columns of focal by calling method overlap. Multiplies the overlap values to 0.8 and formats it with 2 decimal places. Creates a header and writes the processed list on the csv file.



Method that determines the overlap based on the ecoregions and depthRangeMinMax. If two parameters are the same values, returns 1.0. If the it is not, it populates the ecoregions and depthRangeMinMax by calling method EcologicalRegion and depthRangeFor. Then the retrieved values will be compared by calling determineOverlap method.



Method that determines the region.



Method that determines the depthRangeMinMax.



Method that checks the depthRangeMinMax variable if it suffices the format and throws error message if it fails.



Method that checks the overlap. Calls method matchingEcoRegions and overlappingDepthRange methods for comparing the regions and depthRangeMinMax. Returns overlap values based on the constants of enum Overlap.



Method that returns true if two parameters are equal. Returns fales if not.



Method that compares the values of the depthRangeMinMax. Returns true if all the conditions were met.



Method that set in what ecology field will it get the values from the releases table.

### **Group.java**



Method that creates the getter and setter methods for taxa, name and type.

### **GroupType.java**



Declares group type constants from GroupTypeConstants.java.

### **GroupTypeConstants.java**



Declares group type constants.

### **LtlBiomassUtil.java**



Declares necessary import parameters and static variables for LtlBiomassUtil.java.



Sets values for variables Months, Latitudes, Longtitudes and LtlBiomass. Calls methods writeMonths, writeLatitudes, writeLongitudes, and writeLtlBiomass.



Loops the values of the months, ltl, lat and lng. Creates an arrayFloat based on the loop values.



Sets default values for longitudes.



Set default values for Latitudes.



Sets default values for Months.

### **Main.java**



Declares the necessary import parameters for Main.java. 

Method that sets the port to localhost:8080.



Method that runs when the server is shutting down. Catches exception and throws error message.



Method that sets the servlet mapping and registering resources.



Method that sets namespaces using hashmap class.

### **Msg.java**



Declares the necessary import parameters for Msg.java. Implements getters and setters for the class.

### **OsmoseProperty.java**



Creates constants for predation ingestion rate max and predation efficiency critical.

### **Ping.java**



Declares the necessary import parameters for Ping.java. Implements a method and calls class Msg.

### **PropertyMapper.java**



Checks the input stream parameter line by line. Sets default NA value if the length of line is greater than 3.

### **PropertyMapping.java**



Interface that accepts table name, column name, mapped name and default value that is being called in PropertyMapper class.

### **StreamFactory.java**



Interface for outputStream name.

### **StreamFactorylmpl.java**



Class that outputs the parameter name with string value baseDir.

### **Taxon.java**



Class for Taxon that declares getter and setter methods for name, url and selection criteria.

### **TraitFinder.java**



Interface that accepts taxon, fishbaseMapping, and table names. Declares collection list availableTables and findUsedTables.

### **TraitFinderFishBaseAPI.java**



Declares the necessary import parameters for TraitFinderFishbaseAPI.java. Creates a log using java utility Logger.



Declares static variable httpClient, mapProperties. Declares hashmap type speciesProperties. Checks input parameters and calls doMapping method then calls method valueFromTableResults passing tableResultMap, tableName and columnName parameter. Checks if the result is not blank then adds value to speciesProperties hashmap.



Accepts hashmap resultJsons, tableName, and columnName which is being checked if has values and returns the value for the method call.



Declares static methods for FishBase Ropensci API.



Checks whether the value should be retrieved from FishBase or SeaLifeBase and changed the query parameter accordingly.



Checks the parameter speciesName if it matches the a certain pattern then changes it the capitalization if genus or species is needed.



findTraitsStatic method that accepts taxon and fishbaseMapping.



Gets the mapping of the FishBase fields to OSMOSE.



Checks if the tableNames variable is not on the list of retrieved mappedName, it adds the tableName to the new array list.



Accepts taxon, fishbasemapping and tablenames collection list parameter. Loops the values of tablenames and gets the HttpResponse based on the URI. Saves the results in a treemap. If the checking fails, it will show an error message.



Set timeouts for socket time and connect time.



Checks the URI then retrieves the table name and throws exception error if it fails to access the URI.



Method that accepts taxon, fishbasemapping and tableNames list then calls findTraitsStatic method passing the parameters.



Checks the available tables from the URI by calling availableTablesStatic method. Throws error message if it fails to access it.



Creates a new list of table names from calling findUsedTablesStatic method.

### **ValidateConfig.java**



Not being used.

### **ValidateConfigV2.java**



Not being used.

### **ValueFactory.java**



Interface that accepts name and group.

### **ValueFactoryCalculated.java**



Declares necessary imports and variables for ValueFactoryCalculated.java



Creates a hashmap with field name and it value calculated by calling different methods.



Calculates the critical predation by dividing the value from fishbase table popqb.maintqb to default value of predation.ingestion.rate.max.sp. Checks if it has valid value by calling isValidPredationEfficiency method.



Checks if the ingestionRateParsed variable is equal to zero then logs a warning message. Checks if the maintQB value retrieved for FishBase table is greater than or equal to the ingestionRateParsed then logs a warning message. If all fails, return true value.



Validates fecundity fields RelFecundityMean and SpawningCycles to output relative fecundity.



Calculates the value of egg weight.



Calculates the sex ratio by dividing spawning.sexratiomid value to 100.



Calculates estimateAmax by calling calculateAmax method.



Checks different conditions for the values of species.to.sp, species.lInf.sp, poplw.LenghtMin, species.K.sp, species.lifespan.sp, estimate.AgeMin and estimate.AgeMax. If the first conditions which checks too, lInf, lengthMIn and k.sp fields were met, it calls method calcMax and returns the value. If the second condition which checks the lifespan, ageMin and ageMax were met, it multiplies values for lifespan by adjusted value of ageMin then divide it by ageMax. If all conditions fails, return 1.0 value.



Accepts to, Loo, LengthMin and K then calculate it based on the formula.



Checks the parameter name if it contains a specified key or checks the group list for values.



Message for failed calculation.

### **ValueFactoryDefault.java**



Sets defaults values for OSMOSE parameters.



Sets defaults values for OSMOSE parameters.

### **ValueFactoryFishbaseAPI.java**



Declares necessary import parameters and variables for ValueFactoryFishbaseAPI.java.



Instantiates the TraitFinderFishbaseAPI. Creates a list of taxon but checks first there is a taxa value. If not, it gets the group name. Creates a collection based on availableTables method of TraitFinderFIshBaseAPI. Loops the taxon values. Inside it, it creates a list of tables passing the availableNames set and calling findUsedTablesStatic method and removed all the values that are missing on the availableTables set. It then creates a map by calling findTraits method passing the taxon, getMappingInputStream and table values. It then gets the values that are missing on the availableTables set. Throws exception if something fails on retrieving the values.

### **ValueFactoryFishbaseBase.java.**



Declares methods and variables to be used for the FishBase API mapping.

### **ValueFactoryFishbaseCache.java**



Declares necessary import parameters and variables for ValueFactoryFishbaseCache.java.



Sets default cache version to v0.2.1.



Gets available tables from the releases.



Creates a list of species and changes the url with http://fishbase.org/summary/.



If checking conditions for tables were met, it then creates a URI for each table name that is needed for data retrieval. Starts a stopwatch class. Gets the values based on the spec codes by calling collectValuesForSpecCodes method, passing columnName, URI and list of spec codes. Based on the values retrieved, it will then call selectGroupValueUsingOrderedTaxonList.



Scans through the list of values of each spec codes then omits the blank values.



Retrieves the values of each spec code pass on the list from the releases tables.



Checks the URI parameter and throws error message.



Sets the URI to direct to the releases tables.



Checks the parameter name and group if both contains values. Set to null if not.



Gets the list of the available tables names from table\_names.tsv file on the version used.



Reference the mapping csv file and setter for groups.

### **ValueFactoryMappingDefault.java**



Declares necessary import parameters and variables for ValueFactoryMappingDefault.java



Calls populateDefaults method if the defaults variable is equal to null.



Checks the parameters passed if it has a default value then adds the values to a tree map.

### **ValueFactoryMappingDefaultsForGroup.java**



Checks if the group type is set to have default values like zooplankton and phytoplankton.

### **ValueFactoryNA.java**



Replaces no values for NA when being called.

### **ValueFactoryProxy.java**



Loops the valueFactory values then checks if it has blank values.

### **ValueFactorySexRationConstraints.java**



Declares necessary import parameters and variables for ValueFactorySexRationConstraints.java.



Checks the species.sexration.sp values and sets defaults values if it is less than 0.1 or greater than 0.9.



Replace no values with NA.

## **Fb-osmose-bridge/src/main/resources/com/github/jhpoelen/fbob**

### **Osmose\_config**

Folder created based on the data retrieved from FIshBase tables, processed data and some default value files.

### **Fishbase-mapping-phytoplankton.csv**

Mapping of FishBase tables to OSMOSE for phytoplankton

### **Fisbase-mapping-zooplankton.csv**

Mapping of FishBase tables to OSMOSE for zooplankton

### **Fishbase-mapping.csv**

Mapping of FishBase tables to OSMOSE for species, fecundity, spawning, popqb, predation, planton, estimate, ecology, mortality and poplw.

### **Logging.properties**

Configuration for logs that is inside Procfile.