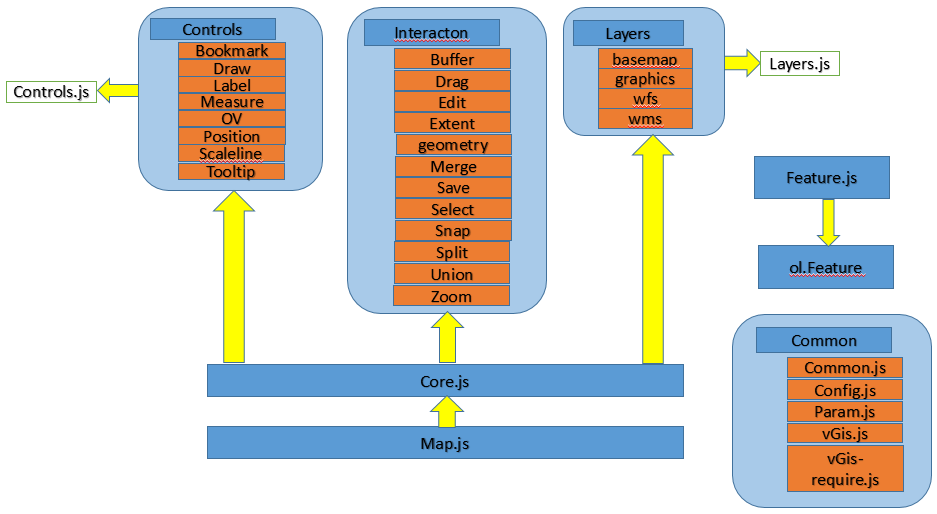
**vGIS library – Openlayer 3**

# Tổng quan



# Controls

## Bookmark

// danh sách bookmark

*this.listBookMark*

// lấy bookmark

*this.getBookMark(name)*

// tạo bookmark

*this.createBookMark*

// xóa bookmark

*this.deleteBookMark(name)*

// zoom tới vị trí bookmark

*this.zoomToExtend(extend)*

## Draw

// tạo control

this.create(geometryType, sourceVector)

// kết thúc việc vẽ

this.drawendFunction

## Label

## Measure

// tạo control

this.create

// bắt đầu vẽ lên bản đồ

this.drawStart

// kết thúc quá tình vẽ, hiển thị khoảng cách hoặc diện tích đo lường

this.drawEnd

// hủy control

this.destroy

## Overview

// tạo control

this.create

## Possition

// tạo control

this.create

## Scaleline

// tạo control

this.create

// lấy đơn vị bản đồ trên thanh scale

this.getUnit

// thiết lập đơn vị trên thanh scale

this.setUnit

## Tooltip

## Control.js

Các control trên kế thừa prototype từ Contorl.js

// contructor

function ControlClass(map) {

this.map = map;

};

// thiết lập map

ControlClass.prototype.getMap

// xác định xem map có chứa control ko?

ControlClass.prototype.isContainInteraction

// tạo control – empty function

ControlClass.prototype.create

// hủy control

ControlClass.prototype.destroy

# Interaction

## Buffer

## Drag

Kéo thả đối tượng trên bản đồ

// contructor – save: vGis.Interactions.SaveClass

app.Drag(save)

## Edit

// tạo đối tượng Edit với lớp bản đồ và kiểu geometry cho trước

this.create(layer, geomType)

// hủy đối tượng – action: nhận các giá trị: **add – update vertext**

this.destroy(action)

this.editVertext(layer)

## Extent

// tạo extent từ vị trí x,y với 1 khoảng cách cho trước tolerance

this.createExtentByPoint(x, y, tolerance)

this.fullExtent

this.backExtent

this.nextExtent

## Geometry

// tạo geometry dạng điểm

this.createPoint(x, y)

// tạo geometry dạng đường

this.createPolyline(coordinates)

// tạo geometry dạng hình tròn

this.createCircle(center, radius)

## Merge

## Save

// lưu đối tượng vào DB

// Params:

// p: nhận 3 giá trị **insert | update | delete**

// f: feature

// formatGML: định dạng GML để thiết lập sourve

// mapServiceWFS: url service WFS

this.saveFeature = function (p, f, formatGML, mapServiceWFS)

// lưu toàn bộ dữ liệu đã thay đổi vào DB

this.saveFeatures

// đưa feature cần lưu vào trong DB vào tập hợp features

// Params:

// action: nhận 3 giá trị **insert | update | delete**

// layer, feature: lớp bản đồ chứa feature cần lưu

this.pushFeature = function (action, layer, feature)

// định dạng dữ liệu lưu trữ toàn bộ feature để lưu vào DB

// định dạng:

// this.data[action][layerID].features = [];

// this.data[action][layerID].formatGML = formatGML;

// this.data[action][layerID].mapServiceWFS = params.mapService;

this.data = []

## Select

// tạo đối tượng – typeSelection: geometry type

this.create(typeSelection)

// hủy đối tượng

this.destroy

// thiết lập lớp bản đồ để select

this.setLayer(layer)

// lưu đối tượng vào tập hợp featureSelection

this.push(feature)

## Snap

// tạo đối tượng

this.create(layer)

// hủy đối tượng

this.destroy

// thiết lập khoảng cách bắt dính đối tượng

this.setTolerance(tolerance)

## Split

// tạo đối tượng

this.create

// thiết lập lớp split

this.setLayer(layer)

// spilit – location: vị trí cần cắt

this.split = function (location)

## Union

## Zoom

// tạo công cụ zoom

this.create

// phóng to bản đồ

this.zoomIn

this.fixZoomIn

// thu nhỏ bản đồ

this.zoomOut

this.fixZoomOut

// phóng tới tại vị trí x,y – polyline – polygon - multypolygon

this.zoomToPoint(x, y)

this.zoomToPolyline(geometry)

this.zoomToPolygon(geometry)

this.zoomToMultyPolygon(geometry)

# Layers

## Basemap

// tạo lớp

this.create

## Graphic Layer

// tạo lớp

this.create

## WMS Layer

// tạo lớp

this.create

## WFS Layer

// tạo lớp

this.create

## Layers.js

Các định dạng layer trên sẽ kế thừa prototype từ lớp này

// contructor

function LayerClass(map, layer, type) {

this.map = map;

this.layer = layer;

this.type = type; // vGis.Config.layerType

};

// lấy đối tượng map

LayerClass.prototype.getMap

// thiết lập map

LayerClass.prototype.setMap(map)

// lấy đối tượng layer

LayerClass.prototype.getLayer

// thiết lập đối tượng layer

LayerClass.prototype.setLayer(layer)

// ẩn/hiện lớp bản đồ

LayerClass.prototype.toogleVisible

LayerClass.prototype.getVisible

LayerClass.prototype.setVisible(visible)

// độ tỏng suốt của lớp bản đồ - opacity

LayerClass.prototype.getOpacity

LayerClass.prototype.setOpacity(opacity)

// thiết lập index của lớp trong bản đồ

LayerClass.prototype.setZIndex

// tạo lớp nền Open Street Map

LayerClass.prototype.createOSM

// tạo lớp định dạng Image WMS

LayerClass.prototype.createImageWMS(mapService, style, featureBase, featureType)

// tạo lớp định dạng Tile WMS

LayerClass.prototype.createTileWMS(mapService, style, featureBase, featureType)

// tạo lớp định dạng WFS

LayerClass.prototype.createWFS(mapService, style, featureBase, featureType, projection, geometryName, geometryType)

// tạo lớp Graphic

LayerClass.prototype.createGraphicLayer(options)

# Common

## Common.js

Các function dùng chung để thực hiện việc tính toán

## Config.js

Lưu trữ các cấu hình hệ thống, các lớp bản đồ, các thao tác với bản đồ

## Param.js

Tạo các tham số đầu vào

vGis.Params =

{

map: null, // mapClass

coreGis: null, // coreClass

baselayers: [], // danh sach cac base layer - basemapClass

wmslayers: [], // danh sach wms layer - wmsClass

wfslayers: [], // danh sach wfs layer - wfsClass

}

## vGis.js

Khai báo thư viện vGis và đưa các file js vào hệ thống

## vGis-require.js

Đưa thư viện ol và goog.require

# Core.js

Là lớp lõi của thư viện

Lớp này sẽ gọi các lớp khác để thực hiện các thao tác trên bản đồ

# Map.js

Chức đối tượng Core.js

// tạo đói tượng map

this.create

// tạo lớp bản đồ nền  
this.setBasemap = function (basemapLayer)

// đưa và gỡ bỏ các lớp bản đồ

this.addLayer(layer)

this.removeLayer(layer)

// tìm kiếm lớp bản đồ

this.findLayerWMS = function (layerID)

this.findLayerWFS = function (layerID)

// thiết lập các controls lên trên map

this.setOV

this.setScale

this.setPosition