# ARCHITETTURA

Sviluppo basato su ambiente javascript con

* Node.js
* Webpack
* Babel

Librerie principali:

* Vue
* Leaflet
* Element <http://element.eleme.io/#/en-US>
* Axios

## Componenti

### GV

Modulo: GV.js

Namespace per applicazione GeoViewer.

Definito come oggetto globale in .entry\_build.js.

Esporta i metodi principali, utilizzati dalle applicazioni:

* init
* utils
* globals
* …

Esporta l’array dei bottoni “Buttons”

### Init

Modulo: init.js

Metodo di inizializzazione della applicazione:

* inizializza la configurazione
* istanzia e ritorna istanza Vue con componente App (gv-app) nel div con id=gv-container.

Viene utilizzato da applicazioni con passaggio di oggetto di configurazione

Definito come oggetto globale **GV.init()**

### App

Modulo: vue/App.js

Componente root per applicazione GeoViewer

Definito come oggetto globale **GV.app**

### Config

Modulo: config.js

Configurazione della applicazione

Definito come oggetto globale **GV.config**

Contiene la configurazione e i metodi di accesso e impostazione.

### Globals

Modulo: globals.js

Contiene le definizioni globali di default

Definito come oggetto globale **GV.globals** in GV.js

### Util

Funzioni di utilità

Raccolte in moduli singoli nella dir src/util

Definite come oggetti globali **GV.utils** in GV.js

## BUILD

Build per produzione: npm run build

### Configurazione

E:\Tomcat6\webapps\ROOT\geoviewer2\build\webpack.prod.conf.js

### Entry Point

Entry point per build applicazione è src/index.js.

Definisce oggetto globale GV (*window.GV*)

Effettua “require” dei moduli

* CSS
* Polyfill
* Direttive
* Bottoni
* Moduli base
  + init.js
  + mountLegend.js

## Sviluppo Applicazioni

Creare cartella all’interno di pages/apps con dentro i file

* **index.html** - file html della app
* **index.css -** file css della app
* **index.js -** file js della app
  + Contiene configurazione
  + Richiama GV.init passando configurazione
* **index.dev.js -** file js per ambiente di sviluppo
  + contiene le require dei moduli
    - \_build\_entry.js
    - index.js
    - index.css

Lanciare la procedura dev.bat con argomento <app>. Es: *dev geoportale*

## Sviluppo Bottoni

Creare file js in cartella buttons

Effettuare require del js del bottone in index.js

### Bottoni leaflet

Ritornano il relativo L.control

Es: zoom.js

### Bottoni Vue

Bottoni che contengono un componente Vue

Estendono L.Control e su onAdd creano un div che contiene il componente Vue

Effettuano import async del componente Vue e impostano la configurazione che ne determinerà il caricamento alla instanziazione del bottone.

## Workflow

Applicazioni chiamano GV.init passando oggetto di configurazione

GV.init istanzia componente <gv-app> e fa rendering in div #gv-container

Componente <gv-app>

* crea oggetto globale GV.app=this (se stesso)
* richiama config.set passando oggetto di configurazione (prop options di <gv-app>)

# NOTE

## Definizione Componenti VUE

Definire i componenti Vue nel componente **App**, altrimenti se compilo una applicazione che utilizza una parte dei componenti la libreria mi viene compilata senza quegli elementi.

## Gestione ordinamento layer

<http://leafletjs.com/reference-1.0.0.html#map-createpane>

<https://esri.github.io/esri-leaflet/examples/layer-ordering.html>

<http://leafletjs.com/examples/map-panes.html>

<http://gis.stackexchange.com/questions/178622/z-index-of-geojsons-and-imageoverlays-in-leaflet>

<http://fiddle.jshell.net/nathansnider/4gmqf0ug/>

Memorizzare variabile lastZIndex su oggetto GV.Map inizializzare a 11

Calcolare zIndex del livello in RlLayer.initialize con GV.app.map.lastZIndex + 1

Salvare nuovo in GV.app.map.lastZIndex

Ogni livello in un suo pane: utilizzare

map.createPane(<layer>);

map.getPane(<layer>).style.zIndex = <zIndex>;

Indicare nome pane nelle options di creazione del layer

## GESTIONE CSS

css/libs/ contiene le librerie css (bootstrap ecc.....)

css/fonts/ contiene i font (cambiare path in css se necessario)

css/images/ contiene le immagini

css/base.css stili di base (comprende leaflet)

**compilazione**

tutto quello che è dentro libs/ viene unificato in libs.css

base.css + libs.css viene unificato in gv.css

gv.css viene minificato in gv.min.css

pagine includono gv.min.css

## Filtro livelli GeoJSON

Creata funzione setFilter in LayerFactory

Prende in input filters: array di condizioni di filtro

Se filters=null viene annullato filtro

TODO: creare layer GV.geoJson che estende L.geoJson e implementa gestione filtro

## Cluster

Utilizzato: <https://github.com/Leaflet/Leaflet.markercluster>

Implementato in LayerFactory per layer di tipo JSON.

Impostare proprietà *cluster* della configurazione del layer

cluster: true

oppure

cluster: {

     options: {

          opt1: ...

     }

}

Per opzioni vedi <https://github.com/Leaflet/Leaflet.markercluster#using-the-plugin>

Per customizzare Icona vedi <https://github.com/Leaflet/Leaflet.markercluster#customising-the-clustered-markers>

## Non Tiled MWS

<https://github.com/Leaflet/Leaflet/issues/558>

Implementata gestione per wms non tiled, basata su:

<https://github.com/ptv-logistics/Leaflet.NonTiledLayer>

Altre implementazioni

<https://github.com/heigeo/leaflet.wms>(anche getFeatureInfo)

<https://gist.github.com/Zverik/5757305>

<https://gist.github.com/fnicollet/5764080>

## Gestione evento click su mappa

Impostabile in configurazione con parametro 'click' delle mapOptions

Finora gestito 'info' = getFeatureInfo in GV.App che chiama activate che registra evento e associa a request.

## INFO

Implementata funzione di info.

Disabilitata su schermi piccoli

## DRAG PANEL

Utilizzata libreria interact.js

<http://interactjs.io/>

<https://github.com/taye/interact.js>

## Eventi Mappa Leaflet

In componente Map.vue per ogni evento di L.map viene emesso un evento su componente root (GV.app)

Vedi <https://github.com/KoRiGaN/Vue2Leaflet>

# DOCUMENTAZIONE LIBRERIE

<http://leafletjs.com/reference-1.0.0.html>

<http://mapsk.in/>

<http://vuejs.org/guide/>

<http://vuejs.org/api/>

<http://element.eleme.io/#/en-US>

# LEAFLET – PLUGIN VARI

<https://github.com/mapbox/mapbox-gl-leaflet>

<https://github.com/aratcliffe/Leaflet.contextmenu>

<https://github.com/davidchouse/Leaflet.NavBar>

<https://github.com/domoritz/leaflet-locatecontrol>

<https://github.com/bambrikii/leaflet-layer-tree-plugin>

<https://github.com/Leaflet/Leaflet.heat>

<https://github.com/MrMufflon/Leaflet.Elevation>

<http://www.liedman.net/leaflet-routing-machine/tutorials/>

<https://github.com/nationalparkservice/Leaflet.Draw>

<https://github.com/ebrelsford/Leaflet.jsonstyles>

<https://github.com/mapbox/leaflet-omnivore> (CSV/GPX/KML/WKT/TopoJSON) NO JSONP - by MapBox)

<https://github.com/gregallensworth/L.Control.BootstrapModal>

<https://github.com/makinacorpus/Leaflet.Spin>

<https://github.com/turbo87/sidebar-v2/>

<https://github.com/Leaflet/Leaflet.fullscreen/tree/gh-pages/dist>

<https://github.com/naomap/leaflet-fusesearch>

<http://humangeo.github.io/leaflet-dvf/>

<https://github.com/TolonUK/Leaflet.EdgeBuffer>

<https://github.com/digidem/leaflet-side-by-side>

<https://github.com/SpatialServer/Leaflet.MapboxVectorTile>

<https://github.com/mukil/Leaflet.annotate>

<https://github.com/Turbo87/sidebar-v2/>

<https://github.com/Geovation/labelgun/blob/master/README.md> LabelGun

<https://pnmcartodesign.wordpress.com/2017/05/02/leaflet-geojson-feature-search/>

<https://bl.ocks.org/zross> Esempi con D3

<http://muxlab.github.io/map-effects-100/Leaflet/22_bootstrap-tooltip.html> (bootstrap tooltip)

<http://muxlab.github.io/map-effects-100/Leaflet/16_responsive-popup.html> Responsive  popup

<http://muxlab.github.io/map-effects-100/> Effetti

<https://github.com/jwasilgeo/Leaflet.Canvas-Flowmap-Layer> Flowmap (flusso tra oggetti)

# ICONE / FONT

<https://www.mapbox.com/maki-icons/>

<https://github.com/mapbox/maki/tree/master/icons>

<http://mapsk.in/>

<https://github.com/lvoogdt/Leaflet.awesome-markers>

<http://ionicons.com/>

# OL3/CESIUM

Versione ol3-cesium

<http://openlayers.org/ol3-cesium/>

<https://github.com/openlayers/ol3-cesium>

<https://map.geo.admin.ch/>

<https://vimeo.com/142114617>

<https://vimeo.com/112935177>

<link rel="stylesheet" href="ol.css" type="text/css">  
<script src="Cesium/Cesium.js"></script>  
<script src="ol3cesium.js"></script>

var ol3d = new olcs.OLCesium({map: map}); // map is the ol.Map instance

var scene = ol3d.getCesiumScene();

scene.terrainProvider = new Cesium.CesiumTerrainProvider({  
  url: 'https://assets.agi.com/stk-terrain/world'  
});

ol3d.setEnabled(true);

Vedi anche

<http://viglino.github.io/ol3-ext/>

<https://astuntechnology.github.io/osgis-ol3-leaflet/index.html>