

Geomajas Client GWT 2.0 - TODOs

- 1 General remarks
- 2 GWT2 Client
 - 2.1 GWT2-API
 - 2.2 GWT2-Impl
 - 2.3 GWT2-Server-Extension
- 3 Plugins
 - 3.1 Plugin: Core Widgets
 - Current Situation
 - Layers and their legends
 - Proposal
 - 3.2 Plugin: Editing
 - 3.3 Plugin: Geocoder
 - 3.4 Plugin: Print
 - 3.5 Plugin: Rasterizing
 - 3.6 Plugin: WMS
 - 3.7 Plugin: Graphics Editing

1 General remarks



- @since declaration needs to be 2.0.0 everywhere! It also needs to be removed from any methods, but only appear on class/interface definitions.
- Make sure all widgets abide by whatever is decided here: [GWT widget definitions](#)
- Fix JIRA.... Perhaps this needs a more detailed explanation. We should create a new JIRA project (GWT2) and then see which issues to copy to it (from various projects).
- **Documentation!!**







What?	Who?
Touch Support	Dosi
GWT API/Impl	Pieter
Plugin: Core Widgets	Oliver + David
Plugin: Editing	?
Plugin: Geocoder	?
Plugin: Print	JanDM
Plugin: Rasterizing	Frank
Plugin: WMS	Pieter
Plugin: Graphics Editing	JanV

Who does what? Whenever in doubt: Assign to Frank



2 GWT2 Client

2.1 GWT2-API




- Should CanvasContainer and CanvasShape be a part of the API? Can it be removed? In the long run, we could replace it with some sort of client-side layer that has a Canvas based renderer. Should be added to a ContainerManager interface which groups all containers
 -  **PURE-189** - Getting issue details... STATUS
- TransformableWidgetContainer: same remark as CanvasContainer. Should be added to a ContainerManager interface which groups all containers.
 -  **PURE-189** - Getting issue details... STATUS
- Tile and TileCode: they are placed under the package gfx. Is that the correct place?

-  **PURE-190** - Getting issue details... STATUS
- Reverse the names MapConfiguration and MapOptions:
 -  **PURE-191** - Getting issue details... STATUS
- Should the MapHint system be presented through a singleton service instead of map specific?
 -  **PURE-191** - Getting issue details... STATUS
- Remove OpacitySupported? I believe the opacity changing should be a part of any layer:
 -  **PURE-192** - Getting issue details... STATUS
- Replace HasLegendWidget with a factory system?
 -  **PURE-193** - Getting issue details... STATUS
- ViewPort.isInitialized delete this method.
- MapConfiguration.setLayerAnimation => move to LayersModelRenderer:
 -  **PURE-191** - Getting issue details... STATUS

2.2 GWT2-Impl

- What is the EntryPoint for? Remove it?
 -  **PURE-197** - Getting issue details... STATUS
- The touch controllers have been disabled since the animated navigation. They should be adapted and tested again. Otherwise we should remove them.
 -  **PURE-198** - Getting issue details... STATUS



2.3 GWT2-Server-Extension

- Add setInstance method to GeomajasServerExtension
 -  **PURE-196** - Getting issue details... STATUS
- GeomajasServerExtension.createLayer should return a ServerLayer instance.
 - Done
- Rename MapOptionsExt to MapConfigurationExt?
 -  **PURE-191** - Getting issue details... STATUS
- Do we really need an ExceptionDialog widget here?
 -  **PURE-194** - Getting issue details... STATUS

3 Plugins

3.1 Plugin: Core Widgets

Current Situation

- CloseableDialogBox, MessageBox, ToolTipBox: Should move to YAWL.
 -  **PURE-203** - Getting issue details... STATUS
- FeatureSelectBox: Follow widget structure.
 -  **PURE-204** - Getting issue details... STATUS

Layers and their legends

Different layers may build their legends in different ways. Many widgets may exist that build legends for certain layers. How can the DefaultLayerPanel for example, know which widget to use? I think we need a factory system that helps determine how to build a legend for a certain layer. The default may simply build an Image widget from layers that have LegendUrlSupported. Then extra factories, stored in some registry may build legend widgets for other layer types.

To be discussed

Proposal








To be discussed: At this point I would like to propose we create the following base packages wherein all widgets should come:

- org.geomajas.gwt2.widget.client.feature: Widgets that display a single feature in some way.
- org.geomajas.gwt2.widget.client.layer: Widgets that display a single layer in some way. (layer panel showing visibility toggle & legend, table for features, ...)
- org.geomajas.gwt2.widget.client.map: Widgets that display the map (all layers) in some way.


To be discussed: For legend related widgets, I propose the following list:

- org.geomajas.gwt2.widget.client.layer.legend.RasterServerLayerLegend: A widget that displays the legend for a RasterServerLayer.
- org.geomajas.gwt2.widget.client.layer.legend.VectorServerLayerLegend: A widget that displays the legend for a VectorServerLayer. Has the option to react to map events (make itself dynamic).
- org.geomajas.gwt2.widget.client.layer.DefaultLayerPanel: A widget that displays the layer title and a checkbox to toggle visibility. It also displays a legend. (how does it know what legend widget to use???)
- org.geomajas.gwt2.widget.client.map.DefaultLayersModelPanel: A widget that displays a "DefaultLayerPanel" for each layer in the map.
- org.geomajas.gwt2.widget.client.map.DefaultLayersModelDropDown: A widget that displays a DropDown button. When opened, a "DefaultLayersModelPanel" comes out.


3.2 Plugin: Editing





- Very unclear API. Also remove all references to @FutureApi.
 -  **GEP-76** - Getting issue details... STATUS
- Revisit the GeometryEditor construct.
 -  **GEP-76** - Getting issue details... STATUS
- Split up the examples
 -  **GEP-75** - Getting issue details... STATUS
- Examples contains a GeometryToShape converter. Should that not be a part of the Geomajas client API?
 -  **PURE-199** - Getting issue details... STATUS
- What SnappingAlgorithms should be part of the API? Do we need a factory to shield the implementations from API? => Only 2: NearestEdgeSnapAlgorithm, NearestVertexSnapAlgorithm
 -  **PURE-200** - Getting issue details... STATUS
- What's the purpose of the GeometryOperationService? How is it connected to the rest? Move to geomajas-server-extension? Same with splitting and merging?
 - The problem is that we want the basic editing code to work independently from the Geomajas Server. Should we make this separation immediately?
 -  **PURE-202** - Getting issue details... STATUS
- Shouldn't we call this "geometry editing"? It's more to the point.
- Split the editing project in client api and server extension.
 -  **PURE-201** - Getting issue details... STATUS

3.3 Plugin: Geocoder


- Widget should follow the rules as explained on Confluence.
 -  **PURE-205** - Getting issue details... STATUS

3.4 Plugin: Print










- Is it Print or Printing? We should rename where necessary.
- There is a default template builder, but no easy way to provide your own templates. Should we not provide a template building API? Some factories perhaps.
- What is the PrintingLayout? Seems like a SmartGWT construct.
 -  **PURE-206** - Getting issue details... STATUS
- PrintPanel: should follow widget rules.

-  **PURE-206** - Getting issue details... STATUS
- PrintingMessages: Place it in a i18n folder.
 -  **PURE-207** - Getting issue details... STATUS
- Refactor Printing into a starting point for creating a print service and template etc. It should not be an EntryPoint.
 -  **PURE-208** - Getting issue details... STATUS
- Almost no @Api annotations (only in the PrintingLayout).
- There are no examples yet.
 -  **PURE-209** - Getting issue details... STATUS


3.5 Plugin: Rasterizing

- I don't understand the need for this plugin. Does it need to override the .gwt.xml definitions of something else? Should this not be a part of the GWT2-ServerExtension?
 -  **PURE-210** - Getting issue details... STATUS

3.6 Plugin: WMS

- Do not let the browser cache WMS GetCapabilities requests (add no-cache headers in the request)
 -  **WMSCL-37** - Getting issue details... STATUS
- Add support for Dates in GetFeatureInfo requests (when using the Feature format)
 -  **WMSCL-38** - Getting issue details... STATUS
- Add support for dealing with BigDecimal in GetFeatureInfo requests (when using the Feature format)
 -  **WMSCL-38** - Getting issue details... STATUS
- Let the client decide if geometries returned through a GetFeatureInfoCommand should be simplified or not (and how many points should be allowed per polygon).
 -  **WMSCL-39** - Getting issue details... STATUS
- Automatic check to see if a WMS layer also has WFS support
 -  **WMSCL-40** - Getting issue details... STATUS
- Add WFS search mechanism
 -  **WMSCL-40** - Getting issue details... STATUS
- Add maximum width parameter to the legend configuration object to limit the width of legend images.
- Initial visibility setting does not work (bug)
 -  **WMSCL-41** - Getting issue details... STATUS
- Add support for style changing. The WMS layer must know what styles are available.
 -  **WMSCL-42** - Getting issue details... STATUS
- Add legend widget
 -  **WMSCL-43** - Getting issue details... STATUS
- ...and many more


3.7 Plugin: Graphics Editing


- This is the link between Graphics project and Editing plugin
- Adding plugin to GWT2 requires renaming of the projects in gwt2 project pom
=> work done, see GGEP-5 branch. This implied re-adding lost class of Editing Plugin: `org.geomajas.plugin.editing.gwt.client.GeometryEditorFactory`
 - Make sure the "Editing" class is used instead of the GeometryEditorFactory. Remove the GeometryEditorFactory.
 -  **PURE-211** - Getting issue details... STATUS

- Adding graphics editing example jar project to gwt2 example
=> some solution suggested: see GGEP-6 branch. This implied implementing changes of ViewPort in class `org.geomajas.plugin.graphicsediting.gwt2.example.client.annotation.AnnotationContainer`

- Needs to be reviewed:  **PURE-212** - Getting issue details...


- Add Documentation: create new documentation project and fill

-  **PURE-216** - Getting issue details...

- Code has been changed  **PURE-119** - Getting issue details... , with specific extras to be mentioned:

- renamed code project to `geomajas-client-gwt2-plugin-graphicsediting-impl` (added `-impl`) and directory named to `impl`
- added a new example project. Difference between the two example projects: adding editing functionality via a) object controller b) action (will be shown in cog dropdown menu)
- extended documentation

- TODO: rework the `GraphicsEditingUtil` class: is now a collection of static values of stroke/fill parameters of polygon/line on creation + a `createClickToStopEditService`.

-  **PURE-220** - Getting issue details...