

JS_framework V 1.0

Readme Document

Oct 16, 2013

CyberTech Systems and Software Ltd.



Introduction

This document provides information on how to use the new template framework describing its various components and their configuration details.

This framework is intended to be used as the foundation while creating new JavaScript/H5 templates. It implements AMD and is currently built for ArcGIS JavaScript API 3.7 and Dojo 1.9.1.

This Framework by design supports mobile, tablet and desktop browsers and has provisions ready for implementing localization.

The Framework comes with the following basic widgets:

- Locator
- Geolocation
- Basemap Switcher
- Share
- Help

The framework can be extended further by way of adding application specific widgets.

Following sections of the document will help developers

- To understand Framework Structure
- Basic configuration settings
- Steps to add & configure new widgets

Framework Folder Structure

The framework consists of the following folders and files:

- esri<TemplateName>: It is the root folder of the application. While <TemplateName> needs to be replaced with the name of template.
 - default.htm: This is the landing page which loads dojoConfiguration, bootStrapper, ArcGIS API for JavaScript and the styles used in application
 - config.js: All configuration settings for the application are contained by this file
 - coreLibrary folder: This folder contains core library files like dojoConfiguration.js, bootStrapper.js etc.
 - dojoConfiguration.js: This file initializes default dojo configuration attributes by creating a dojoConfig object

- bootStrapper.js: This file loads the application configuration settings from configuration file and invokes startup() function from widgetLoader.js to load widgets configured in the configuration file
- widgetLoader.js: This file is responsible to load widgets configured in Header Widget Settings of config file
- nls folder: This folder contains locale files for the entire application
- themes folder: This folder contains all theme element like images and css in respective sub-folders
 - images folder: This folder contains images for entire application
 - styles folder: This folder contains all the css files for all supported devices like mobile, tablet and desktop
- widgets folder: This folder contains the widgets used in the application. Each widget further has its own template folder and its JavaScript file

Configuration settings

The Configuration file (config.js) is located at root folder i.e. esri<TemplateName>.

Application level settings like Application title, icon, favicon/home screen icon can be set in this file. This file can further be used to set

- Basemap – Look for tag name ‘BaseMapLayers’
- Default extent – Look for tag name ‘DefaultExtent’
- Operational layers – Look for tag name ‘OperationalLayers’
- Locator service – Look for tag name ‘LocatorSettings’
- Sharing URLs – Look for tag name ‘MapSharingOptions’

Steps to add application specific widgets

- Add widget’s template and js files in widget folder – root_framework_folder\widgets
- Add corresponding nls files in nls folder – root_framework_folder\nls
- Add widget’s css files in theme-styles folder – root_framework_folder\themes\styles
- Add corresponding images in theme-images folder – root_framework_folder\themes\images
- Add widget’s settings in configuration file – root_framework_folder\config.js

Configuring newly added widgets

Header Widgets can be configured by changing AppHeaderWidgets tag in configuration file (root_framework_folder\config.js). Below are the mandatory settings for widget

- Title: Name of the widget, will be displayed as title of widget

- WidgetPath: Path of the widget respective to the widgets package
- MapInstanceRequired: true if widget is dependent on the map instance

Example:-

```
{  
    Title: "locator",  
    WidgetPath: "widgets/locator/locator",  
    MapInstanceRequired: true  
}
```

Non header widgets needs to be configured in relevant sections of configuration file.