|  |
| --- |
| [Type the company name] |
| Kendo Paged Application Framework |
| User Guide & API Documentation |
|  |
| **Lucas.Lopatka** |
| **1/21/2014** |

|  |
| --- |
| [Type the abstract of the document here. The abstract is typically a short summary of the contents of the document. Type the abstract of the document here. The abstract is typically a short summary of the contents of the document.] |

Contents

[What is kpaf? 5](#_Toc378321567)

[Initial configuration 5](#_Toc378321568)

[Requirements 5](#_Toc378321569)

[jQuery 1.9+ (recommended) 5](#_Toc378321570)

[Kendo UI Web 5](#_Toc378321571)

[Web Browser 5](#_Toc378321572)

[Optional components 5](#_Toc378321573)

[Adding the kpaf framework 5](#_Toc378321574)

[Configuring the bootstrap 5](#_Toc378321575)

[Accessing framework componenets 6](#_Toc378321576)

[Application structure 6](#_Toc378321577)

[Binding basics 6](#_Toc378321578)

[Component access 6](#_Toc378321579)

[Accessing pages 6](#_Toc378321580)

[Accessing modules 6](#_Toc378321581)

[Accessing event handlers 6](#_Toc378321582)

[Creating a new page 7](#_Toc378321583)

[Configuring routes 7](#_Toc378321584)

[Creating a page layout 7](#_Toc378321585)

[Adding event handlers 7](#_Toc378321586)

[Adding modules 7](#_Toc378321587)

[Adding blocks 8](#_Toc378321588)

[Advanced use 8](#_Toc378321589)

[Pages 8](#_Toc378321590)

[Adding a custom page event handler (built-in events) 8](#_Toc378321591)

[Modules 8](#_Toc378321592)

[Creating a module 8](#_Toc378321593)

[Blocks 8](#_Toc378321594)

[Creating a block 8](#_Toc378321595)

[Behaviors 8](#_Toc378321596)

[Using a predefined behavior 8](#_Toc378321597)

[Defining a new behavior 8](#_Toc378321598)

[Kendo 8](#_Toc378321599)

[Adding a popup window to a page 8](#_Toc378321600)

[Nesting a grid inside a grid 8](#_Toc378321601)

[Adding a custom popup editor to a grid 8](#_Toc378321602)

[Config 11](#_Toc378321603)

[Using custom tags 11](#_Toc378321604)

[Adding custom events 11](#_Toc378321605)

[Customizing Kendo widgets 11](#_Toc378321606)

[API 12](#_Toc378321607)

[Conventions 12](#_Toc378321608)

[Config 12](#_Toc378321609)

[Built-in “tags” 12](#_Toc378321610)

[Built-in “attributes” 12](#_Toc378321611)

[Built-in “collections” 12](#_Toc378321612)

[Events 12](#_Toc378321613)

[Components 12](#_Toc378321614)

[App 12](#_Toc378321615)

[App.Loader 16](#_Toc378321616)

[App.URL 16](#_Toc378321617)

[App.Router 16](#_Toc378321618)

[App.Routers 18](#_Toc378321619)

[App.Routers.Adapters 18](#_Toc378321620)

[App.EventHandler 18](#_Toc378321621)

[App.Observable 20](#_Toc378321622)

[App.ErrorHandler 20](#_Toc378321623)

[App.EventHandlers 21](#_Toc378321624)

[App.EventHandlers.Adapters 21](#_Toc378321625)

[App.Helpers 23](#_Toc378321626)

[App.Utilities 24](#_Toc378321627)

[App.Page 29](#_Toc378321628)

[App. Page.Layout 31](#_Toc378321629)

[App.Data 41](#_Toc378321630)

[App.Data.DataSource 42](#_Toc378321631)

[App.Data.Bindings 42](#_Toc378321632)

[App.Widgets.Helpers 42](#_Toc378321633)

[App.Widgets.Behavior 42](#_Toc378321634)

# What is kPaged?

KPaged, or the Kendo Paged Application Framework is an Adaptive Object Model (AOM) renderer for HTML5/JavaScript, designed to work seamlessly with the Kendo UI JavaScript library.

KPaged reduces the complexity of building and maintaining MVVM (Model-View-ViewModel) JavaScript applications, and leverages the Kendo UI JavaScript Framework and Widget Library.

# Initial configuration

## Requirements

### jQuery 1.9+ (recommended)

It may be possible to run the kPaged framework on an older version of jQuery, but it has not been tested…

### Kendo UI Web

### Web Browser

## Optional components

## Adding the kPaged framework

## Configuring the bootstrap



# Accessing framework componenets

## Application structure

The kPaged framework

## Binding basics

## Component access

### Accessing pages

### Accessing modules

### Accessing event handlers

# Creating a new page

## Configuring routes



## Creating a page layout

## Adding event handlers



## Adding modules to a layout

## Adding blocks to a layout

# Advanced use

## Pages

### Adding a custom page event handler (built-in events)

## Modules

### Creating a module

## Blocks

### Creating a block

## Behaviors

### Using a predefined behavior

### Defining a new behavior

## Kendo

### Adding a popup window to a page

### Nesting a grid inside a grid

### Adding a custom popup editor to a grid

Using a custom popup editor, allows us to display more fields in the editor than are shown on the grid.

I haven't found a way to get JavaScript DOM let me inject HTML generated by KendoDOMBuilder into a script tag.

Simple fix -- wrap the popup editor container div in a script tag. This masks the div object from the rendering engine, and lets us keep doing what we're doing.









## Config

### Using custom tags

### Adding custom events

Open up app.config.js and look for the App.Config namespace. To add a page event, add a new value to the pageEvent array. To add a module event, add a new value to the moduleEvents array.

### Customizing Kendo widgets



# API

## Conventions

## Config

### Built-in “tags”

#### Input elements

hidden, text, url, email, select, radio, checkbox, textarea, datepicker, yesno

#### Void elements

base, br, col, command, embed, hr, img, img, input, link, meta, param, source

### Built-in “attributes”

id, name, type, class, style, data, for, min, max, pattern, step, required, requiredIf, requiredIfEmpty, readonly, disabled, href, src, target, rel, block, rows, cols, maxlength

### Built-in “collections”

layout, blocks, panels, forms, fieldsets, fieldgroups, fields, group, modules, children

### Events

#### App

#### Pages

##### executing

##### initialized

##### loaded

##### changed

#### Modules

##### initialized

##### dataBound

##### pageLoaded

#### Blocks

## Components

### App

#### execute

##### params: none

##### usage

##### return: void

#### addPage

##### params: pageName, config

###### String pageName

###### Object config

##### usage

##### return: void

#### addBlock

##### params: blockName, config

###### String blockName

###### Object config

##### usage

##### return: void

#### addModule

##### params: moduleName, config

###### String moduleName

###### Object config

##### usage

##### return: void

#### loader

##### params: none

##### usage

##### return App.Loader

#### load

##### params: type, url, options

###### String type

###### String url

###### Object options

##### usage

##### return

#### getPages

##### params: none

##### usage

##### return ChainableHash

#### getPageNames

##### params: none

##### usage

##### return Array

#### getPage

##### params: pageName

###### String pageName

##### usage

##### return App.Page

#### setPage

##### params: pageName, instance

###### String pageName

###### App.Page instance

##### usage

##### return

#### getCurrent

##### params

##### usage

##### return App.Page

#### setCurrent

##### params: pageName, instance

###### String pageName

###### App.Page instance

##### usage

##### return self

#### getConfig

##### params: key

###### String key

##### usage

##### return Object

#### setConfig

##### params: key, value

###### String key

###### Object value

##### usage

##### return self

#### buildMenu

##### params: selector

###### String selector

##### usage

##### return

#### getRouter

##### params: none

##### usage

##### return App.Router

### App.Loader

#### getResourceEventHandler

##### params: resource

###### String resource

##### usage

##### return

#### setResourceEventHandler

##### params: resource, handler

###### String resource

###### function handler

##### usage

##### return self

#### load

##### params: type, source, options

###### String type

###### String source

###### Object options

##### usage

##### return

### App.URL

### App.Router

#### current

##### params: none

##### usage

##### return

#### routes

##### params: none

##### usage

##### return

#### add

##### params: pattern, callback

###### String pattern

###### function callback

##### usage

##### return self

#### remove

##### params: none – TODO: this cant be right

##### usage

##### return self

#### removeAll

##### params: none

##### usage

##### return self

#### parse

##### params: request

##### usage

##### return self

#### destroy

##### params: none

##### usage

##### return self

#### reset

##### params: none

##### usage

##### return self

### App.Routers

### App.Routers.Adapters

#### Kendo

#### Crossroads

#### Hasher

### App.EventHandler

#### adapter

##### params: none

##### usage

##### return Adapter | false

#### addEventListener

##### params: eventName, callback, scope

###### String eventName

###### function callback

###### String scope

##### usage

##### return

#### removeEventListener

##### params: eventName, callback, scope

###### String eventName

###### function callback

###### String scope

##### usage

##### return

#### hasEventListener

##### params: eventname

###### String eventName

##### usage

##### return

#### dispatch

##### params: eventName, target

##### usage

##### return

#### getEvents

##### params

##### usage

##### return

#### getEvent

##### params: event

##### usage

##### return

#### hasEvent

##### params: event

##### usage

##### return

### App.Observable

#### subscribe

##### params: name, subscriber

###### String name

###### Object subscriber

##### usage

##### return self

#### unsubscribe

##### params: name

###### String name

##### usage

##### return self

#### hasSubscriber

##### params: name

###### String name

##### usage

##### return

#### notify

##### params: eventName, event, data

##### usage

##### return self

### App.ErrorHandler

#### setErrors

##### params: type, data

###### String type

##### usage

##### return

#### log (doesnt do anything)

### App.EventHandlers

### App.EventHandlers.Adapters

#### Signals (works with Crossroads)

##### event

###### params

###### usage

###### return

##### adapter

###### params

###### usage

###### return

##### addEventListener

###### params

###### usage

###### return

##### setEventListener

###### params

###### usage

###### return

##### hasEventListener

###### params

###### usage

###### return

##### dispatch

###### params

###### usage

###### return

##### getEvents

###### params

###### usage

###### return

##### getEvent

###### params

###### usage

###### return

##### hasEvent

###### params

###### usage

###### return

#### EventBus – not completely implemented

##### adapter

###### params

###### usage

###### return

##### addEventListener

###### params

###### usage

###### return

##### removeEventListener

###### params

###### usage

###### return

##### hasEventListener

###### params

###### usage

###### return

##### dispatch

###### params

###### usage

###### return

##### getEvents

###### params

###### usage

###### return

### App.Helpers

#### isEmpty

##### params

##### usage

##### return

#### App.Helpers.String

##### hyphenize

###### params

###### usage

###### return

##### camelize

###### params

###### usage

###### return

#### App.Helpers.JSON (not completely implemented)

#### App.Helpers.URL (not completely implemented)

### App.Utilities

#### Stack

##### pop

###### params: none

###### usage

###### return Object

##### push

###### params: item

Object item

###### usage

###### return void

#### Queue

##### dequeue

###### params: none

###### usage

###### return Object

##### enqueue

###### params: item

Object item

###### usage

###### return void

#### HashTable

##### setItem

###### params: key, value

String key

Object value

###### usage

###### return Object | undefined

##### getItem

###### params: key

String key

###### usage

###### return Object | undefined

##### hasItem

###### params: key

String key

###### usage

###### return

##### removeItem

###### params: key

###### usage

###### return Object | undefined

##### keys

###### params: none

###### usage

###### return Array

##### values

###### params: none

###### usage

###### return Array

##### each

###### params: fn

function fn

###### usage

###### return void

##### clear

###### params: none

###### usage

###### return void

#### ChainableHash

##### *Dependency Injection container Wrapper for App.Utilities.HashTable providing a cleaner interface and supporting method chaining*

##### has

###### params: key

String key

###### usage

###### return

##### get

###### params: key

String key

###### usage

###### return Object

##### set

###### params

###### usage

###### return self

##### remove

###### params

###### usage

###### return self | false

#### Injector

##### *Note: I should probably be using a AMD/CommonJS loader instead*

##### get

###### params

###### usage

###### return

##### register

###### params

###### usage

###### return

##### unregister

###### params

###### usage

###### return

#### Iterator

##### *A basic javascript iterator*

##### next

###### params

###### usage

###### return

##### hasNext

###### params

###### usage

###### return

##### rewind

###### params

###### usage

###### return

##### current

###### params

###### usage

###### return

##### key

###### params

###### usage

###### return

#### RecursiveIterator

##### *Basic recursive iterator*

##### hasChildren

###### params

###### usage

###### return

##### getChildren

###### params

###### usage

###### return

#### createFunction (do we even use this?)

### App.Page

#### Base class for all page instances, and an ancestor class for page controllers

#### getName

##### params: none

##### usage

##### return String

#### getBlock

##### params: block

###### String block

##### usage

##### return App.Page.Layout.Block

#### setBlock

##### params: blockName, block

###### String blockName

###### App.Page.Layout.Block block

##### usage

##### return self

#### hasBlock

##### params: block

###### String block

##### usage

##### return

#### getModules

##### params: none

##### usage

##### return

#### getModule

##### params: moduleName

###### String moduleName

##### usage

##### return

#### setModule

##### params: moduleName, module

###### String moduleName

###### App.Page.Layout.Module module

##### usage

##### return self

#### getConfig

##### params: none

##### usage

##### return Object

#### getFormData

##### params: none

##### usage

##### return Object

#### getErrorHandler

##### params: none

##### usage

##### return App.ErrorHandler

#### getEventHandler

##### params: none

##### usage

##### return App.EventHandler

#### load

##### params: params, url

###### Object params

###### String url

##### usage

##### return self

#### save

##### params: event, callback

###### ? event

###### function callback

##### usage

##### return self

#### validate

##### params

##### usage

### App. Page.Layout

#### Base

#### Blocks

#### Block

##### dataBind

###### params: none

###### usage

###### return void

##### setLayout

###### params: layout

Object layout

###### usage

###### return self

##### getLayout

###### params: layout

###### usage

###### return

##### setBlocks

###### params: blocks

array | Object blocks

###### usage

###### return void

##### render

###### params: none

###### usage

###### return void

##### getId

###### params: none

###### usage

###### return String

##### html

###### params: none

###### usage

###### return String

##### getViewModel

###### params: none

###### usage

###### return kendo.ObservableObject

##### getValidator

###### params: none

###### usage

###### return kendo.Validator

#### Module

##### dataBind

###### params: none

###### usage

###### return void

##### render

###### params: none

###### usage

###### return void

##### getId

###### params: none

###### usage

###### return String

##### html

###### params: none

###### usage

###### return String

##### getViewModel

###### params: none

###### usage

###### return kendo.ObservableObject

##### getValidator

###### params: none

###### usage

###### return kendo.Validator

##### getEventHandler

###### params: none

###### usage

###### return App.EventHandler

#### BlockDirector

##### merge

###### params

###### usage

###### return

##### getPage

###### params

###### usage

###### return

##### getBlock

###### params

###### usage

###### return

##### getBinder

###### params

###### usage

###### return

##### setBinder

###### params

###### usage

###### return

##### getBuilder

###### params

###### usage

###### return

##### setBuilder

###### params

###### usage

###### return

##### isCollection

###### params

###### usage

###### return

##### isVoid

###### params

###### usage

###### return

##### build

###### params

###### usage

###### return

##### getDocument

###### params

###### usage

###### return

#### FieldMap

#### DocumentBuilder

#### DOMBuilder

##### getDocument

###### Returns the document

###### params: none

###### usage

###### return DOM DocumentFragment

##### getRoot

###### Returns the root node

###### params: none

###### usage

###### return DOM Node

##### getCurrent

###### Returns the current node

###### params: none

###### usage

###### return DOM Node

##### setCurrent

###### Sets the current node

###### params: node

DOM Node node

###### usage

###### return self

##### getParent

###### Returns the parent of the current node

###### params: none

###### usage

###### return DOM Node

##### appendNode

###### params: ref, element, attributes

DOM Node ref

String | DOM DocumentFragment element

Object attributes

###### usage

###### return DOM Node

##### addBefore

###### Creates and appends a node inside a specified parent

###### params: ref, element, attributes

DOM Node ref

String | DOM DocumentFragment element

Object attributes

###### usage

###### return DOM Node

##### addAfter

###### Creates a node and inserts it after the specified element

###### params: ref, element, attributes

DOM Node ref

String | DOM DocumentFragment element

Object attributes

###### usage

###### return DOM Node

##### append

###### Creates and appends a node inside a specified parent

###### params: element, attributes, ref

String | DOM DocumentFragment element

Object attributes

DOM Node ref

###### usage

###### return self

##### add

###### Creates a node and inserts it after the specified element

###### params: element, attributes, ref

String | DOM DocumentFragment element

Object attributes

DOM Node ref

###### usage

###### return self

##### before

###### Creates a node and inserts it before the specified element

###### params: element, attributes, ref

String | DOM DocumentFragment element

Object attributes

DOM Node ref

###### usage

###### return self

##### parent

###### Sets the internal current node reference to the parent of the current node

###### params: none

###### usage

###### return DOM Node

##### text

###### Sets the text for a specified node

###### Params: ref, value

###### usage

###### return self

##### setAttributes

###### params: node, attributes

###### usage

###### return self

#### XmlDOMBuilder

#### KendoDOMBuilder

##### getDirector

###### params: none

###### usage

###### return BlockDirector

##### getBinder

###### params: none

###### usage

###### return DOMBinder

##### setAttributes

###### params: node, attributes

###### usage

###### return void

#### DOMBinder

##### getDirector

###### params: none

###### usage

###### return BlockDirector

##### getViewModel

###### params: none

###### usage

###### return kendo.ObservableObject

##### bind

###### params: type, value

String type – TODO: Rename this to key

Object value

###### usage

###### return self

##### unbind – TODO: Not implemented

###### params: key

String key

###### usage

###### return self

#### KendoDOMBinder

##### getDirector

###### params: none

###### usage

###### return BlockDirector

##### getViewModel

###### params: none

###### usage

###### return kendo.ObservableObject

##### getValidator

###### params: none

###### usage

###### return kendo.Validator

##### bindNode

###### params: node, bindings

###### usage

###### return self

##### unbindNode

###### params: key

###### usage

###### return self

##### bind

###### params: selector

###### usage

###### return

##### bindValidation

###### params: selector, rules

###### usage

###### return self

### App.Data

### App.Data.DataSource

#### Factory

#### DataSource

#### EntityDS

#### MethodDS

#### CustomDS

#### RawDS

#### HierarchicalDataSource

#### HierarchicalEntityDS

#### HierarchicalMethodDS

#### HierarchicalCustomDS

#### HierarchicalRawDS

### App.Data.Bindings

### App.Widgets.Helpers

#### PanelBar

##### build

#### SemanticTabStrip

##### build

### App.Widgets.Behavior

#### getEvent

#### execute