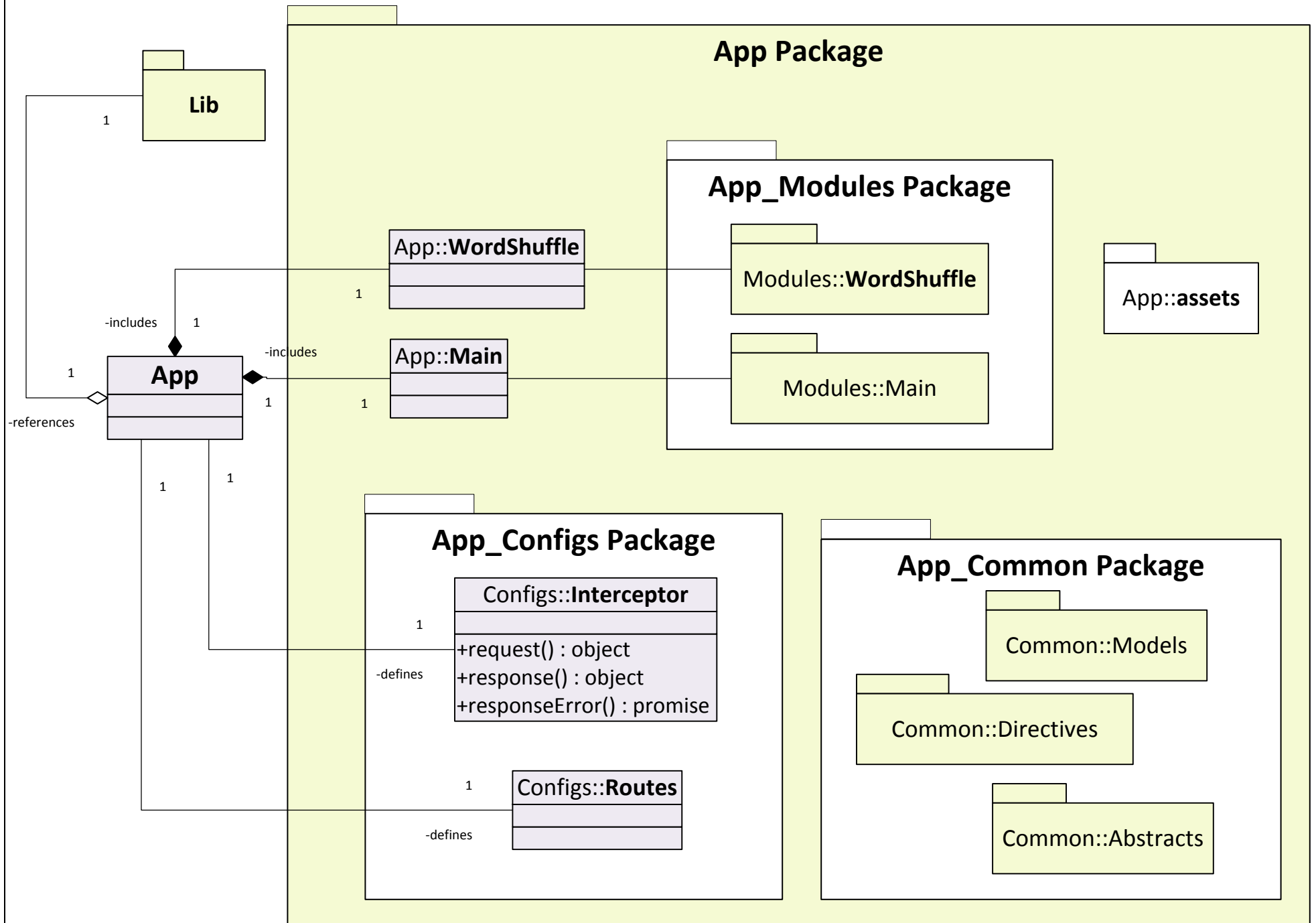
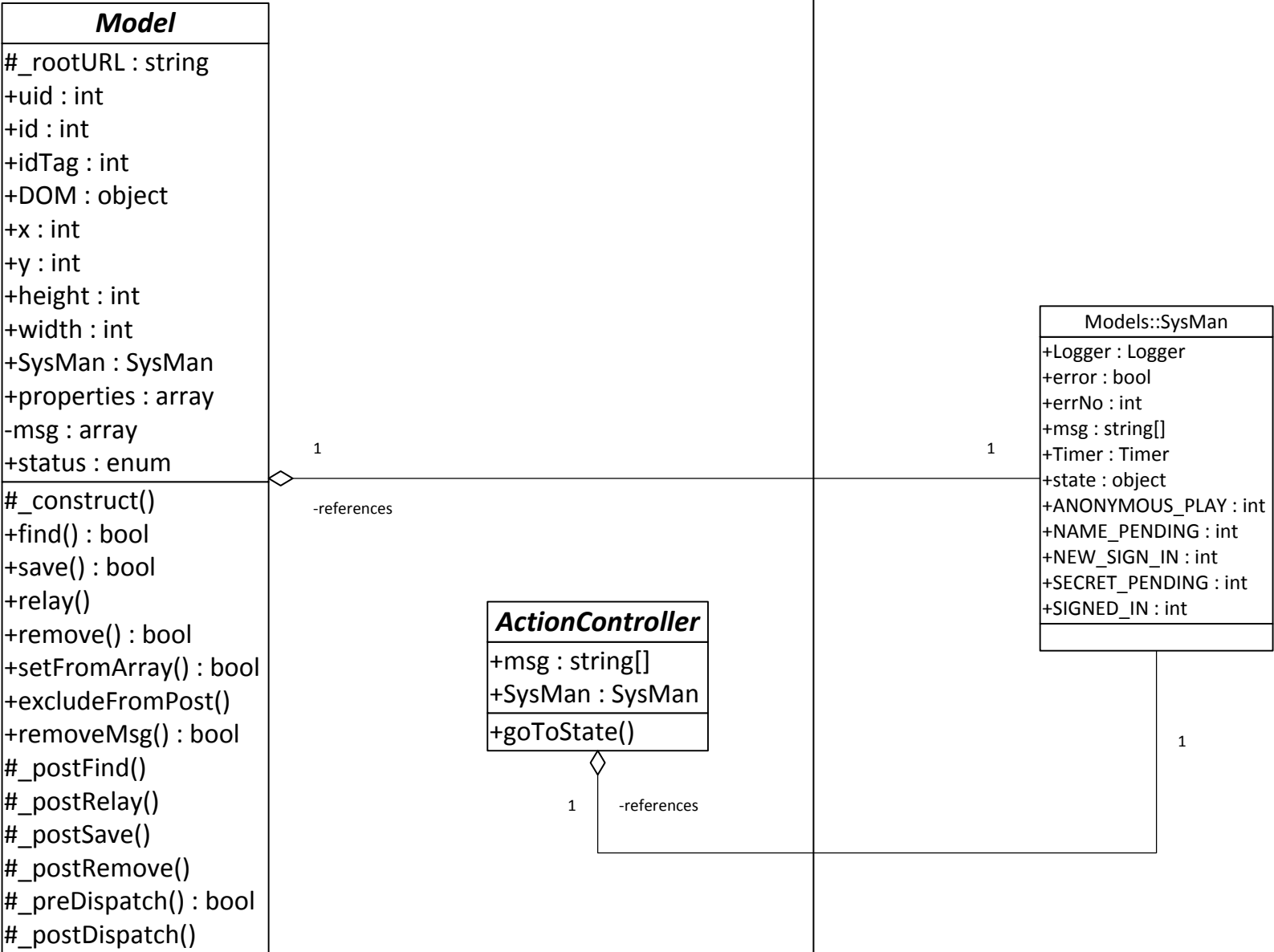


Grace Web Application



App_Common_Abstracts Package



App_Common_Models Package

App_Common_Models_Tools

Tools::Logger
+inError : bool
+stack : object[]
+errMsg : string
+mode : int
+output : string
+ERRNO : object
+TYPE : object
+MODE : object
+entry() : bool
+callOnError()

Tools::Timer
+trip : int
+isTripped : bool
+now : int
+delta : int
+passedTrips : int

SysMan

+Logger : Logger
+error : bool
+errNo : int
+msg : string[]
+Timer : Timer
+state : object
+ANONYMOUS_PLAY : int
+NAME_PENDING : int
+NEW_SIGN_IN : int
+SECRET_PENDING : int
+SIGNED_IN : int

Understanding aggregate relationships “shared” and “composite”: Logger is shared with SysMan. If SysMan deleted, Logger lives on. Timer is a component of SysMan, it dies with it.

-references

1

1

1

-can have zero or many

0..*

Message

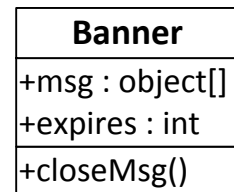
+type : string
+text : string
+TYPES : object

-has a

1

1

App_Common_Directives Package



«bind»



Banner.drv.html

«bind»

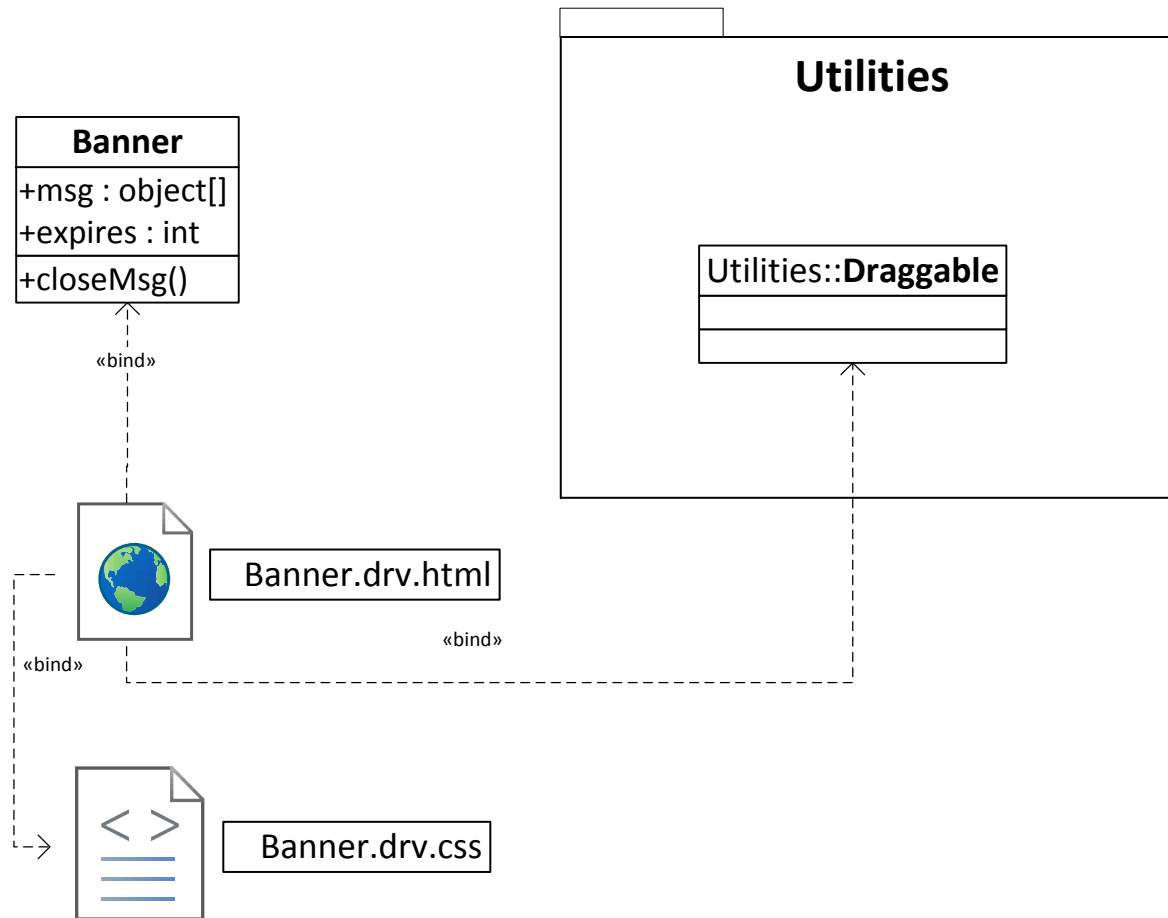
«bind»



Banner.drv.css

Utilities

Utilities::Draggable



appCommonDirectiveBanner.drv.html



msg[0].text

close

mediate icon
display with
msg[0].type

App_Modules_Main (Main Module) Package

Main_Controllers Package

Controllers::**Error**

+errorAction()

«extends»

Controllers::**Index**

+indexAction()

Abstracts::**ActionController**

+msg : string[]

+SysMan : SysMan

+goToState()

Configs

Models

Main_Views Package

layout.tpl.html

«bind»

layout.tpl.css

«bind»

Error.tpl.html

«bind»

«bind»

Error

Error.tpl.html

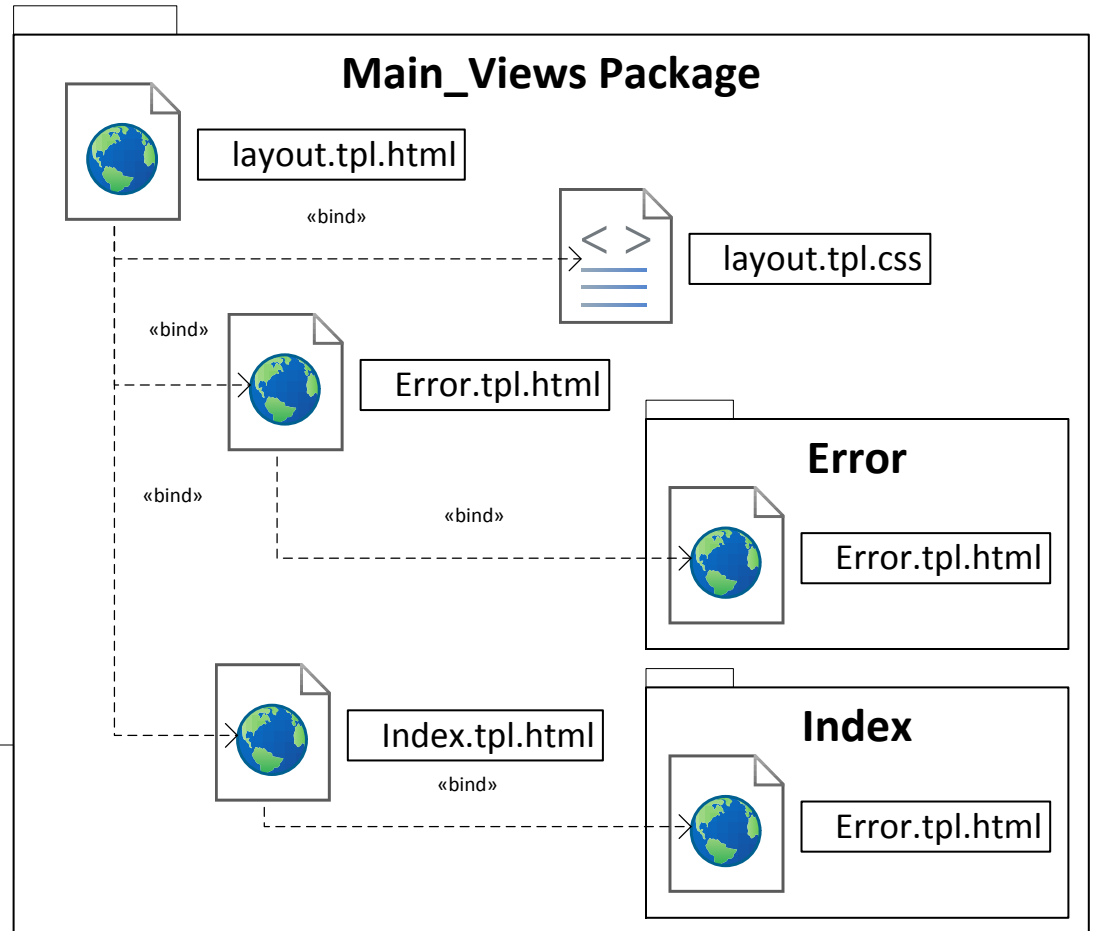
Index

Error.tpl.html

Index.tpl.html

«bind»

The "ui-view" directive binds the module layout, controller, and active templates together





gApp!

WordShuffle

Hello from Grace Application Framework! This text displays when AngularJS "ui.router" loads the Main_Views_Layout template.

ui-view

One of the templates below
depending on URL state,
specifically :controller parameter in
'/m/:module/:controller/:action'

Main_Views Package

Error.tpl.html



Index.tpl.html

Main_Views_Index.tpl.html



layout.tpl.html

«bind»



And now this text displays once "ui.router" loads the Main_View_Index template per the URL state.

ui-view

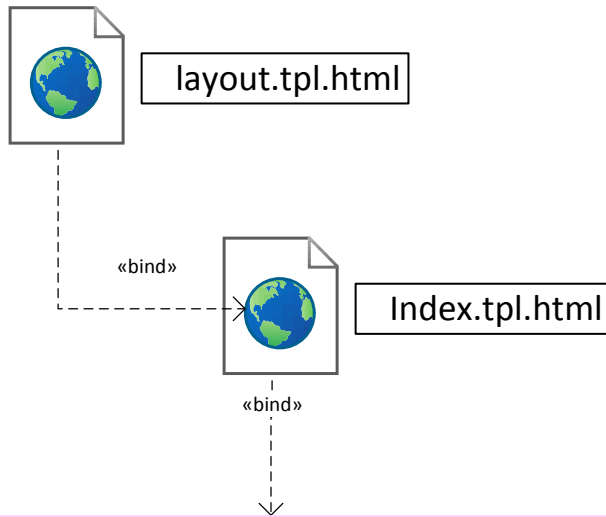
This controller only supports one action; hence only one template displays

Main_Views_Index Package



Index.tpl.html

Main_Views_Index_Index.tpl.html



Finally, "ui.router" hits the action parameter in the URL and loads this template, Main_Views_Index_Index. Roughly, over the next 2-3 months, you will build a new module within this is the Grace Health System web development framework. This process will get you familiar with all the tools that you will be using to create or modify Grace Health System web applications. We will go from the backend to the frontend, covering everything from MySQL server to AngularJS framework.



layout.tpl.html

«bind»



Main_Views_Error.tpl.html

ERROR: It looks like something went wrong with your code!

{{Error.SysMan.Logger.errMsg}}

ui-view

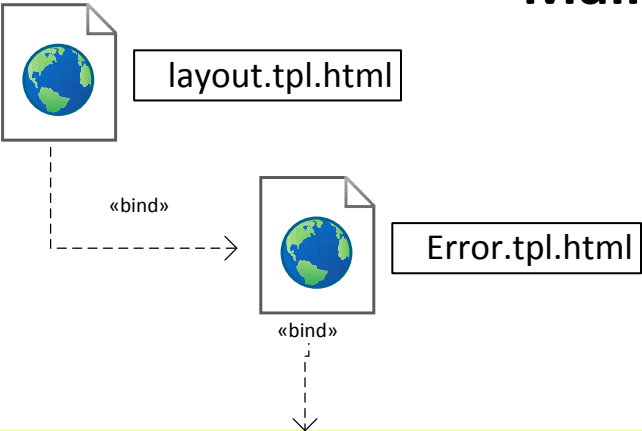
This controller only supports one action; hence only one template displays

Main_Views_Error Package



Error.tpl.html

Main_Views_Error_Error.tpl.html

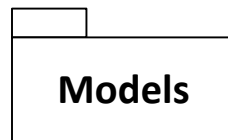
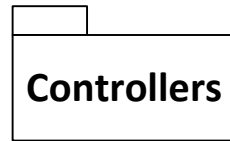


Here is the stack trace before the error:

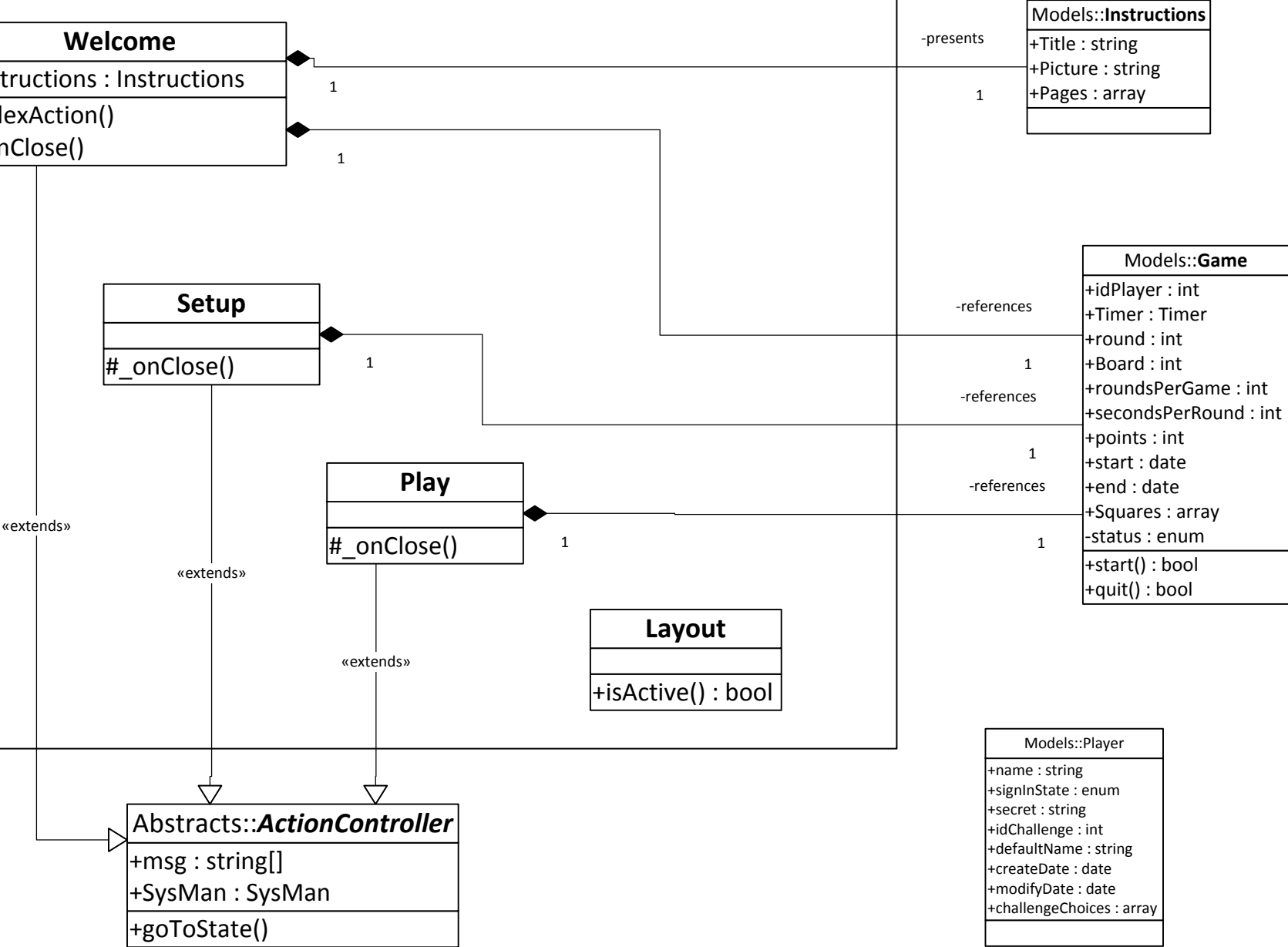
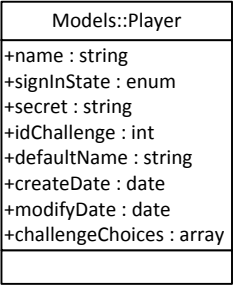
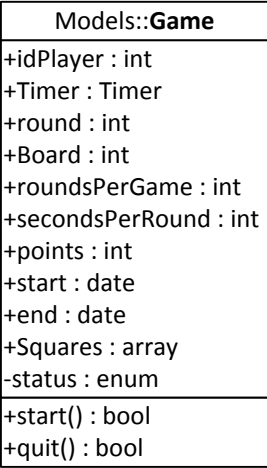
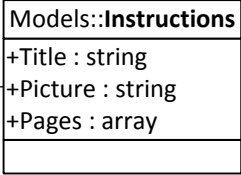
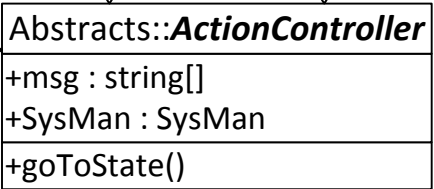
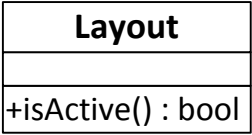
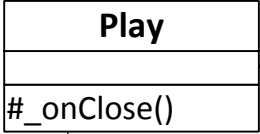
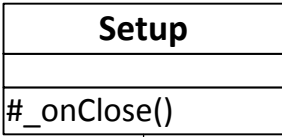
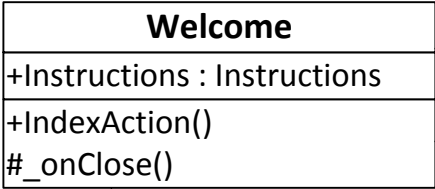
For each item in
Error.SysMan.Logger.stack

| DateTime | Type | Source | Text |
|---------------|---------------|---------------|---------------|
| <<item.date>> | <<item.type>> | <<item.type>> | <<item.text>> |
| <<item.date>> | <<item.type>> | <<item.type>> | <<item.text>> |
| <<item.date>> | <<item.type>> | <<item.type>> | <<item.text>> |

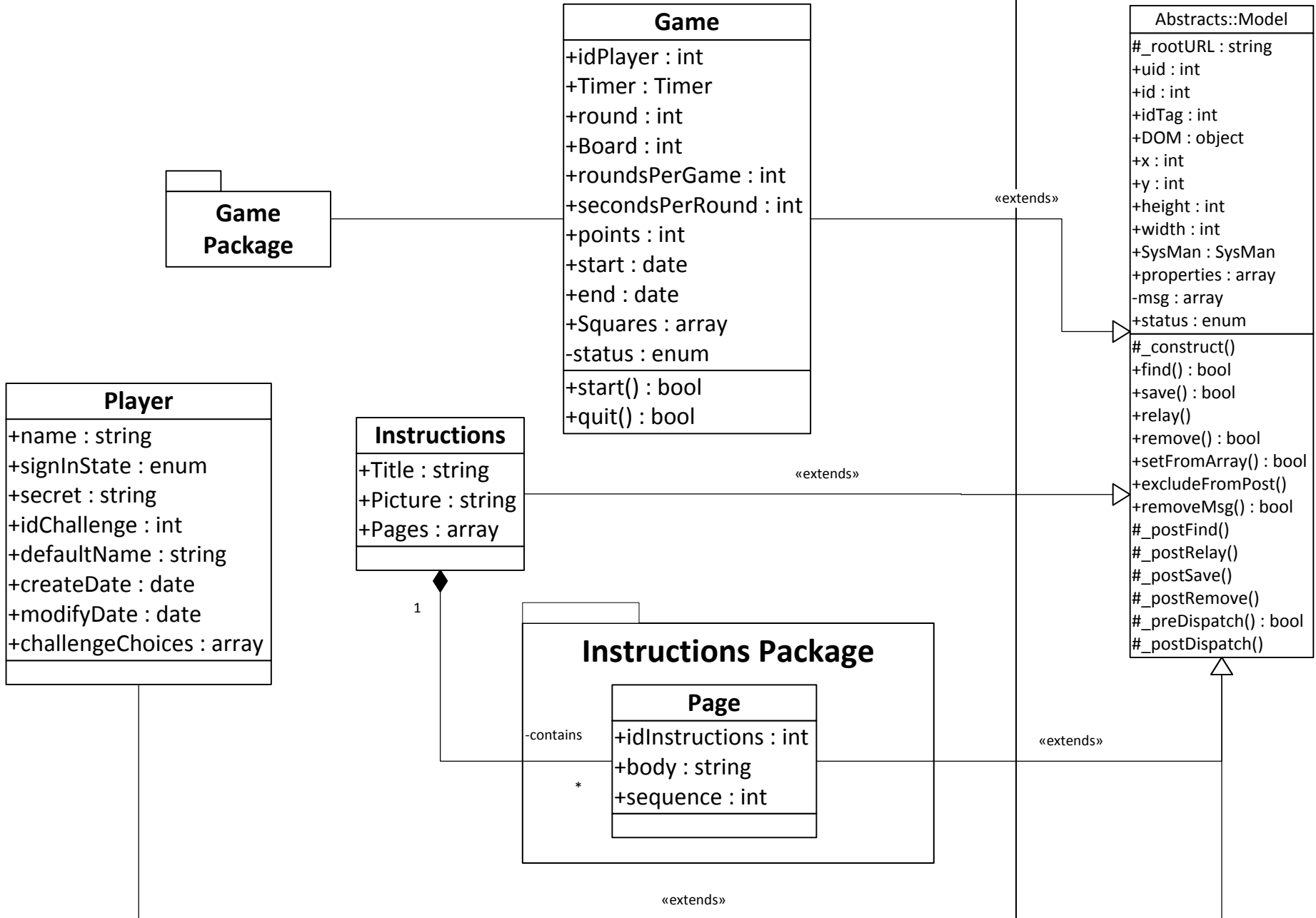
App_Modules_WordShuffle (WordShuffle Module) Package



WordShuffle_Controllers Package



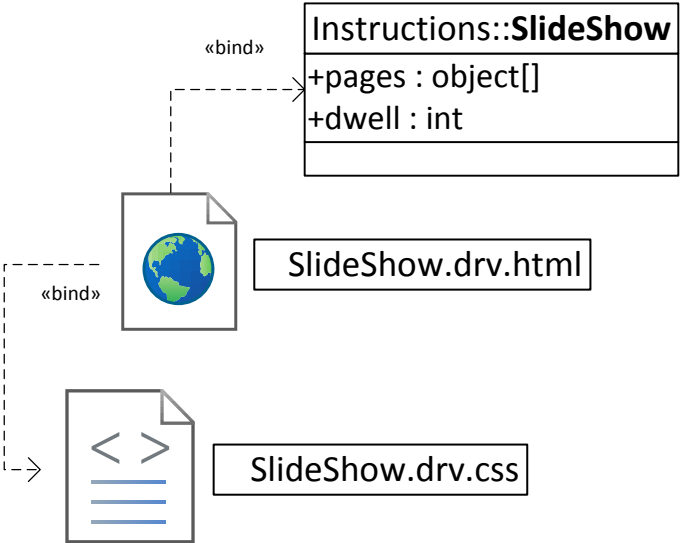
WordShuffle::Models Package



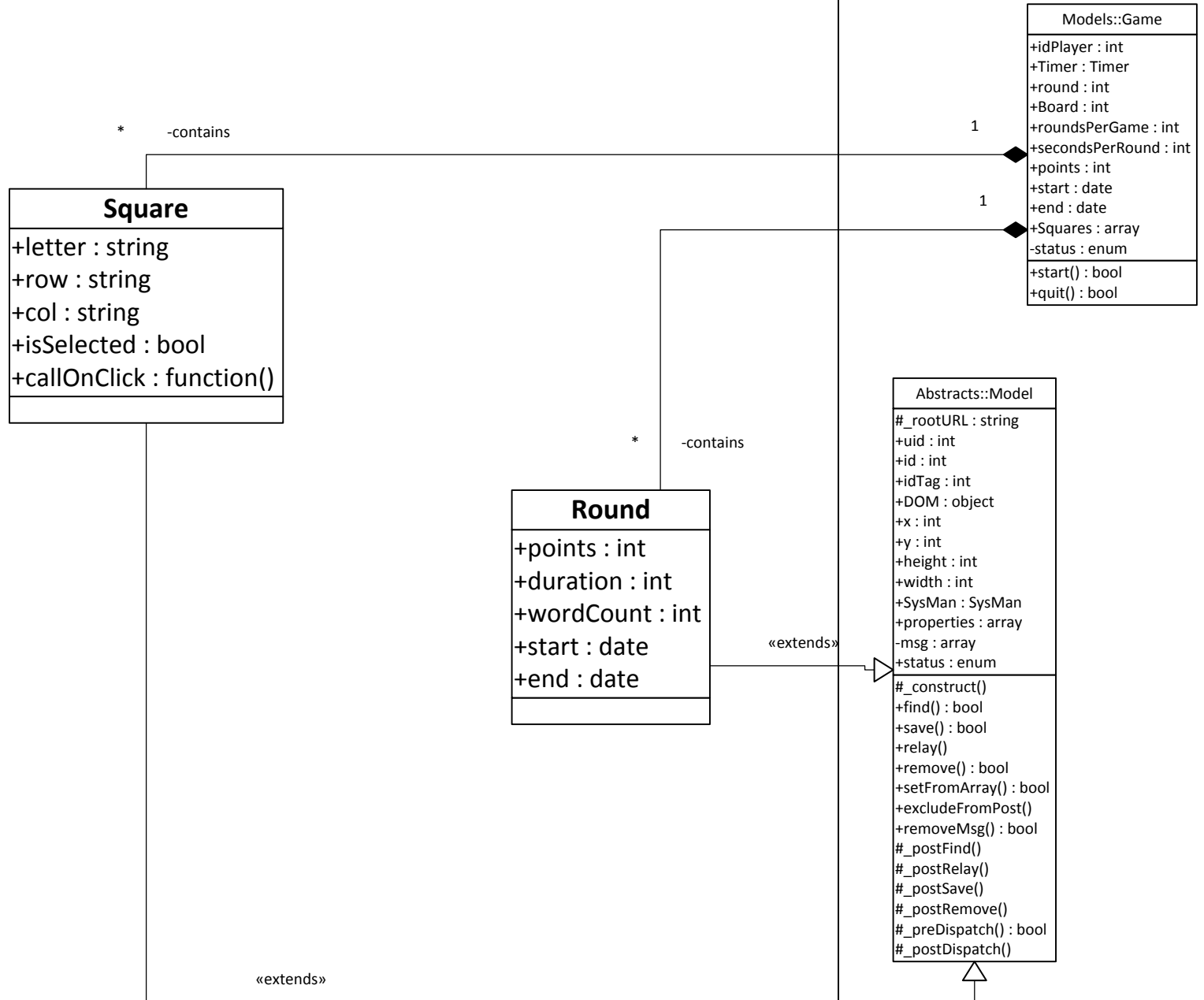
WordShuffle::Directives Package

Game

Instructions



WordShuffle::Models::Game Package



WordShuffle_Models_Game

| | | |
|------------------------|-----------------------------------|--------------------------------|
| idPlayer | {int} | - primary key for player |
| instructions | {string[]} | - html pages explaining game |
| Timer | {App_Library_Models_Timer} | - game timer |
| round | {int} | - current round |
| roundsPerGame | {int} | - number of rounds in the game |
| secondsPerRound | {int} | - seconds in each round |
| points | {int} | - total game points |
| start | {date} | - date/time game started |
| end | {end} | - date/time game ended |
| Rounds | | |
| | {WordShuffle_Models_Game_Round[]} | - array of game rounds |

| | |
|-------------------|---|
| start() | {bool} |
| <i>details:</i> | Starts the game |
| <i>arguments:</i> | none |
| <i>return:</i> | true if start successful |
| quit() | {bool} |
| <i>details:</i> | Quit current game |
| <i>arguments:</i> | none |
| <i>return:</i> | true if quit successful |
| save() | {bool} |
| <i>details:</i> | Saves results of completed game to the database |
| <i>arguments:</i> | none |
| <i>return:</i> | true if save successful |

Wordshuffle_Views Package

[Goto Welcome Page](#)

Welcome

Play

Setup

Welcome.tpl.html

Layout.tpl.html

Welcome.tpl.css

Layout.tpl.css

Setup.tpl.html

Setup.tpl.css

Play.tpl.html

Play.tpl.css

Directives::Banner
+msg : object[]
+expires : int
+closeMsg()

«bind»

«bind»

«bind»

«bind»

«bind»

«bind»

«bind»

Word Shuffle

[Welcome](#) [Setup](#) [Play](#)

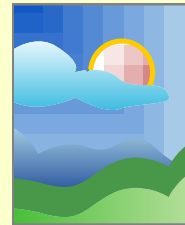
Your Stats

Round High: 156
Round Avg: 91
Avg Pts per Word: 21
Longest Word: 8



Let's Play WordShuffle!

Word Shuffle is an exciting word finding game. Twenty-five letters are randomly distributed over a 5x5 matrix. Your objective is to make as many words as possible with adjacent letters used only once. You can connect letters to the left and right, up and down, and even diagonally, but you can only touch a given letter once as you create the word.

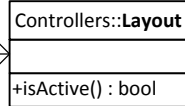


The objective is to make as many words as possible. You get more points for making longer words. The point score follows: 1 point for each letter up to the 4th letter, 2 points for each letter thereafter up to the 6th letter, 4 points for each letter from thereafter.

WordShuffle_Views_Layout.tpl.html

[Goto Welcome Page](#)

«bind»



The `Layout.isActive()` method controls the “active” class for the menu items. The text font and background change color when the associated controller is active as determined by the URL controller state

Word Shuffle

Welcome Setup Play

ui-view

One of the templates below depending on URL state, specifically `:controller` parameter in `‘/m/:module/:controller/:action’`. In this case, the `“:module”` parameter is `“wordshuffle”`.

WordShuffle_Views Package



Welcome.tpl.html



Setup.tpl.html



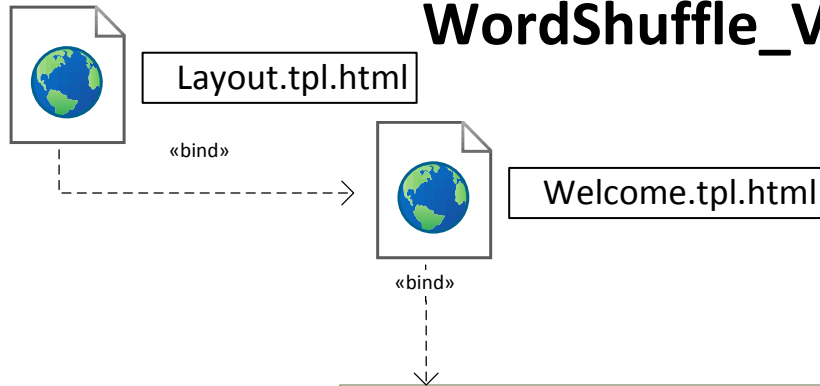
Play.tpl.html

© Copyrighted May 2014 Grace Clinic Holdings

This Layout template binds the Layout controller through the `ng-controller` directive, specifically `ng-controller="WordShuffle_Controllers_Layout as Layout"`. It defines the header and footer for the entire WordShuffle application. It also specifies styling that maintains the look and feel of the WordShuffle application

WordShuffle_VIEWS_Welcome_Index.tpl.html

[Goto Welcome Page](#)



{{Welcome.Instructions.Title}}

{{Welcome.Instructions.Picture}}

Directive accepts "pages" and "dwell" time as inputs. Set dwell to 5000msec.

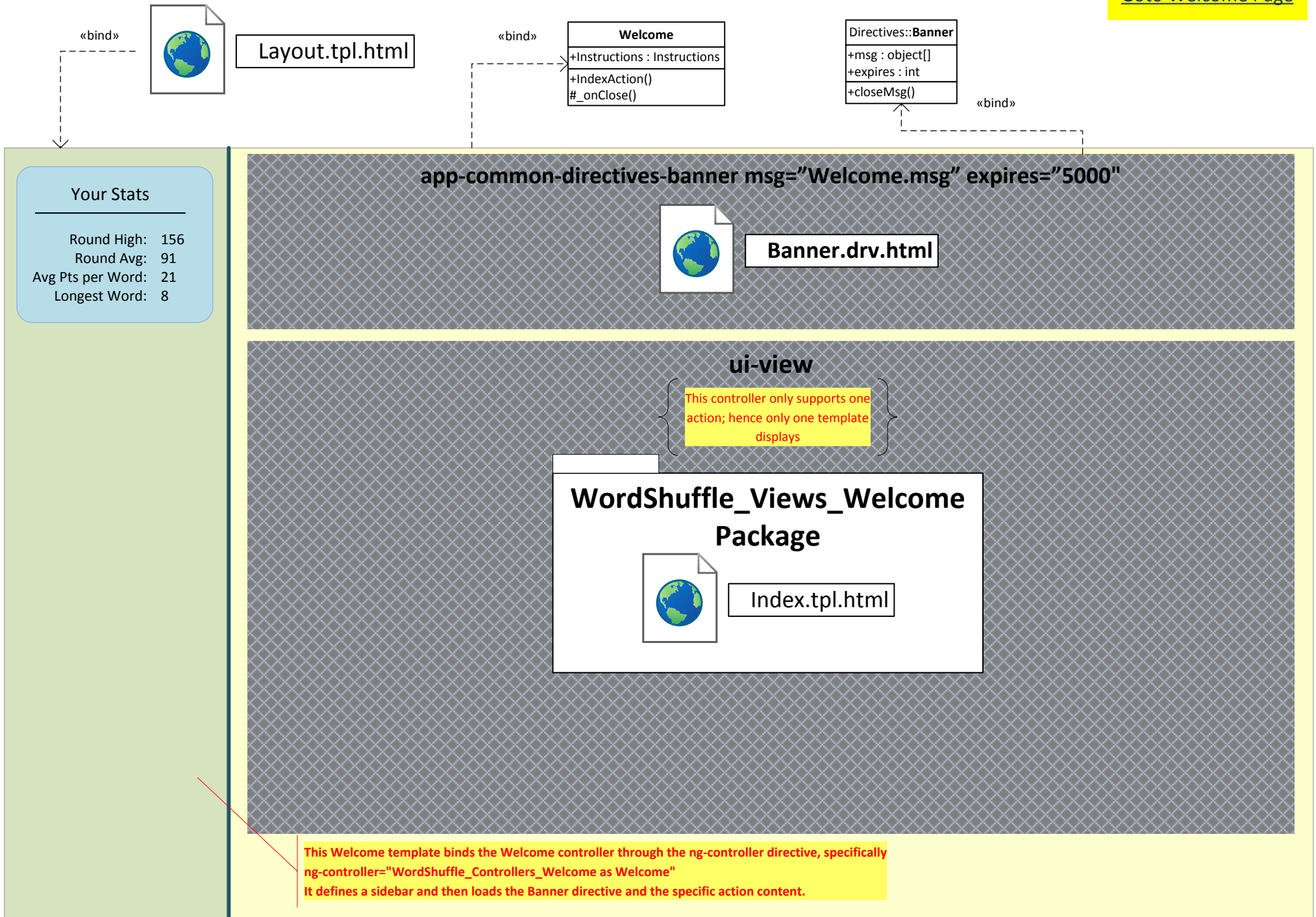
wordshuffle-directives-instructions-slide-show
pages="Welcome.Instructions.Pages" dwell="5000"



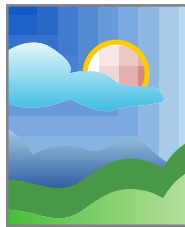
SlideShow.driv.html

WordShuffle_Views_Welcome.tpl.html

[Goto Welcome Page](#)



Word Shuffle is an exciting word finding game. Twenty-five letters are randomly distributed over a 5x5 matrix. Your objective is to make as many words as possible with adjacent letters used only once. You can connect letters to the left and right, up and down, and even diagonally, but you can only touch a given letter once as you create the word.



The objective is to make as many words as possible. You get more points for making longer words. The point score follows: 1 point for each letter up to the 4th letter, 2 points for each letter thereafter up to the 6th letter, 4 points for each letter from thereafter.

continuously rotate
"index" to display
pages[index].body