

Php Traits

Php Traits

- [Aero\\Common\\Traits\\CanExtendFillable](#)
- [Aero\\Common\\Traits\\CastsToJavaScript](#)
- [Aero\\Common\\Traits\\Sluggable](#)
- [Aero\\Catalog\\Traits\\Taggable](#)
- [Aero\\Common\\Traits\\CanHaveAdditionalAttributes](#)
- [Aero\\Catalog\\Traits\\HasPrices](#)
- [Aero\\Store\\Traits\\Seoable](#)
- [Aero\\Responses\\Traits\\HasSections](#)
- [Aero\\Admin\\Traits\\HasPermissions](#)
- [Aero\\Admin\\Traits\\IsExtendable](#)

Aero\\Common\\Traits\\CanExtendFillable

This trait can be added to a model class. When added your model will have the ability to have fillables dynamically added. The trait adds the following methods:

static makeFillable Method

This method accepts a key for the attribute to make fillable.

Aero\\Common\\Traits\\Casts ToJavaScript

This trait can be added to a model class. This trait provides a way of defining attributes that should be kept when casting to javascript. You can define these attributes in a protected **\$castsToJs** array. The trait adds the following methods:

static addCastToJs Method

This static method provides a way to dynamically add attributes to the models **\$castsToJs** array. It expects you to pass the attribute key to be added.

jsSerialize Method

This method returns the value of the **attributesToJs** method.

toJavaScript Method

This method returns the JSON value of the **jsSerialize** method.

attributesToJs Method

This method returns an array of the attributes that should be casted to javascript.

getCastsToJs Method

This method returns the attribute keys that should be cast to javascript.

Aero\\Common\\Traits\\Sluggable

This trait can be added to a model class. The trait adds a relationship to **Aero\\Store\\Models\\Slug** and helper methods for adding a slug to the model.

Aero\\Catalog\\Traits\\Taggable

This trait can be added to a model class. The trait adds the ability to add tags to the model. The trait adds the following methods:

tag Method

This method accepts a **Aero\\Catalog\\Models\\Tag** and will attach the tag to the model without detaching any other tags.

tags Method

This method defines the Eloquent relationship with **Aero\\Catalog\\Models\\Tag**.

untag Method

This method accepts a **Aero\\Catalog\\Models\\Tag** and will detach the tag from the model without detaching any other tags.

Aero\\Common\\Traits\\CanHaveAdditionalAttributes

This trait can be added to a model class. When added your model will have the ability to have additional attributes. The trait adds the following methods:

additional Method

This is the relationship method. This provides the relationship to **Aero\\Common\\Models\\AdditionalAttribute**.

hasAdditional Method

This method accepts a key string parameter and will do a database query to return a boolean value if the additional attribute with the key you provided exists.

scopeHavingAdditional Method

This method is a [Laravel local scope method](#) that will allow you to scope a query on your module to require that the model has a specific additional attribute.

getAdditionalAttribute Method

This method accepts an additional attributes key and allows you to get an additional attributes value using its key.

saveAdditionalAttribute Method

This method accepts key and value parameters and will save the key and value you provide as an additional attribute.

Aero\\Catalog\\Traits\\HasPrices

This trait can be added to a model class. This trait adds prices to the model. The trait adds the following methods:

currentPrices Method

This method returns a collection of the current viable price based upon the date and time.

basePrice Method

This method returns the base price.

prices Method

This method defines the eloquent relationship with **Aero\\Catalog\\Models\\Price**.

futurePrices Method

This method returns a collection of prices that are set for the future.

Aero\\Store\\Traits\\Seoable

This trait can be added to a model class. The trait adds a relationship to **Aero\\Store\\Models\\Seo** and helper methods for adding SEO data to the model.

Aero\\Responses\\Traits\\Has Sections

This trait can be added to a response builder class. It adds useful methods for managing sections of the page. There are other traits (HasBottomSections, HasTopSections, and HasSidebarSections) in the same namespace that add more section helpers (these traits all require that the class uses the HasSections trait as well). The trait adds the following methods:

getSectionKey Method

This method defines the key that is used to store the sections data on the response builder (defaults to 'sections').

setSection Method

This method will set a section. It accepts a key and value and optionally a section key (the section key will default to the **getSectionKey** method).

setSections Method

This method will set the sections. It accepts an array of sections and optionally a section key (the section key will default to the **getSectionKey** method).

getSection Method

This method will return a specified section. It accepts a key for the section and optionally a section key (the section key will default to the **getSectionKey** method).

getSections Method

This method will return a collection of the sections. It optionally accepts a section key (the section key will default to the **getSectionKey** method).

addSection Method

This method accepts a key and value to add to the sections. Using this method will put your section at the top of the sections collection.

addSectionBefore Method

This method works the same way as the **addSection** method but also accepts a key for the section that this new section should be added before.

addSectionAfter Method

This method works the same way as the **addSection** method but also accepts a key for the section that this new section should be added after.

removeSection Method

This method accepts a key of the section to remove and optionally accepts a section key (the section key will default to the **getSectionKey** method).

hasSections Method

This method returns if there are sections. It optionally accepts a section key (the section key will default to the **getSectionKey** method).

Aero\\Admin\\Traits\\HasPermissions

This trait adds the ability to add permissions to a class easily. The trait adds the following methods:

permissions Method

This method accepts an array that will set the permissions required.

allowed Method

This method optionally accepts a user and will return if the user has the required permissions.

Aero\\Admin\\Traits\\IsExtendable

This trait can be added to a class. This trait can be used on Pipelines (more information "[introduction to pipelines](#)")