

What is the combination serializer service?

The combination serializer service is used to serialize and deserialize models.

Serializing

This is used to take a collection of models and convert them into a string that can be used to find a corresponding combination.

Let's say you have a category called `mens`, a manufacturer called `acme` and a `blue` tag in the group `color`. Then if you visit `/mens/acme/blue` on the storefront it will load any products in the category `mens` that have the manufacturer `acme` and are tagged with `blue`.

Perhaps you might rather that it 404s unless a combination for those models exists (as otherwise there may very likely be no products that matches the criteria). This can be done by serializing the slugs and considering them invalid if they don't correspond to an existing combination model:

```
\Aero\Store\Routing\Slugs::setValidator(function (\Illuminate\Database\Eloquent\Collection
$slugs, $resolver) {
    if ($resolver !== 'listings') return true;

    try {
        $serialized = \Aero\Common\Services\CombinationSerializer::serialize(
            $slugs->map(function (\Aero\Store\Models\Slug $slug) {
                return $slug->model;
            })
        );

        return \Aero\Catalog\Models\Combination::where('serialized', $serialized)->exists();
    } catch (\Exception $e) {
        return false;
    }
});
```

Deserializing

This is used to take a combination model and find all of its' component models (collections, categories, manufacturers & tags).

For example, you might want to resolve the category ids for the listings page from the combination:

```
\Aero\Store\Http\Responses\ListingsPage::extend(function
(\Aero\Store\Http\Responses\ListingsPage $page, \Closure $next) {
    $categoryIds = \Aero\Common\Services\CombinationSerializer::deserialize($page-
>getData('combination'))
        ->filter(function (\Illuminate\Database\Eloquent\Model $model) {
            return $model instanceof \Aero\Catalog\Models\Category;
        })
        ->pluck('id');

    // Do stuff...

    return $next($page);
});
```

Revision #2

Created 2026-09-24 10:55:39 UTC by Ethan Payne

Updated 2026-10-02 12:41:04 UTC by Ethan Payne