HiEventsDev / **Hi.Events** Public

Code Issues 128 Pull requests 21 Discussions Actions Projects

Commit `01e1aee`

daveearley and **claude** authored 4 days ago · 6 / 6 · Verified

```
fix: Validate sort_by parameter against allowlist in repositories (#1128)

Co-authored-by: Claude Opus 4.6 (1M context) <noreply@anthropic.com>
```

`develop` (#1128) · v1.7.1-beta

1 parent `8fd3412` commit `01e1aee`

13 files changed +81 -27 lines changed

Top

Filter files…

- backend/app
  - DomainObjects
    - ProductCategoryDomainObject.php
  - Repository/Eloquent
    - AffiliateRepository.php
    - AttendeeRepository.php
    - BaseRepository.php
    - CapacityAssignmentRepository.php
    - CheckInListRepository.php
    - EventRepository.php
    - MessageRepository.php
    - OrderRepository.php
    - ProductCategoryRepository.php
    - ProductRepository.php
    - PromoCodeRepository.php

WaitlistEntryRepository.php

13 files changed +81 -27 lines changed

Search within code

...mainObjects/ProductCategoryDomainObject.php

```
@@ -2,10 +2,40 @@
2  2
3  3   namespace HiEvents\DomainObjects;
4  4
       + use HiEvents\DomainObjects\Interfaces\IsSortable;
       + use HiEvents\DomainObjects\SortingAndFiltering\AllowedSorts;
5  7   use Illuminate\Support\Collection;
6  8
7     - class ProductCategoryDomainObject extends Generated\ProductCategoryDomainObjectAbstract
       + class ProductCategoryDomainObject extends Generated\ProductCategoryDomainObjectAbstract implements IsSortable
8  10  {
       +     public static function getDefaultSort(): string
       +     {
       +         return self::ORDER;
       +     }
       +
       +     public static function getDefaultSortDirection(): string
       +     {
       +         return 'asc';
       +     }
       +
       +     public static function getAllowedSorts(): AllowedSorts
       +     {
       +         return new AllowedSorts([
       +             self::ORDER => [
       +                 'asc' => __('Order Ascending'),
       +                 'desc' => __('Order Descending'),
       +             ],
       +             self::CREATED_AT => [
       +                 'asc' => __('Oldest First'),
       +                 'desc' => __('Newest First'),
       +             ],
```

```
      32  +             self::NAME => [
      33  +                 'asc' => __('Name A-Z'),
      34  +                 'desc' => __('Name Z-A'),
      35  +             ],
      36  +         ]);
      37  +     }
      38  +
 9    39          public ?Collection $products = null;
10    40
11    41          public function setProducts(Collection $products): void
```

...Repository/Eloquent/AffiliateRepository.php

```
@@ -44,8 +44,8 @@ public function findByEventId(int $eventId, QueryParamsDTO $params): LengthAware
44    44              }
45    45
46    46              $this->model = $this->model->orderBy(
47        -                 column: $params->sort_by ?? AffiliateDomainObject::getDefaultSort(),
48        -                 direction: $params->sort_direction ?? 'desc',
      47  +                 column: $this->validateSortColumn($params->sort_by, AffiliateDomainObject::class),
      48  +                 direction: $this->validateSortDirection($params->sort_direction, AffiliateDomainObject::class),
49    49              );
50    50
51    51              return $this->paginateWhere(
```

.../Repository/Eloquent/AttendeeRepository.php

```
@@ -85,8 +85,8 @@ public function findByEventId(int $eventId, QueryParamsDTO $params): LengthAware
85    85                  $this->applyFilterFields($params, AttendeeDomainObject::getAllowedFilterFields(), prefix: 'attendees');
86    86              }
87    87
88        -         $sortBy = $params->sort_by ?? AttendeeDomainObject::getDefaultSort();
89        -         $sortDirection = $params->sort_direction ?? AttendeeDomainObject::getDefaultSortDirection();
```

```
88  +         $sortBy = $this->validateSortColumn($params->sort_by, AttendeeDomainObject::class);
89  +         $sortDirection = $this->validateSortDirection($params->sort_direction, AttendeeDomainObject::class);
90  90
91  91            if ($sortBy === AttendeeDomainObject::TICKET_NAME_SORT_KEY) {
92  92                $this->model = $this->model
```

.../app/Repository/Eloquent/BaseRepository.php

```
@@ -7,6 +7,7 @@
7   7   use BadMethodCallException;
8   8   use Carbon\Carbon;
9   9   use HiEvents\DomainObjects\Interfaces\DomainObjectInterface;
10  +   use HiEvents\DomainObjects\Interfaces\IsSortable;
10  11  use HiEvents\Http\DTO\QueryParamsDTO;
11  12  use HiEvents\Models\BaseModel;
12  13  use HiEvents\Repository\Eloquent\Value\Relationship;
@@ -54,6 +55,28 @@ public function __construct(Application $application, DatabaseManager $db)
54  55           */
55  56          abstract protected function getModel(): string;
56  57
58  +       /**
59  +        * @param class-string<IsSortable> $domainObjectClass
60  +        */
61  +       protected function validateSortColumn(?string $sortBy, string $domainObjectClass): string
62  +       {
63  +           $allowedColumns = array_keys($domainObjectClass::getAllowedSorts()->toArray());
64  +           $default = $domainObjectClass::getDefaultSort();
65  +
66  +           if ($sortBy === null || !in_array($sortBy, $allowedColumns, true)) {
67  +               return $default;
68  +           }
69  +
70  +           return $sortBy;
71  +       }
72  +
```

```
      73  +     protected function validateSortDirection(?string $sortDirection, string
          $domainObjectClass): string
      74  +     {
      75  +         return in_array(strtolower($sortDirection ?? ''), ['asc', 'desc'], true)
      76  +             ? $sortDirection
      77  +             : $domainObjectClass::getDefaultSortDirection();
      78  +     }
      79  +
57    80          public function setMaxPerPage(int $maxPerPage): static
58    81          {
59    82              $this->maxPerPage = $maxPerPage;
```

...y/Eloquent/CapacityAssignmentRepository.php

```
          @@ -39,8 +39,8 @@ public function findByEventId(int $eventId, QueryParamsDTO
          $params): LengthAware
39    39              }
40    40
41    41              $this->model = $this->model->orderBy(
42        -                 $params->sort_by ??
          CapacityAssignmentDomainObject::getDefaultSort(),
43        -                 $params->sort_direction ??
          CapacityAssignmentDomainObject::getDefaultSortDirection(),
      42  +             $this->validateSortColumn($params->sort_by,
          CapacityAssignmentDomainObject::class),
      43  +             $this->validateSortDirection($params->sort_direction,
          CapacityAssignmentDomainObject::class),
44    44              );
45    45
46    46              return $this->paginateWhere(
```

...pository/Eloquent/CheckInListRepository.php

```
          @@ -140,8 +140,8 @@ public function findByEventId(int $eventId,
          QueryParamsDTO $params): LengthAware
140   140              }
141   141
142   142              $this->model = $this->model->orderBy(
143       -                 $params->sort_by ?? CheckInListDomainObject::getDefaultSort(),
```

```
144     -             $params->sort_direction ?? CheckInListDomainObject::getDefaultSortDirection(),
        143 +         $this->validateSortColumn($params->sort_by, CheckInListDomainObject::class),
        144 +         $this->validateSortDirection($params->sort_direction, CheckInListDomainObject::class),
145     145           );
146     146
147     147           return $this->paginateWhere(
```

...app/Repository/Eloquent/EventRepository.php

```
@@ -81,8 +81,8 @@ public function findEvents(array $where, QueryParamsDTO $params): LengthAwarePag
81      81            }
82      82
83      83            $this->model = $this->model->orderBy(
84      -             $params->sort_by ?? EventDomainObject::getDefaultSort(),
85      -             $params->sort_direction ?? EventDomainObject::getDefaultSortDirection(),
        84 +          $this->validateSortColumn($params->sort_by, EventDomainObject::class),
        85 +          $this->validateSortDirection($params->sort_direction, EventDomainObject::class),
86      86            );
87      87
88      88            return $this->paginateWhere(
```

...p/Repository/Eloquent/MessageRepository.php

```
@@ -46,8 +46,8 @@ public function findByEventId(int $eventId, QueryParamsDTO $params): LengthAware
46      46            }
47      47
48      48            $this->model = $this->model->orderBy(
49      -             $params->sort_by ?? MessageDomainObject::getDefaultSort(),
50      -             $params->sort_direction ?? 'desc',
        49 +          $this->validateSortColumn($params->sort_by, MessageDomainObject::class),
```

```
      50  +             $this->validateSortDirection($params->sort_direction,
          MessageDomainObject::class),
51    51            );
52    52
53    53            return $this->paginateWhere(
```

...app/Repository/Eloquent/OrderRepository.php

```
          @@ -57,8 +57,8 @@ public function findByEventId(int $eventId, QueryParamsDTO
          $params): LengthAware
57    57            }
58    58
59    59            $this->model = $this->model->orderBy(
60        -             column: $params->sort_by ?? OrderDomainObject::getDefaultSort(),
61        -             direction: $params->sort_direction ?? 'desc',
      60  +             column: $this->validateSortColumn($params->sort_by,
          OrderDomainObject::class),
      61  +             direction: $this->validateSortDirection($params->sort_direction,
          OrderDomainObject::class),
62    62            );
63    63
64    64            return $this->paginateWhere(
          @@ -102,9 +102,10 @@ public function findByOrganizerId(int $organizerId, int
          $accountId, QueryParamsD
102   102               ->where('events.organizer_id', $organizerId)
103   103               ->where('events.account_id', $accountId);
104   104
      105 +         $sortBy = $this->validateSortColumn($params->sort_by,
          OrderDomainObject::class);
105   106           $this->model = $this->model->orderBy(
106       -             column: $params->sort_by ? 'orders.' . $params->sort_by : 'orders.'
          . OrderDomainObject::getDefaultSort(),
107       -             direction: $params->sort_direction ?? 'desc',
      107 +             column: 'orders.' . $sortBy,
      108 +             direction: $this->validateSortDirection($params->sort_direction,
          OrderDomainObject::class),
108   109           );
109   110
110   111           return $this->paginateWhere(
```

...tory/Eloquent/ProductCategoryRepository.php

```
@@ -36,10 +36,10 @@ public function findByEventId(int $eventId,
QueryParamsDTO $queryParamsDTO): Col
            }
        }

-       // Apply sorting from QueryParamsDTO
-       if (!empty($queryParamsDTO->sort_by)) {
-           $query->orderBy($queryParamsDTO->sort_by, $queryParamsDTO->sort_direction ?? 'asc');
-       }
+       $query->orderBy(
+           $this->validateSortColumn($queryParamsDTO->sort_by, ProductCategoryDomainObject::class),
+           $this->validateSortDirection($queryParamsDTO->sort_direction, ProductCategoryDomainObject::class),
+       );

        return $query->get();
    }
```

...p/Repository/Eloquent/ProductRepository.php

```
@@ -41,8 +41,8 @@ public function findByEventId(int $eventId, QueryParamsDTO $params): LengthAware
        }

        $this->model = $this->model->orderBy(
-           $params->sort_by ?? ProductDomainObject::getDefaultSort(),
-           $params->sort_direction ?? ProductDomainObject::getDefaultSortDirection(),
+           $this->validateSortColumn($params->sort_by, ProductDomainObject::class),
+           $this->validateSortDirection($params->sort_direction, ProductDomainObject::class),
        );

        return $this->paginateWhere(
```

...Repository/Eloquent/PromoCodeRepository.php

```
@@ -39,8 +39,8 @@ public function findByEventId(int $eventId, QueryParamsDTO $params): LengthAware
39  39          }
40  40
41  41          $this->model = $this->model->orderBy(
42      -           column: $params->sort_by ?? PromoCodeDomainObject::getDefaultSort(),
43      -           direction: $params->sort_direction ?? 'desc',
    42  +           column: $this->validateSortColumn($params->sort_by, PromoCodeDomainObject::class),
    43  +           direction: $this->validateSortDirection($params->sort_direction, PromoCodeDomainObject::class),
44  44          );
45  45
46  46          return $this->paginateWhere(
```

...sitory/Eloquent/WaitlistEntryRepository.php

```
@@ -156,8 +156,8 @@ public function findByEventId(int $eventId, QueryParamsDTO $params): LengthAware
156 156          }
157 157
158 158          $this->model = $this->model->orderBy(
159     -           column: $params->sort_by ?? WaitlistEntryDomainObject::getDefaultSort(),
160     -           direction: $params->sort_direction ?? WaitlistEntryDomainObject::getDefaultSortDirection(),
    159 +           column: $this->validateSortColumn($params->sort_by, WaitlistEntryDomainObject::class),
    160 +           direction: $this->validateSortDirection($params->sort_direction, WaitlistEntryDomainObject::class),
161 161          );
162 162
163 163          return $this->loadRelation(new Relationship(
```

Comments 0

Please sign in to comment.