🏎️ Pole position on the grid 🏁

Awesome new Sylius Grid features to put you in the driver's seat

Sylius

1. Akawaka & Sylius
2. What’s under the hood of the Sylius Stack ?
3. What if… you were Toto Wolff?
4. Add a link to another grid with filtered data
5. Mystery surprise 👀
6. Quick practice before it’s your time to shine!
7. Conclusion

Estelle Gaits

- Web developer @Akawaka
- Sylius Key Contributor
- In June 2016, I was there :

Estelle Gaits
stlgaits · she/her

PHP / Symfony web developer

Edit profile

26 followers · 105 following

Akawaka
France
in/estellegaits

Achievements

x3 x3

Organizations

stlgaits / README.md

Adishatz 👋

I'm Estelle. I am a PHP / Symfony backend developer from France 🇫🇷

Once upon a time, I used to be an MFL Teacher 👩‍🏫 (French & Spanish) in London 🇬🇧

- ⚡ I'm a web developer at Akawaka
- 🌱 I'm currently learning Sylius and Sulu
- 🧞 I would like to learn Sulu, Svelte and GO as well as continue to explore API Platform.

✨ **My battleground**

PHP · SYMFONY · PHPSTORM · PHPSTAN · PHPUNIT · MYSQL · MARIADB · DOCKER · CSS3 · BOOTSTRAP · TAILWINDCSS · JAVASCRIPT · REACT · STIMULUS · JQUERY · VISUALSTUDIOCODE

Loïc Frémont

- Technical Expert @Akawaka
- Sylius Key Contributor
- Sylius Stack Main Maintainer

Akawaka & Sylius

a true 💖 story

- Sylius Partner since 2022
- 4 amazing Sylius Key Contributors :
  - Loïc Frémont
  - Valentin Silvestre
  - Florian Merle
  - Estelle Gaits

Symfony & Sylius experts for your business-critical web projects

We help you build, scale and improve your applications with:

- robust architectures built on **DDD** principles – for software that stand the test of time
- strong quality culture through systematic **testing** and continuous improvement
- flexible engagement models – fixed-price or time-based, solo or embedded in your team

AKAWAKA

Sylius

Search menu...

- Dashboard
- Library
- Configuration
  - Conferences
  - Talks
  - Speakers

Hello
admin@example.com

Dashboard

TALKS
5

SPEAKERS
3

CONFERENCES
5

2 Semaines | Mois | Année

New talks

TITLE
Voluptate architecto voluptatem debitis. Stéphane Decock
Maiores voluptas perspiciatis cupiditate voluptas ut ut quo. Mikołaj Król
Consequuntur modi explicabo voluptatibus debitis officiis nam. Jacques Bodin-Hullin
Dicta aut et id voluptatem quia quod. Manuele Menozzi
Velit perspiciatis assumenda voluptatem tempore. Loïc Caillieux

Show all

New speakers

NAME
Julien Jacottet Mezcalito
Hélèna Gravelier Synolia
Gregor Šink Creatim d.o.o.
Joachim Løvgaard Sylius Core Team Member, Setono
Stephan Hochdörfer bitExpert AG

Show all

What’s under the hood of the Sylius Stack ?

Create a lean & mean back-office in no time with :

- **Sylius Grid bundle**
- Sylius Resource bundle
- Doctrine ORM and Doctrine DBAL drivers for SyliusGridBundle
- Bootstrap Admin UI
- Symfony UX, AssetMapper and more !

What if… you were Toto Wolff?

Let’s create a Formula 1 admin panel using :

- the OpenF1 API
- the Sylius Stack
  - boosted `make:grid` command
  - new Sylius Grid Attributes
    - #[AsGrid]
    - #[AsFilter]

Create a Driver resource

```php
namespace App\Resource;

use Sylius\Resource\Metadata\AsResource;
use Sylius\Resource\Model\ResourceInterface;

#[AsResource]
final readonly class DriverResource implements ResourceInterface
{
    public function __construct(
        public int $number,
        public string $firstName,
        public string $lastName,
        public string $countryCode,
        public string $teamName,
        public string|null $image = null,
    ) {
    }

    public function getId(): int
    {
        return $this->number;
    }
}
```

Create a Driver grid

💡 *Now you can also create a grid based on a non-Doctrine entity, including… Sylius Resources !*

```
symfony console make:grid 'App\Resource\DriverResource'
```

stlgaits commented last week • edited by loic425 Member

The current implementation of the `make:grid` command assumes a Doctrine entity as the only valid input.
This PR introduces enhanced flexibility to the command by allowing developers to generate grids based on resource classes beyond standard Doctrine entities.

With this change, you can use any PHP class—such as a Sylius Resource implementation—as the basis for your grid configuration. This makes the command more versatile and better suited to projects using custom resource patterns or non-standard data sources.
Example usage:

```
symfony console make:grid 'App\BoardGameBlog\Infrastructure\\Sylius\Resource\BoardGameResource'
```

This command will now correctly scaffold a grid using the specified class, even if it is not a traditional Doctrine entity.

```php
namespace App\Grid;

use Sylius\Bundle\GridBundle\Builder\Field\StringField;
use Sylius\Bundle\GridBundle\Builder\GridBuilderInterface;
use Sylius\Bundle\GridBundle\Grid\AbstractGrid;
use Sylius\Component\Grid\Attribute\AsGrid;

#[AsGrid]
final class DriverGrid extends AbstractGrid
{
    public function buildGrid(GridBuilderInterface $gridBuilder): void
    {
        $gridBuilder
            ->addField(TwigField::create('image', 'driver/grid/field/image.html.twig'))
            ->addField(StringField::create('number'))
            ->addField(StringField::create('firstName'))
            ->addField(StringField::create('lastName'))
            ->addField(StringField::create('countryCode'))
            ->addField(StringField::create('teamName'))
        ;
    }
}
```

▦ The brand new #[AsGrid] attribute

Before

```php
use App\Entity\Dummy;
use Sylius\Bundle\GridBundle\Grid\AbstractGrid;
use Sylius\Bundle\GridBundle\Grid\ResourceAwareGridInterface;

final class DummyGrid extends AbstractGrid
implements ResourceAwareGridInterface
{
    public function buildGrid(
        GridBuilderInterface $gridBuilder,
    ): void {
        // ...
    }

    public function getResourceClass(): string
    {
        return Dummy::class;
    }
}
```

After

```php
use App\Entity\Dummy;
use Sylius\Bundle\GridBundle\Grid\AbstractGrid;

#[AsGrid(resourceClass: Dummy::class)]
final class DummyGrid extends AbstractGrid
{
    public function __invoke(
        GridBuilderInterface $gridBuilder,
    ): void
        // ...
    }
}
```

```php
use App\Entity\Dummy;
use Sylius\Bundle\GridBundle\Grid\AbstractGrid;

#[AsGrid(resourceClass: Dummy::class)]
final class DummyGrid extends AbstractGrid
{
    public function __invoke(
        GridBuilderInterface $gridBuilder,
    ): void
        // ...
    }
}
```

Starting slowly

Create a driver 🏎️ Grid Provider

```php
namespace App\Grid;

use Pagerfanta\Adapter\FixedAdapter;
use Pagerfanta\Pagerfanta;
use Pagerfanta\PagerfantaInterface;
use Sylius\Component\Grid\Data\DataProviderInterface;
use Sylius\Component\Grid\Definition\Grid;
use Sylius\Component\Grid\Parameters;

final readonly class DriverGridProvider implements DataProviderInterface
{
    public function getData(Grid $grid, Parameters $parameters): PagerFantaInterface
    {
        // start with an empty paginator
        return new Pagerfanta(new FixedAdapter(0, []));
    }
}
```

```php
// ...
use Sylius\Bundle\GridBundle\Grid\AbstractGrid;
use Sylius\Component\Grid\Attribute\AsGrid;

#[AsGrid]
final class DriverGrid extends AbstractGrid
{
    public function buildGrid(GridBuilderInterface $gridBuilder): void
    {
        $gridBuilder
            ->setProvider(DriverGridProvider::class)
            // ...
        ;
    }
}
```

Sylius

Search menu...

Dashboard

Formula One

Drivers

Dashboard / Drivers

Drivers

Filters

No results found

Adjust your search and try again.

Now, let’s use real data from the API!

```php
namespace App\Grid\Provider;

use Pagerfanta\Adapter\ArrayAdapter;
use Pagerfanta\Pagerfanta;
use Pagerfanta\PagerfantaInterface;
use Sylius\Component\Grid\Data\DataProviderInterface;
use Sylius\Component\Grid\Definition\Grid;
use Sylius\Component\Grid\Parameters;
use Symfony\Contracts\HttpClient\HttpClientInterface;

final readonly class DriverGridProvider implements DataProviderInterface
{
    public function __construct(
        private HttpClientInterface $openF1Client,
    ) {}

    public function getData(Grid $grid, Parameters $parameters): PagerFantaInterface
    {
        return new Pagerfanta(new ArrayAdapter(iterator_to_array($this->getDrivers())));
    }

    private function getDrivers(): iterable
    {
        // ...
    }
}
```

```php
namespace App\Grid\Provider;

use Pagerfanta\Adapter\ArrayAdapter;
use Pagerfanta\Pagerfanta;
use Pagerfanta\PagerfantaInterface;
use Sylius\Component\Grid\Data\DataProviderInterface;
use Sylius\Component\Grid\Definition\Grid;
use Sylius\Component\Grid\Parameters;
use Symfony\Contracts\HttpClient\HttpClientInterface;

final readonly class DriverGridProvider implements DataProviderInterface
{
    public function __construct(
        private HttpClientInterface $openF1Client
    ) {}

    // ...

    private function getDrivers(): iterable
    {
        // ...
    }
}
```

```php
namespace App\Grid\Provider;

use Sylius\Component\Grid\Data\DataProviderInterface;
use Symfony\Contracts\HttpClient\HttpClientInterface;

final readonly class DriverGridProvider implements DataProviderInterface
{
    public function __construct(
        private HttpClientInterface $openF1Client
    ) {}

    // ...

    private function getDrivers(): iterable
    {
        $responseData = $this->openF1Client->request('GET', '/v1/drivers?session_key=9158')->toArray();

        foreach ($responseData as $row) {
            yield new DriverResource(
                number: $row['driver_number'],
                firstName: $row['first_name'],
                lastName: $row['last_name'],
                countryCode: $row['country_code'],
                teamName: $row['team_name'],
                image: $row['headshot_url'],
            );
        }
```

Sylius

Search menu...

Dashboard

Formula One

Drivers

Dashboard / Drivers

Drivers

Filters

Show 10

IMAGE	NUMBER	FIRSTNAME	LASTNAME	COUNTRYCODE	TEAMNAME
	1	Max	Verstappen	NED	Red Bull Racing
	2	Logan	Sargeant	USA	Williams
	4	Lando	Norris	GBR	McLaren
	10	Pierre	Gasly	FRA	Alpine
	11	Sergio	Perez	MEX	Red Bull Racing
	14	Fernando	Alonso	ESP	Aston Martin
	16	Charles	Leclerc	MON	Ferrari
	18	Lance	Stroll	CAN	Aston Martin
	20	Kevin	Magnussen	DEN	Haas F1 Team
	22	Yuki	Tsunoda	JPN	AlphaTauri

Showing 1 to 10 of 20 entries

Previous 1 2 Next

🔎 Filter for specific drivers

using the brand new #[AsFilter] attribute

Before

```php
use Sylius\Component\Grid\Data\DataSourceInterface;
use Sylius\Component\Grid\Filtering\ConfigurableFilterInterface;
use Symfony\Component\Form\Extension\Core\Type\CountryType;

final class CountryFilter implements ConfigurableFilterInterface
{
    public function apply(DataSourceInterface $dataSource, string $name, $data, array $options): void
    {
        // TODO: Implement apply() method.
    }

    public static function getType(): string
    {
        return self::class;
    }

    public static function getFormType(): string
    {
        return CountryType::class;
    }
}
```

After

```php
use Sylius\Component\Grid\Attribute\AsFilter;
use Sylius\Component\Grid\Data\DataSourceInterface;
use Sylius\Component\Grid\Filtering\FilterInterface;
use Symfony\Component\Form\Extension\Core\Type\CountryType;

#[AsFilter(formType: CountryType::class)]
final class CountryFilter implements FilterInterface
{
    public function apply(DataSourceInterface $dataSource, string $name, $data, array $options): void
    {
        // TODO: Implement apply() method.
    }
}
```

Before

```yaml
## config/packages/sylius_grid.yaml
sylius_grid:
  filter:
    dummy: 'path/to/dummy/filter/template'
```

After

```php
use Sylius\Component\Grid\Attribute\AsFilter;
use Sylius\Component\Grid\Filtering\FilterInterface;

#[AsFilter(
    template: 'path/to/dummy/filter/template',
)]
final class DummyFilter implements FilterInterface
{
    // ..
}
```

#[AsFilter]

```php
namespace App\Grid\Filter;

use Sylius\Component\Grid\Attribute\AsFilter;
use Sylius\Component\Grid\Data\DataSourceInterface;
use Sylius\Component\Grid\Filtering\FilterInterface;
use Symfony\Component\Form\Extension\Core\Type\CountryType;

#[AsFilter(
    formType: CountryType::class, // Symfony Form Type to use
    template: '@SyliusBootstrapAdminUi/shared/grid/filter/select.html.twig', // The Twig template
)]
final class CountryFilter implements FilterInterface
{
    public function apply(DataSourceInterface $dataSource, string $name, $data, array $options): void
    {
        // We handle the filtering part in the DriverGridProvider
    }
}
```

Insert the filter into our Grid

```
namespace App\Grid;

use App\Grid\Filter\CountryFilter;
use Sylius\Bundle\GridBundle\Builder\Filter\Filter;
use Sylius\Bundle\GridBundle\Builder\GridBuilderInterface;
use Sylius\Bundle\GridBundle\Grid\AbstractGrid;
use Sylius\Component\Grid\Attribute\AsGrid;

#[AsGrid]
final class DriverGrid extends AbstractGrid
{
    public function buildGrid(GridBuilderInterface $gridBuilder): void
    {
        $gridBuilder
            // ...
            ->addFilter(
                Filter::create('country', CountryFilter::class)
                    ->setFormOptions([
                        'alpha3' => true,
                        'autocomplete' => true,
                    ])
                    ->setLabel('app.ui.country')
            )
            // ...
        ;
```

Actual filtering logic inside the provider

```php
namespace App\Grid\Provider;

use Sylius\Component\Grid\Data\DataProviderInterface;
use Symfony\Contracts\HttpClient\HttpClientInterface;

final readonly class DriverGridProvider implements DataProviderInterface
{
    public function __construct(
        private HttpClientInterface $openF1Client,
    ) {}

    // ...

    private function getDrivers(): iterable
    {
        $responseData = $this->openF1Client->request('GET', '/v1/drivers?session_key=9158')->toArray();

        foreach ($responseData as $row) {
            yield new DriverResource(
                number: $row['driver_number'],
                firstName: $row['first_name'],
                lastName: $row['last_name'],
                countryCode: $row['country_code'],
                teamName: $row['team_name'],
                image: $row['headshot_url'],
```

```php
namespace App\Grid\Provider;
//...

final readonly class DriverApiGridProvider implements DataProviderInterface
{
    // ...

    public function getData(Grid $grid, Parameters $parameters): PagerFantaInterface
    {
        return new Pagerfanta(
            new ArrayAdapter(iterator_to_array(
                $this->getDrivers($parameters->get('criteria')),
            ))
        );
    }

    private function getDrivers(array $criteria): iterable
    {
        $query = ['session_key' => 9158];

        if (!empty($criteria['country'] ?? null)) {
            $query['country_code'] = $criteria['country'];
        }

        $responseData = $this->openF1Client
            ->request(method: 'GET', url: '/v1/drivers', options: ['query' => $query])
            ->toArray()
```

Sylius

Search menu...

- Dashboard
- Formula One
  - Drivers

Dashboard / Drivers

Drivers

Filters

Country

fr

- France
- French Guiana
- French Polynesia
- French Southern Territories
- South Africa
- Central African Republic

IMAGE	NUMBER	FIRSTNAME	LASTNAME	COUNTRYCODE	TEAMNAME
	1	Max	Verstappen	NED	Red Bull Racing
	2	Logan	Sargeant	USA	Williams
	4	Lando	Norris	GBR	McLaren
	10	Pierre	Gasly	FRA	Alpine
	11	Sergio	Perez	MEX	Red Bull Racing
	14	Fernando	Alonso	ESP	Aston Martin
	16	Charles	Leclerc	MON	Ferrari
	18	Lance	Stroll	CAN	Aston Martin
	20	Kevin	Magnussen	DEN	Haas F1 Team

Sylius

Search menu...

Dashboard

Formula One

Drivers

Dashboard / Drivers

Drivers

Filters

Country

France

Filter Reset

Show 10

IMAGE	NUMBER	FIRSTNAME	LASTNAME	COUNTRYCODE	TEAMNAME
	10	Pierre	Gasly	FRA	Alpine
	31	Esteban	Ocon	FRA	Alpine

Showing 1 to 2 of 2 entries

Previous 1 Next

Add a link to another grid with filtered data

Toto Wolff needs to listen in on Lewis Hamilton's team radio 📻

Actions: links with filter params

```
use Sylius\Bundle\GridBundle\Builder\Action\Action;
use Sylius\Bundle\GridBundle\Builder\ActionGroup\ItemActionGroup;

#[AsGrid]
final class DriverGrid extends AbstractGrid
{
    public function buildGrid(GridBuilderInterface $gridBuilder): void
    {
        $gridBuilder
            ->addActionGroup(
                ItemActionGroup::create(
                    Action::create('team_radios', 'show')
                        ->setOptions([
                            'link' => [
                                'route' => 'app_admin_team_radio_index',
                                'parameters' => [
                                    'criteria' => [
                                        'driver_number' => [
                                            'value' => 'resource.number', // driverResource->number
                                        ],
                                    ],
                                ],
                            ],
                        ])
                        ->setLabel('app.ui.show_team_radios')
```

Sylius

Search menu...

- Dashboard
- Formula One
  - Drivers
  - Team radios
  - Sessions

Dashboard / Drivers

Drivers

Filters

Show 10

IMAGE	NUMBER	FIRSTNAME	LASTNAME	COUNTRYCODE	TEAMNAME	ACTIONS
	1	Max	Verstappen	NED	Red Bull Racing	
	2	Logan	Sargeant	USA	Williams	
	4	Lando	Norris	GBR	McLaren	
	10	Pierre	Gasly	FRA	Alpine	
	11	Sergio	Perez	MEX	Red Bull Racing	
	14	Fernando	Alonso	ESP	Aston Martin	
	16	Charles	Leclerc	MON	Ferrari	
	18	Lance	Stroll	CAN	Aston Martin	
	20	Kevin	Magnussen	DEN	Haas F1 Team	

Sylius

Search menu...

- Dashboard
- Formula One
  - Drivers
  - Team radios
  - Sessions

Dashboard / Team radios

Team radios

Filters

Driver number

1

Filter Reset

Show 10

DRIVERNUMBER	DATE	ACTIONS
1	2023-09-15 09:39:02	
1	2023-09-15 09:45:45	
1	2023-09-15 10:04:14	

Showing 1 to 3 of 3 entries

Previous 1 Next

Push, Push, Push!

🧟 New Grid Live Component

it’s alive…

Overview of the new DataTableComponent

```php
#[AsLiveComponent(name: 'sylius_grid_data_table')]
final class DataTableComponent
{
    #[LiveProp(writable: true)]
    public string|null $grid = null;

    #[LiveProp(writable: true)]
    public int $page = 1;

    #[LiveProp(writable: true)]
    public array|null $criteria = null;

    #[LiveProp(writable: true)]
    public array|null $sorting = null;

    #[LiveProp(writable: true)]
    public int|null $limit = null;
}
```

Transform your grid into a Live Component

```yaml
sylius_twig_hooks:
    hooks:
        'sylius_admin.book.index.content.grid':
            data_table:
                component: 'sylius_grid_data_table'
                props:
                    grid: '@=_context.grid'
                    page: '@=_context.page'
                    criteria: '@=_context.criteria'
                    sorting: '@=_context.sorting'
                    limit: '@=_context.limit'
```

Use it in any template

```twig
<!-- templates/book/show/body.html.twig -->
{{ component('sylius_grid_data_table', {
    grid: 'app_admin_book',
    criteria: {
        author: author.id,
    },
}) }}
```

Quick practice before it's your time to shine!

Q1: How can you add a Team filter for the F1 drivers' grid?

```
->addFilter(Filter::create('teamName', TeamFilter::class))
```

- ```
  #[AsFilter(formType: TeamFilterType::class,  template: FilterTemplate::SELECT->value,)]
  final class TeamFilter
  ```
- ```
  use Sylius\Component\Grid\Filtering\ConfigurableFilterInterface;

  final class TeamFilter implements ConfigurableFilterInterface {

  public static function getFormType(): string
  {
      return TeamFilterType::class;
  }
  // ...
  ```
- ```
  use Sylius\Component\Grid\Filtering\FilterInterface;

  #[AsFilter(formType: TeamFilterType::class, template: FilterTemplate::SELECT->value,)]
  final class TeamFilter implements FilterInterface
  ```
  ```

Q2: Find the odd one out.

- `#[AsFilter()]`
- `#[AsGrid()]`
- `#[AsProvider()]`
- `#[AsResource()]`

Q3: The OpenF1 API does not include a **/teams** endpoint, but you would like to create a grid listing all teams. You've created this simple model … :

```
final readonly class Team
{
    public function __construct(
        public string $id,
        public string $name,
        public string|null $color = null,
    ) {
    }
}
```

Q3: … and the 'hard-coded' data provider below.

```
final readonly class TeamGridProvider implements DataProviderInterface
{
    public function getData(Grid $grid, Parameters $parameters): PagerFantaInterface
    {
        $teams = iterator_to_array($this->getTeamsPaginator());

        return new Pagerfanta(new ArrayAdapter($teams));
    }

    public function getTeamsPaginator(): iterable
    {
        $teams = [
            ['name' =>'Ferrari', 'color' => 'F91536'],
            ['name' =>'Mercedes', 'color' => '6CD3BF'],
            ['name' =>'Red Bull Racing', 'color' => '3671C6'],
            // ...
        ];

        foreach ($teams as $team) {
            yield new Team(
                id: $team['name'],
                name: $team['name'],
                color: $team['color'],
            );
```

Q3: What command could you run to generate the missing grid?

- `symfony console create:grid 'App\Model\Team'`
- `symfony console make:resource 'App\Model\Team'`
- `symfony console make:grid 'App\Model\Team'`
- `symfony console make:grid 'App\Resource\TeamResource'`

Conclusion

Congratulations!

https://stack.sylius.com/ - Stack (docs)

https://github.com/Sylius/Stack - Stack

https://github.com/Sylius/SyliusGridBundle - Grid

https://stack.sylius.com/grid/index/installation - Grid (docs)

https://github.com/Sylius/SyliusResourceBundle - Resource

https://openf1.org/ - Open F1 API by br-g (Bruno Godefroy)