part1.md

Part 1: Blog

As a way of exploring the Sylius Resource and Grid bundles we will create a simple blogging system.

Create Post entity

Create a Doctrine ORM entity to represent a blog post.

1. `php app/console server:run`
2. Create a `Post` entity in `src/AppBundle/Entity/Post.php`.
3. Map the fields `$title`, `$body` and `$publishedAt`.

Hints:

- `use Doctrine\ORM\Mapping as ORM`
- `@ORM\Id()`
- `@ORM\GeneratedValue()`
- `@ORM\Column(type="integer")`
- `@ORM\Column(type="string")`
- `@ORM\Column(type="date", nullable=true)`

Resource configuration

Configure the Post entity as a Sylius Resource via. the ResourceBundle.

1. Place file in `app/config/wsc/resources.yml`
2. Configure the model class to be that of the previously created `Post` entity.
3. Include it from `app/config.yml`

Hints:

- `php app/console config:dump-reference sylius_resource`
- `imports: [ { resource: "wsc/resources.yml" } ]`

REST CRUD configuration

Add route configuration which enables a REST API.

1. Create the routing configuration file in `app/config/wsc/routing_api.yml`
2. Import the new routing file from the main `app/config/routing.yml`
3. Configure the FOS REST bundle
4. Create a new resource with the `curl` command.
5. View the resource with the `curl` command.

Hints:

```
app_admin_post_api:
    resource: |
        alias: <resource name>
        except: [ "index" ] # do not generate these routes
        <something here>
    type: sylius.resource_api
    prefix: <route prefix, e.g. admin-api (do not use admin!)>
```

```
$ curl -i -X POST -H "Content-Type: application/json" -d '{"title": "...", ...}' http://127.0.0.1:8000/<your
```

```
$ curl -i -X GET -H "Content-Type: application/json" http://127.0.0.1:8000/app-api/posts/1
```

```
section: api
```

```
fos_rest:
    # ...
    format_listener:
        rules:
            # ...
            -
                path: '^/<route containing your API>'
                priorities: ['json', 'xml'] # enable JSON and XML
                fallback_format: json       # default to JSON
                prefer_extension: true
```

HTML CRUD configuration

Create the routing for the HTML rest interface.

1. Create the routing configuration for the resource in `app/config/wsc/routing.yml`
2. Include the routing from `app/config/routing.yml`
3. Create a new post
4. Verify the new routes with the console.
5. Try creating a new resource (and expect an exception thereafter).

Hints:

```
<route name>: # include "admin" to avoid conflicts later
    resource: |
        alias: <resource name>
        except: [ "show" ] # do not generate this route
        templates: <template set to use>
    type: sylius.resource
    prefix: <route prefix, e.g. admin>
```

```
SyliusAdminBundle:Crud
```

```
$ php app/console debug:router # grep is your friend
```

Grid Configuration

Configure a grid for listing posts (the posts index page).

1. Create the grid configuration file in `app/config/wsc/grids.yml`
2. Add fields for the `title` and `publishedAt` properties of the `Post` entity.
3. Enable actions for creating, updating and deleting entries.
4. Associate the grid with the resource in `app/config/wsc/routing.yml`
5. Import the file from the main `app/config/config.yml` file.
6. Navigate to `http://127.0.0.1:8000/admin/posts`
7. Create, update and delete blog posts!

Hints:

```
$ php app/console config:dump-reference sylius_grid
```

```yaml
sylius_grid:
    grids:
        <grid name, e.g. app_post>:
            driver:
                options:
                    class: <class name of the post entity>
            fields:
                <property name for title>:
                    type: string
                <property name for published date>:
                    type: datetime
                    options:
                        format: Y-m-d
```

```yaml
            actions:
                main:
                    create:
                        type: create
                item:
                    update:
                        type: update
                    delete:
                        type: delete
```

```
# routing.yml
<...>:
     resource: |
         # ...
+        grid: app_post
         # ...
     # ...
```

Backend Menu

Create a menu item in the admin interface.

1. Create the menu listener.
2. Add the menu listener to the DI configuration.

Hints:

```php
<?php

// ... don't forget the namespace!

use Sylius\Bundle\UiBundle\Menu\Event\MenuBuilderEvent;

class <for example, AdminMenuListener>
{
    public function <method name>(MenuBuilderEvent $event)
    {
        $menu = $event->getMenu();

        $blogMenu = $menu
            ->addChild('blog')
            ->setLabel('Blog');
        $blogMenu
            ->addChild('posts', [ 'route' => '<name of the index route for the
            posts>' ])
            ->setLabel('Posts');
    }
}
```

```
services:
    <name of menu listener service, not important>:
        class: <full class name of your listener>
        tags:
            -
                name: kernel.event_listener
                event: sylius.menu.admin.main
                method: <your method name>
```

Create a frontend controller

Create a controller and template for listing the blog posts on the front end.

1. Create the controller.
2. Create the DI configuration for the controller
3. Create the template
4. Add routing for the controller (actually, this must be prefixed with *something* to avoid conflicts with Sylius).
5. Copy the shop layout and override it, adding a link to your blog post list.

```php
<?php

namespace AppBundle\Controller;

use Symfony\Bundle\FrameworkBundle\Templating\EngineInterface;
use Doctrine\ORM\EntityManagerInterface;
use AppBundle\Entity\Post;
use Symfony\Component\HttpFoundation\Request;

class PostController
{
    private $entityManager;
    private $templating;

    public function __construct(
        EntityManagerInterface $entityManager,
        EngineInterface $templating
    )
    {
        $this->entityManager = $entityManager;
        $this->templating = $templating;

    }

    public function indexAction(Request $request)
    {
        $repository = $this->entityManager->getRepository(Post::class);
        $posts = $repository->findAll();

        return $this->templating->render(
            '@App/Post/index.html.twig',
            [
                'posts' => $posts
            ]
        );
    }
}
```

```twig
{% extends "SyliusShopBundle::layout.html.twig" %}

{% block content %}

    {% for post in posts %}
        {# display your post here. be creative. #}
    {% endfor %}

{% endblock %}
```

```
<layout template path here>
```

```
app.controller.admin_post:
    class: <full class name for your controller>
    arguments:
        - @doctrine.orm.default_entity_manager
        - @templating
```