Examples

Practical examples for Derafu Auth

Anonymous · 09/09/2026

Examples

Minimal Setup

```
<?php

require_once 'vendor/autoload.php';

use Derafu\Auth\Configuration\AuthConfiguration;
use Derafu\Auth\Service\KeycloakAuthenticationService;
use Derafu\Auth\Service\SessionService;
use Derafu\Auth\Validator\RouteValidator;
use Derafu\Auth\Middleware\AuthenticationMiddleware;

$config = new AuthConfiguration([
    'keycloak_url' => 'http://localhost:8080',
    'client_id' => 'your-client-id',
    'client_secret' => 'your-client-secret',
    'redirect_uri' => 'http://localhost/auth/callback',
]);

$authService = new KeycloakAuthenticationService($config);
$sessionService = new SessionService($config);
$routeValidator = new RouteValidator($config);

$authMiddleware = new AuthenticationMiddleware(
    $authService,
    $sessionService,
    $routeValidator
);
```

API Examples

REST API with Authentication

```
class UserApiController
{
```

```
    public function getUsers(ServerRequestInterface $request): ResponseInterface
    {
        $user = $request->getAttribute('user');

        if (!$user) {
            return new JsonResponse(['error' => 'Not authenticated'], 401);
        }

        if (!$this->isAuthorized($user, 'user:read')) {
            return new JsonResponse(['error' => 'Access denied'], 403);
        }

        $users = $this->getUserList();

        return new JsonResponse(['users' => $users]);
    }

    private function isAuthorized(array $user, string $permission): bool
    {
        $userPermissions = $user['permissions'] ?? [];

        return in_array($permission, $userPermissions);
    }
}
```

GraphQL with Authentication

```
use GraphQL\Type\Definition\ObjectType;
use GraphQL\Type\Definition\Type;
use GraphQL\Type\Schema;

class AuthenticatedGraphQLSchema
{
    public function createSchema(): Schema
    {
        $userType = new ObjectType([
            'name' => 'User',
            'fields' => [
                'id' => Type::string(),
                'email' => Type::string(),
                'name' => Type::string(),
                'roles' => Type::listOf(Type::string()),
            ],
        ]);

        $queryType = new ObjectType([
            'name' => 'Query',
            'fields' => [
                'me' => [
                    'type' => $userType,
```

```
                    'resolve' => function ($root, $args, $context) {
                        $user = $context['user'] ?? null;

                        if (!$user) {
                            throw new \Exception('Not authenticated');
                        }

                        return $user;
                    },
                ],
            ],
        ]);

        return new Schema([
            'query' => $queryType,
        ]);
    }
}
```

Testing Example

```
use PHPUnit\Framework\TestCase;
use Derafu\Auth\Middleware\AuthenticationMiddleware;

class AuthenticationMiddlewareTest extends TestCase
{
    public function testProtectedRouteRequiresAuthentication()
    {
        $config = new AuthConfiguration([
            'keycloak_url' => 'http://localhost:8080',
            'client_id' => 'test-client',
            'client_secret' => 'test-secret',
            'redirect_uri' => 'http://localhost/auth/callback',
            'protected_routes' => ['/dashboard'],
        ]);

        $authService = $this->createMock(KeycloakAuthenticationService::class);
        $sessionService = $this->createMock(SessionService::class);
        $routeValidator = new RouteValidator($config);

        $middleware = new AuthenticationMiddleware(
            $authService,
            $sessionService,
            $routeValidator
        );

        $request = $this->createMock(ServerRequestInterface::class);
        $request->method('getUri')->willReturn(new Uri('/dashboard'));
```

```
        $handler = $this->createMock(RequestHandlerInterface::class);

        $response = $middleware->process($request, $handler);

        $this->assertEquals(302, $response->getStatusCode()); // Redirect to login.
    }
}
```

Real-World Examples

E-commerce Application

```
class EcommerceAuthController
{
    public function handleLogin(ServerRequestInterface $request): ResponseInterface
    {
        $user = $request->getAttribute('user');

        if (!$user) {
            return new JsonResponse(['error' => 'Not authenticated'], 401);
        }

        if (!$this->hasPermission($user, 'shop:access')) {
            return new JsonResponse(['error' => 'Access denied'], 403);
        }

        $cart = $this->getUserCart($user['sub']);

        return new JsonResponse([
            'user' => $user,
            'cart' => $cart,
            'permissions' => $user['permissions'] ?? [],
        ]);
    }

    private function hasPermission(array $user, string $permission): bool
    {
        $permissions = $user['permissions'] ?? [];

        return in_array($permission, $permissions);
    }
}
```

Admin Panel

```
class AdminPanelController
{
```

```
    public function handleDashboard(ServerRequestInterface $request):
ResponseInterface
    {
        $user = $request->getAttribute('user');

        if (!$user) {
            return new JsonResponse(['error' => 'Not authenticated'], 401);
        }

        if (!$this->isAdmin($user)) {
            return new JsonResponse(['error' => 'Access denied'], 403);
        }

        $stats = $this->getAdminStats();

        return new JsonResponse([
            'user' => $user,
            'stats' => $stats,
        ]);
    }

    private function isAdmin(array $user): bool
    {
        $roles = $user['roles'] ?? [];

        return in_array('admin', $roles) || in_array('super_admin', $roles);
    }
}
```

Last updated on 09/09/2026