

THE INITIAL DESCRIPTION OF THE OBJECT OF INTELLECTUAL PROPERTY
SPACEFEX - Common Project Management Platform Source Code

<?php

```
namespace SORQPAY\Modules\AemiDashboard\Http\Controllers\Customers;

use Illuminate\Contracts\Pagination\LengthAwarePaginator;
use Illuminate\Http\JsonResponse;
use SORQPAY\Modules\AemiDashboard\Http\Controllers\AbstractAemiDashboardController;
use SORQPAY\Modules\AemiDashboard\Services\Customers\AmlService;
use SORQPAY\Modules\Users\Models\Profile\BusinessProfile;
use SORQPAY\Modules\AemiDashboard\Http\Requests\UserLevelRequest;
use SORQPAY\Modules\Users\Repositories\Profile\BusinessProfileRepository;
use SORQPAY\Modules\Users\Services\Dev\UserDeleteService;
use SORQPAY\Modules\Users\Repositories\UserLevelRepository;
use SORQPAY\Modules\Users\Repositories\BusinessSectorRepository;
use SORQPAY\Modules\AemiDashboard\Http\Resources\Customer\CustomerUserLevelResourceCollection;
use
SORQPAY\Modules\AemiDashboard\Http\Resources\Customer\CustomerBusinessSectorResourceCollection;
use Illuminate\Http\Request;

/**
 * Class AbstractCustomersController
 * @package SORQPAY\Modules\AemiDashboard\Http\Controllers\Customers
 */
abstract class AbstractCustomersController extends AbstractAemiDashboardController
{
    /**
     * @var string $type
     */
    protected static $type;
    /**
     * @var BusinessProfileRepository $businessProfileRepository
     */
    protected $businessProfileRepository;
    /**
     * @var AmlService $amlService
     */
    protected $amlService;

    /**
     * @param BusinessProfileRepository $businessProfileRepository
     * @param AmlService $amlService
     */
    public function __construct(BusinessProfileRepository $businessProfileRepository, AmlService $amlService)
    {
        $this->businessProfileRepository = $businessProfileRepository;
        $this->amlService = $amlService;
    }

    /**
     * @OA\Get(
     *   path="/api/en-GB/v1/aemi/dashboard/customers/company/user-level/{business_profile_id}",
     *   operationId="GetUserLevelCustomer",
     *   tags={"Customers"},
     *   summary="Get user_level customer",
     *   description="Get user_level customer. Value by default is 0. Also you can get user_level and other business
     profile info via API command /api/en-GB/v1/aemi/dashboard/customers/company/{business_profile_id}",

```

```

* @OA\Response(
*   response=200,
*   description="Successful operation",
*   @OA\JsonContent(
*     type= "object",
*     @OA\Property( property="level", type="integer", example="123" )
*   )
* ),
* @OA\Response(
*   response=401,
*   description="Unauthenticated",
* ),
* @OA\Response(
*   response=403,
*   description="Forbidden"
* ),
* @OA\Response(
*   response=404,
*   description="Business profile id is incorrect or not found",
* ),
* )
*
* @param BusinessProfile $businessProfile
* @return JsonResponse
*/
protected function getUserLevel(BusinessProfile $businessProfile): JsonResponse
{
    return $this->responseOk([
        'level' => $businessProfile->getUserLevel()
    ]);
}

/**
* @OA\Put(
*   path="/api/en-GB/v1/aemi/dashboard/customers/company/user-level/{business_profile_id}",
*   operationId="PutUserLevelCustomer",
*   tags={"Customers"},
*   summary="Set user_level customer",
*   description="Set user_level customer",
*   @OA\RequestBody(
*     request="setUserLevelNumber",
*     required=true,
*     description="User level value must be a number",
*     @OA\JsonContent(
*       type="object",
*       @OA\Property(property="level", type="integer", example="3")
*     )
*   ),
*   @OA\Response(
*     response=200,
*     description="Successful operation",
*     @OA\JsonContent(
*       type= "object",
*       @OA\Property( property="message", type="integer", example="User level updated" )
*     )
*   ),
*   @OA\Response(
*     response=401,
*     description="Unauthenticated",
*   ),
*   @OA\Response(
*     response=403,
*     description="Forbidden"

```

```
* ),  
* @OA\Response(  
*   response=404,  
*   description="Business profile id is incorrect or not found",  
* ),  
* @OA\Response(  
*   response=422,  
*   description="Invalid input parameters given.",  
*   @OA\JsonContent(  
*     @OA\Property(property="error", type="object",  
*       @OA\Property(property="level", type="array",  
*         @OA\Items(type="string", example="The level must be a number."),  
*     )  
*   ),  
* ),  
* ),  
*)  
  
* @param BusinessProfile $businessProfile  
* @param UserLevelRequest $request  
* @return JsonResponse  
*/  
  
protected function setUserLevel(BusinessProfile $businessProfile, UserLevelRequest $request): JsonResponse  
{  
    $businessProfile->user_level = $request->get('level');  
    $businessProfile->save();  
  
    return $this->responseOk("User level updated");  
}  
  
/**  
 * @OA\Get(  
 *   path="/api/en-GB/v1/aemi/dashboard/customers/company/user-levels?sort=created_at|title|value",  
 *   operationId="Get information about user level's",  
 *   tags={"Customers"},  
 *   summary="Get information about user level's",  
 *   description="Get information about user level's. In case ?sort= parameter was absent sort by value and by  
asc will be applied. ?sort=-value sort by desc.",  
 *   @OA\Response(  
 *     response=200,  
 *     description="Successful operation",  
 *     @OA\JsonContent(  
 *       type= "object",  
 *       @OA\Property(property="data", type="array", example={  
 *         {  
 *           "key": "6230abf5893d8e711e675882",  
 *           "title": "Full",  
 *           "value": 0,  
 *           "updated_at": "2022-03-15T15:08:37.076000Z",  
 *           "created_at": "2022-03-15T15:08:37.076000Z"  
 *         },  
 *         {  
 *           "key": "6230abf5893d8e711e675883",  
 *           "title": "Full-investigated",  
 *           "value": 1,  
 *           "updated_at": "2022-03-15T15:08:37.076000Z",  
 *           "created_at": "2022-03-15T15:08:37.076000Z"  
 *         },  
 *         {  
 *           "key": "6230abf5893d8e711e675886",  
 *           "title": "Limited",  
 *           "value": 2,  
 *           "updated at": "2022-03-15T15:08:37.076000Z",
```

```

*         "created_at": "2022-03-15T15:08:37.076000Z"
*     },
*     {
*         "key": "6230abf5893d8e711e675888",
*         "title": "Denied",
*         "value": 3,
*         "updated_at": "2022-03-15T15:08:37.076000Z",
*         "created_at": "2022-03-15T15:08:37.076000Z"
*     },
* },
* ),
* @OA\Items()
* ),
*         @OA\Property(property="links", type="object",
*         @OA\Property(property="first", type="string", example="https://testapi.quatroit.com/api/en-
GB/v1/aemi/dashboard/customers/company/user-levels?page=1"),
*         @OA\Property(property="last", type="string", example="https://testapi.quatroit.com/api/en-
GB/v1/aemi/dashboard/customers/company/user-levels?page=1"),
*         @OA\Property(property="prev", type="string", example="null"),
*         @OA\Property(property="next", type="string", example="null")
*     ),
* @OA\Property(property="meta", type="object",
*     example={
*         "current_page": 1,
*         "from": 1,
*         "last_page": 1,
*         "links": {
*             {
*                 "url": null,
*                 "label": "« Previous",
*                 "active": false
*             },
*             {
*                 "url": "https://testapi.quatroit.com/api/en-GB/v1/aemi/dashboard/customers/company/user-
levels?page=1",
*                 "label": "1",
*                 "active": true
*             },
*             {
*                 "url": null,
*                 "label": "Next »",
*                 "active": false
*             }
*         }
*     },
*     "path": "https://testapi.quatroit.com/api/en-GB/v1/aemi/dashboard/customers/company/user-levels",
*     "per_page": 15,
*     "to": 4,
*     "total": 4
* },
* ),
* ),
* @OA\Response(
*     response=401,
*     description="Unauthenticated",
* ),
* @OA\Response(
*     response=403,
*     description="Forbidden"
* )
* )
*
* @return CustomerUserLevelResourceCollection
* @throws \Illuminate\Contracts\Container\BindingResolutionException

```

```

*/
protected function getUserLevels(): CustomerUserLevelResourceCollection
{
    return new CustomerUserLevelResourceCollection(
        app()->make(UserLevelRepository::class)->getUserLevelsPaginated()
    );
}

/**
 * @OA\Get(
 *     path="/api/en-GB/v1/aemi/dashboard/customers/company/business-
sectors?sort=created_at|title|sector_id",
 *     operationId="Get business sector's data",
 *     tags={"Customers"},
 *     summary="Get business sector's data",
 *     description="Get business sector's data. In case ?sort= parameter was absent sort by sector_id and by asc
will be applied. ?sort=-sector_id sort by desc.",
 *     @OA\Response(
 *         response=200,
 *         description="Successful operation",
 *         @OA\JsonContent(
 *             type= "object",
 *             @OA\Property(property="data", type="array", example={
 *                 {
 *                     "key": "6230abf5893d8e711e675882",
 *                     "title": "Machinery, motor vehicle, aircraft or ship manufacture/wholesale",
 *                     "sector_id": 0,
 *                     "updated_at": "2022-03-15T15:08:37.076000Z",
 *                     "created_at": "2022-03-15T15:08:37.076000Z"
 *                 },
 *                 {
 *                     "key": "6230abf5893d8e711e675883",
 *                     "title": "Marketing and Media",
 *                     "sector_id": 1,
 *                     "updated_at": "2022-03-15T15:08:37.076000Z",
 *                     "created_at": "2022-03-15T15:08:37.076000Z"
 *                 }
 *             },
 *             @OA\Items()
 *         ),
 *     ),
 *     @OA\Response(
 *         response=401,
 *         description="Unauthenticated",
 *     ),
 *     @OA\Response(
 *         response=403,
 *         description="Forbidden"
 *     )
 * )
 *
 * @return CustomerBusinessSectorResourceCollection
 * @throws \Illuminate\Contracts\Container\BindingResolutionException
 */
protected function getBusinessSectors(): CustomerBusinessSectorResourceCollection
{
    return new CustomerBusinessSectorResourceCollection(
        app()->make(BusinessSectorRepository::class)->getBusinessSectors()
    );
}

/**

```

```

*
* @OA\Get(
*   path="/api/en-GB/v1/aemi/dashboard/customers/{business_profile_id}",
*   operationId="Get info by business_profile_id.",
*   tags={"Customers"},
*   deprecated=true,
*   summary="Get info by business_profile_id.",
*   description="Get info by business_profile_id.",
*   @OA\Response(
*     response=200,
*     description="Successful operation",
*     @OA\JsonContent(
*       type="object",
*       @OA\Property(property="data", type="object", example={
*         "id": "13ada19a-73c3-4413-962d-10657f941134",
*         "personal_profile": {
*           "_id": "623aee20dd550c7fd545768a",
*           "verification_status": "NOT_STARTED",
*           "ca_checked": true,
*           "verified": true,
*           "in_use": true,
*           "status": "CREATED",
*           "blocked": false,
*           "user_id": "13ada19a-73c3-4413-962d-10657f941134",
*           "email": "ntest122212121@test.com",
*           "country_phone_code": "380",
*           "phone": "677745665",
*           "business_profile_id": "623aee20dd550c7fd5457689",
*           "title": "Mr",
*           "first_name": "ffdfdf",
*           "middle_name": "fdf",
*           "last_name": "dfdfdf",
*           "profile_type": "DIRECTOR",
*           "updated_at": "2022-03-23T09:56:00.038000Z",
*           "created_at": "2022-03-23T09:53:36.877000Z",
*           "address1": "hghghgh",
*           "address2": null,
*           "city": "hhhh",
*           "country": "AF",
*           "country_of_birth": "AF",
*           "date_of_birth": "2004-02-04",
*           "politically_exposed": "0",
*           "post_code": "ghgh",
*           "us_person": "0",
*           "jumio": null,
*           "comply_advantage": null,
*           "key": "623aee20dd550c7fd545768a"
*         },
*         "business_profile": {
*           "_id": "623aee20dd550c7fd5457689",
*           "on_boarding_status": "ACCEPTED",
*           "modulr_enabled": false,
*           "cj_enabled": false,
*           "afex_enabled": false,
*           "user_level": 0,
*           "user_id": "13ada19a-73c3-4413-962d-10657f941134",
*           "profile_type": "COMPANY",
*           "reference_code": "SORQPAY",
*           "updated_at": "2022-03-30T13:46:53.653000Z",
*           "created_at": "2022-03-23T09:53:36.873000Z",
*           "address1": "fgfgfg",
*           "address2": null,
*           "business_adult": "0",

```

```

*      "business_art": "0",
*      "business_chemical": "0",
*      "business_cryptocurrency_processing": "0",
*      "business_fx_speculators": "0",
*      "business_gambling": "0",
*      "business_money_processing": "0",
*      "business_nonprofit": "0",
*      "business_precious_metals": "0",
*      "business_sector": "gfgfgfg",
*      "business_type": "LTD",
*      "business_used_cars": "0",
*      "business_weapons": "0",
*      "city": "ggg",
*      "company_name": "ggfgfg",
*      "contact_person": true,
*      "country": "AF",
*      "country_of_incorporation": "AF",
*      "country_phone_code": "380",
*      "email": "ntest122212121@test.com",
*      "first_name": "ffdfdf",
*      "have_financial_institution_status": "0",
*      "have_us_person": "0",
*      "job_title": "DIRECTOR",
*      "last_name": "dfdfdf",
*      "middle_name": "fdf",
*      "operational_address1": null,
*      "operational_address2": null,
*      "operational_city": null,
*      "operational_country": null,
*      "operational_post_code": null,
*      "operational_same": true,
*      "phone": "677745665",
*      "post_code": "fgfgfg",
*      "registration_date": "2022-03-01",
*      "registration_number": "ggfgfg",
*      "tax_id_number": "ggfgfg",
*      "trade_name": "gfgfgg",
*      "website": "fgfggffgfg.com",
*      "business_description": "gggggg",
*      "currency": "EUR",
*      "first_payment_amount": "44",
*      "first_payment_country_code": "AF",
*      "first_payment_currency": "EUR",
*      "first_payment_name": "fgfg",
*      "first_payment_purpose": "fgfgfg",
*      "monthly_turnover": "55",
*      "number_of_payments": "55",
*      "specify": "ggfgfg",
*      "pricing_plan_assigned_at": "2022-03-23T09:56:44.318000Z",
*      "pricing_plan_id": "61c85579bda95c24f30b4373",
*      "customer_id": 10039,
*      "credit_safe": null,
*      "ibanq_enabled": true,
*      "key": "623aee20dd550c7fd5457689"
*  },
*  "directors": {
*      {
*          "_id": "623aee20dd550c7fd545768a",
*          "verification_status": "NOT_STARTED",
*          "ca_checked": true,
*          "verified": true,
*          "in_use": true,
*          "status": "CREATED",

```

```

*         "blocked": false,
*         "user_id": "13ada19a-73c3-4413-962d-10657f941134",
*         "email": "ntest122212121@test.com",
*         "country_phone_code": "380",
*         "phone": "677745665",
*         "business_profile_id": "623aee20dd550c7fd5457689",
*         "title": "Mr",
*         "first_name": "ffdfdf",
*         "middle_name": "fdf",
*         "last_name": "dfdfdf",
*         "profile_type": "DIRECTOR",
*         "updated_at": "2022-03-23T09:56:00.038000Z",
*         "created_at": "2022-03-23T09:53:36.877000Z",
*         "address1": "hghghgh",
*         "address2": null,
*         "city": "hhhh",
*         "country": "AF",
*         "country_of_birth": "AF",
*         "date_of_birth": "2004-02-04",
*         "politically_exposed": "0",
*         "post_code": "ghgh",
*         "us_person": "0",
*         "jumio": null,
*         "comply_advantage": null,
*         "key": "623aee20dd550c7fd545768a"
*     }
* },
*     "shareholders": {
*         {
*             "id": "623aeea54bd196137a259e37",
*             "verification_status": "NOT_STARTED",
*             "ca_checked": true,
*             "verified": false,
*             "in_use": false,
*             "status": "CREATED",
*             "blocked": false,
*             "first_name": "hhhg",
*             "middle_name": "hghgh",
*             "last_name": "ghghgh",
*             "date_of_birth": "2004-02-18",
*             "ownership_stake": "99",
*             "tax_id_number": "gfgfg",
*             "is_ubo": false,
*             "country": "AF",
*             "city": "ggfgfg",
*             "post_code": "fgfgfg",
*             "address1": "fgfgfg",
*             "address2": null,
*             "country_phone_code": "380",
*             "phone": "674455446",
*             "email": "dfdfdfdf2@dfdfdfd.com",
*             "participation_in_other_business": null,
*             "politically_exposed": "0",
*             "us_person": "0",
*             "business_profile_id": "623aee20dd550c7fd5457689",
*             "profile_type": "SHAREHOLDER",
*             "updated_at": "2022-03-23T09:55:59.623000Z",
*             "created_at": "2022-03-23T09:55:49.577000Z",
*             "key": "623aeea54bd196137a259e37",
*             "jumio": null,
*             "comply_advantage": null
*         }
*     },
* },

```

```

*         "ubos": {},
*         "partners": {
*             {
*                 "_id": "623aee63dd550c7fd545768b",
*                 "partner_type": "incoming",
*                 "partner_name": "gggfgfg",
*                 "partner_purpose": "fgfgfg",
*                 "partner_country": "AF",
*                 "partner_monthly_turnover": "555",
*                 "partner_currency": "EUR",
*                 "partner_business_profile_id": "623aee20dd550c7fd5457689",
*                 "updated_at": "2022-03-23T09:54:43.673000Z",
*                 "created_at": "2022-03-23T09:54:43.673000Z",
*                 "key": "623aee63dd550c7fd545768b"
*             },
*             {
*                 "_id": "623aee6b9c62782ca5599225",
*                 "partner_type": "outgoing",
*                 "partner_name": "hhhghgh",
*                 "partner_purpose": "ghghgh",
*                 "partner_country": "AF",
*                 "partner_monthly_turnover": "5555",
*                 "partner_currency": "EUR",
*                 "partner_business_profile_id": "623aee20dd550c7fd5457689",
*                 "updated_at": "2022-03-23T09:54:51.437000Z",
*                 "created_at": "2022-03-23T09:54:51.437000Z",
*                 "key": "623aee6b9c62782ca5599225"
*             }
*         },
*         "blocked": false
*     }
* ),
* ),
* @OA\Response(
*     response=401,
*     description="Unauthenticated",
* ),
* @OA\Response(
*     response=403,
*     description="Forbidden"
* )
* )
*
* @param BusinessProfile $businessProfile
* @return JsonResponse
*/
protected function show(BusinessProfile $businessProfile): JsonResponse
{
    if (!$businessProfile->getUser()) {
        return $this->responseInternalServerError('Broken data, linked user record does not exist');
    }

    return $this->responseOk([
        'data' => $this->amlService->getAmlPageDataForCustomer($businessProfile->getUser())
    ]);
}

/**
 * @OA\Put(
 *     path="/api/en-GB/v1/aemi/dashboard/customers/block/{business_profile_id}",
 *     operationId="BlockingCustomer",

```

```

*   tags={"Customers"},
*   summary="Blocks customer",
*   description="Blocks customer",
*   @OA\Response(
*       response=200,
*       description="Successful operation",
*       @OA\JsonContent(
*           type= "object",
*           @OA\Property( property="message", type="string", example="User has blocked." )
*       )
*   ),
*   @OA\Response(
*       response=401,
*       description="Unauthenticated",
*   ),
*   @OA\Response(
*       response=403,
*       description="Forbidden"
*   )
* )
* @param BusinessProfile $businessProfile
*
* @return JsonResponse
*/
protected function blockUser(BusinessProfile $businessProfile): JsonResponse
{
    $businessProfile->setBlocked(true)
        ->save();

    return $this->responseOk([
        'message' => 'User has blocked.',
    ]);
}

/**
* @OA\Put(
*   path="api/en-GB/v1/aemi/dashboard/customers/unblock/{business_profile_id}",
*   operationId="UnblockingCustomer",
*   tags={"Customers"},
*   summary="Unblocks customer",
*   description="Unblocks customer",
*   @OA\Response(
*       response=200,
*       description="Successful operation",
*       @OA\JsonContent(
*           type= "object",
*           @OA\Property( property="message", type="string", example="User has unblocked." )
*       )
*   ),
*   @OA\Response(
*       response=401,
*       description="Unauthenticated",
*   ),
*   @OA\Response(
*       response=403,
*       description="Forbidden"
*   )
* )
*
* @param BusinessProfile $businessProfile
*
* @return JsonResponse
*/

```

```

protected function unblockUser(BusinessProfile $businessProfile): JsonResponse
{
    $businessProfile->setBlocked(false)
        ->save();

    return $this->responseOk([
        'message' => 'User has unblocked.',
    ]);
}

/**
 * @param Request $request
 * @return LengthAwarePaginator
 */
public function index(Request $request)
{
    return $this->businessProfileRepository
        ->getPaginatedByProfileType(static::$type);
}

/**
 * @OA\Delete(
 *     path="/api/en-GB/v1/aemi/dashboard/customers/{business_profile_id}",
 *     operationId="DeleteCustomers",
 *     tags={"Customers"},
 *     summary="Delete customer",
 *
 *     @OA\Response(
 *         response=200,
 *         description="Successful operation",
 *         @OA\JsonContent(
 *             type="object",
 *             @OA\Property(property="message", type="string", example="User has deleted." )
 *         )
 *     ),
 *     @OA\Response(
 *         response=401,
 *         description="Unauthenticated",
 *     ),
 *     @OA\Response(
 *         response=403,
 *         description="Forbidden"
 *     )
 * )
 *
 * @param BusinessProfile $businessProfile
 * @return JsonResponse
 * @throws \Illuminate\Contracts\Container\BindingResolutionException
 */
public function destroy(BusinessProfile $businessProfile)
{
    app()->make(UserDeleteService::class)->delete($businessProfile->getUser());

    return $this->responseOk([
        'message' => 'User has deleted.',
    ]);
}

```