<<subsystem>>
rest_framework

BasePersmission

+ has_permission(request: Request, view: View): bool

+ has_object_permission(request: Request, view: View, obj: Object): bool

permission_classes

AllowAny

+ has_permission(request: Request, view: View)

IsAuthenticated

+ has_permission(request: Request, view: View)

IsManager

+ has_permission(request: Request, view: View)

IsNormalUser

+ has_permission(request: Request, view: View)

IsCompanyManager

+ has_permission(request: Request, view: View)

IsTechnicalManager

+ has_permission(request: Request, view: View)

IsCustomer

+ has_permission(request: Request, view: View)

IsSpecialist

+ has_permission(request: Request, view: View)

BasicUserPermission

PermissionFactory

- user_type : UserRole

+ __init__(user_type)

+ get_permission_class(): Type[BasicUserPermission]

IsReadOnlyRequest

+ has_permission(request: Request, view: View)

IsPostRequest

+ has_permission(request: Request, view: View)

Request

- _user: AbstractUser

+ __init__(request: HttpRequest)

+ user(): AbstractUser

Response

- status: int

+ __init__(data: Dict, status: int)

+ rendered_content()

View

+ __init__()

+ as_view(): function

+ setup()

+ dispatch()

+ options()

- _allowed_methods(): String[]

APIView

- permission_classes: BasePermission[]

+ __init__()

+ as_view(): function

+ dispatch(): Response

+ options(): Response

+ allowed_methods(): String[]

+ get_render_context(): Dict[String, Object]

+ perform_authentication(request:Request): User

+ check_permissions(request:Request)

+ initialize_request(request:HttpRequest): Request

+ initial(request:Request)

+ handle_exception(exc: Exception): Response

+ finalize_response(request:Request, response:Response): Response

<<use>>
<<use>>