

API-k és mikroszolgáltatások	
Projekt: Navigációs API	GitHub: https://github.com/trolland97/navigation
Projektvezető: Dr. Bilicki Vilmos / Gémes Tamás	
Projektet készítette: Péter Roland	Neptun kód: PYJY2U

OpenAPI elemek megvalósítása	Pontszám	Megjegyzés
<ul style="list-style-type: none"> JSON támogatás 	1	swagger.yaml line: 14
<ul style="list-style-type: none"> Swagger request és response validation használata 	1	swagger.yaml line: 32
<ul style="list-style-type: none"> GET, PUT, POST, DELETE végpontok 	3	swagger.yaml line: 86 swagger.yaml line: 145 swagger.yaml line: 58 swagger.yaml line: 176
<ul style="list-style-type: none"> Újrahasznosítható objektum definíciók használata minden végponthoz 	1	swagger.yaml line: 201
<ul style="list-style-type: none"> Required mezők használata az objektum definíciókban és paraméterekben 	1	swagger.yaml line: 154 swagger.yaml line: 202
<ul style="list-style-type: none"> Query string paraméter használat lekérdezésnél 	1	swagger.yaml line: 121
<ul style="list-style-type: none"> Hibakezelés: Közös hiba definíció használata, minden endpoint által használva, controllerekben megvalósítva 	2	swagger.yaml line: 235
<ul style="list-style-type: none"> Hibakezelés: HTTP hibakódok használata különböző hiba esetekre (pl. hiányzó objektum, autentikációs hiba, hiányzó jogosultság, egyéb szerver hiba), controllerekben megvalósítva 	2	swagger.yaml line: 76
<ul style="list-style-type: none"> Tag-ek használata végpontok csoportosítására 	1	swagger.yaml line: 15
API autentikáció - Swagger Security		

Péter Roland

<ul style="list-style-type: none"> • Globális session alapú autentikáció minden endpointra (API Key) 	5	swagger.yaml line: 29 swagger_security.js line: 6
<ul style="list-style-type: none"> • Login és Signup végpontok, autentikációs kivételekkel 	2	swagger.yaml line: 61 swagger.yaml line: 37
API üzleti logika - Controllers		
<ul style="list-style-type: none"> • Adattárolás (in-memory vagy perzisztens) használata 	2	Db.js
<ul style="list-style-type: none"> • A választott téma szerinti logika megvalósítása (számítás, nem csak CRUD) 	6	Distance_calculator.js
API Gateway használat - Kong		
<ul style="list-style-type: none"> • API Gateway használata reverse proxy-ként 	3	kong_config.sh line: 15
<ul style="list-style-type: none"> • API Key használata kliens azonosításra (mobil, web) 	4	kong_config.sh line: 35
<ul style="list-style-type: none"> • Rate limit használata (globális) 	3	kong_config.sh line: 23
<ul style="list-style-type: none"> • Rate limit használata (klienstől függő - Kong consumer) 	4	kong_config.sh line: 41
<ul style="list-style-type: none"> • Dinamikus terhelés (load balancing) 	4	kong_config.sh line: 11 kong_config.sh line: 7
Docker Compose		
<ul style="list-style-type: none"> • Működő Docker-compose deployment Kong-gal és OpenAPI-val (verzio min 3) 	5	Docker-compose.yml
<ul style="list-style-type: none"> • Verziózott Docker image-ek használata 	3	dockerhub
<ul style="list-style-type: none"> • Health check definiálása az API-hoz 	2	Docker-compose.yml