

OpenAPI elemek megvalósítása	Megjegyzés
<ul style="list-style-type: none"> • Szabványos swagger raw endpoint 	swagger.yaml line: 373
<ul style="list-style-type: none"> • JSON támogatás 	swagger.yaml line: 10, 12
<ul style="list-style-type: none"> • Swagger request és response validation használata 	swagger.yaml line: 30, 37
<ul style="list-style-type: none"> • GET, PUT, POST, DELETE végpontok 	GET: swagger.yaml line: 76 PUT: swagger.yaml line: 222 POST: swagger.yaml line: 24 DELETE: swagger.yaml line: 161
<ul style="list-style-type: none"> • Újrahasznosítható objektum definíciók használata minden végponthoz 	swagger.yaml line: 346
<ul style="list-style-type: none"> • Required mezők használata az objektum definíciókban és paraméterekben 	swagger.yaml line: 348 swagger.yaml line: 315
<ul style="list-style-type: none"> • Query string paraméter használat lekérdezésnél 	swagger.yaml line: 202
<ul style="list-style-type: none"> • Hibakezelés: Közös hiba definíció használata, minden endpoint által használva, controllerekben megvalósítva 	swagger.yaml line: 528
<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 	swagger.yaml line: 149
<ul style="list-style-type: none"> • Tag-ek használata végpontok csoportosítására 	swagger.yaml line: 304
<ul style="list-style-type: none"> • XML támogatás minimum 1 végpontra 	-

API autentikáció - Swagger Security	
<ul style="list-style-type: none"> • Globális session alapú autentikáció minden endpointra (API Key) 	swagger.yaml line: 14
<ul style="list-style-type: none"> • Login és Signup végpontok, autentikációs kivételekkel 	swagger.yaml line: 46 swagger.yaml line: 22
API üzleti logika - Controllers	
<ul style="list-style-type: none"> • Adattárolás (in-memory vagy perzisztens) használata 	app.js line: 28
<ul style="list-style-type: none"> • A választott téma szerinti logika megvalósítása (számítás, nem csak CRUD) 	taxiController.js line: 19, 61
API Gateway használat - Kong	
<ul style="list-style-type: none"> • API Gateway használata reverse proxy-ként 	kongconfig.sh line: 6
<ul style="list-style-type: none"> • API Key használata kliens azonosításra (mobil, web) 	kongconfig.sh line: 52, 56
<ul style="list-style-type: none"> • Rate limit használata (globális) 	kongconfig.sh line: 11
<ul style="list-style-type: none"> • Rate limit használata (klienstől függő - Kong consumer) 	kongconfig.sh line: 42, 47
<ul style="list-style-type: none"> • Dinamikus terhelés (load balancing) 	kongconfig.sh line: 22, 28
Docker Compose	
<ul style="list-style-type: none"> • Működő Docker-compose deployment Kong-gal és OpenAPI-val (verzio min 3) 	docker-compose.yml
<ul style="list-style-type: none"> • Verziózott Docker image-ek használata 	link
<ul style="list-style-type: none"> • Health check definiálása az API-hoz 	docker-compose.yml line: 64, 78