

Formai követelmények (kötelező):

- Kód github repository-ban az oktátónak elküldve
- Kubernetes telepítő fájl(ok) a repository-ban
- Kong API gateway konfigurációs scriptek
- Npm install és swagger project start segítségével az API hiba nélkül elindul lokális gépen
- Minikube-ban a Kubernetes telepítővel + Kong API gateway konfiguráció után Swagger UI-al vagy Postmannel megszólítható a projekt (+jelezni, hogy a hallgató melyikkel tesztelte)
- Dokumentáció: Az alábbi táblázat megjegyzés oszloppal min. 1 mondatos cellákkal, amiben dokumentálva van, hogy az adott követelmény megvalósítása melyik fájl(ok) mely soraiban található.

Maximális pontszám: 60 pont

<https://github.com/berkitamas/simplelight>

OpenAPI elemek megvalósítása	Pontszám	Hely
• Szabványos swagger raw endpoint	1	swagger.yaml - L873
• JSON támogatás	1	swagger.yaml - L21
• Swagger request és response validation használata	1	swagger.yaml - L290
• GET, PUT, POST, DELETE végpontok	4	swagger.yaml - L183 (GET) swagger.yaml - L210 (POST) swagger.yaml - L283 (PUT) swagger.yaml - L318 (DELETE)
• Újrahasznosítható objektum definíciók használata minden végponthoz	2	swagger.yaml - L896
• Required mezők használata az objektum definíciókban és paraméterekben	2	swagger.yaml - L616
• Query string paraméter használat lekérdezésnél	1	swagger.yaml - L817
• Hibakezelés: Közös hiba definíció használata, minden endpoint által	3	swagger.yaml - L1125

használva, controllerekben megvalósítva		
<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 	3	swagger.yaml - L662
<ul style="list-style-type: none"> Tag-ek használata végpontok csoportosítására 	1	swagger.yaml - L23
<ul style="list-style-type: none"> XML támogatás minimum 1 végpontra 	2	-
API autentikáció - Swagger Security		
<ul style="list-style-type: none"> Globális session alapú autentikáció minden endpointra (API Key) 	5	SwaggerSecurity.js
<ul style="list-style-type: none"> Login és Signup végpontok, autentikációs kivételekkel 	2	swagger.yaml - L53
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	parkings.js - L78
API Gateway használat - Kong		
<ul style="list-style-type: none"> API Gateway használata reverse proxy-ként 	3	kong.sh - L18
<ul style="list-style-type: none"> API Key használata kliens azonosításra (mobil, web) 	4	kong.sh - L33
<ul style="list-style-type: none"> Rate limit használata (globális) 	3	kong.sh - L28
<ul style="list-style-type: none"> Rate limit használata (klienstől függő - Kong consumer) 	4	kong.sh - L49
<ul style="list-style-type: none"> Dinamikus terhelés (load balancing) 	4	kong.sh - L3
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	simplelight/tags
<ul style="list-style-type: none"> Health check definiálása az API-hoz 	2	docker-compose.yml - L75

Választható témák:

- Mobil tárca API
- Különbéle böngészések és vásárlások megvalósítása (példa: Simple by OTP)
- Navigációs API
- Cél és forrás (szabad szöveges cím) megjelöléssel útvonal számítás
- A sessionben tárolja az aktuális helyszínt és forrás megjelölés nélkül azt használja
- Angry Birds API
- Egymás elleni játékként