

WALLET API (GNEPGM)

POSTMAN volt használva.

OpenAPI elemek megvalósítása	Pontszám	Megjegyzés
<ul style="list-style-type: none"> Szabványos swagger raw endpoint 	1	Swagger.yaml: 273-301 sor
<ul style="list-style-type: none"> JSON támogatás 	1	Swagger.yaml:11-16 sor
<ul style="list-style-type: none"> Swagger request és response validation használata 	1	Default.yaml: 20-31 sor
<ul style="list-style-type: none"> GET, PUT, POST, DELETE végpontok 	4	Swagger.yaml: 24-299 sor
<ul style="list-style-type: none"> Újrahasznosítható objektum definíciók használata minden végponthoz 	2	Swagger.yaml: 309-364 sor
<ul style="list-style-type: none"> Required mezők használata az objektum definíciókban és paraméterekben 	2	Swagger.yaml: 310-321 sor
<ul style="list-style-type: none"> Query string paraméter használat lekérdezésnél 	1	Swagger.yaml:25-52 sor
<ul style="list-style-type: none"> Hibakezelés: Közös hiba definíció használata, minden endpoint által használva, controllerekben megvalósítva 	3	Swagger.yaml: 355-364 sor
<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: 214-239 sor
<ul style="list-style-type: none"> Tag-ek használata végpontok csoportosítására 	1	Swagger.yaml 17-23 és 25-52 sor
<ul style="list-style-type: none"> XML támogatás minimum 1 végpontra 	2	Swagger.yaml:147-174 sor
API autentikáció - Swagger Security		
<ul style="list-style-type: none"> Globális session alapú autentikáció minden endpointra (API Key) 	5	Swagger.yaml: 305-308 és 121-122 sor App.js: 9-12 sor Swagger_security.js: 1-11 sor
<ul style="list-style-type: none"> Login és Signup végpontok, 	2	Swagger.yaml: 152-153 és

authentikációs kivételekkel		303-304 sor								
API üzleti logika - Controllers										
<ul style="list-style-type: none">Adattárolás (in-memory vagy perzisztens) használata	2	-								
<ul style="list-style-type: none">A választott téma szerinti logika megvalósítása (számítás, nem csak CRUD)	6	Controllerek: Movies.js: 24-43 sor Vignette.js: 24-43 sor								
API Gateway használat - Kong										
<ul style="list-style-type: none">API Gateway használata reverse proxy-ként	3	POSTMAN: url: http://localhost:8001/apis <table><tr><th>KEY</th><th>VALUE</th></tr><tr><td>name</td><td>walletapi</td></tr><tr><td>hosts</td><td>example.com</td></tr><tr><td>upstream_url</td><td>http://mockbin.org</td></tr></table>	KEY	VALUE	name	walletapi	hosts	example.com	upstream_url	http://mockbin.org
KEY	VALUE									
name	walletapi									
hosts	example.com									
upstream_url	http://mockbin.org									
<ul style="list-style-type: none">API Key használata kliens azonosításra (mobil, web)	4	POSTMAN: url: http://localhost:8001/consumers <pre>{ "username": "web", "custom_id": "1" }</pre>								
<ul style="list-style-type: none">Rate limit használata (globális)	3	POSTMAN: url: http://localhost:8001/plugins <table><tr><th>KEY</th><th>VALUE</th></tr><tr><td>name</td><td>rate-limiting</td></tr><tr><td>config.second</td><td>5</td></tr><tr><td>config.hour</td><td>10000</td></tr></table>	KEY	VALUE	name	rate-limiting	config.second	5	config.hour	10000
KEY	VALUE									
name	rate-limiting									
config.second	5									
config.hour	10000									
<ul style="list-style-type: none">Rate limit használata (klienstől függő - Kong consumer)	4	POSTMAN: url: http://localhost:8001/plugins <pre>{ "name": "rate-limiting", "consumer_id": "a696a2dd-5d6d-4492-a526-4c2256d83d9d", "config.second": "10" }</pre>								
<ul style="list-style-type: none">Dinamikus terhelés (load balancing)	4	POSTMAN:								

		<div>url: http://localhost:8001/upstreams/walletservice/targets?target=walletapi2:10010%26weight=50</div> <table><tr><th>KEY</th><th>VALUE</th></tr><tr><td>Content-Type</td><td>application/x-www-form-urlencoded</td></tr><tr><td>Postman-Token</td><td>a654c88d-03cd-4d75-8e2c-f97910dd22ab</td></tr><tr><td>cache-control</td><td>no-cache</td></tr></table>	KEY	VALUE	Content-Type	application/x-www-form-urlencoded	Postman-Token	a654c88d-03cd-4d75-8e2c-f97910dd22ab	cache-control	no-cache
KEY	VALUE									
Content-Type	application/x-www-form-urlencoded									
Postman-Token	a654c88d-03cd-4d75-8e2c-f97910dd22ab									
cache-control	no-cache									
Docker Compose										
<ul style="list-style-type: none">• Működő Docker-compose deployment Kong-gal és OpenAPI-val (verzio min 3)	5	DockerFile és docker-compose.yml								
<ul style="list-style-type: none">• Verziózott Docker image-ek használata	3	docker-compose.yml: 1 sor és 72-82 sor								
<ul style="list-style-type: none">• Health check definiálása az API-hoz	2	docker-compose.yml: 60-64 sor								