

SMART MANAGER

Tesztelés dokumentáció



Készítette:
Magyar Márk József,
Nagy Huba Kende

1. Tartalomjegyzék

2.	Frontend tesztelés megvalósítása	3
3.	Basic Tesztesetek.....	3
3.1	Bejelentkezés és Kijelentkezés.....	3
3.2	Regisztráció	3
4.	Beállítások Tesztelési esetei.....	4
4.1	Jelszó változtatás és visszaállítás	4
4.2	Deaktiválás	4
5.	Chat Teszteset.....	5
5.1	Chat Teszt	5
6.	Tanári felület tesztesetei	6
6.1	Projekt Feladat hozzáadás	6
6.2	Felhasználó hozzáadása a Feladathoz.....	6
6.3	Feladat Hozzáadása a Projekthez	7
6.4	Diák Értékelése.....	7
6.5	Diák Statisztikájának megtekintése	8
7.	Backend tesztelés megvalósítása	9
7.1	Felhasználói adatok lekérése	9
7.2	Feladat frissítése	10
7.3	Feladat komment létrehozása	11
7.4	Beszélgetés lekérdezése	12
7.5	Beadás létrehozása	13
7.6	Üzenet törlése	14
8.	API interaktív specifikáció	14

2. Frontend tesztelés megvalósítása

Minden frontend teszt során automatizált tesztelés történt, amit Robot Framework keretrendszerrel a Ride IDE-ben készült, a SeleniumLibrary web tesztelő könyvtár alkalmazásával.

3. Basic Tesztesetek

Basic tesztesetekben le lett tesztelve a bejelentkezés és a kijelentkezés egyben, illetve a regisztráció külön.

3.1 Bejelentkezés és Kijelentkezés

Teszt neve:	LOGIN_LOGOUT
Automatizált:	IGEN
Framework:	Robot Framework
Böngésző:	Chrome
Teszt lépések:	Böngésző megnyitása
	Bejelentkezés gomb kattintás
	Felhasználónév Bekérése
	Jelszó Bekérése
	Bejelentkezés
Elvárt eredmény:	Felhasználó dashboard oldalra kerül
Eredmény:	PASS

3.2 Regisztráció

Teszt neve:	REGISTER
Automatizált:	IGEN
Framework:	Robot Framework
Böngésző:	Chrome
Teszt lépések:	Böngésző megnyitása
	Regisztráció gomb kattintás
	Felhasználónév Bekérése
	Jelszó Bekérése
	Email Bekérése
	Jelszó Bekérése
	Megerősítő Jelszó Bekérése
	Szerepkör kiválasztása
	ÁSZF elfogadása
	Regisztráció
Elvárt eredmény:	Felhasználó Bejelentkezés oldalra kerül
Eredmény:	PASS

4. Beállítások Tesztelési esetei

Beállítások tesztesetben le lett tesztelve a jelszó megváltoztatása ÉS visszaállítása (tehát ellenőrizve is lett a sikeres át írás) illetve tesztelve lett a fiók deaktiválása.

4.1 Jelszó változtatás és visszaállítás

Teszt neve:	Jelszo_Valtoztatats
Automatizált:	IGEN
Framework:	Robot Framework
Böngésző:	Chrome
Teszt lépések:	Böngésző megnyitása
	Bejelentkezés gomb kattintás
	Felhasználónév bekérése
	Jelszó bekérés
	Bejelentkezés
	Beállítások menüpont kiválasztása
	Fiók menüpont kiválasztása
	Új jelszó megadása
	Új jelszó megadása (újra)
	Mentés gombra kattintás
	Kijelentkezés
	Bejelentkezés gomb kattintás
	Felhasználónév bekérése
	Jelszó bekérés
	Bejelentkezés
	Beállítások menüpont kiválasztása
	Fiók menüpont kiválasztása
	Új jelszó megadása
	Új jelszó megadása (újra)
	Mentés gombra kattintás
Elvárt eredmény:	Felhasználó jelszavának megváltoztatása majd visszaállítása
Eredmény:	PASS

4.2 Deaktiválás

Teszt neve:	DEAKTIVALAS
Automatizált:	IGEN
Framework:	Robot Framework
Böngésző:	Chrome
Teszt lépések:	Böngésző megnyitása
	Bejelentkezés gomb kattintás
	Felhasználónév Bekérése
	Jelszó Bekérés
	Bejelentkezés
	Beállítások menüpont kiválasztása
	Fiók menüpont kiválasztása
	Fiók deaktiválása gomb kattintás
Elvárt eredmény:	Felugró értesítések elfogadása
	Felhasználó a fő oldalra kerül
Eredmény:	PASS

5. Chat Teszeset

A chat teszesetben ellenőrizve lett az üzenetek folyamata a felhasználók között.

5.1 Chat Teszt

Teszt neve:	Uzenet_KULDES_FOGADAS
Automatizált:	IGEN
Framework:	Robot Framework
Böngésző:	Chrome
Teszt lépések:	Böngésző megnyitása
	Bejelentkezés gomb kattintás
	Felhasználónév bekérése
	Jelszó bekérése
	Bejelentkezés
	Üzenetek menüpont kiválasztása
	Felhasználó kiválasztása
	Indítás gomb megnyomása
	Üzenet beírása
	Üzenet küldése
	Kijelentkezés
	Bejelentkezés gomb kattintás
	Felhasználónév bekérése
	Jelszó bekérése
	Bejelentkezés
	Üzenetek menüpont kiválasztása
	Felhasználó kiválasztása
	Indítás gomb megnyomása
	Üzenet beírása
	Üzenet küldése
Elvárt eredmény:	Sikeres Üzenet Küldés És Fogadás
Eredmény:	PASS

6. Tanári felület tesztesetei

A tanári tesztesetek részletesen végig mennek minden lehetséges folyamatban, amit meg tud valósítani a tanár felhasználó.

6.1 Projekt Feladat hozzáadás

Teszt neve:	Tanar_FeladatHozzaadas
Automatizált:	IGEN
Framework:	Robot Framework
Böngésző:	Chrome
Teszt lépések:	Böngésző megnyitása
	Bejelentkezés gomb kattintás
	Felhasználónév bekérése
	Jelszó bekérése
	Bejelentkezés
	Feladatok menüpont kiválasztása
	Új projekt gomb kattintás
	Projekt név megadása
	Projekt leírás megadása
	Mentés Gomb kattintása
Elvárt eredmény:	Sikeres Feladat Hozzáadása / Rögzítése
Eredmény:	PASS

6.2 Felhasználó hozzáadása a Feladathoz

Teszt neve:	Tanar_FeladatonBeluliFelhasznaloHozzaadas
Automatizált:	IGEN
Framework:	Robot Framework
Böngésző:	Chrome
Teszt lépések:	Böngésző megnyitása
	Bejelentkezés gomb kattintás
	Felhasználónév bekérése
	Jelszó bekérése
	Bejelentkezés
	Feladatok menüpont kiválasztása
	Újonnan létrehozott Projekt kiválasztása
	Tag hozzáadása gomb kattintás
	Felhasználó kiválasztása
	Hozzáadás gomb kattintás
Elvárt eredmény:	Sikeres Felhasználó hozzárendelése a feladathoz
Eredmény:	PASS

6.3 Feladat Hozzáadása a Projekthez

Teszt neve:	Tanar_FeladatonBeluliFeladatHozzaadas
Automatizált:	IGEN
Framework:	Robot Framework
Böngésző:	Chrome
Teszt lépések:	Böngésző megnyitása
	Bejelentkezés gomb kattintás
	Felhasználónév bekérése
	Jelszó bekérése
	Bejelentkezés
	Feladatok menüpont kiválasztása
	Újonnan létrehozott Projekt kiválasztása
	Projekten belüli feladatok menüpont kiválasztása
	Új feladat gomb kattintás
	Feladat cím megadása
	Feladat leírás megadása
	Felhasználó hozzárendelése
	Prioritás kiválasztása
	Határidő megadása
	Mentés gomb kattintás
Elvárt eredmény:	Sikeres Feladat Hozzáadása
Eredmény:	PASS

6.4 Diák Értékelése

Teszt neve:	Tanar_DiakErtekelese
Automatizált:	IGEN
Framework:	Robot Framework
Böngésző:	Chrome
Teszt lépések:	Böngésző megnyitása
	Bejelentkezés gomb kattintás
	Felhasználónév bekérése
	Jelszó bekérése
	Bejelentkezés
	Értékelés menüpont kiválasztása
	Feladat kiválasztása
	Feladaton belüli felhasználó kiválasztása
	Pontszám megadása
	Szöveges értékelés megadása
	Értékelés gomb kattintás
Elvárt eredmény:	Sikeres Statisztika hozzáadása
Eredmény:	PASS

6.5 Diák Statisztikájának megtekintése

Teszt neve:	Tanar_DiakStat
Automatizált:	IGEN
Framework:	Robot Framework
Böngésző:	Chrome
Teszt lépések:	Böngésző megnyitása
	Bejelentkezés gomb kattintás
	Felhasználónév bekérése
	Jelszó bekérése
	Bejelentkezés
	Értékelés menüpont kiválasztása
	Diák kiválasztása
	Statisztika megtekintése
Elvárt eredmény:	Sikeress Diák értékelés megtekintése
Eredmény:	PASS

7. Backend tesztelés megvalósítása

A REST API végpontjainak tesztelése az Insomnia API kliens használatával történt. A tesztelés célja az endpointok megfelelő működésének ellenőrzése, valamint, hogy a kérésekre helyes státuszkódot adnak vissza.

7.1 Felhasználói adatok lekérése

Metódus	URL	Body	Válasz
GET	/api/auth/profileData	üres	JSON

Teszt script és eredmények:

```
const body = insomnia.response.json();

insomnia.test('200-as státusz', () => {
    insomnia.expect(insomnia.response.code).to.equal(200);
});

insomnia.test('válasz tartalmazza a felhasználói adatokat', () => {
    insomnia.expect(body.data).to.have.property('user');
    insomnia.expect(body.data.user).to.have.property('id');
    insomnia.expect(body.data.user).to.have.property('felhasznalonev');
    insomnia.expect(body.data.user).to.have.property('email');
});

insomnia.test('Jelszó nincs visszaküldve', () => {
    insomnia.expect(body.data.user).to.not.have.property('jelszo');
});
```

PASS	200-As Státusz After-response Test (0.2 ms)
PASS	Válasz Tartalmazza A Felhasználói Adatokat After-response Test (0.2 ms)
PASS	Jelszó Nincs Visszaküldve After-response Test (0.1 ms)

Válasz minta:

```
{
    "success": true,
    "data": {
        "user": {
            "id": 37,
            "felhasznalonev": "testuser_1772130958562",
            "email": "test_1772130958562@example.com",
            "teljes_nev": "Test User",
            "szerep_tipus": "diak",
            "aktiv": true,
            "elerheto": true,
            "letrehozas_idopont": "2026-02-26T18:35:59.360Z"
        }
    }
}
```

7.2 Feladat frissítése

Metódus	URL	Body	Válasz
PUT	/api/project/feladat/{task_id}	JSON	JSON

Teszt script és eredmények:

<pre>const body = insomnia.response.json(); insomnia.test('200-as státusz', () => { insomnia.expect(insomnia.response.code).to.equal(200); }); insomnia.test('Feladat sikeresen frissítve', () => { insomnia.expect(body).to.have.property('success', true); insomnia.expect(body.data.task.feladat_nev).to.equal('Új feladat név'); insomnia.expect(body.data.task.statusz).to.equal('befejezett'); });</pre>	PASS 200-As Státusz After-response Test (0.2 ms)	PASS Feladat Sikeresen Frissítve After-response Test (0.1 ms)
--	--	---

Válasz minta:

<pre>{ "success": true, "message": "Feladat sikeresen frissítve", "data": { "task": { "id": 36, "projekt_id": 41, "feladat_nev": "Új feladat név", "feladat_leiras": "Feladat leírás teszteléshez", "letrehozo_id": 39, "felelos_id": null, "prioritas": "alacsony", "statusz": "befejezett", "hatarido": "2026-06-29T22:00:00.000Z", "letrehozas_idopont": "2026-02-26T18:54:56.547Z" } } }</pre>
--

7.3 Feladat komment létrehozása

Metódus	URL	Body	Válasz
POST	/api/project/ujFeladatKomment	JSON	JSON

Teszt script és eredmények:

```
const body = insomnia.response.json();

insomnia.test('201-es státusz', () => {
  insomnia.expect(insomnia.response.code).to.equal(201);
});

insomnia.test('Komment létrehozva', () => {
  insomnia.expect(body).to.have.property('success', true);
  insomnia.expect(body.data).to.have.property('comment');
  insomnia.expect(body.data.comment).to.have.property('id');
  insomnia.expect(body.data.comment).to.have.property('komment_szoveg');
});

if (body.success && body.data && body.data.comment) {
  insomnia.environment.set('comment_id', body.data.comment.id);
}
```

PASS
201-Es Státusz
After-response Test (0.1 ms)

PASS
Komment Létrehozva
After-response Test (0.1 ms)

Válasz minta:

```
{
  "success": true,
  "message": "Komment sikeresen létrehozva",
  "data": {
    "comment": {
      "id": 22,
      "feladat_id": 37,
      "felhasznalo_id": 40,
      "komment_szoveg": "Ez egy teszt komment a feladathoz",
      "letrehozas_idopont": "2026-02-26T19:00:14.984Z"
    }
  }
}
```

7.4 Beszélgetés lekérdezése

Metódus	URL	Body	Válasz
GET	/api/messages/beszelgetes?with={user_id}&projektId={project_id}		JSON

Teszt script és eredmények:

```
const body = insomnia.response.json();

insomnia.test('200-as Státusz', () => {
    insomnia.expect(insomnia.response.code).to.equal(200);
});

insomnia.test('Beszélgetés adatai visszakülvé', () => {
    insomnia.expect(body).to.have.property('success', true);
    insomnia.expect(body.data).to.have.property('messages');
    insomnia.expect(body.data.messages).to.be.an('array');
});
```

PASS 200-As Státusz After-response Test (0.1 ms)	PASS Beszélgetés Adatai Visszakülvé After-response Test (0.1 ms)
--	--

Válasz minta:

```
{
    "success": true,
    "data": {
        "messages": [],
        "count": 0
    }
}
```

7.5 Bédas létrehozása

Metódus	URL	Body	Válasz
POST	/api/files /beadas	JSON	JSON

Teszt script és eredmények:

```
const body = insomnia.response.json();

insomnia.test('201-es státusz', () => {
  insomnia.expect(insomnia.response.code).to.equal(201);
});

insomnia.test('Bédas létrehozva', () => {
  insomnia.expect(body).to.have.property('success', true);
  insomnia.expect(body.data).to.have.property('beadas');
  insomnia.expect(body.data.beadas).to.have.property('id');
});

if (body.success && body.data && body.data.beadas) {
  insomnia.environment.set('beadas_id', body.data.beadas.id);
}
```

PASS 201-Es Státusz After-response Test (0.1 ms)	PASS Bédas Létrehozva After-response Test (0.1 ms)
--	--

Válasz minta:

```
{
  "success": true,
  "message": "Sikeress beadás létrehozás",
  "data": {
    "beadas": {
      "id": 39,
      "feladat_id": 39,
      "felhasznalo_id": 42,
      "tanar_id": 42,
      "pontszam": 0,
      "jegy": 1,
      "statusz": "beadva",
      "visszajelzes": "Teszt beadás",
      "bekuldes_idopont": "2026-02-26T19:16:10.566Z",
      "ertekeles_idopont": null
    }
  }
}
```

7.6 Üzenet törlése

Metódus	URL	Body	Válasz
DELETE	/api/messages/torles/{message_id}	üres	JSON

Teszt script és eredmények:

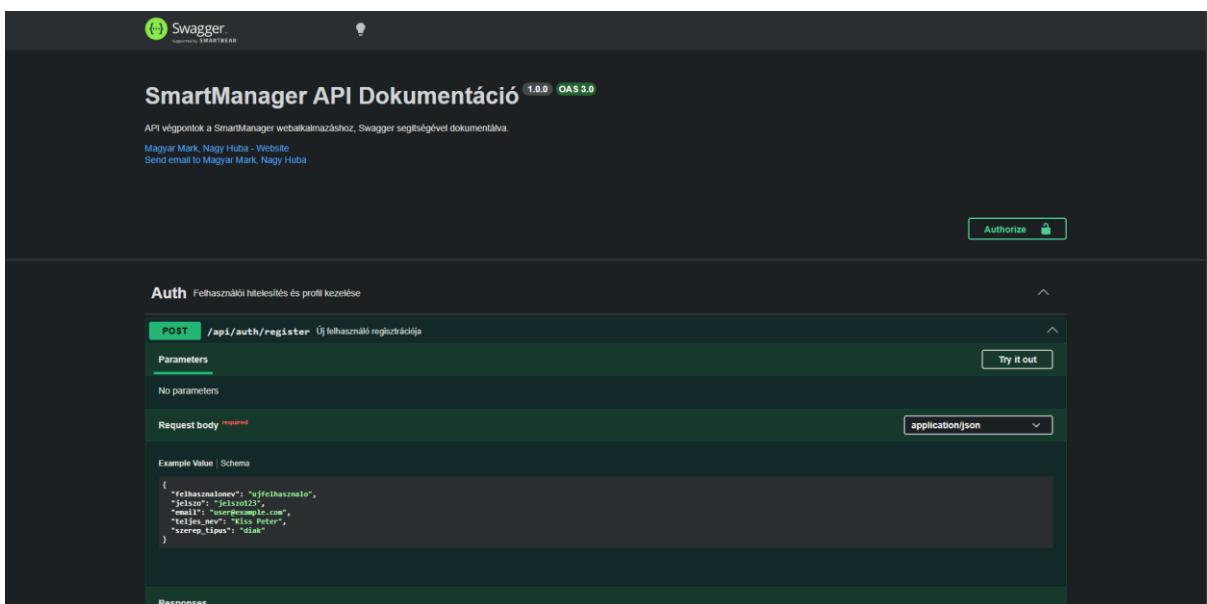
<pre>const body = insomnia.response.json(); insomnia.test('200-as státusz', () => { insomnia.expect(insomnia.response.code).to.equal(200); }; insomnia.test('Üzenet törölve', () => { insomnia.expect(body).to.have.property('success', true); });</pre>	PASS 200-As Státusz After-response Test (0.1 ms)	PASS Üzenet Törölve. After-response Test (0.1 ms)
--	--	---

Válasz minta:

```
{
  "success": true,
  "message": "Üzenet sikeresen törölve"
}
```

8. API interaktív specifikáció

A REST API dokumentációja Swagger alapú interaktív felületen keresztül elérhető. Tartalmazza az elérési útvonalakat, támogatott HTTP méthodusokat, bemeneti paramétereket és válaszstruktúrát. A böngészőbe beírva a <http://localhost:3000/docs> URL-en elérhető



The screenshot shows the SmartManager API Documentation page. At the top, it says "SmartManager API Dokumentáció 1.0.0 OAS 3.0". Below that, there's a "Auth" section for user authentication. The main part of the screen displays the "POST /api/auth/register" endpoint, which is described as "Új felhasználó regisztrációja". It shows "Parameters" (none), "Request body" (with a required field "felhasznalonev" containing "ujfelhasznalo", "jelszwo" containing "jelszof23", "email" containing "nagy@nagy.hu", "teljes_nev" containing "Kiss Péter", and "szerep_tipus" containing "diák"), and "Responses" (none).