

ZÁRÓDOLGOZAT – TARTALMI KIVONAT

Lázár Levente

2024



Magyar Agrár- és Élettudományi Egyetem
Károly Róbert Campus

**Felsőoktatási szakképzés – programtervező informatikus
szak**

API hívások tesztjeinek átvitele Postman-ből ABAP környezetbe

Belső konzulens:	Dr. Pántya Róbert
	Egyetemi adjunktus
Belső konzulens intézete/tanszéke:	MATE Műszaki Intézet
Készítette:	Lázár Levente

Gyöngyös
2024

A záródolgozatom célja, hogy bemutasson egy fejlesztést a munkahelyemen, amely szakmai szempontból megfelel az egyetemi tanulmányaimhoz is. A projekt fő célja egy adott programban (*Postman*) íródott API unit tesztek (ezek a program legkisebb egységei, amik egyetlen konkrét feladatot látnak el) "átköltöztetése" egy másik programozási környezetbe, mivel a nevezett programot a cég a tavalyi évben kivette az elérhető szoftverek közül. Így az én feladatom az lett, hogy ezeket a teszteket átvigyerem a cég saját programozási nyelvébe, és hogy ugyanazt az eredményt produkálva sikeresen futtathatóak legyenek ezek a tesztek, amik a teszt-lefedettségben nagyon sokat segítenek, hiszen ezen esetek a vállalat ügyfeleinek számos rendszerében is hívásra kerülnek. A dolgozatban érinteni fogom az összes felmerülő kihívást és nehézséget, amik a projekt során felléptek, hiszen két különböző rendszer, illetve két különböző programozási nyelv esetéről van szó, eltérő megoldási lehetőségekkel, de a végeredménynek pontosan ugyanolyannak kell lennie, mint az eredeti tesztek esetében. Ezen felül szerepelni fog jelentős mennyiségű forráskód is, mindkét környezetből, hogy pontosan és teljes mértékben be tudjam mutatni az átköltözés menetét, illetve a programozási nyelvek különféle megoldásait egy-egy felmerülő problémára, hiszen amíg a Postman egy kifejezetten API tesztek írására és alkalmazására való szoftver, addig az ABAP egy üzleti programozási környezetet biztosító nyelv, amelyben ezen tesztek átköltöztetése nem is olyan egyszerű, mint azt elsőre gondolnánk.

Összefoglalva, ez a projekt rávilágított a Postman agilis tesztelési képességei és az ABAP robusztus és integrált tesztelési keretrendszere közötti szinergiára. Egyedi deszerializáló metódusok, dinamikus API-hívási funkciók kifejlesztésével és ezen elemek ABAP tesztelési keretrendszerbe történő zökkenőmentes integrálásával a projekt sikeresen átültette a Postman API-tesztek agilitását az ABAP fejlesztés strukturált világába. Ezzel javítva a külső interfészekkel összekapcsolódó ABAP-alkalmazások minőségét és megbízhatóságát. A saját deszerializáló metódusok írására azért volt szükség, mert bár a dolgozatban egy fajta struktúrát, a periódusokat mutattam be, de a projekt során felmerültek egyéb GET kérések is, amelyek már más eredményt adtak vissza egy teljesen más struktúrával, különféle attribútumokkal. Az ABAP nyelvben egy beépített osztályban található ilyen JSON deszerializáló metódus, ahol bár különféle attribútumok megadásával mi is ki tudunk nyerni egyszerűen információt, de mivel a feladatban az értékekkel számolást is kellett végezni, így problémák merültek fel a különféle struktúrák miatt, hiszen nem lehetett alkalmazni ezt a függvényt, mivel az nem pakolta be a kinyert információt egy belső táblába. A belső tábla

létrehozására azért volt szükség, hogy az egyes név/érték párokat ideiglenesen fel tudjam tölteni ilyen táblákba és számítási műveleteket tudjak rajtuk végezni. Ilyen volt a részletesen bemutatott első teszt eset, ahol annak kellett visszajönnie, hogy a JSON válaszban 170 különféle név/érték pár szerepel, nem több és nem is kevesebb. A JSON deszerializáló függvényem nem csupán szétválasztva értelmezte ezen név/érték párokat, hanem egy másik segítő függvény használatával fel is töltötte az előre létrehozott struktúrát. Az ideiglenes tábla pontosan olyan attribútumokat tartalmazott, mint a JSON válaszban látható értékek, minden a megfelelő helyen került feltöltésre, innentől pedig a számítási műveletek nagyon leegyszerűsödtek, hiszen csak azt kellett vizsgálni, hogy annyi elem van-e a táblában, mint ahányat elvárunk a GET kérés válaszából. Innentől ha egy teljesen más struktúrát kapunk válaszul, akkor is tudjuk használni ezen függvényeket, csupán az ideiglenesen létrehozott táblánk struktúráját kell módosítani olyan mezőkkel, amiket a válaszban is kapunk, vagy adott esetben új struktúrát is létre tudunk hozni és onnantól kezdve azzal dolgozhatunk tovább, a deszerializálás és a táblafeltöltés működni fog.

A projekt sikere a Postman API tesztek ABAP-környezetben történő replikálásában nem csak technikai vívmány, hanem stratégiai továbbfejlesztése is a cég fejlesztési életciklusának. Az API-tesztek ABAP környezetbe való beépítésével a fejlesztők és tesztelők mostantól átfogó tesztcsomagot hajthatnak végre egyetlen, egységes környezetben, jelentősen csökkentve a környezetváltást és a manuális többletterhelést, amely korábban a külső API-integrációk érvényesítéséhez szükséges volt.

Ezen kívül ez a törekvés hangsúlyozza az alkalmazkodóképesség fontosságát a szoftverfejlesztésben. Az a képesség, hogy a teszteket le lehet fordítani egy modern API-tesztelési platformról, például a Postman programról az ABAP-környezetbe, amely hosszabb múltra tekint vissza és más fókuszú, azt a sokoldalúságot mutatja, amely szükséges ahhoz, hogy a régi rendszerek naprakészek és teljes mértékben működőképesek legyenek a fejlődő külső szolgáltatások és API-k mellett.