Jelenleg 4 projekt van a solutionben:

* Common
* Extending\_WCF
* TestClient
* TestService

A TestClient és a TestService egy alap WCF-es service és kliens, jelenleg nem sok hasznuk.

A Commonban vannak a teszt osztályok, illetve a tesztek.

Az ExtendingWCF projektben van a lényeg:

## GenSerializer

Ez az osztály generálja leés tárolja az egyes osztályokhoz tartozó sorosítókat.

Gyakorlatilag ez a projekt lelke.

A Serialize, és a Deserialize metódusok amiket ki szeretnék ajánlani, majd a projekt végén, egyelőre tesztelési céllal több metódus is publikus.

A működése a következő:

Meghívják valamelyik metódust a megfelelő paraméterekkel, ekkor a lekéri a kód, hogy van-e már generált sorosító, ha van használja.

Ha nincs a CreateSerializer fut le, ahol kétszer legenerálódik a sorosító, egyszer egy objektumba kerül az eredmény egyszer pedig egy DLL-be, hogy ILSpy-al meg tudjam nézni a generált kódot.

A Discover metódus a feltérképezést végzi, ez megy végig az osztály mezőin (ABC sorrendben), majd az összeshez szükséges kiíró és betöltő metódusokat belegenerálja a kódba.

A discover a primitív típusokra az Emit(De)SerializePrimitive függvényt hívja ami ezeket a típusokat kezeli le.

## IdentityEqualityComparer

A GenSerializerben használom, az equals felüldefiniálására.

Azért kellett, mert az átküldött objektumokat letárolom, és ha még egyszer át kell küldeni, akkor már csak az őt helyettesítő sorszámot küldöm át, és az egyezést referencia szerint nézem.

## IntegerConverter

A benne lévő logika van IL kódban implementálva a GenSerializerben, csak teszt miatt maradt benn.

## My\*akármi\*

## A My-al kezdődő osztályok elavultak. Egy reflection alapú megoldáshoz tartoznak, a projekt kezedetén használtam őket tesztelésre. Ide tartozik a StaticXmlObjectSerializer is.

## UTF8EncodingProxy

Ez az osztály valósítja meg a stringek UTF8 kódolású sorosítását.