

# Elektra kérdőív generálás

## (iFORM technológia)

verzió: 01.

2020.11.16

### Tartalomjegyzék

I. Elektra kérdőív generálás (iFORM).....	2
<b>I.1. A kérdőívek felépítésének alapelvei</b> .....	2
<b>I.2.Javaslatok adat XML generáláshoz</b> .....	5
Mellékletek.....	6
II. Adat XML struktúrája (iFORM).....	7
<b>Rekordmutató</b> .....	7
<b>Adatösszevonás</b> .....	8
III. Adat XML séma (iFORM) .....	11
<b>xmlAdapter - complex type</b> .....	11
<b>Error - complex type</b> .....	11
<b>attachment - complex type</b> .....	12
<b>attachments - complex type</b> .....	13
<b>chapter - complex type</b> .....	14
<b>Data - complex type</b> .....	15
<b>form - complex type</b> .....	16
<b>row - complex type</b> .....	17
<b>table - complex type</b> .....	18
<b>fileSizeLongAdapter - complex type</b> .....	18
<b>formAdaptedMap - complex type</b> .....	19
<b>formKeyElement - complex type</b> .....	19
<b>formMapAdapter - complex type</b> .....	20
<b>sourceEnum - simple type</b> .....	20
IV. Kérdőívsablon XML séma (iFORM).....	21
<b>xmlAdapter - complex type</b> .....	21
<b>attachmentField - complex type</b> .....	21

# I. Elektra kérdőív generálás (iFORM)

Jelen dokumentum a KSH-ELEKTRA rendszeréhez adatszolgáltatást fejleszteni szándékozók számára nyújt technikai magyarázatot az adatszolgáltatásokat tartalmazó import állományok szakszerű előállításához.

A KSH-ELEKTRA rendszer használata egyedi kérdőívek kitöltésére épül. Egy kérdőív példány az egyes adatgyűjtések (OSAP-ok) adott tárgyidőszak szerinti, meghatározott adatszolgáltató jól beazonosítható beszámolási egységére és részegységére vonatkozó információk összességét tartalmazza.

A kérdőívek kitöltőbeli megjelenését, az összegyűjtendő adatok leírását (típus, értékészlet vagy annak forrás megjelölése, kitölthetőség, stb.), az elvégzendő ellenőrzéseket, illetve a kérdőívhez kapcsolódó sűgő/kitöltési útmutató hivatkozásokat OSAP-onként egyetlen állomány foglalja össze, amit a rendszerben "**kérdőívsablon**"-nak hívunk. A kérdőívsablonok XML formátumban tartalmazzák a kitöltőprogram vezérléséhez szükséges információkat. A kitöltő program ezek alapján állítja elő a felhasználók számára az adatok bevitelére szolgáló HTML felületet, de a rendszerhez kapcsolódó fejlesztők is ennek az értelmezésével tudják a helyes importállományokat előállítani. A sablonok értelemszerűen az egyes adatgyűjtések esetében eltérőek, viszont a kérdőívsablont leíró XML állományok struktúrája szabványos XSD-nek megfelelően épül fel (lásd a mellékelt "iform-sablon.xsd" állományt), így az egyes sablonok bárki által logikusan értelmezhetőek, felhasználhatóak az implementáció eszközeitől és nyelvtől függetlenül. A kérdőívsablon esetében irányelv, hogy azok a tulajdonságok kerülnek bele, amik az alapértelmezett jellemzőkhöz képest eltérően vannak beállítva.

A KSH-ELEKTRA rendszerbe egyéb informatikai rendszerekből XML állományok felhasználásával lehet adatot beimportálni. Ezeket az állományokat a továbbiakban "**adat XML**" néven hivatkozunk. A kérdőív kitöltő program is ilyen adat XML-eket állít elő az egyes kérdőívek kitöltésekor. Az adat XML kezelését az egyes adatgyűjtéseknél használt kérdőívsablonok vezérlik.

Az adat XML-ek struktúrája szintén egységes logikájú (a mellékletként csatolt XSD szerint, lásd a mellékelt "iform-adatxml.xsd" állományt), ugyanakkor OSAP-ra vonatkozó kérdőívsablon szerint változó konkrét struktúrájú. Az adat XML-ek belső felépítése ennek következtében a kérdőívsablon elemzésével állítható elő. **Az adat XML-ek kötelezően utf-8 kódolással kell készüljenek.**

## I.1. A kérdőívek felépítésének alapelvei

A kitöltést támogató kiegészítő elemektől eltekintve minden kérdőív tetszőleges számú fejezetből és azon belül tetszőleges számú egyedi beviteli mezőből, vagy táblázatnak látszó komponensstruktúrából (és azok ismétlődéseiből (~táblázat soraiból)) áll. Utóbbiak egymásba is ágyazhatók, így az egyes fejezetek bonyolultsága is elvileg korlátlan, de a gyakorlatban ez természetesen véges, hogy áttekinthető maradjon. A rendszer táblázatokat csak az adat XML-eken belül kezel, valójában az összetartozó adatokat csoportosítja, és ezek ismétlődése esetén az összetartozó mezőkből sorokat képez, amiket táblázatokban fog össze.

Az ismétlődések esetében egyedileg szabályozott viselkedés kérdése, hogy a KSH által előzetesen megszemélyesítéssel létrehozott ismétlődések (sorok) mellett van-e az adatszolgáltatónak joga új sorokat hozzáadni, vagy a megszemélyesítettekben kitörölni.

Azokban az esetekben, ahol nem foglalható össze egyetlen táblázatban az ismétlődő adatok köre, a rendszer lehetőséget ad egész fejezetek több példányban való ismétlődésére. Ez logikailag a táblázat soraival analóg tervezési minta, csak egy magasabb szinten. A fejezetek ismétlődésének viselkedése a táblázat sorok ismételhetségével azonos módon szabályozható egyedileg.

Az adat XML-ben az egyes struktúrák a következő szabályok szerint képződnek le a kérdőívsablon alapján:

1) A kérdőív példány fejezeteinek megfelelően az adat XML-ben pontosan megegyező számú <chapter> példány található. (Az ismételhető fejezetek példányai külön-külön.)

2) Minden egyes kitöltendő komponens vagy sorszám (attól függetlenül, hogy rejtett-e) mező <data> tagként kerül leírásra. Ezen belül mindig 2 elem található:

- <identifier>: megmutatja, hogy a kérdőívsablon melyik konkrét komponenséből származik az adat
- <value>: a konkrét beviteli mezőben rögzített adattartalom.

Az adat elem tartalmazhat egy attribútumot is "s" néven mely az adatok forrását mutatja. A KSH által előkészített adatok esetén értéke "P", ezt az adat XML-ben kötelező megtartani. Adatszolgáltatók által kitöltött adatok esetén az "s" attribútumot nem szabad megadni.

3) A <data> tag-ek táblázatokba (<table>), és azokon belül sorokba (<row>) szerveződnek az alábbi esetekben:

- ismétlődést tartalmazó paneleknél: minden panel ismétlődés példány egy-egy sort hoz létre a táblázatban
- rekordmutató panelek: az ugyanolyan rekordmutató értékkel rendelkező kitölthető komponensek képeznek egy-egy sort a táblázatban
- adatösszevonás több panelből: az összevonásra megjelölt panelen belüli további gyerekpanelekből nem külön-külön táblázat képződik, hanem a közös szülőre hivatkozással egyetlen nagy táblázattá állnak össze az egyes gyerekek ismétlődései (esetleg rekordmutató alapján származtatható sorai).

Utóbbi kettőről részletesen az Adat XML struktúrája fejezetben.

4) A fenti táblázatosítás egymásba ágyazott struktúráknál is lehetséges.

Az így létrejövő adat XML-ek megfelelősége tekintetében jó kiindulási alapot jelent a rendszerből az egyes kérdőívekhez kapcsolódó adat XML-ek kiexportálásával kinyerhető állomány. Az adat XML-ben a kitöltésből származó adatokon kívül előfordulhatnak olyan mezők és adatok, amiket a KSH megszemélyesítéssel tölt fel

az egyes kérdőív-példányokba. Ezek a rendszerből kiexportált adat XML-ekben a <data> tagben s="P" attribútummal kerülnek megjelölésre. Ezeket importból nem lehet módosítani, viszont aki a kérdőív-sablonban definiált ellenőrzések/számolások futtatását tervezné külső rendszerben, annak számolnia kell azzal, hogy ezek az adatok is szükségesek lehetnek.

A KSH-ELEKTRA rendszerbe beépített űrlap-technológiában elérhető <attachments> és <attachment> tag-ek bár szerepelnek az XSD-ben, de alkalmazásukra nem kerül sor.

Alkalmazásra kerül viszont az indokolható hibák megadására a <chapter> tag-eket követően az adat XML végén szereplő <justifiableErrors> tag és annak gyerek elemei, az egyes hiba indoklások azonosítójával, felhasználó által megadott tartalmával és egyéb leíró adataival.

Különösen fontos megemlíteni, hogy a kérdőívek összepárosítására szolgáló mezők szintén a KSH által meghatározottan kerülnek az egyes kérdőív-példányok adat XML-jébe. Ezeket feltétlenül reprodukálni kell a külső rendszerekből származó adat XML-ek esetében is annak érdekében, hogy a különböző importálások során a rendszer befogadja az állományokat. Ilyen mezők két helyen vannak az adat XML-ben:

1. Az adat XML elején a <templateKeys> tag-ekben kerülnek megadásra a sablon azonosító adatai. Adatelemenként egy-egy <element> tag és annak gyerekeként egy-egy <name> és <value> tag páros található.

<b>Adatelemhez tartozó &lt;name&gt; tag értéke</b>	<b>Mező leírása</b>
mc01	OSAP szám
mev	Tárgyév

Példa:

```
<templateKeys>
  <element>
    <name>mc01</name>
    <value>5003</value>
  </element>
  <element>
    <name>mev</name>
    <value>20</value>
  </element>
</templateKeys>
```

Technikai okokból a <templateKeys> gyerek elemei között előfordulhatnak egyéb adatok is, ezeket nem kell megadni, a rendszer a szerver oldalon tárolt értéküket változatlanul megtartja az importálás során minden esetben.

2. Előlap fejezeten levő azonosító mezők. Normál mezőként data tag-be kerülnek.

<b>Adatelemhez tartozó &lt;name&gt; tag értéke</b>	<b>Mező leírása</b>
MHO	Tárgyhó

<b>Adatelemhez tartozó &lt;name&gt; tag értéke</b>	<b>Mező leírása</b>
M003G	Adatszolgáltató KSH törzsszáma
M003	Beszámolási egység azonosítója (törzsszám)
MV42	Beszámolási részegység azonosítója

Bizonyos adatgyűjtéseknél előfordulhat, hogy az MV42 értéke üres, a többi esetben kell értéket tartalmaznia az előlapnak.

Az előállítandó **adat XML fájlok elnevezését** a következő konvenciók szerint kell megtenni:

- OSAP megjelenés nélkül:

"adat\_[OSAP kód 4 karakteren]\_[tárgyév 2 karakteren]\_[fájl egyedi ügyféloldali azonosítója, mely nem tartalmazhat aláhúzás karaktert].xml"

pl: adat\_1040\_12\_akarmilyen-adatszolgáltato-oldali-azonosito.xml.  
Ennél az elnevezési konvenciónál alapértelmezetten 0-s OSAP megjelenést feltételezünk.

- OSAP megjelenéssel:

"adat\_[OSAP kód 4 karakteren]\_[tárgyév 2 karakteren]\_[OSAP megjelenés]\_[fájl egyedi ügyféloldali azonosítója, mely nem tartalmazhat aláhúzás karaktert].xml"

pl: adat\_1040\_12\_1\_akarmilyen-adatszolgáltato-oldali-azonosito.xml.

A rendszer az adatszolgáltató oldali fájlnev részlettel nem foglalkozik, korlátokat nem állít, de a felhasználói operációs rendszer megkötéseit figyelembe kell venni.

Az eddig leírt általános elveken és specifikációkon kívül a fejlesztői tájékoztató egyedi igények szerint tartalmazza még a következőket:

- Az igényelt adatgyűjtésekhez egy-egy kérdőívsablon állomány, fájlok neve:
  - OSAP megjelenés nélkül: <OSAP kód>\_<tárgyév>\_<változat>.xml
  - OSAP megjelenéssel: <OSAP kód>\_<tárgyév>\_<OSAP megjelenés>\_<változat>.xml
- Az igényelt adatgyűjtésekhez egy-egy technikai pdf állomány, ami vizuálisan szemlélteti a kérdőív struktúráját, és az egyes beviteli mezőkhöz tartozó azonosító tulajdonságot. Természetesen ez a sablon XSD alapján a kérdőívsablonból kinyerhető információ.

## 1.2.Javaslatok adat XML generáláshoz

- Mindig ügyeljünk az UTF-8-as karakter kódolásra.
- Célszerű nem a teljes adat XML-t generálni, hanem a KSH-ELEKTRA-ból letölteni a KSH által előkészített adat XML-t és azt minta sablonként használni kitölteni.

- Az elkészült adat XML-en az iform-adatxml.xsd alapján célszerű séma ellenőrzést végezni.
  - Az adat XML KSH-ELEKTRA rendszerbe importálásakor célszerű a kérdőívet a KSH-ELEKTRA-ban megnyitva a "Feltöltés adat XML-ből" funkcióval importálni, majd ellenőrizni a kérdőív helyes kitöltöttségét.
- 

## Mellékletek

## II. Adat XML struktúrája (iFORM)

A kérdőív szerkesztőben lehetőség van olyan beállítások használatára, amely módosítja az adat XML felépítését a sablon struktúrájából elsődlegesen adódó felépítéshez képest. Általánosságban a kérdőív sablonban levő minden kompozit elemből, amely ismételhető, egy táblázat lesz az adat XML-ben, az egyedi mezők pedig az őket tartalmazó táblázat oszlopai, vagy egy fejezet nem táblázatba sorolt önálló gyerekelemei lesznek. Vannak azonban speciális esetek, mint a rekordmutató használata és az adatösszevonás, amelyek extra kapcsolatok kialakítását teszik lehetővé.

### Rekordmutató

A kérdőív szerkesztőben lehetőség van a beviteli mezőknél „Rekordmutató” tulajdonság megadására. Az egy panelen, vagy annak gyerekein elhelyezett, beállított rekordmutatót tartalmazó komponensekből egy táblázat kerül létrehozásra az adat XML-ben, mintha egy ismétlődő panel sorai lennének. Az adat XML-ben így képzett táblázat azonosítója a „RecordPointer\_” prefixből és a tartalmazó kompozit elem azonosítójából fog állni (pl.: „RecordPointer\_torzs0”). A táblázat annyi sort fog tartalmazni, ahány egyedi érték került beállításra a sablonban az érintett beviteli mezők Rekordmutató tulajdonságánál. A sorok elemeit pedig azok a mezők alkotják, amelyek azonos Rekordmutató értékkel rendelkeznek.

Fontos figyelni arra, hogy a fejezet törzse is panelnek minősül ebből a szempontból, így ha közvetlenül a törzsre kerülnek rekordmutatóval ellátott mezők, akkor a törzs egyéb paneljain szereplő további rekordmutatós mezőkkel együtt egy táblázattá képződnek le. Amennyiben külön táblázatokba akarjuk leképezni a mezőket, úgy a törzsön belül egy külön gyerek panelra vagy annak gyerekeire kell elhelyezni az egy táblázatba csoportosítandó mezőket.

A rekordmutatós komponensek mindig a kompozit elemen belül kerülnek csoportosításra, és rekordmutatóként külön táblázat sorba kerülnek. Ennek eredményeképp például egy szabadpanelből csinálhatunk az adat xml-ben egy táblázatot, mely annyi sort tartalmaz, ahány különböző rekordmutató került megadásra. Használata akkor lehet szükséges, ha egy kompozit elemen belül több olyan komponenst kell elhelyezni, melyek ugyanahhoz a mutatóhoz tartoznak, vagy valamiért táblázatként kellene kezelni, de egyébként heterogén sorokat akarunk reprezentálni, és nem szabályos ismétlődéseket. Vagy adat szempontból szabályos, de megjelenés szempontjából nem sima táblázatos megjelenítést igénylő adatokat.

Példa az adat XML struktúrájára:

```
<table>
  <identifier>RecordPointer_torzs0</identifier>
  <row panel="torzs0">
    <identifier>1</identifier>
    <data>
      <identifier>NumericField1</identifier>
      <value>10</value>
    </data>
    <data>
      <identifier>NumericField2</identifier>
```

```

        <value>20</value>
      </data>
    </row>
    <row panel="torzs0">
      <identifier>2</identifier>
      <data>
        <identifier>SingleLineTextField3</identifier>
        <value>kerek csokoládé</value>
      </data>
    </row>
    <row panel="torzs0">
      <identifier>3</identifier>
      <data>
        <identifier>DropDownMenu1</identifier>
        <value>barack</value>
      </data>
    </row>
  </table>

```

## Adatösszevonás

A kérdőív szerkesztőben a kompozit komponensek esetén beállítható az „Adatösszevonás” tulajdonság. Ez a beállítás az adat XML struktúrájára jelent megkötést, hatására az adott kompozit komponensből és minden más komponensből, amit tartalmaz, egyetlen táblázat jön létre az adat XML-ben, melynek azonosítója a „DataAggregation\_” prefixből és az adatösszevonásra kijelölt kompozit elem azonosítójából fog állni (pl.: „DataAggregation\_VerticalPanel1”).

Használata a sablonok megszemélyesítést segíti olyan esetben, amikor a megszemélyesítés szempontjából egy táblázatot kell megszemélyesíteni, de a kérdőívben csoportosítás, számolások megkönnyítése vagy egyéb okokból több külön kompozit komponensre van bontva. Adatösszevonások egymásba ágyazása nem lehetséges, azaz egy adatösszevonásosnak megjelölt panel gyerekein már nem szabad ismételt beállítani.

Példa az adat XML struktúrája:

```

<table>
  <identifier>DataAggregation_VerticalPanel1</identifier>
  <row panel="HorizontalPanel1">
    <data>
      <identifier>NumberingField1</identifier>
      <value>1</value>
    </data>
    <data>
      <identifier>SingleLineTextField2</identifier>
      <value>a</value>
    </data>
    <data>
      <identifier>NumericField3</identifier>
      <value></value>
    </data>
  </row>
  <row panel="HorizontalPanel1">
    <data>
      <identifier>NumberingField1</identifier>
      <value>2</value>
    </data>
    <data>
      <identifier>SingleLineTextField2</identifier>
      <value></value>
    </data>
  </row>
</table>

```



```

        </data>
        <data>
            <identifier>NumericField3</identifier>
            <value></value>
        </data>
    </row>
    <row panel="HorizontalPanel2">
        <data>
            <identifier>NumberingField2</identifier>
            <value>1</value>
        </data>
        <data>
            <identifier>SingleLineTextField4</identifier>
            <value>b</value>
        </data>
        <data>
            <identifier>NumericField4</identifier>
            <value></value>
        </data>
    </row>
    <row panel="HorizontalPanel2">
        <data>
            <identifier>NumberingField2</identifier>
            <value>2</value>
        </data>
        <data>
            <identifier>SingleLineTextField4</identifier>
            <value></value>
        </data>
        <data>
            <identifier>NumericField4</identifier>
            <value></value>
        </data>
    </row>
</table>

```

Amennyiben az összevonandó panelek között van rekordmutatót használó is, akkor a rekordmutató kezelésének szabályait betartva az összevont adat XML-ben már nem kerül külön kiírásra a rekordmutatós jelleg, hanem a megfelelő sor azonosítójában jelenik meg a rekordmutató értéke:

```

<table>
    <identifier>DataAggregation_VericalPanel1</identifier>
    <row panel="HorizontalPanel1">
        <data>
            <identifier>NumberingField1</identifier>
            <value>1</value>
        </data>
        <data>
            <identifier>SingleLineTextField2</identifier>
            <value>sorszámos táblázat első sora</value>
        </data>
        <data>
            <identifier>NumericField3</identifier>
            <value>3.1415</value>
        </data>
    </row>
    <row panel="HorizontalPanel1">
        <data>
            <identifier>NumberingField1</identifier>
            <value>2</value>
        </data>
        <data>

```

```

        <identifier>SingleLineTextField2</identifier>
        <value> sorszamos táblázat második sora </value>
    </data>
    <data>
        <identifier>NumericField3</identifier>
        <value>42</value>
    </data>
</row>
<row panel="PlainPanel1">
    <identifier>1</identifier>
    <data>
        <identifier>SingleLineTextField5</identifier>
        <value>43</value>
    </data>
</row>
<row panel="PlainPanel1">
    <identifier>2</identifier>
    <data>
        <identifier>SingleLineTextField6</identifier>
        <value>43</value>
    </data>
</row>
<row panel="PlainPanel1">
    <identifier>3</identifier>
    <data>
        <identifier>MultilineTextField1</identifier>
        <value>234</value>
    </data>
</row>
</table>

```

A fenti példában a PlainPanel1 szabadpanelen 3 különböző mező egy-egy sort alkot rekordmutatós táblázatként.

# III. Adat XML séma (iFORM)

## xmlAdapter - complex type

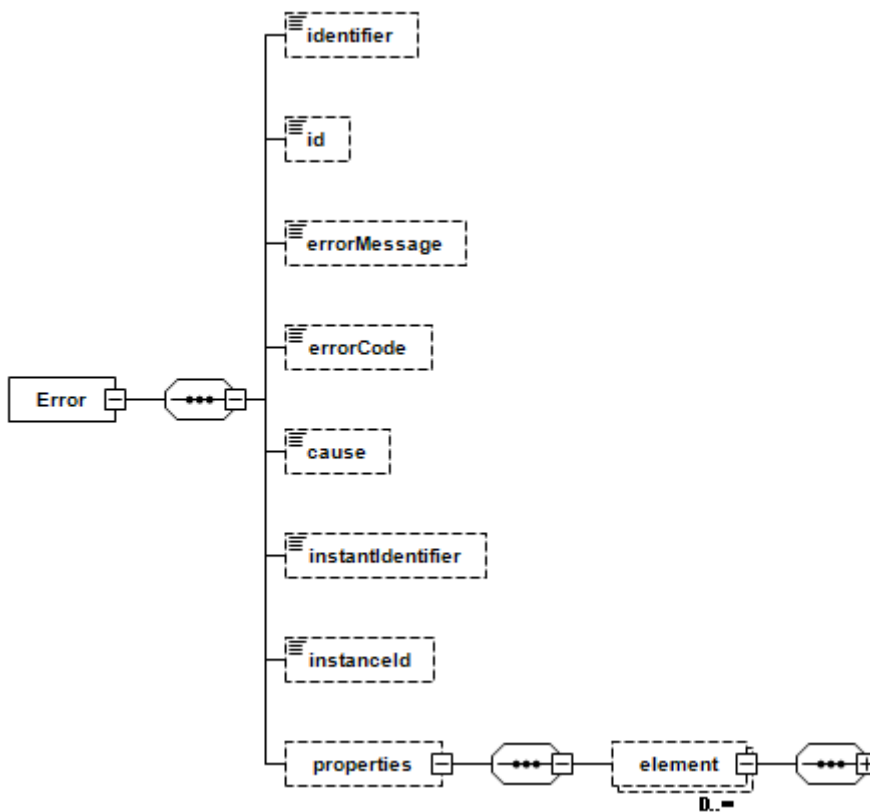
A kulcs/érték adatpárok reprezentálására használt technikai jellegű összetett típusdefiníció (pl keys, properties.)



```
<xs:complexType abstract="true" name="xmlAdapter">
  <xs:sequence/>
</xs:complexType>
```

## Error - complex type

Űrlap ellenőrzéseihez kapcsolódó indokolt hiba



Attribútum neve	Attribútum típusa	Attribútum leírása
identifier	<u>xs:string</u>	Azonosító
id	<u>xs:string</u>	Egyedi azonosító
errorMessage	<u>xs:string</u>	Hibaüzenet

errorCode	<u>xs:string</u>	Hibakód
cause	<u>xs:string</u>	Hiba indoka
instantIdentifier	<u>xs:string</u>	Példány azonosító (nem használt, kivezetése folyamatban)
instanceId	<u>xs:string</u>	Példány azonosító
properties	form:formAdaptedMap	Tulajdonságok

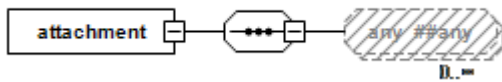
```

<xs:complexType name="justifiableError">
  <xs:annotation>
    <xs:documentation>
      <![CDATA[Úrlap ellenőrzéseihez kapcsolódó indokolt hiba.]]>
    </xs:documentation>
  </xs:annotation>
  <xs:sequence>
    <xs:element minOccurs="0" name="identifier" type="xs:string"/>
    <xs:element minOccurs="0" name="id" type="xs:string"/>
    <xs:element minOccurs="0" name="errorMessage" type="xs:string"/>
    <xs:element minOccurs="0" name="errorCode" type="xs:string"/>
    <xs:element minOccurs="0" name="cause" type="xs:string"/>
    <xs:element minOccurs="0" name="instantIdentifier" type="xs:string"/>
    <xs:element minOccurs="0" name="instanceId" type="xs:string"/>
    <xs:element minOccurs="0" name="properties"
type="form:formAdaptedMap"/>
  </xs:sequence>
</xs:complexType>

```

## attachment - complex type

### Csatolmány leírói



### Attribútum neve Attribútum típusa Attribútum leírása

id	<u>xs:string</u>	Azonosító
fileName	<u>xs:string</u>	Fájl neve
fileSize	<u>xs:string</u>	Fájl mérete

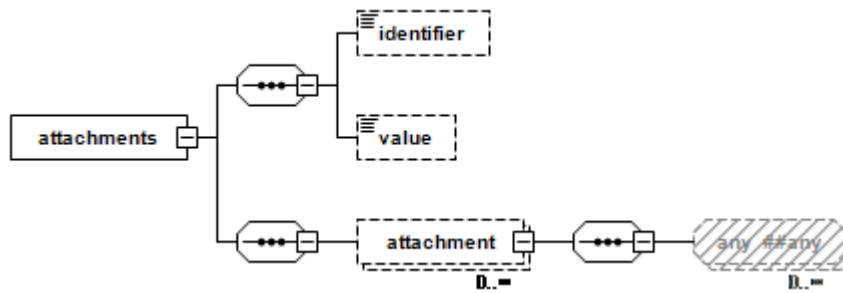
```

<xs:complexType name="attachment">
  <xs:annotation>
    <xs:documentation>
      <![CDATA[Csatolmány leírói]]>
    </xs:documentation>
  </xs:annotation>
  <xs:sequence/>
  <xs:attribute name="id" type="xs:string"/>
  <xs:attribute name="fileName" type="xs:string"/>
  <xs:attribute name="fileSize" type="xs:string"/>
</xs:complexType>

```

## attachments - complex type

### Csatolmányok listája



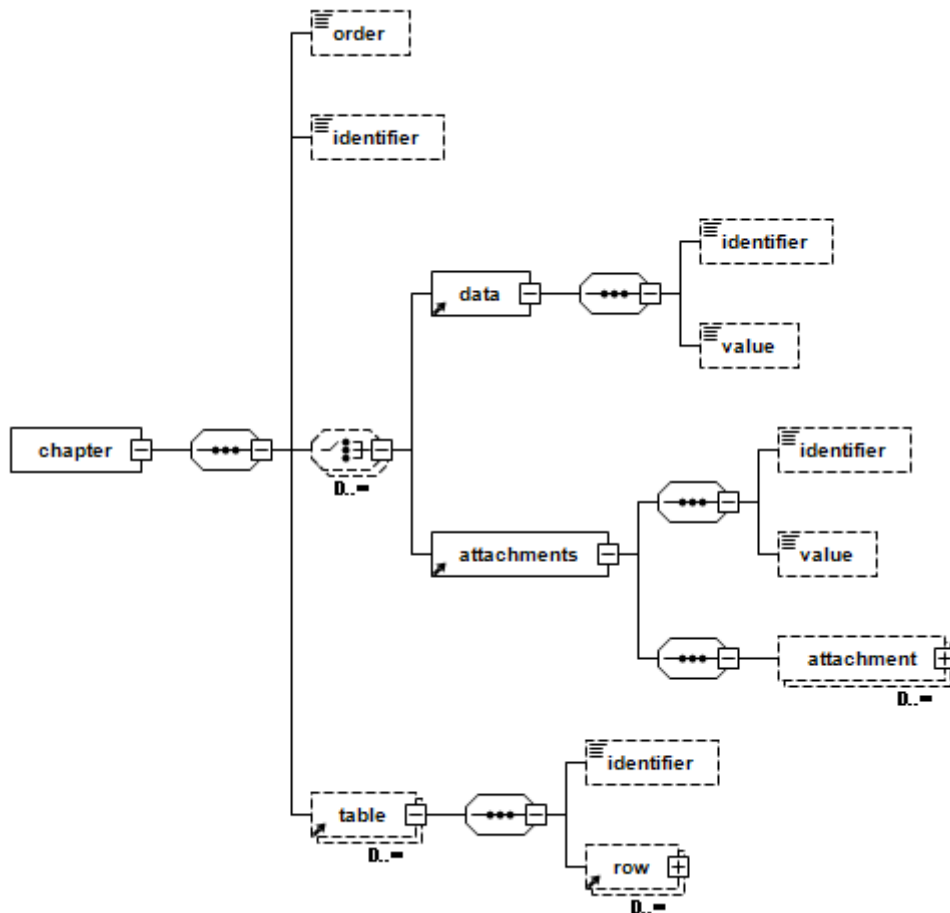
### Attribútum neve Attribútum típusa Attribútum leírása

identifier	<u>xs:string</u>	Azonosító
value	<u>xs:string</u>	Érték
attachment	form:attachment	Csatolmány

```
<xs:complexType name="attachments">
  <xs:annotation>
    <xs:documentation>
      <![CDATA[Csatolmányok listája]]>
    </xs:documentation>
  </xs:annotation>
  <xs:complexContent>
    <xs:extension base="form:data">
      <xs:sequence>
        <xs:element maxOccurs="unbounded" minOccurs="0"
name="attachment" nillable="true" type="form:attachment"/>
      </xs:sequence>
    </xs:extension>
  </xs:complexContent>
</xs:complexType>
```

## chapter - complex type

Úrlap fejezetet reprezentáló osztály



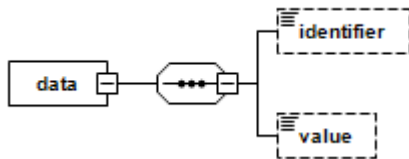
### Attribútum neve Attribútum típusa Attribútum leírása

order	<u>xs:long</u>	Sorrend
identifier	<u>xs:string</u>	Azonosító

```
<xs:complexType name="chapter">
  <xs:annotation>
    <xs:documentation>
      <![CDATA[Úrlap fejezetet reprezentáló osztály.]]>
    </xs:documentation>
  </xs:annotation>
  <xs:sequence>
    <xs:element minOccurs="0" name="order" type="xs:long"/>
    <xs:element minOccurs="0" name="identifier" type="xs:string"/>
    <xs:choice maxOccurs="unbounded" minOccurs="0">
      <xs:element ref="form:data"/>
      <xs:element ref="form:attachments"/>
    </xs:choice>
    <xs:element maxOccurs="unbounded" minOccurs="0" ref="form:table"/>
  </xs:sequence>
  <xs:attribute name="s" type="form:sourceEnum"/>
</xs:complexType>
```

## Data - complex type

### Űrlap adat elem



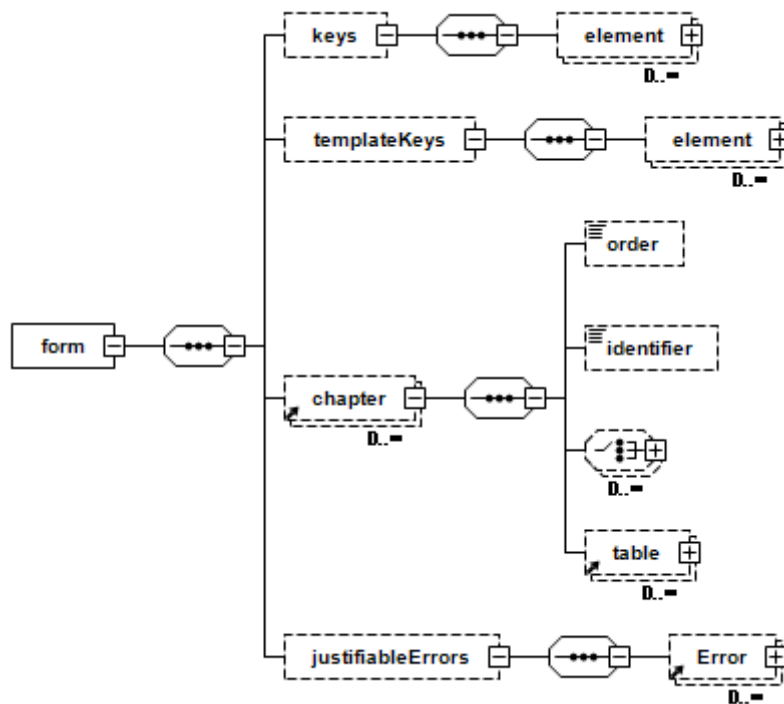
### Attribútum neve Attribútum típusa Attribútum leírása

identifier	xs:string	Azonosító
value	xs:string	Érték

```
<xs:complexType name="data">
  <xs:annotation>
    <xs:documentation>
      <![CDATA[Űrlap adat elem.]]>
    </xs:documentation>
  </xs:annotation>
  <xs:sequence>
    <xs:element minOccurs="0" name="identifier" type="xs:string"/>
    <xs:element minOccurs="0" name="value" type="xs:string"/>
  </xs:sequence>
  <xs:attribute name="s" type="form:sourceEnum">
    <xs:annotation>
      <xs:documentation>
        <![CDATA[Adatok forrása.]]>
      </xs:documentation>
    </xs:annotation>
  </xs:attribute>
</xs:complexType>
```

## form - complex type

Úrlap, az adatXML fő eleme



Attribútum neve	Attribútum típusa	Attribútum leírása
keys	form:formAdaptedMap	Adat XML egyedi azonosítója, kerdoivld és verzió (együttesen bizonylatszám)
templateKeys	form:formAdaptedMap	Sablonban beállított kulcsok
chapter	form:chapter	Fejezet
justifiableErrors	form:Error	Indokolt hibák

```

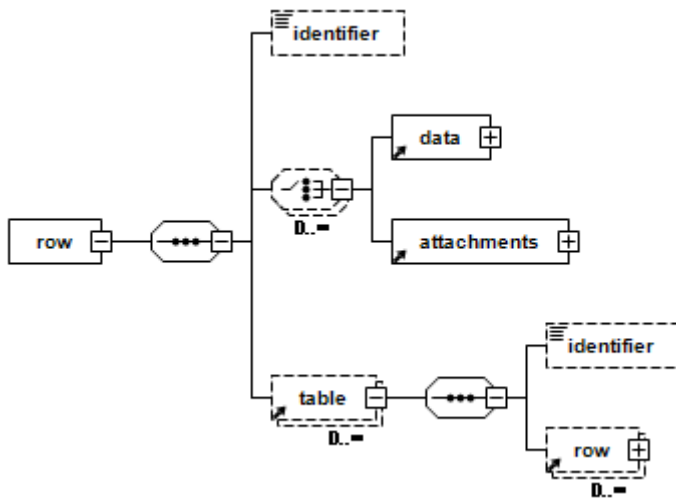
<xs:complexType name="form">
  <xs:annotation>
    <xs:documentation>
      <![CDATA[Úrlap, az adatXML fő eleme.]]>
    </xs:documentation>
  </xs:annotation>
  <xs:sequence>
    <xs:element minOccurs="0" name="keys" type="form:formAdaptedMap"/>
    <xs:element minOccurs="0" name="templateKeys"
type="form:formAdaptedMap"/>
    <xs:element maxOccurs="unbounded" minOccurs="0" ref="form:chapter"/>
    <xs:element minOccurs="0" name="justifiableErrors">
      <xs:complexType>
        <xs:sequence>
          <xs:element maxOccurs="unbounded" minOccurs="0"
ref="form:Error"/>
        </xs:sequence>
      </xs:complexType>
    </xs:element>
  </xs:sequence>
</xs:complexType>

```



## row - complex type

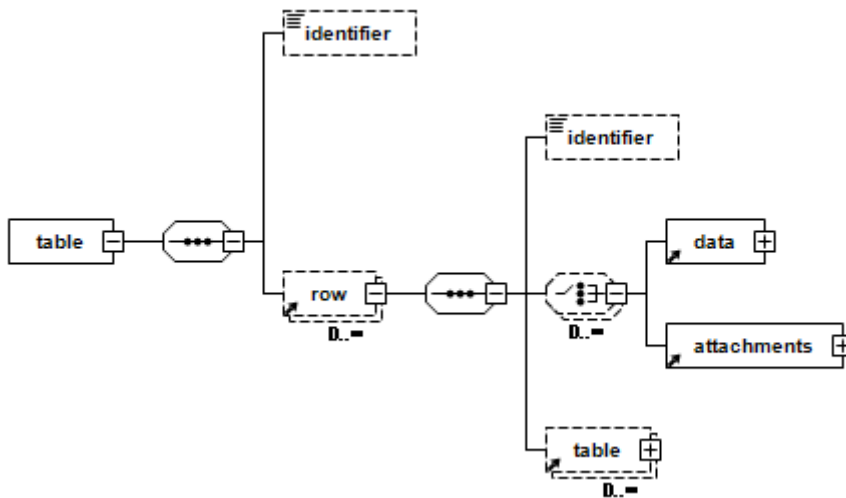
Táblázat egy sora



```
<xs:complexType name="row">
  <xs:annotation>
    <xs:documentation>
      <![CDATA[Táblázat egy sora.]]>
    </xs:documentation>
  </xs:annotation>
  <xs:sequence>
    <xs:element minOccurs="0" name="identifier" type="xs:string"/>
    <xs:choice maxOccurs="unbounded" minOccurs="0">
      <xs:element ref="form:data"/>
      <xs:element ref="form:attachments"/>
    </xs:choice>
    <xs:element maxOccurs="unbounded" minOccurs="0" ref="form:table"/>
  </xs:sequence>
  <xs:attribute name="panel" type="xs:string"/>
  <xs:attribute name="s" type="form:sourceEnum"/>
</xs:complexType>
```

## table - complex type

Űrlap táblázatának adatait tartalmazó osztály



### Attribútum neve Attribútum típusa Attribútum leírása

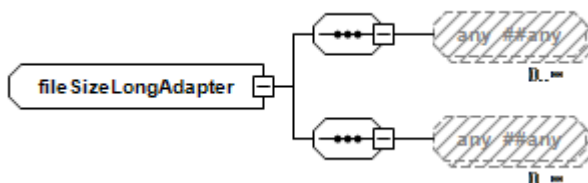
identifier

xs:string

Azonosító

```
<xs:complexType name="table">
  <xs:annotation>
    <xs:documentation>
      <![CDATA[Űrlap táblázatának adatait tartalmazó osztály.]]>
    </xs:documentation>
  </xs:annotation>
  <xs:sequence>
    <xs:element minOccurs="0" name="identifier" type="xs:string"/>
    <xs:element maxOccurs="unbounded" minOccurs="0" ref="form:row"/>
  </xs:sequence>
</xs:complexType>
```

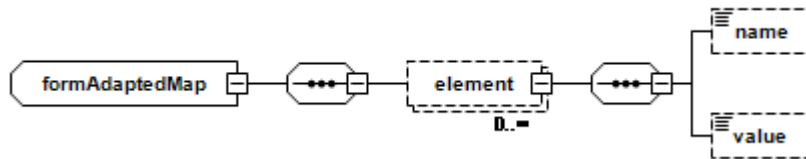
## fileSizeLongAdapter - complex type



```
<xs:complexType name="fileSizeLongAdapter">
  <xs:complexContent>
    <xs:extension base="xmlAdapter">
      <xs:sequence/>
    </xs:extension>
  </xs:complexContent>
</xs:complexType>
```

## formAdaptedMap - complex type

Map típus XML-be képezéséhez és visszaolvasásához adapter osztály



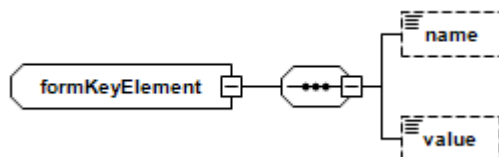
### Attribútum neve Attribútum típusa Attribútum leírása

name	xs:string	Név
value	xs:string	Érték

```
<xs:complexType name="formAdaptedMap">
  <xs:annotation>
    <xs:documentation>
      <![CDATA[Map típus XML-be képezéséhez és visszaolvasásához adapter
osztály.]]>
    </xs:documentation>
  </xs:annotation>
  <xs:sequence>
    <xs:element maxOccurs="unbounded" minOccurs="0" name="element"
nillable="true" type="form:formKeyElement"/>
  </xs:sequence>
</xs:complexType>
```

## formKeyElement - complex type

Kulcs érték párok reprezentálása

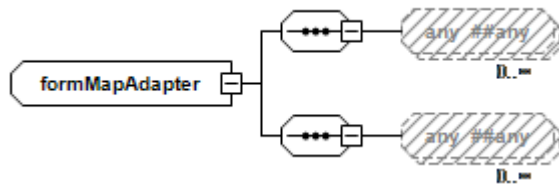


### Attribútum neve Attribútum típusa Attribútum leírása

name	xs:string	Név
value	xs:string	Érték

```
<xs:complexType name="formKeyElement">
  <xs:annotation>
    <xs:documentation>
      <![CDATA[Kulcs érték párok reprezentálása.]]>
    </xs:documentation>
  </xs:annotation>
  <xs:sequence>
    <xs:element minOccurs="0" name="name" type="xs:string"/>
    <xs:element minOccurs="0" name="value" type="xs:string"/>
  </xs:sequence>
</xs:complexType>
```

## formMapAdapter - complex type



```
<xs:complexType name="formMapAdapter">
  <xs:complexContent>
    <xs:extension base="xmlAdapter">
      <xs:sequence/>
    </xs:extension>
  </xs:complexContent>
</xs:complexType>
```

## sourceEnum - simple type

### Adatok lehetséges forrásai

```
<xs:simpleType name="sourceEnum">
  <xs:annotation>
    <xs:documentation>
      <![CDATA[Adatok lehetséges forrásai.]]>
    </xs:documentation>
  </xs:annotation>
  <xs:restriction base="xs:string">
    <xs:enumeration value="P"/>
  </xs:restriction>
</xs:simpleType>
```

## IV. Kérdőívsablon XML séma (iFORM)

### xmlAdapter - complex type

A kulcs/érték adatpárok reprezentálására használt technikai jellegű összetett típusdefiníció (pl keys, properties.)



```
<xs:complexType abstract="true" name="xmlAdapter">  
  <xs:sequence/>  
</xs:complexType>
```

### attachmentField - complex type

Csatolmány mező