
التفاصيل خطوة بخطوة:

المستخدم يختار خدمة معينة 1.

- "المستخدم يفتح التطبيق، يرى قائمة الخدمات، ويختار خدمة مثل "تجديد جواز السفر": الوصف
- لجلب تفاصيل الخدمة API يستدعي Frontend الـ: الإجراء في النظام
- **Frontend:** طلب الـ

text

CollapseWrapCopy

GET /api/services/1

- **Backend:** رد الـ

json

CollapseWrapCopy

```
{
  "success": true,
  "data": {
    "id": 1,
    "serviceName": "تجديد جواز السفر",
    "serviceDescription": "قم بتجديد جواز سفرك بسرعة وسهولة",
    "isAvailable": true
  }
}
```

```
}  
  
}
```

2. النظام يعرض الحقول المطلوبة.

- بعد اختيار الخدمة، يظهر للمستخدم الحقول التي يجب ملؤها: **الوصف**

- **Frontend:** طلب الـ

text

CollapseWrapCopy

GET /api/services/1/fields

- **Backend:** رد الـ

json

CollapseWrapCopy

```
{  
  "success": true,  
  "data": [  
    {  
      "id": 1,  
      "fileName": "الاسم الكامل",
```

```
"description": "يرجى إدخال اسمك بالكامل"
},

{
  "id": 2,
  "fileName": "رقم الجواز",
  "description": "رقم جواز السفر الحالي"
}

]

}
```

- المستخدم يعرف أنه يحتاج لإدخال الاسم الكامل ورقم الجواز: النتيجة

3. النظام يعرض المستندات المطلوبة

- تتطلب بعض الخدمات مستندات مثل صورة الجواز أو إثبات الإقامة: الوصف

- **Frontend:** طلب الـ

text

CollapseWrapCopy

GET /api/services/1/documents

- **Backend:** رد الـ

json

CollapseWrapCopy

```
{
  "success": true,
  "data": [
    {
      "id": 1,
      "documentName": "صورة الجواز",
      "description": "يرجى رفع صورة واضحة من الجواز الحالي"
    },
    {
      "id": 2,
      "documentName": "إثبات الإقامة",
      "description": "يرجى رفع مستند يثبت مكان إقامتك الحالي"
    }
  ]
}
```

- المستخدم يعرف أنه يحتاج لرفع صورة الجواز وإثبات الإقامة: النتيجة

4. المستخدم يملأ البيانات ويرفع المستندات

- "المستخدم يدخل البيانات ويرفع الملفات، ثم يضغط "إرسال: الوصف"

- **Frontend: طلب الـ**

text

CollapseWrapCopy

POST /api/services/1/submit

Content-Type: application/json

- **بيانات الطلب:**

json

CollapseWrapCopy

```
{
  "serviceId": 1,
  "userId": 101,
  "serviceDataRequests": [
    {
      "fieldId": 1,
      "fieldValueString": "أحمد محمد"
    },
    {
      "fieldId": 2,
      "fieldValueString": "A12345678"
    }
  ]
}
```

```
],  
"documents": [  
  {  
    "documentId": 1,  
    "fileUrl": "https://example.com/uploads/passport.jpg"  
  },  
  
  {  
    "documentId": 2,  
    "fileUrl": "https://example.com/uploads/residence_proof.pdf"  
  }  
]  
}
```

5. يـخزن الـBackend الـبيانات

1. Requests: إنشاء طلب جديد في جدول

csharp

CollapseWrapCopy

```
var request = new Request
```

```
{
```

```
    ServiceId = requestDto.ServiceId,
```

```

        UserId = requestDto.UserId,
        Status = "Pending",
        CreatedAt = DateTime.UtcNow

};

await _context.Requests.AddAsync(request);
await _context.SaveChangesAsync(cancellationToken);

```

2. تخزين البيانات في جدول **ServiceData**:

csharp

CollapseWrapCopy

```

var serviceData = requestDto.ServiceDataRequests.Select(data => new ServiceData
{

    RequestId = request.Id,

    FieldId = data.FieldId,

    FieldValueString = data.FieldValueString

}).ToList();

await _context.ServiceData.AddRangeAsync(serviceData);
await _context.SaveChangesAsync(cancellationToken);

```

- مثال البيانات المحفوظة:

text

CollapseWrapCopy

Id | RequestId | FieldId | FieldValueString

1 | 2001 | 1 | "أحمد محمد"

2 | 2001 | 2 | "A12345678"

3. تخزين المستندات في جدول **ServiceDocuments**:

csharp

CollapseWrapCopy

```
var serviceDocuments = requestDto.Documents.Select(doc => new ServiceDocument
{
    RequestId = request.Id,
    DocumentId = doc.DocumentId,
    FileUrl = doc.FileUrl
}).ToList();
await _context.ServiceDocuments.AddRangeAsync(serviceDocuments);
await _context.SaveChangesAsync(cancellationToken);
```

- مثال المستندات المحفوظة

text

CollapseWrapCopy

Id | RequestId | DocumentId | FileUrl

1 | 2001 | 1 | "https://example.com/uploads/passport.jpg"

2 | 2001 | 2 | "https://example.com/uploads/residence_proof.pdf"

Frontend؟ كيف يتعامل الـ


1. عند دخول صفحة الخدمة:

- جلب الحقول GET /api/services/{id}/fields.
- جلب المستندات GET /api/services/{id}/documents.

2. عند إدخال البيانات ورفع المستندات:

- يجمع البيانات.
- يحوّل الملفات إلى روابط (fileUrl).
- يرسل POST /api/services/{id}/submit.

3. عند النجاح:

- "إعرض رسالة:  تم إرسال الطلب بنجاح"

