جامعة الشهيد حمّه لخضر - الوادي
Université Echahid Hamma Lakhdar - El-Oued

جامعة الشهيد حمّه لخضر - الوادي
Université Echahid Hamma Lakhdar - El-Oued

الجمهورية الجزائرية الديمقراطية الشعبي
وزارة التعليم العالي والبحث العلمي
جامعة الشهيد حمه لخضر- الوادي

كلية العلوم الدقيقة قسم الإعلام

الآلي

مذكرة نهاية التخرج

شهادة على الحصول متطلبات ضمن تدخل

ليسانس أكاديمي

اقتراح نظام متكامل لإدارة الوثائق عبر الإنترنت لفائدة المكتبة المركزية بجامعة الوادي

من إنجاز الطلبة:

- غنابزية حسان
- حنانشة سيف الإسلام
- غرايسة سعد

الاستاذ المشرف:

بـــالي معـــاذ

نوقشت يوم 06\10 \ 2020 أمام اللجنة المكونة من الاساتذة:

حمودة مريم	أستاذة محاضرة ب	رئيسة
شروق قطاس	أستاذة محاضرة ب	مقررة
بالي معاذ	أستاذ محاضر ب	مشرفا

الموسم الجامعي 2020/2019

الجمهورية الجزائرية الديمقراطية الشعبية

وزارة التعليم العالي والبحث العلمي

جامعة الشهيد حمه لخضر- الوادي

كلية العلوم الدقيقة **قسم الإعلام الآلي**

مذكرة نهاية التخرج

تدخل ضمن متطلبات الحصول على شهادة

ليسانس أكاديمي

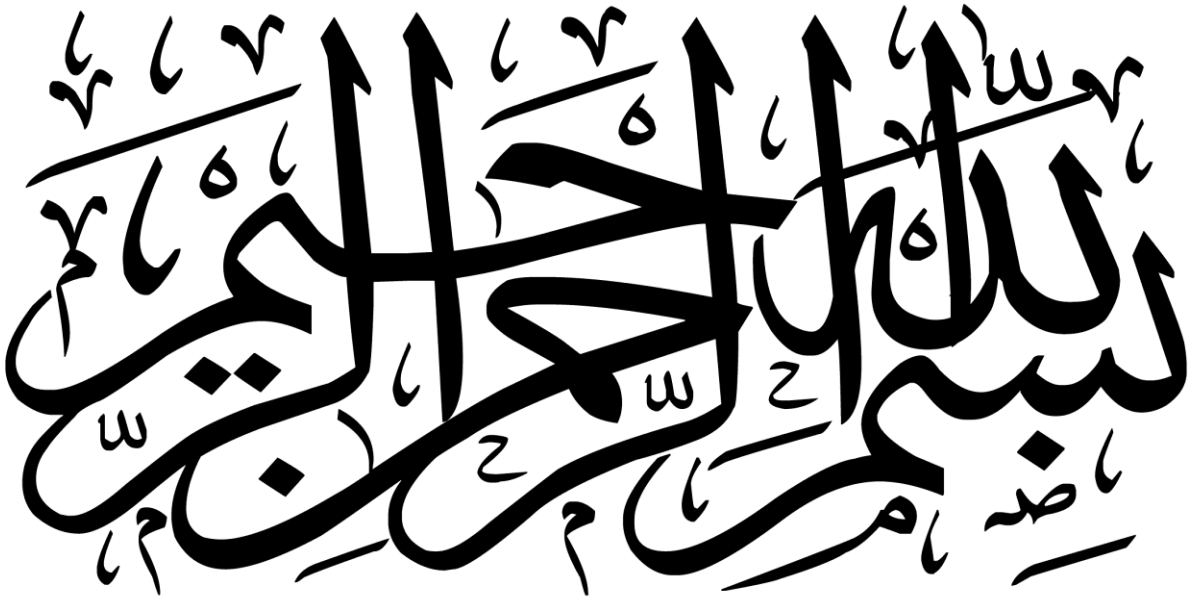
منصة للنشر الوثائقي عبر الانترنت للمكتبة المركزية للجامعة

من إنجاز الطلبة: **الاستاذ المشرف:**

- **غرايسة سعد** **بـــالي معـــاذ**
- **حنانشة سيف الإسلام**
- **غنابزية حسان**

نوقشت يوم __ \ __ \ 2019 أمام اللجنة المكونة من الاساتذة: ا

.....	.....	**رئيسا**
.....	.....	**مقررا**


بسم الله الرحمن الرحيم

الإهداء

إلى رمز التفاني والإخلاص، إلى من وضعتني على طريق الحياة وجعلتني ربط الجأش، وراعتني حتى صرت كبيراً حفظها الله وأطال عمرها

أمي الحبيبة

إلى منبت الخير والتضحية والإيثار

والدي الكريم

إلى مثال العطاء والكبرياء والتضحية

إخواني وأخواتي

إلى من سهر معنا وأرشدنا و تعب من أجلنا أستاذنا المتواضع و الخلوق

الدكتـــــــور بـــــــــــــالي معـــــــــــاذ

إلى كل الأساتذة الكرام أدام الله عليكم بالصحة و العافية

أهدي إليكم بحثي في

وأبلغكم إن إنهائي عملي لم يكن ليتم لولا دعمكم، وأتمنَّى أن ينال رضاكم.

شكر وتقدير

أول مشكور هو الله عز وجل

ثم والداي على كل مجهوداتهم منذ ولادتي إلى هذه اللحظات

أنتم كل شيء أحبكم في الله أشد الحب

إلى إخوتي و أخواتي

ويسرني أن أوجه شكري لكل من أرشدني أو وجهني أو ساهم معي في إعداد هذا البحث

بإيصالي للمراجع والمصادر المطلوبة في أي مرحلة من مراحله

وأشكر على وجه الخصوص أستاذي الفاضل الدكتور

بالي معاذ

على مساندتي وإرشادي بالنصح والتصحيح

كما أن شكري موجه لكل أساتذة قسم الإعلام آلي و إدارة كلية العلوم الدقيقة بجامعة الشهيد حمه لخضر

لتوفير أفضل الأحوال التي تلائم الطلبة أثناء الدراس

ملخص

يعرف نظام الأرشفة و إدارة السجلات إلكترونياً في المكتبات الجامعية العديد من المشاكل والعوائق أهمها نقص توفير الخدمات و وصعوبة التواصل مع المنخرطين ,لذلك قمنا في هذه المذكرة باقتراح نظام آلي يخلصنا من عديد مشاكل الارشفة , يسهل التعامل مع المنخرطين , يوفر بحث متقدم , والاعتماد على مزامنة آلية للمحتوى الرقمي .

abstract

The electronic archiving and records management system in university libraries knows many problems and obstacles, the most important of which is the lack of service provision and the difficulty of communicating with those involved, so in this memorandum we proposed an automated system that saves us from many archiving problems, facilitates dealing with those involved, provides advanced search,and relies on synchronization A mechanism for digital content.

Resume

Le système d'archivage électronique et de gestion des documents dans les bibliothèques universitaires connaît de nombreux problèmes et obstacles, dont le plus important est le manque de prestation de services et la difficulté de communiquer avec les personnes impliquées.Nous avons donc proposé dans ce mémorandum un système automatisé qui nous évite de nombreux problèmes d'archivage, facilite le traitement des personnes impliquées, fournit une recherche avancée et repose sur la synchronisation. Un mécanisme pour le contenu numériqu

فهرس المحتويات

فهرس المحتويات.....
فهرس الأشكال .....
مقدمة عامة 1
I دراسة الموجود 3
I- دراسة الموجود:..... 4
مقدمة..... 4
1- المكتبة المركزية للجامعة..... 4
1-1-تعريف المكتبة المركزية للجامعة: 4
1-2-أهدافها:..... 4
2-الأنظمة الالكترونية في المكتبة المركزي 4
2-1-نظام ال SYNGEB:..... 4
2-2-نظام الأرشفة الإلكتروني Catalogue :..... 5
3 - الأنظمة المستخدمة في المكتبات العالمية..... 7
3-1-نظام الدي سبيس Dspace..... 7
3-2-نظام iDOX Standard 9
3-3-نظام DocuWare 10
3-4-التقنيات المستخدمة في أنظمة الارشفة الالكترونية..... 10
خاتمة :..... 18
II- التصميم..... 19
مقدمة..... 20
1-التقنيات المستخدمة 20
1-1-تعريف UML (Unified Modeling Language) :..... 20
1-2-عملية التطوير (Processes de développement):..... 21
1-3- طريقة 2TUP (2TRACK UNIFIED PROCESS): 21
1-4- حالات استخدام Diagram Case Use :..... 22
2-الأنظمة الفاعلة في النظام: 23
2-1-مخطط العام لحالات الاستخدام في النظام: 23
2-2-نموذج التتابع لتحديث النظام الالي :..... 24
2-3- نمذجة مخطط النشاط للمسؤول: 27
2-4-نمذجة مخطط النشاط لتطبيق الاندرويد:..... 28
2-5-مخطط التسلسلي لتطبيق الاندرويد :..... 29
خاتمة :..... 30

III- التنفيذ 31
مقدمة: 32
1 - بيئة العمل 32
1-1-مكونات بيئة العمل 32
1-2- شرح مكونات بيئة العمل 33
2- اعداد سرفر و شرح كيفية عمله 36
2-1- تحديد بيئة العمل. 36
2-2- شرح مكونات بيئة العمل. 36
3- شرح واجهات البرامج..... 48
3-1-شرح api نظام الدي سبيس.(DSpace) 48
3-2- المجتمعات الكبرى والصغرى (المكتبة العامة و المكتبة الخاصة): 50
3-3-المجموعات (الكليات) 58
3-4-العناصر (محافظ الكتب): 63
3-5-شرح سرفر nodejs الخاص بالدي سبيس (.DSpace)..... 74
3-6-شرح سرفر nodejs العام. 79
4- شرح واجهات تطبيق الاندرويد 85
4-1-تسجيل الدخول 85
4-2-البحث 86
4-3-الصفحة الرئسية لتطبيق الاندرويد 87
5- شرح واجهات التطبيق المكتبي 88
5-1-تسجيل الدخول للتطبيق المكتبي..... 88
5-2-الصفحة الرئسية للتطبيق المكتبي: 89
الخاتمة 89
الخاتمة العامة..... 90
قائمة المصادر والمراجع..... 91
الملحق 93

فهرس الأشكال

الشكل I-1: الواجهة الرئيسية لموقع الأرشفة الإلكترونية (Catalogue) 6
الشكل I-2: مجالات استخدام REST API.....12
الشكل II-1:مخطط يوضح كيفية الرسم باستخدام UML.....21
الشكل II-2: مخطط حلات الاستخدام العام للنظام.....23
الشكل II-3: مخطط التتابع لتحديث النظام الالي.....24
الشكل II-4: جدول يوضح مفاهيم البيانات المرسلة.....25
الشكل II-4: مثال على شكل json لمفاهيم البيانات المرسلة. 26
الشكل II-4: مخطط النشاط للمسؤول.....27
الشكل II-5: مخطط النشاط لتطبيق الاندرويد.....28
الشكل II-6: مخطط التسلسلي لتطبيق الاندرويد.....29
الشكل II-7: مخطط لنمذجة نظامنا.....30
الشكل III-1: مخطط عام يلخص عملنا في هذا النظام 32
.....
الشكل III-2: اضافة nodemon في العنصر scripts 38
الشكل III-3: بنية المشروع.....39
الشكل III-4: طريقة استدعاء بعض ال framework 40
الشكل III-5: طريقة اضافة ال Middlewares الخاصة بالمشروع 40
الشكل III-6: طريقة اضافة الملفات التي تتعامل مع السيرفر 40
الشكل III-7: تشغيل السرفر 41
الشكل III-8: ارسال طلب الي السرفر 43
الشكل III-9: الحالة 200.....44
الشكل III-10: الحالة 400.....44
الشكل III-11: الحالة 500.....44
الشكل III-12: الحالة 404.....45
الشكل III-13: الحالة 500.....45
الشكل III-14: الحالة 403.....46
الشكل III-15: الرد من السرفر في الحالة 200.....46
الشكل III-16: الرد من السرفر في احد الحالات الخطاء.....47
الشكل III-17: ال localhost في المتصفح 47
الشكل III-18: ال localhost في ال postman 48
الشكل III-19:الواجهة الاساسية.....48
الشكل III-20: واجهة تسجيل الخروج.....49
الشكل III-21: واجهة حالة الجلسة 49
الشكل III-22: واجهة التحقق من تشغيلDSpace 50

الشكل III-23: واجهة الحصول على مصفوفة المجتمعات الكبيرة والصغيرة 51
الشكل III-24: واجهة اضافة مجتمع كبيرة 51
الشكل III-25: واجهة اضافة مجتمع صغير الى مجتمع كبير.....52
الشكل III-26: واجهة الحصول على مصفوفة المجتمعات العليا.....53
الشكل III-26: واجهة الحصول على مصفوفة المجتمعات العليا.....53
الشكل III–27: واجهة الحصول على مجتمع كبير واحد 54
الشكل III-28: واجهة الحصول على مجتمع صغير.....55
الشكل III-29: واجهة تحديث على المجتمع الكبير 55
الشكل III-30: واجهة الحصول على المجتمع الكبير بعد تحديثه.....56
الشكل III-31: واجهة حذف مجتمع الكبير 56
الشكل III-32: واجهة الحصول على المجتمع الكبير بعد حذفه.....57
الشكل III-33: واجهة الحصول على مجموعات مجتمع صغير واحد.....57
الشكل III-34: واجهة الحصول على المجتمعات الصغيرة لمجتمع كبير واحد 58
الشكل III-35: واجهة إضافة مجموعة.....59
الشكل III-36: واجهة الحصول على مصفوفة المجموعات.....60
الشكل III-37: واجهة الحصول على مصفوفة المجموعات.....61
الشكل III-38: واجهة اضافة عنصر في مجموعة.....61
الشكل III-39: واجهة تحديث مجموعة.....62
الشكل III-40: واجهة الحصول على مجموعة بعد التحديث.....62
الشكل III-41: واجهة حذف مجموعة.....63
الشكل III-42: واجهة الحصول على مجموعة بعد الحذف.....63
الشكل III-43: واجهة إضافة عنصر.....64
الشكل III-44: واجهة الحصول على العناصر 65
الشكل III-45: واجهة الحصول على البيانات الوصفية للعنصر 65
الشكل III-46: واجهة إضافة ملف للعنصر 66
الشكل III-47: واجهة الحصول على بيانات ملفات العنصر الواحد 67
الشكل III-48: واجهة الحصول على بيانات ملفات العنصر 68
الشكل III-49: واجهة الحصول على بيانات ملفات العنصر بعد التحديث 68
الشكل III-50: واجهة حذف البيانات الوصفية للعنصر 69
الشكل III-51: واجهة حذف العنصر.....69
الشكل III-52: واجهة إضافة اسم للملف.....70
الشكل III-53: واجهة الحصول على الملفات.....70
الشكل III-54: واجهة الحصول على رابط الملف 71
الشكل III-55: واجهة فتح الملف.....71
الشكل III-56: واجهة تحديث بيانات الملف.....72
الشكل III-57: واجهة تحديث الملف 72

الشكل III-58: واجهة حذف الملف 73
الشكل III-59: واجهة بنية المشروع 74
الشكل III-60: واجهة الملف الاساسي للسرفر 75
الشكل III-61: واجهة الاخطاء المحتملة 75
الشكل III-62: واجهة اظافة object json الى DSpace باستخدام postman 76
الشكل III-63: واجهة الدالة التي تقوم باظافة object json 77
الشكل III-64: واجهة الدالة التي تقوم باظافة object json 78
الشكل III-65: واجهة ملف التسميات العامة 78
الشكل III-66: واجهة بنية المشروع 79
الشكل III-67: واجهة تسجيل الدخول 80
الشكل III-68: واجهة التسجيل من أجل المدير 80
الشكل III-69: واجهة التسجيل من أجل المشرف 81
الشكل III-70: واجهة التسجيل من أجل الطالب 82
الشكل III-71: واجهة تسجيل الدخول للمسؤول 83
الشكل III-72: ادارة المسائيل 83
الشكل III-73: ادارة المنخرطين 84
الشكل III-73: ادارة الموظفين 84
الشكل III-74: تسجيل الدخول لتطبيق الاندرويد 85
الشكل III-75: واجهة البحث لتطبيق الاندرويد 86
الشكل III-76: واجهة الصفحة الرئيسة لتطبيق الاندرويد 87
الشكل III-77: تسجيل الدخول للتطبيق المكتبي 88
الشكل III-78: الصفحة الرئيسية للتطبيق المكتبي 89

مقدمة عامة

فرض التطور التكنولوجي الهائل في مجال تقنيات المعلومات على مستوى المكتبات، استخدام النظم الآلية لضبط الحجم الهائل من الإنتاج الفكري، و تقديم خدمات معلومات متنوعة بسرعة و بدقة عالية، حيث أن المكتبات في الوقت الراهن لا تقاس بكميات الأوعية الموجودة بداخلها، و لكن بالخدمات التي تقدمها هذه المكتبة، و كيفية تقديم هذه الخدمات إلى المستفيدين بأسرع وقت ممكن و في أي مكان، متخطية الحواجز الزمنية و المكانية التي كانت تمثل أهم حواجز انتشار المعلومات و الاستفادة منها، و لذلك أصبح من الضروري أن تتجه المكتبات إلى هذا التطور العلمي الحديث و هو الثورة التكنولوجية المتمثلة في البرمجيات و النظم الآلية المعتمدة على الحاسبات الآلية لدعم شتى العمليات و الوظائف التي تعمل بكفاءة و سرعة فائقة و سهولة. و من هذا المنطلق أردنا تطبيق هذه المفاهيم على موضوع دراستنا وهو اقتراح نظام متكامل لإدارة الوثائق عبر الإنترنت لفائدة المكتبة المركزية بجامعة الوادي.

الاشكالية

تسعى المكتبات لتطوير أنظمتها وخدماتها والحصول على احسن نظام ممكن ولكن

من خلال دراستنا لهذا الموضوع وجدنا العديد من المشاكل والصعوبات أهمها:

- رداءة نظام الأرشفة الإلكتروني الحالي, وعدم توافقه مع نظام ادارة المكتبة (syngeb).
- صعوبة التواصل بين إدارة المكتبة والطالب.

إضافة إلى ذلك لاحظنا عدم وجود موقع يتيح للطالب الإستفادة من خدمات المكتبة.

ما هي الحلول لعلاج هذه المشاكل.

النتائج المنتظرة

من خلال هذا العمل نامل ان نحقق جملة من الاهداف والنتائج وهي :

- إطلاق منصة للأرشفة الإلكترونية للمكتبة المركزية باستخدام نظام الدي سبيس (Dspace)

- توفير بيئة أكثر تنظيم لإدارة محتوى المكتبة.

- برمجة نظام تحديث آلي بين الـ(Syngeb) و(Dspace) يسمح بالتحديث الدوري للمحتوى الرقمي .

- تطبيق اندرويد يسهل التواصل بين المكتبة و الطالب .

هيكل و أجزاء البحث: سنقسم العمل إلى ثلاثة فصول أولها دراسة الموجود و الذي من خلاله سنتعرف بشكل دقيق على مختلف تفاصيل طريقة التعامل بهذا النظام و ذلك من خلال استراتيجية تعتمد على دراسة نظام المكتبة المركزية كما أننا سنقوم باستخراج أهم المشاكل الموجودة بالنظام و تقديم حلول أما الفصل الثاني فسيكون للتصميم حيث سنستعمل لغة النمذجة الموحدة حتى نتمكن من إعداد قاموس معطيات و في الفصل الأخير سنقوم بالعمل التطبيقي و إنجاز الأنظمة المذكورة سلفا, حيث يسمح لنا بحل للعديد من المشاكل.

I دراسة الموجود

1- المكتبة المركزية للجامعة
2- الأنظمة المستخدمة في المكتبة المركزية
3- الأنظمة المستخدمة في المكتبات العالمية

I- دراسة الموجود:

مقدمة

من خلال هذا الفصل سنقوم بدراسة دقيقة للنظام الموجود لجمع المعلومات اللازمة ,والالمام بكافة الجوانب المهمة و الكافية لكي نقوم عمل نظام متطور يتماشى مع متطلبات المكتبة.

1- المكتبة المركزية للجامعة

1-1-تعريف المكتبة المركزية للجامعة:

هي المكتبة الرئيسية للجامعة وتقوم بمهام الإدارة العامة للمكتبات الجامعية والإشراف الفني والتبادل ودعم المكتبات والكميات والأقسام بالمكتبيين المؤهلين والرصيد المكتبي والمكتبة المركزية تنطوي تحتها مكتبات كل الكليات والأقسام التابعة للجامعة و تعتبر هيكل من هياكل الجامعة وأيضا من أهم الوسائل التى تعتمد عليها الجامعة فى تحقيق أهدافها ، فهى تقوم بتوجيه مجموعاتها من المواد المكتبية لمساندة المناهج والمقررات الدراسية بالجامعة ، وتقوم بتقديم خدماتها للطلاب والباحثين تلبيةً لأغراض الدراسة والبحث .

1-2-أهدافها:

توفير مصادر المعرفة لخدمة التخصصات العلمية المختلفة بالجامعة.

ربط نشاط الجامعة بمتطلبات خطط الجامعة.

تطوير النظم المكتبية بما يتفق مع التطورات الحديثة في مجال خدمات المكتبات والمعلومات.

2-الأنظمة الالكترونية في المكتبة المركزي

2-1-نظام ال SYNGEB:

a) **المواصفات:** هو برنامج كامل وتطوّري يسمح بمعالجة جميع أنواع الوثائق (الكتب والدوريات والأطروحات والمقالات ...إلخ). يسمح هذا البرنامج كذلك بالمعالجة السريعة و الفعالة للوثائق الخاصة بك
b) **اللغات المعتمدة:** العربية والفرنسية.
c) **الدعم:** قرص التثبيت.
d) **مميزاته:**

فهرسة جميع أنواع المستندات: فهرسة بسيطة وفهرسة متدرجة.

إدارة التزويد: الطلبات المسبقة ، الطلبات (الكتب والدوريات) ، الهبات ، التبادلات ، الودائع وإدارة الميزانية.

إدارة المخزون وطباعة الفروق .

إدارة الاعارة الداخلية / الخارجية والإعارة بين المكتبات (ILL): يتم التسجيل عن طريق الرموز الشريطية أو لوحة المفاتيح .

يعتمد مبدأ عمله على نظام Client–Server .

e) عيوبه :

قاعدة بياناته تحتاج الى حيز كبير من وسائط التخزين .

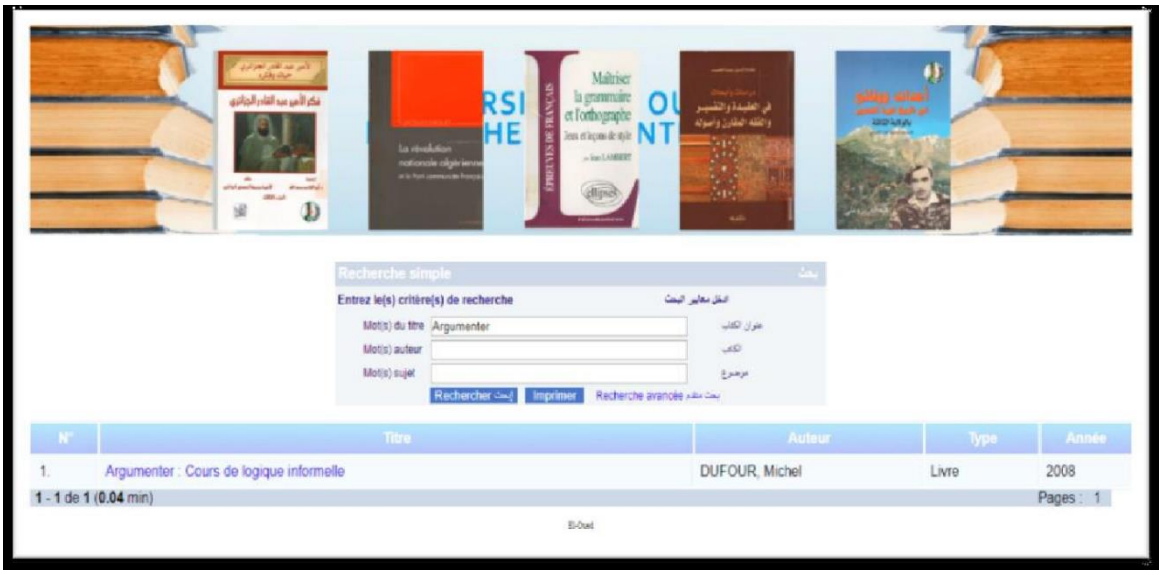
عدم توفير حماية لقاعدة البيانات .

عدم توفر التحديثات التقنية للنظام

2-2-نظام الأرشفة الإلكتروني Catalogue :

هو موقع إلكتروني (catalogue.univ–eloued.dz) يستخدم لنشر الفهرس الإلكتروني للمكتبة المركزية . ومن أهم الوظائف التي يتيحها هذا الموقع :

- البحث على المراجع بمختلف الصيغ (بحث عادي ومتقدم).
- اتاحة صور لأغلفة المراجع.
- توفير فهرس مصور لمحتويات المراجع.



الشكل I-1: الواجهة الرئيسية لموقع الأرشفة الإلكترونية (Catalogue)

مميزاته:

تنظيم الوثائق وربطها بنظام الفهرسة أليا حسب حقول خاصة .

تعدد وسائل الإدخال والإخراج حسب رغبة المستفيد .

بحث واسترجاع retrieval Search سريع جدا لما تم حفظه من صور للوثائق .

بحث متقدم متعدد المعايير: حسب العنوان، الكاتب، الموضوع، دار النشر،

عام النشر، نوع الملف,اللغة, . ISBN،ISSN.

نسبة الدخول إلى الموقع عالية.

عيوبه:

- صعوبة تحديث المحتوى على هذا النظام .
- عدم وجود مزامنة آلية بينها وبين نظام SYNGEB.

- لا يحتوي على الميزات التي تدعم (Search Engine Optimization) نظرا لأن تصميمه محلي أي لا يتوفر على المميزات الموجودة في الأنظمة المستخدمة في المكتبات العالمية.

3 - الأنظمة المستخدمة في المكتبات العالمية

3-1-نظام الدي سبيس Dspace

a) **تعريفه:** حزمة برمجية لإدارة المستودعات الرقمية مفتوحة المصدر تحديدا تستخدم لإنشاء مستودعات الوصول الحر لنشر المحتوى الرقمي الأكاديمي أو المؤسساتي. يتشابه Dspace وأنظمة إدارة ، المحتوى إلى حد ما مع أنظمة إدارة الوثائق، يقدم مجموعة خدمات كنظام لإدارة الأرشيف الرقمي، مع التركيز على التخزين على المدى الطويل، والإتاحة وحفظ المحتوى الرقمي. كما تركز على تقديم المحتوى الرقمي للمستخدمين النهائيين وتوفر مجموعة كاملة من الأدوات لإدارة (Digital re المحتوى. كما يعد الخيار الأكثر استخداما لبناء المستودعات الرقمية positories) للمؤسسات الأكاديمية، ويرجع ذلك إلى هيكلته القابلة لتعديل لتلبية احتياجات المستفيدين المستقبلية. يساعد البرنامج في إنشاء وفهرسة واسترجاع ك ويتمتع بتوافقية عالية للعمل مع ، كافة أشكال المحتوى الرقمي لمعظم نظم التشغيل، كما يلتزم البرنامج بالمعايير الدولية للبيانات الوصفيةالتالية.

المطور الأصلي	مخابر إتش بي, ومعهد ماساتشوستس للتكنولوجيا
المطور	Durespace
أول إصدار	نوفمبر 2002
آخر إصدار	x.7 2018
حالة التطوير	مستمر
لغة التطوير	Java
نظام التشغيل	مختلف اللغات
النوع	إدارة المستودعات المؤسسية
الترخيص	ترخيص BSD
الموقع الرسمي	www.dspace.org

b) البنية الهيكلية لنظام Dspace :

- الوحدة (Community) : تتألف الوحدات (المجموعات الرئيسة) من عدة المجموعات، ولا يمكن إيداع الوثائق فيها .
- المجموعة (Collection) : (المجموعات الفرعية) ولا تضم مجموعات أصغر، وفيها تجري عملية إيداع الوثائق .
- الوثيقة-العنصر (Item): و هي الوثائق المكونة للمجموعة، فالرسالة العلمية هي وثيقة واحدة داخل مجموعة الرسائل العلمية.
- سجل توصيف الوثيقة (Dublin Core metadata): هو ملف به يجري وصف الوثيقة ونظا م DSpace القائم على حقول الدبلن كور الأساسية للوصف الببليوغرا في ويمكّن النظام من تعديل سجل الوصف الببليوغرافي، ليواكب حاجات وصف كل مجموعة .
- التجميع (Bundle): و هو تجميع لمجموعة من الوثائق، يقوم به نظام DSpace ليكون ملفات ال HTML لعرض الوثيقة بوساطة شبكة الإنترنت، حيث يعرض الوثيقة الرقمية والسجل الببليوغرافي الخاص.

- الملف الرقمي (Bitstream) : و هو الملف الرقمي للوثيقة، فبعد توصيف الرسالة العملية , مثلا بوساطة سجل الوصف يجري إيداع ملفها الرقمي في الصيغ (formats) المتعارف عمليا للحفظ الرقمي ، zip، PDF،jpg وهذا الملف من الهيكمة مهم جدا حيث يمكن تطبيق الكثير من السياسات عليه ، كما يمكن لمجموعات من المستفيدين الاطلاع الكامل أو الاطلاع على البيانات الوصفية، والملف الرقمي و هذا ما يعد الأساس لأية مجموعة رقمية بغض النظر عن النظام الذي يقوم بإدارتها .
- معلومات صيغة الملف الرقمي (BitsreamFormat): يجري في هذا الجزء من الهيكلة تحديد صيغة الملف الرقمي، وحيث يتعرف نظام DSpace آليا على أكثر الصيغ المتعارف عليها، يطمع المتصفح على أية البرامج الضرورية لقراءة الملف، ويعتبر نظام DSpace نظاما مرنا ,حيث من تعريف الصيغ غير الموجودة بالنظام , و يمكّن المستخدم من إضافتها لتصبح جزءاً من النظام .

2-3-نظام iDOX Standard

تعريف idox standard : هو نظام إدارة وثائق مُتعدد اللغات ويُمكن إستخدامه على الويب أو الويندوز أو كتطبيق علي الهاتف الذكي. هو نظام إدارة وثائق إلكتروني يقوم بالإلتقاط والأرشفة والفهرسة وإستعادة الوثائق من خلال خيارات بحث مُتقدمة كما يجعل عملية مُعالجة الوثيقة آلياً.

المُميزات الرئيسية:

- مُتعدد اللغات.
- إستخدام API مما يوفر سهولة الدمج.
- إدخال وثائق إلكترونية.
- مسح فردي وجماعي.
- تعريف المستخدم بأنواع المستندات والفهرسة.
- فهرسة للنص بالكامل لكل الوثائق.
- الحصول على نتائج فورية للبحث.

- التعامل مع مُحتوي الملفات.
- إنشاء وإدارة المجلدات والمجلدات فرعية وأدراجها إلكترونياً.
- تخزين الوثائق بطريقة مُنظمة.

3-3-نظام DocuWare

تعريفه : هو نظام لإدارة الوثائق وتخزين الوثائق إلكترونيا ، يدعم 13 لغة ، وقد تم تركيبه في أكثر من 6،000 مؤسسه مع عشرات الآلاف من المستخدمين في جميع أنحاء العالم .

المميزات :

- تنفيذ متواضع من الإدارة وتكاليف التشغيل.
- إرتفاع قدرة الأمان للوثيقة .

و من أهم مميزات DocuWare أن سيرفر ه متوفر في ثلاث اصدارات مختلفة. الاصدارات تختلف في مداها من الميزات ، وهي مصممة لتلبية مختلف متطلبات الأعمال.

3-4-التقنيات المستخدمة في أنظمة الارشفة الالكترونية

A. عموميات :

a) واجهة برمجة التطبيقات (API)

هي خدمة تُقدّمها المواقع أو البرمجيّات الخاصة بالشركات الكبرى لتسهيل إضافة بعض من مميزاتها لبرامج أخرى.

B. خدمات الويب (Web service)

طريقة لنقل الدوال والإجراءات بين اجهزة مختلفة ,وغالبا يكون عبر استخدام HTTP , وله عدة صيغ و أشهرها رسالة SOAP و REST API.

a) HTTP:

هو بروتوكول نقل النص التشعبي (HyperText Transfer Protocol) وهو مجموعة من القواعد لنقل الملفات ، مثل النصوص والصور الرسومية والصوت والفيديو وملفات الوسائط المتعددة الأخرى على شبكة الويب العالمية و يعمل على رأس مجموعة بروتوكولات TCP / IP (بروتوكولات الأساس للإنترنت).

b) xml:

ليست لغة برمجيَّة إنَّها تنتمي لعائلة لغات الترميز "Markup Languages" الَّتي تنتمي إليها أيضًا لغة HTML المعروفة.

c) JSON :

ليس لغة برمجة، ولكن هو صيغة أو طريقة لتمثيل البيانات وهي مدعومة في معظم لغات البرمجة، حيث يتم تحويل أي نص مكتوب بصيغة JSON إلى كائن - Object، والعكس.

C. خدمة REST API

تعريفه :

إختصارال Representational state transfer هو نوع من أنواع ال API حيث يقوم هذا النوع بنقل البيانات بين العميل والخادم " Client وServer" " عن طريق البرتكول HTTP , فجميع العمليات تتم من خلال هذا البرتكول وعندما نقول عمليات فنحن نقصد العمليات الشائعة بعالم البرمجة و هي CREATE READ UPDATE DELETE إختصاراً لكلمة CRUD.

الشكل I-2: مجالات استخدام REST API

a) أهم الدوال (methods) التي يعتمد عليها الـ API Rest

أولا علينا أن نفهم طريقة عمل الـ api بغض النظر على برمجتها, فتقريبا جميع الـ apis الخاصة بالويب بالعالم تتفق على مجموعة من الأبجديات ومنها أنها تدعم لغة ,.,,json ، أي أنها تتبادل البيانات عن طريق

لغة json. أيضا يجب فهم كيفية عمل نظام CRUD في API أو ما يسمى اختصار ل UPDATE DELETE READ CREATE

هذه الدوال تترجم من خلالها نوع الطلب المرسل من client إلى server حيث سيفهم الـ API من خلال مسار رابط http + نوع الـ methods سيفهم ماذا تريده بالضبط ويقوم بمعالجة طلبك وتتلخص تعليمات البروتوكول http في التالي:

- GET تستخدم لجلب البيانات من السيرفر (قراءة البيانات READ)
- POST لإضافة بيانات (إنشاء محتوى CREATE).
- PUT تعديل بيانات موجودة بالفعل (التعديل UPDATE).
- DELETE لحذف بيانات موجودة بالسرفر(الحذف DELETE).

b) أهم النقاط التي تعتمد عليها الـ API Rest

يتم استخدام ال method المناسب لكل عملية طلب من طرف العميل Client فإذا أردت عرض معلومات يجب استخدام GET أما إذا كنت تريد التعديل استخدام PUT , اذا تريد الانشاء استخدم POST .

- header : تستخدم هذه الدالة لجلب ال request header من المتصفح وتخرج مصفوفة .
- body:يحتوي هذا العنصر على محتوى الذي تريد إظهاره لزوارك، سواءً كان نصاً، أو صورة، أو فيديو، أو لعبة، أو ملف صوتي، أو أياً كان.
- sission:تمكنك الـ Session في ASP.NET من تخزين متغيرات في صفحة معينة ومن ثم استرجاعها في صفحة أخرى أي أنها تسمح لك بتخزين متغيرات عند الانتقال بين الصفحات.

يتم إرسال البيانات من العميل client إلى الخادم server سواءاً كانت عن طريق parametres في الرابط أو عن طريق Body , و اذا اردت ارسال اي معلومات اخرى مثل مواثيق التحقق Authentication فيتم ارسالها عن طريق الـ Header

في الREST API لا حاجة لحفظ البيانات في session وذلك لأن جميع الطلبات يجب أن تكون state lees بمعنى أنها لا تحفظ أي بيانات او اي حالة في السيرفر والتواصل يكون مباشرة بين العميل والخادم عن طريق البرتوكول بشكل مباشر .

c) ميزات REST API:

فصل تام بين ال Client و server بحيث أن ال Client لا يستطيع ان يقوم بالتعديل أو الاضافة لأكواد السيرفر.

امكانية تشغيل التطبيق على عدة اجهزة Cross-device app بحيث قد يكون صفحة ويب او تطبيق موبايل او برنامج سطح مكتبي او اي تطبيق على أي منصة يمكنه التعامل وارسال طلبات HTTP .

سرعة تبادل البيانات بين العميل والخادم Client/Server بحكم اعتمادها على برتوكول HTTP .

d) العيوب :

نظام الحماية يحتاج الى مراقبة وتحقق من طرف المطور

.C DSpce REST API

DSpace REST API Server path: /rest

Index

- GET / - Return this page.
- GET /test - Return the string "REST api is running" for testing purposes.
- POST /login - Method for logging into the DSpace RESTful API. You must post the parameters "email" and "password".

  Example:

  "email=test@dspace&password=pass". Returns a JSESSIONID cookie which can be used for future authenticated requests.
- GET /shibboleth-login - Method for logging into the DSpace RESTful API with Shibboleth. You must configure Shibboleth to pass the Shibboleth session to this endpoint. This will return you a JSESSIONID cookie which must be included in future requests.
- GET /status - Method for retrieving information on the current authenticated user. The request must include the JSESSIONID cookie.
- GET /logout - Method for logging out of the DSpace RESTful API. The request must include the JSESSIONID cookie.

Communities

- GET /communities - Return an array of all communities in DSpace.
- GET /communities/top-communities - Returns an array of all top-leve communities in DSpace. GET /communities/{communityId} - Returns a community with the specified ID.

- GET /communities/{communityId}/collections – Returns an array of collections of the specified community.
- GET /communities/{communityId}/communities – Returns an array of subcommunities of the specified community.
- POST /communities – Create a new top-level community. You must post a community.
- POST /communities/{communityId}/collections – Create a new collection in the specified community. You must post a collection.
- POST /communities/{communityId}/communities – Create a new subcommunity in the specified community. You must post a community.
- PUT /communities/{communityId} – Update the specified community.
- DELETE /communities/{communityId} – Delete the specified community.
- DELETE /communities/{communityId}/collections/{collectionId} – Delete the specified collection in the specified community.
- DELETE /communities/{communityId}/communities/{communityId2} – Delete the specified subcommunity (communityId2) in the specified community (communityId).

Collections

- GET /collections – Return all DSpace collections in array.
- GET /collections/{collectionId} – Return a collection with the specified ID.
- GET /collections/{collectionId}/items – Return all items of the specified collection.
- POST /collections/{collectionId}/items – Create an item in the specified collection. You must post item.

- POST /collections/find-collection - Find a collection by name.
- PUT /collections/{collectionId} - Update the specified collection. You must post a collection.
- DELETE /collections/{collectionId} - Delete the specified collection from DSpace.
- DELETE /collections/{collectionId}/items/{itemId} - Delete the specified item (itemId) in the specified collection (collectionId).

Items

- GET /items - Return a list of items.
- GET /items/{item id} - Return the specified item.
- GET /items/{item id}/metadata - Return metadata of the specified item.
- GET /items/{item id}/bitstreams - Return bitstreams of the specified item.
- POST /items/find-by-metadata-field - Find items by the specified metadata value.
- POST /items/{item id}/metadata - Add metadata to the specified item.
- POST /items/{item id}/bitstreams - Add a bitstream to the specified item.
- PUT /items/{item id}/metadata - Update metadata in the specified item.
- DELETE /items/{item id} - Delete the specified item.
- DELETE /items/{item id}/metadata - Clear metadata of the specified item.
- DELETE /items/{item id}/bitstreams/{bitstream id} - Delete the specified bitstream of the specified item.

Bitstreams

- GET /bitstreams - Return all bitstreams in DSpace.

- GET /bitstreams/{bitstream id} – Return the specified bitstream.
- GET /bitstreams/{bitstream id}/policy – Return policies of the specified bitstream.
- GET /bitstreams/{bitstream id}/retrieve – Return the contents of the specified bitstream.
- POST /bitstreams/{bitstream id}/policy – Add a policy to the specified bitstream.
- PUT /bitstreams/{bitstream id}/data – Update the contents of the specified bitstream.
- PUT /bitstreams/{bitstream id} – Update metadata of the specified bitstream.
- DELETE /bitstreams/{bitstream id} – Delete the specified bitstream from DSpace.
- DELETE /bitstreams/{bitstream id}/policy/{policy_id} – Delete the specified bitstream policy.

خاتمة :

من خلال هذا ال فصل نكون قد جمعنا أكبر عدد ممكن من المعلومات حول طريقة عمل المكتبة المركزية وبعد اطلاعنا على الأنظمة المستخدمة للمكتبات العالمية وقع اختيارنا على نظام الدي سبيس (Despase) وذلك لمميزاته المذكورة سابقا.

II- التصميم

II- التصميم

1- التقنيات المستخدمة

2- الأنظمة الفاعلة في النظام

مقدمة

من خلال هذا الفصل سوف نقوم بنمذجة دقيقة للحصول على الجودة العالية, ولتحقيق نظام جيد او لتطوير مشروع متقدم من الضروري وضع نموذج لهذا النظام مع اتباع طريقة او منهج للتصميم, وكذلك سوف نقوم بتنفيذ نموذج لنظامنا مع تحفيز اختيارنا لعملية التصميم

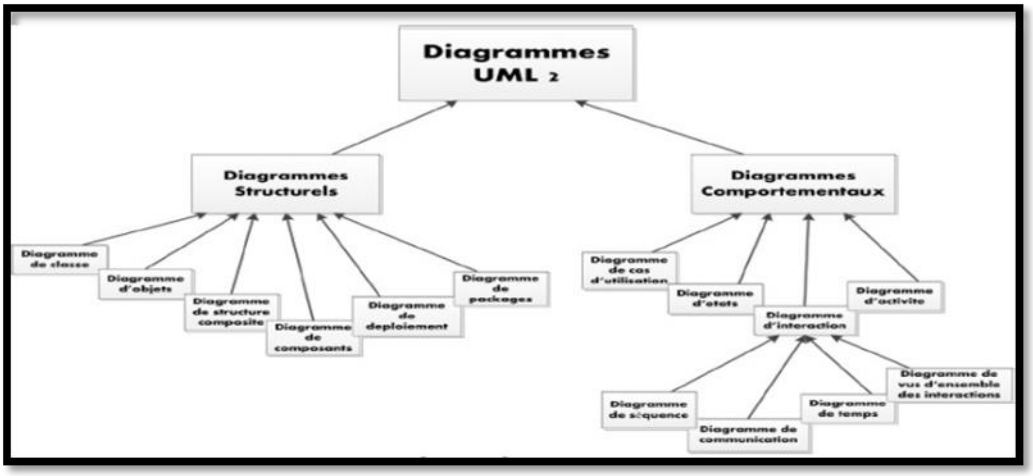
وتم اقتراح العديد من منهجيات التصميم بطريقة نموذجية ودقيقة للأنظمة المطورة حيث تعتمد هذه الطرق على مجموعة من المراحل والخطوات التي تشكل عملية التصميم.وبالتالي سنقوم بنمذجة نظام مترابط باستخدام لغة التصميم UML التي سنقوم باستعمالها في الجزء الوظيفي, وفي الجزء التقني سنحدد احتياجات النظام من اجل ضمان عمله في أحسن الظروف و دون حدوث اي أعطال ومشاكل تقنية محتملة.

1-التقنيات المستخدمة

1-1-تعريف (Unified Modeling Language) UML :

هي لغة نمذجة موحدة تستخدم لنمذجة تطبيق برمجي بطريقة قياسية في إطار التصميم الموجه للكائن.
يمكن لـ UML تغطية دورة حياة البرنامج من مواصفات المتطلبات إلى التشفير من خلال توفير وسائل متعددة لوصف الممثلين ونمذجتهم واستخدام النظام وسلوك الكائن وتدفق الرقابة الداخلية للعمليات ومكونات التنفيذ وعلاقاتها ، بغض النظر عن أساليب التنفيذ. UML هو معيار النمذجة الكائن الأكثر استخداماً اليوم, يتم تنظيمها حول ثلاثة عشر نوعًا من الرسوم البيانية ، كل منها مكرس من خلال تمثيل المفاهيم الخاصة لنظام

البرمجيات. تنقسم هذه الأنواع من المخططات إلى مجموعتين رئيسيتين: المخططات الهيكلية والمخططات السلوكيةكما هو موضح في الشكل أدناه :



الشكل II-1:مخطط يوضح كيفية الرسم باستخدام UML.

2-1-عملية التطوير (Processes de développement):

العمليات الموحدة(Les processes unifies):

عملية موحدة هي عملية مبنية على UML (لغة النمذجة الموحدة). بشكل أكثر تحديدًا ، هذه هي أفضل الممارسات لتطوير الكائن .
في دراستنا اخترنا العمل مع عملية TUP - لأنها تستهدف مشاريع باي حجم ، ولها مكان كبير في مجال التكنولوجيا ومخاطر المشاريع.

3-1- طريقة 2TUP (2TRACK UNIFIED PROCESS):

هي عملية تطوير البرمجيات تقوم بتنفيذ طريقة العملية الموحدة. تقدم TUP2 دورة تطوير تفصل الجوانب التقنيةعن الجوانب الوظيفية.
يبدا بدراسة اولية تتكون اساسا من تحديد العناصر الفاعلة التي تتفاعل مع النظام المراد بناءه والرسائل التي يتبادلها الممثلون والنظام ,لإنتاج المواصفات ونموذج السياق ,ثم يتم توضيح العملية حول ثلاث مراحل اساسية:

- ✓ **الجزء التقني(Une branche technique)**:

يستفيد الجزء الفني من المعرفة الفنية و / أو القيود الفنية.التقنيات المطورة للنظام مستقلة عن الوظائف المراد تنفيذها.

- ✓ **الجزء الوظيفي (Une branche fonctionnelle)**:

يستفيد الجزء الوظيفي من معرفة أعمال الشركة, جمع هذا الفرع احتياجات العمل ، مما ينتج عنه نموذج يركز على أعمال المستخدم النهائي.

- ✓ **الجزء من التحقيق (Une branche de conception et réalisation):**

تتكون مرحلة التنفيذ من الجمع بين الفرعين ، مما يسمح بتصميم أحد التطبيقات وأخيرًا تقديم حل يتكيف مع الاحتياجات.

بعد تحديد الاحتياجات وانتقاد الحلول القائمة ، ندرس في بقية الفصل مرحلة تصميم نظامنا.

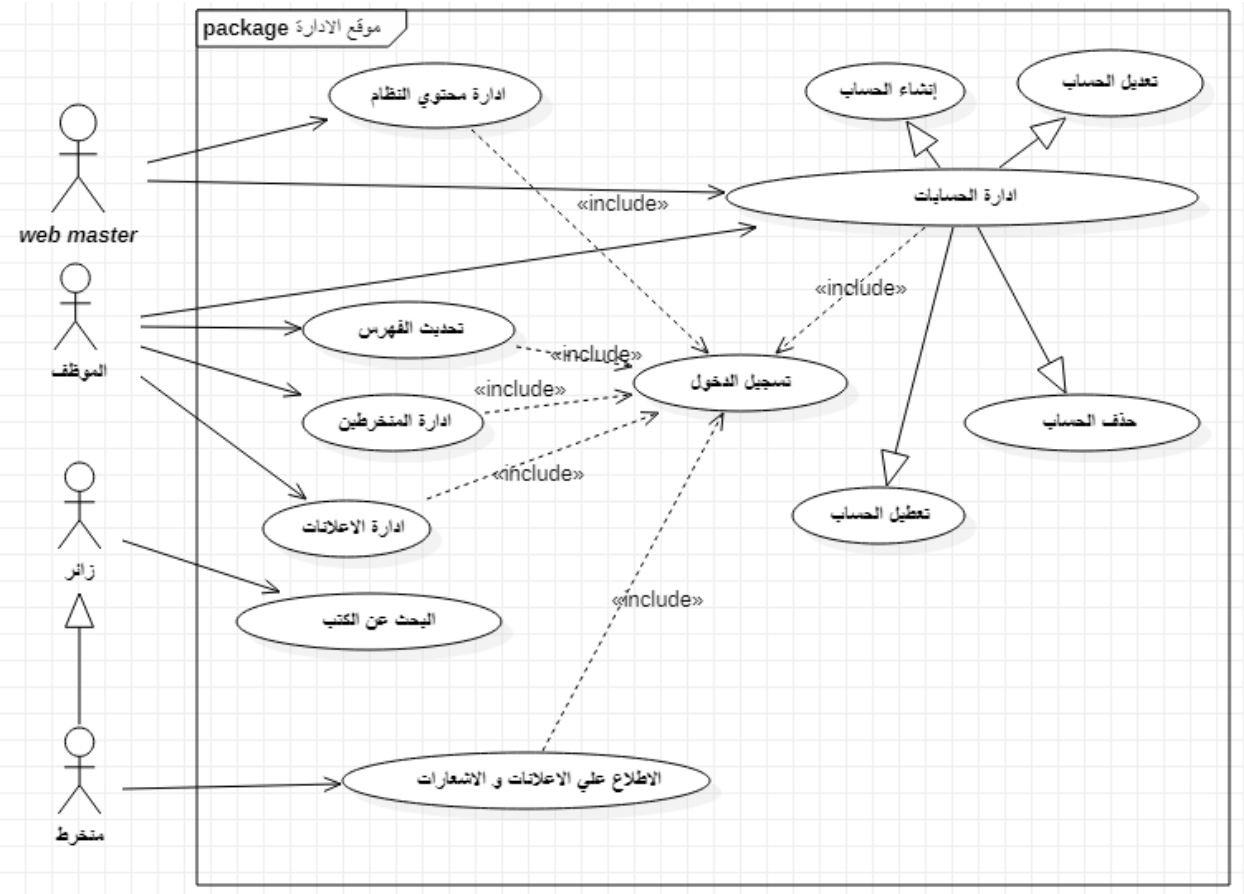
4-1- حالات استخدام Diagram Case Use :

هي وصف لخطوات أو اجراءات بين المستخدم ونظام البرمجيات الذي يقود المستخدم إلى شيء أكثر تجريد مثل نظام برمجة خارجي أو عملية مفيد. يمكن أن يكون المستخدم أو الممثل شخص ا أو شيئا يدوية .المطلب الثاني: مخطط الفئة class de Diagram هو نوع من ثابت هيكل الرسم التي تصف بنية النظام من خالل إظهار طبقات النظام، وخصائصها، والعمليات والعالقات بين الكائنات المطلب الثالث: مخطط التسلسل sequence de Diagramيعرض الرسم البياني التسلسلي تفاعالت الكائنات مرتبة في تسلسل زمني .يصور الكائنات والفئات المتضمنة في السينار يو وتسلسل الرسائل المتبادلة بين الكائنات الالزمة لتنفيذ وظيفة السيناريو. بإدراكات حالة الستخدام في العرض المنطقي للنظا ترتبط المخططات التسلسلية عادة م قيد التطوير . ا تسمى المخططات التسلسلية أحيان مخططات الحداث أو سيناريو هات الحدث . ُظهر المخطط التتابعي، كخطوط عمودية متوازية وكأسهم أفقية ، الرسائل المتبادلة بينها ، بالترتيب الذي تحدث به.يسمح ذلك بتحديد سيناريوهات وقت التشغيل البسيطة بطريقة رسومية.

2-الأنظمة الفاعلة في النظام:

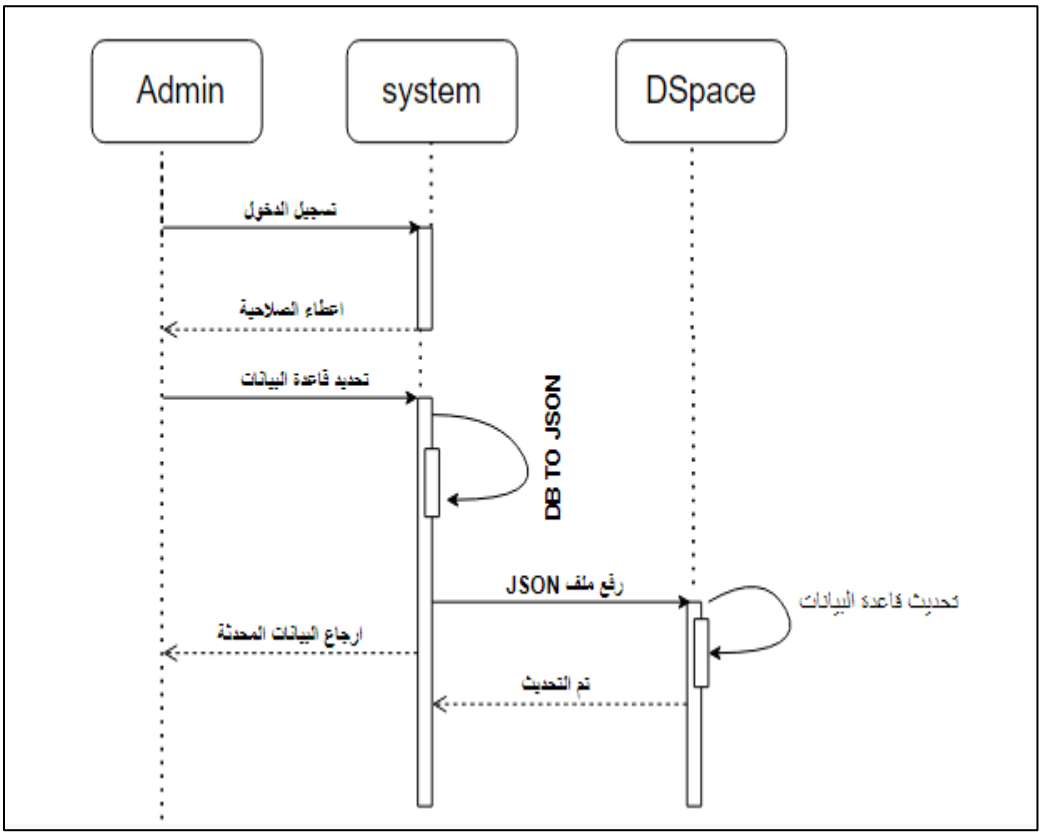
2-1-مخطط العام لحالات الاستخدام في النظام:

في هذا المخطط لدينا اربعة اشخاص



الشكل II-2: مخطط حلات الاستخدام العام للنظام

2-2-نموذج التتابع لتحديث النظام الالي :



الشكل II-3: مخطط التتابع لتحديث النظام الالي

المعلومات لتسجيل الدخول هي :

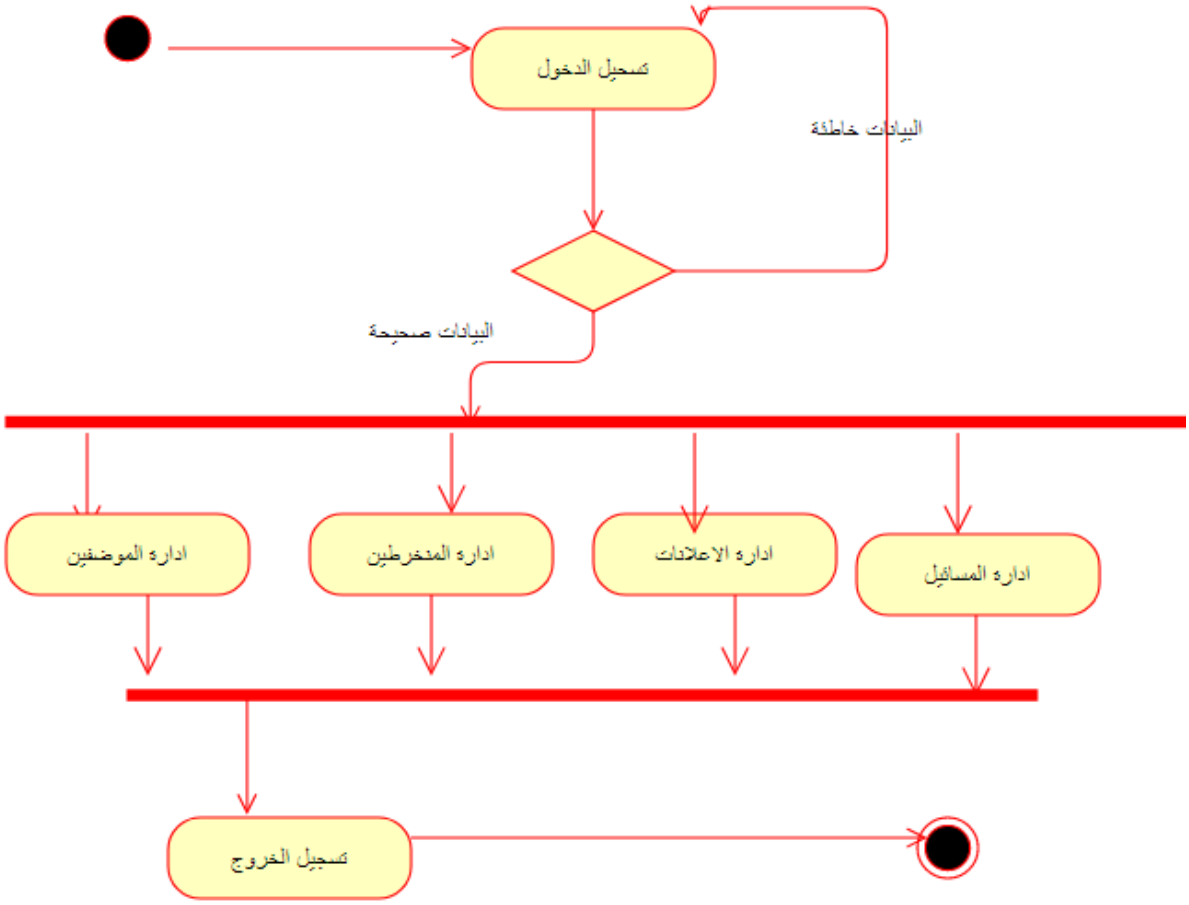
- البريد الاكتروني.
- كلمة المرور.

json		**- syngeb**
communityInfo	name	dbo_PARAMETRE.AGENCE
collection	name	dbo_SPECIALITE.SPE_LIBELLE_AR
	shortDescreption	id_sengeb
metadata	dc.title	DOC_TITRE_PROPRE
	dc.title.alternative	DOC_TITRE_COMPLEMENT
	dc.contributor.author	dbo_NOTICE_AUTEUR--X--dbo_VEDETTE.dbo_VEDETTE.VED_NOM
	dc.type	dbo_TYPE_DOCUMENT.TYP_TYP_LIBELLE_AR
	dc.contributor.author	dbo_VEDETTE.VED_NOM
	dc.language	LAN_ID
	dc.contributor.editor	DOC_LIEU_EDITION
	dc.date.issued	DOC_ANNEE
	dc.contributor.subject	dbo_NOTICE_EXEMPLAIRE--X--dbo_VEDETTE.dbo_VEDETTE.VED_NOM
	dc.identifier.isbn	DOC_ISBN
	dc.identifier.other	dbo_NOTICE_CLASSIFICATION.CLA_ID
databitstreams	name	docment_[DOC_ID].pdf
	description	DOC_TITRE_PROPRE, dbo_TYPE_DOCUMENT.TYP_TYP_LIBELLE_AR
	file	base64

الشكل II-4: جدول يوضح مفاهيم البيانات المرسلة.

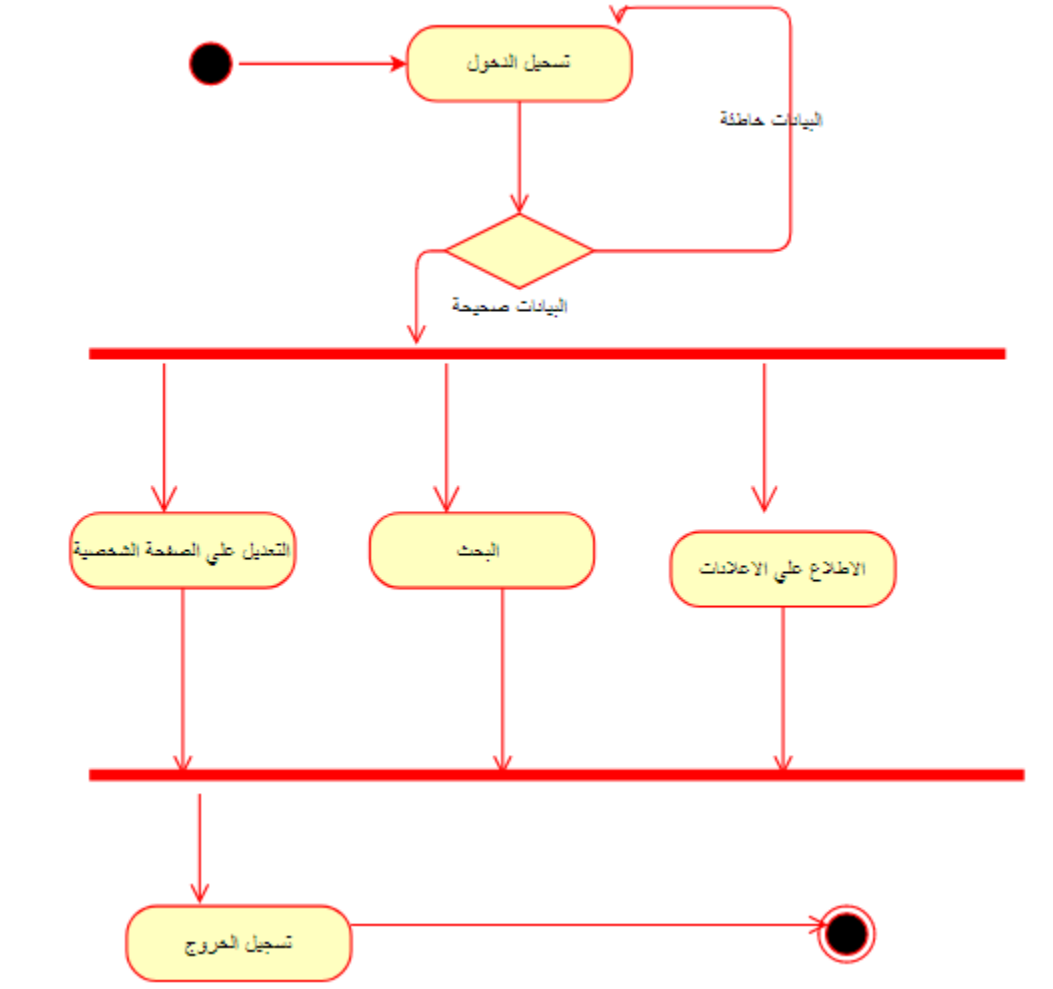
الشكل II-4: مثال على شكل json لمفاهيم البيانات المرسلة.

3-2- نمذجة مخطط النشاط للمسؤول:



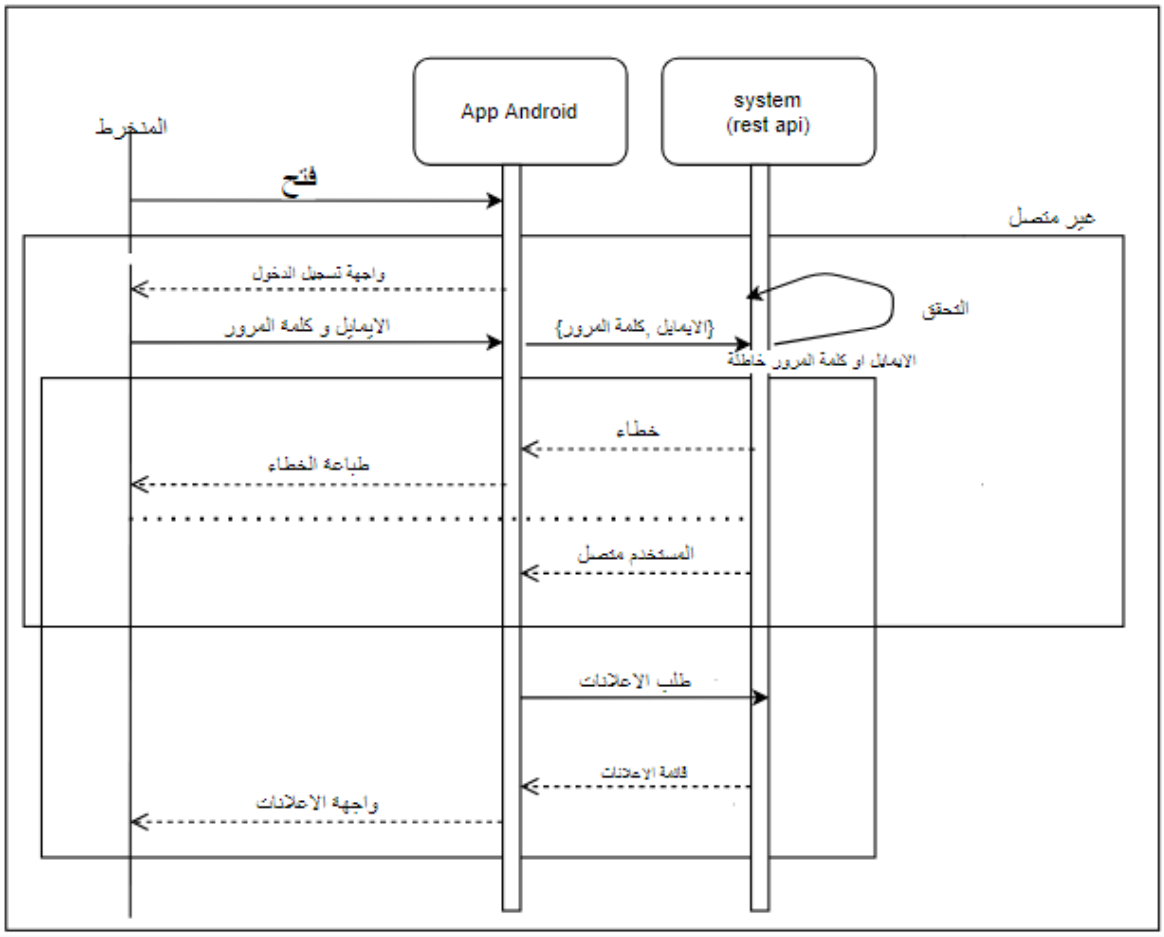
الشكل II-4: مخطط النشاط للمسؤول

4-2-نمذجة مخطط النشاط لتطبيق الاندرويد:



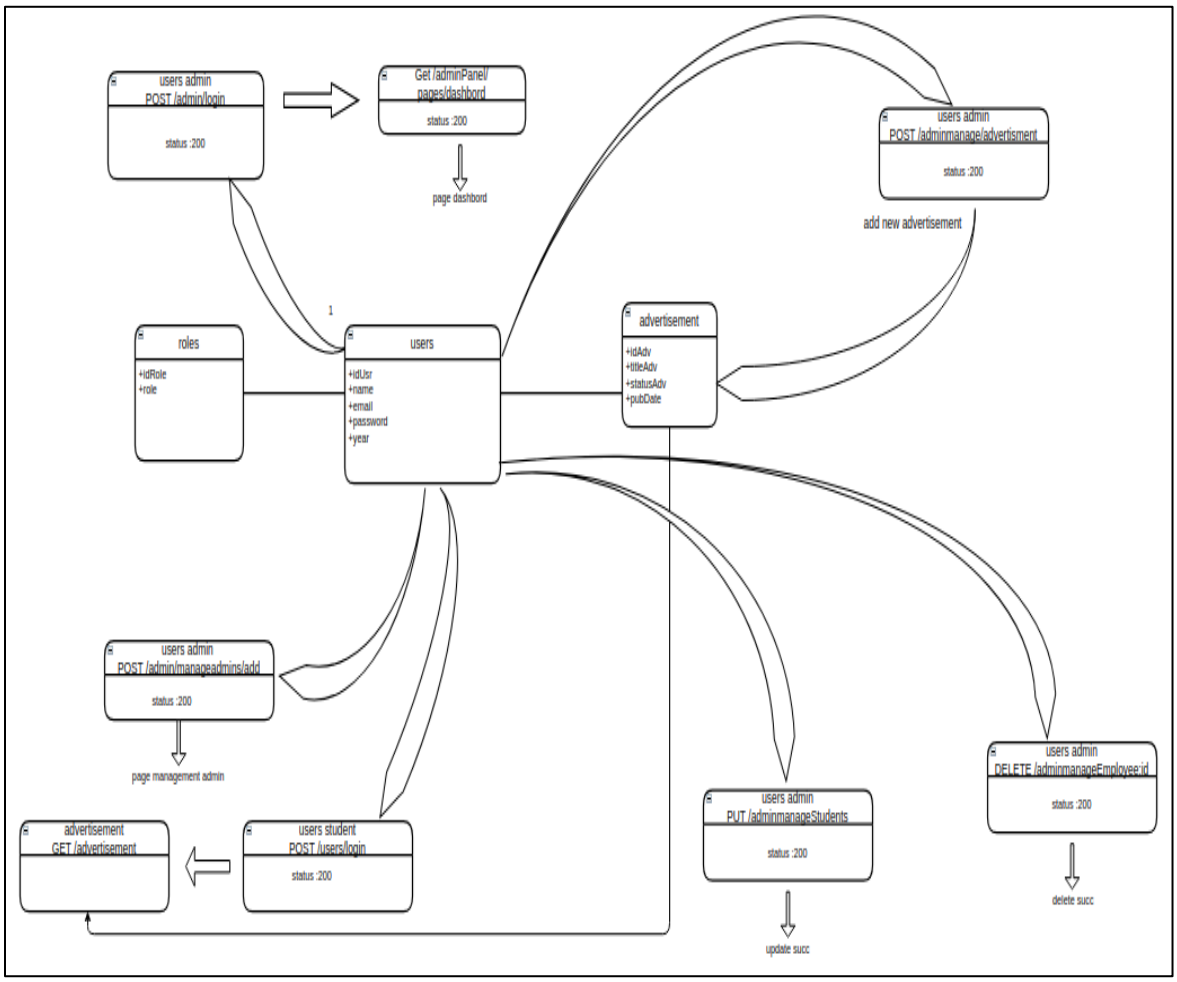
الشكل II-5: مخطط النشاط لتطبيق الاندرويد

5-2-مخطط التسلسلي لتطبيق الاندرويد :



الشكل II-6: مخطط التسلسلي لتطبيق الاندرويد

لقد اعتمدنا الطريقة المقترحة من طرف Visual Paradigm [1]:



الشكل II-7: مخطط لنمذجة نظامنا

خاتمة :

في سياق هذا الفصل، شرعنا في نمذجة نظامنا قبل البدء في تنفيذه. ثم تقديم نتائج النمذجة على هيئة صور UML.

في الفصل التالي، سنناقش بالتفصيل تنفيذ نظامنا.

[1] https://www.visual-paradigm.com/support/documents/vpuserguide/276/3420/85154_modelingrest.html

III-التنفيذ

1 : بيئة العمل.

2 :اعداد سرفر و شرح كيفية عمله.

3 : شرح واجهات البرامج.

4 : تطبيق الاندرويد.

5 : التطبيق المكتبي.

مقدمة:

بعد العمل النظري ،تحديد متطلبات التطبيق والتصميم بدقة سنحاول في هذا الفصل تنفيذ ما ورد في التصميم وتجسيد النظام عمليا و بدقة ،سوف نقوم بانشاء برنامج مكتبي ، تطبيق اندرويد و موقع ويب وهذا الأخير هو النقطة الرابطة بين باقي التطبيقات ببعضها عن طريق API

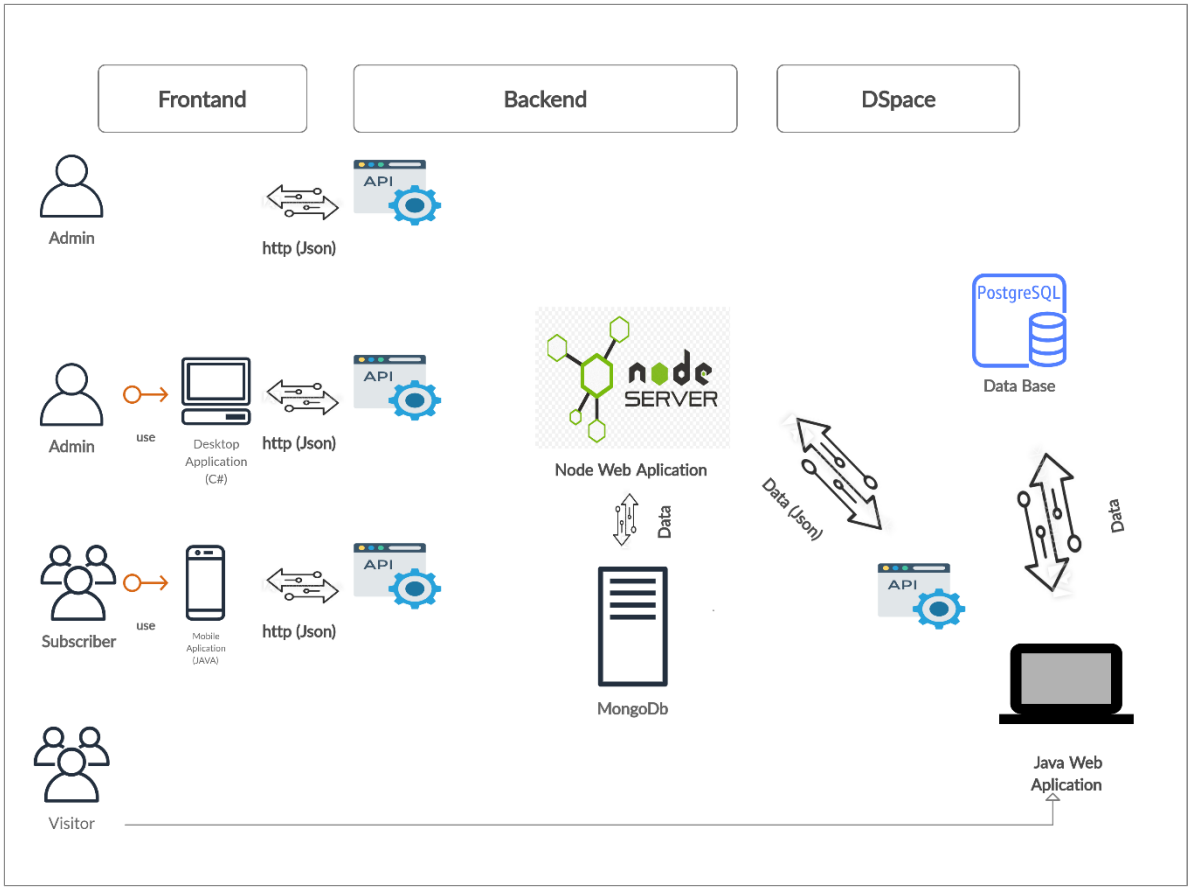
وهذا الجزء هو خاتمة مشروعنا سنقوم بانجاز التصميم اعتمادا على جملة من المعايير منها بيئة العمل ، نظام ادارة قواعد البيانات ، ادوات تصميم الموقع

1 - بيئة العمل

في هذا الجزء سوف نقدم وصف بيئة العمل اللازمة لتطوير التطبيقات من خلال سياق العمل المطلوب.

1-1-مكونات بيئة العمل .

الشكل التالي يوضح مكونات بيئة و طريقة عمل نظامنا.



الشكل III-1: مخطط عام يلخص عملنا في هذا النظام

2-1- شرح مكونات بيئة العمل

- **HTML5**

لغة ترميز النص التشعبي) بالإنجليزية(Hyper Text Markup Language :، هي لغة ترميز تستخدم في إنشاء وتصميم صفحات ومواقع الويب، وتعتبر هذه اللّغة من أقدم اللّغات وأوسعها استخداما في تصميم الهيكلة الاولية لصفحات الويب. وهذا الاصدار الخامس منها نظرا لتقدمه و توسعات استخداماته، و هي عبارة عن مجموعة من الاوسمة الفاتحة و الغالقة لتشكل لنا ملف html ويكون امتداد الملف ""html.

ويتم تشغيل هذا الملف عن طريق المتصفحات او احد برامج كتابة الاكواد عن طريق تطبيقات الهواتف او تطبيقات سطح المكتب.

- **CSS3**

أوراق الأنماط المتتالية (Cascading Style Sheets)

هي لغة تنسيق لصفحات الويب او لتصميم التطبيقات وصممت خصيصا لعزل الشكل العام لتصميم عن باقي الاكواد وينطبق ذلك عن كل شيئ ذو جمالية في الواجهة من خطوط، صور و خلفيات ...الخ. يمكن كتابة اكوادها اما داخل ملف إتش تي إم إل او تضمينه فيه و كتابة الاكواد في ملف خارجي، و يكون امتداد الملف "..css"

- **Bootstrap**

كان يسمى سابقا "تويتر بوتستراب" (Twitter Bootstrap) هي مجموعة من الأدوات مفتوحة المصدر لإنشاء مواقع الويب وتطبيقاته. قامت شركة تويتر بتطويرها مع

Jacob Thornton وأنشئت لمساعدة المصممين والمطورين على بناء منتجات مذهلة بسرعة وفعالية الهدف منها توفير مكتبة واسعة مرنة وموثقة بشكل جيد للتصاميم "مكررة الاستعمال" المبنية باستخدام HTMLو CSSو JavaScriptكي يقوم المطورون باستعمالها والإبداع باستخدامها. توسع المشروع ليصبح يعرف بـ Bootstrap ويستقل عن تويتر لأنه كما صرح المسؤولون أصبح مشروعا كبير يتضمن العشرات من العناصر حيث أصبح المشروع الأكثر شعبية على موقع GitHub بعد أن حصل على أكثر من 13,000 متابع و2000 تطوير فرعي. وهذا الاصدار هو اخر اصدار لها.

- **JavaScrip**

هي لغة برمجة عالية المستوى تستخدم أساساً في متصفحات الويب لإنشاء صفحات أكثر تفاعلية. يتم تطويرها حالياً من طرف شركة موزيلا. تُستخدَم هذه الاخيرة لتوفير تطبيقات ويب بما في ذلك الألعاب؛ وهي مُستعمَلة من أغلبية المواقع، وتدعمها جميع المتصفحات تقريبًا دون الحاجة إلى إضافات خارجية ، و يوجد بها العديد من framework

منها:

- Node.js

- Vue.js
- React

- **REST Api**

هو اختصار لـ .REpresentational State Transfer إنه أسلوب معماري لأنظمة الوسائط التشعبية الموزعة وقد قدمه روي فيلدنج لأول مرة في عام 2000 في أطروحته الشهيرة. مثل أي نمط معماري آخر ، فإن REST لديها أيضًا 6 قيود إرشادية خاصة بها والتي يجب تلبيتها إذا كانت الواجهة بحاجة إلى الإشارة إليها على أنها .RESTful هذه المبادئ مذكورة أدناه.

- العميل - الخادم
- قواعد تقديم الطلب
- قابل للتخزين المؤقت
- واجهة موحدة
- نظام الطبقات
- كود عند الطلب (اختياري)
- حالات عدة عمليات شائعة واشهرها:
- انشاء ،قراءة ، تحديث ، حذف

create,read,update,delete والاسم المختصر لها Crud.

- **MongoDB**

الاسم مشتق من("humongous")وهي نظام قاعدة بيانات مفتوحة المصدر مطورة ومدعومة من gen. 10وهي جزء من عائلة أنظمة قواعد بيانات .NoSQL بدلاً من تخزين البيانات في جداول كما معروف في قاعدة البيانات المترابطة، تقوم مونغو دي بي بتنظيم البيانات كمستندات شبيه JSON بمخططات ديناميكية (مونغو دي بي تدعي بصيغة (BSON فتجعل دمج البيانات في أنواع محددة من التطبيقات أسهل وأسرع.

بدات شركة 10 genبتطوير مونغو دي بي في أكتوبر2007، وهي تستخدم الآن من قبل شبكات فاياكوم سي بي إس الإعلامية المحلية, كريجزليست, فورسكوير و آدهار. وهي تعتبر أشهر نظام إدارة قواعد بيانات NoSQL.

- **NodeJs**

هو نظام برامج مصمم لكتابة تطبيقات إنترنت قابلة للتوسع كخوادم الويب. تم اختياره بواسطة InfoWorld لجائزة تقنية العام في 2012.أنشئت Node.js على يد ريان دال ابتداءً في عام 2009، وقامت برعاية نموهاJoyent ، مشغله. يتألف Node.js من في

8التابع لجوجل مع العديد من المكتبات المدمجة. وأخذت بعض مواصفات مشروع.CommonJS تكتب برامج node.js بلغة الجافاسكربت، باستخدام نمط حدثّي التوجه، إدخال وإخراج غير متزامنين للحد من

النفقات وتحقيق أكبر قدر من قابلية التوسع. وعلى عكس أغلب برامج الجافاسكريبت فهي لا تشتغل على متصفح الويب، ولكن عوضا عن ذلك فيتم تشغيلها من طرف الخادم، وهذا هو اخر اصدار لها.

- **Github**

gitهو نظام تحكم في الإصدار ، لكن ماذا يعني ذلك؟ عندما ينشئ المطورون شيئًا ما (تطبيق ، على سبيل المثال) ، فإنهم يجرون تغييرات مستمرة على الكود ، ويطلقون إصدارات جديدة حتى وبعد الإصدار الرسمي الأول (غير التجريبي). Git هو نظام التحكم في الإصدار المفضل لمعظم المطورين ، نظرًا لأنه يتمتع بمزايا متعددة على الأنظمة الأخرى المتاحة. يقوم بتخزين تغييرات الملفات بشكل أكثر كفاءة ويضمن سلامة الملفات بشكل أفضل. ومن اهم مزاياه انه يمكن تطوير الكود بعدد كبير من المبرمجين الذين يعملون في نفس المشروع بتطويره وفي نفس الملفات.

hub هو امتداد لبوابة سطر الأوامر يساعدك على أداء مهام GitHub اليومية دون

مغادرة الجهاز.

- **Docker**

تقنية جديدة ونوعية شبيهة بالنظم الافتراضية (بالإنجليزية:(Virtual System ولكنها تقوم بعملية إنشاء نسخ افتراضية للنظام، وذلك من خلال إنتاج حاويات تتشارك فيها النواة مع النظام المضيف، حيث أن هذه الحاويات تلعب دور غلاف حاوي لتطبيق ما، بحيث يصبح قائما بذاته، مستقل؛ أي أن هذا التطبيق آو المشروع بكامل احتياجاته ليعمل يكون داخل حاوية (قد تكون حاوية واحدة أو عدة حاويات تتخاطب في ما بينها), وبالتالي فان ذلك يقلل من استهلاك موارد الجهاز . الجدير بالذكر أيضا أن القائمين على مشروع Docker قاموا بتوفير ما يسمى بـ Docker Hub يمكن ان تتشارك فيه صور المشروع (إذا كان مفتوح المصدر), حيث يوجد به الكثير من الحلول المفتوحة المصدر والقابلة للاستخدام.

- **DSpace 6**

هو عبارة عن حزمة برمجيات مستودع مفتوح المصدر تُستخدم عادةً لإنشاء مستودعات وصول مفتوحة للمحتوى العلمي أو المنشور الرقمي. بينما يشترك DSpace في بعض الميزات المتداخلة مع أنظمة إدارة المحتوى وأنظمة إدارة المستندات ، فإن برنامج مستودع DSpace يخدم حاجة معينة كنظام أرشيف رقمي ، يركز على التخزين طويل الأجل والوصول إلى المحتوى الرقمي والحفاظ عليه.وهذا هو لاصدار قبل الاخير.

2- اعداد سرفر و شرح كيفية عمله

في هذا الجزء سوف نقوم بدراسة عملية عن اللغات المستخدمة لبرمجة هذا السرفر وشرح كيفية عمله.

2-1- تحديد بيئة العمل.

بيئة العمل:

- Npm
- NodeJs
- Framwork NodeJs
- Middleware NodeJs
- Postman
- localhost

2-2- شرح مكونات بيئة العمل.

- **Npm**

هو أكبر سجل برمجيات في العالم ، ويحتوي السجل على أكثر من 800000 حزمة كود ، ويستخدم مطورو المصادر المفتوحة npm لمشاركة البرامج،يتم استخدامها لتثبيت الحزم التي يمكن ان نحتاجها في مشروعنا.

- **NodeJs**

هي بيئة تشغيل مفتوحة المصدر تعمل عبر الأنظمة الأساسية لتطوير تطبيقات الخادم والشبكات. تتم كتابة تطبيقات Node.js بلغة JavaScript ، ويمكن تشغيلها خلال وقت تشغيل Node.js على أنظمة التشغيل OS X و Microsoft Windows و Linux.

يوفر Node.js أيضًا مكتبة غنية من وحدات JavaScript المتنوعة التي تبسط تطوير تطبيقات الويب باستخدام Node.js إلى حد كبير. أهم جوانب إطار عمل الويب هي - بنيته وميزاته (مثل دعم التخصيص والمرونة وقابلية التوسع والأمان والتوافق مع المكتبات الأخرى وما إلى ذلك.

هذه هي معادلة بسيطة عنها:

Node.js = Runtime Environment + JavaScript Library

- **Framwork NodeJs**

هو عبارة عن منصة لتطوير تطبيقات البرمجيات. يوفر أساسًا لمطورين بناء برامج لمنصة معينة عليه. على سبيل المثال ، قد يتضمن إطار العمل فئات ووظائف محددة مسبقًا يمكن استخدامها لمعالجة المدخلات وإدارة الأجهزة والتفاعل مع برامج النظام. يعمل هذا على تبسيط عملية التطوير لأن المبرمجين لا يحتاجون إلى إعادة اختراع العجلة في كل مرة يطورون فيها تطبيقًا جديدًا. يشبه إطار العمل واجهة برمجة التطبيقات (API)، على الرغم من أن إطار العمل من الناحية الفنية يتضمن واجهة برمجة تطبيقات. كما يوحي الاسم ، يعمل إطار العمل كأساس للبرمجة ، بينما توفر واجهة برمجة التطبيقات الوصول إلى العناصر التي يدعمها إطار العمل. قد يشتمل إطار العمل أيضًا على مكتبات رموز ومترجم وبرامج أخرى مستخدمة في عملية تطوير البرامج.

ومن بين إطارات العمل المستعملة لدينا:

- Express.JS
- Socket.io
- **Middleware NodeJs**

غالبًا ما تُستخدم البرامج الوسيطة في سياق إطار عمل Express.js وهي مفهوم أساسي لـ .node.js باختصار ، إنها في الأساس وظيفة لها حق الوصول إلى كائنات الطلب والاستجابة لتطبيقك. الطريقة التي أرغب في التفكير بها هي سلسلة من "الفحوصات / الشاشات المسبقة" التي يمر بها الطلب قبل أن يتعامل معه التطبيق. على سبيل المثال ، Middlewares ستكون مناسبة لتحديد ما إذا كان الطلب قد تمت مصادقته قبل أن ينتقل إلى التطبيق ويعيد صفحة تسجيل الدخول إذا لم تتم مصادقة الطلب أو لتسجيل كل طلب. يتوفر الكثير من البرامج الوسيطة التابعة لجهات خارجية والتي تتيح مجموعة متنوعة من الوظائف.

ومن بين الوسائط التي نستعملها:

- body-parser
- express-validator
- cookie-parser
- axios
- **Postman**

هو أداة تطوير البرمجيات. يعد Postman أحد أنجح الأدوات المستخدمة في فحص .API بدأ في عام 2012 كمشروع صغير لتحليل وتفسير سير عمل API في التجربة والتحسين. API تعني واجهة برمجة التطبيقات التي تمكن البرامج من التفاعل مع بعضها البعض من خلال طلبات .API يتيح Postman للمستخدمين التعاون مع شركائهم داخل مساحات عمل الفريق.

- **localhost**

المضيف المحلي هو الاسم الافتراضي الذي يصف عنوان الكمبيوتر المحلي المعروف أيضًا باسم عنوان الاسترجاع. على سبيل المثال ، كتابة: localhost ping من شأنه أن يقوم باختبار عنوان IP المحلي 127.0.0.1 (عنوان الاسترجاع). عند إعداد خادم ويب أو برنامج على خادم ويب ، يتم استخدام 127.0.0.1 لتوجيه البرنامج إلى الجهاز المحلي."

a) بناء سرفر.nodejs

الخطوات التي نقوم بها لبناء السرفر:

- ✓ نقوم بفتح البرنامج الذي سنقوم بكتابة الاكواد عليه وقد اخترنا Visual Studio Code .
- ✓ نقوم بانشاء مشروع جديد في github ثم تنزيله ثم فتحه باستخدام vs code
- ✓ نقوم بتهيئة المشروع كالتالي:

* نقوم بالادخال الامر التالي في الترمينال npm init لانشاء ملف package.json

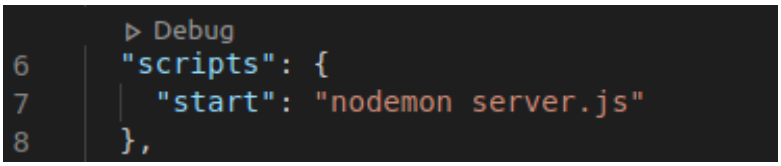
* نقوم بتثبيت حزمة nodemon لسهولة تشغيل السرفر وذلك بإدخال الامر

التالي :

```
nodemon npm install -g
```

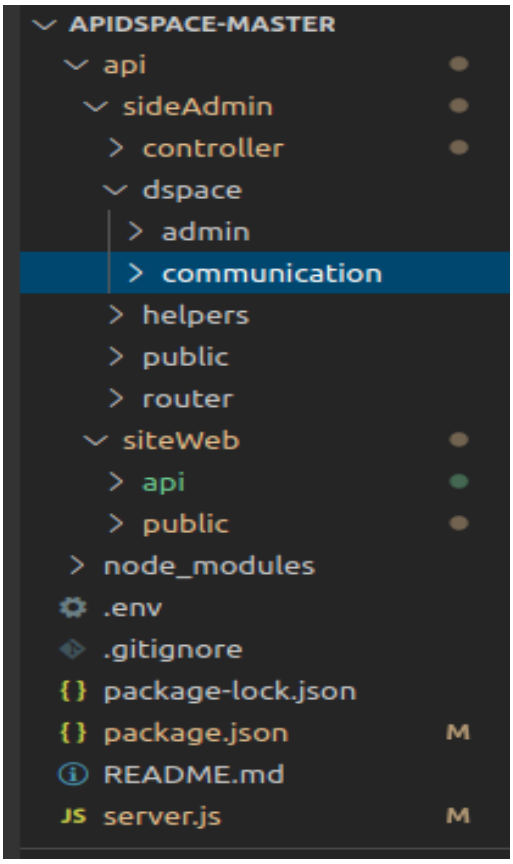
*نقوم بتعديل في ملف الذي تم إنشاؤه باستخدام الخطوة الأولى

واظافة start nodemon server.js في العنصر scripts



الشكل III-2: اضافة nodemon في العنصر scripts

- ✓ نقوم بتحديد بنية المشروع و تكون كالتالي:



الشكل III-3: بنية المشروع

- ✓ نقوم بانشاء ملف بامتداد .js وهو سيكون الملف الأساسي لتشغيل السرفر وتسميته ب app او server كتسمية مثالية.
- ✓ نقوم باستدعاء بعض framework و Middlewares التي نحتاجها لتشغيل السرفر في الملف التي يكون الملف الأساسي للسرفر server.js ويجب ان تكون قد تم تثبيتها سابقا.

مثلا

express , mongoose ,morgan ,body-parser,..etc

هناك طرق لاستدعائهم وهذه احداهم:

```
1   const express = require('express')
2   const bodyParser = require("body-parser");
3   require("dotenv").config();
4   var request = require("request");
5   var fs = require("fs");
6   var http = require('http');
```

الشكل III-4: طريقة استدعاء بعض ال framework

- ✓ نقوم بتحديد المنفذ الذي يستخدمه السرفر .
- ✓ نقوم بانشاء دالة تقوم بالاتصال بقواعد البيانات ثم نظيفها الى ملف الاساسي .
- ✓ نقوم باظافة Middlewares الخاصة بالمشروع.

```
// middlewares

app.use(bodyParser.urlencoded({ extended: false }));
app.use(bodyParser.text());// This will parse your file
app.use(bodyParser.json({limit:'50mb'}));
app.use(bodyParser.urlencoded({
  extended: true
}));
app.use(express.static(__dirname + '/api/siteWeb/public/'));
```

الشكل III-5: طريقة اضافة ال Middlewares الخاصة بالمشروع

- ✓ نقوم باظافة الملفات التي يتم الوصول اليها عند تشغيل السرفر.

```
28  // routes middleware
29  const routercommunities = require('./api/sideAdmin/router/communities')
30  const routercollections = require('./api/sideAdmin/router/collections')
31  const routerItems = require('./api/sideAdmin/router/items')
32  const routerAdmin = require('./api/sideAdmin/router/main.add.information')
33  const routerSearch = require('./api/siteWeb/api/router/searchitem')
34
35
36  //const dsSite = require('./api/siteWeb/public/index.html')
37  //routes
38  app.use("/api",routercommunities,routercollections,routerItems);
39
40  app.use("/api",routerAdmin,routerSearch);
41
```

الشكل III-6: طريقة اضافة الملفات التي تتعامل مع السيرفر

- ✓ نقوم بتشغيل السرفر باستخدام nodemon بالشكل التالي:

الشكل III-7: تشغيل السرفر

b) تحديد و شرح البنية العامة لسرفر ب.nodejs

يكون في سرفر nodejs بنية أساسية لا يمكن التخلي عليها وهي كالتالي :

- api
- controller
- router
- model
- node_modules
- server.js
- package.json
- helpers
- package-lock.json

شرح البنية العامة:

- apiهو مجلد الذي يحتوي على كل ملفات ومجلدات السرفر.
- controllerهو المجلد الذي يكون فيه دوال السرفر.
- routerهو المجلد الذي يكون فيه استدعاء الدوال الموجودة في.controller
- modelهو المجلد الذي يكون فيه ملفات الخاصة بقواعد البيانات.
- helpersهو المجلد الذي يكون فيه تعريف الأخطاء التي يمكن ان تحدث في السرفر.
- node_modulesهو المجلد الذي يكون فيه مجلدات و ملفات الحزم المثبته.
- server.jsهو الملف الأساسي للسرفر ويكون فيه كل router التي يمكن الوصول اليها في هذا السرفر.
- package.jsonيحتوي على معلومات عن المشروع مثل اسم المطور، اسم المشروع ، مسار ملف التشغيل الأساسي ، الحزم المثبته واصداراتها و ...الخ.
- package-lock.jsonيصف الشجرة الدقيقة التي تم إنشاؤها ، بحيث تكون عمليات التثبيت اللاحقة قادرة على إنشاء أشجار متطابقة ، بغض النظر عن تحديثات التبعية الوسيطة.هذا الملف مخصص للالتزام به في مستودعات المصدر ، ويخدم أغراضًا مختلفة:

وصف تمثيل واحد لشجرة التبعية بحيث يضمن أعضاء الفريق وعمليات النشر والتكامل المستمر تثبيت نفس التبعيات تمامًا.

توفير وسيلة للمستخدمين "للسفر عبر الزمن" إلى الحالات السابقة للوحدات node_modules دون الحاجة إلى الالتزام بالدليل نفسه.

لتسهيل رؤية أكبر للتغييرات الشجرية من خلال اختلاف التحكم في المصدر المقروء.من اجل تحسين عملية التثبيت من خلال السماح لـ npm بتخطي دقة البيانات الوصفية المتكررة للحزم المثبتة مسبقًا... الخ.

c) شرح Postman

- ✓ شرح مبدا عملPostman
- ✓ ارسال طلبات الى السرفر
- ✓ الرد من السرفر
- ✓ **شرح مبدا عمل:Postman**

هو عبارة عن client بالنسبة للسرفر و يتم استخدامه لاختبار الدوال الموجود في api قبل ربطها بأحد الأجهزة الأخرى. يمكن للمرء إرسال طلبات للاتصال بـ APIs والانضمام إليها. يمكن لهذه المكالمات استرداد المعلومات وإضافتها وحذفها وتحديثها. هذه الأداة مفيدة لتطوير أو تحليل أو فحص API ، أو الاختلاط مع واجهة برمجة تطبيقات خارجية لإرسال الطلبات .

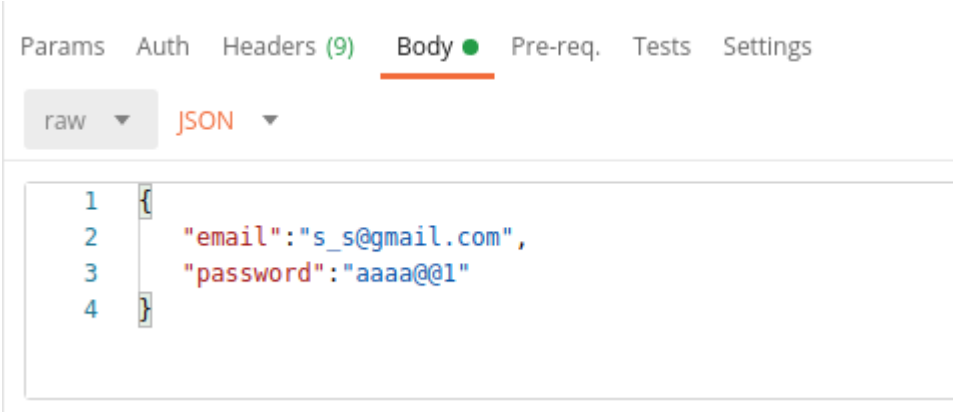
مثال:

- ✓ **ارسال طلبات الى السرفر:**

يمكن ارسال طلب الى api بعدة صيغ منها:

* ارسال.body
* ارسال.headers
* ارسال.params

مثال:



الشكل III-8: ارسال طلب الي السرفر

هناك عدة صيغ لارسال لبيانات الى السرفر منها:

* row
* from-data
* binary

✓ **الرد من السرفر:**

يكون الرد بعدة صيغ حسب ما تم تحديده في السرفر

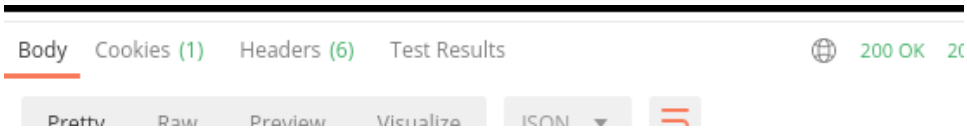
وعادة يكون الرد بالحالات التالية:

a. الحالة 200 : يشير رمز استجابة حالة نجاح HTTP 200 OK إلى أن الطلب قد نجح. تكون استجابة 200 قابلة للتخزين المؤقت افتراضيًا.يعتمد معنى النجاح على طريقة طلب:HTTP

- GET تم جلب المورد وإرساله في نص الرسالة.
- HEADرؤوس الكيان موجودة في نص الرسالة.
- POSTيتم إرسال المورد الذي يصف نتيجة الإجراء في نص الرسالة.
- TRACEيحتوي نص الرسالة على رسالة الطلب كما استلمها الخادم.

غالبًا ما تكون النتيجة الناجحة لـ PUT أو DELETE ليست 200 OK ولكن لا توجد محتويات 204 (أو 201 تم إنشاؤها عند تحميل المورد لأول مرة.(

مثال:

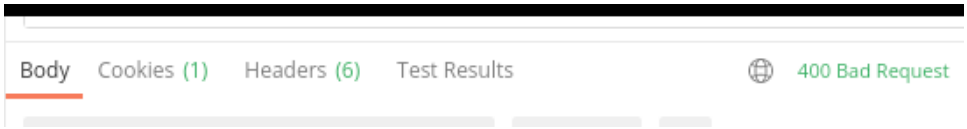


الشكل III-9: الحالة 200

b. الحالة 400:

يشير رمز حالة استجابة طلب غير صالح لبروتوكول نقل النص التشعبي 400 (HTTP) إلى أن الخادم لا يمكنه أو لن يقوم بمعالجة الطلب بسبب شيء يُنظر إليه على أنه خطأ عميل (على سبيل المثال ، بناء جملة طلب غير صحيح أو تأطير رسالة طلب غير صالح أو طلب مخادع التوجيه).

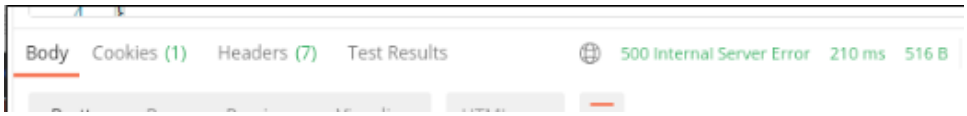
مثال:



الشكل III-10: الحالة 400

c. الحالة 500: يشير رمز الاستجابة لخطأ خادم خطأ داخلي في الخادم HyperText Transfer Protocol (HTTP) 500إلى أن الخادم واجه حالة غير متوقعة منعته من تلبية الطلب.استجابة الخطأ هذه هي استجابة "شاملة" عامة. عادة ، يشير هذا إلى أن الخادم لا يمكنه العثور على رمز خطأ 5 xx أفضل للاستجابة. في بعض الأحيان ، يسجل مسؤولو الخادم استجابات خطأ مثل رمز الحالة 500 مع مزيد من التفاصيل حول الطلب لمنع حدوث الخطأ مرة أخرى في المستقبل .

مثال:

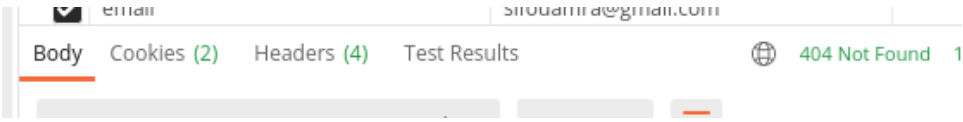


الشكل III-11: الحالة 500

d. الحالة 404: يشير رمز استجابة خطأ العميل HTTP 404 Not Found إلى أن الخادم لا يمكنه العثور على المورد المطلوب. غالبًا ما تسمى الروابط التي تؤدي إلى صفحة 404 بالروابط المعطلة أو الميتة ويمكن أن تخضع لتعفن الارتباط.

لا يشير رمز الحالة 404 إلى ما إذا كان المورد مفقودًا بشكل مؤقت أو دائم. ولكن إذا تمت إزالة مورد نهائيًا.

مثال:



الشكل III-12: الحالة 404

e. الحالة 401: يشير رمز استجابة حالة خطأ العميل غير المصرح به HTTP 401 إلى أن الطلب لم يتم تطبيقه لأنه يفتقر إلى بيانات اعتماد المصادقة الصالحة للمورد الهدف.

يتم إرسال هذه الحالة مع رأس مصادقة يحتوي على معلومات حول كيفية التفويض بشكل صحيح.هذه الحالة مشابهة لـ 403 ، لكن في هذه الحالة ، المصادقة ممكنة.

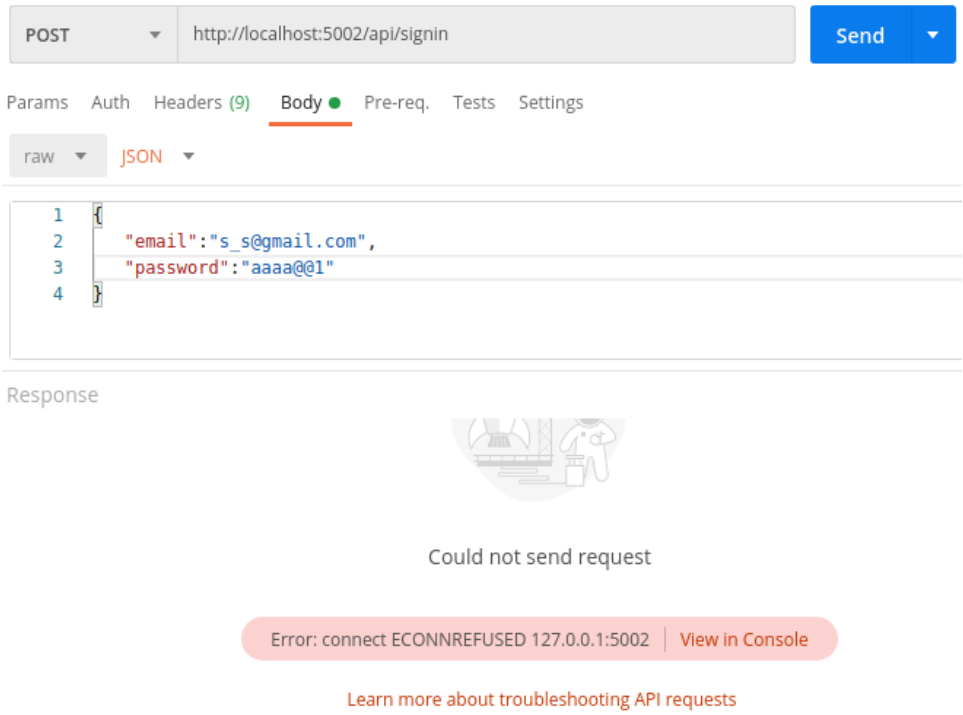
مثال:

الشكل III-13: الحالة 500

f. الحالة 403: يشير رمز استجابة حالة خطأ العميل المحظور HTTP 403 إلى أن الخادم فهم الطلب ولكنه يرفض الإذن به.هذه الحالة مشابهة للحالة 401 ، ولكن في هذه الحالة ، لن تحدث إعادة المصادقة أي فرق. الوصول ممنوع بشكل دائم ومرتبط بمنطق التطبيق ، مثل الحقوق غير الكافية لمورد.

ملاحظة: هناك حالة شاذة عن هذه الحالات و تكون عندما يكون السرفر غير مشغل.

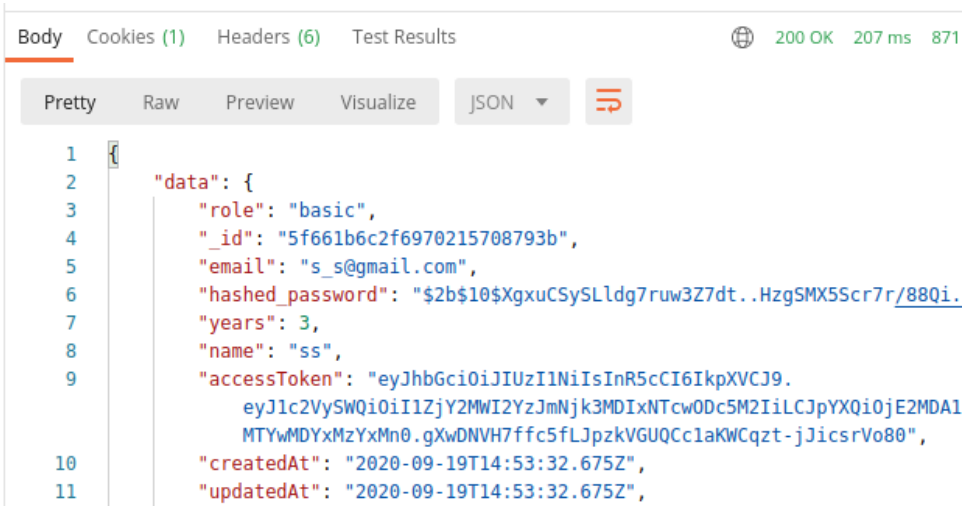
مثال:



الشكل III-14: الحالة 403

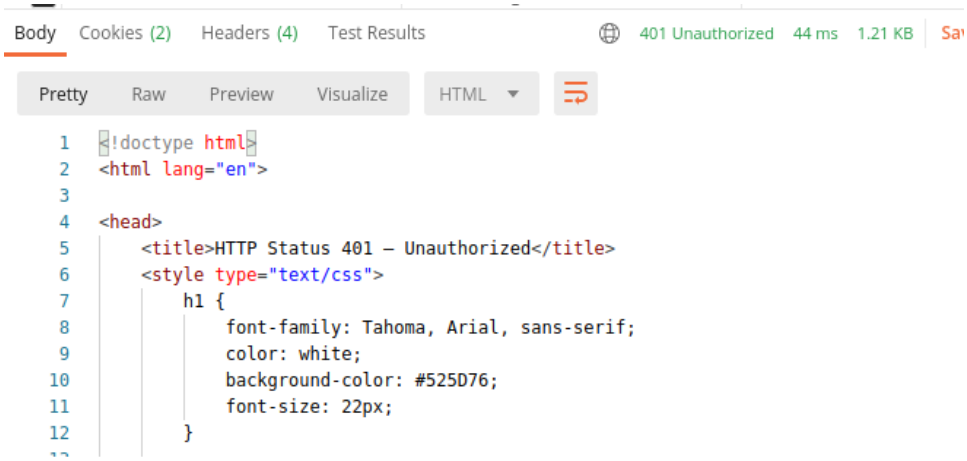
ملاحظة: يكون بالاظافة الى الحالة الرد الخاص بالسرفر اذا كان الامر ناجحا او هناك خطأ.

مثال عندما تكون الحالة نجاح:



الشكل III-15: الرد من السرفر في الحالة 200

مثال عندما تكون أحد حالات الخطا:



الشكل III-16: الرد من السرفر في احد الحالات الخطاء

d) شرحlocalhost

1. شرح مبدا عمله.
2. امثلة عن استخداماته.

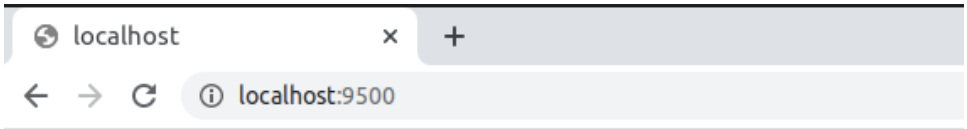
1. شرح مبدا عمله:

سوف يعتمد العنوان المطلوب على بروتوكول الإنترنت المستخدم. استدعاء المضيف المحلي باستخدام IPv4 ، وهو IPv4 الأكثر شيوعًا ، سيكون له نطاق من 127.0.0.0 إلى 127.255.255.255 ولكنه سيكون عادةً افتراضيًا إلى 127.0.0.1. ويستعرضه بناءا على رقم المنفذ لاستعراض ما تم استظافته.

2. امثلة عن استخداماته:

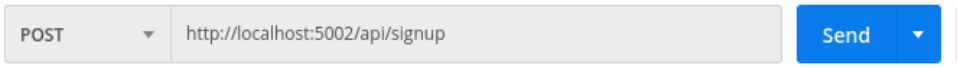
يتم عرض localhost ذلك بإدخالhttp://localhost:"port"/path

في المتصفح:



الشكل III-17: ال localhost في المتصفح

او في Postman او في احد برامج الاختبار الأخرى:



الشكل III-18: ال localhost في ال postman

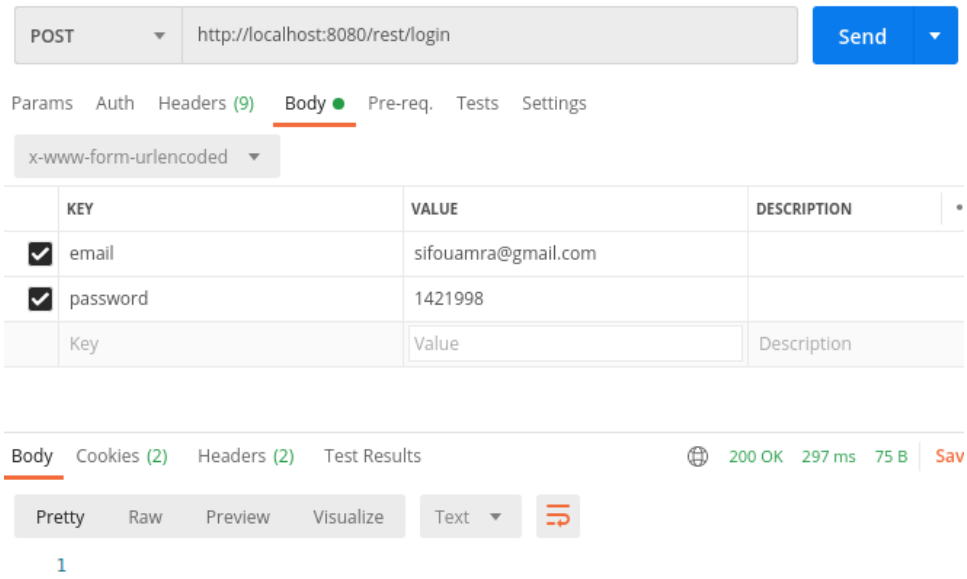
3- شرح واجهات البرامج.

في هذا الجزء سوف نقوم بانهاء مشروعنا به وذلك بشرح الواجهات الاساسية للتطبيقات التي تم برمجتها.

3-1-شرح api نظام الدي سبيس.(DSpace)

✓ تسجيل الدخول:

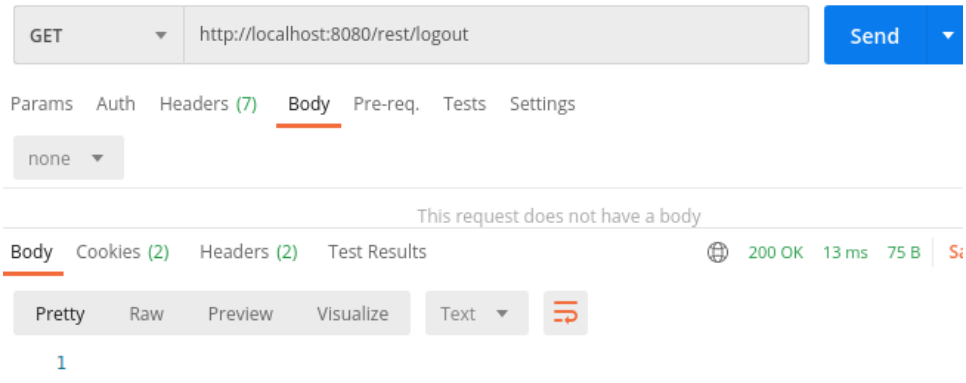
تمثل هذه الواجهة كيفية تسجيل الدخول الى الدي سبيس.



الشكل III-19:الواجهة الاساسية

✓ تسجيل الخروج:

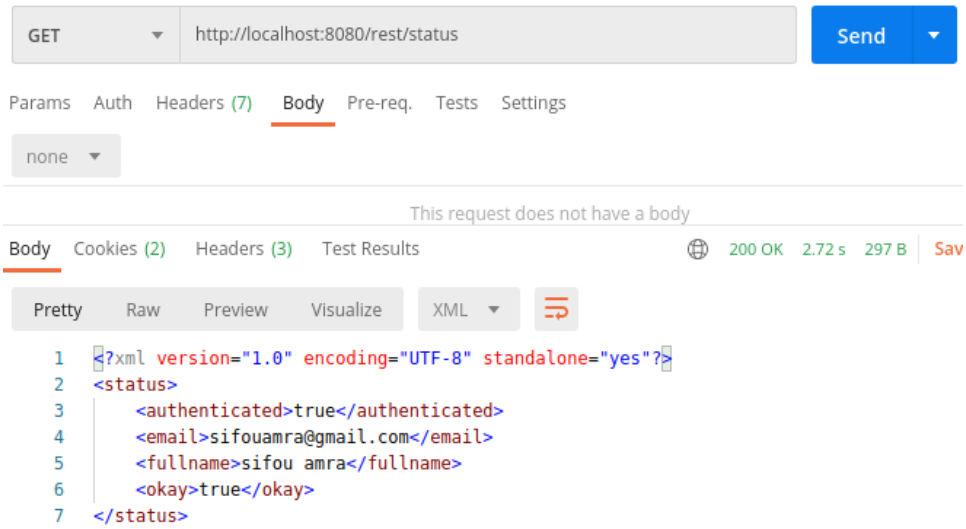
تمثل هذه الواجهة كيفية تسجيل الخروج من الدي سبيس.



الشكل III-20: واجهة تسجيل الخروج

✓ حالة الجلسة:

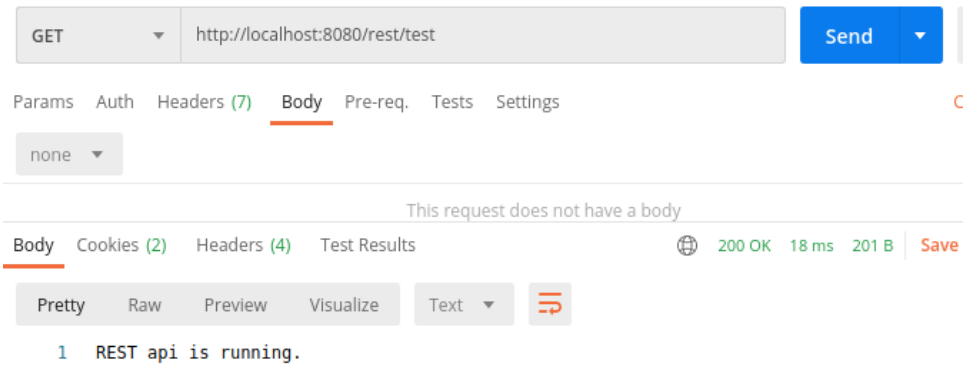
تمثل هذه الواجهة كيفية الاطلاع على حالة جلسة الدخول الى الدي سبيس.



الشكل III-21: واجهة حالة الجلسة

✓ التحقق من تشغيل:DSpace

تمثل هذه الواجهة كيفية التحقق من تشغيل DSpace و ذلك إعادة السلسلة "REST api قيد التشغيل" لأغراض الاختبار

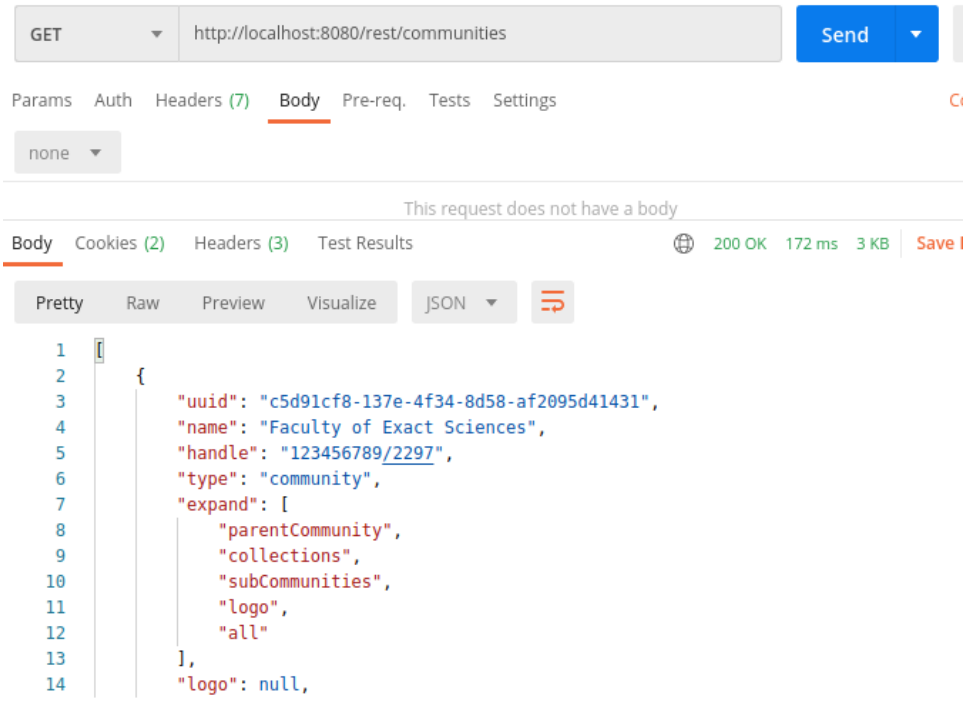


الشكل III-22: واجهة التحقق من تشغيلDSpace

3 -2- المجتمعات الكبرى والصغرى (المكتبة العامة و المكتبة الخاصة):

✓ الحصول على مصفوفة المجتمعات الكبيرة والصغيرة:

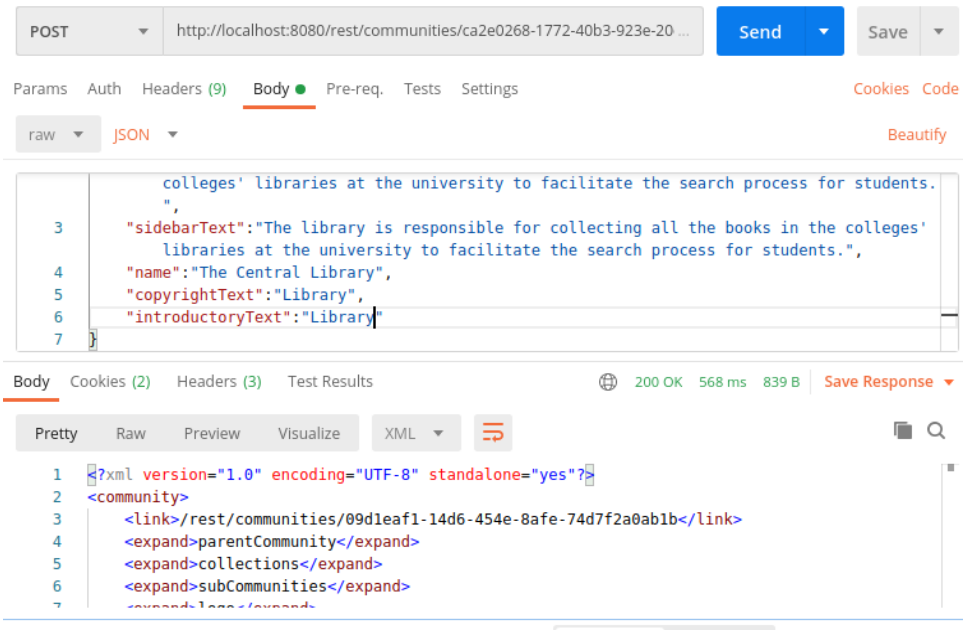
تقوم بإرجاع مصفوفة تحتوي كافة المجتمعات في الدي سبيس.



الشكل III-23: واجهة الحصول على مصفوفة المجتمعات الكبيرة والصغيرة

✓ اضافة مجتمع:

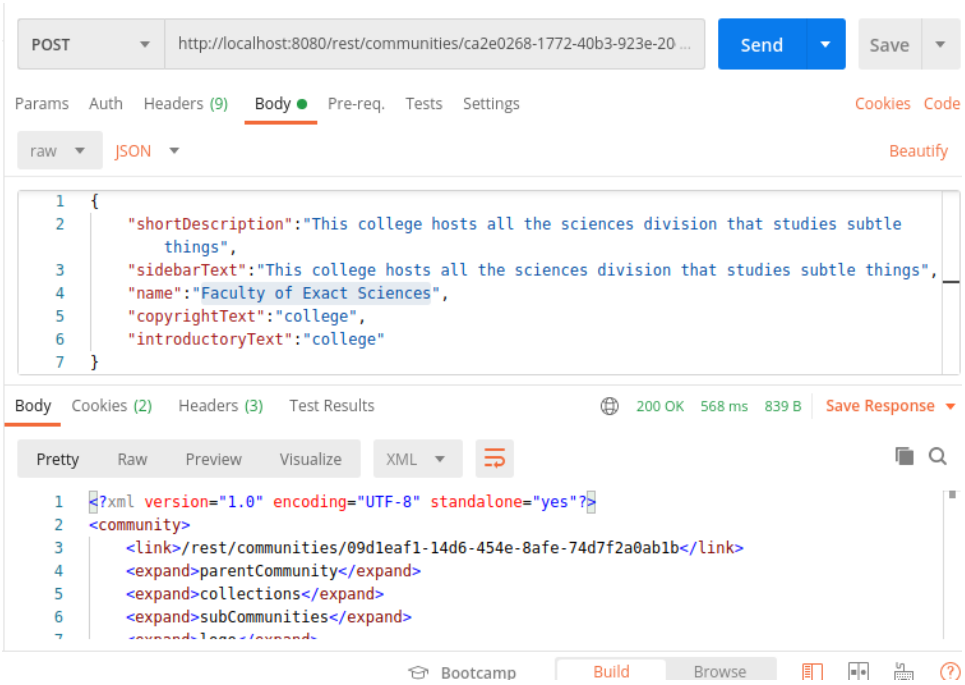
تقوم باضافة مجتمع او مكتبة العامة في الدي سبيس.



الشكل III-24: واجهة اضافة مجتمع كبيرة

✓ اضافة مجتمع صغير الى مجتمع كبير:

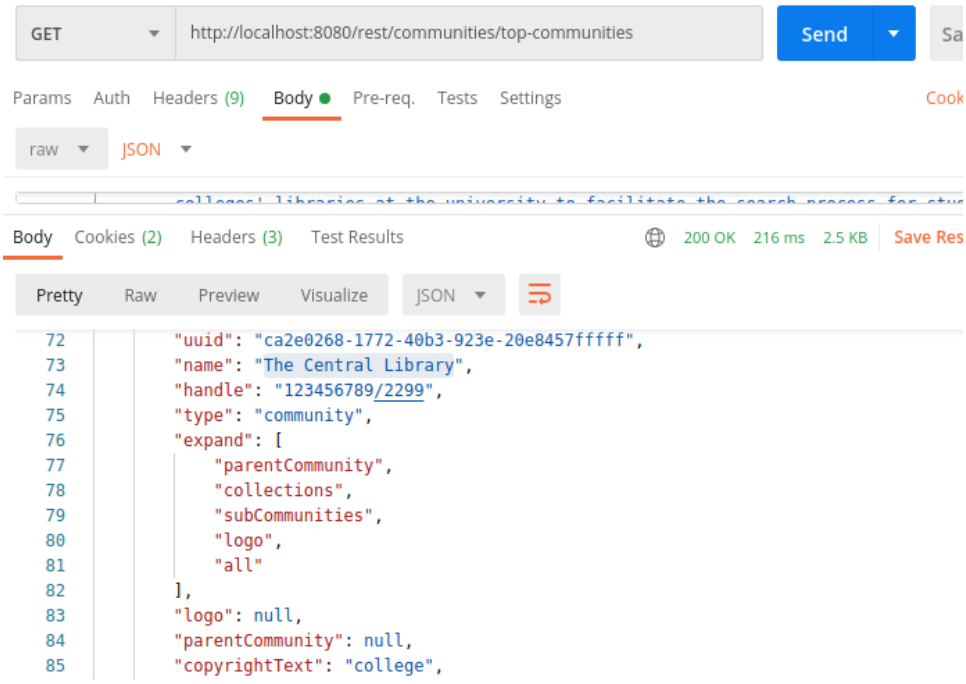
تقوم باضافة المجتمع الصغير في الدي سبيس.



الشكل III-25: واجهة اضافة مجتمع صغير الى مجتمع كبير

✓ الحصول على مصفوفة المجتمعات العليا:

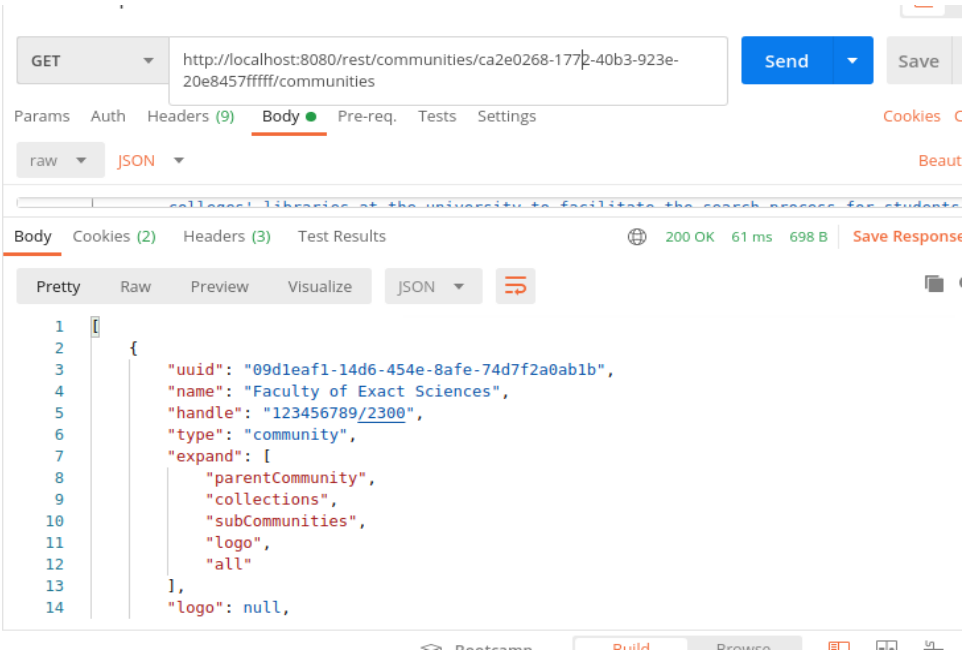
تقوم بإرجاع مصفوفة تحتوي كافة المجتمعات العليا فقط في الدي سبيس.



الشكل III-26: واجهة الحصول على مصفوفة المجتمعات العليا

✓ الحصول على مصفوفة المجتمعات الصغرى:

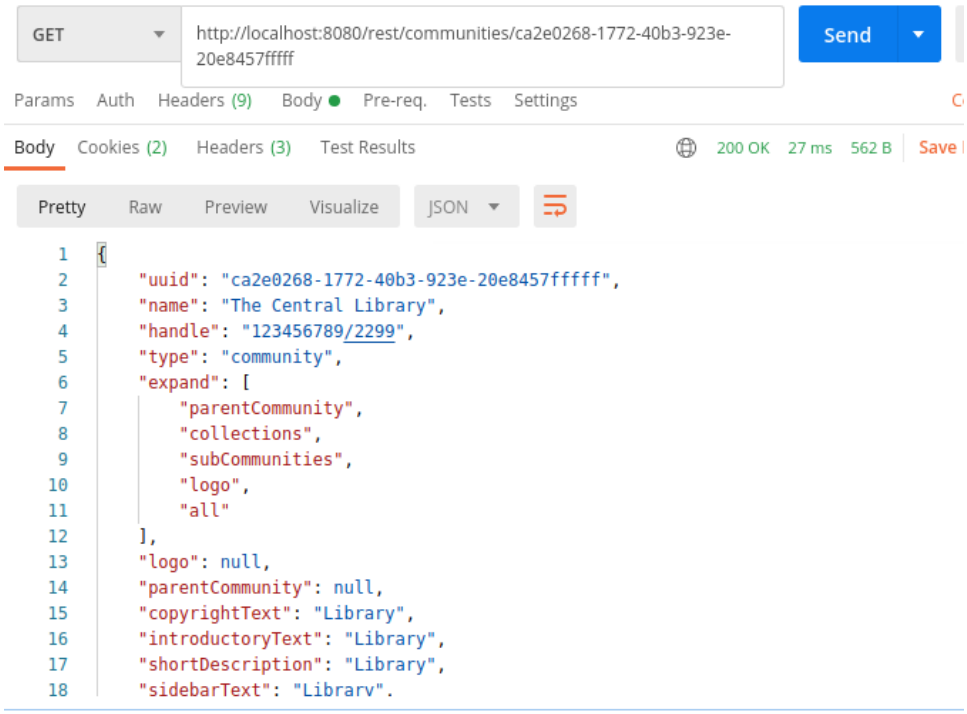
تقوم بإرجاع مصفوفة تحتوي كافة المجتمعات الصغرى فقط في الدي سبيس.



الشكل III-34: واجهة الحصول على مصفوفة المجتمعات العليا

✓ الحصول على مجتمع كبير واحد:

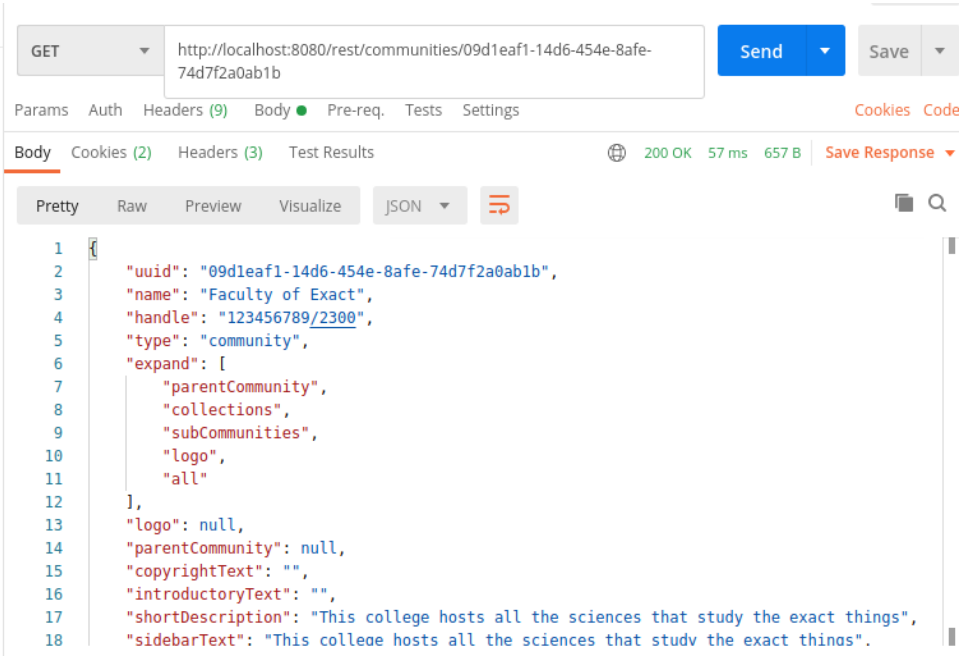
تقوم بإرجاع عنصر وحيد يحتوي على مجتمع كبير واحد باسعمال id الخاص به في الدي سبيس.



الشكل III–27: واجهة الحصول على مجتمع كبير واحد

✓ الحصول على مجتمع صغير واحد:

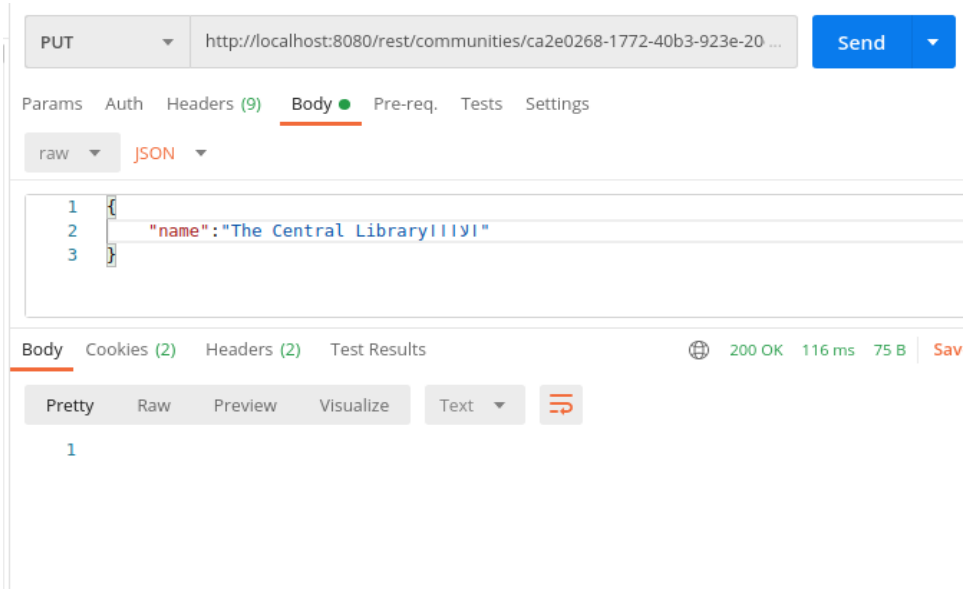
تقوم بإرجاع عنصر وحيد يحتوي على مجتمع صغير واحد باسعمال id الخاص به في الدي سبيس.



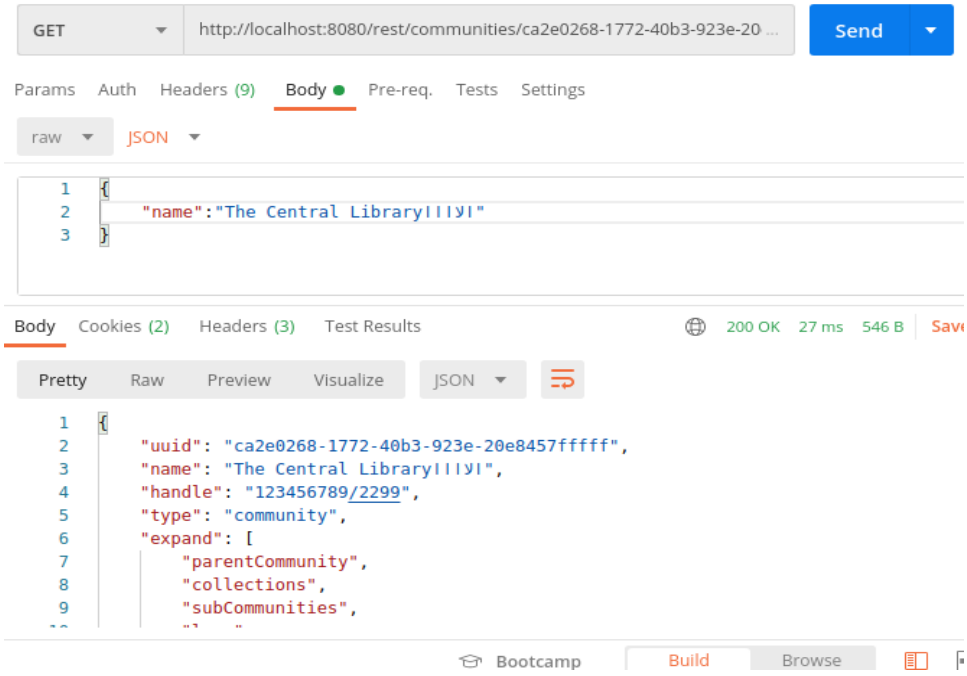
الشكل III-28: واجهة الحصول على مجتمع صغير

✓ تحديث على المجتمع الكبير:

تقوم بتحديث المعلومات الموجودة في هذا المجتمع سوا معلومات تم إدخالها سابقا او تكون جديدة المهم ان يكون المجتمع موجودا بالفعل وذلك باستخدام id الخاص به في الدي سبيس.



الشكل III-29: واجهة تحديث على المجتمع الكبير



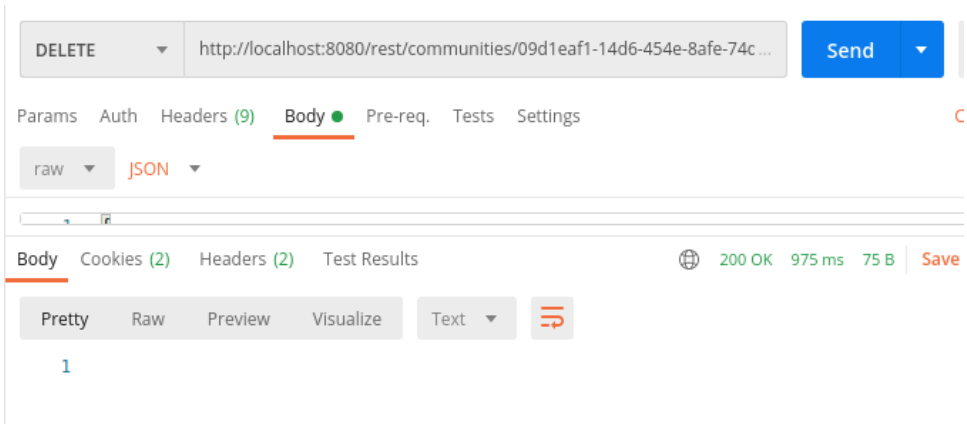
الشكل III-30: واجهة الحصول على المجتمع الكبير بعد تحديثه

✓ تحديث على المجتمع الصغير:

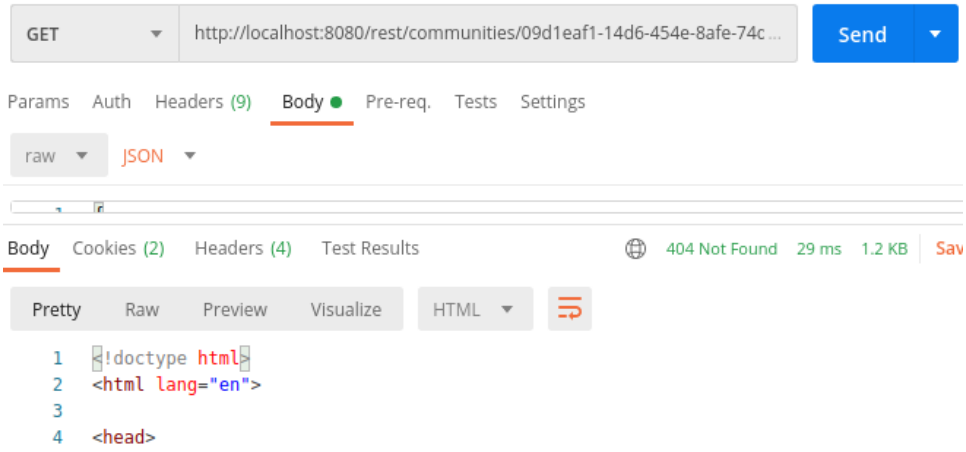
تقوم بتحديث المعلومات الموجودة في هذا المجتمع سوا معلومات تم إدخالها سابقا او تكون جديدة المهم ان يكون المجتمع موجودا بالفعل وذلك باستخدام id الخاص به في الدي سبيس. نفس شكلي المجتمع الكبير.

✓ حذف مجتمع الكبير:

تقوم بحذف المجتمع من الدي سبيس باستعمال id الخاص به.



الشكل III-31: واجهة حذف مجتمع الكبير



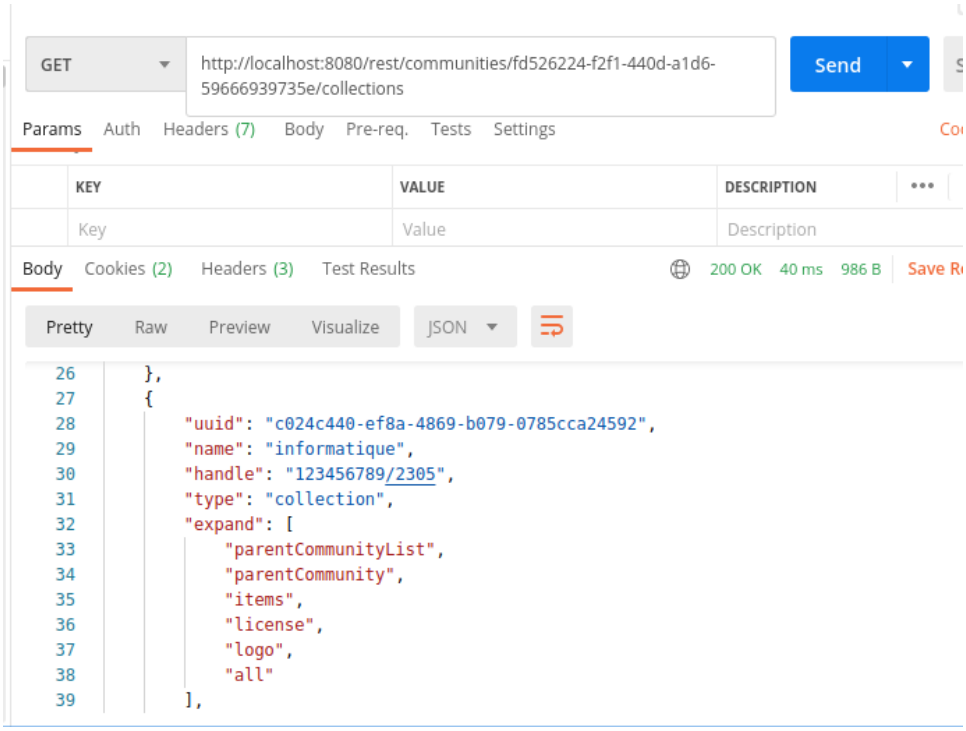
الشكل III-32: واجهة الحصول على المجتمع الكبير بعد حذفه

✓ حذف مجتمع الصغير:

تقوم بحذف المجتمع من الدي سبيس باستعمال id الخاص به نفس شكلي المجتمع الكبير.

✓ الحصول على مجموعات مجتمع صغير واحد:

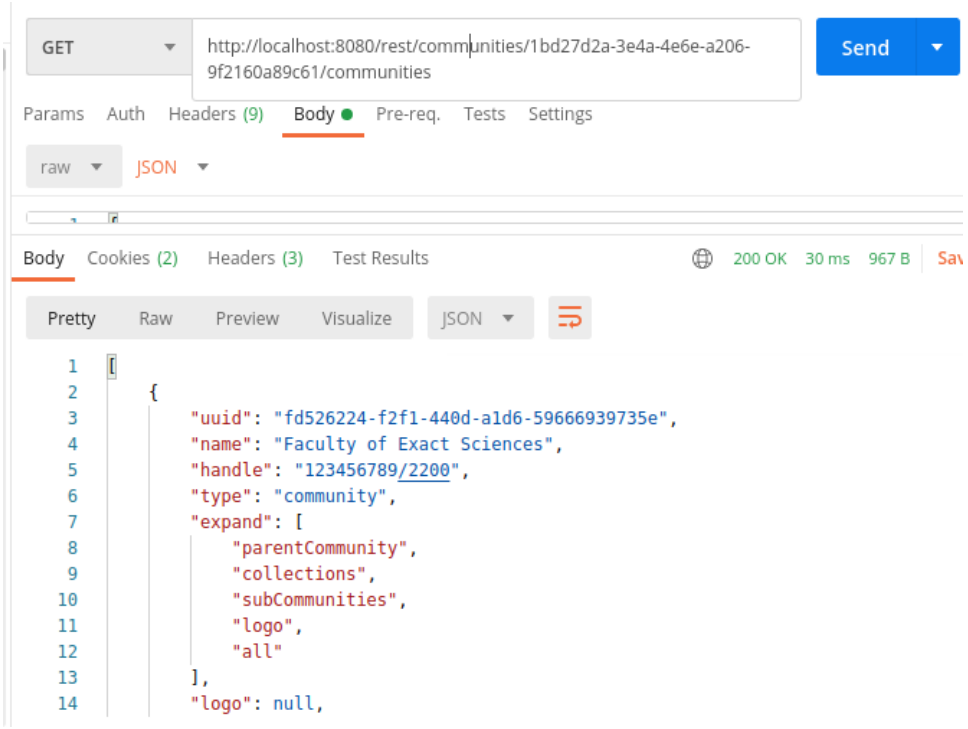
تقوم بإرجاع مصفوفة تحتوي على مجموعات المجتمع صغيرالواحد باسعمال id الخاص به في الدي سبيس.



الشكل III-33: واجهة الحصول على مجموعات مجتمع صغير واحد

✓ الحصول على المجتمعات الصغيرة لمجتمع كبير واحد:

تقوم بإرجاع مصفوفة تحتوي على المجتمعات الصغيرة لمجتمع كبير واحد باسعمال id الخاص به في الدي سبيس.

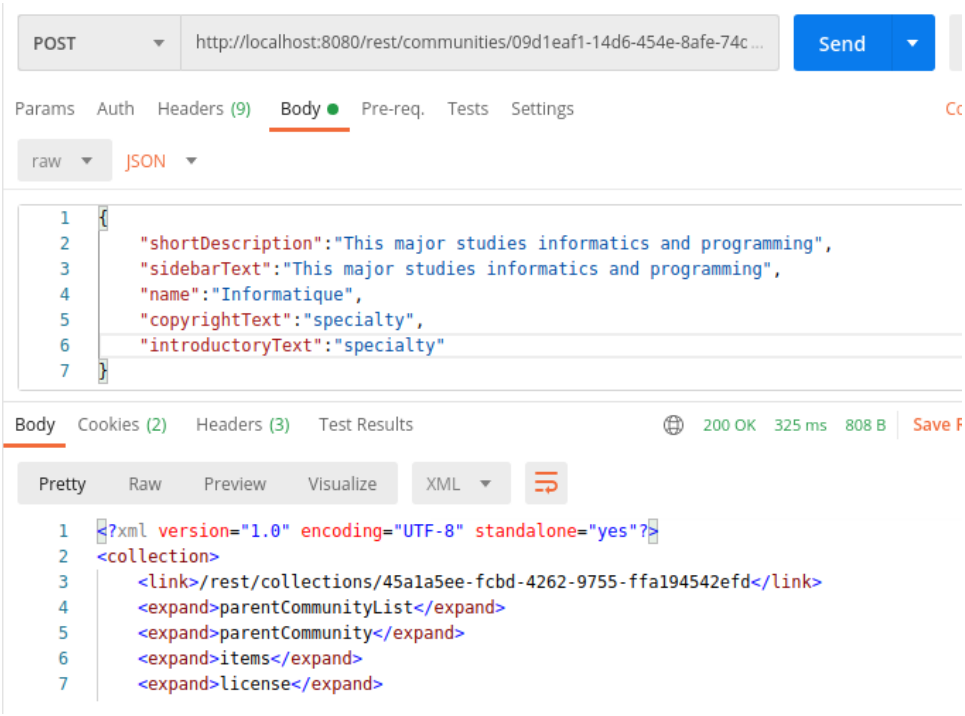


الشكل III-34: واجهة الحصول على المجتمعات الصغيرة لمجتمع كبير واحد

3 -3-المجموعات (الكليات)

✓ إضافة مجموعة:

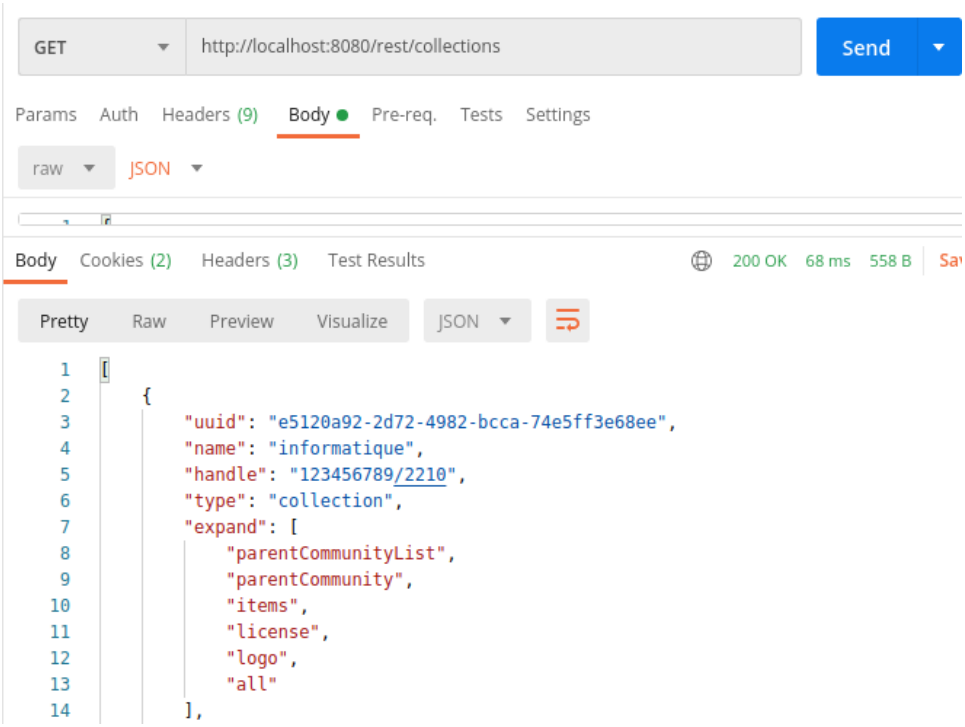
تقوم بإضافة مجموعة في الدي سبيس.



الشكل III-35: واجهة إضافة مجموعة

✓ الحصول على مصفوفة المجموعات:

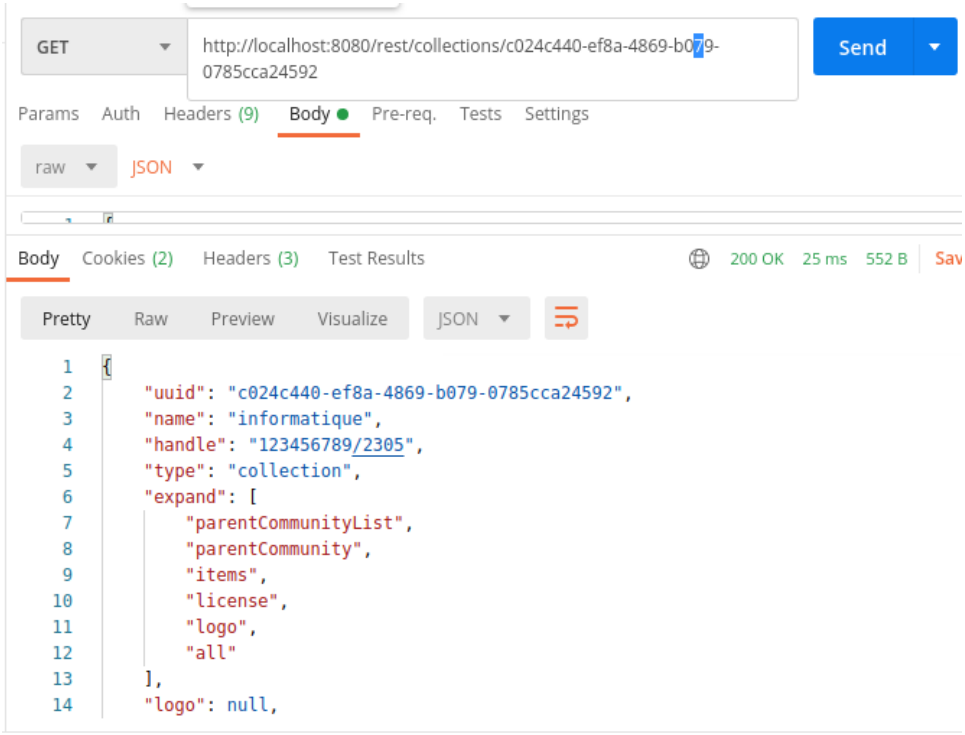
تقوم بإرجاع مصفوفة تحتوي كافة المجموعات في الدي سبيس.



الشكل III-36: واجهة الحصول على مصفوفة المجموعات

✓ الحصول على مجموعة واحدة:

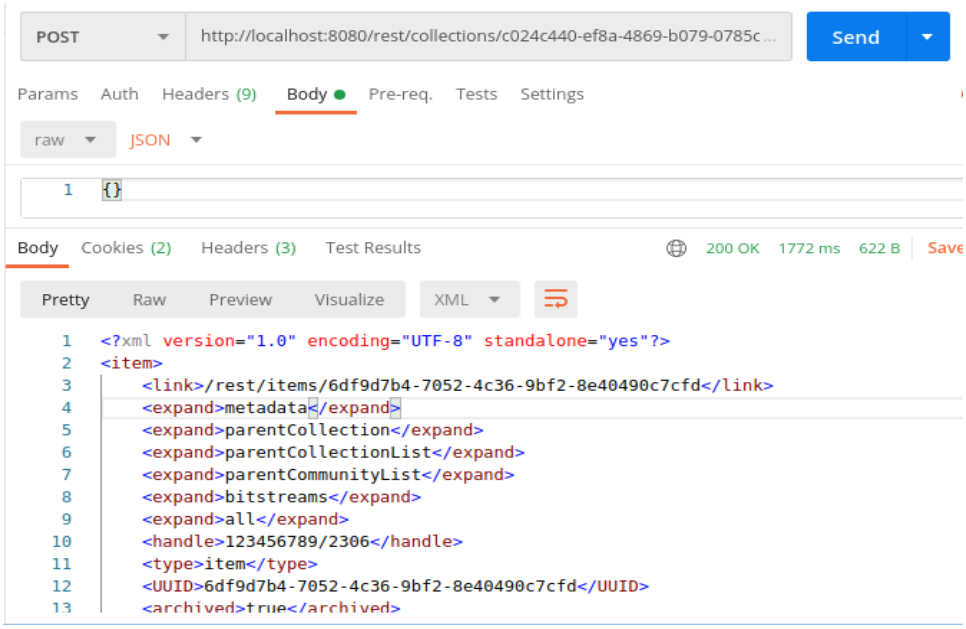
تقوم بإرجاع مجموعة واحدة باستخدام id الخاص بها في الدي سبيس.



الشكل III-37: واجهة الحصول على مصفوفة المجموعات

✓ اضافة عنصر في مجموعة :

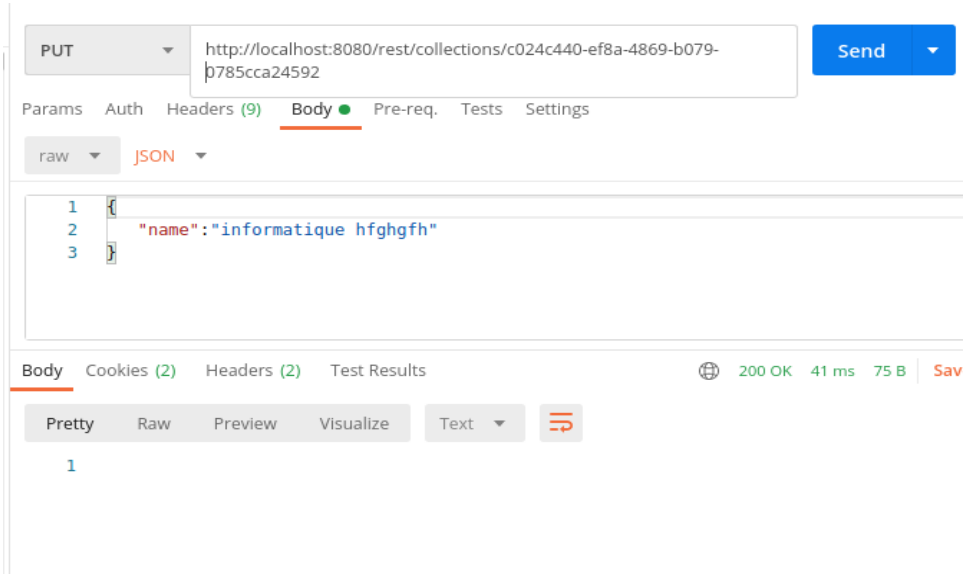
تقوم بإضافة عنصر في مجموعة باستخدام id الخاص بها في الدي سبيس.



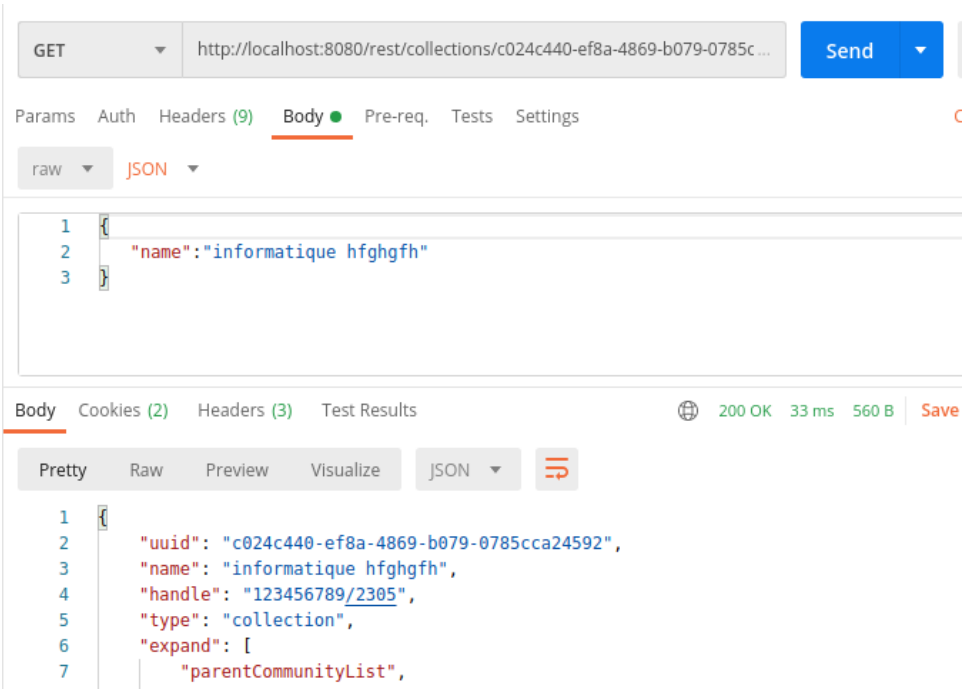
الشكل III-38: واجهة اضافة عنصر في مجموعة

✓ تحديث مجموعة :

تقوم بتحديث مجموعة باستخدام id الخاص بها في الدي سبيس.



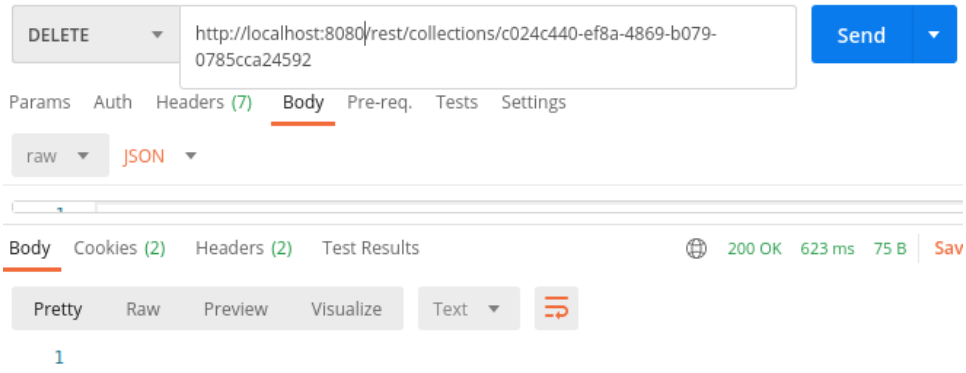
الشكل III-39: واجهة تحديث مجموعة



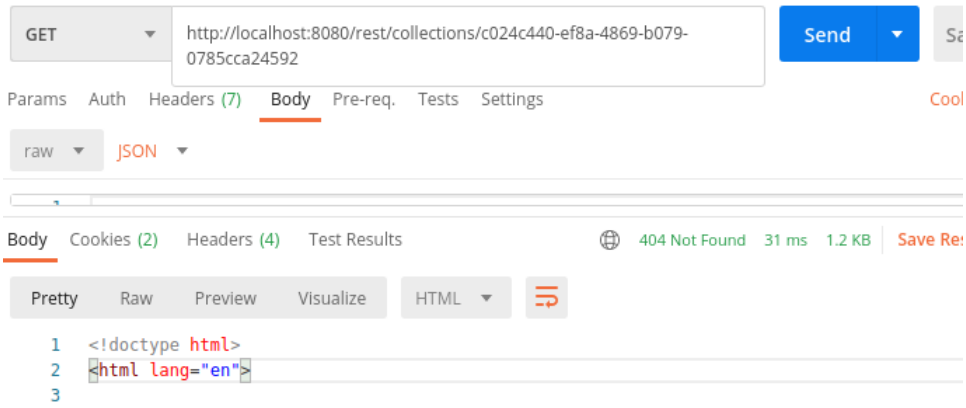
الشكل III-40: واجهة الحصول على مجموعة بعد التحديث

✓ حذف مجموعة :

تقوم بحذف مجموعة باستخدام id الخاص بها في الدي سبيس.



الشكل III-41: واجهة حذف مجموعة

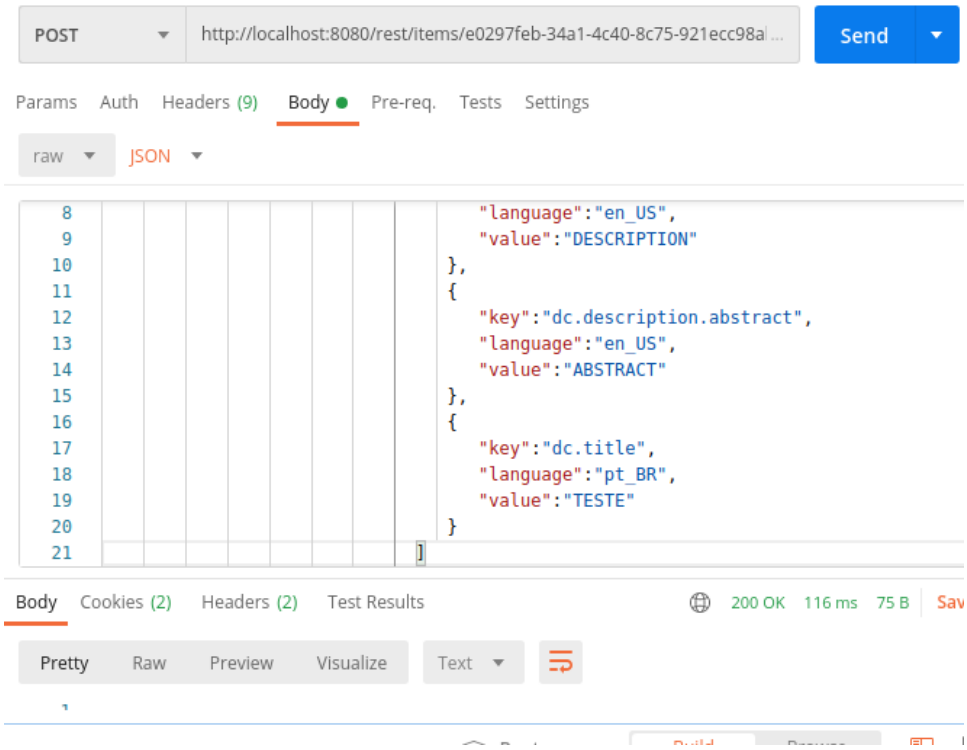


الشكل III-50: واجهة الحصول على مجموعة بعد الحذف

3-4-العناصر (محافظ الكتب):

✓ إضافة عنصر:

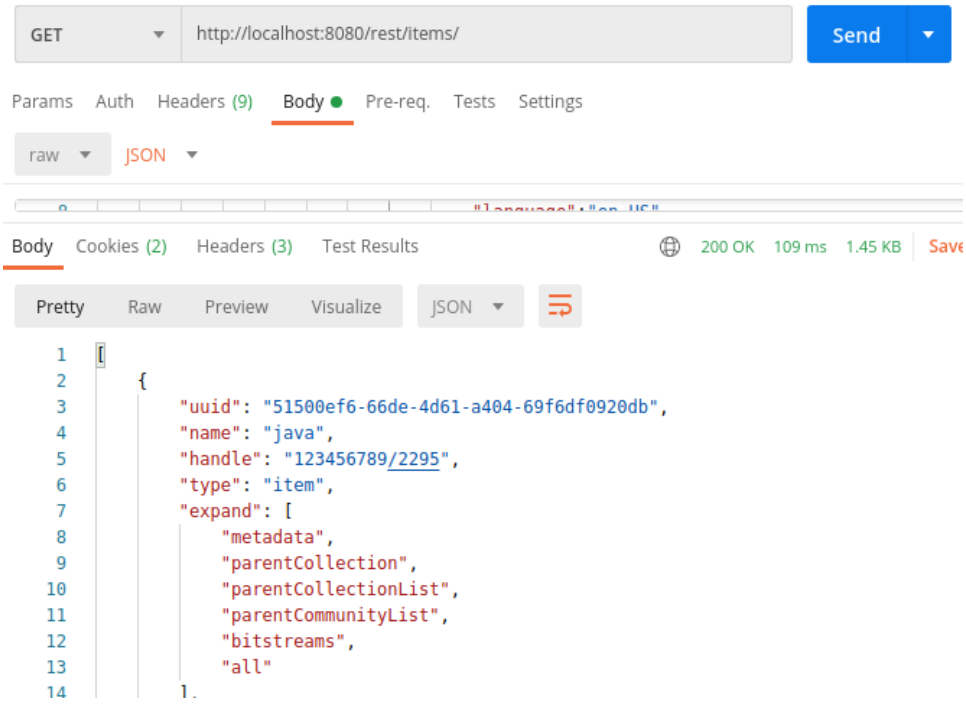
نقوم باظافة بيانات وصفية له وهو موجود بالفعل عن طريق id الخاص به في الدي سبيس.



الشكل III-42: واجهة إضافة عنصر

✓ الحصول على العناصر:

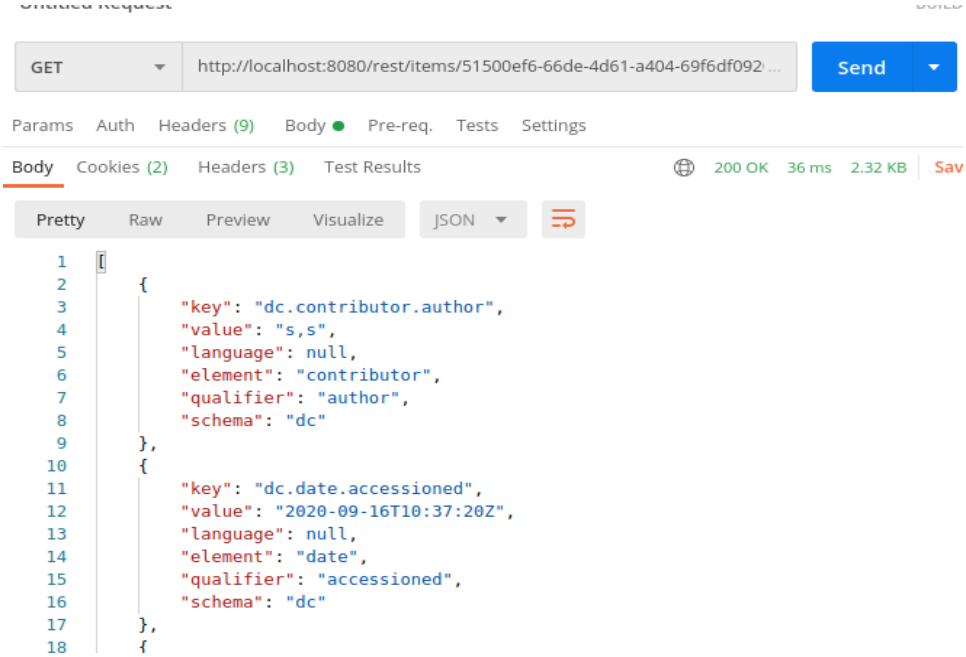
نقوم بالحصول على العناصر في الدي سبيس.



الشكل III-43: واجهة الحصول على العناصر

✓ الحصول على البيانات الوصفية للعنصر:

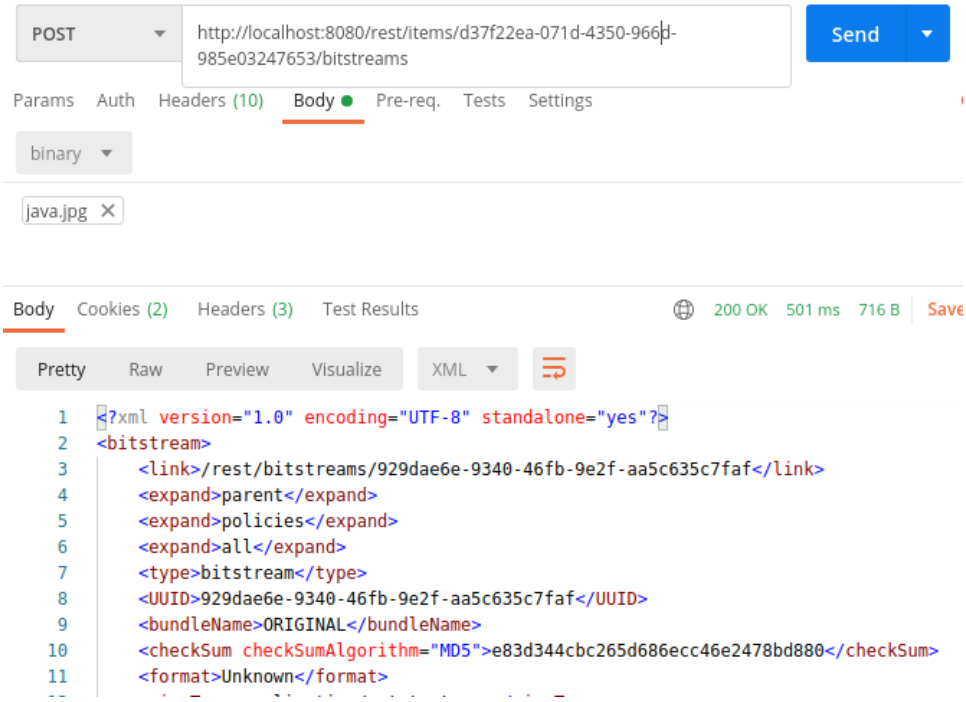
نقوم بالحصول على البيانات الوصفية لعنصر ما باستخدام id الخاص به في الدي سبيس.



الشكل III-44: واجهة الحصول على البيانات الوصفية للعنصر

✓ إضافة ملف للعنصر:

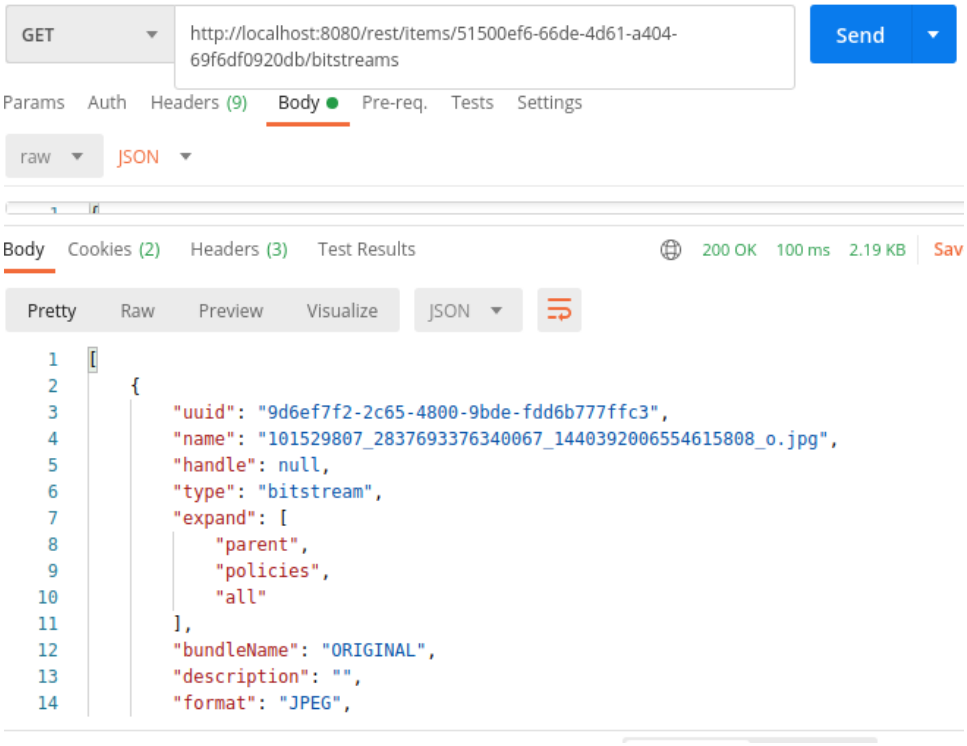
نقوم بإضافة ملف العنصر المحدد باستخدام id الخاص به في الدي سبيس.



الشكل III-45: واجهة إضافة ملف للعنصر

✓ الحصول على بيانات ملفات العنصر الواحد:

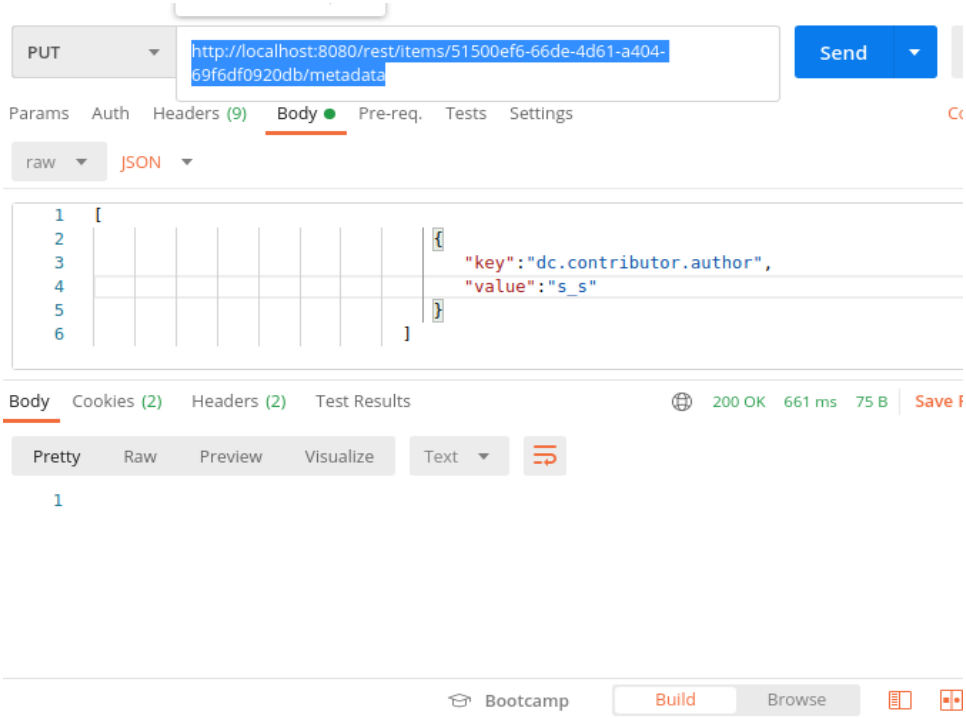
نقوم بالحصول على بيانات ملفات العنصر المحدد باستخدام id الخاص به في الدي سبيس.



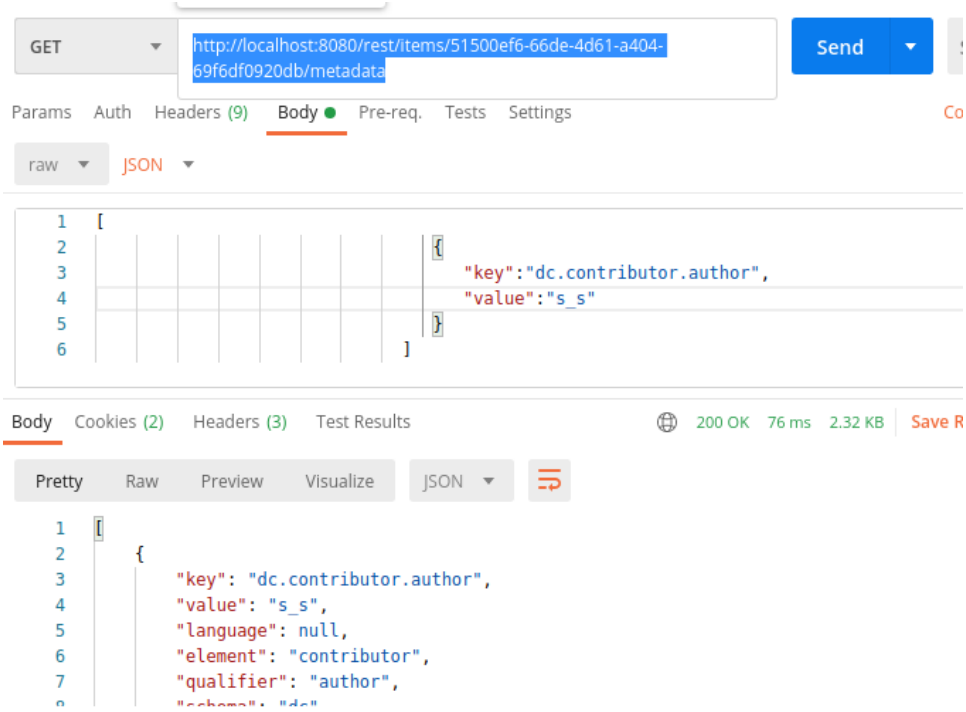
الشكل III-46: واجهة الحصول على بيانات ملفات العنصر الواحد

✓ تحديث البيانات الوصفية العنصر:

نقوم بتحديث البيانات الوصفية للعنصر باستخدام id الخاص به في الدي سبيس.



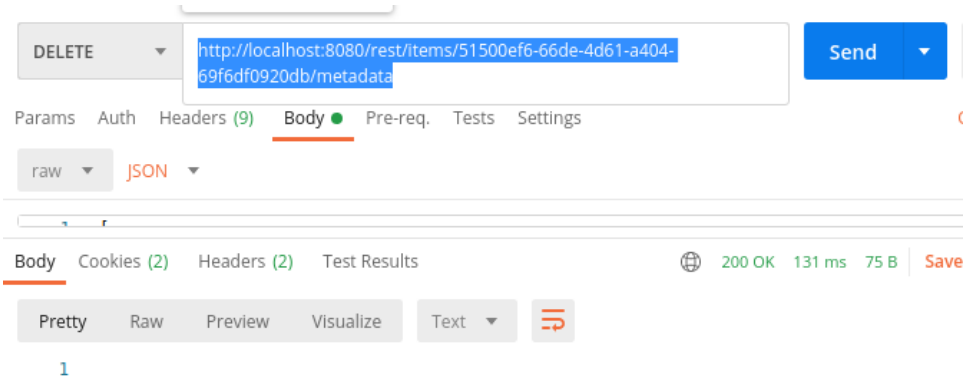
الشكل III-47: واجهة الحصول على بيانات ملفات العنصر



الشكل III-48: واجهة الحصول على بيانات ملفات العنصر بعد التحديث

✓ حذف البيانات الوصفية للعنصر:

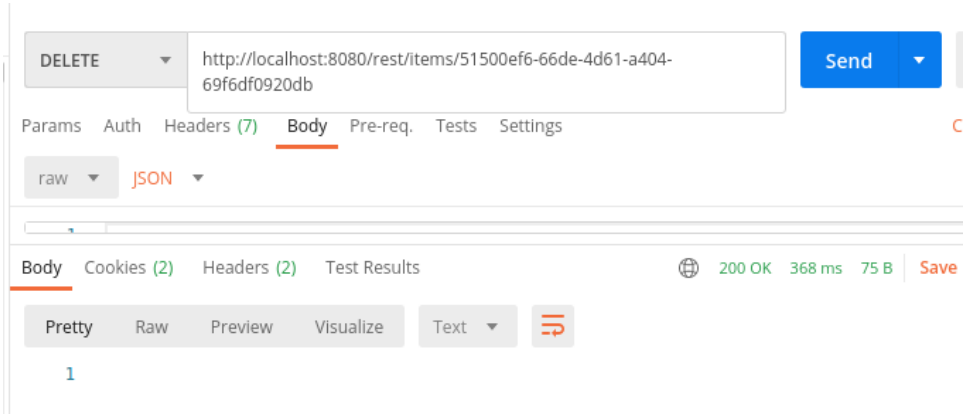
نقوم بحذف البيانات الوصفية للعنصر باستخدام id الخاص به في الدي سبيس.



الشكل III-49: واجهة حذف البيانات الوصفية للعنصر

✓ حذف العنصر:

نقوم بحذف العنصر باستخدام id الخاص به في الدي سبيس.

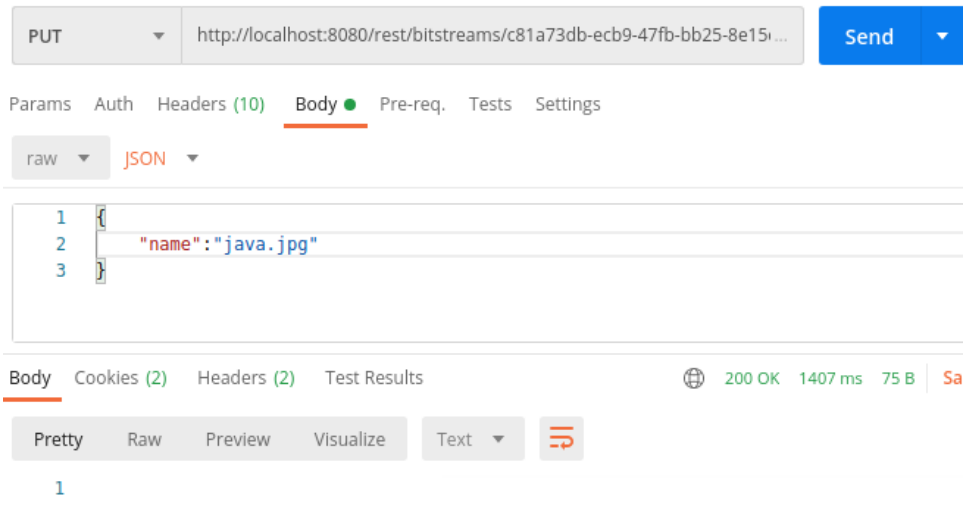


الشكل III-50: واجهة حذف العنصر

ملفات العناصر:

✓ إضافة اسم للملف:

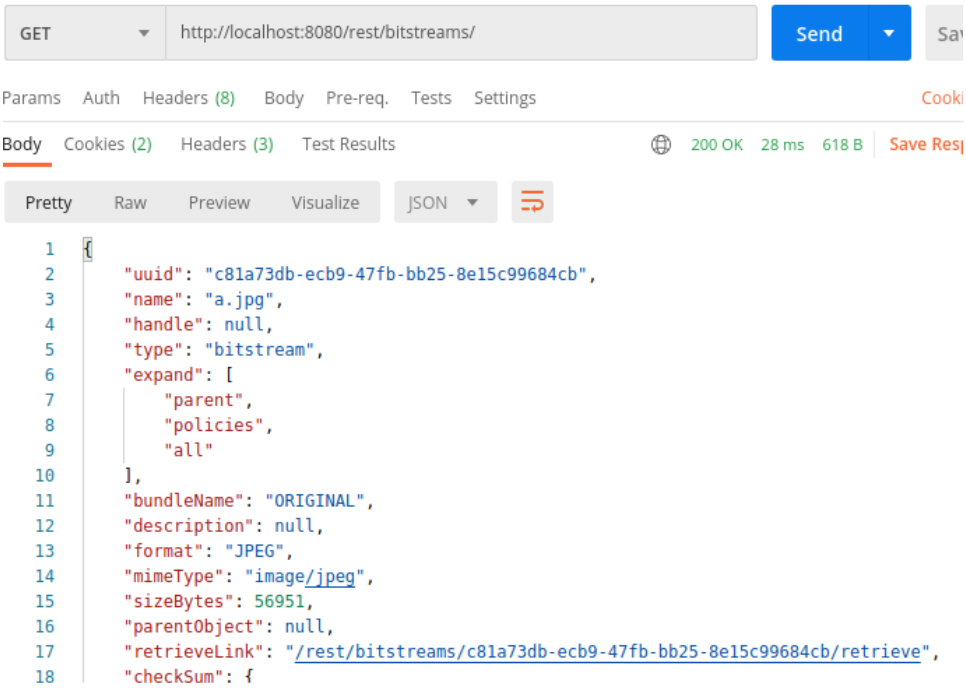
نقوم بإضافة اسم للملف وذلك بتحديث الملف الموجود مسبقا من أجل اسمه وذلك باستخدام id الخاص به في الدي سبيس.



الشكل III-51: واجهة إضافة اسم للملف

✓ الحصول على بيانات الملفات:

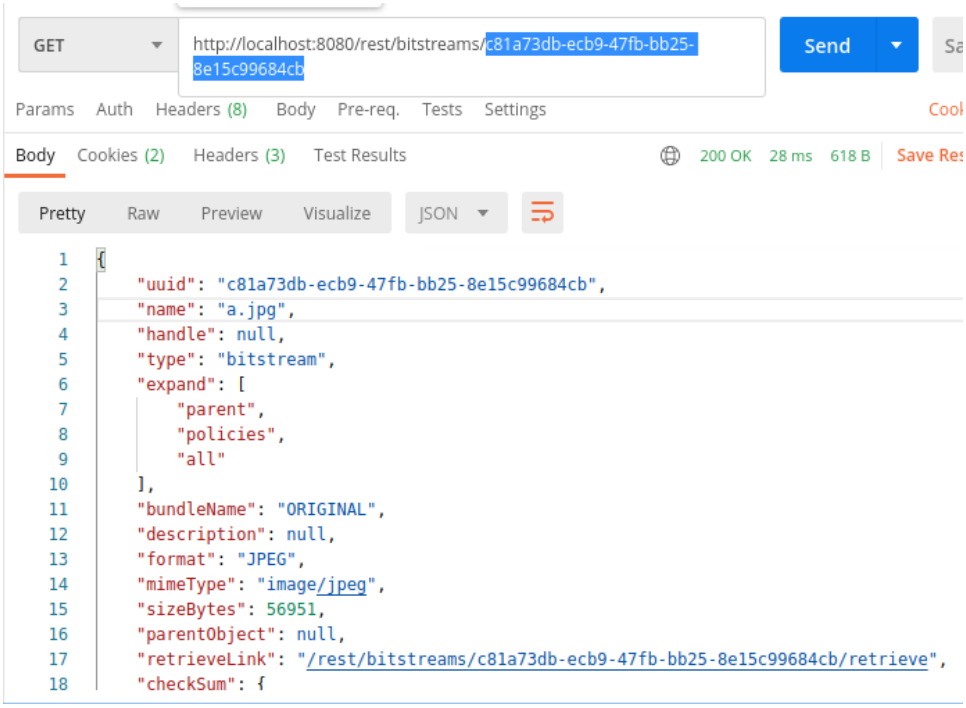
نقوم بالحصول على البيانات الخاصة بالملفات الموجودة في الدي سبيس.



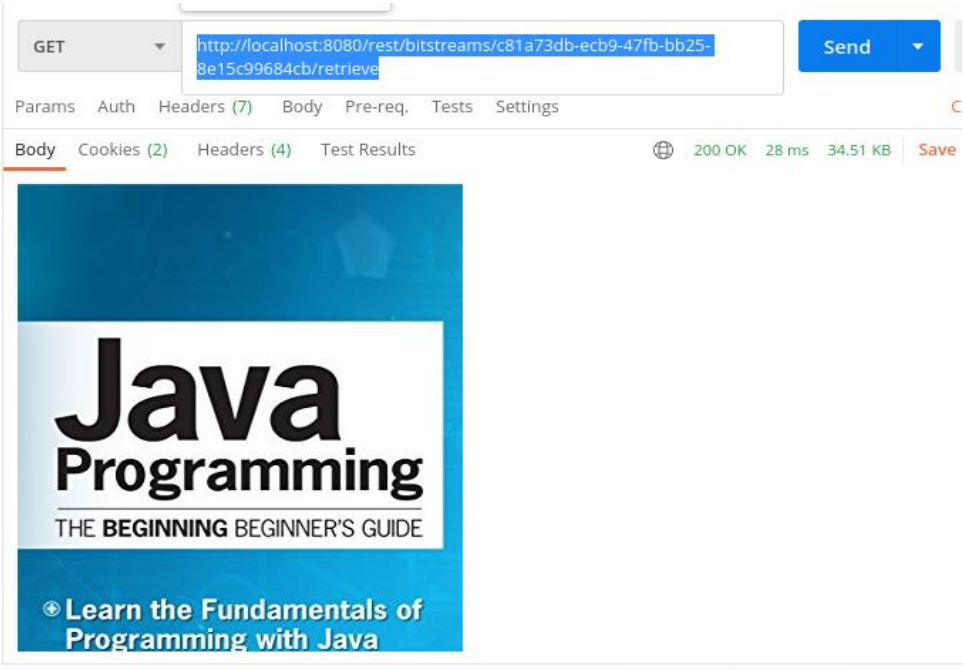
الشكل III-52: واجهة الحصول على الملفات

✓ الحصول على الملف و فتحه:

نقوم بالحصول على الملف للحصول على الرابط الذي يمكننا من خلاله الوصول الى ملفه وذلك ب id الخاص به في الدي سبيس.



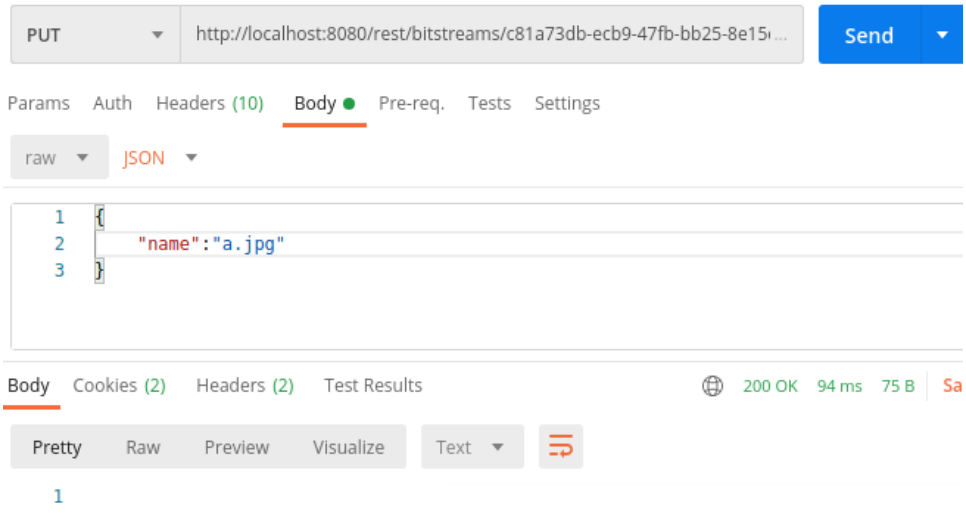
الشكل III-53: واجهة الحصول على رابط الملف



الشكل III-54: واجهة فتح الملف

✓ تحديث بيانات الملف:

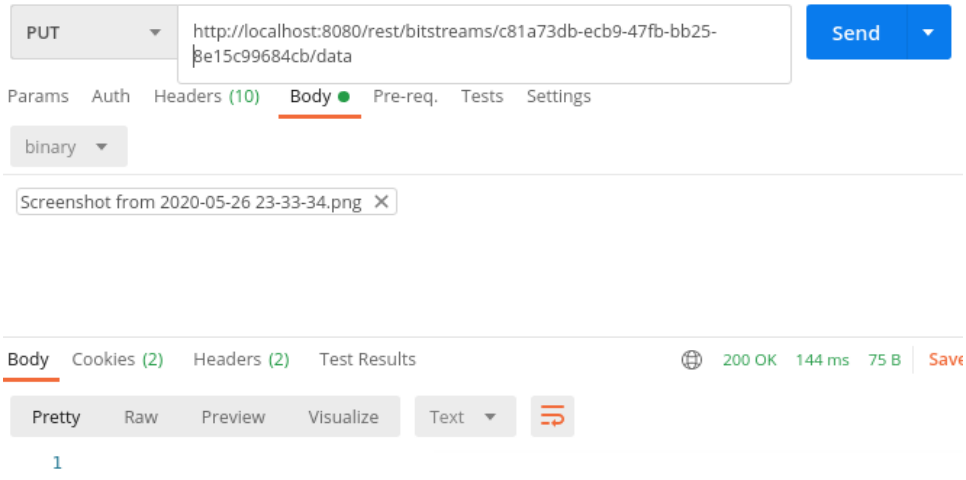
نقوم بتحديث بيانات الملف وذلك باستخدام id الخاص به في الدي سبيس.



الشكل III-55: واجهة تحديث بيانات الملف

✓ تحديث الملف:

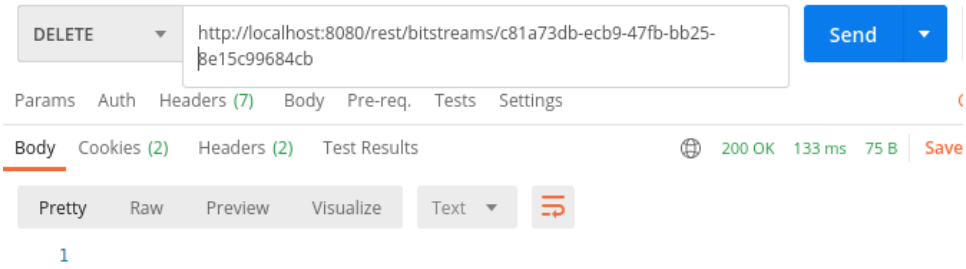
نقوم بتحديث الملف وذلك باستخدام id الخاص به في الدي سبيس.



الشكل III-56: واجهة تحديث الملف

✓ حذف الملف:

نقوم بحذف الملف وذلك باستخدام id الخاص به في الدي سبيس.

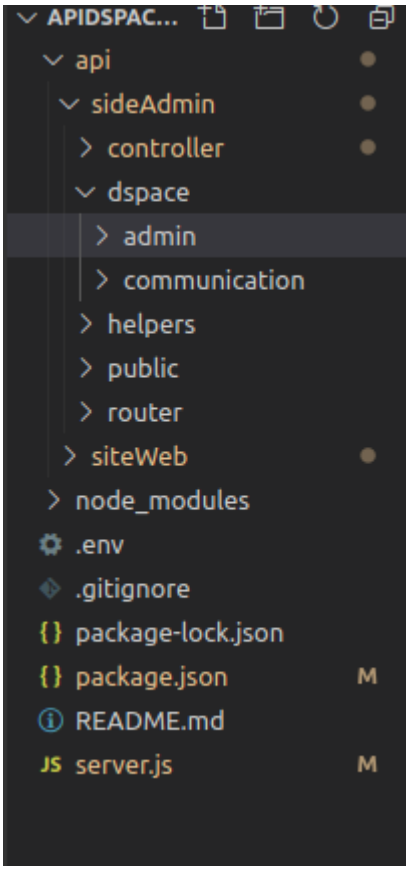


الشكل III-57: واجهة حذف الملف

5-3-شرح سرفر nodejs الخاص بالدي سبيس (.DSpace)

✓ بنية المشروع:

هنا نشاهد البنية الكاملة للسرفر.



الشكل III-58: واجهة بنية المشروع

✓ الملف الاساسي للسرفر:

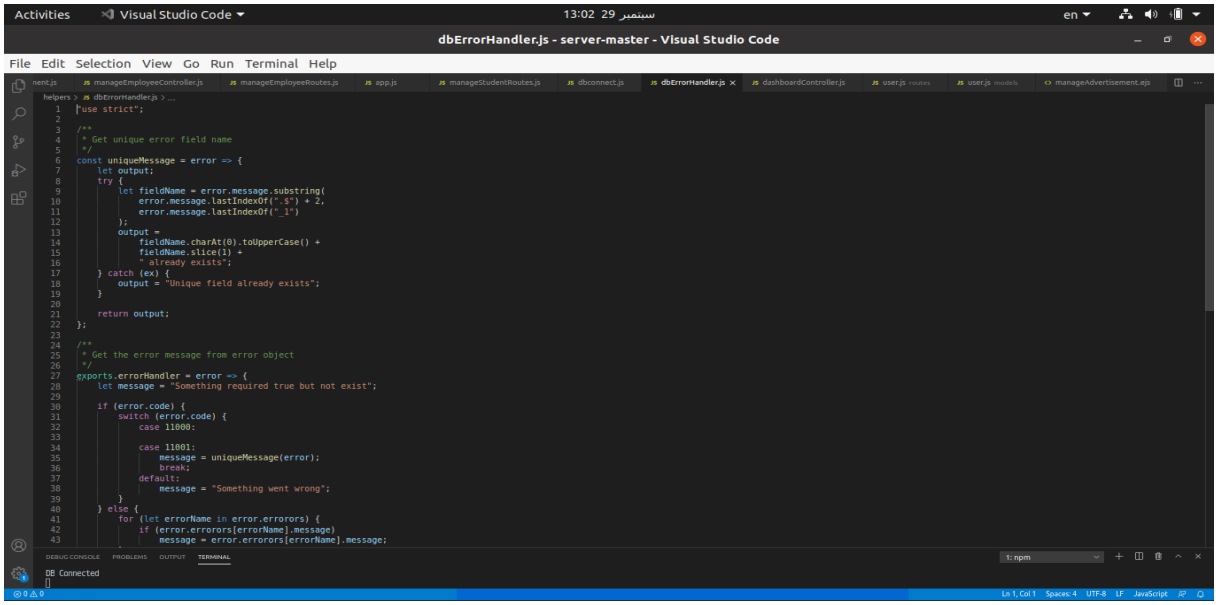
هنا نشاهد الكود الخاص بالملف الأساسي للسرفر.

```
server.js > ...
1   const express = require('express')
2   const bodyParser = require("body-parser");
3   require("dotenv").config();
4   var request = require("request");
5   var fs = require("fs");
6   var http = require('http');
7   //app
8   const app = express()
9   // middlewares
10  app.use(bodyParser.urlencoded({ extended: false }));
11  app.use(bodyParser.text());// This will parse your file
12  app.use(bodyParser.json({limit:'50mb'}));
13  app.use(bodyParser.urlencoded({
14    extended: true
15  }));
16  app.use(express.static(__dirname + '/api/siteWeb/public/'));
17  // import routes
18  const routercommunities = require('./api/sideAdmin/router/communities')
19  const routercollections = require('./api/sideAdmin/router/collections')
20  const routerItems = require('./api/sideAdmin/router/items')
21  const routerAdmin = require('./api/sideAdmin/router/main.add.information')
22  const routerSearch = require('./api/siteWeb/api/router/searchitem')
23  //routes
24  app.use("/api",routercommunities,routercollections,routerItems);
25  app.use("/api",routerAdmin,routerSearch);
26  const port = process.env.PORT || 9500
27  app.listen(port, () => {
28    console.log(`Server is running on port ${port}`);
29    console.log(`Database is connect`);
30  })
31
```

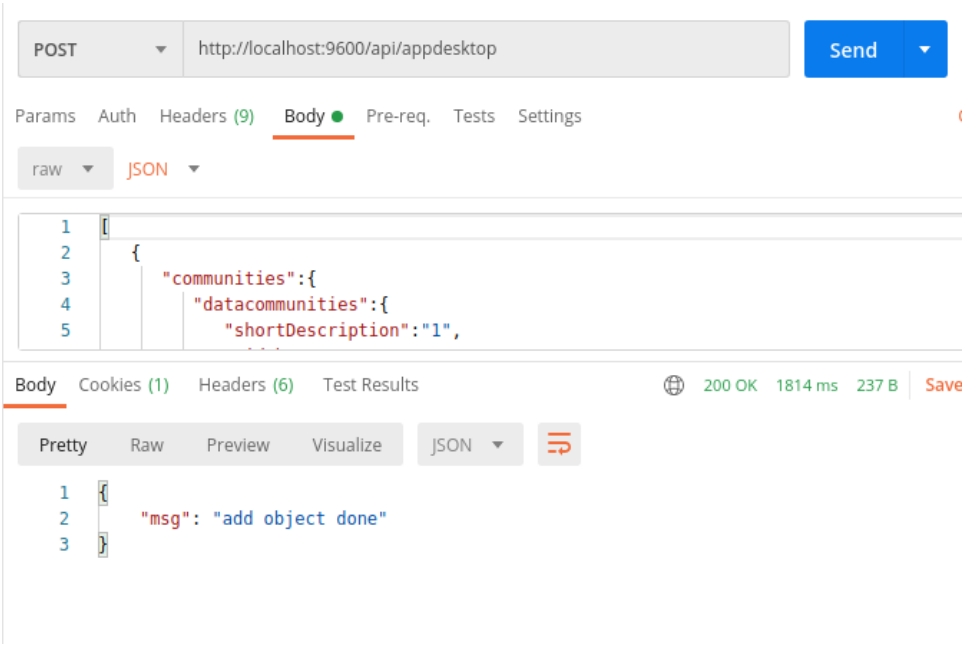
الشكل III-59: واجهة الملف الاساسي للسرفر

- ✓ object json

هنا نشاهد ملف الاخطاء المحتملة .



الشكل III-60: واجهة الاخطاء المحتملة



الشكل III-61: واجهة اظافة object json الى DSpace باستخدام postman

✓ الدالة التي تقوم باظافة :object json

هنا نشاهد الدالة التي تقوم بمعالجة البيانات المرسلة من تطبيق سطح المكتب و ارسالها الى api DSpace .

```
exports.addObjectAppDesktop = async (req, res) => {
  var checkDelete = false

  let arrayObjectJson = JSON.parse(JSON.stringify(req.body))


  switch(Array.isArray(arrayObjectJson)){
    case false:
      //errorPrint
        return res.send({Error:"Is not Array"})
    case true:
        if(Array.isArray(arrayObjectJson) && arrayObjectJson.length === 0)
        //errorPrint
            return res.send({Error:"Array App Desktop is empty"})
    case true:{

      await loginToDspace().then(async (token) =>{

          for(const value of arrayObjectJson) {


          var idLevelCommunities = await getIdLevelCommunities(value.communities.datacomm
          .then((resolve) =>{


            return resolve

          }).catch(err =>{
            return {Error:err}
          })
```

الشكل III-62: واجهة الدالة التي تقوم باظافة object json

✓ دالة تسجيل الدخول الى :dspace

هنا نشاهد الدالة التي تقوم بتسجيل الدخول وانشاء جلسة جديدة.

```
const axios = require('axios')
const qs = require('qs')
require("dotenv").config();

// login function
const loginToDspace =  function loginToDspace(){
  let url = process.env.DATABASE+"login";

  return new Promise((resolve, reject) => {
    axios({
    method: "post",
    url: url,
    data: qs.stringify({
      email: process.env.EMAIL,
      password: process.env.PASSWORD
    }),


    headers: {
      "content-type": "application/x-www-form-urlencoded;charset=utf-8" },

  }).then((response) => {

      let token  = response.headers["set-cookie"][0];
      resolve(token);
  },
    (error) => {
      console.log(error);
      reject();
    }
```

الشكل III-63: واجهة الدالة التي تقوم باظافة object json

✓ ملف التسميات العامة:

هنا نشاهد التسميات العامة التي نحتاجها في المشروع.

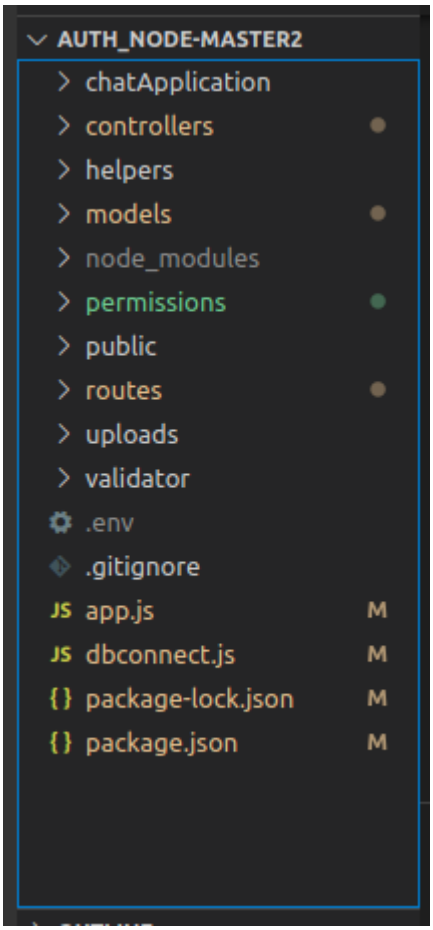
```
PORT=9600
DATABASE=http://localhost:8080/rest/
COMMUNITIES=communities/
COLLECTIONS=collections/
ITEMS=items/
EMAIL=sifouamra@gmail.com
PASSWORD=1421998
API=http://localhost:5002/api/collections
NAME=The Central Library
```

الشكل III-64: واجهة ملف التسميات العامة

6-3-شرح سرفر nodejs العام.

✓ بنية المشروع:

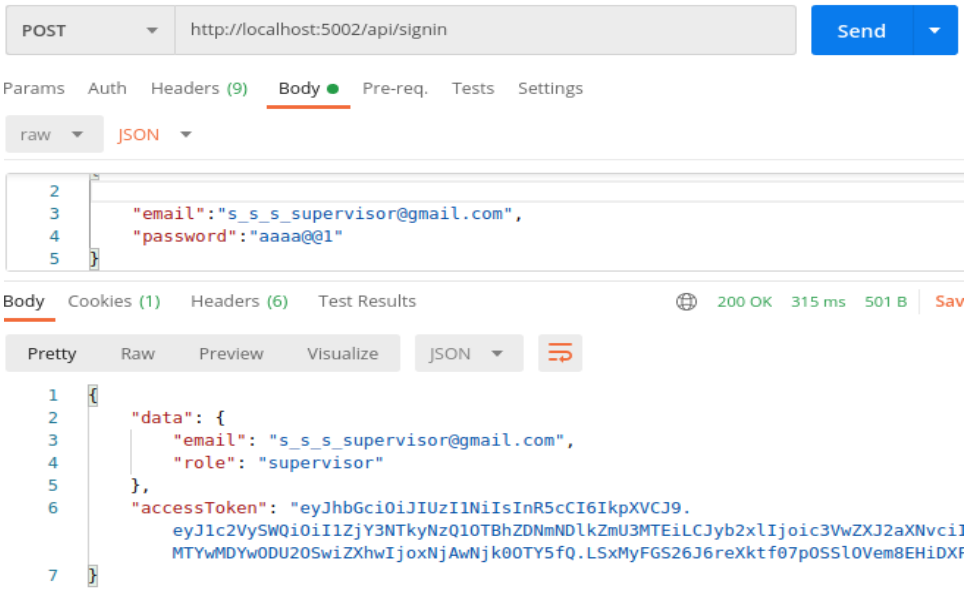
هنا نشاهد البنية العامة للمشروع.



الشكل III-65: واجهة بنية المشروع

✓ تسجيل الدخول:

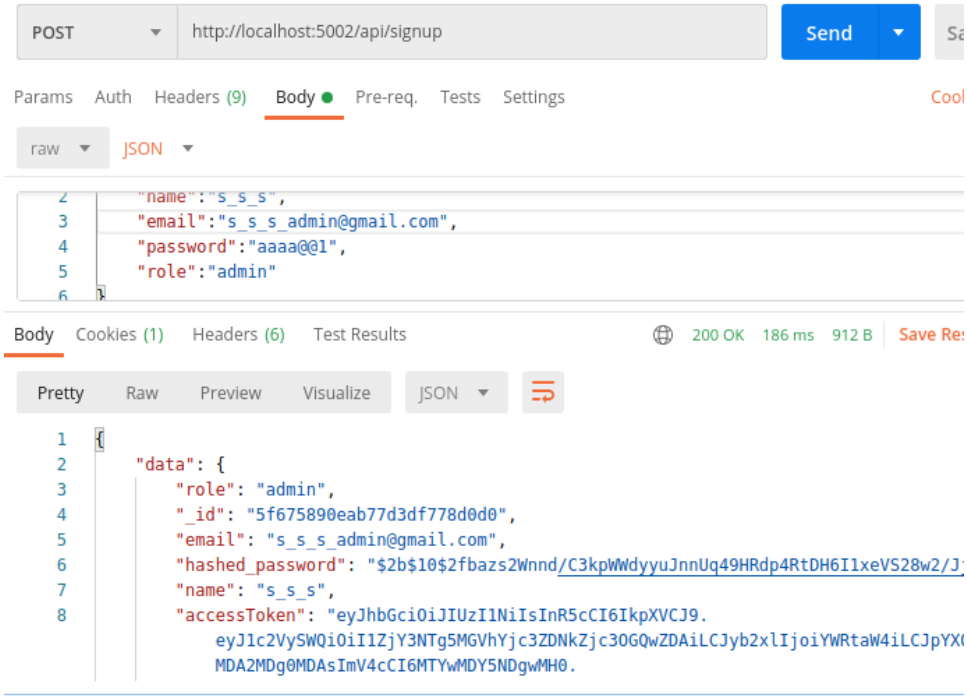
هنا نشاهد متطلبات تسجيل الدخول بالنسبة للمستخدمين.



الشكل III-66: واجهة تسجيل الدخول

✓ التسجيل من أجل المدير:

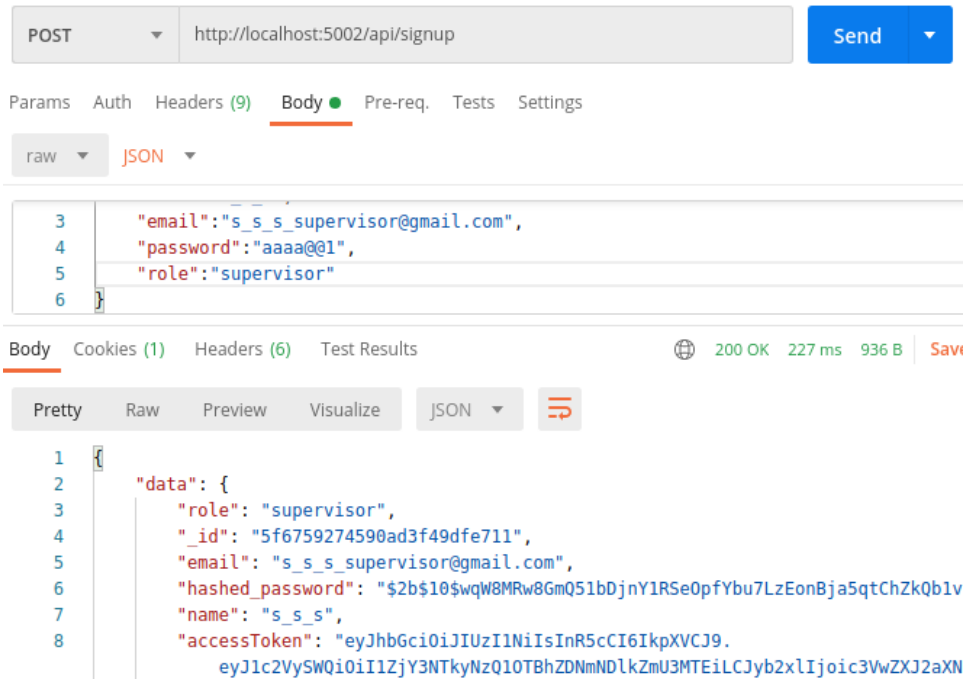
هنا نشاهد كيفية اظافة ادمن فقط اذا كان الشخص ادمن.



الشكل III-67: واجهة التسجيل من أجل المدير

✓ التسجيل من أجل المشرف:

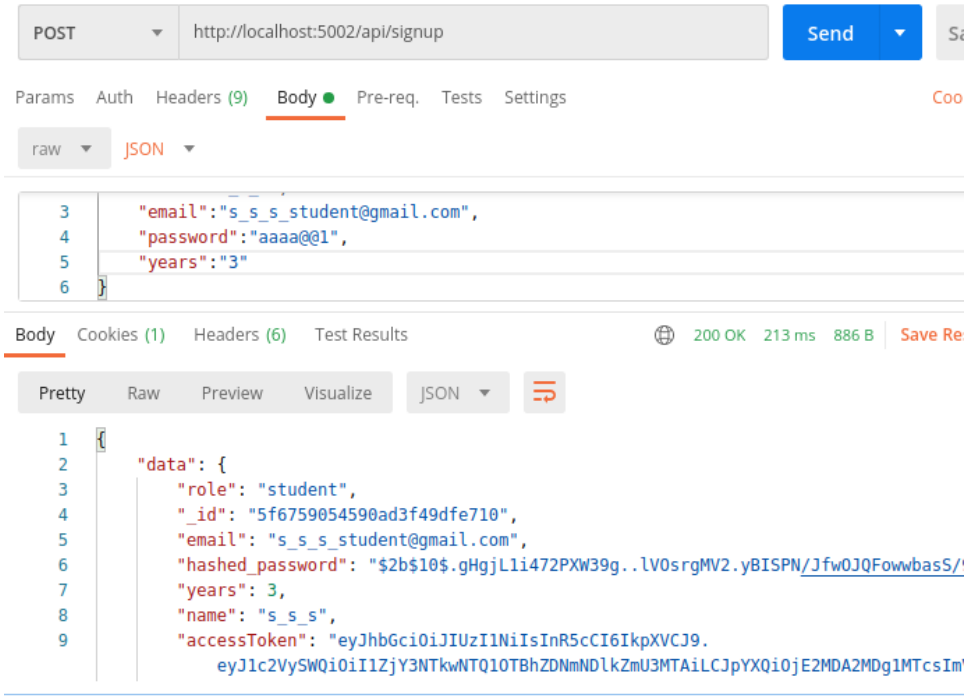
هنا نشاهد تسجيل المشرف فقط اذا كان المستخدم مدير.



الشكل III-68: واجهة التسجيل من أجل المشرف

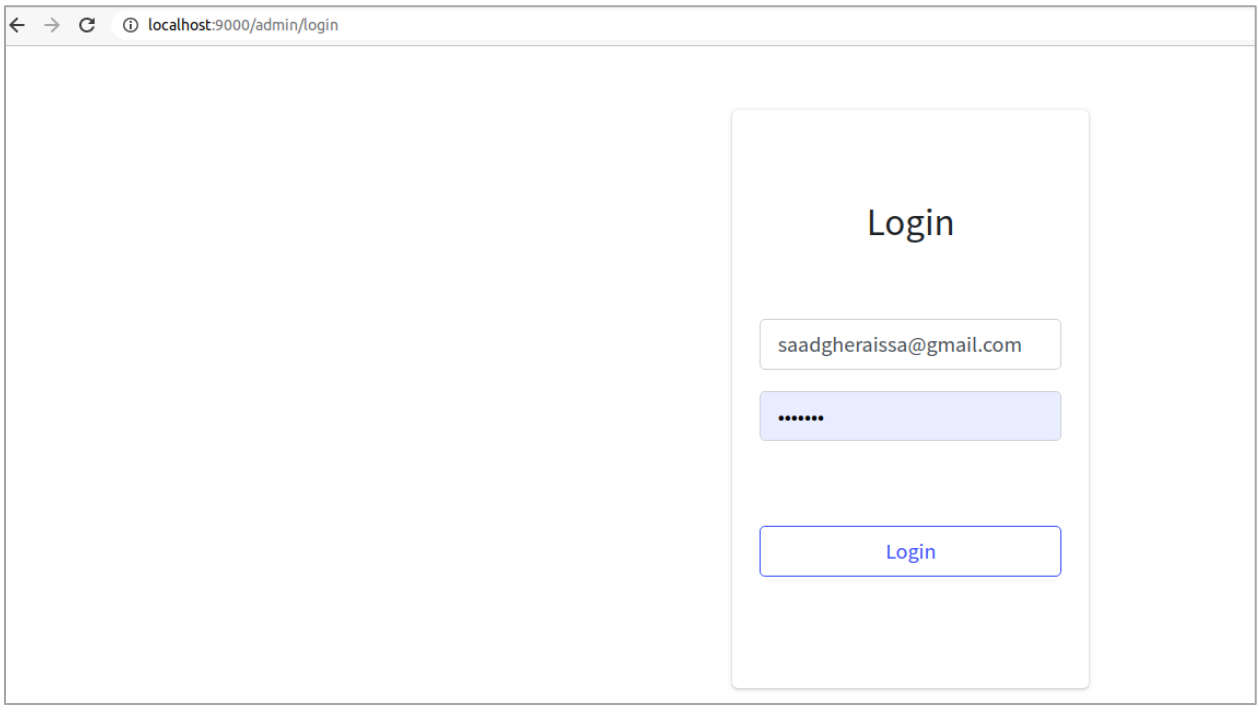
✓ التسجيل من أجل الطالب:

هنا يستطيع الجميع التسجيل من أجل طالب.



الشكل III-69: واجهة التسجيل من أجل الطالب

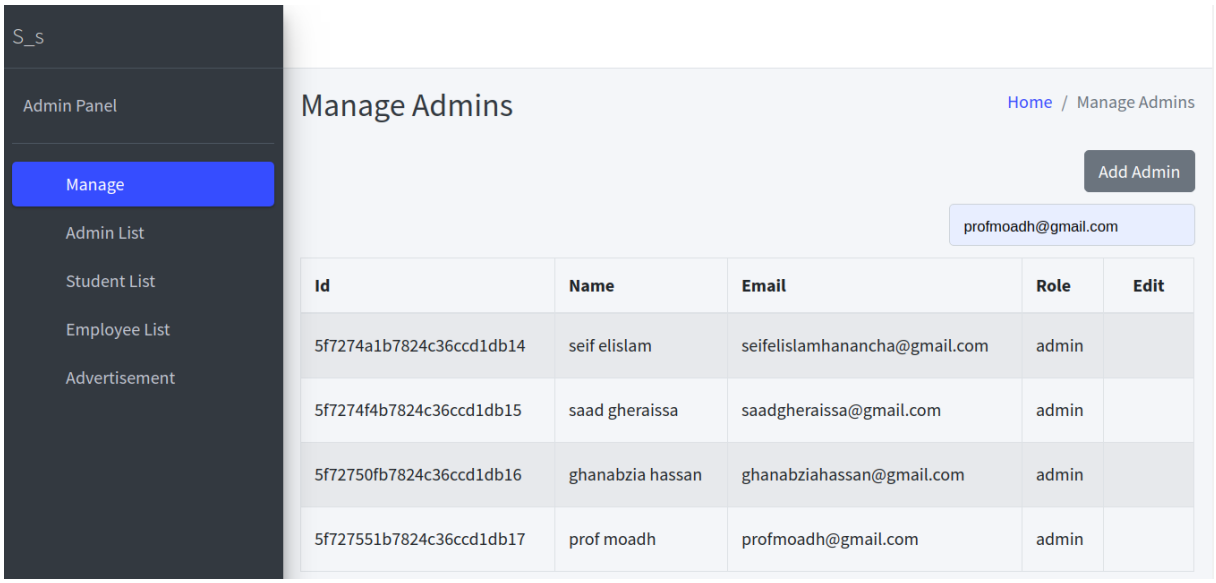
✓ واجهة تسجيل الدخول للمسؤول



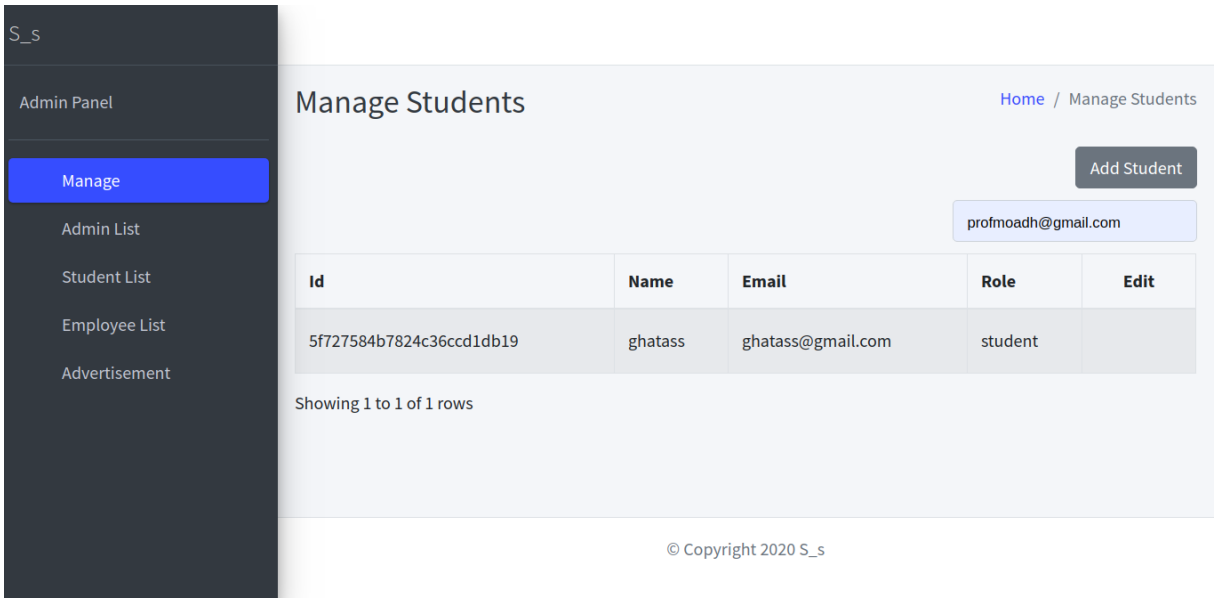
الشكل III-70: واجهة تسجيل الدخول للمسؤول

✓ المسؤول

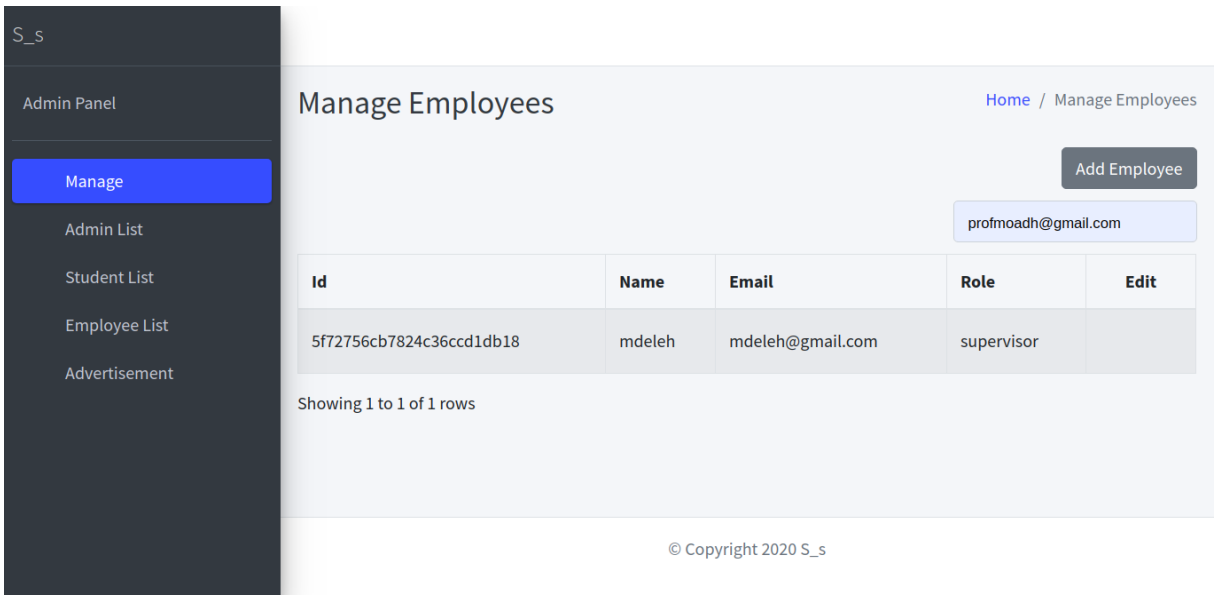
هنا نشاهد الصلاحيات المختلفة للمسؤول في نظامنا.



الشكل III-71: ادارة المسائيل



الشكل III-72: ادارة المنخرطين

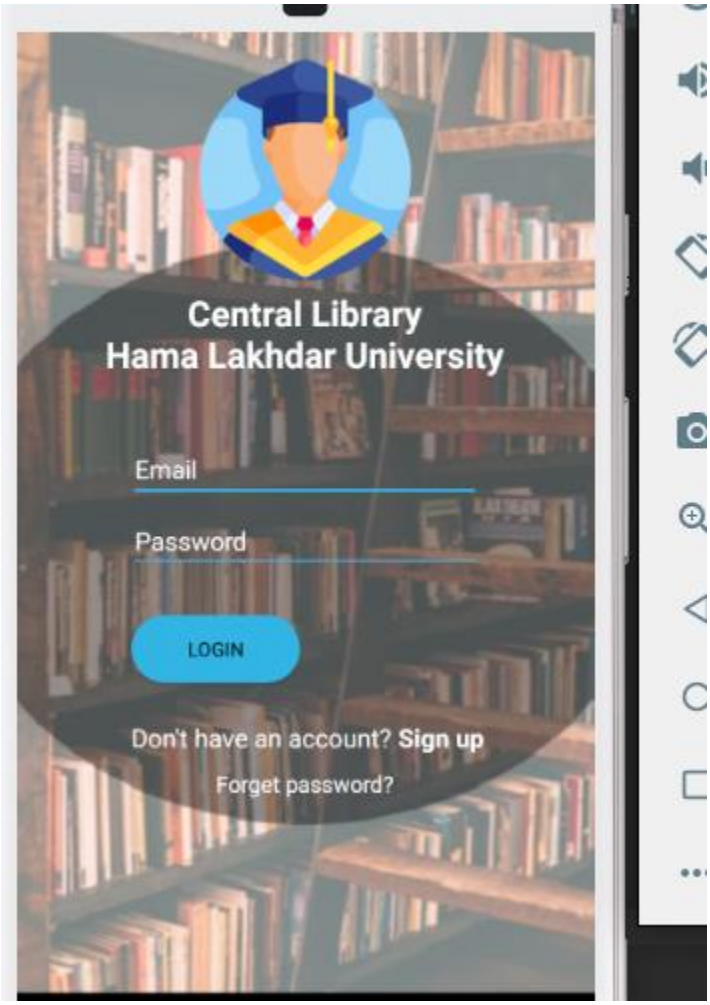


الشكل III-73: ادارة الموظفين

4- شرح واجهات تطبيق الاندرويد

4-1-تسجيل الدخول

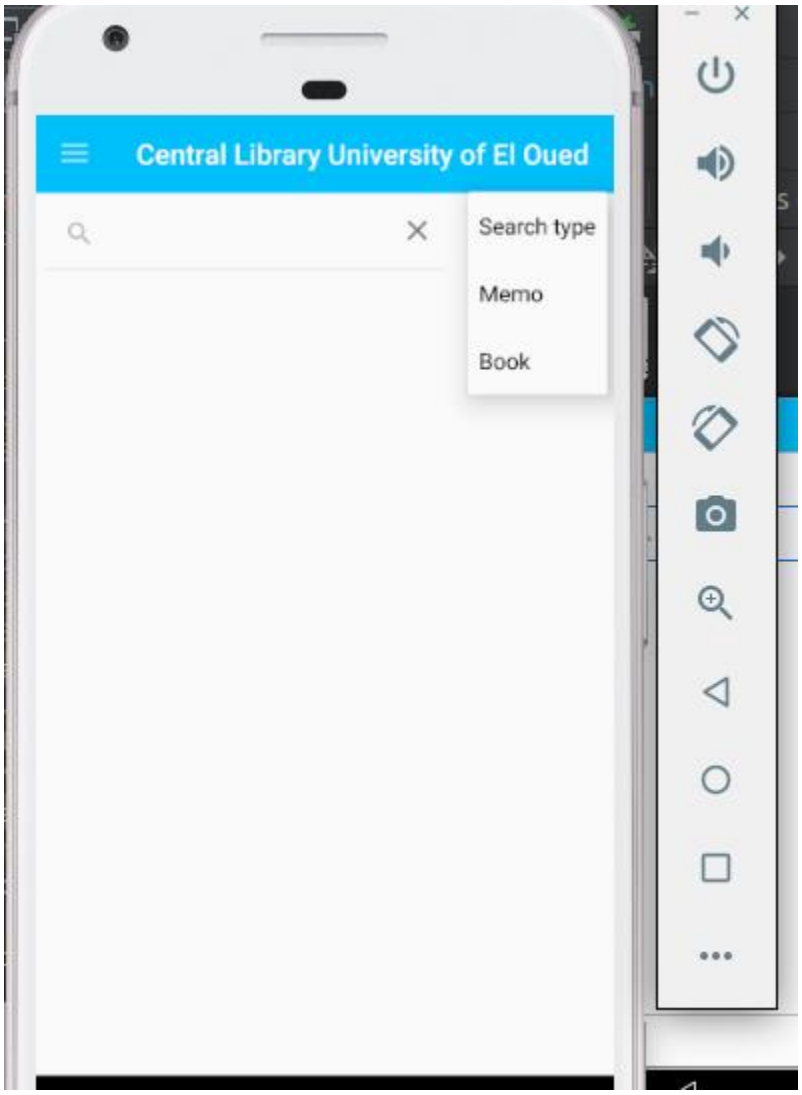
من هنا يمكننا الدخول الي تطبيق الاندرويد عبر ادخال البريد الالكتروني و كلمة المرور كما هو موضح في الشكل ادناه بالنسبة للمخرطين:



الشكل III-74: تسجيل الدخول لتطبيق الاندرويد

2-4-البحث

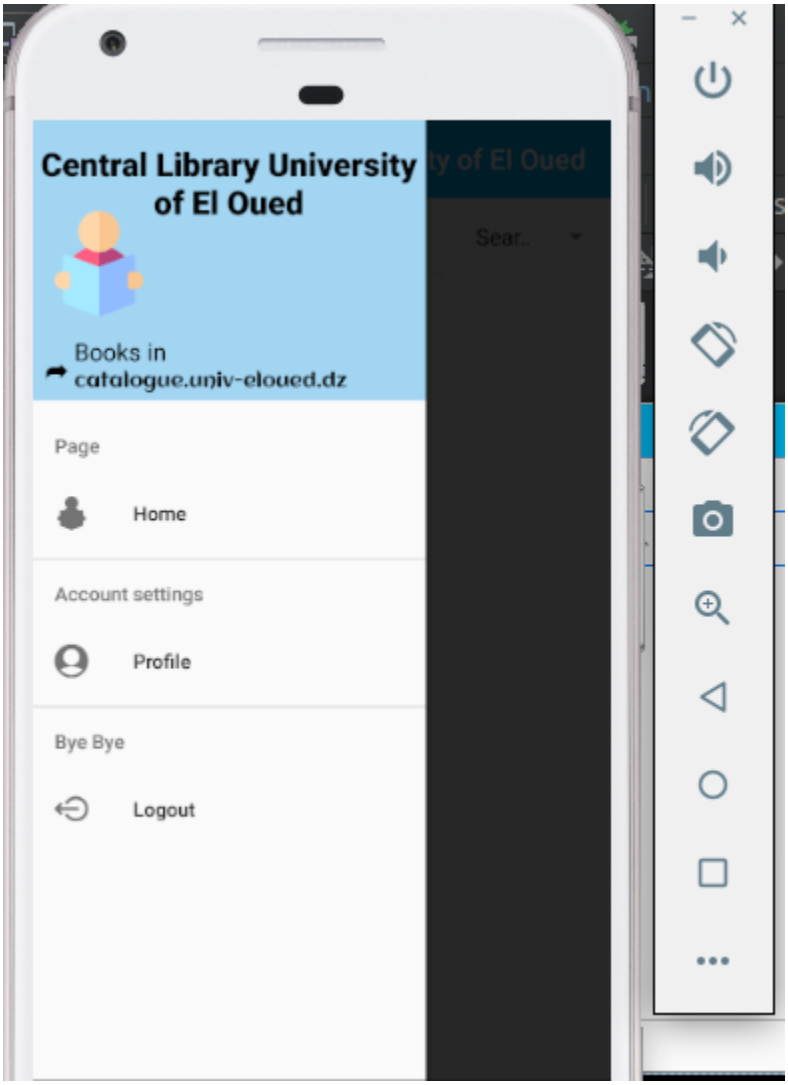
بالضغط علي ايقونة البحث يمكننا البحث علي الكتب الموجودة في نظامنا



الشكل III-75: واجهة البحث لتطبيق الاندرويد

3-4-الصفحة الرئسية لتطبيق الاندرويد

في الصفحة الرئسية يمكننا الاطلاع علي الاعلانات بالضغط علي ال (home) و يمكننا ايضا تسجيل الخروج بالظغط علي (logout) و في ال (profile) يمكننا الاطلاع علي الملف الشخصي و تغيير كلمة المرور

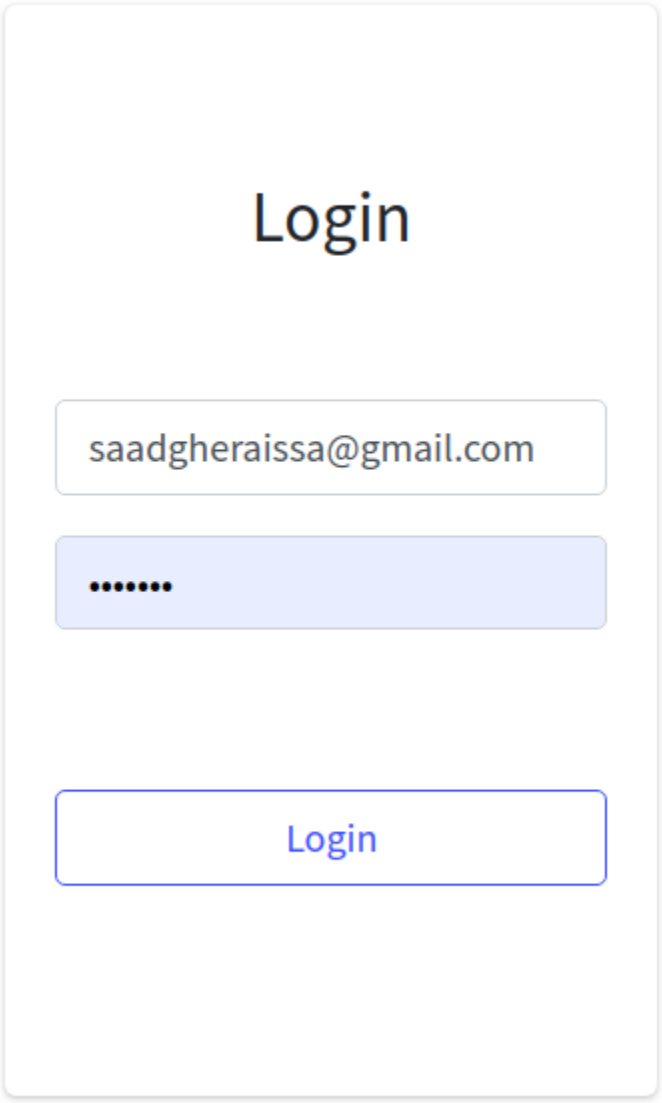


الشكل III-76: واجهة الصفحة الرئيسة لتطبيق الاندرويد

5- شرح واجهات التطبيق المكتبي

5-1-تسجيل الدخول للتطبيق المكتبي

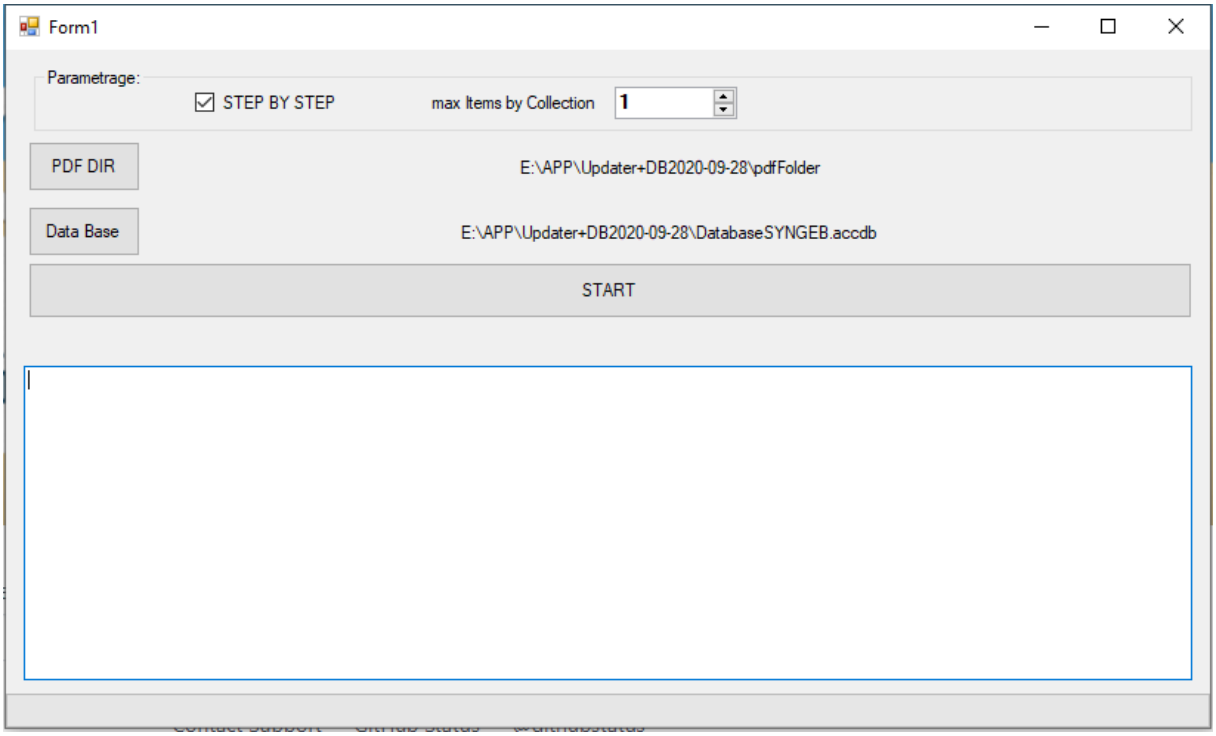
من خلال اداع البريد الالكتروني و كلمة المرور صحيحة يمكننا الدخول الي التطبيق المكتبي.



الشكل III-77: تسجيل الدخول للتطبيق المكتبي

2-5-الصفحة الرئسية للتطبيق المكتبي:

في الصفحة الرئسية للطبيق المكتبي بالضغط علي زر(Data Base) يمكننا اختيار ملف من قاعدة البيانات السنجاب و بالضغط علي زر (PDF DIR) يمكننا اختيار مجلد صور الكتب و في (max items by collections) يمكننا تحديد عدد الكتب ومعلوماتها التي يتم تجهيزها ثم ارسالها و في الاخير بالضغط علي الزر (start) يمكننا ارسال البيانات الي ال REST API.



الشكل III-78: الصفحة الرئيسية للتطبيق المكتبي

الخاتمة:

من خلال هذا الفصل نكون قد انهينا عبئ موضفو المكتبة المركزية من نقل البيانات من السنجاب الي كاتلوغ يدويا الي نقل آلي حديث.

الخاتمة العامة

من خلال الدراسة التي قمنا بها بكل مراحلنا قد انتهنينا من تطوير مشروع ادراة الارشفة الالكترونية لموقع المكتبة المركزية , وقد انتهت مشاكلها فقد قمنا بتوضيف كفائتنا لدراسة الجزء النظري منه ثم الانتقال الي تصميمه ودراسة الاحتمالات الممكنة له ثم تنفيذ المشروع بكفائة و اباستعمال اسلوب لغة النمذجة و ذالك بانشاء موقع ويب و تطبيق اندرويد وتطبيق مكتبي و لقد كان هذا المشروع جد مفيد لنا بتعلم اشياء جديدة تطوير البرمجيات و التعامل مع نظام عالمي لادارة المكتبات مع توضيف المعلومات النظرية و التطبيقية في سنوات الدراسة.

قائمة المصادر والمراجع

Département de l'Information Scientifique et Technique,CERIST "Le LOgiciel SYNGEB
https://www.dist.cerist.dz/logiciels.php [visité le 20/09/2020]
DocuWare document management and workflow automation , www.docuware.com , [visité le 11/09/2020]
DocuWare , http://www.wsiarabia.com/Products/DocuWare/tabid/65/language/ar-JO/Default.aspx , [visité le 11/09/2020]
What Is GitHup , and What Is It Used For, KORBIN BROWN, https://www.howtogeek.com/180167/htg-explains-what-is-github-and-what-do-geeks-use-it-for [visité le 21/09/2020]
What is npm, Installing npm ,https://www.w3schools.com/whatis/whatis_npm.asp, [visité le 13/09/2020]
Body-parser middleware in Node.js, Installation of body-parser module, https://www.tutorialspoint.com/nodejs/nodejs_introduction.htm , [visité le 20/09/2020].
What is Frameworks? [Definition] Types of Frameworks, Vijay Singh, https://hackr.io/blog/what-is-frameworks, [visité le 21/09/2020]
Body-parser middleware in Node.js, gouravhammad, https://www.geeksforgeeks.org/body-parser-middleware-in-node-js/, [visité le 14/06/2020].
How to Install Git on Ubuntu 18.04, https://linuxize.com/post/how-to-install-git-on-ubuntu-18-04,[visité le 12/05/2020].
npm-package-lock.json , A manifestation of the manifest , https://developer.mozilla.org/en-US/docs/Web/HTTP/Status/500
MDN Web Docs. The HTTP 200 OK , https://developer.mozilla.org/en-US docs/Web/HTTP/Status/200 , [visité le 15/08/2020].
MDN Web Docs, The HTTP 404 Not Foun , https://developer.mozilla.org/en-US/docs/Web/HTTP/Status/404 , , [visité le 18/08/2020].
MDN Web Docs, 403 Forbidden ,https://developer.mozilla.org/en-US/docs/Web/HTTP/Status/403 ,[visit le 18/08/2020].
What Is My IP Address , What is a localhost , https://whatismyipaddress.com/localhost ,[visit le 12/07/2020].
PDF.CO, Postman Tutorial for Beginner and Advanced Developers , https://pdf.co/blog/postman-tutorial , [visit le 04/08/2020].

Help Desk Geek, What Is Localhost and How Can You Use It? , Joseph Boone https://helpdeskgeek.com/networking/what-is-localhost-and-how-can-you-use-it/ ,[visit le 03/06/2020].
Docker for DSpace Testing and Development , Installing Docker, https://dspace-labs.github.io/DSpace-Docker-Images/documentation/tutorialSetup.html , [visit le 03/05/2020].
Docker, Install Docker Engine on Ubuntu, https://docs.docker.com/engine/install/ubuntu/ ,[visit le 15 /05/ 2020].
Docker , Install Docker Compos , https://docs.docker.com/compose/install/, [visit le 17 /05/ 2020].
w3schools.com, The language for building web pages, https://www.w3schools.com/ ,[visit le 17/05/2020].
Restfulapi, REST API Tutorial, https://restfulapi.net/ ,[visit le 09/09/2020].
UW-Milwaukee Help Desk, Postman , https://kb.uwm.edu/uwmhd/page.php?id=95041 ,[visit le 04/08/2020].
Digitalcraft , What is Postman, and Why Should I Use It? , https://www.digitalcrafts.com/blog/student-blog-what-postman-and-why-use-it ,[visit le 05/08/2020].
How-To Geek, What Is GitHub, and What Is It Used For?, https://www.howtogeek.com/180167/htg-explains-what-is-github-and-what-do-geeks-use-it-for/ , ,[visit le 04/07/2020].
ماهى JSON وكيفية كتابتها وفيما تستخدم , عمر العربي , https://www.motwr.com/_ , تم الاطلاع عليه يوم 12 سبتمبر 2020.
المنتجات و البرمجيات ,مركز البحث في الإعلام العلمي و التقني , http://www.cerist.dz/index.php/ar/produits-logiciels-ar/1003-syngeb , تم الاطلاع عليه يوم 11 سبتمبر 2020.
- نظام iDOX Standard http://www.idoxsoft.com/ar/package/%D9%86%D8%B8%D8%A7%D9%85-idox-/standard تم الاطلاع عليه يوم 12 سبتمبر 2020.
ما هو HTTP , https://motaber.com/%d9%85%d8%a7-%d9%87%d9%88-http/ , تم الاطلاع عليه يوم 25 سبتمبر.
النظم الآلية و الخدمات المكتبية, حيدرة عبد القادر , https://bplpadrar.dz/frm/threads/198/ , تم الاطلاع عليه يوم 12 سبتمبر 2020.

ملحق:

- تثبيت و اعداد نظام الدي سبايس(DSpace)

1-2- تثبيت الدي سبايس (DSpace 6)

في هذا الجزء سوف نقدم كيفية تثبيت ال DSpace علي ال Docker في Ubuntu 20.04

❖ تثبيت Docker Engine on Ubuntu 20.0

You can install Docker Engine in different ways, depending on your need

- Install using the repository
- Install from a package

We will use the first way

✓ Set up the repository

a) Update the apt package index and install packages to allow apt to use a repository over HTTPS:

```
$ sudo apt-get update
$ sudo apt-get install \
apt-transport-https \
ca-certificates \
curl \
gnupg-agent \
software-properties-common
```

a) Add Docker's official GPG key:

```
$ curl -fsSL https://download.docker.com/linux/ubuntu/gpg | sudo apt-key add -
```

b) Use the following command to set up the **stable** repository.

```
$ sudo add-apt-repository \
  "deb [arch=amd64] https://download.docker.com/linux/ubuntu \
  $(lsb_release -cs) \
  stable"
```

- ✓ Install Docker Engine

b) Update the apt package index, and install the *latest version* of Docker Engine and containerd, or go to the next step to install a specific version:

```
$ sudo apt-get update
$ sudo apt-get install docker-ce docker-ce-cli containerd.io
```

c) To install a *specific version* of Docker Engine, list the available versions in the repo, then select and install:

-List the versions available in your repo:

```
$ apt-cache madison docker-ce

 docker-ce | 5:18.09.1~3-0~ubuntu-xenial | https://download.docker.com/linux/ubuntu xenial/stable amd64 Packages
docker-ce | 5:18.09.0~3-0~ubuntu-xenial | https://download.docker.com/linux/ubuntu xenial/stable amd64 Packages
docker-ce | 18.06.1~ce~3-0~ubuntu | https://download.docker.com/linux/ubuntu xenial/stable amd64 Packages
docker-ce | 18.06.0~ce~3-0~ubuntu | https://download.docker.com/linux/ubuntu xenial/stable amd64 Packages
      ...
```

- ✓ Install a specific version using the version string from the second column, for example, 5:18.09.1~3-0~ubuntu-xenial.

```
$ sudo apt-get install docker-ce=<VERSION_STRING> docker-ce-cli=<VERSION_STRING> containerd.io
```

3.Verify that Docker Engine is installed correctly by running the hello-world image

```
$ sudo docker run hello-world
```

And after that we Verify Installation by tape this commends:

Verify Git Install

```
git version
```

Verify Docker Installation

```
docker version
```

Verify Docker Compose Installation

```
docker-compose version
```

Check terminal access to a Docker image

```
docker run -it --rm ubuntu /bin/bash
```

تثبيتDSpace Docker Compose Files

```
cd
git clone https://github.com/DSpace-Labs/DSpace-Docker-Images.git
cd DSpace-Docker-Images/docker-compose-files/dspace-compose
```

Start Using DSpace-Docker and Go to the Docker Compose Directory

```
cd
cd DSpace-Docker-Images/docker-compose-files/dspace-compose
```

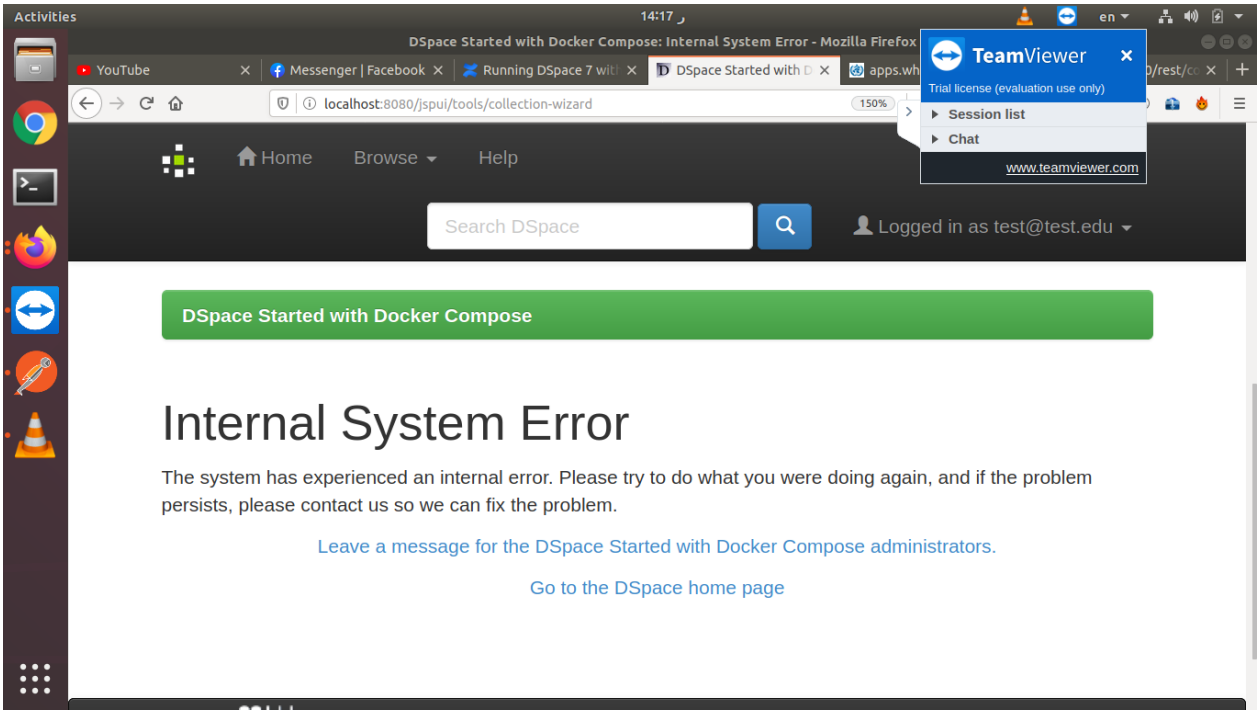
Start DSpace 6

```
docker-compose -p d6 -f docker-compose.yml -f d6.override.yml pull
docker-compose -p d6 -f docker-compose.yml -f d6.override.yml up -d
```

It will take 2-5 minutes for the services to start completely. Once they have started, you can view them on the web. The first time you start DSpace, it will load some default AIP content. This process takes an additional 3-5 minutes

http://localhost:8080/xmlui

and when we go to create communities we get this error



الشكل1: الخطاء المحتمل بعد تثبيت الدي سبايس

Now let’s fix this error;

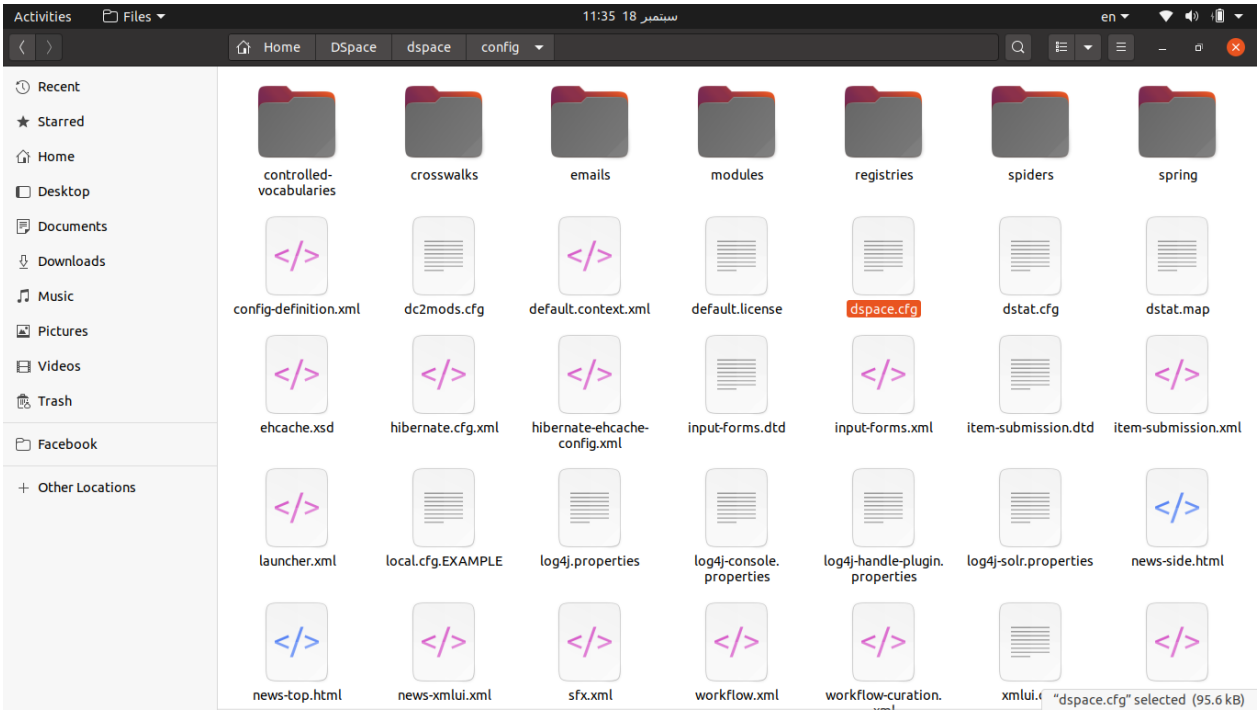
1.running DSpace instances

```
docker exec dspace /dspace/bin/dspace index-discovery
```

2.Create an additional admin user

```
docker exec -it dspace /dspace/bin/dspace create-administrator
```

Go to the dspace.cfg file in DSpace folder



الشكل2: مكان وجود ملف dspace.cfg

Replace example to your mail

```
109
110 ##### Email settings ######
111
112 # SMTP mail server (allows DSpace to send email
    notifications)
113 mail.server = smtp.example.com
114
115 # SMTP mail server authentication username and password
    (if required)
116 mail.server.username = ******gmail.com
117 mail.server.password = ******
118
119 # SMTP mail server alternate port (defaults to 25)
120 mail.server.port = 465
121
122 # From address for mail
123 # All mail from the DSpace site will use this 'from'
    address
```

الشكل3: التوضيح بدقة مكان التعديل علي mail server

Add your mail and your password

```
113 mail.server = smtp.gmail.com
114
115 # SMTP mail server authentication username and password
    (if required)
116 mail.server.username = ****@Exemple.com
117 mail.server.password = **********
118
119 # SMTP mail server alternate port (defaults to 25)
120 mail.server.port = 465
121
122 # From address for mail
123 # All mail from the DSpace site will use this 'from'
    address
124 mail.from.address = dspace-noreply@myu.edu
125
126 # Name of a pre-configured Session object to be fetched
    from a directory.
127 # This overrides the Session settings above.  If none
```

الشكل4: اضافة اسم المستخدم وكلمة المرور

Change the default post 25 to 465

```
118
119# SMTP mail server alternate port (defaults to 25)
120mail.server.port = 25
121
122# From address for mail
123# All mail from the DSpace site will use this 'from'
   address
124mail.from.address = dspace-noreply@myu.edu
125
126# Name of a pre-configured Session object to be fetched
   from a directory.
127# This overrides the Session settings above.  If none
   can be found, then DSpace
128# will use the above settings to create a Session.
129#mail.session.name = Session
130
131# When feedback is submitted via the Feedback form, it
```

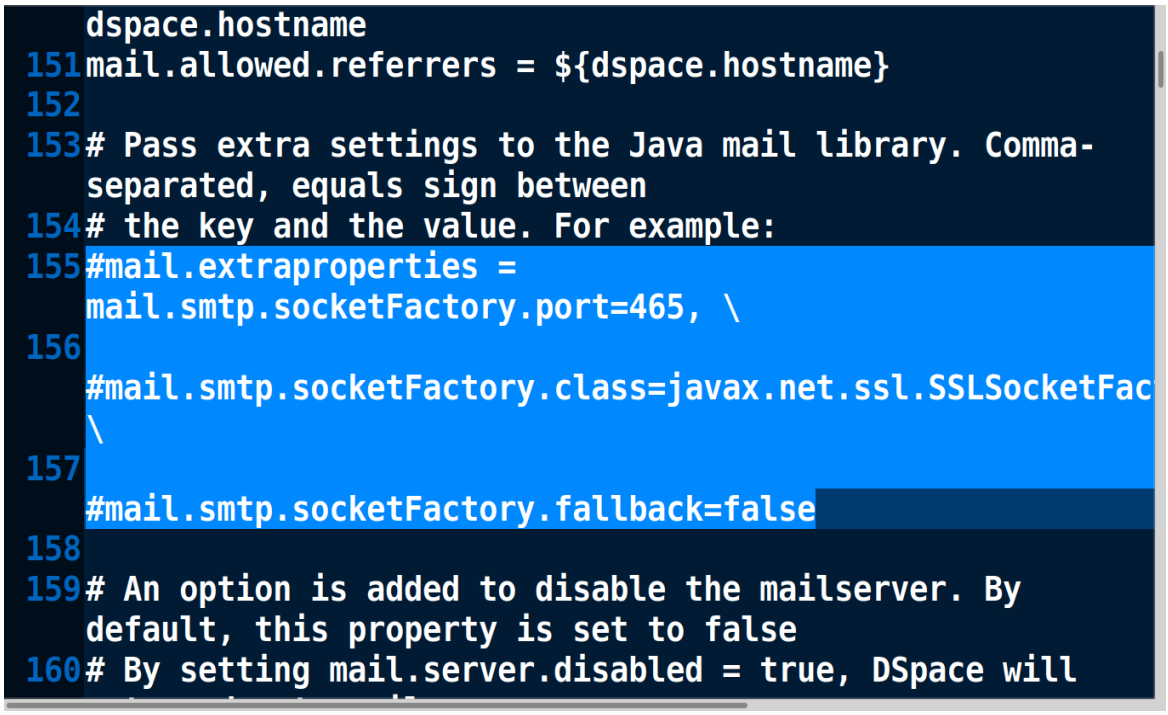
الشكل5: مكان وجود البورت

Like that

```
116 mail.server.username = faresfahim39@gmail.com
117 mail.server.password = 020202
118
119 # SMTP mail server alternate port (defaults to 25)
120 mail.server.port = 465
121
122 # From address for mail
123 # All mail from the DSpace site will use this 'from'
    address
124 mail.from.address = dspace-noreply@myu.edu
125
126 # Name of a pre-configured Session object to be fetched
    from a directory.
127 # This overrides the Session settings above.  If none
    can be found, then DSpace
128 # will use the above settings to create a Session.
129 #mail.session.name = Session
```

الشكل 6: تغيير البورت

Delete these symbols [#] from the beginning of these three lines



الشكل7: مكان حذف رمز #

It's becomes like that

```
151 mail.allowed.referrers = ${dspace.hostname}
152
153 # Pass extra settings to the Java mail library. Comma-
    separated, equals sign between
154 # the key and the value. For example:
155 mail.extraproperties = mail.smtp.socketFactory.port=465,
    \
156
    mail.smtp.socketFactory.class=javax.net.ssl.SSLSocketFact
    \
157
    mail.smtp.socketFactory.fallback=false
158
159 # An option is added to disable the mailserver. By
    default, this property is set to false
160 # By setting mail.server.disabled = true, DSpace will
    not send out emails.
```

الشكل8: حذف رمز #

و من هنا يمكننا استخدام الدي سبايس بشكل عادي.

- **تثبيت مكونات بيئة العمل.**
  - تثبيتnodejs

Open terminal or press Ctrl + Alt + T

for install node.js use the following command

```
sudo apt install nodejs
```

Once installed, verify it by checking the installed version using the following command

```
node -v or node –version
```

- *تثبيت:Npm*

for install npm use the following command

```
sudo apt install npm
```

Once installed, verify it by checking the installed version using the following command

```
npm -v or npm –version
```

- كيفية تثبيت الحزم باستخدام npm

هناك العديد من الحزم مثلا نريد تثبيت body-parser نقوم بالخطوات التالية:

1. نقوم بفتح terminal وذلك ب.Ctrl + Alt + T
2. نقوم بتحديد مسار مجلد المشروع مثلا cd project ثم. cd ApiDspace

ليكون المسار النهائي .home/project/ApiDspace

3. نقوم بادخال الامر التاليnpm install body-parser
4. نتحقق انه تم التثبيت.npm version body-parser

- **تثبيت:Postman**

Open terminal or press Ctrl + Alt + T

For install Postman

```
sudo snap install postman
```

For Create a symbolic link.

```
sudo ln -s /opt/Postman/Postman /usr/bin/postman
```

- تثبيت اداة:Git

Start by updating the package index

```
sudo apt update
```

Run the following command to install Gi

```
sudo apt install git
```

Verify the installation by typing the following command which will print the Git version

```
git –version
```