كلية تقنية المعلومات
قسم / تقنيات الويب

بحث بعنوان :

AUTHORIZATION IN REST USING JAX-RS

اعداد :
أميمة معتوق البنداق
هبة سليمان كريمة
مقدم لدكتور الفاضل :
عبد الناصر ضياف

محتويات البحث

01 المقدمة :

في عالم تطوير التطبيقات الحديثة، تعتبر خدمات الويب RESTful واحدة من الأساليب الأكثر شيوعًا لبناء وتطوير التطبيقات الموزعة. تتيح خدمات الويب RESTful للتطبيقات المختلفة التواصل والتفاعل بشكل فعال من خلال استخدام بروتوكول HTTP ومبادئ REST (Representational State Transfer). ومع تزايد تعقيد التطبيقات وانتشار استخدام Microservices، يصبح التأكد من أمان هذه الخدمات أمرًا بالغ الأهمية.

من بين الجوانب الأساسية للأمان في خدمات الويب، يعتبر التفويض (Authorization) أحد الأعمدة الرئيسية التي تضمن أن المستخدمين لديهم الأذونات المناسبة للوصول إلى الموارد الحساسة أو تنفيذ عمليات معينة.
يوفر JAX-RS، وهو API قوي لتطوير خدمات الويب RESTful باستخدام جافا، أدوات وميزات متعددة لتطبيق وإدارة التفويض بطريقة مرنة وقابلة للتخصيص.

يهدف هذا البحث إلى تقديم نظرة شاملة على عملية التفويض في خدمات REST باستخدام JAX-RS. سنستعرض المفاهيم الأساسية للتفويض، الطرق المختلفة لتطبيق الأذونات باستخدام التعليقات التوضيحية (Annotations)، والفلاتر (Filters)، والمعترضات (Interceptors)، بالإضافة إلى مناقشة مزايا وعيوب استخدام JAX-RS في هذا السياق. من خلال هذا البحث، نسعى لتوفير فهم متكامل لأفضل الممارسات والاستراتيجيات التي يمكن اعتمادها لضمان تأمين خدمات الويب RESTful بفاعلية.

02 ماهو التفويض (Authorization) :

في بيئة الويب و خدمات الويب RESTful ، التفويض له أهمية حيوية. فمن خلاله، تتمكن المنصات من التحكم في الوصول إلى الموارد والبيانات الحساسة، وضمان أن يتم التعامل معها فقط من قبل الأشخاص والكيانات المخولة. هذا يساعد على منع الاختراقات، وحماية الخصوصية، وتحقيق التوازن بين الأمن والسهولة في الاستخدام

ويعتبر Authorization احد المفاهيم الاساسية التي تضمن الامن والحماية. إنه الآلية التي من خلالها يتم تحديد وتنظيم ما يُسمح للمستخدمين الوصول إليه أو القيام به، مما يضمن الأمن والحماية للموارد والبيانات الحساسة.

التفويض (Authorization)

هو عملية تحديد ما إذا كان للمستخدم الحق في الوصول إلى مورد معين أو إجراء عملية معينة.

تشمل المفاهيم الأساسية لـ(Authorization) تخصيص الأدوار للمستخدمين، منح الأذونات للوصول إلى موارد محددة، وتحديد السياسات التي تنظم هذه الأذونات.

كيف يمكن تنفيد(Authorization) ب أستخدام JAX-RS ؟؟

نحتاج في بداية الامر لتعريف ماذا المقصود ب JAX-RS

03 ماهو JAX-RS:

مكتبة من جافا لتطوير خدمات ويب RESTful. تقدم مجموعة من التعليقات التوضيحية (annotations) لتبسيط بناء وإدارة خدمات الويب.

كاتعريف اخر : هو إطار عمل يساعد في بناء خدمات الويب RESTful باستخدام جافا.

04 تطبيق Authorization في JAX-RS :

يمكن تنفيد التفويض في REST ويب بأستخدام JAX-RS عن طريق تقنيات الاتية :-

- Annotation استخدام التعليقات التوضيحية لتحديد الأدوار والأذونات المطلوبة للوصول إلى موارد معينة.
- Filters استخدام الفلاتر للتحقق من الأذونات قبل تنفيذ الموارد.
- Interceptors استخدام المعترضات لمعالجة طلبات HTTP قبل وبعد استدعاء الموارد.

- تطبيق Authorization باستخدام Annotations

Annotations شائعة هيا :

@RolesAllowed	تحدد الأدوار المسموح لها بالوصول إلى مورد معين.
@PermitAll	تسمح للجميع بالوصول إلى المورد.
@DenyAll	تمنع الجميع من الوصول إلى المورد.

- تطبيق Authorization باستخدام Filters :

الهدف الرئيسي من استخدام (Filters) في عملية (Authorization) هو ضمان أن الطلبات الموجهة للخادم تمتلك الأذونات الصحيحة للوصول إلى الموارد المطلوبة. الفلاتر تساعد في اعتراض الطلبات وفحص الأذونات قبل السماح بالوصول إلى الموارد المحمية.

هناك أنواع رئيسية ل Filters :-

1. فلاتر الطلب (Request Filters)

- **ContainerRequestFilter**

يقوم بأعتراض وينفذ قبل تسليم الطلب إلى مورد JAX-RS.

الشكل العام له :-

filter(ContainerRequestContext)

2. فلاتر الاستجابة (Response Filters)

- **ContainerResponseFilter**

يعترض وينفذ بعد معالجة الطلب وقبل إرسال الاستجابة إلى العميل.

الشكل العام له :-

filter(ContainerRequestContext , ContainerResponseContext)

خطوات تنفيذ عملية التفويض باستخدام الفلاتر (Filters) في JAX-RS:

1. إنشاء الفلتر (Filter):

تقوم بإنشاء فلتر ينفذ واجهة " ContainerRequestFilter".

في هذه الفلتر، يتم فحص الطلبات قبل وصولها إلى الموارد

2. استخراج بيانات التفويض (Authorization):

يتم استخراج البيانات من الطلب مثل الرموز المميزة (tokens) أو بيانات التوثيق الأخرى. عادةً ما يتم الحصول على هذه البيانات من **Header Authorization.**

3. التحقق من الأذونات :

- يتم التحقق من الأدوار أو الصلاحيات الممنوحة للمستخدم بناءً على البيانات المستخرجة.
- يتم مقارنة الأذونات المطلوبة للوصول إلى المورد مع الأذونات المتاحة للمستخدم.

4. منع الوصول غير المصرح به :

في حال عدم تطابق الأذونات، يتم منع الوصول عن طريق إرجاع استجابة غير مصرح به

أو(Response.Status.UNAUTHORIZED)

(Response.Status.FORBIDDEN).

5. السماح بالوصول المصرح به:

في حال تطابق الأذونات، يتم السماح بالوصول إلى المورد المطلوب.

باستخدام هذه الخطوات، يمكننا تنفيذ عملية التفويض بطريقة فعالة وآمنة في خدمات الويب RESTful باستخدام JAX-RS.

- تطبيق Authorization باستخدام Interceptors :

في JAX-RS، يتم استخدام الـ Interceptors لمعالجة الطلبات والاستجابات عند نقاط محددة في دورة حياتها، مما يتيح تعديل محتوى الكيانات (entities) قبل قراءتها أو كتابتها. يمكن استخدام Interceptors لتطبيق التفاوض (Authorization) لتحسين الأمان والأداء، وأيضا لتعديل البيانات بطرق معينة مثل التشفير أو فك التشفير.

أنواع Interceptors:

- ReaderInterceptor:
  يعترض ويعدل الكيانات قبل قراءتها من الطلب .
- WriterInterceptor:
  يعترض ويعدل الكيانات قبل كتابتها إلى الاستجابة .

خطوات تنفيذ التفاوض باستخدام Interceptors:

1. إنشاء ReaderInterceptor

هذا الـ Interceptor يعترض الطلبات ويعدل محتوى الكيانات قبل قراءتها.

2. إنشاء WriterInterceptor

هذا الـ Interceptor يعترض الاستجابات ويعدل محتوى الكيانات قبل كتابتها.

3. تسجيل الـ Interceptors

لجعل الـ Interceptors تعمل، يجب تسجيلها في تطبيق JAX-RS الخاص بالمستخدم . يمكن القيام بذلك باستخدام التعليقات التوضيحية "@Provider"، أو يمكن تسجيلها يدوياً في فئة التكوين

العمليات التي تقوم بها Interceptors :

- التحكم في البيانات : التحقق وتعديل البيانات قبل الوصول إلى الموارد أو إرسالها للعميل.
- التشفير وفك التشفير : تشفير البيانات عند الإرسال وفك تشفيرها عند الاستقبال لتحسين الأمان.
- تسجيل الطلبات والاستجابات : تسجيل محتوى الطلبات والاستجابات للتصحيح أو التحليل.
- إضافة رؤوس : تعديل رؤوس HTTP قبل إرسال الاستجابة، مثل نوع المحتوى أو إعدادات التخزين المؤقت.

الخلاصة :

باستخدام JAX-RS Filters وInterceptors، يمكننا تحسين أمان وفعالية تطبيقات الويب من خلال التحقق، التعديل، التشفير، وتسجيل البيانات بمرونة عالية. هذه الأدوات تتيح لنا التحكم الكامل في دورة حياة الطلبات والاستجابات، مما يعزز من قدرة التطبيقات على تلبية متطلبات الأمان والأداء المتقدمة.

أيضا هناك اليات اخري لتطبيق عملية التفويض منها :-

- سياسات الأمان التقريرية (Declarative Security):

يمكن استخدام ملفات التكوين مثل web.xml لتحديد الأذونات والأدوار بشكل تقريري.

- التفويض باستخدام OAuth 2.0 وJWT:-

استخدام OAuth 2.0 للسماح بالتفويض الخارجي عبر رموز الوصول (Access Tokens).

أيضا JWT (JSON Web Tokens) :

يتم استخدام JWT للتحقق من الأذونات في طلبات HTTP

- الأمان التقدمي (Progressive Security):-

يثمتل في تنفيد مستويات مختلفة من الأمان بناءً على حساسية الموارد ومتطلبات الأذونات.

05 مميزات وعيوب :-

تطبيق Authorization في REST باستخدام JAX-RS يوفر العديد من المزايا ولكن يأتي أيضًا مع بعض العيوب. فيما يلي مميزات وعيوب استخدام JAX-RS للتفويض في خدمات REST:

المميزات :-

1- تكامل جيد مع Java EE وJakarta EE :

JAX-RS هو جزء من مواصفات Java EE (والآن Jakarta EE)، مما يجعله متكاملاً بشكل جيد مع مكونات Java الأخرى مثل EJB وJPA

2 . تعليقات توضيحية بسيطة ومباشرة :

JAX-RS يوفر تعليقات توضيحية مثل @PermitAll ، @RolesAllowed، والDenyAll@ التي تسهل تطبيق الأذونات على مستوى الخدمة أو الموارد.

3. قابلية التمدد باستخدام Filters وInterceptors

4. قابلية التخصيص:

يمكنك تخصيص منطق التفويض بناءً على احتياجات التطبيق

5. دعم التكامل مع أطر أخرى:

JAX-RS يمكن دمجه مع أطر عمل أخرى مثل Spring Security لتوفير طبقة إضافية من الحماية والأذونات.

6. إدارة الأدوار والأذونات بشكل مركزي:

باستخدام ملفات التكوين مثل web.xml أو application.properties، يمكنك إدارة الأدوار والأذونات بشكل مركزي ومرن.

العيوب :-

1. تعقيد الإدارة مع زيادة عدد الميكروسيرفيسز:

مع زيادة عدد الخدمات المصغرة، قد يصبح من الصعب إدارة الأذونات والأدوار عبر جميع الخدمات بشكل مركزي.

2. تحديات التحديث والصيانة

3. توافق التعليقات التوضيحية

4. الأمان:

استخدام الأذونات المستندة إلى التعليقات التوضيحية فقط قد لا يكون كافيًا لتأمين التطبيق بالكامل. من الضروري دمج إجراءات أمان إضافية مثل التحقق من الرموز (tokens) وجلسات العمل (sessions).

06 الخاتمة :-

استخدام التفويض في REST باستخدام JAX-RS يوفر وسيلة قوية ومرنة للتحكم في الوصول إلى الموارد والبيانات الحساسة. من خلال الجمع بين الأدوار والأذونات، يمكن للتطبيقات تحديد وضبط المستخدمين المسموح لهم بالتفاعل مع خدمات معينة وإجراء عمليات محددة. تطبيق الفلاتر والمُعترضات (Interceptors) يسهل التحقق من الصلاحيات وتطبيق السياسات الأمنية اللازمة. هذا النهج يعزز من أمان التطبيق ويضمن أن البيانات والوظائف متاحة فقط للمستخدمين المصرح لهم.

07 المراجع :-

- https://developer.mozilla.org/
- https://stackoverflow.com/
- https://chatgpt.com/