JWT SECURITY SPRING

سالم المطيري

2024 December 23

JWT مع Security Spring

JWT (Token Web JSON) هو آلية مصادقة وتفويض حديثة وآمنة تُستخدم بشكل خاص في التطبيقات عديمة الحالة (Stateless). يتميز ببنية بسيطة تتكون من 3 أجزاء:

- Header: يحتوي نوع التوكن (JWT) وخوارزمية التشفير (مثل HS256).
- Payload: يخزن بيانات المستخدم (مثل معرف المستخدم، الصلاحيات، تاريخ انتهاء الصلاحية).
- Signature: يضمن عدم تعديل التوكن باستخدام المفتاح السري أو مفتاح عام/خاص.

كيف يعمل JWT مع Spring Security؟

١. المستخدم يُدخل بياناته (اسم مستخدم وكلمة مرور).

٢. الخادم يتحقق ويُنشئ JWT يحتوي على بيانات المستخدم.

٣. يتم إرسال التوكن إلى العميل وتخزينه (عادةً في LocalStorage أو كوكيز آمنة).

٤. يُرسل العميل التوكن مع كل طلب في الهيدر (Authorization: Bearer <token>).

٥. يتحقق الخادم من صحة التوكن وصلاحيته قبل معالجة الطلب.

المزايا

- عديم الحالة (Stateless): لا حاجة لتخزين الجلسات على الخادم، مما يحسن الأداء ويزيد من القابلية للتوسع.
- صغير الحجم: سهل الإرسال عبر الشبكات.
- آمن: يعتمد على التوقيع لضمان صحة التوكن.
- متعدد المنصات: يعمل مع أي لغة برمجة.

الاعتبارات الأمنية

- استخدم :HTTPS لحماية التوكن أثناء النقل.
- حدد وقت انتهاء التوكن: لتقليل المخاطر في حالة السرقة.
- تجنب تخزين التوكن في LocalStorage إن أمكن: لتجنب هجمات XSS.
- إعداد قائمة سوداء :(Blacklist) لإبطال التوكنات عند الحاجة.

التحديات والحلول

التحدي	الحل
صعوبة إلغاء التوكن	استخدم قائمة سوداء لإبطال التوكنات.
حساسية المفتاح السري	قم بتحديث المفاتيح بانتظام وحمايتها.
حجم التوكن الكبير	احتفظ فقط بالمعلومات الضرورية.

الخلاصة

JWT هو أداة قوية لتأمين التطبيقات الحديثة، خاصة تلك التي تعتمد على APIs. RESTful عند الالتزام بأفضل الممارسات، يحقق توازنًا مثاليًا بين الأمان والأداء.