Java adalah salah satu bahasa pemrograman yang paling banyak digunakan saat ini bersaing dengan Python dan JavaScript.

Fitur baru JAVA 19

Bagian ini menjelaskan beberapa peningkatan di Java SE 19 dan JDK 19. Dalam beberapa kasus, deskripsi menyediakan tautan ke informasi detail tambahan tentang masalah atau perubahan. API yang dijelaskan di sini disediakan dengan Oracle JDK. Ini mencakup implementasi lengkap Platform Java SE 19 dan API Java tambahan untuk mendukung pengembangan, debugging, dan pemantauan aplikasi Java. Sumber informasi lain tentang peningkatan penting dan fitur baru di Java SE 19 dan JDK 19 adalah Java SE 19 (JSR) Spesifikasi Platform, yang mendokumentasikan perubahan spesifikasi yang dibuat antara Java SE 17 dan Java SE 19. Dokumen ini mencakup deskripsi fitur baru dan peningkatan yang juga merupakan perubahan spesifikasi. Deskripsi juga mengidentifikasi potensi masalah kompatibilitas yang mungkin dialami saat bermigrasi ke JDK 19.

core-libs/java.lang

* Mendukung Unicode 14.0. Rilis ini meningkatkan dukungan Unicode ke 14.0, yang mencakup hal berikut: Kelas java.lang.Charactermendukung Basis Data Karakter Unicode level 14.0, yang menambahkan 838 karakter, dengan total 144.697 karakter. Penambahan ini termasuk 5 skrip baru, dengan total 159 skrip, serta 37 karakter emoji baru. Kelas java.text.Bididan java.text.Normalizer mendukung tingkat 14.0 dari Unicode Standard Annexes, #9 dan #15, masing-masing. Paket ini java.util.regexmendukung Extended Grapheme Clusters berdasarkan level 14.0 Unicode Standard Annex #29 Untuk detail lebih lanjut tentang Unicode 14.0, lihat catatan rilis Konsorsium Unicode .

core-libs/java.lang

* Properti sistem baru untuk System.out dan System.err

Dua properti sistem baru, stdout.encoding dan stderr.encoding, telah ditambahkan. Nilai dari properti sistem ini adalah pengkodean yang digunakan oleh output standar dan aliran kesalahan standar ( System.out dan System.err). Nilai default dari properti sistem ini bergantung pada platform. Nilai mengambil nilai native.encodingproperti saat platform tidak menyediakan aliran untuk konsol. Properti dapat diganti pada opsi baris perintah peluncur (dengan -D) untuk mengaturnya ke UTF-8 tempat yang diperlukan.

core-libs/java.net

* Dukungan Pengikatan Saluran HTTPS untuk Java GSS/Kerberos( JDK- 8279842 ) Dukungan telah ditambahkan untuk token pengikatan saluran TLS untuk otentikasi Negosiasi/Kerberos melalui HTTPS melalui javax.net.HttpsURLConnection. Token pengikatan saluran semakin dibutuhkan sebagai bentuk keamanan yang ditingkatkan. Mereka bekerja dengan berkomunikasi dari klien ke server pemahaman klien tentang pengikatan antara keamanan koneksi, seperti yang diwakili oleh sertifikat server TLS, dan kredensial otentikasi tingkat yang lebih tinggi, seperti nama pengguna dan kata sandi. Server kemudian dapat mendeteksi jika klien telah tertipu oleh MITM dan mematikan sesi atau koneksi. Fitur ini dikendalikan melalui properti sistem baru jdk.https.negotiate.cbt yang dijelaskan sepenuhnya di Networking Properties .

core-libs/java.time

* Format Tanggal-Waktu Tambahan( JDK- 8176706 )

Format tanggal/waktu tambahan sekarang diperkenalkan di java.time.format.DateTimeFormatter/DateTimeFormatterBuilderkelas. Dalam rilis sebelumnya, hanya 4 gaya yang telah ditentukan sebelumnya, yaitu FormatStyle.FULL/LONG/MEDIUM/SHORTtersedia. Sekarang pengguna dapat menentukan gaya fleksibel mereka sendiri dengan DateTimeFormatter.ofLocalizedPattern(String requestedTemplate) metode baru ini. Sebagai contoh, DateTimeFormatter.ofLocalizedPattern("yMMM") menghasilkan formatter, yang dapat memformat tanggal dengan cara lokal, seperti "Feb 2022" di US lokal. Metode pendukung DateTimeFormatterBuilder.appendLocalized(String requestedTemplate) juga disediakan.

core-libs/Java.util:collections

* Metode Baru untuk Membuat HashMaps dan HashSet yang Dialokasikan Sebelumnya ( JDK- 8186958 )

Metode pabrik statis baru telah diperkenalkan untuk memungkinkan pembuatan HashMap dan instance terkait yang telah dialokasikan sebelumnya untuk mengakomodasi jumlah pemetaan atau elemen yang diharapkan. Setelah menggunakan HashMap.newHashMapmetode ini, jumlah pemetaan yang diminta dapat ditambahkan ke yang baru dibuat HashMap tanpa mengubah ukurannya. Ada metode pabrik statis baru yang serupa untuk LinkedHashMap, WeakHashMap, HashSet, dan LinkedHashSet. Kumpulan lengkap metode baru adalah:

HashMap.newHashMap

LinkedHashMap.newLinkedHashMap

WeakHashMap.newWeakHashMap

HashSet.newHashSet

LinkedHashSet.newLinkedHashSet

Konstruktor int-argument untuk kelas-kelas ini mengatur "kapasitas" (ukuran tabel internal) yang tidak sama dengan jumlah elemen yang dapat ditampung. Kapasitas terkait dengan jumlah elemen dengan perhitungan sederhana namun rawan kesalahan. Untuk alasan ini, program harus menggunakan metode pabrik statis baru ini daripada konstruktor int-argument.

hotspot/kompiler

* Dukungan untuk Perlindungan PAC-RET di Linux/AArch64 ( JDK -8277204)

Dukungan untuk perlindungan PAC-RET pada platform Linux/AArch64 telah diperkenalkan. Saat diaktifkan, OpenJDK akan menggunakan fitur perangkat keras dari ekstensi ARMv8.3 Pointer Authentication Code (PAC) untuk melindungi dari serangan Return Orientated Programming (ROP). Untuk informasi selengkapnya tentang ekstensi PAC, lihat "Menyediakan perlindungan untuk perangkat lunak kompleks" atau bagian "Otentikasi penunjuk dalam status AArch64" di Arm ARM . Untuk memanfaatkan fitur ini, OpenJDK pertama harus dibangun dengan flag konfigurasi --enable-branch-protectionmenggunakan GCC 9.1.0+ atau LLVM 10+ . Kemudian, flag runtime -XX:UseBranchProtection=standardakan mengaktifkan perlindungan PAC-RET jika sistem mendukungnya dan javabiner dikompilasi dengan perlindungan cabang diaktifkan; jika tidak, bendera diabaikan secara diam-diam. Atau, -XX:UseBranchProtection=pac-retjuga akan mengaktifkan perlindungan PAC-RET, tetapi dalam kasus ini jika sistem tidak mendukungnya atau javabiner tidak dikompilasi dengan perlindungan cabang diaktifkan, maka peringatan akan dicetak.

hotspot/waktu proses

* Pembuatan Arsip CDS Otomatis( JDK- 8261455 )

Opsi JVM -XX:+AutoCreateSharedArchive dapat digunakan untuk secara otomatis membuat atau memperbarui arsip CDS untuk suatu aplikasi. Sebagai contoh:

java -XX:+AutoCreateSharedArchive -XX:SharedArchiveFile=app.jsa -cp app.jar App

Arsip yang ditentukan akan ditulis jika tidak ada, atau jika dibuat oleh versi JDK yang berbeda

security-libs/java.security

* Windows KeyStore Diperbarui untuk Menyertakan Akses ke Lokasi Mesin Lokal( JDK-6782021)

Dukungan Windows KeyStore di penyedia SunMSCAPI telah diperluas untuk menyertakan akses ke lokasi mesin lokal. Jenis keystore baru adalah:

"Windows-MY-LOCALMACHINE"

"Windows-ROOT-LOCALMACHINE"

Jenis keystore berikut juga ditambahkan, memungkinkan pengembang untuk memperjelas bahwa mereka memetakan ke pengguna saat ini:

"Windows-MY-CURRENTUSER" (sama seperti "Windows-MY")

"Windows-ROOT-CURRENTUSER" (sama seperti "Windows-ROOT")

security-libs/java.security

* Pisahkan SEQUENCE di X509Certificate::getSubjectAlternativeNames dan X509Certificate::getIssuerAlternativeNames di otherName( JDK-8277976)

Implementasi JDK dari X509Certificate::getSubjectAlternativeNames dan X509Certificate::getIssuerAlternativeNames telah ditingkatkan untuk mengembalikan bidang type-id dan tambahan dari file . Bidang dikembalikan sebagai String jika dikodekan sebagai string karakter atau sebagai array byte, yang berguna karena menghindari harus mengurai bentuk nama yang disandikan ASN.1 DER.valueotherNamevalue

security-libs/javax.net.ssl

* Skema Tanda Tangan TLS( JDK- 8280494 )

Java SE API baru, javax.net.ssl.SSLParameters.getSignatureSchemes()dan javax.net.ssl.SSLParameters.setSignatureSchemes(), telah ditambahkan untuk memungkinkan aplikasi menyesuaikan skema tanda tangan yang digunakan dalam koneksi TLS atau DTLS individual. Perhatikan bahwa penyedia yang mendasarinya dapat menentukan skema tanda tangan default untuk setiap koneksi TLS atau DTLS. Aplikasi juga dapat menggunakan properti sistem "jdk.tls.client.SignatureSchemes" dan/atau "jdk.tls.server.SignatureSchemes" yang ada untuk menyesuaikan skema tanda tangan default khusus penyedia. Jika tidak null, skema tanda tangan yang diteruskan ke setSignatureSchemes()metode akan menggantikan skema tanda tangan default untuk koneksi TLS atau DTLS yang ditentukan. Perhatikan bahwa penyedia mungkin belum diperbarui untuk mendukung API baru dan dalam hal ini mungkin mengabaikan skema tanda tangan yang ditetapkan. Penyedia JDK SunJSSEmendukung metode ini. Disarankan agar penyedia pihak ketiga menambahkan dukungan untuk metode ini saat mereka menambahkan dukungan untuk JDK 19 atau rilis yang lebih baru.

security-libs/jdk.security

* Tambahkan Opsi -providerPath ke jarsigner( JDK-8281175)

Opsi baru -providerPathtelah ditambahkan ke jarsigner. Seseorang dapat menggunakan opsi ini untuk menentukan jalur kelas dari implementasi keystore alternatif. Ini dapat digunakan bersama dengan -providerClassopsi.

security-libs/org.ietf.jgss:krb5

* Opsi Baru untuk ktab untuk Menyediakan Garam Non-default( JDK-8279064)

Dua opsi baru ditambahkan ke ktabperintah saat menambahkan entri keytab baru. Saat ktab -a username password -s altsaltdipanggil, altsaltdigunakan sebagai pengganti garam default. Saat ktab -a username password -fdipanggil, alat akan menghubungi KDC untuk mengambil garam yang sebenarnya digunakan.

xml/jaxp

* Batas Pemrosesan XML Baru (JDK- 8270504 (bukan publik))

Tiga batas pemrosesan telah ditambahkan ke pustaka XML. Ini adalah:

jdk.xml.xpathExprGrpLimit

Deskripsi: Membatasi jumlah grup yang dapat berisi ekspresi XPath.

Jenis: bilangan bulat

Nilai: Sebuah bilangan bulat positif. Nilai kurang dari atau sama dengan 0 menunjukkan tidak ada batas. Jika nilainya bukan bilangan bulat, a NumberFormatExceptiondilemparkan. Standar 10.

jdk.xml.xpathExprOpLimit

Deskripsi: Membatasi jumlah operator yang dapat dikandung oleh ekspresi XPath.

Jenis: bilangan bulat

Nilai: Sebuah bilangan bulat positif. Nilai kurang dari atau sama dengan 0 menunjukkan tidak ada batas. Jika nilainya bukan bilangan bulat, a NumberFormatExceptiondilemparkan. Standar 100.

jdk.xml.xpathTotalOpLimit

Deskripsi: Membatasi jumlah total operator XPath dalam XSL Stylesheet.

Jenis: bilangan bulat

Nilai: Sebuah bilangan bulat positif. Nilai kurang dari atau sama dengan 0 menunjukkan tidak ada batas. Jika nilainya bukan bilangan bulat, a NumberFormatExceptiondilemparkan. Standar 10000.

* Prosesor yang didukung

jdk.xml.xpathExprGrpLimitdan jdk.xml.xpathExprOpLimitdidukung oleh prosesor XPath. Ketiga batasan tersebut didukung oleh prosesor XSLT.

* Mengatur property

Untuk prosesor XSLT, properti dapat diubah melalui file TransformerFactory. Sebagai contoh,

TransformerFactory factory = TransformerFactory.newInstance();

factory.setAttribute("jdk.xml.xpathTotalOpLimit", "1000");

Untuk prosesor XPath, properti dapat diubah melalui file XPathFactory. Sebagai contoh :

XPathFactory xf = XPathFactory.newInstance();

xf.setProperty("jdk.xml.xpathExprGrpLimit", "20");

Untuk prosesor XPath dan XSLT, properti dapat diatur melalui properti sistem dan jaxp.propertiesfile konfigurasi yang terletak di confdirektori instalasi Java. Sebagai contoh

System.setProperty("jdk.xml.xpathExprGrpLimit", "20");

atau dalam jaxp.properties berkas, jdk.xml.xpathExprGrpLimit=20