

UUID

JAVA

UUID는 "Universally Unique Identifier"의 약자로, 범용 고유 식별자를 의미한다.

이는 일반적으로 소프트웨어에서 객체를 고유하게 식별하는 데 사용되는데 Java에서는 `java.util.UUID` 클래스를 통해 UUID를 생성하고 조작할 수 있다.

UUID는 128비트 숫자로 구성되어 있으며, 16진수로 표현되고 16비트씩 8개의 그룹으로 나누어져 있다.

예시

```
550e8400-e29b-41d4-a716-446655440000
```

여기에서 첫 번째 그룹은 32비트, 나머지 7개의 그룹은 16비트씩 구성되는데 이러한 구성은 UUID를 보다 쉽게 다룰 수 있도록 도와준다.

UUID는 생성 시스템의 시간, MAC 주소 및 무작위 수 등을 통해 랜덤하게 생성 되고 이로 인해 UUID는 매우 높은 확률로 고유하게 생성된다.

추가

시간 정보를 기반으로 하는 UUID의 경우, 생성 순서도 유지된다.

Java에서 UUID 생성

```
import java.util.UUID;

UUID uuid = UUID.randomUUID();
System.out.println(uuid.toString());
```

- UUID 클래스를 import
- `randomUUID()` 메소드를 호출하여 UUID 객체 생성
- `toString()` 메소드를 호출하여 UUID를 문자열로 변환하여 출력

UUID는 데이터베이스의 고유 식별자, 웹 서비스의 세션 식별자 등 다양한 용도로 사용된다고 한다.

Spring - UUID 내부

@NonNls

```
public final class UUID implements java.io.Serializable, Comparable<UUID> {
```

Explicit serialVersionUID for interoperability.

```
private static final long serialVersionUID = -4856846361193249489L;
```

Static factory to retrieve a type 4 (pseudo randomly generated) UUID. The UUID is generated using a cryptographically strong pseudo random number generator.

Returns: A randomly generated UUID

```
public static UUID randomUUID() {
    SecureRandom ng = Holder.numberGenerator;

    byte[] randomBytes = new byte[16];
    ng.nextBytes(randomBytes);
    randomBytes[6] &= 0x0f; /* clear version */
    randomBytes[6] |= 0x40; /* set to version 4 */
    randomBytes[8] &= 0x3f; /* clear variant */
    randomBytes[8] |= 0x80; /* set to IETF variant */
    return new UUID(randomBytes);
}
```

```
public int compareTo(UUID val) {
    // The ordering is intentionally set up so that the UUIDs
    // can simply be numerically compared as two numbers
    return (this.mostSigBits < val.mostSigBits ? -1 :
        (this.mostSigBits > val.mostSigBits ? 1 :
            (this.leastSigBits < val.leastSigBits ? -1 :
                (this.leastSigBits > val.leastSigBits ? 1 :
                    0))));
}
```