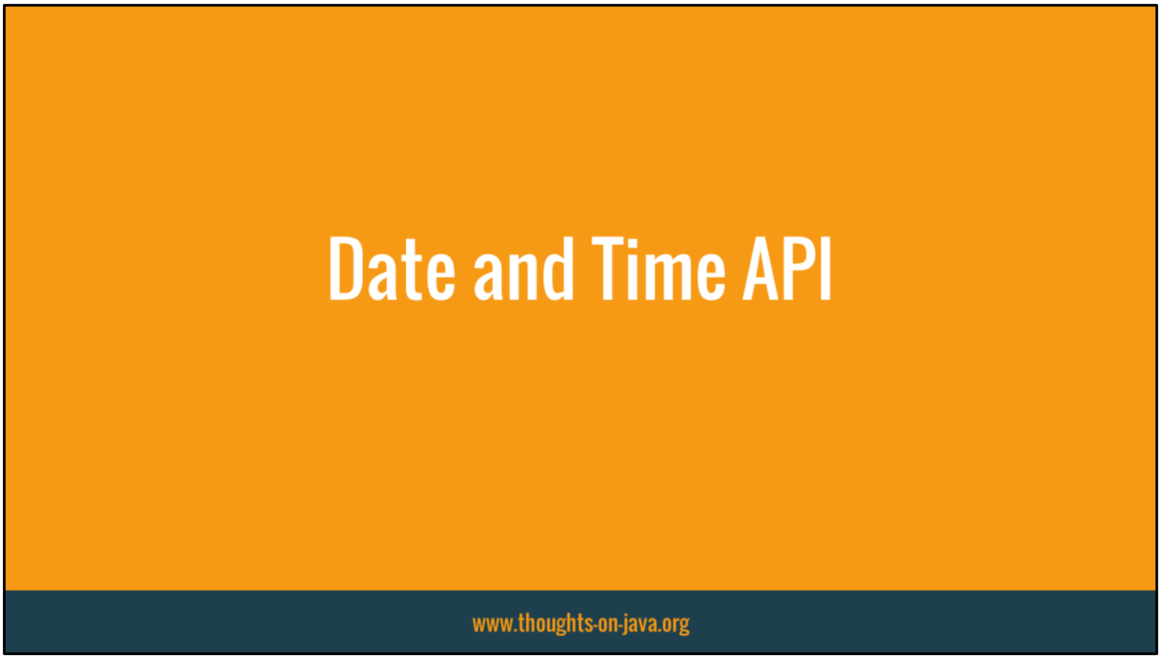
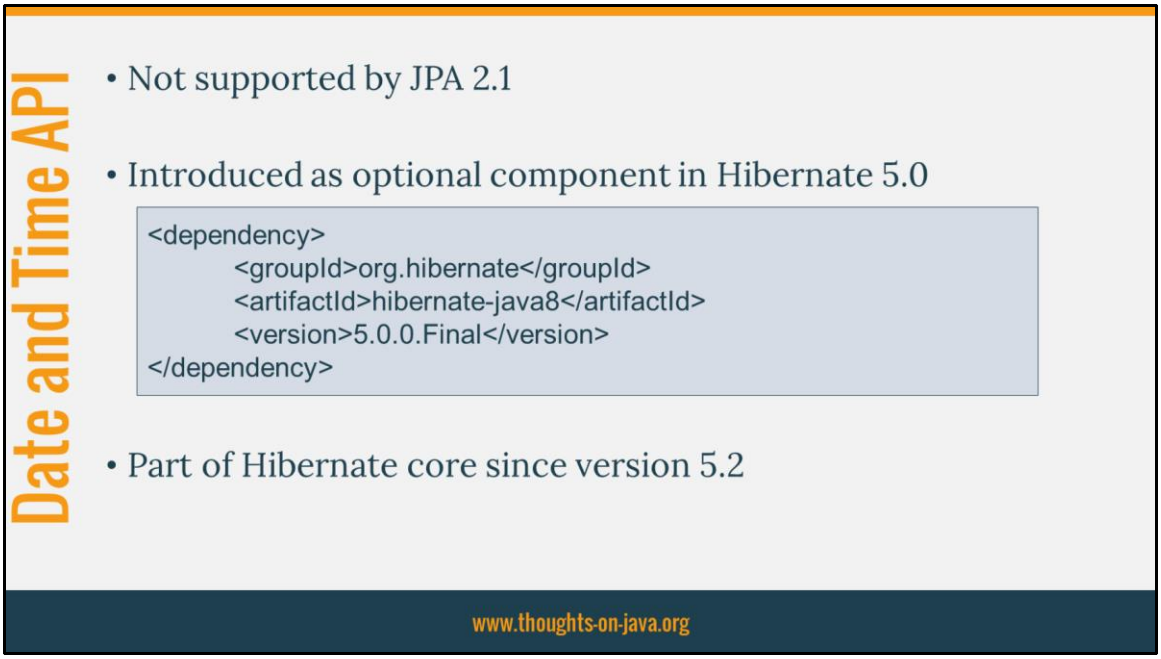


Date and Time API

www.thoughts-on-java.org

Java 8 finally introduced a new Date API. The old *java.util.Date* had a lot of flaws that often made it painful to use.
With the new Date and Time API, this is no longer the case. But unfortunately, JPA and older Hibernate versions don't provide direct support for it. You have to implement your own *AttributeConverter* to map the new classes to the correct data types. I show you how to do that in the *AttributeConverter* module.

Since the release of Hibernate 5.0, you don't need to do that anymore. It now provides proprietary support for the new classes of the Date and Time API.



Date and Time API

- Not supported by JPA 2.1
- Introduced as optional component in Hibernate 5.0

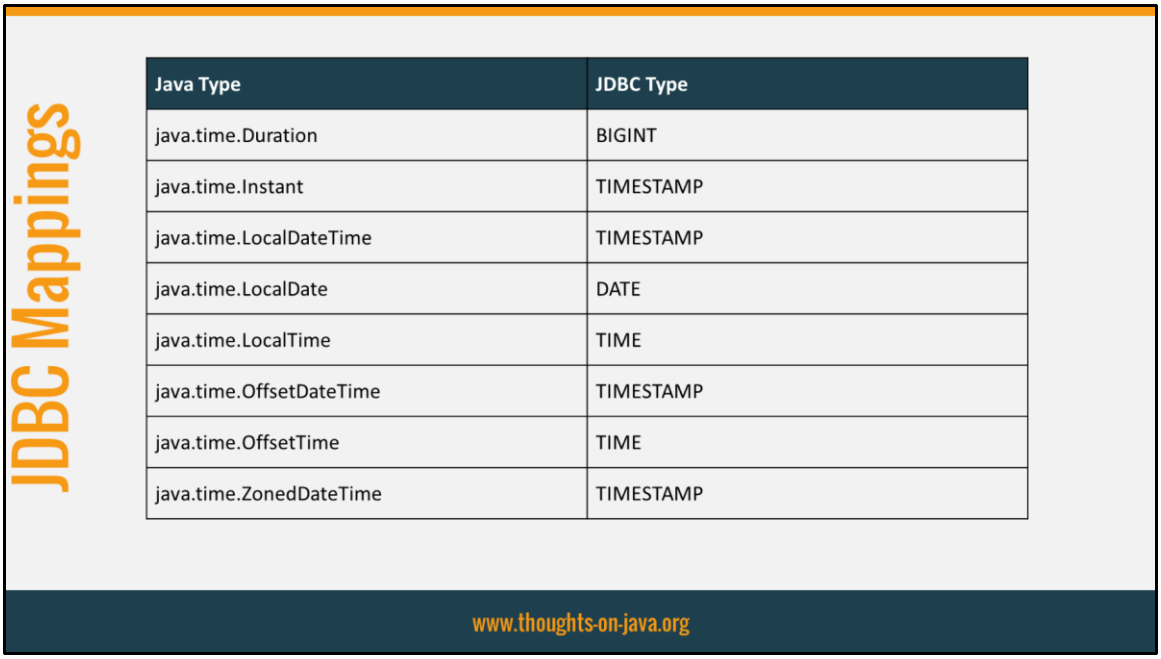
```
<dependency>
        <groupId>org.hibernate</groupId>
        <artifactId>hibernate-java8</artifactId>
        <version>5.0.0.Final</version>
</dependency>
```

- Part of Hibernate core since version 5.2

www.thoughts-on-java.org

The Hibernate core in version 5.0 was still based on Java 7 and didn’t provide any support for Java 8 classes. But the Hibernate team released the optional *hibernate-java8.jar* file which you need to add to your classpath.

Since version 5.2, the Hibernate core is based on Java 8 and the *hibernate-java8.jar* has become part of the Hibernate core.

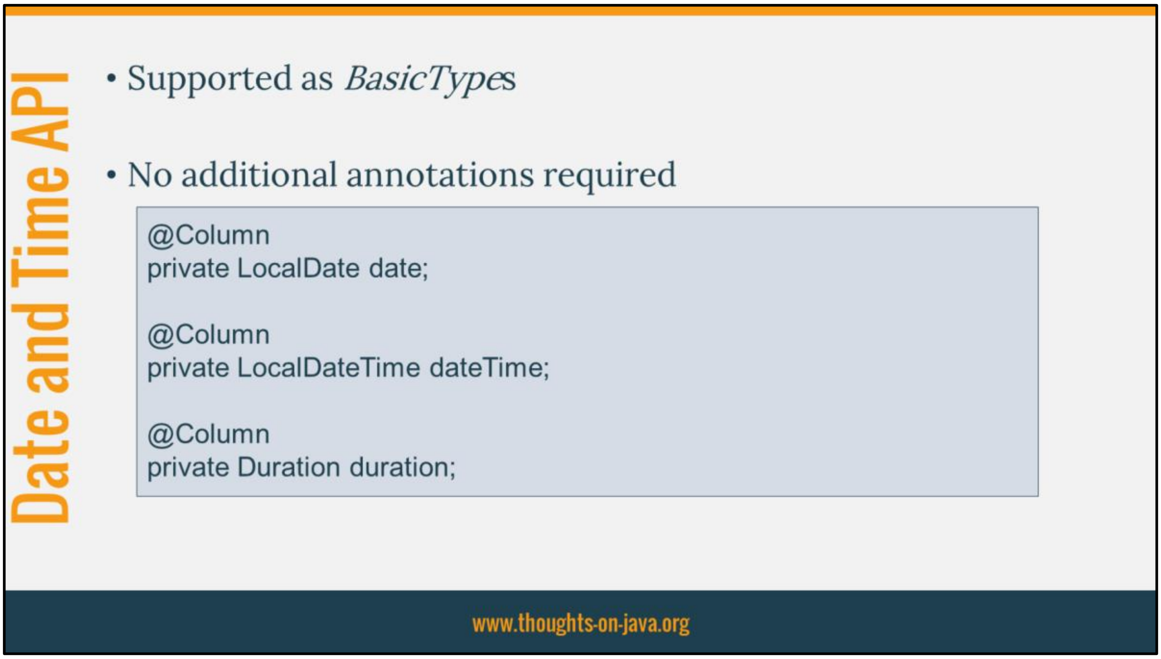


JDBC Mappings

Java Type	JDBC Type
java.time.Duration	BIGINT
java.time.Instant	TIMESTAMP
java.time.LocalDateTime	TIMESTAMP
java.time.LocalDate	DATE
java.time.LocalTime	TIME
java.time.OffsetDateTime	TIMESTAMP
java.time.OffsetTime	TIME
java.time.ZonedDateTime	TIMESTAMP

www.thoughts-on-java.org

Here you can see an overview of the supported classes of the Date and Time API and the JDBC types to which they get mapped by Hibernate.



Date and Time API

- Supported as *BasicTypes*

- No additional annotations required

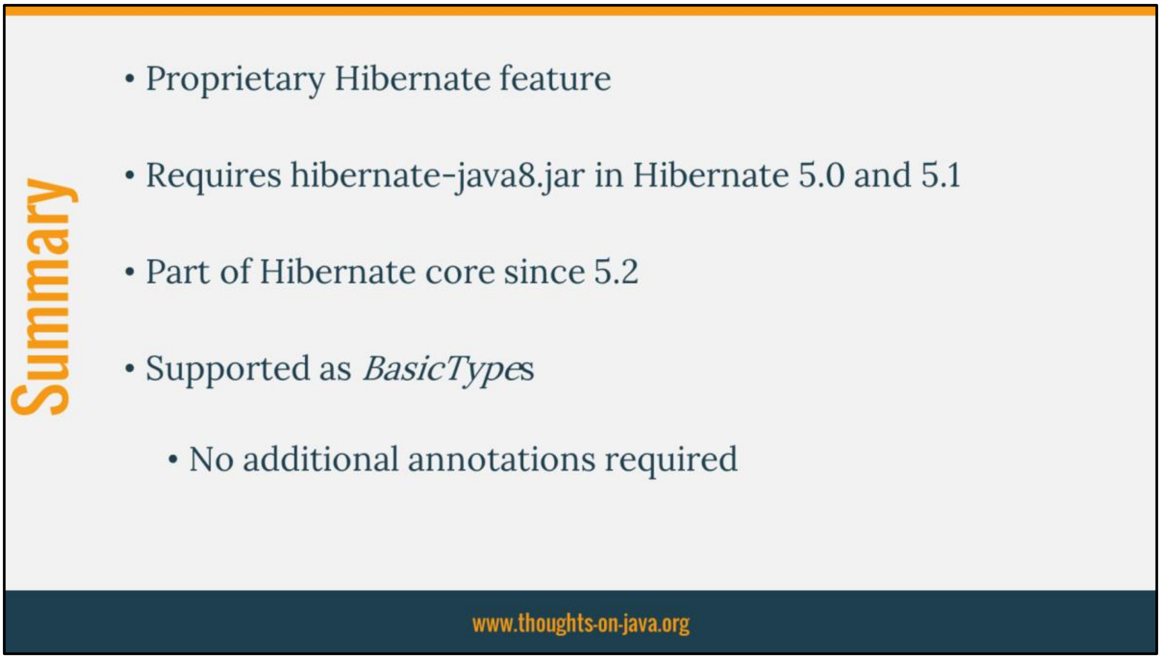
```
@Column
private LocalDate date;

@Column
private LocalDateTime dateTime;

@Column
private Duration duration;
```

www.thoughts-on-java.org

Hibernate supports the new classes as *BasicTypes*. The only thing you have to do is to map the entity attribute to a database column. You can see an example for *LocalDate*, *LocalDateTime* and *Duration* here on the slide. Hibernate doesn't require any additional annotations. That makes the new classes easier to use than the old *java.util.Date.*

Demo

www.thoughts-on-java.org



Summary

- Proprietary Hibernate feature
- Requires hibernate-java8.jar in Hibernate 5.0 and 5.1
- Part of Hibernate core since 5.2
- Supported as *BasicType*s
  - No additional annotations required

www.thoughts-on-java.org

Hibernate 5 added the support of the Date and Time API classes as a proprietary feature. The code base of the Hibernate core in version 5.0 and 5.1 still uses Java 7 and shipped the support for the new Java 8 classes in a separate jar file. This file got removed in version 5.2 when the Hibernate team switched their code base to Java 8. Hibernate supports the new classes as *BasicType*s which makes them easy to use. The mapping requires no additional annotations. The only thing you have to do is to map the entity attribute to a database column.

Exercises

www.thoughts-on-java.org