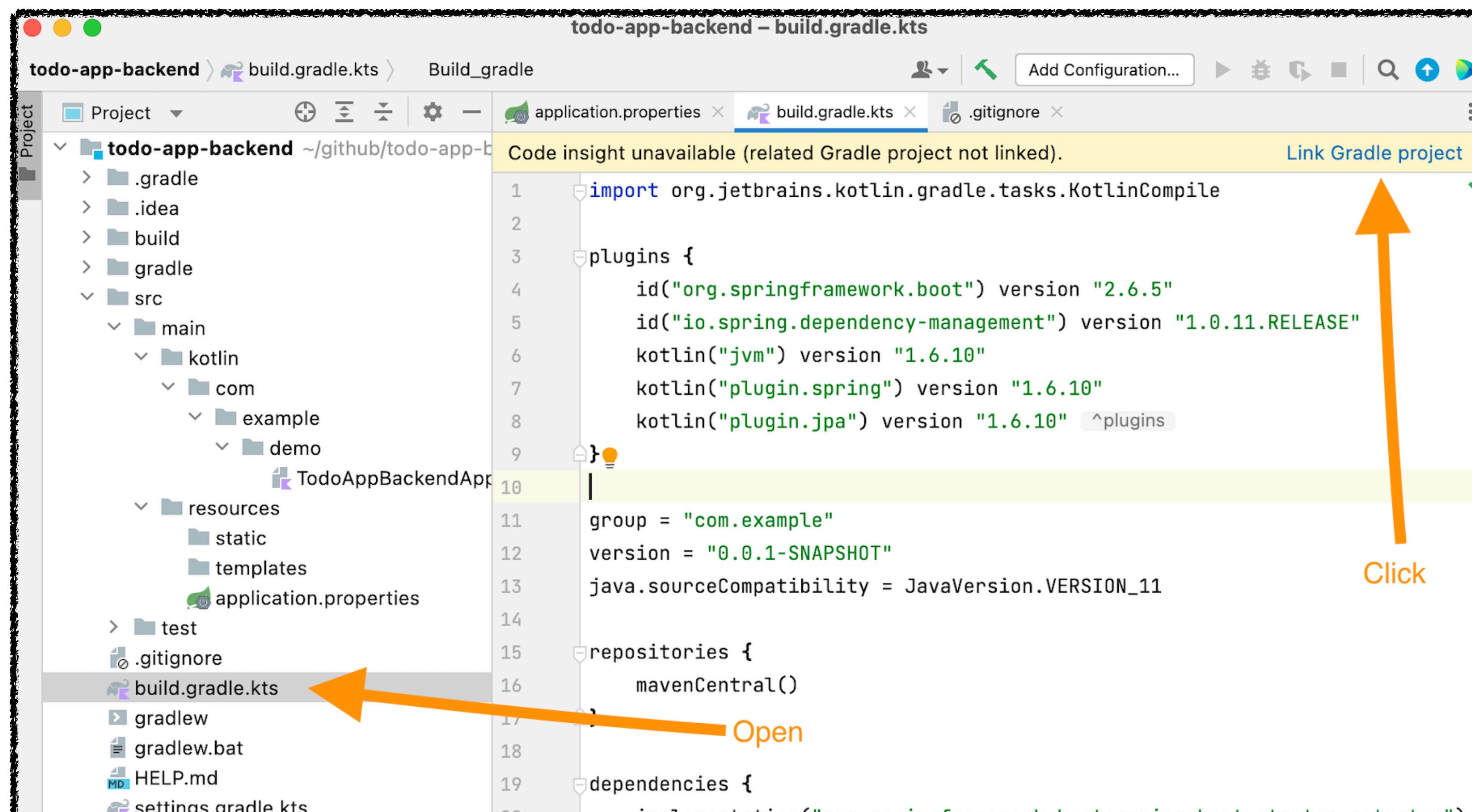


TODO APP Backend

Configure Database

Exercise2

- Check if Gradle project is correctly linked to IntelliJ



Configure Database

Exercise2

todo-app-backend – build.gradle.kts

todo-app-backend > build.gradle.kts > Build_gradle

application.properties build.gradle.kts .gitignore

Code insight unavailable (related Gradle project not linked). [Link Gradle project](#)

```
import org.jetbrains.kotlin.gradle.tasks.KotlinCompile
```

Project

todo-app-backend ~/github/todo-app-b

- .gradle
- .idea
- build
- gradle
- src
 - main
 - kotlin
 - com
 - example
 - demo
 - resources
 - static
 - templates
 - application.properties
 - test
 - .gitignore
- build.gradle.kts
- gradlew
- gradlew.bat
- HELP.md
- settings.gradle.kts
- todo-app-backend.iml

Favorites

- christian...
- Applicati...
- github
- Projects
- Desktop
- Screens...
- Documents
- Downloads

iCloud

- iCloud Dri...
- Shared

Locations

- OneDrive

Tags

Name Size Kind Date Added

Name	Size	Kind	Date Added
> build		-- Folder	Heute, 08:45
todo-app-backend.iml	382 bytes	Document	Heute, 08:44
> src		-- Folder	Heute, 08:44
settings.gradle.kts	26 bytes	Atom Document	Heute, 08:44
> gradle		-- Folder	Heute, 08:44
HELP.md	1 KB	Markdo...ument	Heute, 08:44
build.gradle.kts	1 KB	Atom Document	Heute, 08:44
gradlew	8 KB	Unix Ex...able File	Heute, 08:44
gradlew.bat	3 KB	Batch File	Heute, 08:44

New Folder Cancel Open

demo – build.gradle.kts (demo)

todo-app-backend > build.gradle.kts > Build_gradle

application.properties build.gradle.kts (demo) .gitignore

```
import org.jetbrains.kotlin.gradle.tasks.KotlinCompile
```

```
plugins { this: PluginDependenciesSpecScope
    id("org.springframework.boot") version "2.6.5"
    id("io.spring.dependency-management") version "1.0.11.RELEASE"
    kotlin(module: "jvm") version "1.6.10" Should see gradle tab now
```

Project

todo-app-backend [demo] ~/github/todo-app-b

- External Libraries
- Scratches and Consoles

Gradle

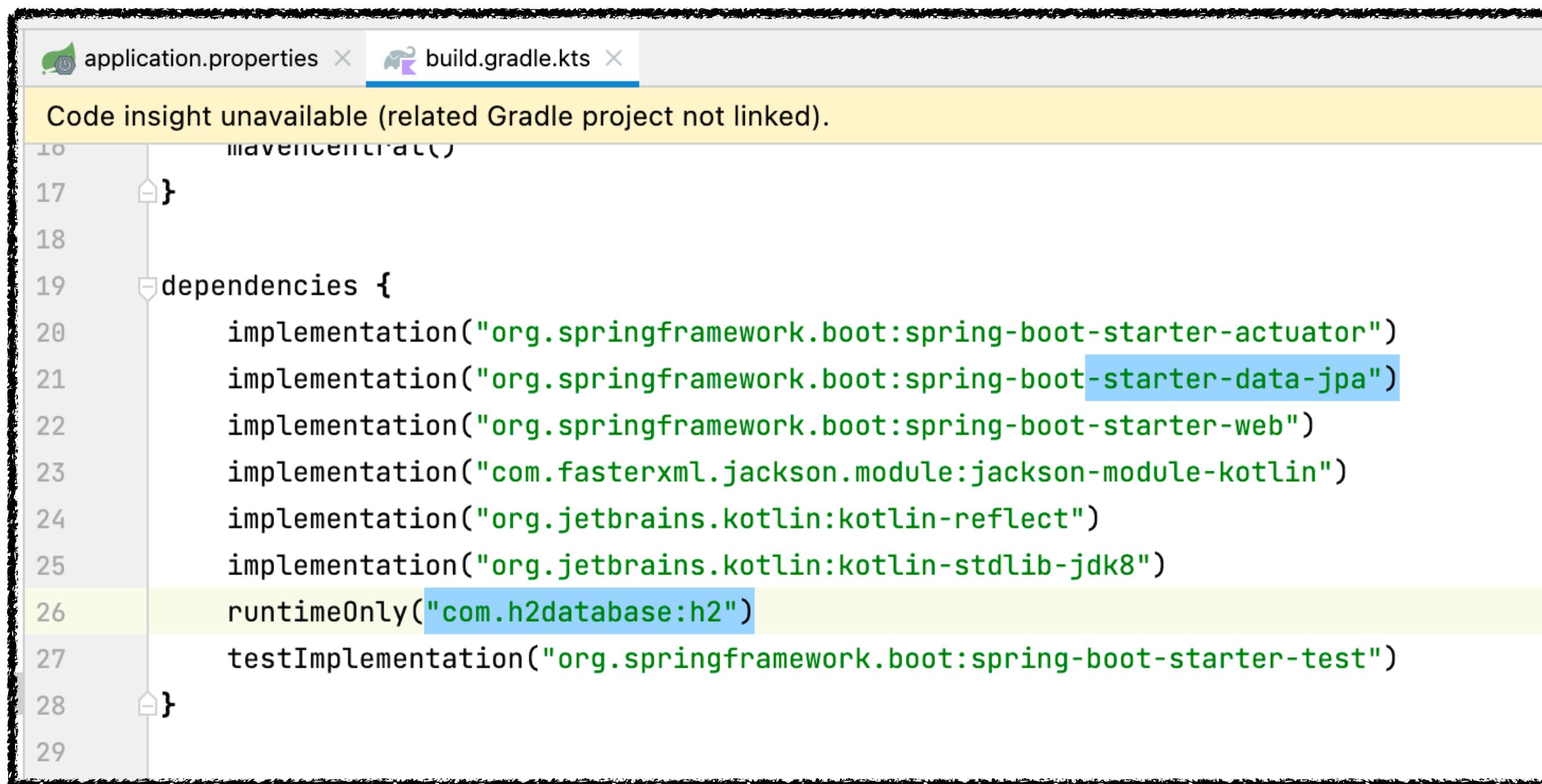
Gradle

demo

Configure Database

Exercise2

- Check if required dependencies are there



The screenshot shows an IDE interface with two tabs: "application.properties" and "build.gradle.kts". The "build.gradle.kts" tab is active. A yellow banner at the top states "Code insight unavailable (related Gradle project not linked)". The code in the file is as follows:

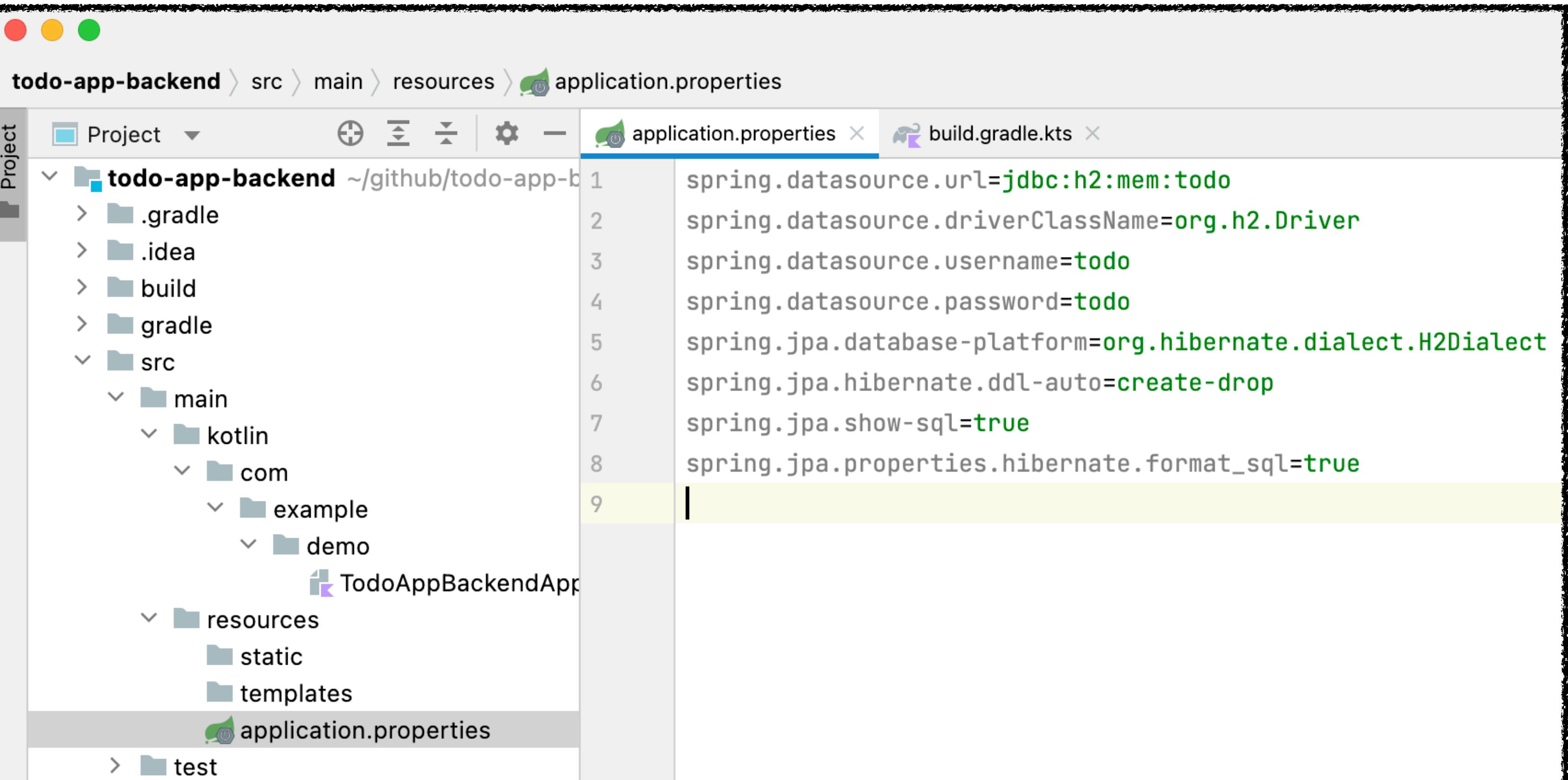
```
16
17 }
18
19 dependencies {
20     implementation("org.springframework.boot:spring-boot-starter-actuator")
21     implementation("org.springframework.boot:spring-boot-starter-data-jpa")
22     implementation("org.springframework.boot:spring-boot-starter-web")
23     implementation("com.fasterxml.jackson.module:jackson-module-kotlin")
24     implementation("org.jetbrains.kotlin:kotlin-reflect")
25     implementation("org.jetbrains.kotlin:kotlin-stdlib-jdk8")
26     runtimeOnly("com.h2database:h2")
27     testImplementation("org.springframework.boot:spring-boot-starter-test")
28 }
29
```

The line "implementation("org.springframework.boot:spring-boot-starter-data-jpa")" is highlighted with a blue rectangle. The line "runtimeOnly("com.h2database:h2")" is highlighted with a yellow rectangle.

Configure Database

Exercise2

- Add database connection string and settings to application.properties



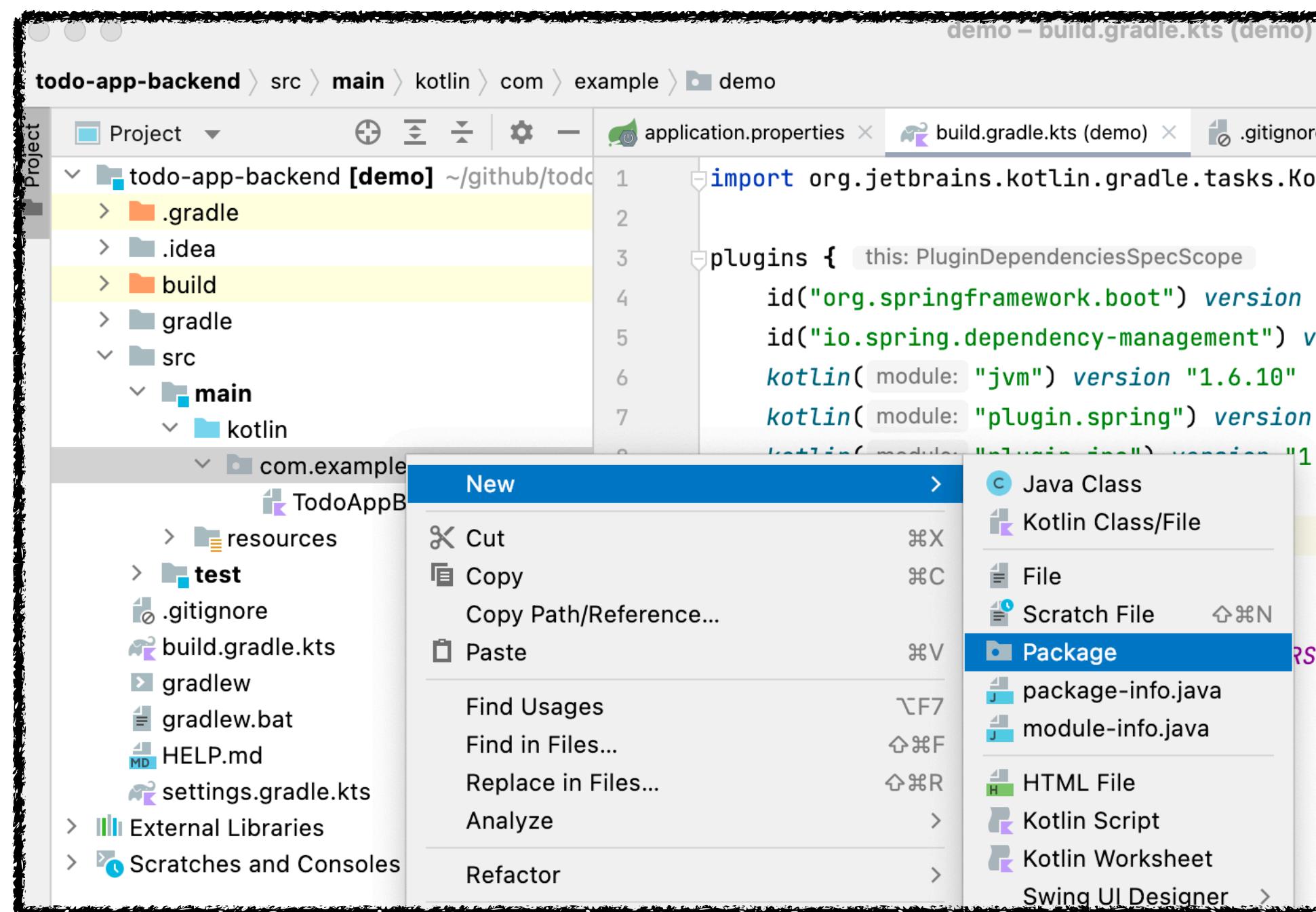
The screenshot shows a code editor window with the file `application.properties` open. The project structure on the left shows a folder named `todo-app-backend` containing `.gradle`, `.idea`, `build`, `gradle`, and `src`. The `src` folder contains `main`, which has `kotlin`, `com`, `example`, and `demo` subfolders. The `TodoAppBackendApp` class is visible under `demo`. The `resources` folder contains `static`, `templates`, and `application.properties`. The `application.properties` file itself is shown with the following content:

```
spring.datasource.url=jdbc:h2:mem:todo
spring.datasource.driverClassName=org.h2.Driver
spring.datasource.username=todo
spring.datasource.password=todo
spring.jpa.database-platform=org.hibernate.dialect.H2Dialect
spring.jpa.hibernate.ddl-auto=create-drop
spring.jpa.show-sql=true
spring.jpa.properties.hibernate.format_sql=true
```

Configure Database

Exercise2

- Add package where we can put our database entities



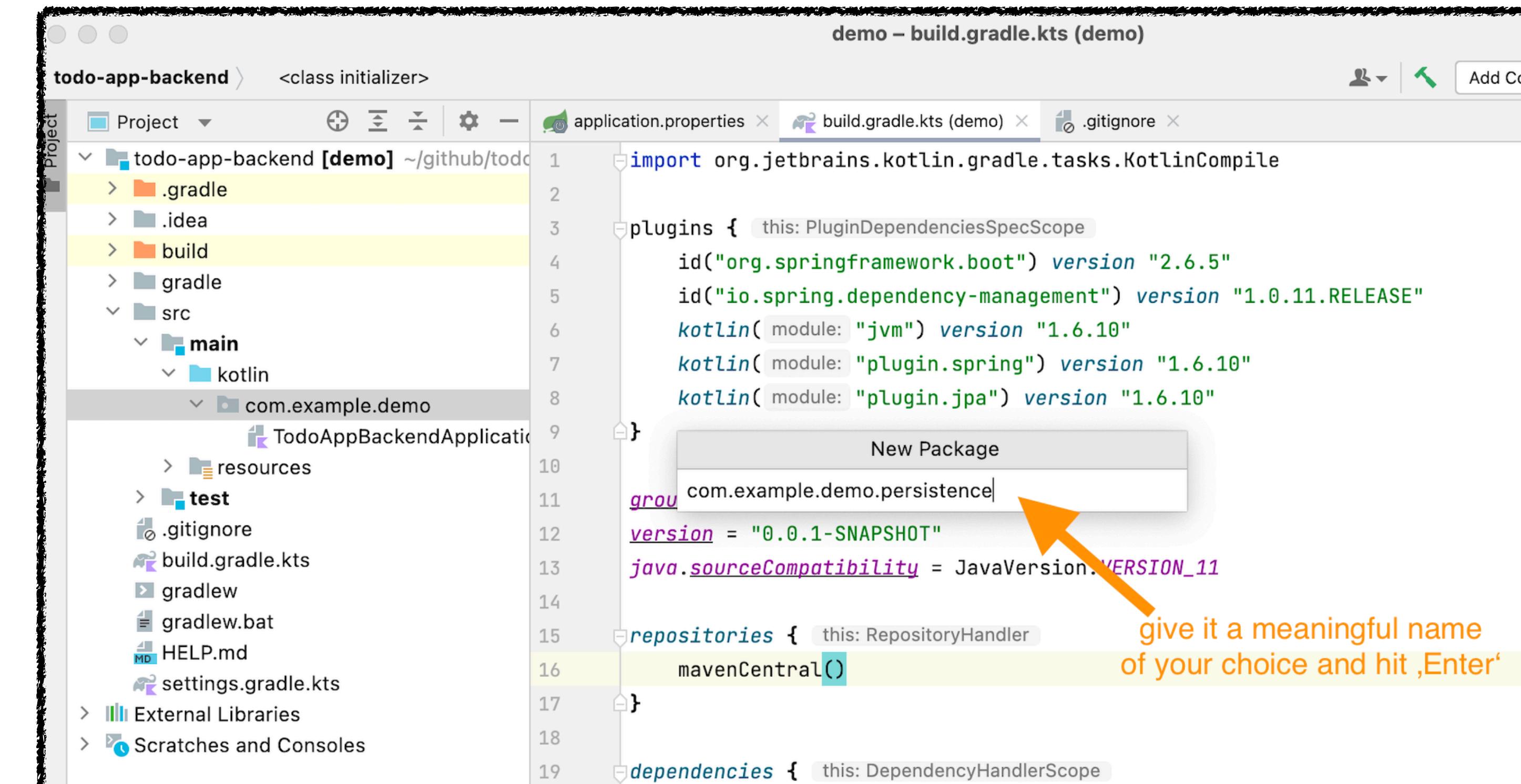
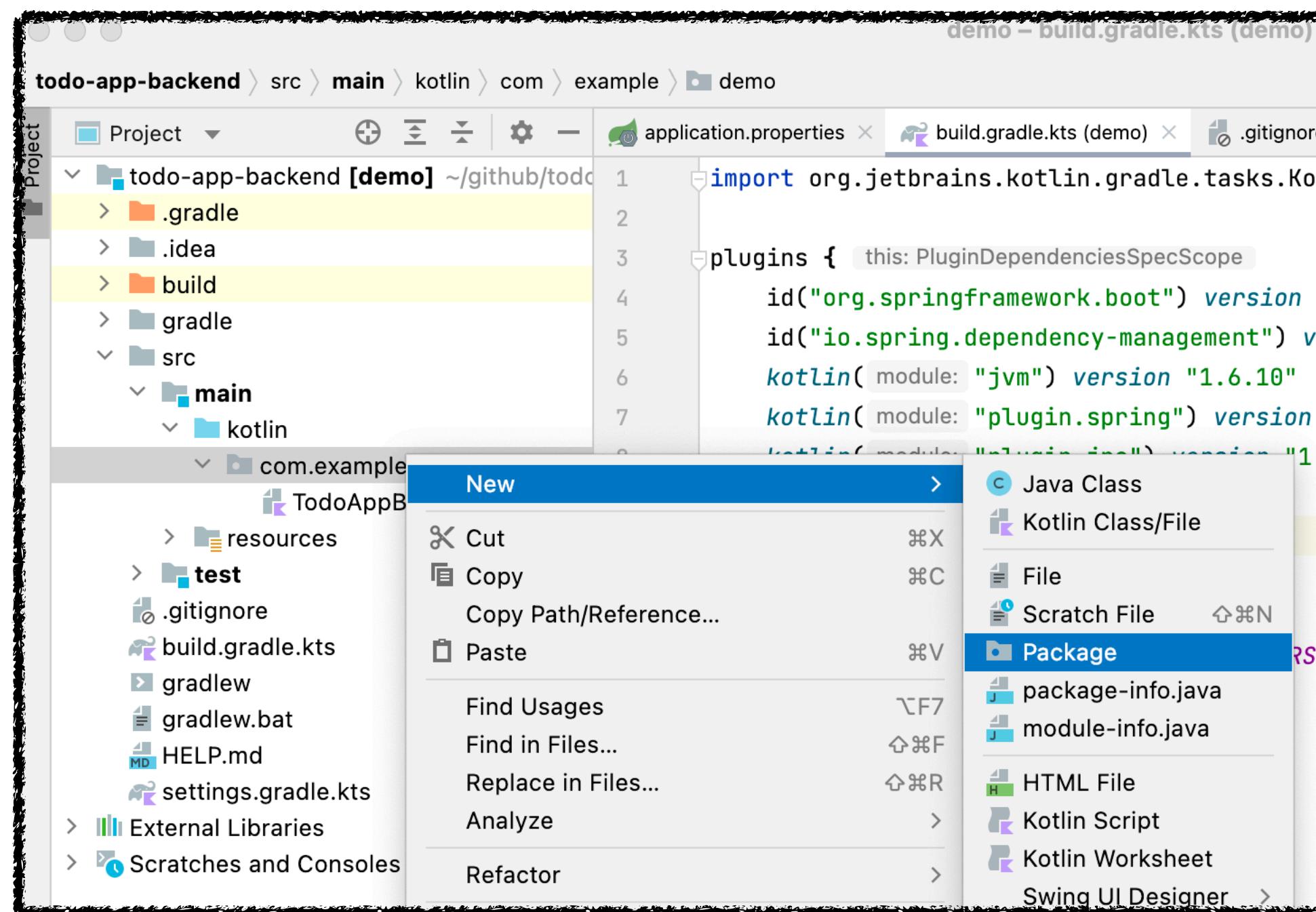
```
demo – build.gradle.kts (demo)
todo-app-backend > <class initializer>
Project application.properties build.gradle.kts (demo) .gitignore
todo-app-backend [demo] ~/github/todo
> .gradle
> .idea
> build
> gradle
> src
> main
> kotlin
> com.example
> TodoAppBackendApplication
> resources
> test
> .gitignore
> build.gradle.kts
> gradlew
> gradlew.bat
> HELP.md
> settings.gradle.kts
> External Libraries
> Scratches and Consoles

import org.jetbrains.kotlin.gradle.tasks.KotlinCompile
plugins {
    id("org.springframework.boot") version "2.6.5"
    id("io.spring.dependency-management") version "1.0.11.RELEASE"
    kotlin("jvm") version "1.6.10"
    kotlin("plugin.spring") version "1.6.10"
    kotlin("plugin.jpa") version "1.6.10"
}
group = "com.example.demo"
version = "0.0.1-SNAPSHOT"
java.sourceCompatibility = JavaVersion.VERSION_11
repositories {
    mavenCentral()
}
dependencies {
```

Configure Database

Exercise2

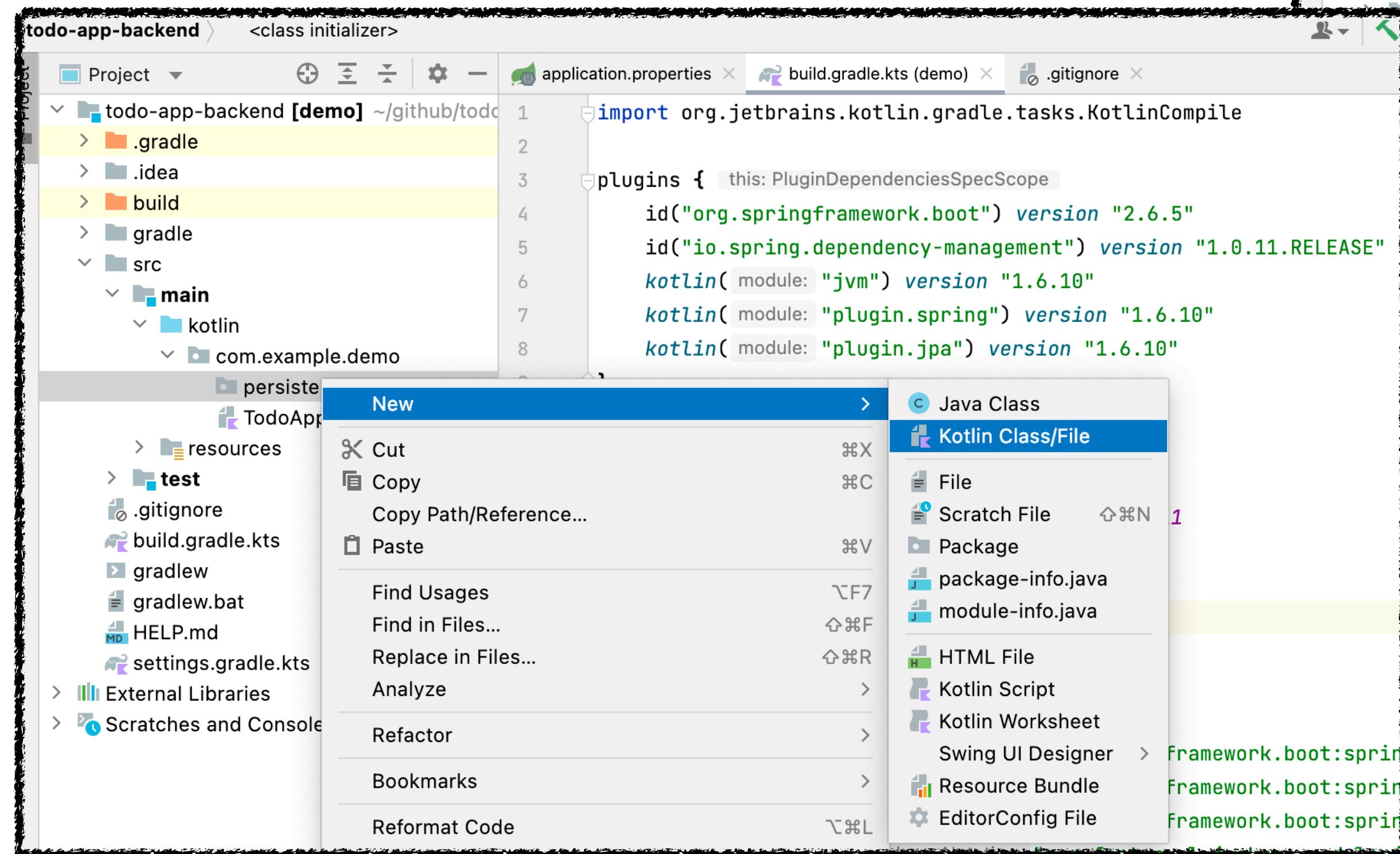
- Add package where we can put our database entities



Configure Database

Exercise2

- Add Todo Class to our package



The screenshot shows the 'build.gradle.kts' file in the 'todo-app-backend' project. The 'com.example.demo' package under 'src/main/kotlin' is highlighted. A 'New Kotlin Class/File' dialog is open, with 'Class' selected. The code in the file includes imports for org.jetbrains.kotlin.gradle.tasks.KotlinCompile, org.springframework.boot, io.spring.dependency-management, kotlin.jvm, kotlin.plugin.spring, and kotlin.plugin.jpa.

```
import org.jetbrains.kotlin.gradle.tasks.KotlinCompile

plugins {
    id("org.springframework.boot") version "2.6.5"
    id("io.spring.dependency-management") version "1.0.11.RELEASE"
    kotlin("jvm") version "1.6.10"
    kotlin("plugin.spring") version "1.6.10"
    kotlin("plugin.jpa") version "1.6.10"
}

group = "com.example.demo"
version = "0.0.1-SNAPSHOT"
java.sourceCompatibility = JavaVersion.VERSION_1_8

repositories {
    mavenCentral()
}

dependencies {
    implementation("org.springframework.boot:spring-boot-starter-data-jpa")
    implementation("com.fasterxml.jackson.module:jackson-module-kotlin")
    implementation("org.jetbrains.kotlin:kotlin-reflect")
    implementation("org.jetbrains.kotlin:kotlin-stdlib")
}
```

Configure Database

Exercise2

- Make to Todo class an Entity

```
todo-app-backend > src > main > kotlin > com > example > demo > persistence > TodoEntity
Project application.properties build.gradle.kts (demo) TodoEntity.kt .gitignore
todo-app-backend [demo] ~github/todo
> .gradle
> .idea
> build
> gradle
src
> main
  > kotlin
    > com.example.demo
      > persistence
        > TodoEntity
TodoAppBackendApplication
> resources
> test
.gitignore
build.gradle.kts
gradlew
gradlew.bat
HELP.md

package com.example.demo.persistence

import javax.persistence.*

@Table(name = "todo")
@Entity(name = "todo")
class TodoEntity(
    @Id
    @GeneratedValue(strategy = GenerationType.IDENTITY)
    var id: Int = 0,
    var task: String,
    var completed: Boolean,
)
```

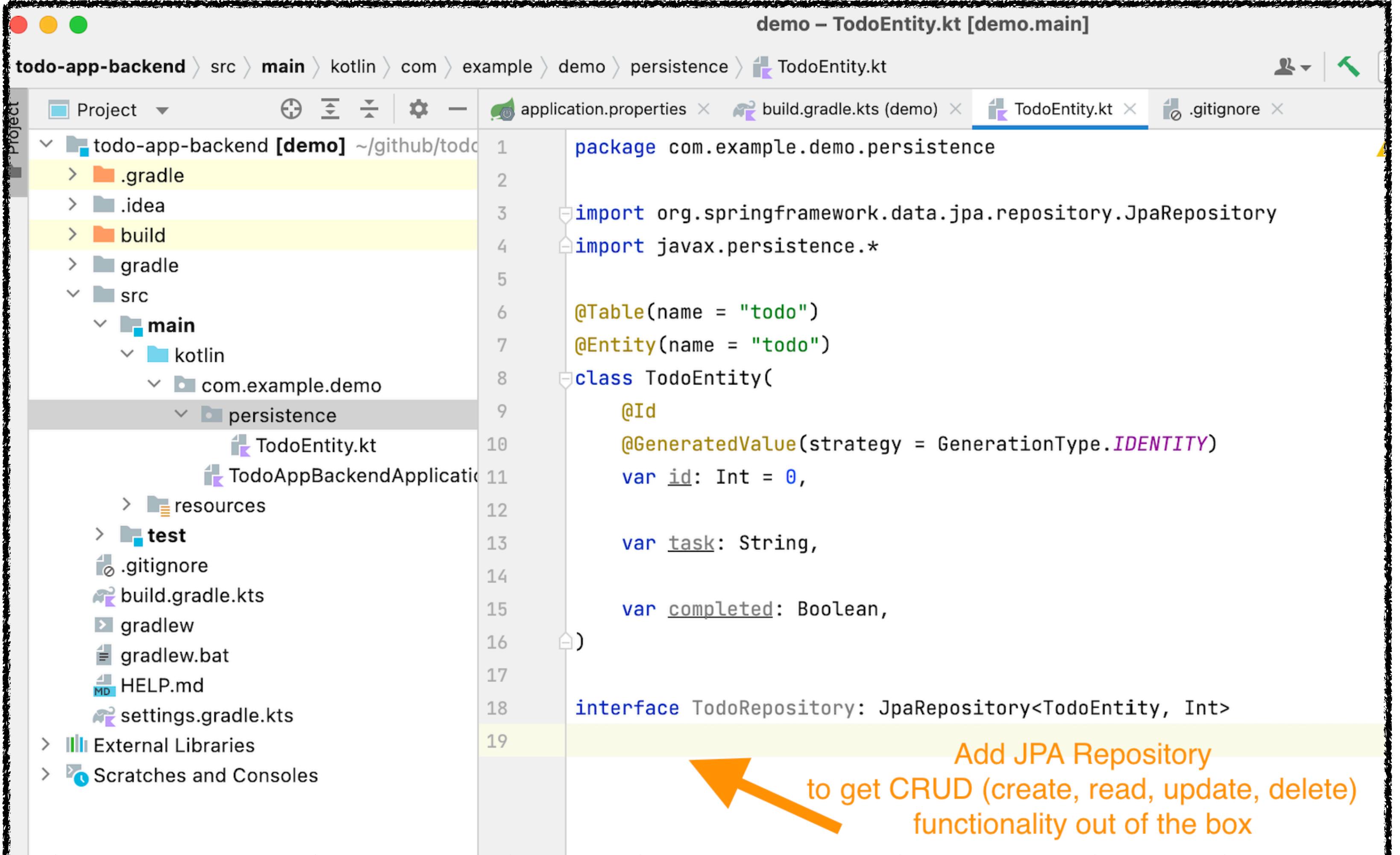
```
Terminal: Local
2022-03-31 09:35:13.103 INFO 46221 --- [           main] com.zaxxer.hikari.HikariConfig       : H2Dialect
Hibernate:
drop table if exists todo CASCADE
Hibernate:
create table todo (
    id integer generated by default as identity,
    completed boolean not null,
    task varchar(255),
    primary key (id)
)
2022-03-31 09:35:13.530 INFO 46221 --- [           main] o.h.e.t.j.p.i.JtaPlatformInitiator : org.hibernate.engine.transaction.jta.platform.internal.NoJtaPlatform
```

You should see our table will be created
if we run our application again

Configure Database

Exercise2

- Add JPA Repository



demo – TodoEntity.kt [demo.main]

todo-app-backend › src › main › kotlin › com › example › demo › persistence › TodoEntity.kt

```
package com.example.demo.persistence
import org.springframework.data.jpa.repository.JpaRepository
import javax.persistence.*

@Table(name = "todo")
@Entity(name = "todo")
class TodoEntity(
    @Id
    @GeneratedValue(strategy = GenerationType.IDENTITY)
    var id: Int = 0,
    var task: String,
    var completed: Boolean,
)

interface TodoRepository: JpaRepository<TodoEntity, Int>
```

Add JPA Repository
to get CRUD (create, read, update, delete)
functionality out of the box