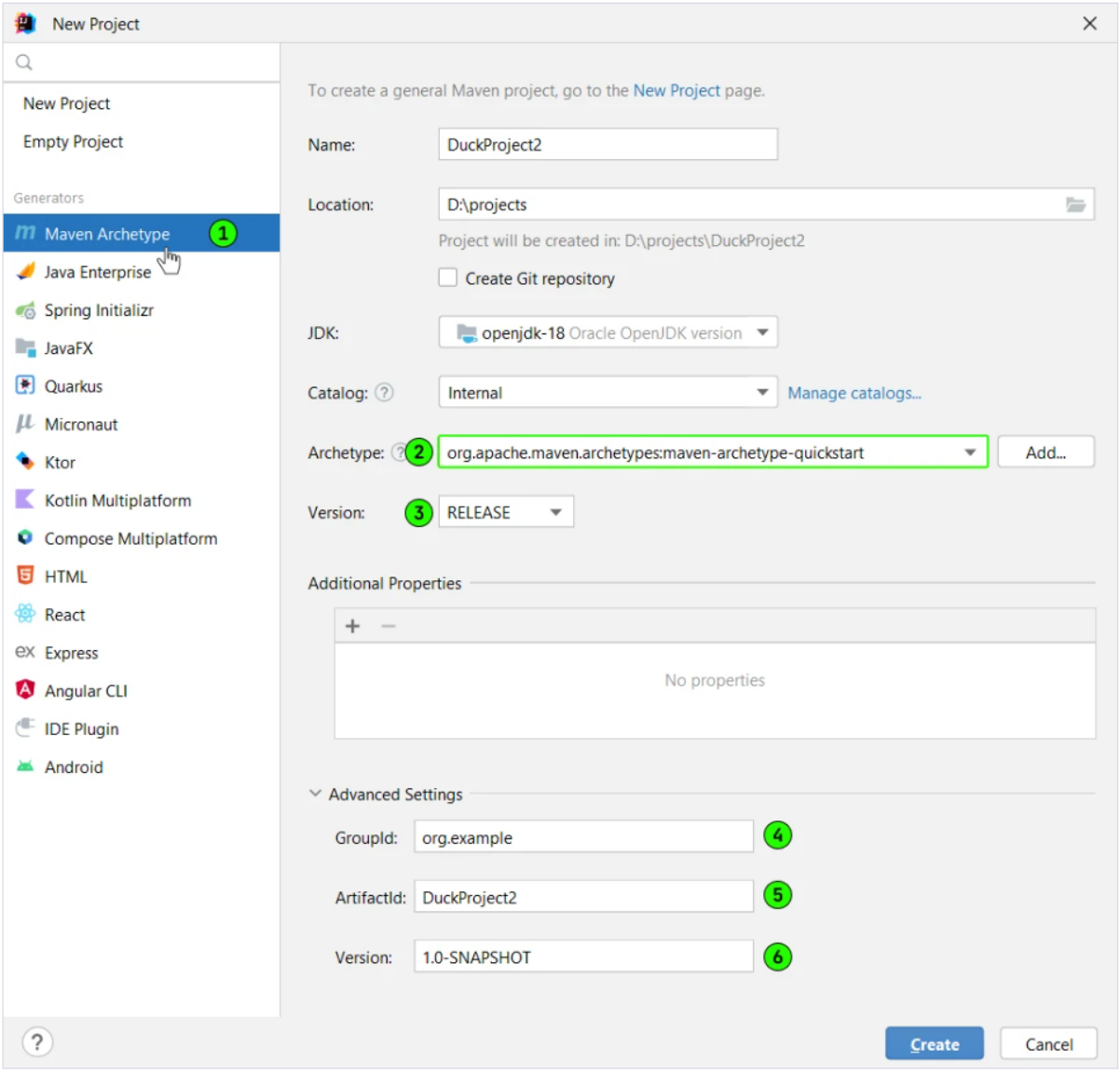
Archetypes in Maven

Introduction to archetypes

There is another way to create a Maven project in IDEA - based on an archetype:



Here it is proposed to create a project based on one of the existing **archetypes** . What are these archetypes and why do we need them?

Maven has standardized project templates - such templates are called archetypes. Remember the starting structure of the project - **the src** , **java** , **test** folders, and so on? So this folder structure is set using the archetype.

There are sample templates on the official Maven website . With their help, you can generate different start-up projects - a simple application, a plugin, a website.

The list of available archetypes can be obtained by running the following command in the console: mvn archetype:generate

Popular archetypes

The most popular archetypes are:

- **maven-archetype-quickstart** ;
- **maven-archetype-** site
- **maven-archetype-webapp** ;
- **maven-archetype-j2ee-simple** ;
- **jpa-maven-archetype** ;
- **spring-mvc-quickstart** .

If you want to create an empty Java project, then use the **maven-archetype-quickstart** archetype . It was the result of his work that you saw when creating a project in IDEA in the last lecture.

If you want to create a web application that will run inside a web server, display HTML pages and all that, then you can safely take the **maven-archetype-webapp** archetype as a basis .

You can use the maven-archetype-site archetype to create a site . Or even the maven-archetype-site-simple archetype if a very simple site is expected. Try different options and see which one you like best.

To work with Hibernate or JPA, you can use the **jpa-maven-archetype** archetype .

And finally, there is also a special archetype for working with Spring - **spring-mvc-quickstart** . It will be very useful for beginners. More similar information on the latter can be found here, at the link .

Why are archetypes good? They wean to write projects from scratch. Nobody writes projects just in Java anymore. Modern projects are written on a technology stack: a list of 5-10 frameworks and a couple of dozen libraries is the modern “language in which I write”.

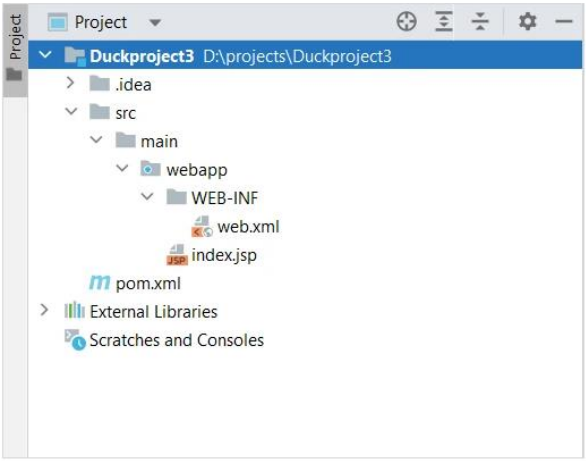
Web application on Maven

Separately, I would like to dwell on the **maven-archetype-webapp** archetype .

This is a classic web application written in Java. And although it is a little outdated after the popularity of Spring, you, as a beginner, will find it very useful. What do you need to know about this archetype?

It allows you to build a simple **web application** - the build result will be **a .war file** . Deploy can be configured so that your web application is immediately added to Tomcat. Finally, you can experiment with primitive servlets and JSPs.

If you create a project based on this archetype, you will get the following folder structure:



Here are a few interesting things:

- webapp folder;
- WEB-INF folder;
- web.xml file;
- index.jsp

Firstly, there is a **webapp** folder (from Web Application), in which all the resources of your web application will be stored.

Secondly, the **web.xml** file is the web application deployment descriptor . It describes how your web application should interact with the web server and its clients.

Thirdly, there is an **index.jsp** file , which is a fairly simple form of a servlet. It's working and you can experiment with your first JSP servlet by changing it.

We will talk about all this in more detail in the topic dedicated to servlets and web applications.