

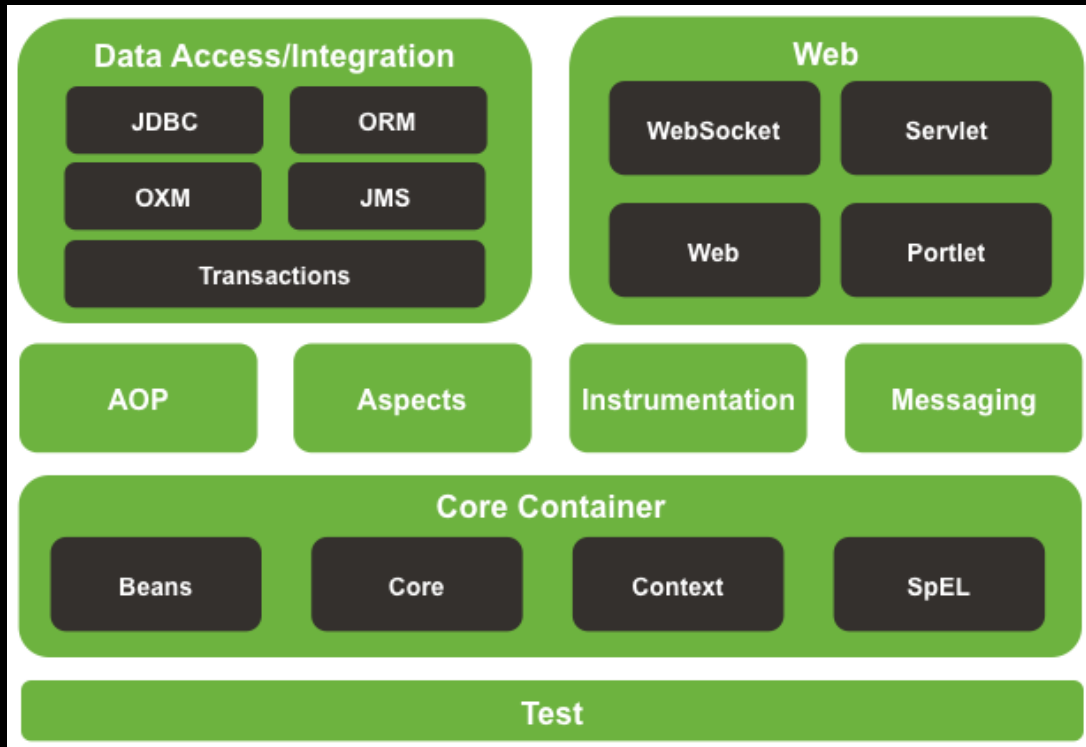
# Spring Framework

## Introduction

# The Spring Framework is everywhere



# The Spring Framework Modules



Spring is huge. In fact, this is a whole container of frameworks that allow you to perform tasks of any complexity - from working with the database to testing procedures.

The Spring Framework 5.0 consists of features organized into about 20 modules.

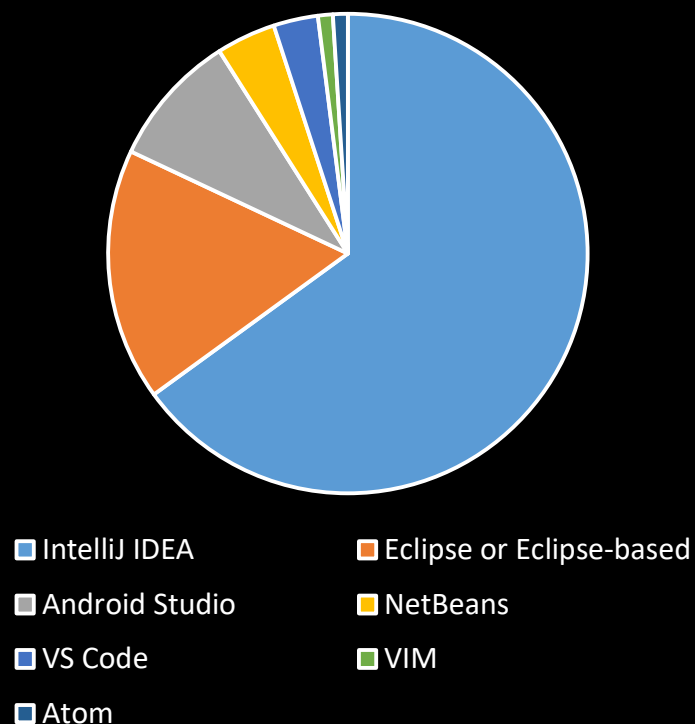
These modules are grouped into:

- Core Container,
- Data Access/Integration,
- Web,
- AOP (Aspect Oriented Programming),
- Instrumentation,
- Messaging,
- Test.



# 65% of Java developers choose IntelliJ IDEA

Which IDE / editor do you use the most for Java development?

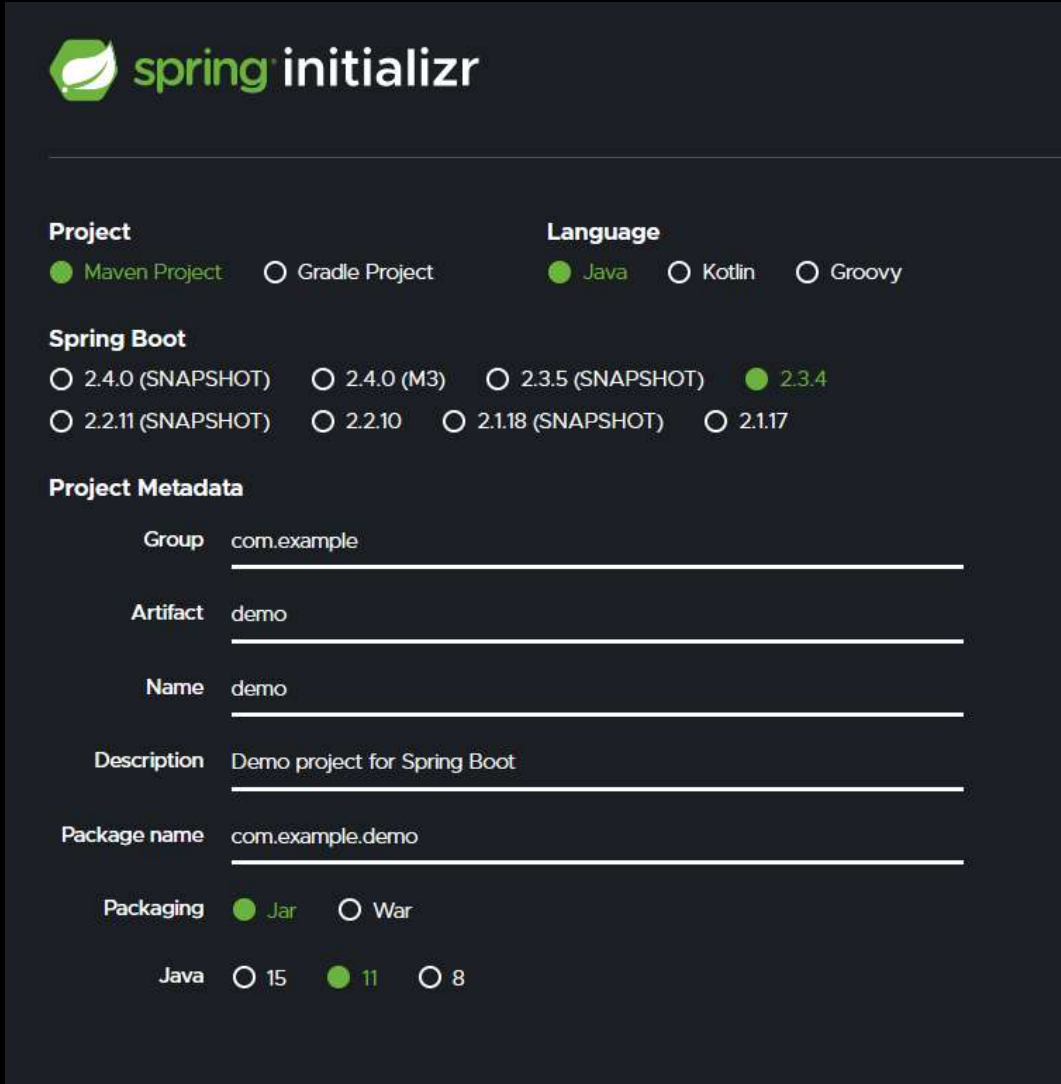


# Spring Initializr

*Spring Initializr* provides a simple and intuitive web UI to create and configure and generate *Spring* based application. This tool makes it easy for developers to generate an initial project structure without worrying much about the project structure and dependencies. On a high-level *Spring Initializr tools* take cares of the following points for any *Spring* based application

- Project structure;
- Dependencies required to start the work;
- build script (*Maven* or *Gradle*) to build the application;
- Language and version (*Initializr* will add correct dependencies based on the version);
- packaging (*war* or *jar*);
- *Spring Initializr* is available on the web, most of the *IDE* has a built-in integration with the *Initializr*.

# Initial project setup



The image shows the Spring Initializr web interface for creating a new project. It features a dark theme with green accents. The form is organized into sections: Project, Language, Spring Boot, Project Metadata, and Packaging. The 'Project' section has radio buttons for 'Maven Project' (selected) and 'Gradle Project'. The 'Language' section has radio buttons for 'Java' (selected), 'Kotlin', and 'Groovy'. The 'Spring Boot' section has radio buttons for various versions, with '2.3.4' selected. The 'Project Metadata' section includes text input fields for 'Group' (com.example), 'Artifact' (demo), 'Name' (demo), 'Description' (Demo project for Spring Boot), and 'Package name' (com.example.demo). The 'Packaging' section has radio buttons for 'Jar' (selected) and 'War'. At the bottom, there are radio buttons for 'Java' versions 15, 11 (selected), and 8.

**Project**

☒ Maven Project ☐ Gradle Project

**Language**

☒ Java ☐ Kotlin ☐ Groovy

**Spring Boot**

☐ 2.4.0 (SNAPSHOT) ☐ 2.4.0 (M3) ☐ 2.3.5 (SNAPSHOT) ☒ 2.3.4

☐ 2.2.11 (SNAPSHOT) ☐ 2.2.10 ☐ 2.1.18 (SNAPSHOT) ☐ 2.1.17

**Project Metadata**

Group

Artifact

Name

Description

Package name

**Packaging**

☒ Jar ☐ War

Java ☐ 15 ☒ 11 ☐ 8

This is the simplest Initializr view with minimal options to build your project. Let's quickly discuss these fields for a basic understanding.

- **Project** – This gives us the option to select the project as *Maven* or *Gradle* based. In this post, we choose Maven. This selection generates a *pom.xml* file for our project.
- **Language** – The programming language to use. You have the option to select *Groovy* or *Kotlin*.
- **Spring Boot** – *Spring Boot* version.
- **Group** – The groupId attribute in *Apache Maven*, also known as the id of the project's group.
- **Artifact** – The name of the project, also known by the artifactId attribute in *Apache Maven*.
- **Description** – Description of the project.
- **Package Name** – Root package of the project. If not specified, the value of the *Group* attribute used.
- **Java** – the *Java* version to use.

# Connect GitHub repository

