# ENTERPRICE COMPUTING USING FULL STACK

## 23MX24

## 1. Installation Process for Java Full Stack

**Latest version :**

IntelliJ IDEA 2023.3.

**Websites for download :**

<https://www.jetbrains.com/idea/download/?section=windows>

**Hardware and Software Requirements :**

|  |  |  |
| --- | --- | --- |
| **Requirements** | Minimum | Recommended |
| RAM | 2 GB of free RAM | 8 GB of total system RAM |
| CPU | Any modern CPU | Multi-core CPU. IntelliJ IDEA supports multithreading for different operations and processes making it faster the more CPU cores it can use. |
| Disk Space | 3.5 GB | SSD drive with at least 5 GB of free space |
| Operating System | Officially released 64-bit versions of the following:   * Microsoft Windows 10 1809 or later   Windows Server 2019 or later   * macOS 11.0 or later * Any Linux distribution that supports Gnome, KDE , or Unity DE.   IntelliJ IDEA is not available for the Linux distributions that do not include [GLIBC](https://ftp.gnu.org/gnu/libc/) 2.27 or later. | Latest 64-bit version of Windows, macOS, or Linux (for example, Debian, Ubuntu, or RHEL) |

We do not need to install Java to run IntelliJ IDEA because JetBrains Runtime is bundled with the IDE (based on JRE 17). However, to develop Java applications, a standalone JDK is required.

**Tools and Configurations detail**

Set up Java Environment Variables:

* Update JAVA\_HOME and PATH environment variables.

Integrated Development Environment (IDE) Setup:

* Configure IDE with JDK.
* Install necessary plugins for Java development.

**Sample apps development**

* **Initialize Java Project:**

Open IntelliJ IDEA and create a new Maven project.

Choose the "maven-archetype-quickstart" archetype.

Fill in the GroupId and ArtifactId.

* **Backend Development (Server):**

Add dependencies for Spring Boot and Spring Web in your **pom.xml**.

Create a **TodoController.java** class for your server.

Define a simple endpoint to return a message.

* **Frontend Development:**

Create a **resources/static/index.html** file for your frontend.

Write basic HTML to display a message.

* **Build and Run:**

Open the **Application.java** class generated by IntelliJ IDEA.

Add the **@SpringBootApplication** annotation.

Right-click on the **Application** class and choose "Run 'Application'."

Visit http://localhost:8080 in your browser.

## 2. Installation Process for Javascript Full Stack

**Latest version :**

January 2024 (version 1.86)

**Websites for download :**

<https://code.visualstudio.com/download>

**Hardware and Software Requirements :**

[Hardware](https://code.visualstudio.com/docs/supporting/requirements#_hardware)

Visual Studio Code is a small download (< 200 MB) and has a disk footprint of < 500 MB. VS Code is lightweight and should easily run on today's hardware.

We recommend:

* 1.6 GHz or faster processor
* 1 GB of RAM

[Platforms](https://code.visualstudio.com/docs/supporting/requirements#_platforms)

VS Code is supported on the following platforms:

* Windows 10 and 11 (64-bit)
* macOS versions with Apple security update support. This is typically the latest release and the two previous versions.
* Linux (Debian): Ubuntu Desktop 20.04, Debian 10
* Linux (Red Hat): Red Hat Enterprise Linux 8, Fedora 36

**Additional Requirements for Linux**

* GLIBCXX version 3.4.25 or later
* GLIBC version 2.28 or later

**Tools and Configurations detail**

Javascript Tools for Visual Studio. Use JavaScript to quickly develop reusable web services that are directly deployable to Azure. Take advantage of first-class debugging, profiling, and unit testing integration as well as a wide selection of project templates to help you get started fast.

* Install Visual Studio:

Visual Studio

* Create JavaScript Project:

Choose a template based on your project needs.

* Configure IntelliSense:

Ensure IntelliSense is working for JavaScript.

* Debugging:

Set up debugging for JavaScript code.

* Run Project:

Use "Start Debugging" or F5 to run the project.

* Task Runner Explorer :

Run tasks using Task Runner Explorer.

* Browser Link :

Enable Browser Link for real-time updates.

* npm Package Manager :

Integrate npm packages into the project.

* Build and Bundling :

Configure build processes using Webpack or Grunt.

* NuGet Package Manager:

Manage JavaScript libraries via NuGet.

* Source Control Integration:

Use Git or other version control systems.

* Custom Build Scripts :

Integrate custom build scripts.

* Explore Additional Features:

Utilize features like Solution Explorer.

**Sample apps development**

1. **Initialize Node.js Project:**
   * Open Visual Studio Code and create a new folder for your project.
   * Open the terminal in VSCode and run **npm init -y** to create a **package.json** file.
2. **Backend Development (Server):**
   * Install Express.js using **npm install express**.
   * Create a **server.js** file for your server logic.
   * Define routes, handle requests, and integrate with databases if needed.
3. **Frontend Development:**
   * Create an **index.html** file for your static page.
   * Style your page using CSS in a separate **styles.css** file.
   * Write frontend logic using JavaScript in a **app.js** file.
4. **Integration:**
   * Connect the frontend and backend using RESTful APIs.
   * Ensure proper communication between layers.
   * Define API endpoints for data exchange.
5. **Build and Run:**
   * Set up npm scripts in your **package.json** for build and run tasks.
   * For development, consider using tools like nodemon for automatic server restarts.
   * Run the project using **npm start** or your defined script.