Datasheet

Development Tools in the Payara® Platform

The Payara® Platform - Production-Ready,
Cloud Native and Aggressively Compatible.

Development tools are a true companion of developers and no matter what platform you use, there's always a set of essential tools that developers can use to make their lives easier.

While you're free to use as many as needed, it's important to choose tools that align with your project's requirements and help improve your productivity so you can complete the project on time.

The Payara Platform provides a wide variety of development tools to simplify development tasks, enhance the developer experience, and accelerate developer's productivity. This datasheet will discuss many of the useful and no-cost development tools offered by the Payara Platform and the developer community.

Build Tool Plugin

Payara's build tool plugin can automate the creation of war archives or executable Uber JAR bundle of the application and run the Payara Micro application in-place from source code.

Payara Micro Maven Plugin

The Payara Micro Maven Plugin provides Payara Micro support in Maven, which can start and stop a Payara Micro instance and create an Uber JAR bundle of the application.

To incorporate the maven plugin into your project POM file, add the plugin snippet in the **plugins** section of your **pom.xml**, as shown in the following example:

```
<plugins>
    ...
    <plugin>
        <groupId>fish.payara.maven.plugins</
        groupId>
        <artifactId>payara-micro-maven-plugin</
        artifactId>
```

Server Requirements

- As a direct replacement for GlassFish Server Open Source Edition, Payara Server uses the same basic system requirements:
- JDK8u163 or above
- 512MB RAM

Support for Any Operating System Running One of the Following Java Virtual Machines

- Oracle JDK8 (u162+)
- Azul Zulu JDK8 (u162+)
- OpenJDK JDK8 (u162+)
- Oracle JDK 11 (11.0.4+)
- Azul Zulu JDK11 (11.0.4+)
- OpenJDK JDK11 (11.0.4+)

Related Products & Services

- Payara Micro - microservice and cloud environments
- Payara Cloud – PaaS for 1-click cloud deployments
- Payara Accelerator Consultancy – to help you get the most out of your Java EE/ Jakarta EE applications

```
        <version>1.0.5</version>
        <configuration>
            <deployWar>true</deployWar>
        </configuration>
    </plugin>
    ...
</plugins>
```

To modify the default behavior of the plugin, add the configuration options in payara-micro- maven-plugin's **configuration** section of your **pom.xml**, as shown in the following example:

Configuration Example:

```
<configuration>
    <useUberJar>true</useUberJar>
    <payaraVersion>5.192</payaraVersion>
    <daemon>true</daemon>
    <deployWar>false</deployWar>
    <contextRoot>myApp</contextRoot>
    <commandLineOptions>
        <option>
        <key>--autoBindHttp</key>
        <value>true</value>
        </option>
    </commandLineOptions>
</configuration>
```


Plugin Goals

bundle: This goal bundles the attached project's artifact into an Uber JAR with the specified configuration.
Usage: `mvn payara-micro:bundle`

start: This goal starts Payara Micro with the specified configuration.
Usage: `mvn payara-micro:start`

stop: This goal stops Payara Micro with the specified configuration.
Usage: `mvn payara-micro:stop`

dev: The dev goal ensures rapid compilation & deployment on file save and streamlines development by enabling auto-deployment, live reload, persistent session state, and readable logging. It extends the start goal with optimized settings for efficient web application development.
Usage: `mvn payara-micro:dev`

Payara Micro Gradle Plugin

The Payara Micro Gradle Plugin provides Payara Micro support in Gradle, which can start and stop Payara Micro instances and create an Uber JAR bundle of the application.

To incorporate the Gradle plugin into your project GRADLE file, add the plugin snippet in the **plugins**

section of your **build.gradle**, as shown in the following example:

```
plugins {
  id "fish.payara.micro-gradle-plugin" version "1.0.2"
}
```

To configure the plugin, add the **payaraMicro** extension block with the configuration tags in your **build.gradle**, as shown in the following example:

Configuration Example:

```
payaraMicro {
    deployWar = true
    useUberJar = false
    daemon = false
    commandLineOptions = [port: 8787]
    javaCommandLineOptions = [Dtest: 'test123', ea:true]
}
```

Plugin Tasks:

bundle: This task bundles the attached project's artifact into an Uber JAR with the specified configuration.
Usage: `gradle microBundle`

start: This task starts Payara Micro with the specified configuration.
Usage: `gradle microStart`

stop: This task stops Payara Micro with the specified configuration.
Usage: `gradle microStop`

Integrated Development Environment

Writing business logic is one of the most important tasks for a developer when building an application. This is generally done within an Integrated Development Environment (IDE).

Benefits of IDEs

Reduces setup time:

By using an IDE, developers don't need to spend time configuring multiple tools or constantly switching tools if they want to download the latest version of the Payara Platform, manage (start/ stop) the platform, add the new database driver, or view the database tables, for example.

Faster development tasks:

Tighter integration of all development tasks maximizes developer productivity beyond just helping with setup tasks. For example, automatically reloading an application onto Payara Server or Payara Micro after making some changes in the source file.

Continuous learning:

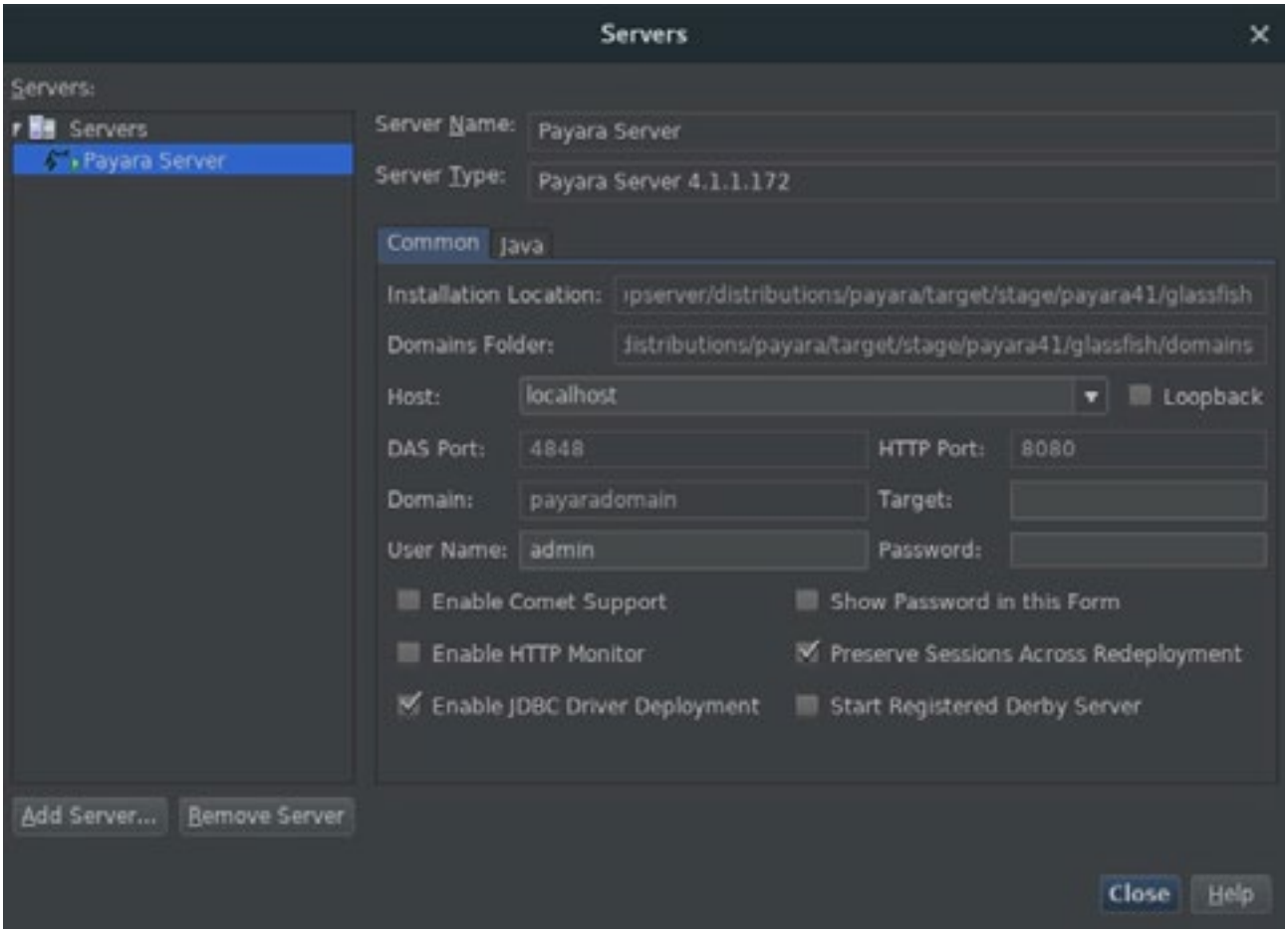
An IDE also supports continuous learning by providing helpful insights, educating developers and keeping them up-to-date. For instance, the IDE's help topics in the Payara Dev tools are constantly being updated, together with Payara Micro sample project templates.

The main goals of an IDE are improving ease of use, offering effective platform support, and improving developer productivity. When talking about ease of use, a common IDE that comes to mind is Apache NetBeans IDE.

Apache NetBeans IDE

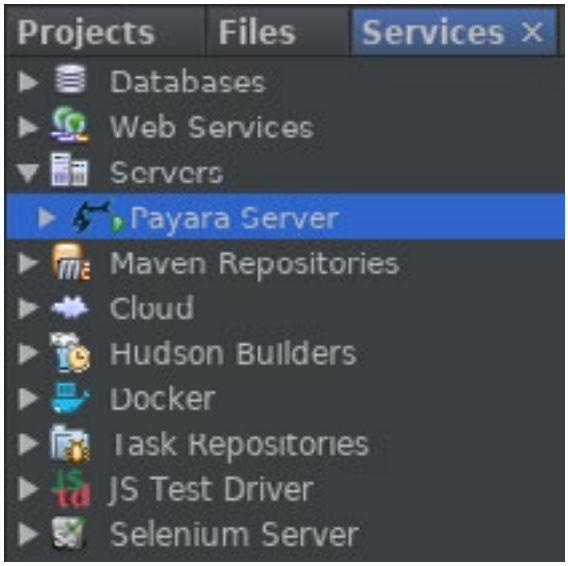
Apache NetBeans IDE 11.1 provides out-of-the-box support for Payara Server and Payara Micro.

Apache NetBeans enables developers to download the latest version of Payara Server, which can be easily configured and added to the NetBeans IDE.



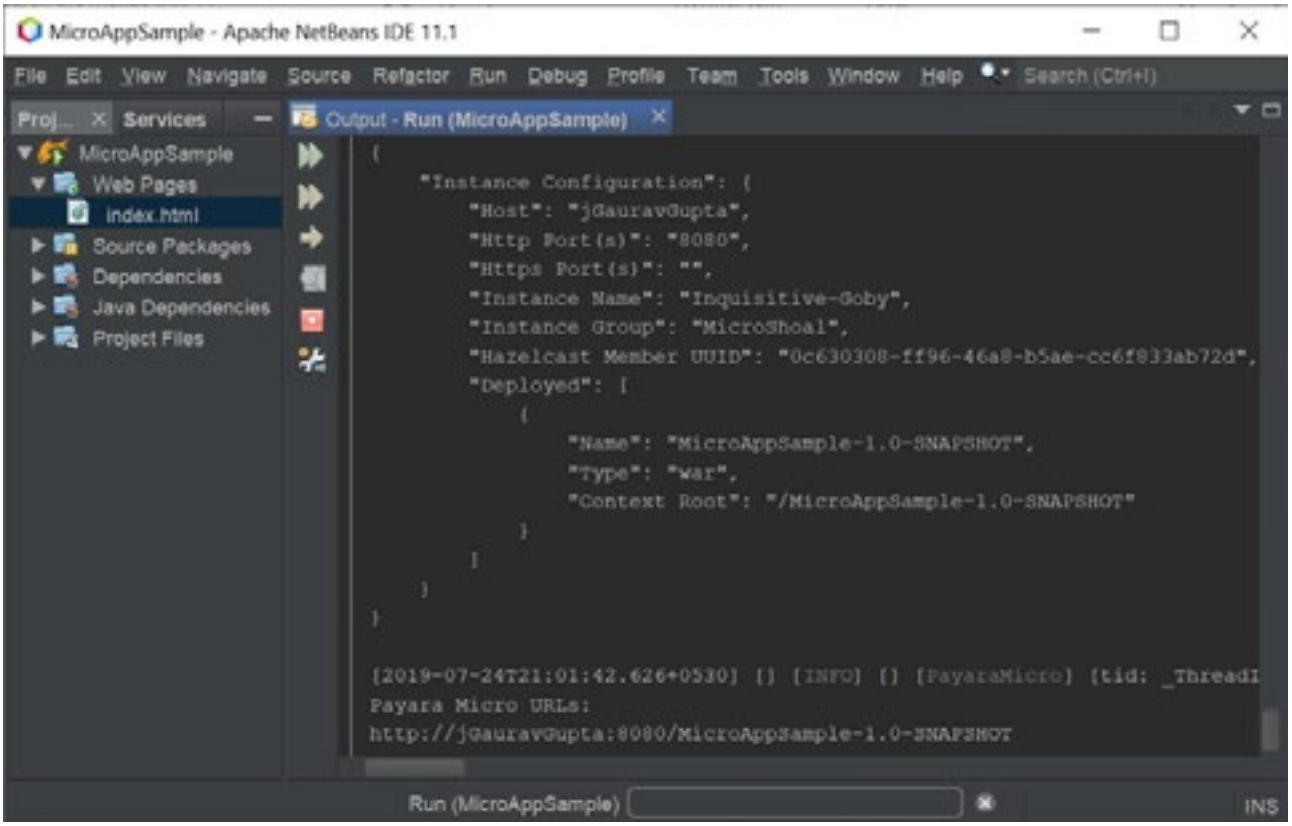
Once Payara Server is added to NetBeans it can be found under **Servers** with the configured name. Right-clicking on the server lets you do the following things:

- Start or stop the server.
- Start debugging the server.
- Open admin console.
- Open server log.



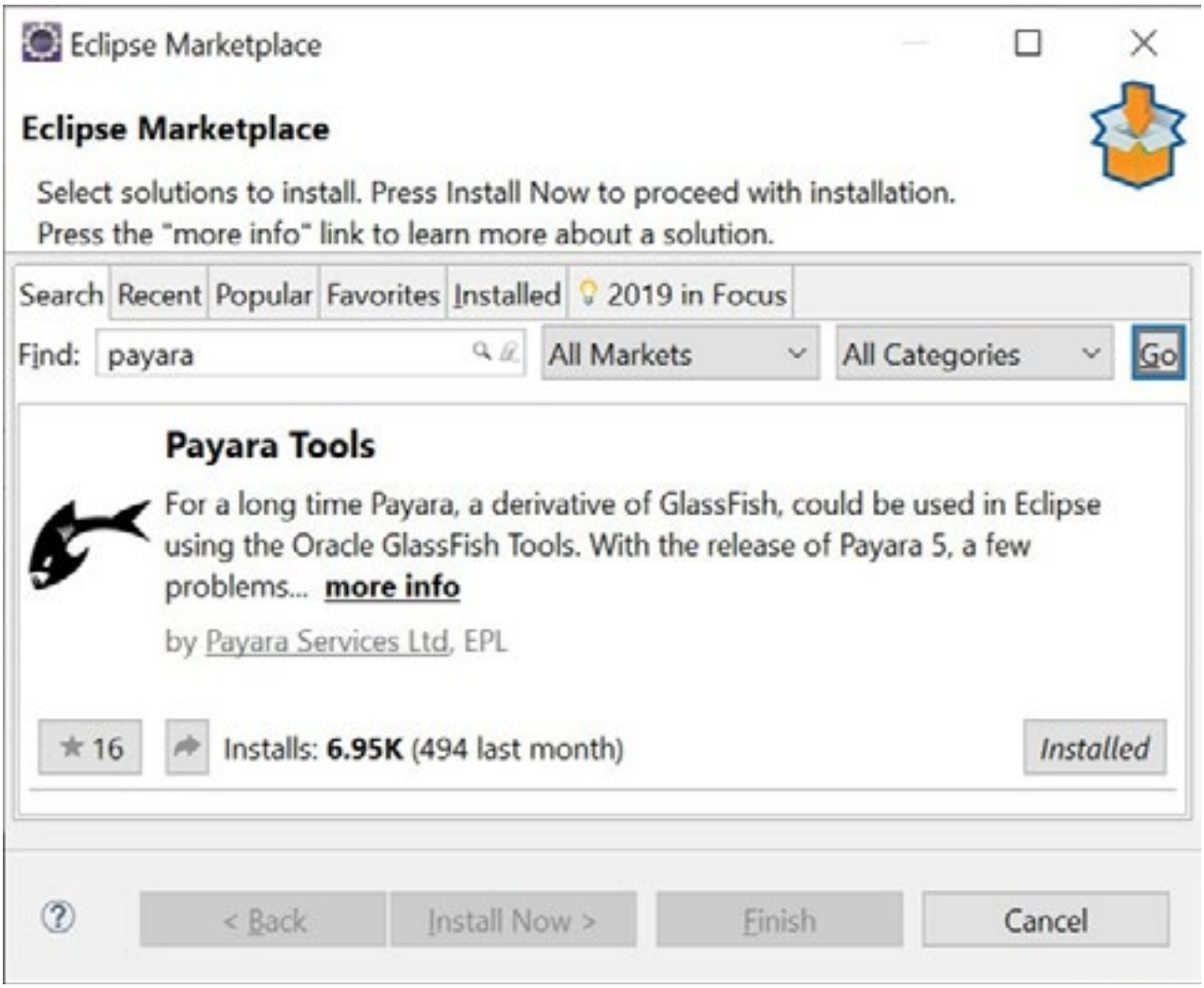
An application can be deployed and undeployed onto the Payara Server easily from within NetBeans IDE by selecting the target Payara Server in the application properties.

Apache NetBeans IDE also provides first-class support to create and run Payara Micro maven web applications, which is the microservices-ready version of Payara Server.

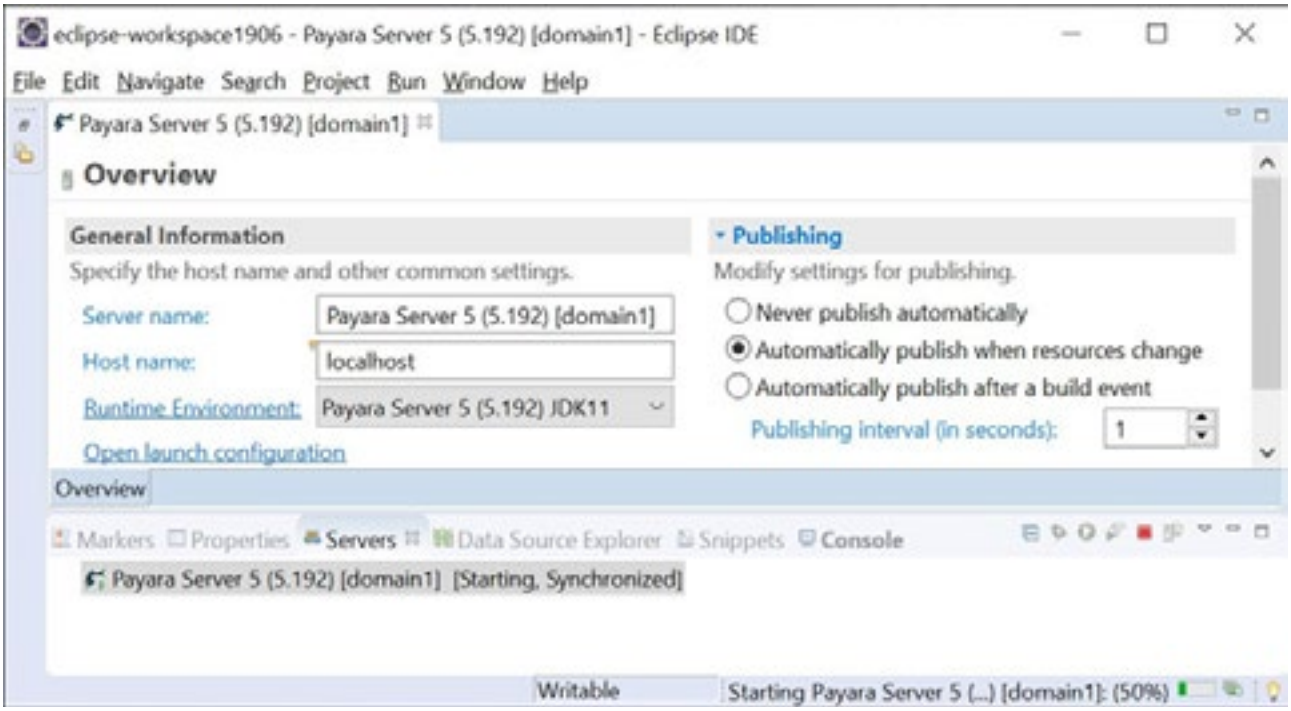


Eclipse IDE

Payara Server and Payara Micro can be easily configured and managed in Eclipse IDE by just installing the Payara Tools from the Eclipse Marketplace.

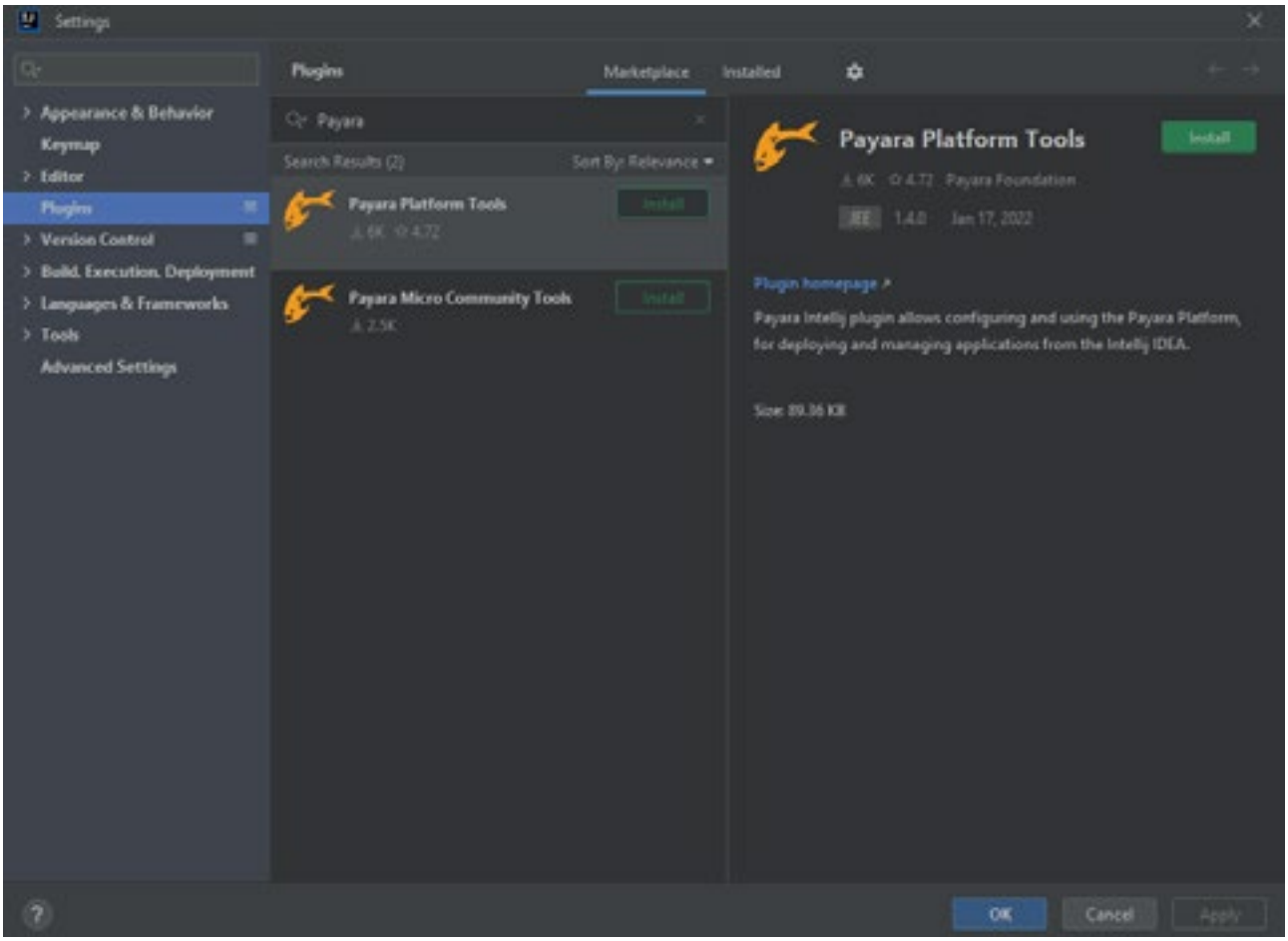


Payara Tools in Eclipse IDE enables publishing of projects to Payara Server, as well as controlling Payara Server from **Servers tab** within Eclipse IDE.



IntelliJ IDEA Ultimate

IntelliJ IDEA is developed by JetBrains. Payara IntelliJ IDEA tools provide int egration of the entire Payara Platform, which encompasses Payara Server, Payara Micro and Payara Cloud, into IntelliJ IDEA Ultimate. To get started, you need to download and install the Payara IntelliJ tools before you can use them. These are available on JetBrains Marketplace. We have a detailed tutorial for how to create, debug and deploy a Maven web application to the Payara Platform. You can follow the same process and Payara Platform configuration steps for a Gradle web application.



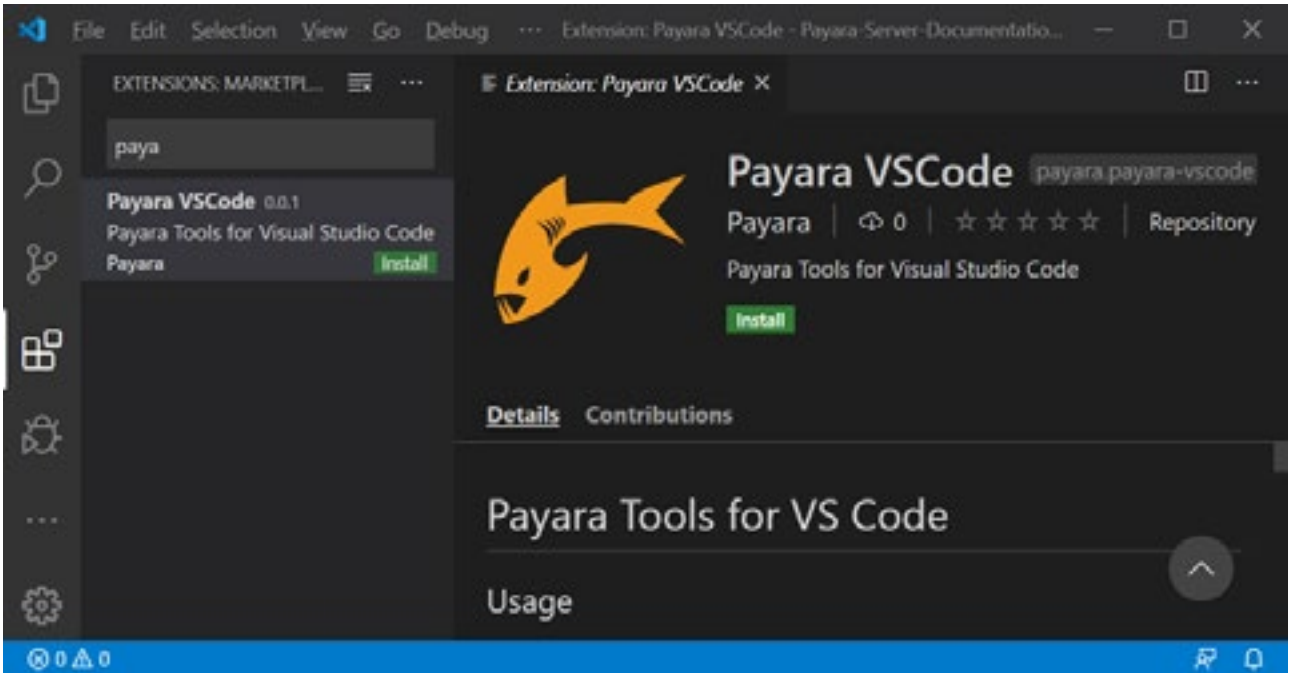
IntelliJ IDEA Community

Payara IntelliJ Community tools offer integration with Payara Micro and Payara Cloud. To use Payara Server features, IntelliJ IDEA Ultimate Edition is needed.

You can install Payara IntelliJ Community tools from JetBrains Marketplace to enable the integration of Payara Micro and Payara Cloud into your Community Edition setup. You can create, debug and deploy Payara Micro applications as well as deploy the Jakarta EE application to the Payara Cloud following a similar process to that in the Ultimate Edition.

VSCODE

The Payara VS Code Extension integrates both Payara Server and Payara Micro with Visual Studio Code, providing a streamlined development experience within the lightweight IDE. This extension allows you to easily deploy, debug and manage Payara Micro applications as well as Payara Server instances directly from VS Code, making it easier to work with the Payara Platform in a more flexible, code-centric environment.



Application Generator

Payara Starter Archetype

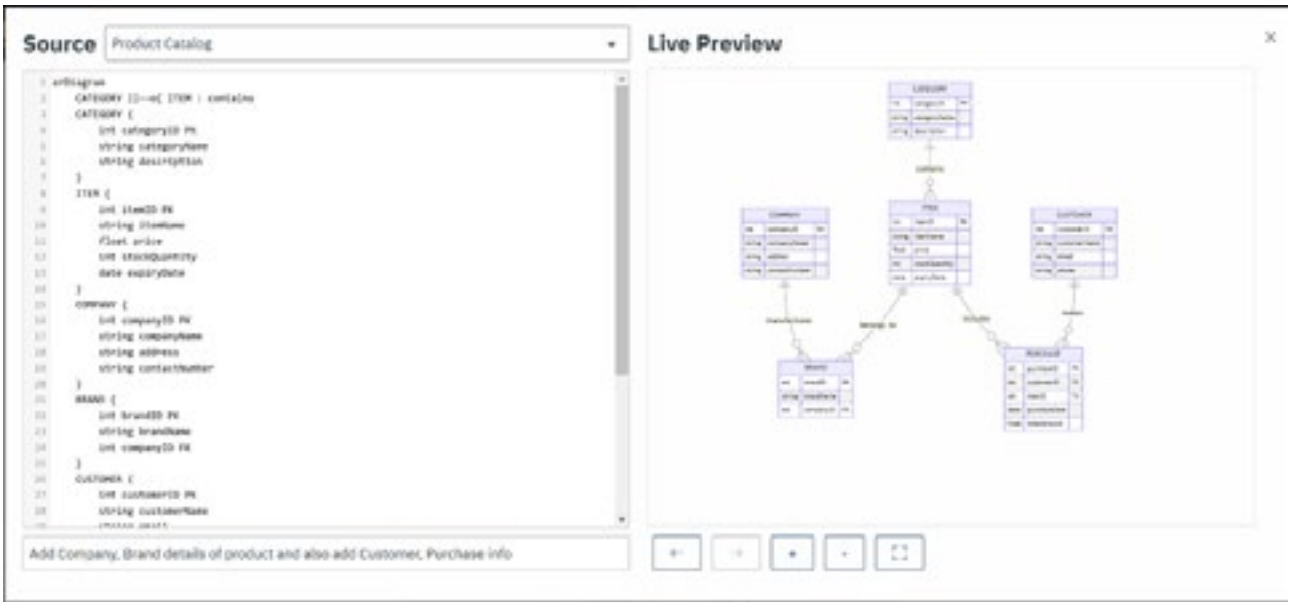
The Payara Starter Archetype is a foundational tool that simplifies the setup of Jakarta EE applications by providing a template-based approach. It helps developers quickly generate the structure of Maven and Gradle projects, enabling fast application development. To create a new project from the archetype, you can run the following command:

```
mvn archetype:generate -DarchetypeGroupId=fish.payara.starter
-DarchetypeArtifactId=payara-starter-archetype -DarchetypeVersion=1.0-beta12
```

Payara Starter (https://start.payara.fish/)

Payara Starter is an online platform that accelerates full-stack Jakarta EE application creation by generating custom project structures. Leveraging tools like Generative AI and ER diagram support, it enables developers to visually design and customize their applications, streamlining development and reducing setup time.

You may generate the Payara Micro maven application by visiting the Payara Starter (https://start.payara.fish/) page:



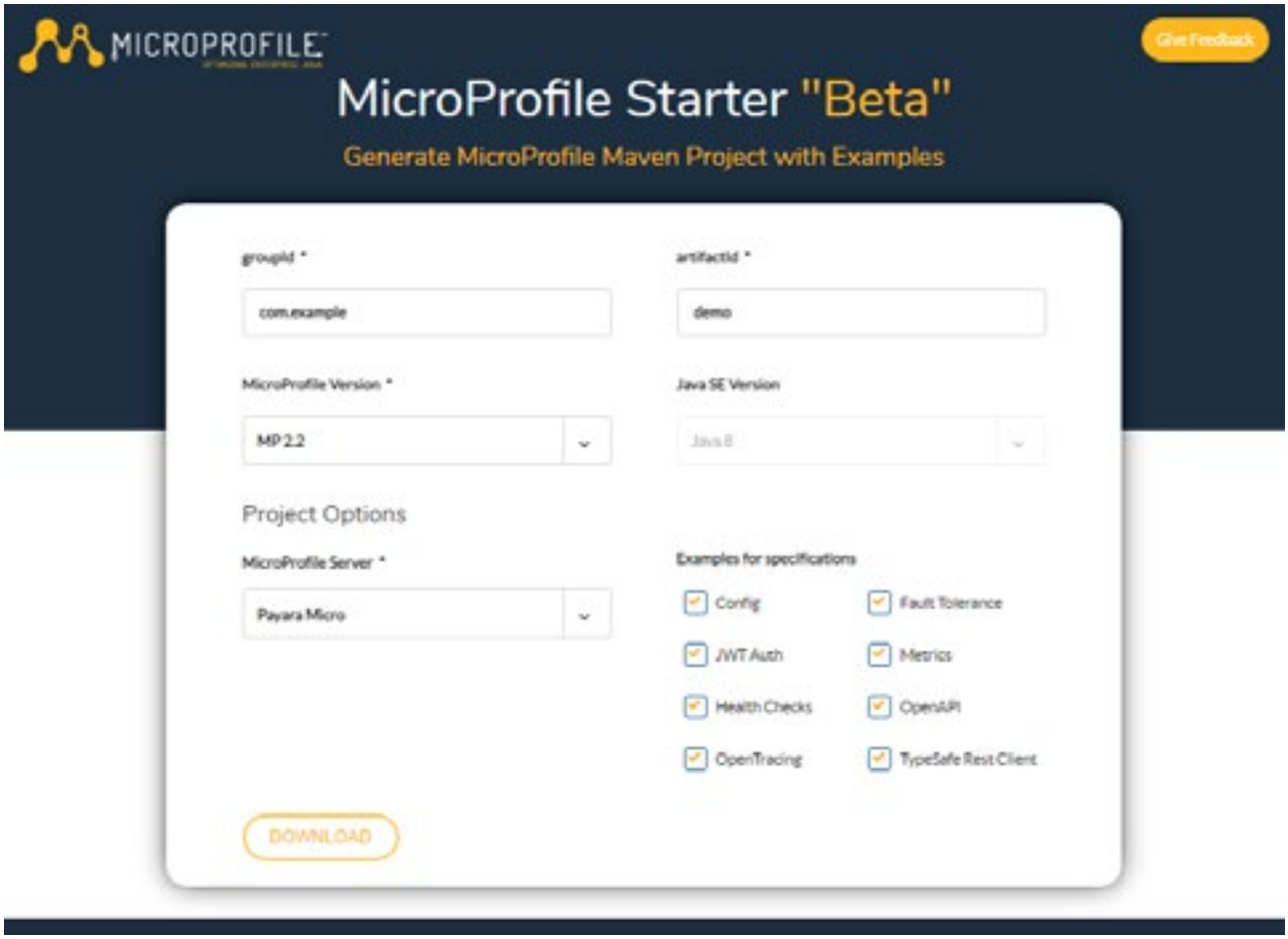
Jeddict (community tool)

Jeddict is an open-source Jakarta EE and MicroProfile full-stack application generator and modeler. Jeddict provides out-of-the-box support for Payara Platform.

MicroProfile Starter (community tool)

MicroProfile Starter is an open-source MicroProfile maven application generator.

You may generate the Payara Micro maven application by visiting the MicroProfile Starter (https://start.microprofile.io/) page:



Select the MicroProfile Version and **Payara Micro** in MicroProfile Server option to download the Payara Micro project.

Boost Productivity with Payara Platform Dev Tools

We all can agree that having the right development tools can tremendously boost productivity, irrespective of which development platform you choose. That's why selecting the right tools at the beginning of a project will help a great deal in successfully creating your application.

Payara Platform and Dev Tools Resources

Try Payara Platform. Experience the benefits of developing Java EE applications in our Java application server.
Download: https://www.payara.fish/downloads

Just getting started with Payara? Watch a video tutorial, read technical overviews and resources to get the most out of Payara Server.
Learn more: http://info.payara.fish/getting-started-with-payara-server-useful-resources

Try Development Tools.
Apache NetBeans: https://netbeans.apache.org/download/
Eclipse Payara tools: https://marketplace.eclipse.org/content/payara-tools
VSCODE: https://marketplace.visualstudio.com/items?itemName=Payara.payara-vscode
IntelliJ Ultimate: https://plugins.jetbrains.com/plugin/15114-payara-ultimate-tools
IntelliJ Community: https://plugins.jetbrains.com/plugin/15445-payara-community-tools

Payara Micro Maven Plugin:
https://mvnrepository.com/artifact/fish.payara.maven.plugins/payara-micro-maven-plugin
Payara Micro Gradle Plugin: https://plugins.gradle.org/plugin/fish.payara.micro-gradle-plugin

Payara Starter Archetype:
https://mvnrepository.com/artifact/fish.payara.starter/payara-starter-archetype
Payara Starter: https://start.payara.fish/
MicroProfile Starter: https://start.microprofile.io
Jeddict: https://jeddict.github.io/
Learn more: https://docs.payara.fish/documentation/ecosystem/ecosystem.html

Get involved. Join the Payara Community and help feed the fish!

Payara Platform and its development tools are, and always will be, open source and we want your ideas, feedback and collaboration for ensuring Payara Platform is the best option for production Java EE/Jakarta EE applications.

Learn more: https://www.payara.fish/community

About Payara Services, Ltd

Payara Services is a global open source company and a recognized leader in the creation of innovative infrastructure software for today and tomorrow. We are proud to nurture and grow an open and collaborative community that builds on the needs of all to advance our software and services while providing support, stability, and security.

Our engaged team operates with the freedom and support to develop industry-leading products and services that enable our users to create world-class solutions across a diverse range of industries.

We help shape the future of the industry through our direct contributions to Jakarta EE and Eclipse MicroProfile® as Eclipse Foundation Solutions Members and members of the Project Management Committee.

sales@payara.fish

UK: +44 800 538 5490
Intl: +1 888 239 8941

www.payara.fish