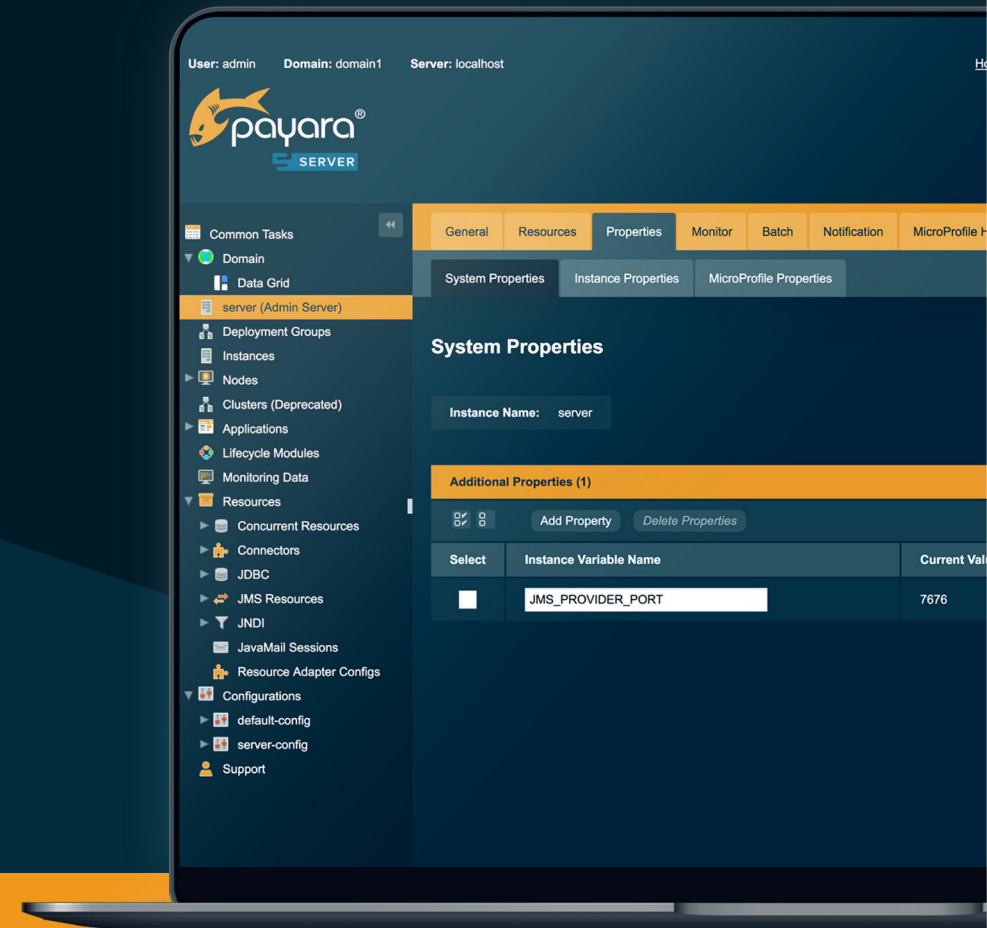




Payara Platform Features Catalog

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



Contents

Payara Platform Features Catalog	4
Admin Console	6
Admin Console Auditing Service	6
AMX Beans	6
Asadmin Recorder	6
Auto-naming	6
Certificate Management	6
Classloader Configuration	7
Concurrent CDI Bean Loading	7
Data Grid Encryption	7
Docker Instance	7
Docker Nodes	7
Domain Data Grid	8
EJB over HTTP	8
High Availability	8
JCache	8
JMX Monitoring	8
Log Message Formatting	8
Log Rotation	9
MicroProfile Config	9
MicroProfile Config Sources	9
MicroProfile Fault Tolerance	9

MicroProfile Health Check	9
MicroProfile JWT Authentication	9
MicroProfile Metrics	9
MicroProfile Open Tracing	10
MicroProfile OpenAPI	10
MicroProfile Rest Client	10
Monitoring Service	10
Notification Service	10
Notifiers	10
Notifiers API	10
OSGI Deployment	11
Password Alias	11
Payara Health Check	11
Payara InSight	11
Persistent EJB Timers	11
Request Tracing	11
REST API	11
REST Monitoring	11
Secure Administration	12
Slow SQL Logger	12
SQL Trace Listeners	12
SQL Tracing	12

Payara Platform Features Catalog

The Payara Platform Features Catalog offers an overview of the features provided by Payara Platform products. The majority of our features are available in both Payara Server and Payara Micro software and in both the Community and Enterprise Editions, but if a specific feature is only available in one of our products, it is noted in the title.

Features in Payara Server only:



Features in Payara Enterprise Edition only:



Almost all features available in the Payara Platform Enterprise Edition may be manually added and built in Payara Platform Community Edition as it's an open source project, but may or may not be supported in the next release.

Features Catalog Last Updated: **11th January 2020**



Payara Platform Enterprise

Software.Security.Stability.Support

Payara Server Enterprise Payara Micro Enterprise

Robust. Reliable. Supported. Small. Simple. Serious.
 The best application platform for production Jakarta EE (Java EE) apps.
 The platform of choice for containerized Jakarta EE (Java EE) microservices deployments.



Payara Platform Community

Open source server runtimes for containerized Jakarta EE and MicroProfile applications.

Payara Server Community Payara Micro Community

Cloud-native, innovative open source middleware platform supporting Jakarta EE applications in any environment: on premise, in the cloud, or hybrid.
 Open source middleware platform for containerized Jakarta EE applications requiring no installation, configuration, or code rewrites.

Enterprise Features



- Enterprise Support**
- Choice of 24x7, 10x5, or Migration & Project Support provided by Payara Engineers
 - Customer Hub
 - Customer Knowledge Base
 - OpenJDK Support
 - Upgrade tooling



- Stable and Secure**
- 10-Year Software Lifecycle
 - Designed for production environments
 - Security alerts, fixes, & Hot fixes
 - Downported patches
 - Full support for ecosystem components
 - New features added after stability confirmed



- Open Source Assurance**
- Payara Enterprise subscriptions include the Open Source Assurance program to protect customers developing and deploying open source solutions.



- Enterprise Release Schedule**
- Monthly Releases

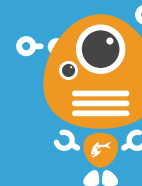
Community Features



- Community Support**
- Get assistance on Payara Forum
 - Raise bugs on GitHub
 - Tech Blog
 - User Guides & Datasheets



- Innovation**
- Rapid access to new features
 - Software experimentation
 - Suitable for development projects



- Open Source**
- Payara Community users can download and install Payara Server or Payara Micro without a contract, but there is no Open Source Assurance program.



- Community Release Schedule**
- Frequent Releases

Admin Console



The Admin Console is an easy-to-navigate graphical interface to configure Payara Server and your applications that includes help documentation.

Admin Console Auditing Service



If you are concerned with the changes that are performed on the domain, you can activate the Admin Console Auditing Service in Payara Server. It will keep track of all actions that are executed (manually or through the [asadmin CLI](#) tool) so that you can later determine if any problem was linked to a certain command that was executed. [Learn more.](#)

AMX Beans



Use AMX Beans to access, configure and monitor Payara Server with a vendor neutral protocol. It can be accessed by client applications such as JConsole, VisualVM, or Azul Mission Control.

Asadmin CLI



The asadmin CLI is a command line interface to perform quick administrative tasks from a script. Using the asadmin CLI eliminates potential errors when performing manual actions with the [Admin Console](#).

Asadmin Recorder



The Asadmin Recorder feature will allow you to record the actions you take in the Payara Server [Admin Console](#) as asadmin commands. Once enabled, the asadmin command for the corresponding action will be recorded to a file. Configuring your Payara Server and instances is ideally performed using asadmin commands rather than the Admin Console. Repeating manual actions in the Admin Console can cause errors and is time consuming. The Asadmin Recorder allows you to learn about the corresponding asadmin commands of the manual actions you perform so that you can automate the setup of the domain in the future. [Learn more.](#)

Auto-naming



Each resource needs a unique name so that it can be identified. In autoscaling environments, this requirement become a burden and the Auto-Naming features helps by automatically assigning names to ensure there is no conflict. [Learn more.](#)

Certificate Management



Certificate Management asadmin commands in Payara Server Enterprise makes it easier and faster to install and update SSL certificates while preventing mistakes. [Learn more.](#)

Classloader Configuration



Using the Classloader Configuration delegation options it is possible to include other versions of some dependencies that are included within the server. This allows updating frameworks like Mojarra and EclipseLink other than the ones provided by Payara Server. [Learn more.](#)

Concurrent CDI Bean Loading

One of the tasks that take some time when you deploy the application is to identify all CDI beans and prepare the Bean Management system. To speed this process, it can be performed in a multi-threaded way. However, there are some cases where this multi-threaded approach results in some issues and cannot be used. [Learn more.](#)

Data Grid Encryption

The Data Grid Encryption offers end-to-end encryption to store and exchange sensitive information between instances securely. Information exchanged with Data Grid Encryption can only be read by the Payara instance. [Learn more.](#)

Deployment Group



The Deployment Group is the clustering feature of Payara Server. It can be used as a target for deployments and other resources so that each are performed on the instances assigned to the Deployment Group. Payara Server's Deployment Groups are an extremely flexible form of clustering as each instance can be part of one or more Deployment Groups. When a Deployment Group is restarted, a rolling restart of each instance is performed to ensure the information contained in the [Domain Data Grid](#) is kept. [Learn more.](#)

Docker Instance



A Payara instance created on a Docker node is a Docker Instance. When the container starts, the instance can join a [Deployment Group](#). At registration time, the instance can use a specified configuration available on the Domain and when it joins a Deployment Group, all application assigned to that Deployment Group are also deployed on the Docker Instance. [Learn more.](#)

Docker Nodes



The Docker Node feature in Payara Platform allows you to create additional nodes and Payara instances running within a Docker container. Both your instances and Docker container are under the control of the Domain Server and can be managed using either the [Admin Console](#) or the asadmin CLI tool. [Learn more.](#)

Domain Data Grid

The Domain Data Grid makes it easy to create a cluster of Payara instances running your application and works right out-of-the-box in variety of network topologies with little to no configuration needed for Amazon AWS, Google Cloud, Microsoft Azure, Kubernetes, and LAN environments. The Domain Data Grid can also share and store information for your application, so it is accessible by all instances. [Learn more.](#)

EJB over HTTP



The standard EJB remoting protocol requires a heavyweight client and the protocol has some issues when using an environment performing NAT (Network Address Translation). The Lite Remote EJB based on HTTP overcomes both problems but has some limitations on the functionality that can be achieved. [Learn More.](#)

High Availability

High Availability for the Domain Data Grid ensures there is always a backup of application and system information, such as web session information, stored in the event an instance goes down. The data is redistributed when a new instance joins the Domain Data Grid. [Learn more.](#)

JCache

With the JCache functionality you can define how data is stored in a Cache in a standard way. Since the JCache implementation in Payara Platform is provided by Hazelcast, which also is used in the [Domain Data Grid](#) functionality, you have a robust data grid acting as a caching tier.

JMX Monitoring



JMX Monitoring in Payara Server allows you to gather information from MBeans and push it to external systems for processing and reporting without the need for external applications such as JConsole or VisualVM. [Learn more.](#)

Log Message Formatting

Log Message Formatting allows easy integration of logging messages in the Payara Platform, even when the log files are processed by an external Log Management Tool that changes the format of the logging messages.

Log Rotation

Log Rotation ensures log files are kept to a manageable size in the Payara Platform. [Learn more.](#)

MicroProfile Config

MicroProfile Config is ideal for cloud-native applications as it allows you to modify the application configuration outside your application without needing the application to be repackaged and redeployed. Payara Platform provides several out-of-the-box config sources you can use in your application to find information about the Payara runtime environment. It integrates with the [asadmin CLI](#) tool and we have also added the automatic reload of the properties after a time interval as a functionality. [Learn more.](#)

MicroProfile Config Sources

Use MicroProfile Config Sources to lookup a value associated with a configuration key. The values can be managed through the Admin Console, asadmin CLI, or the external system the value is stored in, like a Vault.

MicroProfile Fault Tolerance

Using microservices poses a new set of challenges as they need to cope with the distributed aspects. Services can be unavailable, slow to respond, or there could be network issues. With the MicroProfile Fault Tolerance policies, you can configure what needs to be done in those cases, in a declarative way. It allows you to define retries, timeouts, default values and the concept of a circuit breaker. [Learn more.](#)

MicroProfile Health Check

Using MicroProfile Health Check with Payara Platform allows developers to create their own health checks for their applications and automate the verification of their system's health status using their custom checks. [Learn more.](#)

MicroProfile JWT Authentication

MicroProfile JWT Authentication specification defines an extension on OAuth2 and OpenIdConnect to allow authentication and authorization of user requests that are secure and tokens that do not require external validation. MicroProfile JWT Authentication in Payara Platform increases the independence of applications and allows for features like single sign-on by identifying the caller in a request without having a single point of failure to validate the request. [Learn more.](#)

MicroProfile Metrics

MicroProfile Metrics provide detailed insight into the status of the application or your system. They can be the basis to determine the health of your system and make it possible to analyze the time series of values to identify trends and act before the system experiences problems. MicroProfile Metrics exposes the metrics in the Prometheus format by default (the most popular open-source product for gathering metrics). [Learn more.](#)

MicroProfile Open Tracing

The MicroProfile Open Tracing allows easier tracing of the flow of a request in microservices environments, helping you understand the different steps of a user request and why it is slow at various parts of the process. The MicroProfile Open Tracing functionality is limited to JAX-RS calls and is expanded within the Payara Platform in the [Request Tracing Service](#) which can be used with many other types of requests. [Learn more.](#)

MicroProfile OpenAPI

The MicroProfile OpenAPI provides a standard interface for documenting the exposed RESTful APIs. Documentation of your RESTful APIs is important for the parties that are using them. The OpenAPI document describes this in a standardized way and it can be generated from your application code containing the JAX-RS endpoints. [Learn more.](#)

MicroProfile Rest Client

The MicroProfile Rest Client provides a type-safe approach to invoke RESTful services over HTTP. Calling another endpoint in code is a very technical oriented task with the Jakarta EE RestClient Builder. Using the MicroProfile Rest Client, the call looks like any other JVM (Java Virtual Machine) call but in fact it is a remote call where parameters and return value are converted to and from JSON format (or the format defined on the method). [Learn more.](#)

Monitoring Service

The Payara Platform Monitoring Service identifies possible issues for troubleshooting, checks performance against specified rules, and sends alerts of potential problem areas. [Learn more.](#)

Notification Service

Notification Service works in the background to allow services such as [JMX Monitoring](#), [Payara Health Check](#), and [Request Tracing](#) to send messages to the [notifiers](#). [Learn more.](#)

Notifiers

Payara Platform Notifiers deliver monitoring and alert messages to external systems. Payara Enterprise has a more comprehensive list of notifiers so the system your company uses is likely supported. [Learn more.](#)

Notifiers API

The Notifier API allows third parties to create Payara Platform notifiers for their system and allows developers to customize notifiers for their needs. [Learn more.](#)

OSGI Deployment



If you develop OSGI modules and Web or Enterprise artifacts within your enterprise, you can run them both on Payara Server. There is no need to look for two systems and can use the functionality and features you are already familiar with.

Password Alias

Password Alias allows use of sensitive data (such as passwords) in asadmin commands and the [Admin Console](#) while keeping them hidden from users who should not have access to them. [Learn more.](#)

Payara Health Check

Payara Health Check automatically monitors the health of the system and alerts you in the event of abnormal conditions. When used with the [Notification Service](#), you can detect undesired behavior, predict possible failures, and trigger alerts to an external system with little to no impact on your application's performance.

Payara InSight



The Payara InSight monitoring console provides a visual representation of all metrics collected by Payara Server Enterprise, including [MicroProfile Metrics](#) defined via annotations, making it possible to track application-level performance and status. [Learn more.](#)

Persistent EJB Timers

Using a Persistent EJB Timer ensures an execution is performed even when the server is not running at the scheduled time. The execution is performed in that case as soon as the platform is ready again. The Payara Platform also has the option to store the information in the [Domain Data Grid](#). This removes the requirement to have a separate database to store the information. Using the Domain Data Grid also means that the information is available if one of the instances is running but the information is lost when the entire domain is down. [Learn more.](#)

Request Tracing

Request Tracing in the Payara Platform helps you understand the cause of slow application performance. It's compatible with the [MicroProfile OpenTracing](#) specification so you can use any data generated in any OpenTracing tool (such as Zipkin and Jaeger). [Learn more.](#)

REST API



The Payara Server REST API can be integrated in your applications configuration data and interact with the Payara Server configuration through a REST interface. [Learn more.](#)

REST Monitoring



REST Monitoring in Payara Server makes monitoring data accessible over a standard HTTP connection. [Learn more.](#)

Secure Administration



Secure Administration prevents unauthorized access when performing Payara Server configuration remotely by requiring an SSL encrypted connection with a username and password.

Slow SQL Logger

Detect and log slow SQL statements with the Slow SQL Logger in Payara Platform. Not only learn why some requests are slow but to also receive the executed query so you can improve performance within the application or database. [Learn more.](#)

SQL Trace Listeners

Develop custom auditing, error handling, and monitoring components with SQL Trace Listeners. Payara Platform SQL Trace Listeners track calls to the database and can be enabled globally on a data source if the class implementing it is on the server's classpath, or they can be enabled on application-specific data sources by including them in the application's WAR or EAR file. [Learn more.](#)

SQL Tracing

SQL Tracing in the Payara Platform gives you a detailed look at what the application is doing to identify the cause of issues and a view of all queries performed. [Learn more.](#)

Variable Replacement

Simplify configuration maintenance with Variable Replacement. Create a single configuration that references Environment variables, JVM system properties, and MicroProfile Config variables to adapt to each environment (test, acceptance, production, etc). [Learn more.](#)

