

Payroll Engine Web Application

☞ This application is part of the [Payroll Engine](#).

The Web Application provides full access to the Payroll Engine. For a better understanding of the working concepts, it is recommended to read the [Payroll Engine Whitepaper](#).

Features

The functions of the web app are divided into features:

Feature	Group	Description
Tasks	General	Manage the user tasks
Employee Cases	General	Add new employee case
Company Cases	General	Add new company case
National Cases	General	Add new national case
Global Cases	General	Add new global case
Reports	General	Build reports ¹⁾
Payrun Results	Payrun	View and Export payrun results
Payrun Jobs	Payrun	Start and manage payrun jobs
Payruns	Payrun	Add or change a payrun
Payrolls	Payroll	Add or change payroll and clusters
Payroll Layers	Payroll	Add or change payroll layer
Regulations	Payroll	Add or remove regulation
Regulation	Payroll	Derived regulation edit
Shared Regulations	Administration	Manage shared regulations
Tenants	Administration	Add or change a tenant
Users	Administration	Add or change an users ^{2) 3)}
Calendars	Administration	Add or change a payroll calendar
Divisions	Administration	Add or change a division
Employees	Administration	Add or change an employee
Webhooks	Administration	Add or change webhooks and messages
Logs	Administration	View the tenant logs ⁴⁾
User Storage	System	Manage the local user storage

- 1) Based on [FastReports](#).
- 2) The available functions can be assigned to the user.
- 3) The "Administrator" user type can manage all functions.
- 4) Tenant logs are generated by the regulations and should not be confused with the application log.

Web Application Server

In order to run the backend server, the web host must support the execution of .NET Core applications. Follow these steps to start the [IIS Express](#) service for local development:

- [Dotnet](#) using the binary file:

```
dotnet <PathToBin>/PayrollEngine.WebApp.Server.dll --urls=https://localhost:7179/
```

- [Dotnet](#) using the project file, using the working path [Server/](#):

```
dotnet run --urls=https://localhost:7179/
```

- Visual Studio Solution [PayrollEngine.WebApp.sln](#) using the debugger.

User Login

When logging in for the first time, the user must choose a password that complies with the following rules:

- at least 8 characters
- 1 numeric character
- 1 lower case character
- 1 upper case character
- 1 special character

Developer login

In debug mode the web application can be started with an automatic login. The desired context is defined with the program settings [AutoLogin](#), [StartupTenant](#) and [StartupUser](#). In this mode it is possible to switch between the tenants.

Application Configuration

The server configuration file [appsettings.json](#) contains the following settings:

Startup Configuration

Setting	Description	Type	Default
StartupCulture	The web application process culture	string	System culture

Setting	Description	Type	Default
<code>StartupTenant</code>	The startup tenant using the auto login option ¹⁾	string	-
<code>StartupUser</code>	The startup user using the auto login option ¹⁾	string	-
<code>AutoLogin</code>	Automatic logon using the <code>StartupTenant</code> and <code>StartupUser</code> ¹⁾	bool	false
<code>ClearStorage</code>	Clear the user storage	bool	false

¹⁾ Only in debug mode

App Configuration

Setting	Description	Type	Default
<code>AppTitle</code>	The application title	string	Payroll Engine
<code>AppImage</code>	The application image	string	Payroll Engine image
<code>AppImageDarkMode</code>	The application dark mode image	string	Payroll Engine image
<code>DarkMode</code>	Default dark mode at startup	bool?	User system setting
<code>AdminEmail</code>	The administration email for error pages	string	-
<code>ProductUrl</code>	The product url	string	-
<code>PreferredCultures</code>	Preferred cultures in dropdown lists ¹⁾	string[]	-
<code>DefaultFeatures</code>	The default features for new users	string[]	-
<code>AllowTenantSwitch</code>	Allow to switch between tenants	bool	false
<code>LogHttpRequests</code>	Log Http request to file	bool	false
<code>LogCaseChanges</code>	Add case changes to the tenant log	bool	false
<code>SessionTimeout</code>	Web application user session timeout	timespan	10 minutes
<code>ExcelExportMaxRecords</code>	Maximum count of excel export rows	int	10'000
<code>MaxDownloadSize</code>	Maximum download size	long	512'000

¹⁾ Preferred cultures examples: ["en-US", "en-AU", "de"]

Payroll Http Configuration

Setting	Description	Type	Default
<code>BaseUrl</code>	The backend base url	string	
<code>Port</code>	The backend url port	string	
<code>Timeout</code>	The backend request timeout	TimeSpan	100 seconds

Setting	Description	Type	Default
ApiKey	The backend API key	string	

The Payroll HTTP client configuration can be declared in the following locations.

Priority	Source	Description
1.	Environment variable PayrollApiConnection	Connection string with the HTTP client configuration
2.	Environment variable PayrollApiConfiguration	HTTP client configuration JSON file name
3.	File apisettings.json	HTTP client configuration JSON file located in the program folder
4.	File appsettings.json	HTTP client configuration from the program configuration JSON file

Serilog

File and console logging with [Serilog](#).

It is recommended that you save the application settings within your local [User Secrets](#).

Application Logs

The web application server stores its logs in the application folder [logs](#).

Api Key

If a key is required to access the backend API, it must be obtained from one of the following sources (in order of priority):

1. Environment variable [PayrollApiKey](#).
2. From the [Payroll HTTP configuration](#).

Input Attributes

The case input attributes can be used to control the behavior of user input.

☞ [Input Attributes Reference](#).

Docker Support

Build the Docker image:

```
docker build -t payroll-webapp .
```

Run with backend connection:

```
docker run -p 8081:5001 \
-e BackendAddress="http://localhost:5000" \
payroll-webapp
```

Verify web app is accessible at <http://localhost:8081>

Solution projects

The .NET Core application consists of the following projects:

Name	Type	Description
PayrollEngine.WebApp.Shared	Library	Shared resources
PayrollEngine.WebApp.Core	Library	Core types and services
PayrollEngine.WebApp.ViewModel	Razor Library	View model objects
PayrollEngine.WebApp.Presentation	Razor Library	Presentation components
PayrollEngine.WebApp.Server	Exe	Web application server with pages and dialogs

Third party components

- UI with [MudBlazor](#) - license [MIT](#)
- Storage with [LocalStorage](#) - license [MIT](#)
- Logging with [Serilog](#) - license [Apache 2.0](#)