|  |
| --- |
|  |
| Raumplanung |
|  |
| **Technische Dokumentation** |

**Stand: 31.08.2020, Version: 0.1**

**Inhaltsverzeichnis:**

[1 Allgemein 4](#_Toc50064029)

[1.1 Einleitung 4](#_Toc50064030)

[1.2 Gesamtsystem 4](#_Toc50064031)

[1.3 Eingesetzte Produkte 5](#_Toc50064032)

[1.3.1 Microsoft. NET Framework 5](#_Toc50064033)

[1.3.2 Microsoft .NET Core 5](#_Toc50064034)

[1.3.3 Vue.js 5](#_Toc50064035)

[2 Import (gesicherter Bereich) 6](#_Toc50064036)

[2.1 SOAP Webservice 6](#_Toc50064037)

[2.1.1 Konfigurationsmöglichkeiten 6](#_Toc50064038)

[2.1.2 Sicherheitsaspekte 8](#_Toc50064039)

[2.2 Import Clients 8](#_Toc50064040)

[2.2.1 Import CLI 9](#_Toc50064041)

[2.2.2 Import GUI 11](#_Toc50064042)

[3 Portal (öffentlicher Bereich) 13](#_Toc50064043)

[3.1 Übersicht **Fehler! Textmarke nicht definiert.**](#_Toc50064044)

[3.2 Authentifizierung 13](#_Toc50064045)

[3.2.1 Übersicht 14](#_Toc50064046)

[3.2.2 Fehlercodes 15](#_Toc50064047)

[3.3 Autorisierung 14](#_Toc50064048)

[3.4 Konfiguration WebApi 16](#_Toc50064049)

[4 Systembetrieb 16](#_Toc50064050)

[5 Anhang 17](#_Toc50064051)

[5.1 WSDL der Import-Webschnittstelle 17](#_Toc50064052)

[5.2 Klassenmodell OvgRlp.Justizportal.Import.Core.dll 17](#_Toc50064053)

**Historie**

|  |  |  |  |
| --- | --- | --- | --- |
| **Datum** | **Bearbeitet von** | **Beschreibung** | **Bemerkung** |
| 31.08.2020 | Bamberger | Erstellung | Initiale Erstellung der Dokumentation |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

# Allgemein

## Einleitung

Die neue Raumplanung des NJZKO ermöglicht es Räume transparent zu verwalten. Sie wird von der IT-Abteilung des OVG ständig weiterentwickelt. Aktuell ist es noch notwendig zur Buchung von Terminen eine E-Mail zu versenden. Bitte verwenden Sie einen aktuellen Browser wie Google Chrome oder Mozilla Firefox, da der Internet Explorer nicht mehr unterstützt wird. Unter [http://rema.ovgvg.jmrlp.de](http://rema.ovgvg.jmrlp.de/) haben Sie die Möglichkeit sich zu informieren ob Räume belegt sind und wie lange. Zusätzlich können Sie die angezeigten Räume filtern und zwischen Monats-, Wochen- und Tagesansicht wechseln. Ihre Startseite ist hierbei die Planungsübersicht. Über die Navigation an der linken Seite können Sie auf andere Ansichten wechseln. Rechts oben sehen Sie den ersten Buchstaben Ihres Namens, diesen Menüpunkt können Sie anklicken um Informationen zu Ihrem Profil im Programm zu erhalten.

## Gesamtsystem

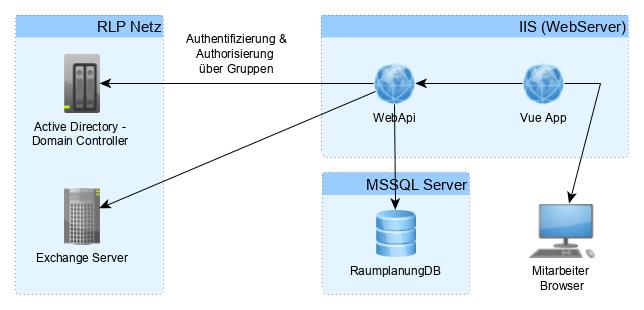


Abbildung 1: Übersicht Gesamtsystem

## Eingesetzte Produkte

### Microsoft. NET Framework

Als Framework für die Import-Komponenten kommt .NET Framework[[1]](#footnote-1) zum Einsatz. Dabei handelt es sich um eine Entwicklungsplattform zum Erstellen von Anwendungen vorrangig im Microsoft Umfeld. Es besteht aus der Common Language Runtime (CLR) und der .NET Framework-Klassenbibliothek.

Der Datenzugriff ist mit dem Entity Framework[[2]](#footnote-2) per Code-First Strategie realisiert.

### Microsoft .NET Core

Als Webframework zur Realisierung der WebApi kommt Microsoft .NET-Core[[3]](#footnote-3) zum Einsatz. Die WebApi ist grundsätzlich als REST[[4]](#footnote-4) WebService realisiert. Mit Standardfunktionen wird durchgängig auf Dependency-Injektion[[5]](#footnote-5) auf Konstruktur-Level gesetzt. Um komplexe Abhängigkeitsprobleme beim Datenaustausch zu vermeiden wird AutoMapper[[6]](#footnote-6) verwendet, dieser wandelt komplexe Objekte in flache Austauschformate und Umgekehrt.

### Vue.js

Als Front-End-Webapplikationsframework kommt Vue 2[[7]](#footnote-7) zum Einsatz. Die UI ist mit Hilfe der Komponenten Bibliothek Vuetify realisiert, der selbst definierte CSS-Code ist somit auf ein Minimum reduziert. Die Web-Anwendung ist in Form eines Fat-Client konzipiert. Bei der Initialisierung werden alle relevanten Daten geladen und per Vuex-Orm in eine Relatione Struktur gebracht, innerhalb einer Flux-Architektur (Single source of truth). Vuex-Orm greift hierbei direkt auf die Rest-Schnittstellen zu und hält somit unter Verwendung der CRUD-Operationen den lokalen Datenstand aktuell mit jenem auf dem Server. Bei jeder CRUD-Operation an einem Termin, werden die Termindatensätze aktualisiert.

# Web-API

Der Import selbst kann in 2 Bereiche, welche in den folgenden Abschnitten detailliert beschrieben werden, klassifiziert werden.

Zum einen existiert eine zentrale Web-Schnittstelle welche die zu Importierenden Akteneinsichten in einem definierten Schema entgegen nimmt.

Zum anderen existieren Clients welche dieses Schema bzw. die Web-Schnittstelle bedienen.

## SOAP Webservice

Der Webservice basiert auf SOAP[[8]](#footnote-8) und bildet die zentrale Komponente für alle zu Importierenden Akteneinsichten. Sie ist innerhalb eines Domänennetzes angesiedelt bzw. gehostet, basiert auf dem WCF-Framework (vgl. Abschnitt 1.3.1) und kapselt jegliche Zugriffe auf die zugehörige Justizportal-Datenbank.

Eine entsprechende Beschreibung der angebotenen Dienste ist dem Anhang zu entnehmen (vgl. WSDL aus Abschnitt 6.1)

### Konfigurationsmöglichkeiten

In Programmpaket des „SoapImporter“ befindet sich eine Datei „web.config“ wo diverse Konfigurationen zum Webservice vorgenommen werden können.

#### Verbindung zur Justizportal Datenbank



Abbildung 1: Verbindungszeichenfolge zur Justizportal Datenbank

Unter dem Tag „connectionStrings“ muss für „JustizportalConnectionString“ die Verbindungszeichenfolge zur Datenbank des Justizportals konfiguriert werden

#### Anwendungseinstellungen

Unter dem Tag „applicationSettings“ können weitere Anwendungseinstellungen vorgenommen werden



Abbildung 2: Anwendungseinstellungen des Soap Webservice

|  |  |
| --- | --- |
| Einstellung | Beschreibung |
| LogFile | Hier kann eine konkrete Logdatei angegeben werden wo jeder Importschritt je Import protokolliert wird.  Alternativ kann ein Verzeichnis angegeben werden, in dem Fall werden in dem Verzeichnis je eine Logdatei für „Info“, „Warning“, „Error“ und „Fatal“ angelegt. |
| LogDbConnectionString | Hier kann in Verbindung mit LogDbDatatable eine Datenbankinstanz angegeben werden.  Die Logs werden folglich in eine Datenbank protokolliert. |
| LogDbDatatable | Hier kann in Verbindung mit LogDbConnectionString eine Tabelle angegeben werden.  Die Logs werden folglich in diese Tabelle protokolliert. |
| ArchiveFileDirectory | Hier kann optional ein Archiv-Pfad für die importierten Akteneinsichten angegeben werden.  Dabei kann auf das YYYY- (für die Jahreszahl) und MM-Pattern (für die 2-Stellige Monatszahl) zurückgegriffen werden. |
| PermissionGroups | Hier müssen eine oder mehrere Sicherheitsgruppen aus dem Active Directory angegeben werden.  Anfragen an den Webservice können nur von Benutzer gestellt werden, welche sich in den jeweiligen Sicherheitsgruppen im AD befinden. |

Tabelle 1: Beschreibung Anwendungseinstellungen Import Webservice

### Sicherheitsaspekte

Grundsätzlich wurde der Import SOAP Webservice für einen Einsatz in sicheren Umgebungen konzipiert. Darüber hinaus fordert der Webservice eine Bindung mit Windows-Authentifizierung auf Nachrichtenebene[[9]](#footnote-9).

In der Konfigurationsdatei (vgl. Abschnitt 2.1.1) erfolgt die Eintragung einer oder mehrerer Sicherheitsgruppen aus dem Active Directory (AD). Wird der Webservice von einem Nutzer aufgerufen welcher nicht über die Gruppen zugehörig ist, werden Anfragen mit einer entsprechenden Fehlermeldung abgewiesen.

## Import Clients

Die Import Clients sind letztendlich dazu da, die Daten einer Akteneinsicht zu strukturieren und über die Schnittstelle des SOAP Webservice zu Importieren.

Die bis vorhandenen Anwendungen „JpImportCLI“ und „JpImportGUI“ liegen 3 verschiedene Import Typen zugrunde, ein entsprechendes Datenmodell befindet sich im Anhang (vgl. Abschnitt 6.2).

|  |  |
| --- | --- |
| ImportTyp | Beschreibung |
| Eureka | Ein spezielles Format (kombination aus PDF-Dateien und einer XML mit Metadaten), welches von der Fachanwendung Eureka-Fach ausgelagert wird.  Die Metadaten sind komplett in der XML angegeben. Ein automatisierter Import kann erfolgen. |
| EIP | Export aus Anwendung EIP. Im Exportordner liegt eine Zip Datei welche wiederum eine Ordnerstruktur.  „eAkte\*\01 - Akte\02 - Hauptakte\“ besitzt.  Die Metadaten müssen werden teilweise vorgeschlagen. |
| Native | Wenn keiner anderer Import Typ gefunden wurde greift dieser Import-Typ, welcher alle Dateien eines Ordners als Importdaten erkennt.  Die Metadaten müssen explizit angegeben werden. |

Tabelle 2: vorhandene Import-Typen

### Import CLI

Die Konsolenanwendung „JpImportCLI.exe“ bietet die Möglichkeit eine Akteneinsicht per Konsole zu importieren. Da Konsolenanwendungen eher als nicht intuitiv und nicht als Benutzerfreundlich gelten, ist die Import CLI eher für den automatisierten Import gedacht.

#### Konfigurationsmöglichkeiten

In Programmpaket der „JpImportCLI“ befindet sich eine Datei „JpImportCLI.exe.config“ wo diverse Konfigurationen inkl. der Bindung eingetragen werden können.

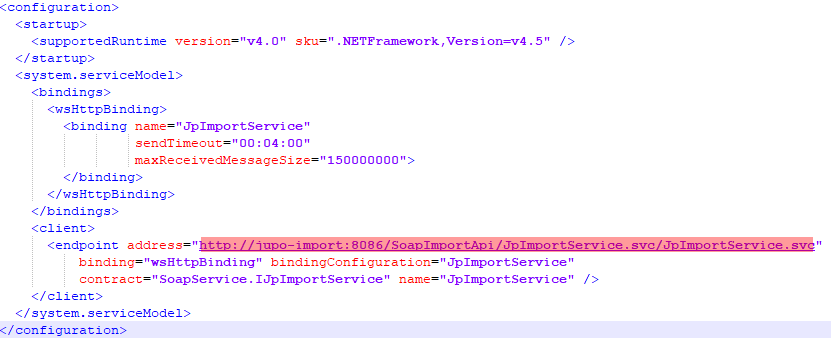


Abbildung 3: Konfigurationsmöglichkeiten Import CLI

Unter dem Tag „endpoint“ muss die Adresse des SOAP Webservice eingetragen werden.

#### Befehlsparameter

Für den Import einer Akteneinsicht existieren verschiedene Befehlsparameter, welche zur Vervollständigung je nach Import-Typ angegeben werden müssen.

|  |  |
| --- | --- |
| Parameter | Beschreibung |
| -i, --import | Hier kann ein Verzeichnis-Pfad mit einer Ansammlung von Import Dateien angegeben werden. |
| --fi, --fileimport | Datei-Pfad für eine einzelne Datei zum Import. |
| --az, --aktenzeichen | Aktenzeichen der Akteneinsicht. |
| --kr, --kurzrubrum | Kurzrubrum der Akteneinsicht. |
| --bn, --benutzername | Benutzername für die Akteneinsicht. |
| --pw, --passwort | Passwort für die Akteneinsicht. |
| --bd, --bereitstellung | Bereitstellungsdatum der Akteneinsicht, wenn nicht angegeben wird das Tagsdatum als default genommen. |
| --ad, --ablauf | Ablaufdatum der Akteneinsicht. |
| -h, -?, --help | Hilfe anzeigen. |
| -v, --version | Anzeige der aktuellen Import CLI Programmversion. |

Tabelle 3: Befehlsparameter der Import CLI

### Import GUI

Die Grafische Anwendung „JpImportGUI“ bietet dem Endanwender die Möglichkeit eine Akteneinsicht manuell zu Importieren. Die Anwendung kann entweder manuell oder mit einem Verzeichnis als Parameter gestartet werden. Die Metadaten können in der Oberfläche überschrieben werden. Für das Passwort steht ein kleiner Passwortgenerator (Button neben dem entsprechenden Feld) zur Verfügung.

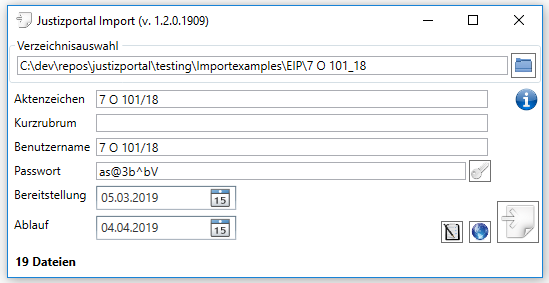
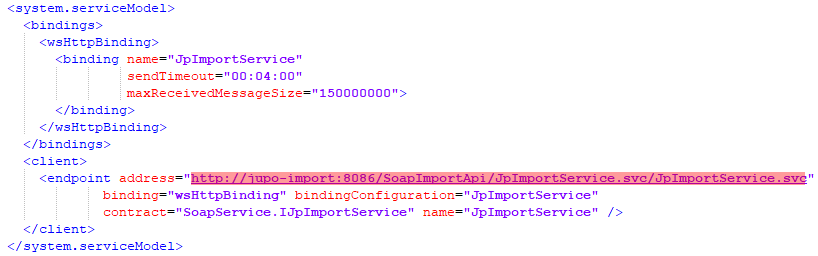


Abbildung 4: Programmmaske Import GUI

#### Konfigurationsmöglichkeiten

In Programmpaket der „JpImportGUI“ befindet sich eine Datei „JpImportGUI.exe.config“ wo diverse Konfigurationen inkl. der Bindung eingetragen werden können.



Unter dem Tag „endpoint“ muss die Adresse des SOAP Webservice eingetragen werden.

#### Anwendungseinstellungen

Weiterhin können in der Datei „JpImportGUI.exe.config“ unter dem Tag „applicationSettings“ können weitere Anwendungseinstellungen vorgenommen werden

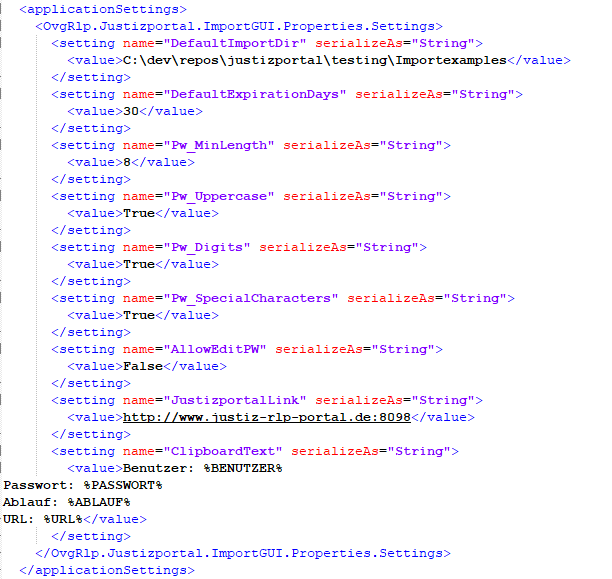


Abbildung 5: Anwendungseinstellungen der Import GUI

|  |  |
| --- | --- |
| Einstellung | Beschreibung |
| DefaultImportDir | Hier kann ein Verzeichnis angegeben werden, welches im Auswahldialog der Verzeichnisauswahl vorgeschlagen wird. |
| DefaultExpirationDays | Hier kann eine Anzahl von Tagen eingetragen werden für den Vorschlag zur Dauer der Akteneinsicht. |
| ClipboardText | Hier kann ein freier Text hinterlegt werden, welcher mittels einer Schaltfläche in den Zwischenspeicher geladen wird. Im Text können die folgende Variablen verwendet werden:  %AKTENZEICHEN%  %KURZRUBRUM%  %BENUTZER%  %PASSWORT%  %BEREITSTELLUNG%  %ABLAUF%  %URL% |
|  |  |
|  |  |

Tabelle 4: Beschreibung Anwendungseinstellungen

# Grafische Benutzerschnittstelle

Zur Verwendung wird der Google Chrome Browser empfohlen aufgrund seiner Verwendung in der Justiz und seines aktuellen und sicheren Funktionsumfangs. Andere moderne Browser funktionieren größtenteils korrekt, allerdings ist mit Funktionseinschränkungen zu rechnen. Der Internet Explorer wird explizit nicht unterstützt. Im Mozilla Firefox funktioniert die „Termin drucken“ Funktion nur über den Umweg eines automatischen Downloads, im MS Edge funktioniert sie gar nicht.

## Authentifizierung

Die Authentifizierung erfolgt passiv über das Active Directory. Es ist keine aktive Handlung des Nutzers erforderlich. Zugriffe sind nur von angemeldeten Nutzern innerhalb der Domain möglich, durch diese Authentifizierung ist auch eine Identifizierung und Authentifizierung der Nutzer realisiert. Beim ersten „kontakt“ der Anwendung mit einem Nutzer, wird für diesen ein Nutzerkonto in der Datenbank erstellt. Ein solcher Kontakt kann zB. durch den Aufruf der Webseite durch den Benutzer entstehen oder auch durch das Auswählen eines Nutzers als Ansprechpartner für einen Termin. Gespeichert wird neben Namen und Organisationseinheit auch die ID des Nutzers im AD. Diese ermöglicht eine Verbindung zwischen den Informationen aus der AD und dem Nutzer Datensatz in der DB.

## Autorisierung

Generell kann jeder Nutzer innerhalb der Domain Termine ansehen, zur Aufteilung der Berechtigungen existieren drei Rollen, diese müssen Gruppen in der AD zugewiesen werden. Die Berechtigung von

Bei allen Anfragen an die WebApi, außer an den Endpunkt „./token“ muss vom WebClient ein gültiges Token mitgegeben werden. Sollte kein Token mitgegeben werden, das mitgegeben Token nicht mehr gültig sein oder der zugehörige Benutzer keinen Zugriff auf die Ressource haben, wird die Anfrage mit dem http-Statuscode 401 (Unauthorized) zurückgewiesen.

### Übersicht

Um Zugriff auf die verschiedenen Ressourcen der WebApi zu erlangen, muss der WebClient sich am Endpunkt „./token“ authentifizieren. Dieser Endpunkt ist öffentlich zugänglich und erwartet die Übergabe eines Benutzernamens und Passwortes.

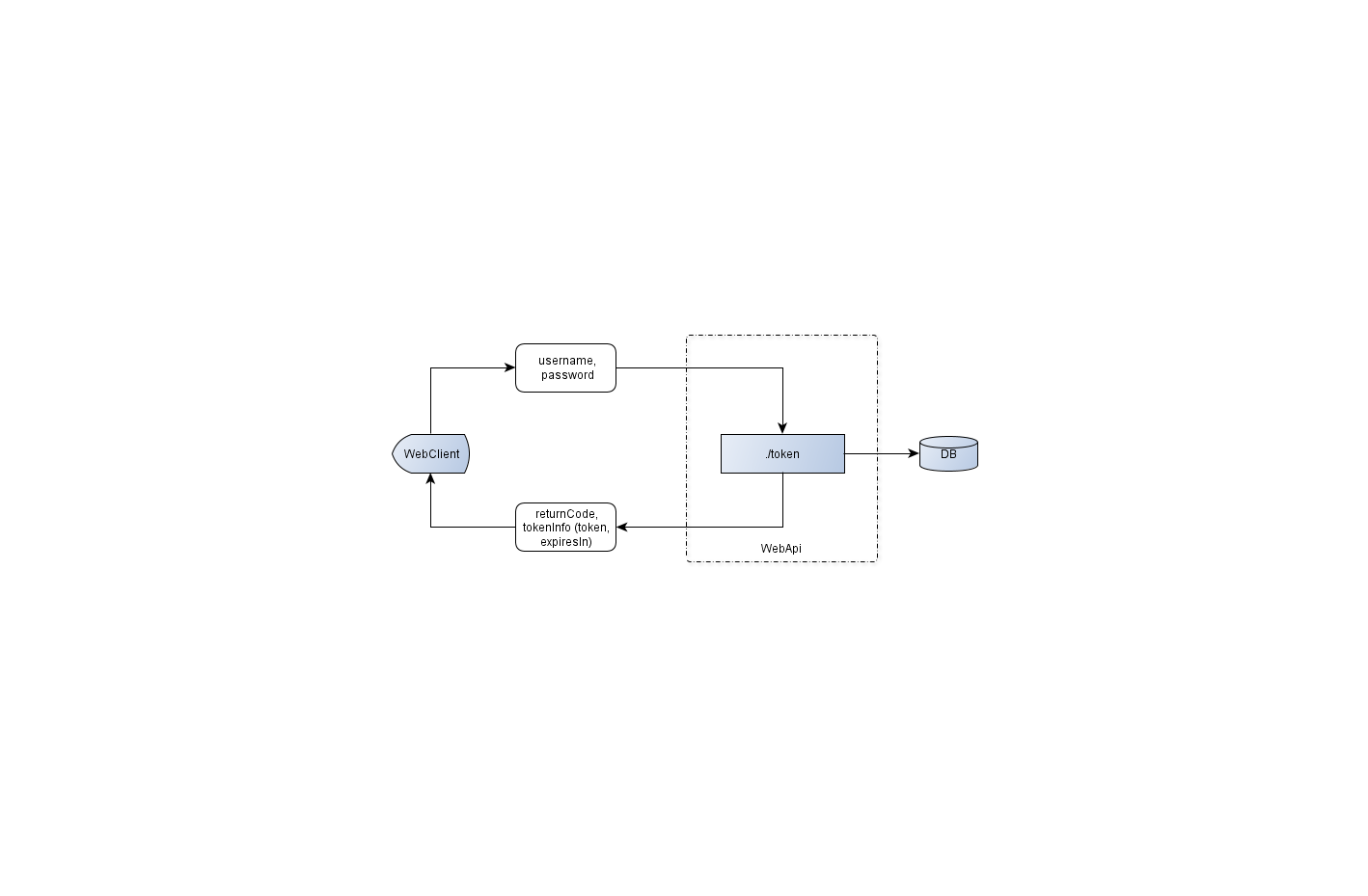


Abbildung 6: Authentifizierung WebClient gegen öffentliche WebApi

Wenn die Authentifizierung erfolgreich ist, erhält der WebClient ein gültiges Token und die Information, wann dieses abläuft.

### Fehlercodes

Wenn die Anmeldung nicht erfolgreich ist, wird dem Client ein Fehlercode und zusätzliche Informationen zurückgegeben.

|  |  |
| --- | --- |
| Fehlercode | Beschreibung |
| ERROR\_WRONG\_CREDENTIALS | Benutzername oder Kennwort sind falsch. Als zusätzliche Information wird die Anzahl der verbleibenden Versuche (remaining) bis zur Sperrung des Accounts übergeben. |
| ERROR\_USER\_EXPIRED | Benutzername und Kennwort sind zwar richtig aber die Akteneinsicht ist bereits abgelaufen. |
| ERROR\_USER\_LOCKED | Benutzername und Kennwort wurden zu oft falsch eingegeben und der Account wurde gesperrt. Als zusätzliche Information wird die Dauer in Sekunden (unlockedIn) zurückgegeben wenn der Account wieder freigeschaltet wird. |

Tabelle 5: Fehlercodes Authentifizierung

## Konfiguration WebApi

todo: Beschreiben der einstellbaren Parameter der WebApi

# Installation

Um das Raumplanungssystem zu installieren werden folgende Komponenten vorrausgesetzt:

* Eine MSSQL Datenbank Instanz, auf welcher das Datenbankschema aufgesetzt wird.
* Ein Aktive Directory mit Gruppen für die Rollen:
  + Administrator
  + Bearbeiter
  + Benutzer

Wobei auch mehrere Rollen einer Gruppe zugeordnet werden können.

* Einen Webserver, bisher IIS, welcher die .Net Core Anwendung ausführen und die Web-App ausliefern kann.
* Einen Email-Server sowie ein Konto welches von diesem E-Mails verschicken kann.

# Updates

Um Updates möglichst einfach zu ermöglichen gibt es im vue-Projektordner ein script innerhalb der package.json: „npm run build-all“.

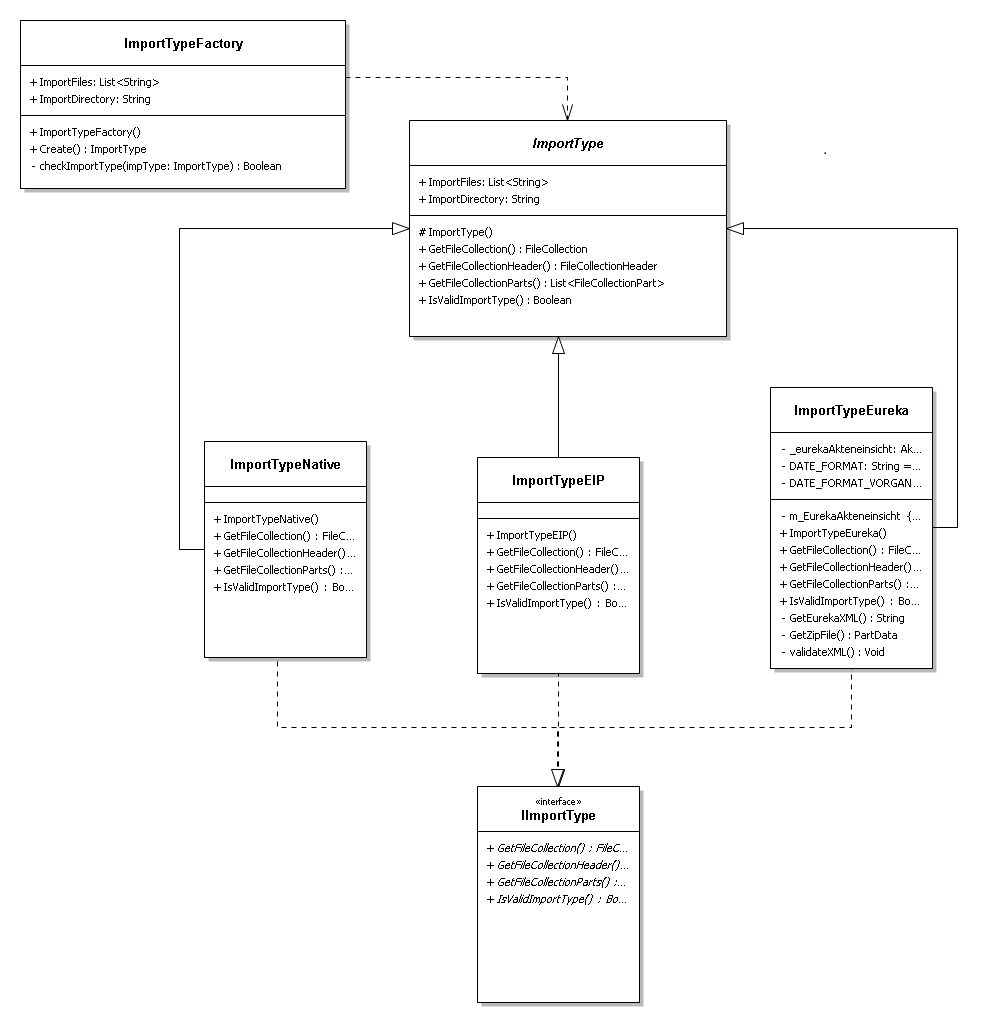
Dieses erstellt sowohl die Web-Anwendung als auch die Webserver Software, kopiert alles in einen Ordner und erstellt im Wurzelverzeichnis ein Zip-Archiv mit Namen und Versionsnummer des Programms. Die erforderlichen Konfigurationsdatei sind innerhalb diese Archivs umbenannt, sie enthalten das Wort „template“ zwischen Dateinamen und Dateiendung. Um eine neuen Instanz aufzusetzen müssen diese Dateien also umbenannt und mit eigenen Daten gefüllt werden. Falls sich mit einem Update die Inhalte der Konfigurationsdateien ändern können diese umbenannten Konfigurationsdateien als Referenz herangezogen werden.

Der Inhalt des Zip-Ordners kann also zum Updaten einfach in das Wurzelverzeichnis einer Instanz kopiert werden. Falls Konfigurationsänderungen oder Datenbankschemaanpassungen notwendig sind müssen diese noch vorgenommen werden. Das Datenbankschema lässt sich updaten durch Verwendung des aktuellsten Update-Scripts im Ordner „Rema.DbAccess -> Migrations“ des Projekts.

# Anhang

## WSDL der Import-Webschnittstelle

## Klassenmodell OvgRlp.Justizportal.Import.Core.dll



1. <https://docs.microsoft.com/de-de/dotnet/framework> [↑](#footnote-ref-1)
2. <https://docs.microsoft.com/de-de/ef/core/> [↑](#footnote-ref-2)
3. <https://docs.microsoft.com/de-de/dotnet/core> [↑](#footnote-ref-3)
4. <https://de.wikipedia.org/wiki/Representational_State_Transfer> [↑](#footnote-ref-4)
5. <https://de.wikipedia.org/wiki/Dependency_Injection> [↑](#footnote-ref-5)
6. <https://automapper.org/> [↑](#footnote-ref-6)
7. <https://vuejs.org/> [↑](#footnote-ref-7)
8. <https://de.wikipedia.org/wiki/SOAP> [↑](#footnote-ref-8)
9. <https://docs.microsoft.com/de-de/dotnet/framework/wcf/samples/message-security-windows> [↑](#footnote-ref-9)