

Kontrakte

Operation: `signIn(Id:string, Password:string)` : Authentifizierungsobjekt

Beschreibung: Prüft die Anmeldedaten und gibt dem Nutzer ein Authentifizierungsobjekt

Vorbedingung:

- Der Nutzer ist nicht angemeldet
- Der Nutzer verfügt über eine Id und ein Passwort
- Die Id und das Passwort werden mit einem Nutzerprofil assoziiert

Nachbedingung:

- Das Ergebnisobjekt wurde zurückgegeben
- Der Server hat Id und Passwort mit einem Nutzerprofil assoziiert
- Der Nutzer hat Zugriff auf die Anwendung entsprechend der Zugriffsrechte des Nutzerprofils

Ausnahme: Die Daten existieren nicht im System, Server ist überlastet

Ergebnis: Authentifizierungsobjekt

Operation: `signOut()`

Beschreibung: Der Nutzer wird von der Anwendung abgemeldet

Vorbedingung:

- Der Nutzer ist angemeldet

Nachbedingung:

- Der Nutzer hat (ohne erneute Anmeldung) keinen Zugriff auf Anwendung
 - Der Nutzer befindet sich auf der Anmeldeseite
-

Operation: updateProfile(FirstName:string, LastName:string, Picture:string, Rate-Card-Level:integer, Skills:string, Projects:string, Id:string)

Beschreibung: Übernimmt die Änderungswünsche des Nutzers an dessen/deren Profil, speichert diese und gibt einen Hinweistext zurück

Vorbedingung: Die Daten konnten an das Modell mit dessen Validierungs-Attributen gebunden werden, d.h.:

- Das Bild ist valide (Auflösung und Größe entsprechend dem korrekten Format)
- Die Veränderung erfüllt das Eingabeformat (First-/LastName nur aus Buchstaben, Pflichtfelder nicht-leer)
- Pflichtfelder sind FirstName, LastName, Rate-Card-Level
- ID darf nicht verändert werden
- Es wurde eine "echte" Änderung vollzogen (mindestens ein Detail wurde verändert)

Nachbedingung:

- Alle eingegeben Daten wurden übernommen und mit den zugehörigen Feldern (FirstName-> Vorname) im (aktualisierten) Nutzerprofil assoziiert
- Falls sich das Bild geändert hat, wurde es hochgeladen und verlinkt

Ausgabe: „Ihre Änderungen wurden übernommen“

Ausnahme: Eingabe an der Stelle ... nicht erlaubt

Operation: createOffer(Title:string)

Beschreibung: Erstellt ein neues Angebot mit dem Titel „Title“

Vorbedingung:

- Es liegt kein Angebot mit diesem Namen vor
- Titel muss aus Buchstaben und Ziffern bestehen (erlaubte Zusatzzeichen: '-', '/', '&', '_', '#', '+')
- Der Nutzer hat die entsprechenden Zugriffsrechte, um ein Angebot erstellen zu können

Nachbedingung:

- Ein Angebot mit dem Titel wurde erstellt

Ausgabe: "Angebot wurde erfolgreich erstellt!"

Ausnahme: Angebot mit selben Namen bereits vorhanden; kein Titel

Operation: deleteOffer(Offer:offer)

Beschreibung: Sucht und löscht übergebenes Angebot

Vorbedingung:

- Es existiert ein Angebot mit den Attributen des Objekts

Nachbedingung:

- Angebot liegt nicht mehr vor
- dazugehörige Dokumentenkonfiguration wurde gelöscht
- verknüpfte Mitarbeiterprofil-Kopien wurden gelöscht

Ausgabe: „Angebot wurde gelöscht“

Ausnahme: Das Angebot existiert nicht mehr (in dieser Form)

Operation: updateOffer(Offer:offer)

Beschreibung: Übernimmt die Änderungswünsche für das übergebene Angebot

Vorbedingung:

- Die Daten wurden an das Modell mit dessen Validierung Notizen gebunden
- Es existiert ein Angebot mit angegebenem Namen
- Eingaben müssen dem Eingabeformat entsprechen
- Titel muss aus Buchstaben und Ziffern bestehen (erlaubte Zusatzzeichen: '-', '/', '&', '_', '#', '+')

Nachbedingung:

- Alle eingegeben Daten wurden übernommen und mit den zugehörigen Feldern (Offer.Titel-> Angebot.Titel) im (aktualisierten) Angebot assoziiert
- Altes Angebot existiert nicht mehr

Ausgabe: „Ihre Änderungen wurden übernommen“

Ausnahme: Das Angebot existiert nicht • kein Titel

Operation: Add(Offer:offer, Employee:employee)

Beschreibung: Speichert eine Kopie der Daten des Mitarbeiters im Angebot, die auch die Originaldaten referenziert

Vorbedingung:

- Das Angebot und der Mitarbeiter existieren

Nachbedingung:

- Die Mitarbeiterliste des Angebots wurde um den Mitarbeiter mit seinen Profildaten erweitert.

Ausgabe: „Der Mitarbeiter <e.Name> wurde zum <O.Name> hinzugefügt“

Ausnahme: Angebot und Mitarbeiter konnten nicht mehr gefunden werden

Operation: Remove(Offer:offer, Employee:employee)

Beschreibung: Löscht den Mitarbeiters aus dem Angebot

Vorbedingung:

- Der Mitarbeiter ist Teil des Angebots

Nachbedingung:

- Der Mitarbeiter ist nicht mehr in der Mitarbeiterliste des Angebots.

Ausgabe: „Der Mitarbeiter <e.Name> wurde entfernt“

Ausnahme: Die Assoziation existiert schon nicht mehr

Operation: updateDocumentConfig(Offer:offer)

Beschreibung: Ändert/erstellt für das Angebot eine Konfigurations-Optionen. Der Nutzer entscheidet durch "die Auswahl von Elementen" was im Dokument angezeigt wird und legt deren Reihenfolge fest

Vorbedingung:

- Die Daten wurden an das Modell mit dessen Validierung Notizen gebunden
- Angebot existiert
- Rate-Card-Level enthält nur eine gültige Zahl (1-5)

Nachbedingung:

- es wurde eine Konfiguration erstellt/geändert und diese im Angebot gespeichert

Operation: generateDocument(Offer:offer): Document

Beschreibung: Erstellt anhand des Angebots und der Konfiguration eine Word Datei

Vorbedingung:

- das Angebot mit der dazugehörigen Configuration existiert
- Configuration ist valide

Nachbedingung:

- es wurde ein Dokument generiert **Ergebnis:** Word Dokument

Operation: createProject(Project:project)

Beschreibung: Erstellt ein neues Projekt mit dem Namen <project.Title> und den eingegebenen Daten

Vorbedingung:

- Es liegt kein Projekt mit diesem Namen vor
- es wurde ein Titel mit passendem Format (bestehend aus Buchstaben und Zahlen und den erlaubten Zusatzzeichen: '-', '/', '&', '_', '#', '+') eingegeben
- Die Beschreibung ist höchstens 400 Zeichen lang
- Das Startdatum liegt vor dem Enddatum, beide liegen zwischen dem 01.01.2011 und dem 01.01.2100
- die Projektzwecke bestehen aus mindestens zwei und höchstens 200 Zeichen
- die Projekttätigkeiten bestehen aus mindestens zwei und höchstens 50 Zeichen

Nachbedingung:

- Alle eingegeben Daten wurden übernommen und mit den zugehörigen Feldern (Project.Title-> Projekt.Titel) im (aktualisierten) Projekt assoziiert

Ausnahme: Projekt mit selben Namen bereits vorhanden

Operation: deleteProject(Project:project) (optional)

Beschreibung: sucht und löscht übergebenes Projekt und löscht aus jedem Profil den Projektnamen

Vorbedingung:

- Nutzer hat Vertriebs-/Adminrechte
- es existiert ein Projekt mit angegebenem Namen

Nachbedingung:

- Projekt liegt nicht mehr vor

Ausgabe: Hinweistext „Projekt wurde gelöscht“

Ausnahme: kein Projekt liegt vor • Fehler beim Löschen auf Datenbank

Operation: updateProject(Project:project)

Beschreibung: einem Nutzer hat Änderungswünsche an ein Projekt übergeben, das System speichert diese und gibt einen Hinweistext zurück

Vorbedingung:

- Nutzer hat Rechte zum bearbeiten eines Projekts
- Die Daten wurden an das Modell mit dessen Validierung Notizen gebunden
- es existiert ein Projekt mit angegebenem Namen
- Eingabe entsprechend passendem Format (bestehend aus Buchstaben und Zahlen und den erlaubten Zusatzzeichen: '-', '/', '&', '_', '#', '+')

Nachbedingung:

- Alle eingegeben Daten wurden übernommen und mit den zugehörigen Feldern (Project.Title-> Projekt.Titel) im (aktualisierten) Projekt assoziiert

Ergebnis: Projekt

Ausgabe: Hinweistext „Ihre Änderungen wurden übernommen“

Ausnahme: kein Projekt liegt vor • Änderung nicht erlaubt (leeres Namensfeld etc)

Operation: updateBasicDataSet(JSON:JSON)

Beschreibung: Der Nutzer verändert mit Hilfe einer JSON Datei die Datenbasis des Systems

Vorbedingung:

- Der Nutzer hält eine zur Datenbasis unterschiedliche datenbasis.json auf seinem Rechner
- Der Aufbau dieser widerspricht nicht den Regeln des Systems.
(Format gleich der datenbasis.json das bereits im System existiert)

Nachbedingung:

- Das System hat die Datenbank entsprechend aktualisiert (Daten in den passenden Tabellen und Feldern vorhanden).
- Falls Skills gelöscht wurden, wurden
 - diese auch bei allen Mitarbeitern
 - und allen Angeboten gelöscht

Ausgabe: „Ihre Änderungen wurden übernommen“

Ausnahme: JSON konnte nicht deserialisiert werden