

# Kontrakte

Operation	<b>makeBuchungsvorschlag(bw:Buchungswunsch)</b>
Beschreibung	Berechnet mögliche Buchungsvorschläge, die den Buchungswunsch erfüllen und nicht mit dem Gesamtbelegungsplan in Konflikt stehen.
Vorbedingung	Der zu bw assoziierte Buchungswunsch ist erfüllbar. Der Gesamtbelegungsplan ist konsistent.
Nachbedingung	Eine Menge von Buchungsvorschlagsobjekten set(bv) wurde generiert. Buchung wurde ohne Konflikte erstellt.
Ergebnisse	Menge von Buchungsvorschlägen set(bv)
Ausnahmen	~
Ausgaben	~
Typ	Systemoperation
Querverweise	~
Anmerkungen	~

Operation	<b>starteSimulation(js:JSON-Datei)</b>
Beschreibung	Startet die Simulation abhängig von der JSON-Datei.
Vorbedingung	Die zu js assoziierte JSON-Datei ist erfüllbar.
Nachbedingung	Simulation wird mit den richtigen Parametern der JSON-Datei gestartet. Es muss ein Ergebnis geben.
Ergebnisse	Auslastungsszenario
Ausnahmen	~
Ausgaben	~
Typ	Systemoperation
Querverweise	~
Anmerkungen	~

Operation	gebeStandortdatenZurück()
Beschreibung	Sucht mit Standort zusammenhängende Informationen zusammen und gibt sie zurück.
Vorbedingung	Gültiger Standort wurde ausgewählt.
Nachbedingung	<ul style="list-style-type: none"> <li>- Standort wurde mit darin befindlichen Ladezonen in Verbindung gesetzt;</li> <li>- Steckertypen aller dieser Ladezonen wurden aufgelistet;</li> <li>- Ladesäulen innerhalb des Standorts wurden gezählt;</li> <li>- Auslastung des Standorts wurde aus Buchungsplan herausgezogen.</li> </ul>
Ergebnisse	Anzahl der Ladesäulen, Informationen zu Ladezonen innerhalb des Standorts, vorhandene Steckertypen und Auslastung des Standorts.
Ausnahmen	~
Ausgaben	~
Typ	Systemoperation
Querverweise	makeBuchungsvorschlag(bw:Buchungswunsch)
Anmerkungen	~

Operation	visualisiereSimulation()
Beschreibung	Stellt Informationen aus Simulation graphisch dar.
Vorbedingung	Simulation wurde erfolgreich durchgeführt.
Nachbedingung	Aus Simulation gewonnene Informationen wurden mithilfe von Graphen, Tabellen und Diagrammen dargestellt.
Ergebnisse	Graphen, Diagramme, Tabellen.
Ausnahmen	~
Ausgaben	~
Typ	Systemoperation
Querverweise	starteSimulation(), simuliereBuchungen(), evaluiereAuslastungsszenario(Berechnung des Auslastungsszenarios)
Anmerkungen	~

Operation	<b>deleteBuchung(bu:Buchung)</b>
Beschreibung	Löscht die ausgewählte Buchung und gibt die Ressourcen (Zeitraum, Ladesäule) wieder frei.
Vorbedingung	Buchung existiert und liegt in der Zukunft.
Nachbedingung	Buchung wurde gelöscht.
Ergebnisse	Boolean
Ausnahmen	~
Ausgaben	~
Typ	Systemoperation
Querverweise	~
Anmerkungen	~

Operation	<b>generiereBenutzer(mail:String, id:String, pw:String, role:int)</b>
Beschreibung	Legt einen Benutzer mit den übergebenen Parametern an und pflegt sie ins System ein.
Vorbedingung	Benutzer existiert noch nicht. Constraints der Parameter müssen erfüllt sein.
Nachbedingung	Benutzer wurde erstellt.
Ergebnisse	Boolean
Ausnahmen	~
Ausgaben	~
Typ	Systemoperation
Querverweise	~
Anmerkungen	~

Operation	<b>editBuchung(bu:Buchung, la:Ladestand, nd:Nötige Distanz, st:Start, en:Ende)</b>
Beschreibung	Aktualisiert eine bestehende Buchung mit einem oder mehreren geänderten Parametern.
Vorbedingung	Eine valide Buchung besteht, neue Buchung ist im Rahmen des möglichen.
Nachbedingung	Eine aktualisierte valide Buchung wurde bearbeitet, und hat jetzt neue Parameterwerte.
Ergebnisse	Liste der Buchungen
Ausnahmen	~
Ausgaben	~
Typ	Systemoperation
Querverweise	legeBuchungAn(bv:Buchungsvorschlag), deleteBuchung(bu:Buchung)
Anmerkungen	~

Operation	<b>sendeTermin(be:Benutzer, st:Start, en:Ende)</b>
Beschreibung	Sendet einen Termin an einen Benutzer.
Vorbedingung	Benutzer hat eine Buchung akzeptiert.
Nachbedingung	Benutzer hat eine Terminnotification bekommen, Buchung wurde ohne Konflikte erstellt.
Ergebnisse	Erfolg(Boolean)
Ausnahmen	~
Ausgaben	Terminnotification an Benutzer
Typ	Systemoperation
Querverweise	legeBuchungAn(bv:Buchungsvorschlag)
Anmerkungen	~

Operation	<b>starteSimulation(dauer, Fahrzeuganzahl, Fahrzeugtypen, Ladezone, Standort, Streuung)</b>
Beschreibung	Berechnet eine Simulation die durch einen Verwalter gestartet wird mit Paramatern die durch diesen eingegeben wurden.
Vorbedingung	Es gibt einen Planer, es existiert mindestens eine Ladezone worauf das angewendet wird, alle Eingaben sind Konsistent.
Nachbedingung	Es muss ein Ergebnis geben egal ob zuviele Buchungen gab oder zu wenige.
Ergebnisse	Berechnung Auslastung
Ausnahmen	~
Ausgaben	Berechnung wird an evaluiereAuslastungsszenario geschickt.
Typ	Systemoperation
Querverweise	~
Anmerkungen	~

Operation	<b>evaluiereAuslastungsszenario(Berechnung des Auslastungsszenarios)</b>
Beschreibung	Wertet das Auslastungsszenario aus.
Vorbedingung	Es muss ein Auslastungsszenario geben.
Nachbedingung	Es muss Konsitent sein.
Ergebnisse	Information für die Visualisierung des Auslastungsszenarions.
Ausnahmen	~
Ausgaben	~
Typ	Systemoperation
Querverweise	starteSimulation()
Anmerkungen	~

Operation	<b>legeBuchungAn(bv:Buchungsvorschlag)</b>
Beschreibung	Nutzer wählt einen der Buchungsvorschläge aus.
Vorbedingung	<ul style="list-style-type: none"> <li>- Vorschläge passen zum Gesamtbelegungsplan;</li> <li>- Gesamtbelegungsplan ist konsistent;</li> <li>- Vorschläge müssen auch trotz parallelem Zugriff korrekt verteilt werden.</li> </ul>
Nachbedingung	Gesamtbelegungsplan ist konsistent. Buchungsvorschlag wurde akzeptiert.
Ergebnisse	Visuelles Feedback ob Buchung erfolgreich angelegt.
Ausnahmen	Durch parallele Zugriff und oder zu langes Warten wurde der Vorschlag an mehrere Nutzer gesendet und ausgewählt. Dann darf die Buchung nur bei einem eingetragen werden.
Ausgaben	~
Typ	Systemoperation
Querverweise	makeBuchungsvorschlag(bw:Buchungswunsch)
Anmerkungen	~

Operation	<b>generiereBuchungen(Ladezonenplan, Anzahl Fahrzeuge, Batterie, SoC Start, Soc Ziel, Verteilung(Stoßzeiten, min/max Streuung), Tickzeit)</b>
Beschreibung	System generiert Abfolge von Buchungen um definierten Ladezonenplan auszuwerten.
Vorbedingung	Konstanz bei den Parametern.
Nachbedingung	~
Ergebnisse	Eine zeitliche Abfolge von Buchungen, die der definierten Verteilung entspricht
Ausnahmen	~
Ausgaben	~
Typ	Systemoperation
Querverweise	starteSimulation(dauer, Fahrzeuganzahl, Fahrzeugtypen, Ladezone, Standort, Streuung)
Anmerkungen	~