

Beschreibung HGKV- Services (V3):

Diese Beschreibung gilt für die Services V3

Die Services können aktuell über die Adresse <http://services.hgkv.at/hgkvservice-300.asmx> aufgerufen werden. (hier können auch beispielhafte Aufrufe des Service angezeigt werden.)

Die WSDL des Services erreichen sie <http://services.hgkv.at/hgkvservice-300.asmx?wsdl>

Zugriff auf das Service

Alle Services müssen mit einem zusätzlichen Header aufgerufen werden:

```
<AuthenticationHeader xmlns="http://db.hgkv.at/">  
  <UserName>[ihr Benutzername]</UserName>  
  <Password>[ihr Kennword]</Password>  
</AuthenticationHeader>
```

Kurzinformation und Ablauf der Aufrufe:

- 1) **GetContracts:** liefert eine Liste der verfügbaren Verträge
 - a. **GetContractInfo:** liefert eine Blitzinfo des gewählten Vertrags
 - b. **GetContractContent:** liefert das Textinhaltsverzeichnis des Vertrags
 - i. **GetContractText:** liefert den jeweiligen Text, -1 für gesamten Text
 - c. **GetContractBonusGroups:** liefert eine Liste aller Zulagen des Vertrags
 - i. **GetContractBoni:** eine Liste aller Zulagen für die jeweilige Gruppe, nach Datum
 - ii. **GetContractBoniById** analog GetContractBoni, Abfrage nach uniqueid
 - d. **GetContractWageGroups:** liefert eine Liste der Lohngruppen
 - i. **GetContractWages:** liefert alle Löhne einer Lohngruppe
 - ii. **GetContractWage:** liefert eine Lohngruppe
 - iii. **GetContractWagesById** analog GetContractWages, Abfrage nach uniqueid
 - iv.
 - e. **GetContractMilageGroups:** liefert eine Liste der Kilometergeld Gruppen
 - i. **GetContractMilages:** Liefert das Kilometergeld pro Gruppe
 - ii. **GetContractMilages** analog GetContractMilages, Abfrage nach uniqueid
 - iii.
 - f. **GetContractUpdates:** Liefert eine Liste der Updates zum betroffenen Vertrag
- 2) **GetUpdates:** Liefert eine komplette Liste der Updates der betroffenen Verträge.

Administrative Methoden:

- 3) **GetCustomerContracts:** Alle verfügbaren KVs anzeigen, je Parameter im oder außerhalb des Pakets
- 4) **AddCustomer:** Einen Kunden für ihr Unternehmen anlegen

5) **AddContract**: Einen Vertrag für den jeweiligen Kunden anlegen

Methoden:

GetContracts :

Aktuelle Liste aller KVs mit Bundesländern

Parameter: keine

Rückgabe: `ContractInfo[]`

GetContractInfo :

Blitzinfo eines KVs anzeigen.

Parameter: `int contractId`

Rückgabe: `Info`

GetContractContent :

Text- Inhaltsverzeichnis eines KVs anzeigen

Parameter: `int contractId`

Rückgabe: `ContractInfo[]`

GetContractText :

Text eines KVs abrufen, der Index kann mit -1 angegeben werden, damit alle Absätze aufgerufen werden.

Parameter: `int contractid,`
`int index (Absatz Nummer, aus GetContractContent)`

Rückgabe: `String (HTML)`

GetContractBonusGroups :

Die aktuelle Liste aller Zulagen eines Vertrags

Parameter: `int contractId`

Rückgabe: `ContractInfo[]`

GetContractBonni :

Die aktuelle Liste aller Zulagen eines Vertrags, nach Gruppen

Parameter: `int contractId`
`int groupId (aus GetContractBonusGroups)`
`DateTime validFrom (gültig ab Datum)`

Rückgabe: `WagesInfo []`

GetContractBonById :

Die aktuelle Liste aller Zulagen eines Vertrags, nach Gruppen

Parameter: `int uniqueId`
`DateTime validFrom (gültig ab Datum)`

Rückgabe: `WagesInfo []`

GetContractWageGroups :

Lohngruppen eines KVs abrufen.

Parameter: `int contractId` (KV Nummer)

Rückgabe: `ContractInfo[]`

GetContractWage

Lohn einer angegebenen Lohngruppe eines KVs zu einem bestimmten Zeitpunkt abrufen

Parameter: `int contracted` (KV Nummer),
`int groupId` (Lohngruppen Nummer aus `GetContractWageGroups`), #
`DateTime validFrom` (gültig ab Datum),
`int step` (Progression)

Rückgabe: `WagesInfo`

GetContractWagesById :

Lohn einer angegebenen Lohngruppe mit allen Progressionsstufen eines KVs zu einem bestimmten Zeitpunkt abrufen.

Parameter: `int uniqueId` (KV Nummer),
`DateTime validFrom` (gültig ab Datum)

Rückgabe: `WagesInfo[]`

GetContractMilageGroups :

Kilometergeld Gruppen

Parameter: `int contractId`

Rückgabe: `ContractInfo[]`

GetContractMilages :

Kilometergeld pro Guppe

Parameter: `int contractId`
`int groupId` (Kilometergeldgruppen Nummer aus `GetContractMilageGroups`),
`DateTime validFrom` (gültig ab Datum)

Rückgabe: `ContractInfo[]`

GetContractMilagesById :

Kilometergeld pro Guppe

Parameter: `int uniqueId`
`DateTime validFrom` (gültig ab Datum)

GetContractUpdates :

Vertrags Updates

Parameter: `int contractId`

Rückgabe: `UpdateInfo[]`

GetUpdates :

Alle Kv Update Informationen

Parameter: `int maximumDisplay`

Rückgabe: `UpdateInfo[]`

Administrative Methoden:

GetCustomerContracts:

Alle verfügbaren KVs anzeigen, je Parameter im oder außerhalb des Pakets

Parameter: bool inverse (true für alle KVs nicht im Paket, false für alle KVs im Paket)

Rückgabe: ContractInfo[]

AddCustomer:

Einen Sub- Kunden anlegen

Parameter: string loginName (Benutzername des neuen Kunden),
 string loginPass (Kennwort des neuen Kunden),

Rückgabe: true bei Erfolg

AddContract:

Einen Vertrag für den aktuellen Kunden anlegen

Parameter: int contractId (KV Nummer),

Rückgabe: true bei Erfolg

GetCustomerContractList:

Die Liste der Verträge aller Subkunden, bzw. die Liste des gegebenen Subkunden abrufen

Parameter: int subCustomerId (Kundennummer Subkunde), null für alle Subkunden

Rückgabe: XmlDaten

Datenstruktur:

```
<customer customer="[Kundennummer Hauptkunde]">
  <subCustomer subCustomer="[Kundennummer Subkunde 1]"
loginId="[LoginName]">
    <contracts id="[KvNr]" orderDate="[Bestelldatum]" />
  </subCustomer>
  <subCustomer subCustomer="[Kundennummer Subkunde 2]"
    <contracts id="[KvNr]" orderDate="[Bestelldatum]" />
    <contracts id="[KvNr]" orderDate="[Bestelldatum]" />
  </subCustomer>
</customer>
```

CustomerChangePassword:

Kennwort für einen Kunden ändern

Parameter: string userName, string oldPassword, string newPassword

Rückgabe: true bei Erfolg