

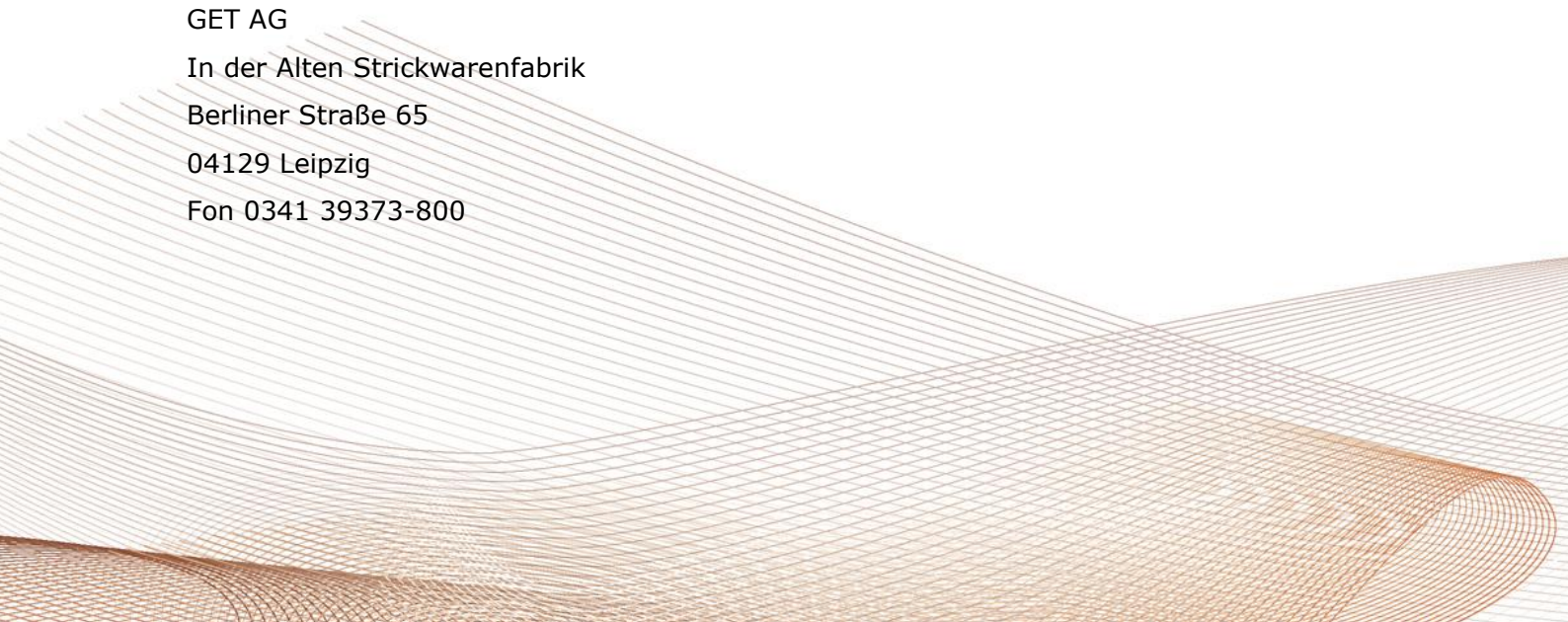
Gas Tarif Web Service

Schnittstellen-Beschreibung

Version 2.0

Leipzig, 22.03.2024

GET AG
In der Alten Strickwarenfabrik
Berliner Straße 65
04129 Leipzig
Fon 0341 39373-800



Inhaltsverzeichnis

Inhaltsverzeichnis.....	1
1 Beschreibung	5
2 Einbindung und API Beschreibung	5
3 Methoden-Übersicht	7
3.1 getBerechneteGrundversorgungstarife	7
3.2 getBerechneteTarife	7
3.3 getParameter	7
3.4 getTarifkarte.....	7
3.5 getUniversalinformation	8
3.6 getVersion	8
3.7 getVersorger	8
3.8 getVersorgerkarte	9
3.9 getVersorgerTarifvarianten	9
3.10 getAbgaben	9
4 Fehlermeldungen	9
5 Objekt-Definitionen	10
5.1 BasisAntwort	10
5.2 Meldung	10
5.3 getBerechneteGrundversorgungstarife Request	10
5.4 getBerechneteTarife Request	13
5.5 getBerechneteTarife Response	20
5.5.1 BerechneteTarifeErgebnisContainer	20
5.6 getUniversalinformation Request	28
5.7 getUniversalinformation Response	30
5.7.1 UniversalinformationErgebnis	30
5.7.2 Unterobjekte	30
5.8 getParameter Request	31
5.9 getParameter Response	31
5.9.1 Typ	31
5.9.2 Param.....	31

5.10	getTarifkarte Request	32
5.11	getTarifkarte Response	33
5.11.1	TarifkarteContainer	33
5.11.2	TarifkarteDaten.....	33
5.11.3	KundenArten	35
5.11.4	TarifkarteBAI	36
5.11.5	PreisStaffel.....	37
5.11.6	ZählerKosten	38
5.12	getVersion Response	39
5.13	getVersorger Request	39
5.14	getVersorger Response	40
5.14.1	Versorger	40
5.15	getVersorgerkarte Request	40
5.16	getVersorgerkarte Response	41
5.17	getVersorgerTarifvarianten Request.....	41
5.18	getVersorgerTarifvarianten Response.....	42
5.18.1	Versorger	43
5.19	getAbgaben Request.....	43
5.20	getAbgaben Response.....	43
5.20.1	Abgabe	44
5.21	Objekt-Definition IdObjGruppe.....	44
5.21.1	IntIdObj.....	45
5.21.2	DoubleIdObj	45
5.21.3	StringIdObj	45
5.21.4	DateIdObj	45
5.21.5	BasisIdObj	45
6	Weitere Informationen.....	46
6.1	Ausgeschlossene Tarife	46
6.2	Brutto-Netto-Werte	46
6.3	Stichtagsberechnung	46
6.4	Vergleichstarif/Ersparnis	46
6.4.1	Vergleichsrichtlinie	47
6.5	Universalinformationen	47

6.5.1	AGB, Vertragsunterlagen, Logos.....	48
6.6	Sortierung und Paging	48
6.7	Filtern nach Marken IDs	48
6.8	Tarif-Definitionen der GET AG	49
6.8.1	Ökogastarif	49
6.8.2	Pakettarif	50
6.8.3	Fixpreistarif	50
6.8.4	Preisgarantie	50
6.8.5	Energiepreistarife	50
6.9	Optionale Parameter/Objekte.....	51
6.10	Portal-Richtlinie.....	51
6.11	Fehlersuche	51
6.12	ortsscharfe bzw. bis zu hausnummernscharfe Tarifzuordnung	52
6.12.1	Nutzung der Ortsschärfe	52
6.12.2	Nutzung der bis zu hausnummernscharfen Tarifzuordnung	53
6.13	Verwendung der Bulkberechnung	54
6.14	Rundungen über berechnungsKonfiguration	55
6.15	Änderungshistorie	56
6.15.1	V1.3	56
6.15.2	V1.4	56
6.15.3	V1.5	56
6.15.4	V2.0	57

Dokumentenhistorie

Version	Datum	Änderung von	Bemerkung
2.0.0.1	23.06.2017	Sandy Quoos	Erstellung
2.0.0.2	26.02.2020	Patrick Schmorde	Auslagerung technische Dokumentation
2.0.0.2	20.06.2023	Daniel Sonne	Erweiterung einstellbare Rundungen

1 Beschreibung

Die GET AG stellt einen Webservice zu Gastarifen bereit. Die zur Verfügung stehenden Methoden sind unter Kapitel 3 aufgelistet.

Diese Dokumentation dient hauptsächlich dem inhaltlichen Verständnis der Methoden, Objekte, Parameter und Daten. Es handelt sich dabei nicht um eine Definition der Schnittstelle.

2 Einbindung und API Beschreibung

Bitte nutzen Sie folgende WSDL und WADL für die Anbindung des Webservice.

WSDL für Testservice (SSL)

<https://webservicetest1.ag-server.de/getagWSCore/jaxws/GasTarifService20x0?wsdl>

WSDL für produktiven Service (SSL)

<https://webservice1.ag-server.de/getagWSCore/jaxws/GasTarifService20x0?wsdl>

WADL für produktiven Service (SSL)

<https://webservice1.ag-server.de/getagWSCore/jaxrs/GasTarifService20x0/application.wadl>

WADL für Testservice (SSL)

<https://webservicetest1.ag-server.de/getagWSCore/jaxrs/GasTarifService20x0/application.wadl>

Weiterhin steht Ihnen eine API Dokumentation per Swagger zur Verfügung:

Produktiver Service

<https://webservice1.ag-server.de/getagWSCore/swagger-ui.html?urls.primaryName=Gas%20Tarif%20Webservice%202.0>

Test-Service

<https://webservicetest1.ag-server.de/getagWSCore/swagger-ui.html?urls.primaryName=Gas%20Tarif%20Webservice%202.0>

Weitere technische Merkmale und Anforderungen, wie z.B. zu Zertifikaten, Authentifizierung, Freischaltung oder Hinweise zum Anfrageverhalten finden Sie in der **Technischen Dokumentation** zu den Webservices.

3 Methoden-Übersicht

3.1 getBerechneteGrundversorgungstarife

Eingabe: <[getBerechneteGrundversorgungstarifeRequest](#)>

Ausgabe: <[getBerechneteTarife Response](#)>

Mit dieser Methode können mit nur wenigen Requestparametern Privat- oder Gewerbe-Grundversorgungstarife ermittelt werden, wobei die Anzahl der Tarife pro Versorger dabei auf 2 limitiert ist.

3.2 getBerechneteTarife

Eingabe: < [getBerechneteTarife Request](#) >

Ausgabe: <[getBerechneteTarife Response](#)>

Die Methode liefert berechnete Tarife mit einer Vielzahl an Filtermöglichkeiten, welche durch bestimmte Eingabe-Parameter getroffen werden können.

3.3 getParameter

Eingabe: <[Parameter Request](#)>

Ausgabe: <[Parameter Response](#)>

Die Methode liefert die Namensauflösung aller ParameterIDs, z.B. für Kündigungsfristen, Boni und Aufschläge, Einheiten, Kundenarten, Vertragslaufzeiten und Preisgarantiekomponenten.

3.4 getTarifkarte

Eingabe: <[getTarifkarte Request](#)>

Ausgabe: <[getTarifkarte Response](#)>

Die Methode liefert zu den übergebenen Parametern alle Informationen des Tarifes zurück.

3.5 getUniversalinformation

Eingabe: <[getUniversalinformation Request](#)>

Ausgabe: <[getUniversalinformation Response](#)>

Die Universalinformation besteht aus einem [Zusatzinformationsobjekt](#). Es ist für Informationen gedacht, die für eine Portaldarstellung benötigt werden, auch um ein etwaiges Backend abzulösen (z.B. AGB, Flyer, Logos, Farbcode). Es ist nicht dafür vorgesehen, Preisbestandteile oder Daten für die Tarifberechnung aufzunehmen und zu übertragen.

Einstieg ist die Nutzung der Methode [getBerechneteTarife](#), um an die nötigen Inputs für diese Methode zu gelangen.

3.6 getVersion

Eingabe: <>

Ausgabe: <[getVersion Response](#)>

Die Methode liefert Informationen über den Aktualitätsstand der Daten und Software, sowie die Version des Webservices.

3.7 getVersorger

Eingabe: <[getVersorger Request](#)>

Ausgabe: <[getVersorger Response](#)>

Die Methode liefert (bundesweit) alle Versorger mit gültigen Tarifen. Eine Filterung ist z.B. nach versorgerID und versorgerSitz möglich, wobei die Parameter in UND Verbindung zueinander stehen.

3.8 getVersorgerkarte

Eingabe: <[getVersorgerkarte Request](#)>

Ausgabe: <[getVersorgerkarte Response](#)>

Die Methode liefert Stammdateninformationen des Versorgers oder wahlweise der Marke zurück.

3.9 getVersorgerTarifvarianten

Eingabe: <[getVersorgerTarifvarianten Request](#)>

Ausgabe: <[getVersorgerTarifvarianten Response](#)>

Die Methode liefert ortsbezogen alle gültigen Tarifvarianten der entsprechenden Lieferanten.

3.10 getAbgaben

Eingabe: <[getAbgaben Request](#)>

Ausgabe: <[getAbgaben Response](#)>

Die Methode liefert alle derzeit gültigen Abgaben und Steuern, auch deren historische Werte.

4 Fehlermeldungen

Siehe Beschreibung der einzelnen [Fehlercodes](#).

5 Objekt-Definitionen

5.1 BasisAntwort

Parameter	Typ	Beschreibung
dynamischeAngaben	object[]	key/value Methode, siehe Objekt dynamischeAngaben
fehlermeldung	object	Siehe Objekt Meldung
warnung	object	Siehe Objekt Meldung
antwortStatus	int	fehlerCode (siehe Fehlermeldungen)
itag	string	Interne Kennzeichnung zur Hilfe bei der Fehlersuche (beachte Fehlersuche)

5.2 Meldung

Parameter	Typ	Beschreibung
dynamischeAngaben	object[]	key/value Methode, siehe Objekt dynamischeAngaben
code	int	fehlerCode (siehe Fehlermeldungen)
beschreibung	string	Interne Kennzeichnung zur Hilfe bei der Fehlersuche (beachte Fehlersuche)

5.3 getBerechneteGrundversorgungstarife Request

Parameter	Typ	Beschreibung
plz	string	Postleitzahl der Verbrauchsstelle. Bei Übergabe von „00000“ werden die bundesweiten Grundversorgungstarife berechnet.
verbrauch	int	Gesamtverbrauch in kWh
stichtag	date	Stichtag für Tarifgültigkeit (Format: 2017-01-01); keine

		historischen Abfragen möglich; siehe Informationen zum Stichtag
kundenart	int	0 = Privat; 1 = Gewerbe; 2 = Landwirtschaft; 4 = Sonderabnehmer
tariftyp	int	Liste von IDs; 1 = Normal; 2 = Heizgas; 3 = Kochen und Warmwasser Bsp.: >1,2<
leistung	double	Jahreshöchstleistung in kW
versorgerIDs	string	Liste von VersorgerIDs (beschränkt das Ergebnis auf Tarife der übergebenen Versorger); Bsp.: >123,456,<
netzbetreiber	string	Verbandsnummer des Netzbetreibers
netzgebiet	string	Interne Netznummer der GET AG
grundversorger	string	Verbandsnummer des Grundversorgers
grenzenBeiATignorierenBis	int	0 = kWh-Obergrenze des Versorgers berücksichtigen; sonst eigene Vorgabe
uid	string	Vom Nutzer wählbare ID zur Kennzeichnung der Anfrage und Zuordnung zum Ergebnis
dynamische Angaben	object[]	key/value Methode, siehe Objekt dynamischeAngaben

Der Request dieser Methode entspricht inhaltlich dem Request der Methode [getBerechneteTarife](#), wobei die nicht sichtbaren Parameter mit folgenden Werten vorbelegt sind:

Parameter	Wert
ort	
limit	2
tarifeVonGrundversorgern	1
grundversorgungstarife	1
handlingTarife	3
vergleichsTarifID	
vergleichsVariantenID	
vergleichsRichtlinie	1
oeko	3
fixpreis	3
vorkasse	3
paket	3
portalRichtlinie	3
bonus	3
preisgarantie	3
sachbonus	3
kaution	3
sonderabschlag	3
vertragslaufzeitBis	0
vertragsverlaengerung	0
kuendigungsfrist	0
bestandskundenTarif	0
preisgarantieDauer	0
zertifikate	
markenIDs	
tarifcode	
tarifRootIDsMindestfilter	

tarifRootIDs	
vertriebskanalIDs	
bAIextendedWhiteList	
bAIextendedBlackList	
kennzahl	0
sortierung	1
seitenGroesse	0
seitenNr	1

Der Response dieser Methode ist analog zu [getBerechneteTarife Response](#).

5.4 getBerechneteTarife Request

Parameter	Typ	Beschreibung
sortierung	int	0 = Sortierung nach Gesamtpreis exklusive Bonus 1 = Sortierung nach Gesamtpreis inklusive Bonus 2 = Sortierung nach Gesamtpunktzahl der Kennzahlen absteigend, Gesamtpreis inklusive Bonus 3 = Sortierung nach Kennzahlgruppe "Preis" absteigend Gesamtpreis, inklusive Bonus 4 = Sortierung nach Kennzahlgruppe "Qualität" absteigend Gesamtpreis, inklusive Bonus 5 = Sortierung nach Kennzahlgruppe "Kundenbewertung" absteigend, Gesamtpreis exklusive Bonus 6 = Sortierung nach Gesamtpunktzahl der Kennzahlen absteigend, Gesamtpreis exklusive Bonus

		7 = Sortierung nach Kennzahlgruppe "Preis" absteigend, Gesamtpreis exklusive Bonus 8 = Sortierung nach Kennzahlgruppe "Qualität" absteigend, Gesamtpreis exklusive Bonus 9 = Sortierung nach Kennzahlgruppe "Kundenbewertung" absteigend, Gesamtpreis exklusive Bonus
plz	string	Postleitzahl der Verbrauchsstelle
ort	string	Ortsname der Verbrauchsstelle
netzbetreiber	string	Verbandsnummer des Netzbetreibers
netzgebiet	string	Interne Netznummer der GET AG
grundversorger	string	Verbandsnummer des Grundversorgers
verbrauch	int	Gesamtverbrauch in kWh
leistung	double	Jahreshöchstleistung in kW
kundenart	int	0 = Privat; 1 = Gewerbe; 2 = Landwirtschaft; 4 = Sonderabnehmer
tariftyp	int	Liste von IDs (Bsp.: >1,2<) 1 = Normal; 2 = Heizgas; 3 = Kochen und Warmwasser
limit	int	Begrenzung der Anzahl von Sondertarifen je Versorger und je Marke (exkl. der Grundversorgungstarife); 0 = unbegrenzt negativer Wert = das Limit wird nur auf den Versorger, nicht auf etwaige Marken angewendet
grenzenBeiATIgnorierenBis	int	0 = kWh-Obergrenze des Versorgers

		berücksichtigen; sonst eigene Vorgabe
tarifeVonGrundversorgern	int	0 = nein; 1 = ja; 3 = egal (0 = keine Tarife von Grundversorgern, 1 = nur Tarife von Grundversorgern, 3 = Tarife von allen Versorgern)
grundversorgungstarife	int	0 = nein; 1 = ja; 3 = egal; (0 = keine GV-Tarife, 1 = nur GV-Tarife, 3 = GV- und Sonder-Tarife)
handlingTarife	int	0 = nein; 1 = ja; 3 = egal; (0 = keine abschließbaren Tarife, 1 = nur abschließbare Tarife, 3 = ohne Beschränkung)
vergleichsTarifID	int	tarifID zur Berechnung der Ersparnis; siehe Informationen zu Vergleichstarif/Ersparnis
vergleichsVariantenID	string	VariantenID des Vergleichstarifs
vergleichsRichtlinie	int	Gibt an wie der Vergleichstarif erzeugt werden soll, siehe Vergleichstarif Ersparnis
vergleichsVersorgerID	int	Sucht Vergleichstarif nur zu der übergebenen VersorgerID und berücksichtigt dabei die anderen Requestparameter
stichtag	date	Stichtag für Tarifgültigkeit (Format: 2017-01-27); keine historischen Abfragen möglich; siehe Informationen zum Stichtag
oeko	int	0 = nein weder Biogasanteil noch

		Klimaschutzprojekte 1 = ja entweder Biogasanteil oder Klimaschutzprojekte 3 = egal Biogasanteil, Klimaschutzprojekte oder keines von beiden siehe Information zu Ökotarif
fixpreis	int	0 = nein; 1 = ja; 3 = egal; siehe Information zu Fixpreistarif
vorkasse	int	0 = nein; 1 = ja; 3 = egal
paket	int	0 = nein; 1 = ja; 3 = egal; siehe Information zu Pakettarif
portalRichtlinie	int	1 = Portalrichtlinie „Allgemein“ 3 = Portalrichtlinien unberücksichtigt lassen (Infos dazu siehe Portal-Richtlinie)
bonus	int	0 = Filterung: Rückgabe des Gesamtpreises exkl. Erstjahresboni, darüber hinaus gehende Boni werden berücksichtigt 1 = Filterung: Rückgabe des Gesamtpreises inkl. Erstjahresboni, auch darüber hinaus gehende Boni werden berücksichtigt 3 = Filterung: alle Boni und Rabatte werden berücksichtigt 4 = Filterung: Rückgabe des Gesamtpreises exkl. Erstjahresboni, keine Anpassung der Mindestlaufzeit; Neukunden-, Wechsel-, Onlinebonus

		und Einzugsermächtigung sind stets Bedingung 5 = Filterung: Rückgabe des Gesamtpreises inkl. Erstjahresboni, keine Anpassung der Mindestlaufzeit; Neukunden-, Wechsel-, Onlinebonus und Einzugsermächtigung sind stets Bedingung 6 = Filterung: wie 5, aber Treueboni werden nur angezeigt, nicht eingerechnet 7 = Filterung: Tarife mit Bonus werden weggefiltert. Der Filter wird nicht auf den Vergleichstarif angewendet.
sachbonus	int	0 = nein; 1 = ja; 3 = egal
preisgarantie	int	0 = nein; 1 = ja; 3 = egal; siehe Information zu Preisgarantie
kaution	int	0 = nein; 1 = ja; 3 = egal
sonderabschlag	int	0 = nein; 1 = ja; 3 = egal
vertragslaufzeitBis	int	maximale Mindestlaufzeit in Monaten; 0 = ohne Beschränkung
vertragsverlaengerung	int	Vertragsverlängerung in Monaten; 0 = ohne Beschränkung
kuendigungsfrist	int	maximale Kündigungsfrist; 0 = egal; 1 = 2 Wochen; 2 = 1 Monat; 3 = 6 Wochen; 4 = 2 Monate; 5 = 3 Monate
bestandskundenTarif	int	0 = nein; 1 = ja; 3 = egal
preisgarantieDauer	int	minimale Dauer der Preisgarantie in Monaten; 0 = ohne Beschränkung

versorgerIDs	string	Liste von VersorgerIDs (beschränkt das Ergebnis auf Tarife der übergebenen Versorger); Bsp.: >123,456,<
zertifikate	string	Liste von ZertifikatIDs 2 = Zertifikat ohne Angabe 3 = TÜV - Nord 4 = TÜV - Süd 5 = TÜV-Rheinland 6 = TÜV-Saarland 7 = Öko-Trend 8 = Öko-Plus 9 = Grünes Gas Label 10 = KlimaINVEST Bsp.: >5,9<
seitenGroesse	int	Anzahl der Ergebnisse auf der Ergebnisseite, wobei der Vergleichstarif ggf. zusätzlich hinzu kommt; 0 = komplettes Ergebnis
seitenNr	int	auszugebende Seite in Abhängigkeit von der SeitenGröße; Eingabe > 0
markenIDs	string	Liste von markenIDs (beschränkt das Ergebnis auf Tarife der übergebenen Marken); Bsp.: >1,6,<
tarifcode	string	Liste von Tarifcodes (beschränkt das Ergebnis auf Tarife der übergebenen Tarifcodes); Bsp.: > 6434, 3436,< Wichtig: Da Tarifcodes nicht eindeutig über alle Energieversorger sind, ist

		immer die zusätzliche Übergabe der versorgerID erforderlich.
tarifRootIDsMindestfilter	string	Liste von Tarif RootIDs um Verbrauchertipp oder Monatstarif zusätzlich zum eigentlichen Ergebnis immer anzuzeigen. Bsp.: > 481730, 481680,<
tarifRootIDs	string	Liste von tarifRootIDs (beschränkt das Ergebnis auf Tarife der übergebenen tarifRootIDs); Bsp.: > 481730, 481680,<
vertriebskanalIDs	string	Liste von vertriebskanalIDs (beschränkt das Ergebnis auf Tarife der übergebenen vertriebskanalIDs); Bsp.: >0,1,<
uid	string	Vom Nutzer wählbare ID zur Kennzeichnung der Anfrage
regionalVersorgerUmkreis	int	Regionalversorgersuche im Umkreis von X Kilometer um die im Request übergebene PLZ. Es wird sowohl der Grundversorger als auch die Versorger mit Stammsitz in dem Umkreis gefiltert / ausgegeben. <diese Funktion ist noch in Entwicklung>
BAIextendedWhiteList	string	Liste von BAI bedingungIDs zur Einschränkung auf z.B. nur Onlinetarife (beschränkt das Ergebnis auf Tarifvarianten der übergebenen BAI ID); Bsp.: > 16,132,...<

		Die Liste muss ohne Leerzeichen übergeben werden
BAIextendedBlackList	string	Liste von BAI bedingungIDs um Tarifvarianten vom Ergebnis auszuschließen Bsp.: > 16,132,...< Die Liste muss ohne Leerzeichen übergeben werden
details	int	0 = keine Details (Default) 1 = detaillierte Aufstellung der Netzentgeltberechnung bei Energiepreistarifen
kennzahl	int	für die individuelle Erstellung von Indizes über die Ergebnisse Kennzahl ID=1 Simplex
dynamische Angaben	object[]	key/value Methode, siehe Objekt dynamische Angaben

5.5 getBerechneteTarife Response

Parameter	Typ	Beschreibung
BasisAntwort	object	Siehe Objekt BasisAntwort
ergebnisContainer	object	siehe Objektbeschreibung BerechneteTarifeErgebnisContainer
ergebnisseGesamtAnzahl	int	Gesamtanzahl der Tarife zu den Parametern

5.5.1 BerechneteTarifeErgebnisContainer

Parameter	Typ	Beschreibung
versorgerInformationen	object	Siehe Objekt VersorgerInformation
plz	string	Postleitzahl der Verbrauchsstelle

ort	string	Ortsname der Verbrauchsstelle
ergebnis	object	siehe Objektbeschreibung Ergebnis
ergebnisseAnzahl	int	Gesamtanzahl der berechneten Tarife im Ergebnis-Objekt
ergebnisseSeiteAnzahl	int	Anzahl der Tarife zu den Parametern, abhängig von Seitengröße
dynamischeAngaben	object[]	key/value Methode, siehe Objekt dynamischeAngaben
uid	string	Vom Nutzer wählbare ID zur Kennzeichnung der Anfrage und Zuordnung zum Ergebnis

5.5.1.1 VersorgerInformationen

Lokationsbezogene Versorgerinformation (Netzbetreiber, Grundversorger)

Parameter	Typ	Beschreibung
marktrollen	object	Siehe Objekt Marktrolle
gebiet	string	Internes Gebiet
versorgerID	int	Interne ID des Versorgers
versorgerName	string	Name des Versorgers
dynamischeAngaben	object[]	key/value Methode, siehe Objekt dynamischeAngaben

5.5.1.2 BerechneteTarifeErgebnis

Parameter	Typ	Beschreibung
aufschlaege (0:n Beziehung)	object	Aufschläge des Tarifs als Objekt BerechneteTarifeBAI
boni (0:n Beziehung)	object	Boni des Tarifs als Objekt BerechneteTarifeBAI
informationen (0:n Beziehung)	object	Informationen zum Tarif als Objekt BerechneteTarifeBAI

kennzahlErgebnis	object	siehe Objektbeschreibung kennzahlErgebnis
netzEntgeltBerechnung	object	siehe Objektbeschreibung netzEntgeltBerechnung
margeBerechnung	object	siehe Objektbeschreibung MargeBerechnung
anzeigeBis	date	Vertriebsende des Tarifs
anzeigeVon	date	Vertriebsbeginn des Tarifs
arbeitspreisNetto	decimal	Verbrauchsarbeitspreis in Euro/Jahr netto
arbeitspreisBrutto	decimal	Verbrauchsarbeitspreis in Euro/Jahr brutto
arbeitspreisCtKWhNetto	decimal	Arbeitspreis in ct/kWh netto
arbeitspreisCtKWhBrutto	decimal	Arbeitspreis in ct/kWh brutto
ersparnisNetto	decimal	Ersparnis zum Vergleichstarif in Euro/Jahr netto
ersparnisBrutto	decimal	Ersparnis zum Vergleichstarif in Euro/Jahr brutto
gesamtpreisNetto	decimal	Gesamtpreis in Euro/Jahr netto
gesamtpreisBrutto	decimal	Gesamtpreis in Euro/Jahr brutto
gesamtpreisOhneBoniNetto	decimal	Gesamtpreis ohne Boni in Euro/Jahr netto
gesamtpreisOhneBoniBrutto	decimal	Gesamtpreis ohne Boni in Euro/Jahr brutto
grundpreisNetto	decimal	Grundpreis in Euro/Jahr netto
grundpreisBrutto	decimal	Grundpreis in Euro/Jahr brutto
gueltigBis	date	Gültigkeitsbeginn des Tarifs
gueltigVon	date	Gültigkeitsende des Tarifs
isGrundversorgungsTarif	boolean	ist Grundversorgungstarif
isOekogasTarif	boolean	ist Ökogastarif; siehe Information zu Ökotarif

isPaketTarif	boolean	ist Pakettarif; siehe Information zu Pakettarif
isVergleichsTarif	boolean	ist der Vergleichstarif; siehe Informationen zu Vergleichstarif/Ersparnis
isVorkasseTarif	boolean	ist Vorkasseprodukt
kuendigungsfrist	string	Kündigungsfrist als Freitext
kuendigungsfristID	int	ID des Kündigungsfreitextes
leistungspreisNetto	decimal	Leistungspreis in Euro/Jahr netto
leistungspreisEuroKWNetto	decimal	Leistungspreis in Euro/kW netto
leistungspreisBrutto	decimal	Leistungspreis in Euro/Jahr brutto
leistungspreisEuroKWBrutto	decimal	Leistungspreis in Euro/kW brutto
markenID	int	ID der Marke zur Zuordnung; 0 = keine Markenzuordnung
paketMehrverbrauchsPreisNetto	decimal	Preis für Mehrverbrauch bei Pakettarif in ct/kWh netto
paketMehrverbrauchsPreisBrutto	decimal	Preis für Mehrverbrauch bei Pakettarif in ct/kWh brutto
paketMenge	int	Paketmenge in kWh
paketPreisNetto	decimal	Paketpreis in Euro/Jahr netto
paketPreisBrutto	decimal	Paketpreis in Euro/Jahr brutto
tarifID	int	ID des Tarifs
tarifInfo	string	Tarifinformation
tarifName	string	Name des Tarifs
tarifRootID	int	RootID des Tarifs
variantenID	string	VariantenID
versorgerID	int	ID des Versorgers
versorgerName	string	Name des Versorgers oder vorrangig der Marke
vertragslaufzeit	string	Mindestlaufzeit als Freitext
vertragslaufzeitID	int	ID des Mindestlaufzeitfreitextes

vertragsverlaengerung	string	Vertragsverlängerung als Freitext
vertragsverlaengerungID	int	ID des Vertragsverlängerungsfreitextes
tarifCode	string	Tarifcode des Tarifs
arbeitspreisEnergiepreisNetto	decimal	Energiepreis Verbrauchsarbeitspreis in Euro/Jahr netto
arbeitspreisEnergiepreisBrutto	decimal	Energiepreis Verbrauchsarbeitspreis in Euro/Jahr brutto
grundpreisEnergiepreisNetto	decimal	Energiepreis Grundpreis in Euro/Jahr netto
grundpreisEnergiepreisBrutto	decimal	Energiepreis Grundpreis in Euro/Jahr brutto
arbeitspreisEnergiepreisCtKWhNetto	decimal	Energiepreis Arbeitspreis oder in ct/kWh netto
arbeitspreisEnergiepreisCtKWhBrutto	decimal	Energiepreis Arbeitspreis oder in ct/kWh brutto
uid	string	Vom Nutzer wählbare ID zur Kennzeichnung der Anfrage und Zuordnung zum Ergebnis
dynamische Angaben	object[]	key/value Methode, siehe Objekt dynamische Angaben

5.5.1.2.1 BerechneteTarifeBAI

Entspricht dem Objekt [TarifkarteBAI](#), ergänzt um:

Parameter	Typ	Beschreibung
calcWertNetto	decimal	Höhe des berechneten BAI netto
calcWertBrutto	decimal	Höhe des berechneten BAI brutto

5.5.1.2.2 kennzahlErgebnis

Die Definition der IDs dieser Methode finden Sie in der Methode `getParameter` unter `Param_Tarif_Gas_Simplex`

Parameter	Typ	Beschreibung
id	int	ID der Kennzahl
wert	double	Wert der Kennzahl
kennzahlErgebnisBestandteil	object []	siehe Objektbeschreibung KennzahlErgebnis
dynamischeAngaben	object[]	key/value Methode, siehe Objekt dynamischeAngaben

5.5.1.2.3 NetzEntgeltBerechnung

Parameter	Typ	Beschreibung
abgaben	object	Siehe Objekt Abgaben
entgeltInformationen	object	Siehe Objekt EntgeltInformationen
steuern	object	Siehe Objekt Steuern
arbeitspreis	decimal	Arbeitspreis in ct/kWh netto
entgeltFuerAbrechnung	decimal	Abrechnungskosten inkl. zusätzliche Abrechnungen in €/Jahr netto
entgeltFuerEinbauBetriebWartungMesstechnik	decimal	Messstellenbetriebskosten in €/Jahr netto
entgeltFuerMessungUndAbleseung	decimal	Summe aus Mess- und Ablesungskosten in €/Jahr netto
entgeltZaehlerpreisAbleseung	decimal	Summe aus entgeltFuerEinbauBetriebWartungMesstechnik und entgeltFuerMessungUndAbleseung in €/Jahr netto, da es VNB gibt, die keine Aufsplittung der Messpreise veröffentlichen
gesamtpreis	decimal	Gesamtpreis Entgelt in €/Jahr netto
grundpreis	decimal	Grundpreis in €/Jahr netto

konzessionsabgabe	decimal	Konzessionsabgabe in ct/kWh
leistungspreis	decimal	Leistungspreis in €/Jahr netto
druckstufe	int	Druckstufe, siehe getParameter
zaehlerTyp	int	zu den Standardeingangsparametern der Netzentgeltberechnung passender Zählertyp, siehe getParameter
zaehlerGroesseBis	double	zu den Standardeingangsparametern der Netzentgeltberechnung passende Zählergröße bis des Preisblatts.
zaehlerGroesseVon	double	zu den Standardeingangsparametern der Netzentgeltberechnung passende Zählergröße von des Preisblatts.
dynamischeAngaben	object[]	key/value Methode, siehe Objekt dynamischeAngaben

5.5.1.2.3.1 Entgeltinformation

Parameter	Typ	Beschreibung
entgeltID	int	GET AG interne (eindeutige) ID des teuersten Entgelts für die angegebenen Parameter bei mehreren möglichen Ergebnissen
entgeltRootID	int	GET AG interne Historien-ID des Entgelts
entgeltName	string	Name des Entgelts
gueltigBis	date	Gültigkeit des Entgelts (Enddatum), wenn kein Enddatum bekannt ist, wird der Wert auf „31.12.9999“ gesetzt.

gueltigVon	date	Gültigkeit des Entgelts (Startdatum)
dynamischeAngaben	object[]	key/value Methode, siehe Objekt dynamischeAngaben

5.5.1.2.3.2 Marktrolle

Parameter	Typ	Beschreibung
marktrolleID	int	ID der Marktrolle, siehe getParameter
verbandsnummer	string	13-stellige BDEW/VDEW-Nummer oder ILN
verbandsnummerKurz	string	kurze Nummer des BDEW/VDEW (Netzbetreibernummer)
dynamischeAngaben	object[]	key/value Methode, siehe Objekt dynamischeAngaben

5.5.1.2.3.3 Abgaben

Parameter	Typ	Beschreibung
id	int	ID-Parameter der Abgaben, siehe getParameter
name	string	Name der Abgaben (EEG, KWK, Sonderkundenumlage, Offshore-Haftungsumlage, Regelenergieumlage)
wert	decimal	Abgabenhöhe
dynamischeAngaben	object[]	key/value Methode, siehe Objekt dynamischeAngaben

5.5.1.2.3.4 Steuern

Parameter	Typ	Beschreibung
id	int	ID-Parameter der Steuern, siehe getParameter
name	string	Name der Steuer (Einkommensteuer, Körperschaftsteuer, Grunderwerbsteuer, etc.)

		Stromsteuer, Umsatzsteuer)
wert	decimal	Steuersatz
dynamischeAngaben	object[]	key/value Methode, siehe Objekt dynamischeAngaben

5.5.1.2.4 MargeBerechnung

Parameter	Typ	Beschreibung
id	int	ID 1 = Beschaffung ID 2 = Marge
wertNetto	decimal	Wert in Euro netto
wertBrutto	decimal	Wert in Euro brutto
wertCtKWhNetto	decimal	Wert in Cent je kWh netto
wertCtKWhBrutto	decimal	Wert in Cent je kWh netto
dynamischeAngaben	object[]	key/value Methode, siehe Objekt dynamischeAngaben

5.5.1.3 Zertifikate

Parameter	Typ	Beschreibung
zertifikat	string	Name des Zertifikats
zertifikatID	int	ID des Zertifikats
dynamische Angaben	object[]	key/value Methode, siehe Objekt dynamische Angaben

5.6 getUniversalinformation Request

Parameter	Typ	Beschreibung
verbrauch	int	Gesamtverbrauch in kWh
stichtag	date	Stichtag für Tarifgültigkeit (Format: 2010-01-27); keine historischen Abfragen möglich;

		siehe Informationen zum Stichtag
tarifcode	string	Liste von Tarifcodes (beschränkt das Ergebnis auf Tarife der übergebenen Tarifcodes); Bsp.: > 6434, 3436,< Wichtig: Da Tarifcodes nicht eindeutig über alle Energieversorger sind, ist immer die zusätzliche Übergabe der versorgerID erforderlich.
tarifRootIDs	string	Liste von tarifRootIDs (beschränkt das Ergebnis auf Tarife der übergebenen tarifRootIDs); Bsp.: > 481730, 481680,<
versorgerIDs	string	Liste von VersorgerIDs (beschränkt das Ergebnis auf Tarife der übergebenen Versorger); Bsp.: >123,456,<
vertriebskanalIDs	string	Liste von vertriebskanalIDs (beschränkt das Ergebnis auf Tarife der übergebenen vertriebskanalIDs); Bsp.: >1,2,< 0=vertriebskanalweit gültig
zusatzinformationIDs	string	Liste von IDs der Zusatzinformationen (beschränkt das Ergebnis auf Tarife der übergebenen IDs); siehe Param_Tarif_Zusatzinformation in Methode getParameter z.B. 1 = AGB des Tarifs
dynamischeAngaben	object[]	key/value Methode, siehe Objekt dynamischeAngaben

5.7 getUniversalinformation Response

Parameter	Typ	Beschreibung
BasisAntwort	object	Siehe Objekt BasisAntwort
ergebnisse	object	siehe Objektbeschreibung getUniversalinfomation Ergebnis
ergebnisseCount	int	Gesamtanzahl der Tarife zu den Parametern

5.7.1 UniversalinformationErgebnis

Parameter	Typ	Beschreibung
tarifID	int	TarifID des Tarifs
tarifRootID	int	TarifRootID des Tarifs
versorgerID	int	ID des Versorgers
markenID	int	ID der Marke
zusatzInformation	object[]	Zusatzinformationen als Objekt Zusatzinformation
dynamischeAngaben	object[]	key/value Methode, siehe Objekt dynamischeAngaben

5.7.2 Unterobjekte

5.7.2.1 zusatzInformation

Parameter	Typ	Beschreibung
zusatzinformationID	int	ID der Zusatzinformation, siehe Param_Tarif_Zusatzinformation in Methode getParameter z.B. 1 = AGB des Tarifs
vertriebskanalID	int	ID des Vertriebskanals
tarifcode	string	Tarifcode des Tarifs
zusatzInfo	string	Zusatzinfo, siehe getParameter

		z.B. 6= Link zu AGB des Tarifs
gueltigVon	date	Zusatzinformation gültig ab
gueltigBis	date	Zusatzinformation gültig bis
verbrauchVon	int	Zusatzinformation gültig ab Verbrauch in kWh
verbrauchBis	int	Zusatzinformation gültig bis Verbrauch in kWh
dynamischeAngaben	object[]	key/value Methode, siehe Objekt dynamischeAngaben

5.8 getParameter Request

Aufruf ohne weiteren Parameter!

5.9 getParameter Response

Parameter	Typ	Beschreibung
BasisAntwort	object	Siehe Objekt BasisAntwort
typ	object	Siehe Objekt Typ

5.9.1 Typ

Parameter	Typ	Beschreibung
dynamischeAngaben	object[]	key/value Methode, siehe Objekt dynamischeAngaben
param	object	siehe Objektbeschreibung Param
name	string	Name des Typs

5.9.2 Param

Parameter	Typ	Beschreibung
dynamischeAngaben	object[]	key/value Methode, siehe Objekt dynamischeAngaben

id	int	ID des Parameters
name	string	Name des Parameters
beschreibung	string	Beschreibung des Parameter

5.10 getTarifkarte Request

Parameter	Typ	Beschreibung
rootID (erforderlich)	int	RootID des Tarifs
markenID	int	zum Zuordnen einer bestimmten Marke, siehe Filter nach MarkenIDs
stichtag	date	Stichtag für Tarifgültigkeit (Format: 2017-01-27); keine historischen Abfragen möglich; siehe Informationen zum Stichtag
plz (erforderlich)	string	Postleitzahl der Verbrauchsstelle
ort	string	Ortsname der Verbrauchsstelle
netzbetreiber	string	Verbandsnummer des Netzbetreibers
netzgebiet	string	Interne Netznummer der GET AG
grundversorger	string	Verbandsnummer des Grundversorgers
variantenID	string	ID der Tarifvariante in Abhängigkeit der BAIs
ausgabeKonfiguration	int	0 - auf das teuerste Ergebnis beschränkt: Gesamtpreis ohne Bonus absteigend (siehe 1.5er) 1 - alle Ergebnisse zulassen
dynamische Angaben	object[]	key/value Methode, siehe Objekt dynamische Angaben

5.11 getTarifkarte Response

Parameter	Typ	Beschreibung
BasisAntwort	object	Siehe Objekt BasisAntwort
tarifkarten	object	Siehe Objekt TarifkarteContainer

5.11.1 TarifkarteContainer

Parameter	Typ	Beschreibung
versorgerInformationen	object	Siehe Objekt VersorgerInformation
tarifkarte	object	Siehe Objekt TarifkarteDaten
plz	string	Postleitzahl der Verbrauchsstelle
ort	string	Ortsname der Verbrauchsstelle

5.11.2 TarifkarteDaten

Parameter	Typ	Beschreibung
aufschlaege (0:n Beziehung)	object	Aufschläge des Tarifs als Objekt TarifkarteBAI
boni (0:n Beziehung)	object	Boni des Tarifs als Objekt TarifkarteBAI
informationen (0:n Beziehung)	object	Informationen zum Tarif als Objekt TarifkarteBAI
zaehlerKosten (0:n Beziehung)	object	siehe Objektbeschreibung Zählerkosten
kundenArten (0:n Beziehung)	object	siehe Objektbeschreibung KundenArten
preisStaffel	object	siehe Objektbeschreibung PreisStaffel
dsAbzugspauschaleBrutto	decimal	Durchschnittspreis-Abzugspauschale Maximum in Euro/Jahr Brutto
dsAbzugspauschaleNetto	decimal	Durchschnittspreis-Abzugspauschale Maximum in Euro/Jahr Netto
dsPreisHoechstBrutto	decimal	Durchschnittshöchstpreis am Tarif in

		ct/kWh brutto
dsPreisHoechstNetto	decimal	Durchschnittshöchstpreis am Tarif in ct/kWh netto
dsPreisMindestBrutto	decimal	Durchschnittsmindestpreis am Tarif in ct/kWh brutto
dsPreisMindestNetto	decimal	Durchschnittsmindestpreis am Tarif in ct/kWh Netto
fristVertragsabschluss	date	Vertragsabschlussfrist
fruehesteBelieferung	date	frühester Lieferbeginn
gueltigBis	date	Gültigkeitsende des Tarifs
gueltigVon	date	Gültigkeitsbeginn des Tarifs
isBestandskundenTarif	boolean	ist Bestandskundentarif
isErsatzversorgungsTarif	boolean	ist Ersatzversorgungstarif
isFixpreisTarif	boolean	ist Fixpreistarif
isGrundversorgungsTarif	boolean	ist Grundversorgungstarif
isHeizgasTarif	boolean	ist Heizgastarif
isKochenUndWarmwasser Tarif	boolean	ist Tarif für Kochen und Warmwasser
isPaketTarif	boolean	ist Pakettarif
isVorkasseTarif	boolean	ist Vorkassentarif
kontingentFixpreisangebot	int	Kontingent Fixpreisangebot in kWh
kuendigungsfrist	string	Kündigungsfrist als Freitext
kuendigungsfristID	int	ID des Kündigungsfreitextes
mindestleistungspreis Brutto	decimal	Mindestleistungspreis am Tarif in Euro/Jahr brutto
mindestleistungspreis Netto	decimal	Mindestleistungspreis am Tarif in Euro/Jahr netto
mindestrechnungspreisBrutto	decimal	Mindestrechnungspreis in Euro/Jahr Brutto
mindestrechnungspreisNetto	decimal	Mindestrechnungspreis in Euro/Jahr Netto

statusID	int	siehe Fehlermeldungen
tarifID	int	ID des Tarifs
tarifInfo	string	Tarifinformationen am Gastarif
tarifName	string	Name des Tarifs
tarifRootID	int	RootID des Tarifs
versorgerID	int	ID des Versorgers
versorgerName	string	Name des Versorgers oder vorrangig der Marke
vertragslaufzeit	string	Mindestlaufzeit als Freitext
vertragslaufzeitBegrenzung	int	Vertragslaufzeitbegrenzung in Monaten; 0 = unbegrenzt
vertragslaufzeitID	int	ID des Mindestlaufzeitfreitextes
vertragsverlaengerung	string	Vertragsverlängerung als Freitext
vertragsverlaengerungID	int	ID des Vertragsverlängerungsfreitextes
variantenID	string	ID der Tarifvariante in Abhängigkeit der BAIs
tarifcode	string	Tarifcode des Tarifs
dynamische Angaben	object[]	key/value Methode, siehe Objekt dynamische Angaben

5.11.3 KundenArten

Parameter	Typ	Beschreibung
art	string	Name der Kundenart
artID	int	ID der Kundenart
dynamische Angaben	object[]	key/value Methode, siehe Objekt dynamische Angaben

5.11.4 TarfkarteBAI

BAI-Objekt der Tarfkarte

Parameter	Typ	Beschreibung
bedingung	string	BAI-Bezeichnung
bedingungID	int	ID der BAI-Bezeichnung
beschreibungstext	string	Beschreibungstext
biogasAnteil	decimal	Anteil an Biogas
einheit	string	Einheit als Freitext
einheitID	int	ID der Einheit
gueltigBis	date	Gültigkeitsende
gueltigVon	date	Gültigkeitsbeginn
id	int	ID des BAIs
preisgarantieBestandteile	string	Preisgarantiebestandteile als Freitext
preisgarantieBestandteileID	int	ID der Preisgarantiebestandteile
preisgarantieBis	date	Gültigkeitsende der Preisgarantie
preisgarantieFuerMonate	int	Anzahl der Monate für die Preisgarantie
verbrauchBis	int	Verbrauchsobergrenze
verbrauchVon	int	Verbrauchsuntergrenze
vertragslaufzeit	string	Vertragslaufzeit als Freitext
vertragslaufzeitID	int	ID der Vertragslaufzeit
wertBrutto	decimal	Höhe des BAI Brutto
wertNetto	decimal	Höhe des BAI Netto
zertifikate (0:n Beziehung)	object	siehe Objektbeschreibung Zertifikate
art	string	Art des BAI
artID	int	ID der Art des BAI
maxWertNetto	decimal	maximale Höhe des BAI netto
maxWertBrutto	decimal	maximale Höhe des BAI brutto
maxWertEinheitID	decimal	Einheit der maximalen Höhe des BAI
dynamische Angaben	object[]	key/value Methode, siehe Objekt dynamische Angaben

5.11.5 PreisStaffel

Parameter	Typ	Beschreibung
arbeitspreisBrutto	decimal	Arbeitspreis in ct/kWh Brutto
arbeitspreisNetto	decimal	Arbeitspreis in ct/kWh Netto
benutzungsdauerBis	double	Benutzungsdauerobergrenze in h
benutzungsdauerVon	double	Benutzungsdaueruntergrenze in h
bezeichnung	string	Staffelbezeichnung
grundpreisBrutto	decimal	Grundpreis in Euro/Jahr Brutto
grundpreisNetto	decimal	Grundpreis in Euro/Jahr Netto
id	int	ID der Preisstaffel
isLeistungspreisGueltig AbNull	boolean	ist Leistungspreis ab 0 kW gültig
isZonenTarif	boolean	Preisstaffel ist Zonung
leistungBis	int	Leistungsobergrenze in kW
leistungVon	int	Leistungsuntergrenze in kW
leistungspreisBrutto	decimal	Leistungspreis in Euro/Jahr Brutto
leistungspreisNetto	decimal	Leistungspreis in Euro/Jahr Netto
leistungspreisSchrittweite	double	Schrittweite der LP-Gültigkeit in kW
verbrauchBis	int	Verbrauchsobergrenze in kWh
verbrauchVon	int	Verbrauchsuntergrenze in kWh
verbrauchGuenstigBis	int	günstige Verbrauchsobergrenze in kWh
verbrauchGuenstigVon	int	günstige Verbrauchsuntergrenze in kWh
arbeitspreisNettoEnergiepreis	decimal	Energiepreis Arbeitspreis in ct/kWh Netto
arbeitspreisBruttoEnergiepreis	decimal	Energiepreis Arbeitspreis in ct/kWh Brutto
grundpreisNettoEnergiepreis	decimal	Energiepreis Grundpreis in Euro/Jahr Netto
grundpreisBruttoEnergiepreis	decimal	Energiepreis Grundpreis in Euro/Jahr Netto

tarifcode	string	Tarifcode des Tarifs
dynamische Angaben	object[]	key/value Methode, siehe Objekt dynamische Angaben

5.11.6 ZählerKosten

Parameter	Typ	Beschreibung
abrechnungJaehrlichBrutto	decimal	Kosten für jährliche Abrechnung in Euro/Jahr Brutto
abrechnungJaehrlichNetto	decimal	Kosten für jährliche Abrechnung in Euro/Jahr Netto
abrechnungMonatlichBrutto	decimal	Kosten für monatliche Abrechnung in Euro/Jahr Brutto
abrechnungMonatlichNetto	decimal	Kosten für monatliche Abrechnung in Euro/Jahr Netto
grundpreisBrutto	decimal	Grundpreis für den Zähler
grundpreisNetto	decimal	Grundpreis für den Zähler
gueltigBis	date	Gültigkeitsende
gueltigVon	date	Gültigkeitsbeginn
id	int	ID der Zählerkostenstaffel
isAbrechnungImMesspreis Jaehrlich	boolean	ist jährliche Abrechnung in Messkosten enthalten
isAbrechnungImMesspreis Monatlich	boolean	ist monatliche Abrechnung in Messkosten enthalten
isImGPEnthalten	boolean	ist im Grundpreis (Verbrauchsstaffel) enthalten
messungUndZaehlungBrutto	decimal	Kosten für Messung und Zählung in Euro/Jahr brutto
messungUndZaehlungNetto	decimal	Kosten für Messung und Zählung in Euro/Jahr netto
zaehlerGroesseBis	double	Zählergröße Obergrenze

zaehlerGroesseVon	double	Zählergröße Untergrenze
zaehlerTyp	string	Typ des Zählers
zaehlerTypID	int	ID des Zählertyps
zusatzAbrechnungBrutto	decimal	Kosten für zusätzliche Abrechnung in Euro brutto
zusatzAbrechnungNetto	decimal	Kosten für zusätzliche Abrechnung in Euro netto
dynamische Angaben	object[]	key/value Methode, siehe Objekt dynamische Angaben

5.12 getVersion Response

Parameter	Typ	Beschreibung
BasisAntwort	object	Siehe Objekt BasisAntwort
softwarestand	decimal (19,0)	Millisekunden zwischen 01.01.1970 und der letzten Software-Aktualisierung (z. B. 1341489661000)
datenbankstand	decimal (19,0)	Millisekunden zwischen 01.01.1970 und der letzten Daten-Aktualisierung (z. B. 1341489661000)
version	string	Version des Webservices

5.13 getVersorger Request

Parameter	Typ	Beschreibung
stichtag	date	Stichtag (Format: 2017-01-27); keine historischen Abfragen möglich; siehe Informationen zum Stichtag
versorgerIDs	string	Liste von VersorgerIDs (beschränkt das Ergebnis auf Tarife der übergebenen Versorger); Bsp.: >123,456,<

verbandsnummer	string	Liste von 13-stellige VDEWID oder ILN des Lieferanten
versorgerName	string	Name / Firmierung des gesuchten Lieferanten / Teilstringsuche ist möglich
versorgerSitz	string	Sitz des Lieferanten
dynamische Angaben	object[]	key/value Methode, siehe Objekt dynamische Angaben

5.14 getVersorger Response

Parameter	Typ	Beschreibung
BasisAntwort	object	Siehe Objekt BasisAntwort
versorger	object	siehe Objekt Versorger

5.14.1 Versorger

Parameter	Typ	Beschreibung
markenID	int	ID der Marke
markenName	string	Name der Marke
versorgerID	Int	ID des Versorgers
versorgerName	string	Name des Versorgers
dynamische Angaben	object[]	key/value Methode, siehe Objekt dynamische Angaben

5.15 getVersorgerkarte Request

Parameter	Typ	Beschreibung
versorgerID	int	ID des Versorgers
markenID	int	ID der Marke
dynamische Angaben	object[]	key/value Methode, siehe Objekt dynamische Angaben

5.16 getVersorgerkarte Response

Alle Angaben werden vorrangig zur Marke und ansonsten zum Versorger ausgegeben.

Parameter	Typ	Beschreibung
BasisAntwort	object	Siehe Objekt BasisAntwort
bemerkung	string	Bemerkung zu Marke, wenn MarkenID<>0
email	string	E-Mail Adresse
fax	string	Faxnummer
firmierung	string	Name
hausnummer	string	Hausnummer
kurzform	string	Kurzform
ort	string	Ort
plz	string	Postleitzahl
postfach	string	Postfach
postfachOrt	string	Postfachort
postfachPLZ	string	Postfachplz
staat	string	Staat
statusID	int	siehe Fehlermeldungen
strasse	string	Straße
telefon	string	Telefon
verbandsnummer	string	13-stellige DVGWID oder ILN
webadresse	string	Homepage
dynamische Angaben	object[]	key/value Methode, siehe Objekt dynamische Angaben

5.17 getVersorgerTarifvarianten Request

Parameter	Typ	Beschreibung
plz	string	Postleitzahl der Verbrauchsstelle
kundenart	int	0 = Privat; 1 = Gewerbe; 2 =

		Landwirtschaft; 4 = Sonderabnehmer
stichtag	date	Stichtag (Format: 2017-01-01); keine historischen Abfragen möglich; siehe Informationen zum Stichtag
tariftyp	int	Liste von IDs; 1 = Normal; 2 = Heizgas; 3 = Kochen und Warmwasser Bsp.: >1,2<
verbrauch	int	Gesamtverbrauch in kWh
versorgerID	int	ID des Versorgers (wird zum Pflichtfeld, wenn im Request keine PLZ übergeben wird)
leistung	double	Jahreshöchstleistung in kW
ort	string	Ortsname der Verbrauchsstelle (kann genutzt werden wenn auch PLZ übergeben wurde)
netzbetreiber	string	Verbandsnummer des Netzbetreibers (kann genutzt werden wenn auch PLZ übergeben wurde)
grundversorger	string	Verbandsnummer des Grundversorgers (kann genutzt werden wenn auch PLZ übergeben wurde)
dynamische Angaben	object[]	key/value Methode, siehe Objekt dynamische Angaben

5.18 getVersorgerTarifvarianten Response

Parameter	Typ	Beschreibung
BasisAntwort	object	Siehe Objekt BasisAntwort
versorger	object	siehe Objekt Versorger

5.18.1 Versorger

Parameter	Typ	Beschreibung
tarife	object	siehe Objekt Tarife
markenID	int	ID der Marke
markenName	string	Name der Marke
versorgerID	Int	ID des Versorgers
versorgerName	string	Name des Versorgers
dynamische Angaben	object[]	key/value Methode, siehe Objekt dynamische Angaben

5.18.1.1 Tarife

Parameter	Typ	Beschreibung
isGrundversorgungsTarif	boolean	ist Grundversorgungstarif
tarifID	int	ID des Tarifs
tarifName	string	Name des Tarifs
variantenID	string	ID der Varianten des Tarifs
dynamische Angaben	object[]	key/value Methode, siehe Objekt dynamische Angaben

5.19 getAbgaben Request

Aufruf ohne weitere Parameter!

5.20 getAbgaben Response

Parameter	Typ	Beschreibung
BasisAntwort	object	Siehe Objekt BasisAntwort
abgabe	object	Siehe Objekt Abgabe
steuer	object	Siehe Objekt Abgabe

5.20.1 Abgabe

Parameter	Typ	Beschreibung
id	int	ID des Abgabentyps aus getParameter
gueltigVon	date	Gültigkeitsbeginn
gueltigBis	date	Gültigkeitsende
lvgStaffel	int	ID der LVG: 0 = keine LVG 1 = LVG Gruppe A 2 = LVG Gruppe B 3 = LVG Gruppe C
wert	decimal	Preis
preisEinheit	string	Cent (aktuelle Abgaben)
mengenAngaben	string	kWh (aktuelle Angaben)
verbrauchVon	int	Verbrauch ab
verbrauchBis	int	Verbrauch bis
dynamische Angaben	object[]	key/value Methode, siehe Objekt dynamische Angaben

5.21 Objekt-Definition IdObjGruppe

Zweck dieses Objektes ist es, auf mögliche zukünftige Anforderungen hinsichtlich Steuerung bzw. Filterung der Ausgabe reagieren zu können, ohne eine Strukturanpassung am Webservice durchführen zu müssen. Vorerst fungiert es als Platzhalter, eine Belegung erfolgt bei konkreten Anforderungen.

Parameter	Typ	Beschreibung
IntObj	object	siehe Objekt IntIdObj
DoubleObj	object	siehe Objekt DoubleIdObj
StringObj	object	siehe Objekt StringIdObj
DateObj	object	siehe Objekt DateIdObj

5.21.1 IntIdObj

Parameter	Typ	Beschreibung
BasisIdObj	object	siehe Objekt BasisIdObj
listenWert	int	Übergabe eines int-Listenwertes
wert	int	Übergabe eines int-Wertes

5.21.2 DoubleIdObj

Parameter	Typ	Beschreibung
BasisIdObj	object	siehe Objekt BasisIdObj
listenWert	double	Übergabe eines double-Listenwertes
wert	double	Übergabe eines double-Wertes

5.21.3 StringIdObj

Parameter	Typ	Beschreibung
BasisIdObj	object	siehe Objekt BasisIdObj
listenWert	string	Übergabe eines string-Listenwertes
wert	string	Übergabe eines string-Wertes

5.21.4 DateIdObj

Parameter	Typ	Beschreibung
BasisIdObj	object	siehe Objekt BasisIdObj
listenWert	date	Übergabe eines date-Listenwertes
wert	date	Übergabe eines date-Wertes

5.21.5 BasisIdObj

Parameter	Typ	Beschreibung
id	int	Übergabe einer ID
name	string	Übergabe eines Namen

6 Weitere Informationen

6.1 Ausgeschlossene Tarife

Fest ausgeschlossen für alle Methoden sind:

- Koppeltarife
- Tarife mit Genossenschaftsanteilen

Außerdem wird die Tarifberechnung in Abhängigkeit vom Login durchgeführt.

6.2 Brutto-Netto-Werte

Die Preise werden bei der Berechnung stets mit und ohne Mehrwertsteuer ausgegeben, unabhängig von der Kundenart.

Auf den Tarfkarten (Rohdaten) werden ebenfalls jeweils Netto- und Bruttowerte ausgegeben.

6.3 Stichtagsberechnung

Als Defaultwert für den Stichtag (Übergabe `<str:stichtag></str:stichtag>`) wird das aktuelle Datum verwendet. Historischen Abfragen (`stichtag < getDate`) sind nicht möglich, der Stichtag ist mindestens das aktuelle Datum.

Generell werden bei Abfragen eventuelle Vertriebsgültigkeiten von Tarifen berücksichtigt.

6.4 Vergleichstarif/Ersparnis

Die Ersparnis wird anhand des Vergleichstarifs ermittelt.

Als Default wird der günstigste Grundversorgungstarif des Gebiets ausgewählt. Ist der Grundversorgungstarif zum übergebenen Verbrauch nicht gültig, so wird der günstigste (Sonder-)Tarif des Grundversorgers heran gezogen. Darüber hinaus kann im request ein gewünschter Vergleichstarif angegeben werden. Dieser muss in der angefragten Lokation gültig sein, alle weiteren Parameter und etwaige Filter werden nicht auf ihn angewendet. Kann der gewünschte Vergleichstarif nicht heran gezogen werden, dann wird als Fallback immer die Defaulteinstellung angewendet.

6.4.1 Vergleichsrichtlinie

Hier sind die Vergleichsrichtlinien beschrieben, die genutzt werden können um die Ausgabe des Vergleichstarifs zu steuern:

- ID 1 = günstigster Grundversorgungstarif der PLZ (wie in Version 1.5)
- ID 2 = günstigstes Sonderprodukt des Grundversorger zu den genutzten Filtern
- ID 3 = günstigster Tarif zu dem in "vergleichsVersorgerID " gewählten Versorger zu den genutzten Filtern
- ID 4 = teuerstes Sonderprodukt des Grundversorger zu den genutzten Filtern
- ID 5 = teuerster Tarif zu dem in "vergleichsVersorgerID " gewählten Versorger zu den genutzten Filtern
- ID 6 = teuerster Tarif über alle Versorger außer dem Grundversorger zu den genutzten Filtern
- ID 7 = günstigster Tarif über alle Versorger außer dem Grundversorger zu den genutzten Filtern

6.5 Universalinformationen

Die Universalinformation besteht aus einem [Zusatzinformationsobjekt](#). Es ist für Informationen gedacht, die für eine Portaldarstellung benötigt werden, auch um ein etwaiges Backend abzulösen (z.B. AGB, Flyer, Logos, Farbcode). Es ist nicht dafür vorgesehen, Preisbestandteile oder Daten für die Tarifberechnung aufzunehmen und zu übertragen.

Einstieg ist die Nutzung der Methode [getBerechneteTarife](#), um an die nötigen Inputs für diese Methode zu gelangen.

6.5.1 AGB, Vertragsunterlagen, Logos

Die GET AG hat keine Datenbank mit Logos und AGB von Energieversorgungsunternehmen. Im Rahmen einer Vertriebspartnerschaft würde man Logos, AGB und Werbemittel vom jeweiligen Versorger erhalten. Über eine CSV Schnittstelle können die Vertragsunterlagen und Werbemittel dann über die entsprechenden IDs und Tarifcodes aus unserem Webservice eindeutig auf einer Webseite zugeordnet werden.

6.6 Sortierung und Paging

Das Ergebnis der Berechnungsmethoden ist nach dem Gesamtpreis aufsteigend sortiert.

Durch die Sortierung des Berechnungsergebnisses ist unter anderem Paging möglich. D.h. über die Angabe von Seitengröße und Seitennummer lassen sich genau die Tarife abrufen, welche in einem Ranking nach dem Gesamtpreis auf der entsprechenden Seite bei vorgegebener Anzahl von Tarifen pro Seite (Seitengröße) erscheinen. Im Ergebnis ist dabei stets der Vergleichstarif enthalten. Über den Parameter `ergebnisseCount` des Ergebnisses ist die Gesamtanzahl der berechneten Ergebnisse zu den übergebenen Parametern ersichtlich.

6.7 Filtern nach Marken IDs

Versorger führen mitunter Marken, denen einzelne oder alle Tarife zugeordnet sind. Über den Parameter `markenID` besteht die Möglichkeit, gezielt nach Tarifen einer Versorgermarke zu filtern.

Methode [getBerechneteTarife](#):

markenID > 0: es werden nur die Tarife eines Versorgers ausgegeben, die der MarkenID zugeordnet sind. Der Versorgername wird durch den Markennamen ersetzt.

markenID = 0 oder leer: es werden alle Tarife des Versorgers zurückgegeben. Sofern der Tarif einer Marke zugeordnet ist, wird diese ausgegeben und der Versorgername wird durch den Markennamen ersetzt.

Methode getTarifkarte: da hier stets ein bestimmter Tarif angezeigt werden soll, wird der Markenfilter in dieser Methode wie folgt angewendet:

markenID = 0: Die Tarifkarte wird nur zurückgegeben, wenn der gewählte Tarif tatsächlich keiner Marke zugeordnet ist.

markenID > 0: Die Tarifkarte wird nur zurückgegeben, wenn der Tarif tatsächlich der angegebenen MarkenID zugeordnet ist. Der Versorgername wird durch den Markennamen ersetzt.

markenID = NULL oder leer: die Tarifkarte wird stets zurückgegeben. Sofern der Tarif einer Marke zugeordnet ist, wird diese ausgegeben und der Versorgername wird durch den Markennamen ersetzt.

6.8 Tarif-Definitionen der GET AG

6.8.1 Ökogastarif

Ein Tarif ist ein Ökogastarif, wenn er laut Anbieter anteilig aus Biogas besteht oder eine Beteiligung an Klimaschutzprojekten vorliegt.

Reine Biogastarife sind durch die Angabe eines Biogasanteils in einem der BAI-Objekte gekennzeichnet.

6.8.2 Pakettarif

Ein Tarif ist ein Pakettarif, wenn der Versorger selbst ihn so bezeichnet.

In diesem Fall besteht der Tarif aus genau 2 Preisstaffeln:

In der ersten Staffeln gilt Grundpreis >0 (Paketpreis) und Arbeitspreis $=0$

In der zweiten Staffel gilt Grundpreis >0 (Paketpreis) und Arbeitspreis > 0 (Preis für Mehrverbrauch)

6.8.3 Fixpreistarif

Ein Tarif ist Fixpreistarif, wenn:

Hauptkriterium

- er nur bis zu einem bestimmten Datum erhältlich ist und
- er nur bis zu einem bestimmten Datum gilt (festes Vertragsende)

Nebenkriterium

- er meistens nicht gekündigt werden muss (läuft also ohne Kündigung aus)
- er ein bestimmtes festes Kontingent in kWh haben kann

6.8.4 Preisgarantie

Ein Tarif ist Tarif mit Preisgarantie, wenn:

Hauptkriterium

- er auch über die Preisgarantiefrist hinaus erhältlich ist und
- er kein festes Enddatum hat

Nebenkriterium

- die Preisgarantie für bestimmte Bestandteile des Tarifs (Steuern, Abgaben etc.) gilt

6.8.5 Energiepreistarife

Energiepreistarife beinhalten keine Kosten für Netznutzung, Abgaben und Umlagen. Der Energiepreis setzt sich zusammen aus Energiearbeitspreis und Energiegrundpreis. Um den Netto-Endverbrauchspreis zu erhalten, werden auf den Energiepreis folgende Komponenten hinzugerechnet: auf den

Energiearbeitspreis der Entgeltarbeitspreis sowie Abgaben und Umlagen. Auf den Energiegrundpreis der Entgeltgrundpreis sowie der Preis für Abrechnung, Betrieb, Messung und Ablesung. Für die Ausgabe von Energiepreistarifen bedarf es des Rechts „Netzentgelte“. Bitte sprechen Sie uns hierfür an.

6.9 Optionale Parameter/Objekte

Default sind die Request-Parameter optional anzugeben. Ist ein Parameter für die Anfrage erforderlich, so ist dies in der WSDL entsprechend gekennzeichnet (required). Die Kennzeichnung der Responseparameter und -objekte als optional richtet sich allein nach der technischen Spezifikation. So ist z.B. der Response Gesamtpreis als optional gekennzeichnet, da technisch möglich, inhaltlich wird dieser Parameter aber stets einen Wert enthalten

6.10 Portal-Richtlinie

1 = Als Portalrichtlinie „Allgemein“ werden die folgenden Kriterien für Tarife definiert:

- 1) Bonussumme exkl. Sofortbonus (brutto) < 100 € wird immer zugelassen
- 2) Wegfilterung der Tarife deren Sofortbonus (brutto) > 25 % des "Gesamtpreis exkl. Bonus (brutto)" sind
- 3) Wegfilterung der Tarife, deren "Neukundenbonus (brutto)" > 15 % des "Gesamtpreis exkl. Bonus (brutto)" sind.

3 = die Portalrichtlinien bleiben unberücksichtigt

Darüber hinaus können auch portalspezifische Richtlinien entworfen werden.

6.11 Fehlersuche

Folgende Fehlercodes können ausgegeben werden:

NA(1000, "Interner Datenfehler.")

FEHLER_1408(1408, "Der Parameter enthält ungültige Daten: ")

FEHLER_1500(1500, "Der Nutzer konnte nicht authentifiziert werden.")

FEHLER_1501(1501, "Nutzer konnte authentifiziert werden, hat aber nicht die erforderlichen Rechte.")

FEHLER_1502(1502, "Unbekannter Fehler bei der Authentifizierung des Nutzers.")

Bitte senden Sie zu jeder Anfrage an den Support der GET AG auch folgende Angaben mit:

- Webservice-Bezeichnung und –Versionsnummer
- FehlerCode
- iTag

6.12 ortsscharfe bzw. bis zu hausnummernscharfe Tarifzuordnung

Im Zusammenspiel mit den Postdaten bzw. demOrtsinformationswebservice der GET AG ist es möglich, die Tarife ortsscharf und teils auch hausnummernscharf zuzuordnen.

Es gelten die Orts- und Straßennamen in postschreibweise aus einer externen Quelle oder aus dem Ortsinformationswebservice (Version 1.0 oder Version 1.3) der GET AG.

Im Folgenden:

WS – Webservice

V – Version

6.12.1 Nutzung der Ortsschärfe

Nutzen Sie zusätzlich zu den bekannten Inputparametern des TarifWS V2.0 den Ort
=> fertig

Damit können Tarife, die im VIS Consult der GET AG kalkuliert und in unseren Datenbanken importiert wurden, bis zu ortsscharf innerhalb des Tarifwebservice ausgegeben werden.

6.12.2 Nutzung der bis zu hausnummernscharfen Tarifzuordnung

- 1.1 PLZ übergeben an den OrtsinformationsWS
(Methode getOrtsinfo (V 1.0) bzw. getOrtsinfoExtended (V1.3))
- 1.2 Test auf Anzahl Netze im Output

Anzahl Netze = 1

PLZ und Verbandsnummer des Netzbetreibers übergeben an den TarifWS
(Methode getBerechneteTarife)
=> fertig

Anzahl Netze > 1

- 2.1 PLZ und Ort übergeben in Methode getOrtsinfo (V 1.0) bzw.
getOrtsinfoExtended (V1.3)
- 2.2 Test auf Anzahl Netze im Output

Anzahl Netze = 1

PLZ und Verbandsnummer des Netzbetreibers übergeben an den
TarifWS (Methode getBerechneteTarife)
=> fertig

Anzahl Netze > 1

- 3.1 PLZ und Ort übergeben in Methode getStrassenByPlzOrt (nur
in V1.3), um Straßennamen zu erhalten
- 3.2 PLZ, Ort und Straße übergeben in Methode
getOrtsinfoExtended (nur in V1.3)
- 3.3 Test auf Anzahl Netze im Output

Anzahl Netze = 1

PLZ und Verbandsnummer des Netzbetreibers übergeben an
den TarifWS (Methode `getBerechneteTarife`)
=> fertig

Anzahl Netze > 1

keine weitere Unterteilung möglich
PLZ übergeben an den TarifWS (Methode
`getBerechneteTarife`)
⇒ man erhält alle Tarife zurück

6.13 Verwendung der Bulkberechnung

Die Methoden `getBerechneteGrundversorgungTarife` und `getBerechneteTarife` ermöglichen eine Bulk-Berechnung zur effizienteren Abfrage größerer Datenmengen.

Es ist möglich innerhalb der Webservice-Anfrage mehrere Param-Objekte zu übergeben. Im Output werden dann ebenfalls mehrere Objekte zurückgegeben, zur Identifizierung der Anfragen kann im Parameter `uid` eine eindeutige Kennung (Typ `string`) übergeben werden.

6.14 Rundungen über berechnungsKonfiguration

In den Methoden [getBerechneteGrundversorgungsTarife](#) und [getBerechneteTarife](#) können optional über eine berechnungsKonfiguration Angaben über die gewünschte Rundung gemacht werden. Dabei ist der wert="X" folgendermaßen zu setzen:

Request - Auszug	Art der Rundung
<code><objs:berechnungsKonfiguration id="20" name="Rundungsmodus" wert="0"> </objs:berechnungsKonfiguration></code>	Rundung für Rohdaten und berechnete Wert ist deaktiviert . Es werden alle Nachkommastellen ausgegeben
<code><objs:berechnungsKonfiguration id="20" name="Rundungsmodus" wert="1"> </objs:berechnungsKonfiguration></code>	Ist gleichzeitig die Defaulteinstellung , wenn die Berechnungskonfiguration nicht übergeben wird. Rohdaten werden mit 4 Nachkommastellen und berechnete Werte mit 2 Nachkommastellen gerundet ausgegeben. Fehlende Nachkommastellen werden mit 0en aufgefüllt.
<code><objs:berechnungsKonfiguration id="20" name="Rundungsmodus" wert="2"> </objs:berechnungsKonfiguration></code>	Rohdaten werden mit 2 Nachkommastellen und berechnete Werte mit 2 Nachkommastellen gerundet ausgegeben. Fehlende Nachkommastellen werden mit 0en aufgefüllt.
<code><objs:berechnungsKonfiguration id="20" name="Rundungsmodus" wert="3"> </objs:berechnungsKonfiguration></code>	Rohdaten werden mit 3 Nachkommastellen und berechnete Werte mit 3 Nachkommastellen gerundet ausgegeben. Fehlende Nachkommastellen werden mit 0en aufgefüllt.

Beispiel für Ausgabe von ungerundeten Werten:

```
...
<paramObjListe sortierung="1">
<objs:berechnungsKonfiguration id="20" name="Rundungsmodus" wert="0">
</objs:berechnungsKonfiguration>
<objs:parameterObj
```

...

6.15 Änderungshistorie

6.15.1 V1.3

- allgemeine Performancesteigerung
- Login-abhängige Tarifberechnung umgesetzt
- Überarbeitung der Inputparameter der Methode [getBerechneteTarife](#)
- 6-Wochenfenster hinzugefügt
- Erweiterung des [BAI](#)-Objekts
- Einführung der [HandlingDetails](#)
- Markenzuordnung hinzugefügt
- komplette Überarbeitung der [Tarifkarte](#) und der [Versorgerkarte](#)

6.15.2 V1.4

- neue Methoden: [getVersorger](#), [getVersorgerTarifvarianten](#), [getParameter](#)
- Erweiterung des Inputs der Methode [getBerechneteTarife](#), z.B.
 - Filter für Versorger, Preisgarantiedauer
 - Paging
- Erweiterung des Outputs von [Ergebnis](#), z.B.
 - Leistungspreis
 - Öko-Zertifikate
 - Brutto-Netto-Werte von Kundenart abhängig
- 6-Wochenfenster entfernt; spezielle Anzeigezeiträume von Tarifen werden jetzt WS-intern berücksichtigt
- produktübergreifende Vereinheitlichung und Bereinigung von Parameterbezeichnungen
- Apache Axis2/Java Version 1.6.0 eingebunden

6.15.3 V1.5

- Erweiterung des Requests der Methode [getBerechneteTarife](#), z.B.
 - Netzbetreiber

- Tariftyp
- Handling
- Portalrichtlinien
- Kündigungsfrist
- Zertifikate
- Erweiterung des Responses von Ergebnis, z.B.
 - Fehlermeldung
- Erweiterung des Responses der Methode getVersion, z.B.
 - Datenbankstand
- Erweiterung des Requests der Methoden getBerechneteTarifebyPLZundVerbrauchKundenart und getBerechneteGrundversorgungTarifebyPLZundVerbrauchKundenart, z.B.
 - Stichtag
- Erweiterung des Requests der Methode getHandlingsDetails
 1. Ort
 2. Netzbetreiber
 3. Grundversorger
- Optimierung der Doku

6.15.4 V2.0

- Entfernen der Methode „getBerechneteTarifebyPLZundVerbrauchKundenart“
- Nutzung der Middleware JAX anstatt AXIS
- Änderung des Filters „portalRichtlinie“
- Umbenennen der Methode „getBerechneteGrundversorgungTarifebyPLZundVerbrauchKundenart“ in „getBerechneteGrundversorgungTarife“
- PLZ im Request der Methode getVersorgerTarifvarianten nun optional, um lokationsunabhängig alle Tarife eines Versorgers zu erhalten
- Erweiterung des Request der Methode getVersorgerTarifvarianten um
 - Ort

- Netzbetreiber
 - Grundversorger
 - tariftyp
- Erweiterung des Response der Methode [getVersorgerTarifvarianten](#) um isGrundversorgungstarif
- Erweiterung des Requests der Methode [getBerechneteGrundversorgungstarife](#):
 - tariftyp
 - oeko
 - uid
 - grenzenBeiATignorierenBis
 - netzbetreiber
 - grundversorger
 - versorgerIDs
- Erweiterung des Response der Methode [getBerechneteGrundversorgungstarife](#):
 - uid
- Möglichkeit mittels [getBerechneteGrundversorgungstarife](#) die Grundversorgungstarife bundesweit zu berechnen
- Hinzufügen der Methode [getUniversalinformation](#)
- Erweiterung des Requests der Methode [getBerechneteTarife](#)
 - markenIDs
 - tarifCode
 - tarifRootIDsMindestfilter
 - vertriebskanal
 - regionalVersorgerUmkreis
 - uid
 - BAIextendedWhiteList
 - BAIextendedBlackList
 - details
 - vergleichsRichtlinie
 - vergleichsVersorgerID

- sortierung
- tarifRootIDs
- sachbonus
- kennzahl
- netzgebiet
- bonus ID 7
- Erweiterung des Response der Methode [getBerechneteTarife](#)
 - tarifCode
 - uid
 - art
 - artID
 - calcWert
 - netzentgeltBerechnung
 - margeBerechnung
 - kennzahlErgebnis
 - ergebnisContainer
 - versorgerInformation
- Entfernen der Methode [getHandlingDetails](#)
- Erweiterung des Response um Boni als [BAI Objekt](#)
- Erweiterung des Request der Methode [getVersorger](#) um
 - versorgerIDs
 - verbandsnummer
 - versorgerName
 - versorgerSitz
- Erweiterung des Response der Methode [getUniversalinformation](#) um
 - netzentgeltBerechnung
- Erweiterung des Requests der Methode [getTarifkarte](#) um
 - rootid
 - Stichtag
 - plz
 - ort
 - netzbetreiber

- grundversorger
- variantenID
- ausgabeKonfiguration
- tarifID im Request entfernt
- Erweiterung des Response der Methode [getTarifkarte](#) um
 - plz
 - ort
 - netzbetreiber
 - grundversorger
 - variantenID
 - tarifcode
- Erweiterung des Requests und Response aller Methoden um das Objekt „[dynamische Angaben](#)“
- [getBerechneteGrundversorgungsTarife](#) und [getBerechneteTarife](#) ermöglichen nun eine Bulk-Berechnung
- Umstellung auf [Basis Authentifizierung](#)
- Erweiterung um die Methode [getAbgaben](#)
- Ausgabe von Energiepreistarifen sofern das Recht „Netzentgelte“ vorhanden ist