

Strom Tarif Web Service Dokumentation

Version 2.0.10

Leipzig, 30.07.2025

GET AG

In der Alten Strickwarenfabrik

Berliner Straße 65 / 04129 Leipzig

Telefon +49 341 39373-800 / Fax -801

get-ag.com / info@get-ag.com

Inhaltsverzeichnis

Inhaltsverzeichnis.....	2
Dokumentenhistorie.....	5
1 Beschreibung	6
2 Einbindung und API Beschreibung	6
3 Methoden-Übersicht	7
3.1 getBerechneteGrundversorgungsTarife.....	7
3.2 getBerechneteTarife	7
3.3 getBundesstrommix.....	7
3.4 getParameter	8
3.5 getTarifkarte.....	8
3.6 getUniversalinformation	8
3.7 getVersion.....	8
3.8 getVersorger	9
3.9 getVersorgerkarte	9
3.10 getVersorgerTarifvarianten	9
3.11 getAbgaben	9
4 Fehlermeldungen	10
5 Objekt-Definitionen	10
5.1 BasisAntwort	10
5.2 Meldung	10
5.3 getBerechneteGrundversorgungsTarife Request	10
5.4 getBerechneteTarife Request	14
5.5 getBerechneteTarife Response	23
5.5.1 ErgebnisContainer	23
5.6 getBundesstrommix Request	31
5.7 getBundesstrommix Response	31
5.8 getParameter Request	32
5.9 getParameter Response	32
5.9.1 Typ	32
5.9.2 Param.....	32
5.10 getTarifkarte Request	33
5.11 getTarifkarte Response	34
5.11.1 TarifkarteAntwort.....	34

5.11.2	TarifkarteContainer	34
5.11.3	TarifkarteDaten.....	34
5.11.4	TarifkarteBAI	36
5.11.5	KundenArten	37
5.11.6	ZeitZone	37
5.11.7	PreisStaffel.....	39
5.11.8	Strommix nach §42 EnWG.....	39
5.11.9	StrommixLight	41
5.11.10	ZählerKosten	41
5.12	getUniversalinformation Request	42
5.13	getUniversalinformation Response	44
5.13.1	getUniversalinformation Ergebnis	44
5.13.2	Unterobjekte	44
5.14	getVersion Response	45
5.15	getVersorger Request	45
5.16	getVersorger Response	46
5.16.1	Versorger	46
5.17	getVersorgerkarte Request	46
5.18	getVersorgerkarte Response	47
5.19	getVersorgerTarifvarianten Request.....	47
5.20	getVersorgerTarifvarianten Response.....	48
5.21	getAbgaben Request.....	49
5.22	getAbgaben Response.....	49
5.22.1	Abgabe	49
6	Weitere Informationen.....	50
6.1	Ausgeschlossene Tarife	50
6.2	Brutto-Netto-Werte	50
6.3	Stichtagsberechnung	51
6.4	Vergleichstarif/Ersparnis	51
6.4.1	Vergleichsrichtlinie	51
6.5	Universalinformationen	52
6.5.1	AGB, Vertragsunterlagen, Logos.....	52
6.6	Sortierung und Paging	53
6.7	Filtern nach Marken IDs	53
6.8	Abfrage von Heizstrom-Tarifen.....	54

6.9	Tarif-Definitionen der GET AG	55
6.9.1	Ökostromtarif	55
6.9.2	Pakettarif	56
6.9.3	Fixpreistarif	56
6.9.4	Preisgarantie	56
6.9.5	Smart-Meter-Tarif	57
6.9.6	Energiepreistarif.....	57
6.9.7	Elektromobilität	57
6.9.8	Börsenpreisabhängige Stromtarife.....	58
6.10	Optionale Parameter/Objekte.....	59
6.11	Portal-Richtlinie.....	59
6.12	Fehlersuche	60
6.13	ortsscharfe bzw. bis zu hausnummernscharfe Tarifzuordnung	62
6.13.1	Nutzung der Ortsschärfe	62
6.13.2	Nutzung der bis zu hausnummernscharfen Tarifzuordnung	63
6.14	Verwendung der Bulkberechnung	64
6.15	Objekt-Definition IdObjGruppe.....	64
6.15.1	IntIdObj.....	65
6.15.2	DoubleIdObj	65
6.15.3	StringIdObj	65
6.15.4	DateIdObj	65
6.15.5	BasisIdObj	65
6.16	Rundungen über dynamischeAngaben.....	66
6.17	Börsenpreiskonfiguration über dynamische Angaben.....	67
6.18	Änderungshistorie	67
6.18.1	V1.3	67
6.18.2	V1.4	68
6.18.3	V1.5	68
6.18.4	V2.0	69

Dokumentenhistorie

Version	Datum	Änderung von	Bemerkung
2.0.1	10.05.2016	Sandy Quoos	Erstellung
2.0.2	01.08.2017	Daniel Sonne	Erweiterung Kennzahl
2.0.3	24.04.2018	Anja Weber	Austausch SSL Zertifikat
2.0.4	26.04.2018	Anja Weber	Erweiterung Erreichbarkeit Testservice
2.0.5	26.09.2018	Daniel Sonne	neue KuendigungsfristID=6 - „1 Monat zum Monatsende“
2.0.6	06.01.2020	Patrick Schmorde	Ergänzende Informationen zur Methode <i>getVersorgerkarte</i>
2.0.7	25.02.2020	Patrick Schmorde	Auslagerung technische Dokumentation
2.0.8	20.06.2023	Daniel Sonne	Erweiterung einstellbare Rundungen
2.0.9	02.09.2024	Gregor Jurke	Erweiterung um börsenpreisabhängige Tarife (dynamische Stromtarife und Flextarife)
2.0.10	30.07.2025	Daniel Sonne	Kleine Anpassung in der Dokumentation des Requestparameter "tariftyp"

1 Beschreibung

Die GET AG stellt einen Webservice zu Stromtarifen 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

<http://webservicetest1.ag-server.de/getagWSCore/jaxws/StromTarifService20x0?wsdl>

WADL für Testservice

<http://webservicetest1.ag-server.de/getagWSCore/jaxrs/StromTarifService20x0/application.wadl?detail=true>

WSDL für produktiven Service (SSL)

<https://webservice1.ag-server.de/getagWSCore/jaxws/StromTarifService20x0?wsdl>

WADL für produktiven Services (SSL)

<https://webservice1.ag-server.de/getagWSCore/jaxrs/StromTarifService20x0/application.wadl?detail=true>

Weiterhin steht Ihnen eine API Dokumentation per Swagger zur Verfügung:

Produktiver Service

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

Test-Service

<http://webservicetest1.ag-server.de/getagWSCore/swagger-ui.html?urls.primaryName=Strom%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

Grundsätzlich sind die Request-Parameter optional. Dort wo ein Parameter zwingend übergeben werden muss, ist dies entsprechend gekennzeichnet.

3.1 getBerechneteGrundversorgungTarife

Eingabe: <[getBerechneteGrundversorgungTarife Request](#)>

Ausgabe: <[getBerechneteTarife Response](#)>

Mit dieser Methode können mit nur wenigen Requestparametern Privat- oder Gewerbe-Grundversorgungstarife zu einer PLZ oder bundesweit 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 getBundesstrommix

Eingabe: < [getBundesstrommix Request](#) >

Ausgabe: <[getBundesstrommix Response](#)>

Die Methode liefert die Zusammensetzung des Gesamtstrommixes von Deutschland.

3.4 getParameter

Eingabe: <[getParameter Request](#)>

Ausgabe: <[getParameter Response](#)>

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

3.5 getTarifkarte

Eingabe: <[getTarifkarte Request](#)>

Ausgabe: <[getTarifkarte Response](#)>

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

3.6 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.

3.7 getVersion

Eingabe: <>

Ausgabe: <[getVersion Response](#)>

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

3.8 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.

3.9 getVersorgerkarte

Eingabe: <[getVersorgerkarte Request](#)>

Ausgabe: <[getVersorgerkarte Response](#)>

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

Dabei werden nur die Versorger berücksichtigt, welche am Tag der Anfrage **aktiv** sind. Zu inaktiven, also abgeschalteten Versorgern werden keine Informationen geliefert.

3.10 getVersorgerTarifvarianten

Eingabe: <[getVersorgerTarifvarianten Request](#)>

Ausgabe: <[getVersorgerTarifvarianten Response](#)>

Die Methode liefert alle gültigen Tarifvarianten der entsprechenden Lieferanten.

3.11 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	string	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. Ohne Übergabe einer PLZ werden die bundesweiten Grundversorgungstarife

		berechnet.
verbrauchHT	int	Gesamtverbrauch oder Verbrauch Hauptzeittarif in kWh
verbrauchNT	int	Verbrauch Nebenzeittarif in kWh; sonst 0 (Gesamtverbrauch)
stichtag	date	Stichtag für Tarifgültigkeit (Format: 2016-01-27); keine historischen Abfragen möglich; siehe Informationen zum Stichtag
kundenart	int	0 = Privat; 1 = Gewerbe; 2 = Landwirtschaft; 3 = Gemein- schaftsanlagen; 4 = Sonderabnehmer
tariftyp	string	Liste von IDs: 1 = Normal; 2 = Schwachlast; 3 = Wärmepumpe; 4 = Heizstrom 5 = Elektromobilität privater Ladetarif Bsp.: >1,2< Bei keiner Übergabe / leer werden alle IDs berücksichtigt!
oeko	int	0 = nein; 1 = ja; 3 = egal; (0 = keine Ökoprodukte, 1 = nur Ökoprodukte, 3 = Ökoprodukte und Normalprodukte); siehe Information zu Ökotarif
zaehlerart	string	Liste von IDs (siehe getParameter) z.B.: 1 = EintarifzaehlerWechselstrom; 2 = EintarifzaehlerDrehstrom; 3 = ZweitarifzaehlerWechselstrom; 4 = ZweitarifzaehlerDrehstrom; 5 = SmartMeter EDL21;

		...
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
versorgerIDs	string	Liste von VersorgerIDs (beschränkt das Ergebnis auf Tarife der übergebenen Versorger); Bsp.: >123,456,<
netzbetreiber	string	Verbandsnummer des Netzbetreibers
grundversorger	string	Verbandsnummer des Grundversorgers
netzgebiet	string	Interne Netznummer der GET AG
dynamischeAngaben	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	10

vergleichsVersorgerID	
fixpreis	3
vorkasse	3
paket	3
portalRichtlinie	3
bonus	3
preisgarantie	3
kautions	3
sonderabschlag	3
smartmeter	0
vertragslaufzeitBis	0
vertragsverlaengerung	0
kuendigungsfrist	0
bestandskundenTarif	0
preisgarantieDauer	0
zertifikate	
seitenGroesse	0
seitenNr	1
lastprofile	
markenIDs	
tarifcode	
tarifRootIDsMindestfilter	
tarifRootIDs	
vertriebskanalIDs	
bAIextendedWhiteList	
bAIextendedBlackList	
sortierung	1
kennzahl	0

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

5.4 getBerechneteTarife Request

Parameter	Typ	Beschreibung
sortierung	int	0 = Sortierung nach gesamtpreisOhneBoni 1 = Sortierung nach gesamtpreis 2 = Sortierung nach Gesamtpunktzahl der Kennzahlen absteigend, gesamtpreis 3 = Sortierung nach Kennzahlgruppe "Preis" absteigend, gesamtpreis 4 = Sortierung nach Kennzahlgruppe "Qualität" absteigend, gesamtpreis 5 = Sortierung nach Kennzahlgruppe "Kundenbewertung" absteigend, gesamtpreis 6 = Sortierung nach Gesamtpunktzahl der Kennzahlen absteigend, gesamtpreisOhneBoni 7 = Sortierung nach Kennzahlgruppe "Preis" absteigend, gesamtpreisOhneBoni 8 = Sortierung nach Kennzahlgruppe "Qualität" absteigend, gesamtpreisOhneBoni 9 = Sortierung nach Kennzahlgruppe "Kundenbewertung" absteigend, gesamtpreisOhneBoni
plz	string	Postleitzahl der Verbrauchsstelle
ort	string	Ortsname der Verbrauchsstelle
netzbetreiber	string	Verbandsnummer des Netzbetreibers
grundversorger	string	Verbandsnummer des

		Grundversorgers
netzgebiet	string	Interne Netznummer der GET AG
verbrauchHT (erforderlich)	int	Gesamtverbrauch oder Verbrauch Hauptzeittarif in kWh
verbrauchNT	int	Verbrauch Nebenzeittarif in kWh; sonst 0 (Gesamtverbrauch)
kundenart	int	0 = Privat; 1 = Gewerbe; 2 = Landwirtschaft; 3 = Gemeinschaftsanlagen; 4 = Sonderabnehmer
tariftyp	string	Liste von IDs; 1 = Normal; 2 = Schwachlast; 3 = Wärmepumpe; 4 = Heizstrom; 5 = Elektromobilität privater Ladetarif Bsp.: >1,2<
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 Wird keine vergleichsRichtlinie übergeben, dann wird default die ID 10 angewendet.
vergleichsVersorgerID	int	Sucht Vergleichstarif nur zu der übergebenen VersorgerID und berücksichtigt dabei die anderen Requestparameter
stichtag	date	Stichtag für Tarifgültigkeit (Format: 2010-01-27); keine historischen Abfragen möglich; siehe Informationen zum Stichtag
oeko	int	0 = nein; 1 = ja; 3 = egal; (0 = keine Ökoprodukte, 1 = nur Ökoprodukte, 3 = Ökoprodukte und Normalprodukte); 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
smartmeter	int	0 = nein; 1 = ja; 3 = egal
vertragslaufzeitBis	int	maximale Mindestlaufzeit in Monaten; 0 = ohne Beschränkung
vertragsverlaengerung	int	maximale 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; 6 = 1 Monat zum Monatsende
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

		4 = RECS Renewable Energy Certificate System 5 = OK Power 9 = Grüner Strom Label e.V. 12 = Ökostromsiegel der LGA (Landesgewerbeanstalt Bayern) 13 = TÜV - Nord 14 = TÜV - Süd 15 = TÜV-Hessen 16 = TÜV-Thüringen 17 = TÜV-Rheinland 18 = Grüner Strom Label e.V. GOLD 19 = Grüner Strom Label e.V. SILBER 20 = EcoTopTen 21 = RenewablePLUS 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
lastprofile		Noch in Planung
markenIDs	string	Liste von markenIDs (beschränkt das Ergebnis auf Tarife der übergebenen Marken); Bsp.: >1,6,< siehe Filterung nach MarkenIDs
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,<
zaehlerart	string	Liste von IDs, z.B.: 1 = EintarifzaehlerWechselstrom; 2 = EintarifzaehlerDrehstrom; 3 = ZweitarifzaehlerWechselstrom; 4 = ZweitarifzaehlerDrehstrom; 5 = SmartMeter EDL21; siehe getParameter
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
kennzahl	int	für die individuelle Erstellung von Indizes über die Ergebnisse Kennzahl ID=1 Simplex
dynamischeAngaben	object[]	key/value Methode, siehe Objekt dynamischeAngaben
details	int	Ermöglicht die Anzeige zusätzlicher Details zum Tarif und/oder einzelner Preiskomponenten: 0 = keine Details (default Wert) 1 = detaillierte Aufstellung der Netzentgeltberechnung bei Energiepreistarifen (Börsenpreis/Beschaffungspreis

		enthalten in Responsewerten „XXXEnergiepreisYYY“) 2 = detaillierte Aufstellung der Netzentgeltberechnung UND der gewählten Beschaffungspreisberechnung (Konfiguration steuerbar über "boersenPreisConfig"-Parameter; (Börsenpreis/Beschaffungspreis nicht enthalten in Responsewerten „XXXEnergiepreisYYY“) Zur Nutzung von nicht-default-Werten dieses Parameters bedarf es des Rechts "dynamische Stromtarife".
boersenPreisConfig (in den dynamischen Angaben des Requests) (!!!) Achtung: Der Wert "1" des boersenPreisConfig-Parameters ist primär dafür gedacht, dynamische Stromtarife (Tarife mit der BAI=208) preislich greifbar zu machen, und sollte idealerweise auch nur für solche verwendet werden, da er eine in die Vergangenheit gerichtete Preisberechnung bedeutet, wogegen klassische Stromtarife i.d.R. in die	int	Dieser Parameter muss im Bereich der dynamischen Angaben mitgegeben werden (Codebeispiel siehe dynamischeAngaben im Abschnitt 6.17): Steuerung der Ausgabe des Beschaffungspreis- bzw. Börsenpreis-Parameters bei dynamischen Stromtarifen: 1 = der Beschaffungspreis entspricht dem Durchschnittswert der Börsenpreise der letzten 365 Tage (vor dem eingegebenen Stichtag) welche nach dem dynamischen BDEW H0 Standardlastprofil gewichtet wurden 2 = der Beschaffungswert wird per se

Zukunft gepreist werden.		auf den Wert 0 gesetzt, es werden also sämtliche Preisbestandteile mit Ausnahme des Börsenpreises ausgegeben
--------------------------	--	--

5.5 getBerechneteTarife Response

5.5.1 ErgebnisContainer

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
marktrolle	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 Ergebnis

Parameter	Typ	Beschreibung
anzeigeBis	date	Vertriebsende des Tarifs
anzeigeVon	date	Vertriebsbeginn des Tarifs
arbeitspreisNetto	decimal	Verbrauchsarbeitspreis in Euro/Jahr netto
arbeitspreisBrutto	decimal	Verbrauchsarbeitspreis in Euro/Jahr brutto
arbeitspreisHTCtKWhNetto	decimal	Arbeitspreis oder HT-Arbeitspreis in ct/kWh netto
arbeitspreisHTCtKWhBrutto	decimal	Arbeitspreis oder HT-Arbeitspreis in ct/kWh brutto
arbeitspreisNTCtKWhNetto	decimal	NT-Arbeitspreis in ct/kWh netto; NULL bei ET
arbeitspreisNTCtKWhBrutto	decimal	NT-Arbeitspreis in ct/kWh brutto; NULL bei ET
aufschlaege (0:n Beziehung)	object	Aufschläge des Tarifs als Objekt berechneteTarifeBAI
boni (0:n Beziehung)	object	Boni des Tarifs als Objekt berechneteTarifeBAI
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	Gültigkeitsende des Tarifs
gueltigVon	date	Gültigkeitsbeginn des Tarifs
informationen (0:n Beziehung)	object	Informationen zum Tarif als Objekt berechneteTarifeBAI
isGrundversorgungstarif	boolean	ist Grundversorgungstarif
isOekostromTarif	boolean	ist Ökostromtarif; 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
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
strommixLight (0:n Beziehung)	object	siehe Objektbeschreibung StrommixLight

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
arbeitspreisEnergiepreis HTNetto	decimal	Energiepreis Verbrauchsarbeitspreis HT in Euro/Jahr netto
arbeitspreisEnergiepreis HTBrutto	decimal	Energiepreis Verbrauchsarbeitspreis HT in Euro/Jahr brutto
arbeitspreisEnergiepreis NTNetto	decimal	Energiepreis Verbrauchsarbeitspreis NT in Euro/Jahr netto
arbeitspreisEnergiepreis NTBrutto	decimal	Energiepreis Verbrauchsarbeitspreis NT in Euro/Jahr brutto
grundpreisEnergiepreisNetto	decimal	Energiepreis Grundpreis in Euro/Jahr netto
grundpreisEnergiepreisBrutto	decimal	Energiepreis Grundpreis in Euro/Jahr brutto
arbeitspreisEnergiepreis HTCtKWhNetto	decimal	Energiepreis Arbeitspreis oder Energiepreis HT-Arbeitspreis in ct/kWh netto
arbeitspreisEnergiepreis HTCtKWhBrutto	decimal	Energiepreis Arbeitspreis oder Energiepreis HT-Arbeitspreis in

		ct/kWh brutto
arbeitspreisEnergiepreis NTCtKWhNetto	decimal	Energiepreis Arbeitspreis oder Energiepreis NT-Arbeitspreis in ct/kWh netto
arbeitspreisEnergiepreis NTCtKWhBrutto	decimal	Energiepreis Arbeitspreis oder Energiepreis NT-Arbeitspreis in ct/kWh brutto
kennzahlErgebnis	object	siehe Objektbeschreibung kennzahlErgebnis
netzEntgeltBerechnung	object	siehe Objektbeschreibung netzEntgeltBerechnung
margeBerechnung	object	siehe Objektbeschreibung margeBerechnung
dynamischeAngaben	object[]	key/value Methode, siehe Objekt dynamischeAngaben

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 Typ `Param_Tarif_Strom_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
apHT	double	Arbeitspreis HT in ct/kWh netto
apNT	double	Arbeitspreis NT in ct/kWh netto
entgeltFuerAbrechnung	double	Abrechnungskosten inkl. zusätzliche Abrechnungen in €/Jahr netto
entgeltFuerEinbauBetriebWartungMesstechnik	double	Messstellenbetriebskosten in €/Jahr netto
entgeltFuerMessungUndAbleseung	double	Summe aus Mess- und Ablesungskosten in €/Jahr netto
entgeltInformationen	object	Siehe Objekt EntgeltInformationen
entgeltZaehlerpreisAbleseung	double	Summe aus entgeltFuerEinbauBetriebWartungMesstechnik und entgeltFuerMessungUndAbleseung in €/Jahr netto, da es VNB gibt, die keine Aufsplittung der Messpreise veröffentlichen
entgelttyp	int	Entgelttyp der zugrundeliegenden Kalkulation
gesamtpreis	double	Gesamtpreis Entgelt in €/Jahr netto
grundpreis	double	Grundpreis in €/Jahr netto
kaHT	double	Konzessionsabgabe zum Hochtarif in ct/kWh
kaNT	double	Konzessionsabgabe zum Niedertarif in ct/kWh
leistungspreis	double	Leistungspreis in €/Jahr netto
steuern	object	Siehe Objekt Steuern
zaehlerart	int	Zählerart der zugrundeliegenden

		Kalkulation
spannungsebeneEntnahme	int	Spannungsebene für Entnahme, siehe getParameter
spannungsebeneMessung	int	Spannungsebene für Messung, siehe getParameter
dynamischeAngaben	object[]	key/value Methode, siehe Objekt dynamischeAngaben

5.5.1.2.3.1 entgeltInformationen

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	double	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 (Energiesteuer, Stromsteuer, Umsatzsteuer)
wert	double	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

5.6 getBundesstrommix Request

Aufruf ohne weitere Parameter!

5.7 getBundesstrommix Response

Parameter Ebene 1	Parameter Ebene 2	Typ	Beschreibung
BasisAntwort		object	Siehe Objekt BasisAntwort
Erneuerbare		object	erneuerbare Energien
	eeg	decimal	Anteil geförderter erneuerbarer Energien nach dem Erneuerbare-Energien-Gesetz in Prozent
	gesamt	decimal	Anteil aller erneuerbaren Energien in Prozent
	sonstige	decimal	Anteil sonstiger erneuerbarer Energien in Prozent
Fossile		object	fossile Energieträger
	erdgas	decimal	Anteil Erdgas in Prozent
	gesamt	decimal	Anteil aller fossiler Energieträger in Prozent
	kohle	decimal	Anteil Kohle in Prozent
	sonstige	decimal	Anteil sonstiger fossiler Energieträger in Prozent
kernkraft		decimal	Anteil Kernenergie in Prozent
radioaktiv		decimal	radioaktiver Abfall in g/kWh
co2		decimal	CO2-Emissionen in g/kWh

stand		int	Jahr des Datenstands
dynamische Angaben		object[]	key/value Methode, siehe Objekt dynamischeAngaben

Die Informationen entsprechen dem §42 Abs. 2 des EnWG 2011 und sind gemäß §42 Abs. 1 des EnWG 2011 angegeben.

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
stichtag	date	Stichtag für Tarifgültigkeit (Format: 2010-01-27); keine historischen Abfragen möglich; siehe Informationen zum Stichtag
markenID	int	zum Zuordnen einer bestimmten Marke, siehe Filter nach MarkenIDs
plz (erforderlich)	string	Postleitzahl der Verbrauchsstelle
ort	string	Ortsname der Verbrauchsstelle
netzbetreiber	string	Verbandsnummer des Netzbetreibers
grundversorger	string	Verbandsnummer des Grundversorgers
netzgebiet	string	Interne Netznummer der GET AG
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
dynamischeAngaben	object[]	key/value Methode, siehe Objekt dynamischeAngaben

5.11 getTarifkarte Response

5.11.1 TarifkarteAntwort

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

5.11.2 TarifkarteContainer

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

5.11.3 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
fristVertragsabschluss	date	Vertragsabschlussfrist
fruehesteBelieferung	date	frühester Lieferbeginn
gueltigBis	date	Gültigkeitsende des Tarifs
gueltigVon	date	Gültigkeitsbeginn des Tarifs
informationen (0:n Beziehung)	object	Informationen zum Tarif als Objekt TarifkarteBAI
isBestandskundenTarif	boolean	ist Bestandskundentarif
isFixpreisTarif	boolean	ist Fixpreistarif
isPaketTarif	boolean	ist Pakettarif
isVorkasseTarif	boolean	ist Vorkassentarif

kontingentFixpreisangebot	int	Kontingent Fixpreisangebot in kWh
kuendigungsfrist	string	Kündigungsfrist als Freitext
kuendigungsfristID	int	ID des Kündigungsfreitextes
kundenArten (0:n Beziehung)	object	siehe Objektbeschreibung KundenArten
mindestrechnungspreisBrutto	decimal	Mindestrechnungspreis in Euro/Jahr Brutto
mindestrechnungspreisNetto	decimal	Mindestrechnungspreis in Euro/Jahr Netto
produkttyp	string	Produkttyp als Freitext
produkttypID	int	ID des Produkttypfreitextes
statusID	int	siehe Fehlermeldungen
strommix (0:n Beziehung)	object	siehe Objektbeschreibung Strommix
tarifArt	string	Tarifart als Freitext
tarifArtID	int	ID des Tarifartfreitextes
tarifID	int	ID des Tarifs
tarifInfo	string	Tarifinformationen am Stromtarif
tarifName	string	Name des Tarifs
tarifRootID	int	RootID des Tarifs
verbrauchGuenstigBis	int	empfohlene Verbrauchsobergrenze in kWh
verbrauchGuenstigVon	int	empfohlene Verbrauchuntergrenze in kWh
versorgerID	int	ID des Versorgers
versorgerName	string	Name des Versorgers oder vorrangig der Marke
vertragslaufzeit	string	Mindestlaufzeit als Freitext
vertragslaufzeitID	int	ID des Mindestlaufzeitfreitextes
vertragslaufzeitbegrenzung	int	Vertragslaufzeitbegrenzung in Monaten; 0 = unbegrenzt
vertragsverlaengerung	string	Vertragsverlängerung als Freitext

vertragsverlaengerungID	int	ID des Vertragsverlängerungsfreitextes
zaehlerKosten (0:n Beziehung)	object	siehe Objektbeschreibung Zählerkosten
zeitZone (0:n Beziehung)	object	siehe Objektbeschreibung ZeitZone
variantenID	string	ID der Tarifvariante in Abhängigkeit der BAIs
dynamischeAngaben	object[]	key/value Methode, siehe Objekt dynamischeAngaben

5.11.4 TarfkarteBAI

BAI-Objekt der Tarfkarte

Parameter	Typ	Beschreibung
bedingung	string	BAI-Bezeichnung
bedingungID	int	ID der BAI-Bezeichnung
beschreibungstext	string	Beschreibungstext
einheit	string	Einheit als Freitext
einheitID	int	ID der Einheit
gueltigBis	date	Gültigkeitsbeginn
gueltigVon	date	Gültigkeitsende
id	int	ID des BAIs
oekostromAnteil	decimal	Anteil des Ökostroms
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 (<i>0:n Beziehung</i>)	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
dynamischeAngaben	object[]	key/value Methode, siehe Objekt dynamischeAngaben

5.11.4.1 Zertifikate

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

5.11.5 KundenArten

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

5.11.6 ZeitZone

Parameter	Typ	Beschreibung
dsAbzugspauschale	decimal	Durchschnittspreis-Abzugspauschale
HoechstBrutto		Maximum in Euro/Jahr Brutto

dsAbzugspauschale HoechstNetto	decimal	Durchschnittspreis-Abzugspauschale Maximum in Euro/Jahr Netto
dsAbzugspauschale MindestBrutto	decimal	Durchschnittspreis-Abzugspauschale Minimum in Euro/Jahr Brutto
dsAbzugspauschale MindestNetto	decimal	Durchschnittspreis-Abzugspauschale Minimum in Euro/Jahr Netto
dsPreisHoechstBrutto	decimal	Obergrenze Durchschnittspreis in ct/kWh Brutto
dsPreisHoechstNetto	decimal	Obergrenze Durchschnittspreis in ct/kWh Netto
dsPreisMindestBrutto	decimal	Untergrenze Durchschnittspreis in ct/kWh Brutto
dsPreisMindestNetto	decimal	Untergrenze Durchschnittspreis in ct/kWh Netto
id	int	ID der Zeitzone
preisStaffel (<i>0:n Beziehung</i>)	object	siehe Objektbeschreibung PreisStaffel
zeitZonenTagTyp	string	Name des TagTyps
zeitZonenTagTypID	int	ID des TagTyps
zeitZonenTemplate	string	Name des ZeitZonenTemplates
zeitZonenTemplateID	int	ID des ZeitZonenTemplates
zeitZonenTyp	string	Name des ZeitzonenTyps
zeitZonenTypID	int	ID des ZeitzonenTyps
zeitZonenZeitraumBis	string	Uhrzeit (hh:mm), bis wann Preisstaffel gültig
zeitZonenZeitraumVon	string	Uhrzeit (hh:mm), ab wann Preisstaffel gültig
dynamischeAngaben	object[]	key/value Methode, siehe Objekt dynamischeAngaben

5.11.7 PreisStaffel

Parameter	Typ	Beschreibung
arbeitspreisBrutto	decimal	Arbeitspreis in ct/kWh Brutto
arbeitspreisNetto	decimal	Arbeitspreis in ct/kWh Netto
bezeichnung	string	Staffelbezeichnung
grundpreisBrutto	decimal	Grundpreis in Euro/Jahr Brutto
grundpreisNetto	decimal	Grundpreis in Euro/Jahr Netto
id	int	ID der Preisstaffel
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
verbrauchBis	int	Verbrauchsobergrenze in kWh
verbrauchVon	int	Verbrauchsuntergrenze in kWh
arbeitspreisNettoEnergiepreisHT	decimal	Energiepreis Arbeitspreis HT in ct/kWh Netto
arbeitspreisNettoEnergiepreisNT	decimal	Energiepreis Arbeitspreis NT in ct/kWh Netto
grundpreisNettoEnergiepreis	decimal	Energiepreis Grundpreis in Euro/Jahr Netto
tarifCode	string	Tarifcode des Tarifs
dynamischeAngaben	object[]	key/value Methode, siehe Objekt dynamischeAngaben

5.11.8 Strommix nach §42 EnWG

Parameter Ebene 1	Parameter Ebene 2	Typ	Beschreibung
Erneuerbare (0:n Beziehung)		object	Erneuerbare

	biomasse	decimal	Anteil von Biomasse in Prozent
	eeg	decimal	Anteil nach EEG gefördert in Prozent
	gesamt	decimal	Gesamtanteil erneuerbare Energien in Prozent
	solar	decimal	Anteil von Solarenergie in Prozent
	wasserkraft	decimal	Anteil von Wasserkraft in Prozent
	windenergie	decimal	Anteil von Windenergie in Prozent
	sonstige	decimal	Anteil sonstiger erneuerbarer Energien in Prozent
	dynamischeAngabe n	object[]	key/value Methode, siehe Objekt dynamischeAngaben
Fossile (0:n Beziehung)		object	Fossile
	braunkohle	decimal	Anteil von Braunkohle in Prozent
	erdgas	decimal	Anteil von Erdgas in Prozent
	gesamt	decimal	Gesamtanteil fossiler Energien in Prozent
	sonstige	decimal	Anteil von Sonstiges in Prozent
	steinkohle	decimal	Anteil von Steinkohle, oder auch Kohle gesamt, in Prozent
	dynamische Angaben	object[]	key/value Methode, siehe Objekt dynamischeAngaben
kernkraft		decimal	Anteil Kernenergie in Prozent
radioaktiv		decimal	radioaktiver Abfall in g/kWh
stand		int	Jahr des Datenstands
co2		decimal	CO2-Emissionen in g/kWh

dynamische Angaben		object[]	key/value Methode, siehe Objekt dynamischeAngaben
--------------------	--	----------	---

5.11.9 StrommixLight

Parameter	Typ	Beschreibung
co2	decimal	CO2-Emissionen in g/kWh
erneuerbare	decimal	Anteil erneuerbaren Energien in Prozent
fossile	decimal	Anteil fossile Energieträger in Prozent
kernkraft	decimal	Anteil Kernenergie in Prozent
radioaktiv	decimal	radioaktiver Abfall in g/kWh
stand	int	Jahr des Datenstands
dynamischeAngaben	object[]	key/value Methode, siehe Objekt dynamischeAngaben

5.11.10 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
gueltigBis (optional)	date	Gültigkeitsende
gueltigVon (optional)	date	Gültigkeitsbeginn
id	int	ID der Zählerkostenstaffel

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
stromwandlersatzBrutto	decimal	Kosten für den Stromwandlern in Euro/Jahr Brutto
stromwandlersatzNetto	decimal	Kosten für den Stromwandlern in Euro/Jahr Netto
tarifschaltungBrutto	decimal	Kosten für Tarifschaltung in Euro/Jahr Brutto
tarifschaltungNetto	decimal	Kosten für Tarifschaltung in Euro/Jahr Netto
zaehlerart	string	Typ des Zählers
zaehlerartID	int	ID des Zählertyps
zusatzAbrechnungBrutto	decimal	Kosten für zusätzliche Abrechnung in Euro Brutto
zusatzAbrechnungNetto	decimal	Kosten für zusätzliche Abrechnung in Euro Netto
dynamischeAngaben	object[]	key/value Methode, siehe Objekt dynamischeAngaben

5.12 getUniversalinformation Request

Parameter	Typ	Beschreibung
verbrauchHT	int	Gesamtverbrauch oder Verbrauch Hauptzeittarif in kWh
verbrauchNT	int	Verbrauch Nebenzeittarif in kWh; sonst 0 (Gesamtverbrauch)
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,<
rechercheTypIDs	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

5.13 getUniversalinformation Response

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

5.13.1 getUniversalinformation Ergebnis

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

5.13.2 Unterobjekte

5.13.2.1 zusatzInformation

Parameter	Typ	Beschreibung
rechercheTypeID	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

5.14 getVersion Response

Parameter	Typ	Beschreibung
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

5.15 getVersorger Request

Parameter	Typ	Beschreibung
stichtag	date	Stichtag (Format: 2010-01-27); 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
dynamischeAngaben	object[]	key/value Methode, siehe Objekt dynamischeAngaben

5.16 getVersorger Response

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

5.16.1 Versorger

Parameter	Typ	Beschreibung
versorger	object	siehe Objekt Versorger
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

5.17 getVersorgerkarte Request

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

5.18 getVersorgerkarte Response

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

Parameter	Typ	Beschreibung
BasisAntwort	Object	Siehe Objekt BasisAntwort
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
strasse	string	Straße
telefon	string	Telefon
verbandsnummer	string	13-stellige VDEWID oder ILN
dynamischeAngaben	object[]	key/value Methode, siehe Objekt dynamischeAngaben

5.19 getVersorgerTarifvarianten Request

Parameter	Typ	Beschreibung
plz	string	Postleitzahl der Verbrauchsstelle
Kundenart	int	0 = Privat; 1 = Gewerbe; 2 = Landwirtschaft; 3 = Gemeinschaftsanlagen; 4 = Sonderabnehmer
stichtag	date	Stichtag (Format: 2010-01-27); keine historischen Abfragen möglich;

		siehe Informationen zum Stichtag
verbrauch	int	Gesamtverbrauch in kWh
versorgerID	int	ID des Versorgers (wird zum Pflichtfeld, wenn im Request keine PLZ übergeben wird)
tariftyp	string	Liste von IDs; 1 = Normal; 2 = Schwachlast; 3 = Wärmepumpe; 4 = Heizstrom 5 = Elektromobilität privater Ladetarif < Bsp.: >1,2< Bei keiner Übergabe / leer werden diese IDs >1,2< berücksichtigt!
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)
dynamischeAngaben	object[]	key/value Methode, siehe Objekt dynamischeAngaben

5.20 getVersorgerTarifvarianten Response

Parameter Ebene 1	Parameter Ebene 2	Parameter Ebene 3	Typ	Beschreibung
BasisAntwort			object	Siehe Objekt BasisAntwort
Versorger			object	Versorger

	markenID		int	ID der Marke
	markenName		string	Name der Marke
	dynamischeAngaben		object[]	key/value Methode, siehe Objekt dynamischeAngaben
	Tarife		object	Tarife
		tarifID	int	ID des Tarifs
		tarifName	string	Name des Tarifs
		variantenID	string	ID der Tarifvariante
		isGrundversorgungstarif	boolean	ist Grundversorgungstarif
		sonstigeAngaben	Object[]	key/value Methode, siehe Objekt dynamischeAngaben
	versorgerID		int	ID des Versorgers
	versorgerName		string	Name des Versorgers

5.21 getAbgaben Request

Aufruf ohne weitere Parameter!

5.22 getAbgaben Response

Parameter	Typ	Beschreibung
BasisAntwort	object	Siehe Objekt BasisAntwort
Abgabe	object	Siehe Objekt Abgabe
Steuer	object	Siehe Objekt Abgabe

5.22.1 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	double	Preis
preisEinheit	string	Cent (aktuelle Abgaben)
mengenAngaben	string	kWh (aktuelle Angaben)

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. Historische 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

ID=10= günstigster GV Tarif der PLZ unter Berücksichtigung des gewählten Tariftyps.

ID=11= teuerster GV Tarif der PLZ unter Berücksichtigung des gewählten Tariftyps.

➔ Wird im Request keine vergleichsRichlinie übergeben, dann wird default die ID 10 angewendet.

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 Tarfkarte wird nur zurückgegeben, wenn der Tarif tatsächlich der angegebenen MarkenID zugeordnet ist. Der Versorgername wird durch den Markennamen ersetzt.

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

6.8 Abfrage von Heizstrom-Tarifen

Um Heizstromtarife abzufragen, verwenden Sie bitte den Tariftyp 4, Wärmepumpentarife erhalten Sie über den Tariftyp 3. Eine zusätzliche Hinzunahme von Tariftyp 2 (Schwachlasttarife) ist nicht erforderlich, diese werden bei Tariftyp 3 und 4 automatisch mit berücksichtigt. Die Tariftypen können grundsätzlich auch miteinander kombiniert werden.

Wird die Anfrage ausschließlich mit einem HT Verbrauch gestellt, wird zur Berechnung des Gesamtarbeitspreises ausschließlich der hinterlegte HT Arbeitspreis zugrunde gelegt. Wird sowohl mit HT als auch mit NT Verbrauch angefragt, wird der jeweils dafür hinterlegte Arbeitspreis zur Berechnung verwendet.

Nachfolgend ein Beispiel-Request für eine Heizstrom-Abfrage:

```
<paramObjListe
  sortierung="0">
<par:parameterObj
  plz="12529"
  ort=""
  grundversorger=""
  netzbetreiber=""
  netzgebiet=""
  verbrauchHT="3500"
  verbrauchNT="3500"
  kundenart="0"
  tariftyp="4"
  limit="0"
  grenzenBeiATignorierenBis="0"
  tarifeVonGrundversorgern="3"
```

```

grundversorgungstarife="3"
handlingTarife="3"
vergleichsTarifID="0"
oeko="3"
fixpreis="3"
vorkasse="3"
paket="3"
portalRichtlinie="3"
bonus="3"
preisgarantie="3"
kaution="3"
sonderabschlag="3"
smartmeter="3"
vertragslaufzeitBis="0"
vertragsverlaengerung="0"
kuendigungsfrist="0"
bestandskundenTarif="3"
preisgarantieDauer="0"
versorgerIDs=""
zertifikate=""
seitenGroesse="0"
seitenNr="0"
lastprofile=""
markenIDs=""
tarifcode=""
tarifRootIDsMindestfilter=""
tarifRootIDs=""
zaehlerart="3"
vertriebskanalIDs=""
uid=""
bAExtendedWhiteList=""
bAExtendedBlackList=""
details="0"
vergleichsRichtlinie="1"
>
</par:parameterObj>
</paramObjListe>

```

6.9 Tarif-Definitionen der GET AG

6.9.1 Ökostromtarif

Ein Tarif ist ein Ökostromtarif, wenn er zu 100% aus regenerativen Energien besteht oder aber aus 100% KWK (egal ob zertifiziert oder nicht). Voraussetzung dafür ist, dass der Anbieter seinen Tarif auch selbst als Ökostrom deklariert.

Ein Tarif ist auch ein Ökostromtarif, wenn er einen Anteil an KWK von maximal 50% hat und der Restanteil mind. aus 50% erneuerbaren Energiequellen stammt und dieser vom TÜV als Ökostrom zertifiziert ist.

Ein Normalstromtarif mit Aufschlag Förderprogramm (Ökoaufschlag) oder Förderprogramm(Ökoaufschlag)-energreen ist kein Ökostromtarif, weil der Strom aus Normalstrom mit einem normalen Strommix besteht und die Aufschläge „nur“ zur Förderung von regenerativen Anlagen genutzt werden. Wenn auf einen Ökostromtarif (im oberen Sinne) ein Förderprogramm (Ökoaufschlag) oder Förderprogramm(Ökoaufschlag)-energreen gewährt wird, dann sind alle entstehenden Tarifvarianten auch Ökostromtarife, weil die zu Grunde gelegte Tarifart bereits Ökostrom ist.

6.9.2 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)

6.9.3 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

6.9.4 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

6.9.5 Smart-Meter-Tarif

Ein Tarif ist ein Smart Meter Tarif, wenn dieser mit einem digitalen Zähler gemessen wird und mind. 2 Zeitzoneinstaffeln hat.

6.9.6 Energiepreistarif

Energiepreistarife beinhalten keine Kosten für Netznutzung, Abgaben und Umlagen. Der Energiepreis setzt sich zusammen aus Energiearbeitspreis und Energiegrundpreis.

Um den Netto-Endverbrauchpreis 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.

6.9.7 Elektromobilität

Mit der stetigen Zunahme der Anzahl der privaten E-Autofahrer und dem Wissen, dass gut 80 % davon ihr E-Auto zu Hause - meist mit einem eigenen Stromzähler - aufladen, entsteht die Möglichkeit eigene Tarife für diese neuen Verbraucher / Strommengen zu kalkulieren, vergleichen und zu vertreiben. Die Tarifdaten Strom werden mit diesem Produkt um die neuen Preise der privaten Elektromobilitätstarife für zu Hause ladende E-Autofahrer erweitert. Es werden sowohl die Preise als auch die zugehörigen Gebiete der Tarife angeboten, so dass die Daten für Kalkulationen, Marktanalysen oder den Vertrieb der eigenen Tarifprodukte genutzt werden können.

6.9.8 Börsenpreisabhängige Stromtarife

Im Zuge der Neuerungen des EnWG ist es nun auch möglich, über den Stromtarif-Webservice nach börsenpreisabhängigen Stromtarifen zu filtern bzw. diese explizit aus Ergebnismengen auszuschließen.

Die GET AG unterscheidet zwischen den folgenden börsenpreisabhängigen Tarifarten im Bereich Strom, welche jeweils durch spezifische BonusAufschlagsBedingungen (BAIs) gekennzeichnet sind und auch über diese gefiltert werden können (siehe Parameter BAIextendedWhiteList bzw. BAIextendedBlackList im Abschnitt 5.4). Die Zugehörigkeiten in diesen Gruppen schließen sich gegenseitig aus - ein Tarif kann also immer nur einer Gruppe angehören und somit maximal eine der beiden BAIs tragen:

- Dynamische Stromtarife (BAI=208)
- Flextarife (BAI=209)

Als "dynamische Stromtarife" gelten bei der GET AG solche Tarife, welche, unmittelbar nach Vertragsbeginn, den tatsächlich angefallenen Verbrauch zu den tatsächlichen Verbrauchszeiten mit den zugehörigen Börsenpreisen verrechnen. Die Abrechnung muss gemäß EnWG mindestens in derselben Granularität erfolgen, wie durch den Börsenpreisverlauf vorgegeben. Es gibt hier also keinerlei Übergangszeit zum Vertragsbeginn, in der abweichende Regelungen gelten können, und in der Regel ist das Vorhandensein eines iMSys / Smartmeters oder eines vergleichbaren Gerätes Grundvoraussetzung, um einen solchen dynamischen Stromtarif abzuschließen. Um dynamische Tarife mit der BAI 208 anzeigen und berechnen zu können, ist ein Erwerb der Datenlizenz „dynamische Stromtarife“ nötig.

„Flextarife“ hingegen orientieren sich zwar am Börsenpreisverlauf, allerdings wird hier nicht der tatsächliche Verbrauch eines Kunden mit den Börsenpreisen verrechnet. Stattdessen greifen Anbieter von Flextarifen auf Verallgemeinerungsregeln zurück, wie bspw. auf eine Durchschnittsberechnung der Börsenpreise des vorherigen Monats (teils auch mit einer Gewichtung nach

einem Standardlastprofil), ohne jedoch direkt zu Vertragsbeginn den tatsächlichen, zeitlichen Verlauf des Verbrauchs eines Kunden mit in die Berechnung der Tarfkosten einzubeziehen. Nach einer Erstlaufzeit findet dann ein Wechsel in einen dynamischen Tarif statt. Flextarife können damit zunächst unabhängig von verbauten Stromzählern angeboten werden, und werden von Versorgern gern genutzt, um etwaige Übergangszeiten, welche durch den Einbau oder Nachweis eines intelligenten Messsystems entstehen, zu überbrücken. Aufgrund der zu Vertragsbeginn geltenden Tarifbedingungen sind diese Tarife auch für die Darstellung auf Vergleichsportalen geeignet.

6.10 Optionale Parameter/Objekte

Optionale Parameter sind mit der Bemerkung „*(optional)*“ und optionale Objekte mit „*(0:n Beziehung)*“ gekennzeichnet.

Dabei richtet sich die Kennzeichnung bei den Requestparametern nach der fachlichen Anforderung. Die Kennzeichnung der Responseparameter und -objekte als optional richtet sich allein nach der technischen Spezifikation.

So ist z.B.:

- die PLZ als Request technisch optional, aber fachlich zwingend notwendig, also nicht als optional gekennzeichnet
- der Response Gesamtpreis als optional gekennzeichnet, da technisch möglich, inhaltlich wird dieser Parameter aber stets einen Wert enthalten

6.11 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.

6.12 Fehlersuche

Nachfolgend werden die einzelnen fehlerCodes mit der entsprechenden Fehlerbegründung aufgelistet:

antwortStatus 1

OK, Berechnung erfolgt

antwortStatus 2

OK, aber mit Warnung abgeschlossen

antwortStatus 3

mit Fehler abgeschlossen, Fehlercode wird ausgegeben:

fehlerCode 0

OK, Berechnung erfolgt.

fehlerCode 500

Der Nutzer konnte nicht authentifiziert werden.

fehlerCode 501

Nutzer konnte authentifiziert werden, hat aber nicht die erforderlichen Rechte.

fehlerCode 600

Nutzer wurde authentifiziert, hat die benötigten Rechte, aber das Limit der Aufrufe ist überschritten.

fehlerCode 700

Interner Datenfehler.

fehlerCode 701

Parameterobjekt ist null

fehlerCode 703

Die Parameterobjekte muessen zur Zuordnung der Ergebnisse im Rueckgabeobjekt eindeutige UUIDs besitzen.

fehlerCode 704

Bei Uebergabe eines tarifcodes muss auch eine versorgerID angegeben werden.

fehlerCode 705

Ohne Uebergabe einer Postleitzahl wird eine VersorgerID erwartet.

fehlerCode 706

Bei Uebergabe eines Ortes muss auch eine PLZ angegeben werden.

fehlerCode 1001

Allgemeiner Fehler bei Entgeltberechnung

fehlerCode 1002

Allgemeiner Fehler bei TarifEntgeltMapper

fehlerCode 1101

Zu den uebergebenen Daten wurde kein passendes Ergebnis berechnet.

fehlerCode 1102

Es konnte kein Ergebnis zugeordnet werden. In der Anfrage müssen Ort/GV/NB spezifiziert werden.

fehlerCode 1201

Die Marke kann dem Versorger nicht zugeordnet werden.

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

- Webservice-Bezeichnung und –Versionsnummer
- FehlerCode
- iTag
- Request

6.13 ortsscharfe bzw. bis zu hausnummernscharfe Tarifzuordnung

Im Zusammenspiel mit den Postdaten bzw. dem Ortsinformationswebservice 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

6.13.1 Nutzung der Ortsschärfe

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

=> fertig

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

6.13.2 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, Hausnummer 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

6.14 Verwendung der Bulkberechnung

Die Methoden `getBerechneteGrundversorgungsTarife` 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.

6.15 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
<code>IntObj</code>	<code>object</code>	siehe Objekt IntIdObj
<code>DoubleObj</code>	<code>object</code>	siehe Objekt DoubleIdObj
<code>StringObj</code>	<code>object</code>	siehe Objekt StringIdObj
<code>DateObj</code>	<code>object</code>	siehe Objekt DateIdObj

6.15.1 IntIdObj

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

6.15.2 DoubleIdObj

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

6.15.3 StringIdObj

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

6.15.4 DateIdObj

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

6.15.5 BasisIdObj

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

6.16 Rundungen über dynamischeAngaben

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><bas:IntObj id="20" name="BERECHNUNGSKONFIGURATION" wert="0"></bas:IntObj></code>	Rundung für Rohdaten und berechnete Wert ist deaktiviert . Es werden alle Nachkommastellen ausgegeben
<code><bas:IntObj id="20" name="BERECHNUNGSKONFIGURATION" wert="1"></bas:IntObj></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><bas:IntObj id="20" name="BERECHNUNGSKONFIGURATION" wert="2"></bas:IntObj></code>	Rohdaten werden mit 2 Nachkommastellen und berechnete Werte mit 2 Nachkommastellen gerundet ausgegeben. Fehlende Nachkommastellen werden mit 0en aufgefüllt.
<code><bas:IntObj id="20" name="BERECHNUNGSKONFIGURATION" wert="3"></bas:IntObj></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:

...

`<par:dynamischeAngaben>`

`<bas:IntObj id="20" name="BERECHNUNGSKONFIGURATION"
wert="0"></bas:IntObj>`

`</par:dynamischeAngaben>`

...

6.17 Börsenpreiskonfiguration über dynamische Angaben

Request - Auszug	Art der Rundung
<code><bas:IntObj name="boersenPreisConfig" wert="1" /></code>	1 = der Beschaffungspreis entspricht dem Durchschnittswert der Börsenpreise der letzten 365 Tage (vor dem eingegebenen Stichtag) welche nach dem dynamischen BDEW H0 Standardlastprofil gewichtet wurden
<code><bas:IntObj name="boersenPreisConfig" wert="2" /></code>	2 = der Beschaffungswert im Tarifiergebnisobjekt wird per se auf den Wert 0 gesetzt, es werden also nur sämtliche Preisbestandteile mit Ausnahme des Börsenpreises ausgegeben

6.18 Änderungshistorie

6.18.1 V1.3

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

6.18.2 V1.4

- neue Methoden: [getVersorger](#), [getVersorgertarifvarianten](#), [getParameter](#)
- Erweiterung des Requests der Methode [getBerechneteTarife](#), z.B.
 - Filter für Versorger, Smartmeter, Preisgarantiedauer
 - Paging
- Erweiterung des Responses von [getBerechneteTarife](#), z.B.
 - Öko-Zertifikate
 - StrommixLight
- Erweiterung des Responses der [Tarifkarte](#), z.B.
 - Preise in Brutto und Netto
 - Zählerkosten
- 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

6.18.3 V1.5

- neue Methode: [getBundesstrommix](#)
- Erweiterung des Requests der Methode [getBerechneteTarife](#), z.B.
 - Netzbetreiber
 - Tariftyp
 - Handling
 - Portalrichtlinien
 - Kündigungsfrist
 - Zertifikate
- Erweiterung des Responses von [getBerechneteTarife](#), z.B.
 - Fehlermeldung
- Erweiterung des Responses der Methode [getVersion](#), z.B.
 - Datenbankstand

- Erweiterung des Requests der Methoden [getBerechneteTarifebyPLZundVerbrauchKundenart](#) und [getBerechneteGrundversorgungsTarifebyPLZundVerbrauchKundenart](#), z.B.
 - Stichtag
 - VerbrauchNT
- Erweiterung des Requests der Methode [getHandlingsDetails](#)
 - Ort
 - Netzbetreiber
 - Grundversorger
- Optimierung der Doku

6.18.4 V2.0

- Nutzung der Middleware JAX anstatt AXIS
- Änderung des Filters „portalRichtlinie“
- Umbenennen der Methode „getBerechneteGrundversorgungsTarifebyPLZundVerbrauchKundenart“ in „[getBerechneteGrundversorgungsTarife](#)“
- Erweiterung des Requests der Methode [getBerechneteGrundversorgungsTarife](#):
 - tariftyp
 - oeko
 - zaehlerart
 - uid
 - grenzenBeiATignorierenBis
 - netzbetreiber
 - grundversorger
 - versorgerIDs
- Erweiterung des Response der Methode [getBerechneteGrundversorgungsTarife](#):
 - uid

- Möglichkeit mittels [getBerechneteGrundversorgungTarife](#) die Grundversorgungstarife bundesweit zu berechnen
- Entfernen der Methode „getBerechneteTarifebyPLZundVerbrauchKundenart“
- Erweiterung des Request der Methode [getVersorgerTarifvarianten](#) um
 - Ort
 - Netzbetreiber
 - Grundversorger
 - PLZ nun optional, um lokationsunabhängig alle Tarife eines Versorgers zu erhalten
 - tariftyp
- Erweiterung des Response der Methode [getVersorgerTarifvarianten](#) um isGrundversorgungTarif
- Erweiterung des Requests der Methode [getBerechneteTarife](#)
 - markenIDs
 - tarifCode
 - tarifRootIDsMindestfilter
 - zaehlerart
 - 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 Response der Methode [getBundesstrommix](#) um
 - sonstige
- 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
 - sonstige erneuerbare Energie
- Erweiterung des Requests und Response aller Methoden um das Objekt [dynamischeAngaben](#)
- [getBerechneteGrundversorgungsTarife](#) und [getBerechneteTarife](#) ermöglichen nun eine Bulk-Berechnung (siehe uid)
- Umstellung auf [Basis Authentifizierung](#)
- Erweiterung um die Methode [getAbgaben](#)
- Änderung der Definition „recherchetyp“ und Hinweis auf die hierfür zu verwendenden IDs aus Param_Tarif_Zusatzinformation in Methode [getParameter](#)
- Ausgabe von Energiepreistarifen sofern das Recht „Netzentgelte“ vorhanden
- Erweiterung von Request und Response der Methode [getBerechneteTarife](#) um dynamische Stromtarife ausgeben zu können:
 - details (um Ausgabe der Netzentgeltberechnung und/oder der Börsenpreise/Beschaffungspreise auszugeben)
 - Erweiterung dynamische Angaben um boersenpreisConfig (um Format der Ausgabe dynamischer Stromtarife zu steuern)