

GET AG Webservices

Technische Dokumentation

Version 1.0.6

Leipzig, 15.07.2024

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

1	Dokumentenhistorie	3
2	Beschreibung	4
3	Erreichbarkeit.....	4
4	Implementierung	5
4.1	"GET AG SOAP & JSON RPC Services" vs. "nativer" RESTful Services	5
4.2	SSL	6
4.3	Freischaltung	7
4.3.1	Feste IP-Adresse	7
4.3.2	Serverlose Datenübertragung	7
4.4	Authentifizierung am Webservice.....	7
4.5	Anfrageverhalten.....	8

1 Dokumentenhistorie

Dokumenten-Version	Datum	Änderung von	Bemerkung
1.0.0	19.02.2020	Patrick Schmorde	Erstellung
1.0.1	26.05.2020	Peter Groß	Aktualisierung Zertifikatinformation
1.0.2	04.06.2020	Peter Groß	Erweiterung Punkt 4. Implementierung um Komprimierung
1.0.3	21.09.2021	Peter Groß	Erweiterung um „GET AG SOAP & JSON RPC Services" vs. "nativer" RESTful Services"
1.0.4	06.07.2022	Peter Groß	Aktualisierung Zertifikatinformation
1.0.5	11.07.2023	Peter Groß	Aktualisierung Zertifikatinformation
1.0.6	15.07.2024	Peter Groß	Aktualisierung Zertifikatinformation

2 Beschreibung

Die GET AG stellt Webservices für unterschiedliche Anwendungen bereit. Diese Dokumentation dient hauptsächlich dazu, die technischen Rahmenbedingungen bei der Einbindung eines Webservices zu erläutern.

Die fachliche Beschreibung der Webservices, inkl. Angabe der verfügbaren WSDL und WADL, erfolgt jeweils in einer spezifischen zum Webservice erstellten Dokumentation.

3 Erreichbarkeit

Die GET AG stellt jeden Service als Produktivservice und als Testservice zur Verfügung.

Der Testservice steht den Implementierungspartnern zur Entwicklungsunterstützung ihrer Anwendungen bereit. Er enthält nicht die aktuellen Daten oder nur einen Teil der Daten und kann aufgrund von Wartungsarbeiten kurzzeitig nicht verfügbar sein.

Die permanente Erreichbarkeit der produktiven Webservices ist durch einen Loadbalancer gegeben. Dabei findet die Datenaktualisierung auf den Servern sequenziell zwischen 3 und 5 Uhr statt.

Die Testservices werden automatisch auf Erreichbarkeit überwacht. Im Gegensatz zu unseren Produktivservices, welche einer 24/7 Überwachung unterliegen, findet die Überwachung der Testservices nur innerhalb unserer Geschäftszeiten (Mo.-Fr. von 9 – 16 Uhr) statt. Für Donnerstags von 10 – 11 Uhr sehen wir planmäßige Wartungsarbeiten an den Testservices vor. Wir bitten dies bei Ihren Planungen zu beachten.

4 Implementierung

Bei der Implementierung sollte der Soap12Endpoint verwendet und die clientseitige Komprimierung genutzt werden. Mit der clientseitigen Komprimierung lässt sich die Größe der zu übertragenden Daten um bis zu 95% reduzieren, was sich positiv auf die Geschwindigkeit des Service / Ihrer Anfragen auswirkt.

Der Webservice unterstützt zudem das Ausliefern von komprimierten HTTP-Antworten. Im konkreten Fall wird eine gzip Komprimierung auf die HTTP-Antwort angewendet, wenn der anfragende Client dem Webservice signalisiert, dass dieser auch gzip-Antworten verstehen und handhaben kann. Diese Signalisierung erfolgt durch das Setzen des HTTP-Headers "Accept-Encoding: gzip" bei jeder HTTP-Anfrage. Wenn der Client also "Accept-Encoding: gzip" im HTTP-Header setzt, wird die HTTP-Antwort serverseitig komprimiert ausgeliefert. Dies reduziert den gröbenseitigen Umfang bzw. das Volumen der Antwort deutlich (zu Lasten der Rechenleistung, da gepackt und entpackt werden muss).

Ein permanentes Nachladen der WSDL oder xsd (Schemaabfragen) sind vom Nutzer unbedingt zu vermeiden, da dies zu Beeinträchtigungen der Services führt. Alternativ sollte dafür der Cache beschrieben werden.

4.1 "GET AG SOAP & JSON RPC Services" vs. "nativer" RESTful Services

Neben den SOAP Services bietet die GET AG auch JSON RPC Services an. Die Struktur der Schnittstelle wird entweder als WADL (ältere Services) oder im OpenApi-Format (neuere Services) geliefert. Mittels diverser Werkzeuge kann ein Client erzeugt oder der Service in eine vorhandene Anwendung integriert werden.

Weiterhin gibt es die Möglichkeit die neueren Services im Swagger-UI zu nutzen. Die JSON RPC Services werden genutzt um die teilweise komplexen / verschachtelten Strukturen der (Bulk-)Berechnungen und Massendatenexporte per Webservice zu ermöglichen. Dabei wird in diesen Fällen per POST die

komplexe Struktur des Requests in den Body gepackt, um teilweise komplexere Anfragestrukturen abzubilden.

(Alle GET AG Services sind aktuell ausschließlich für das Auslesen, Berechnen und Exportieren von größeren Datenmengen konzipiert!)

4.2 SSL

Der produktive Service ist auch über SSL verfügbar. Hier wird das folgende Zertifikat benutzt:

Ausgestellt für

Allgemeiner Name (CN)	webservice1.ag-server.de
Organisation (O)	<kein Teil des Zertifikats>
Organisationseinheit (OU)	Domain Control Validated
Seriennummer	00:E5:C2:C8:58:B1:26:07:F7:D9:52:63:27:41:98:F2:32

Ausgestellt von

Allgemeiner Name (CN)	Sectigo RSA Domain Validation Secure Server CA
Organisation (O)	Sectigo Limited
Organisationseinheit (OU)	<kein Teil des Zertifikats>

Gültigkeitsdauer

Beginn mit	Freitag, 05. Juli 2024
Läuft ab am	Mittwoch, 06. August 2025

Fingerabdrücke

SHA-256-Fingerabdruck	16:6E:AB:49:8D:51:56:10:9C:06:6A:0E:DB:11:09:06:31:E6:E1:4A:E1:2B:4C:40:CE:8D:A8:75:15:A4:B9:92
SHA1-Fingerabdruck	1F:11:0E:2B:94:6D:CE:61:CC:0F:61:19:83:2B:6D:46:03:C9:1E:1C

Die Zertifikate werden regelmäßig aktualisiert. Aktuell findet diese Aktualisierung im Einjahresrhythmus zwischen Juli und Juli statt. Wir informieren Sie aktiv, sobald neue Zertifikate verfügbar sind.

4.3 Freischaltung

4.3.1 Feste IP-Adresse

Der produktive Service muss für die IP-Adresse, über die der Client auf den Service zugreift, freigeschaltet werden. Wenden Sie sich an Ihren Ansprechpartner bei der GET AG und teilen Sie ihm die IP-Adresse oder den IP-Bereich, von der/dem aus Sie zugreifen wollen, mit, um für diesen Service freigeschaltet zu werden. Sie erhalten dann Ihre Authentifizierungsinformationen für den Service.

4.3.2 Serverlose Datenübertragung

Falls Sie die serverlose Datenübertragung / serverless computing (z.B. AWS) nutzen, nehmen Sie bitte folgende clientseitigen Hinweise zur Kenntnis:

<https://medium.com/@matthewleak/aws-lambda-functions-with-a-static-ip-89a3ada0b471>

4.4 Authentifizierung am Webservice

Für alle Aktionen, bei denen Sie auf den Webservice zugreifen, außer bei der Methode getVersion, benötigt der Service Ihre Authentifizierungsinformationen.

Für die Authentifizierung wird das HTTP-Authentifizierungs-Verfahren Basic benutzt.

Die sog. Basisauthentifizierung ist die häufigste Art der HTTP-Authentifizierung. Dazu werden im HTTP-Header „Authorization“ die Authentifizierungsart und weiter der Nutzernamen und das Passwort base64 kodiert übertragen.

Siehe auch <https://www.ietf.org/rfc/rfc2617.txt>

Bsp.:

HTTP Header

Authorization: Basic QmVudXR6ZXJuYW1lOIBhc3N3b3J0

QmVudXR6ZXJuYW1lOIBhc3N3b3J0 ist die resultierende base64-Kodierung für
Benutzername:Passwort
Beide werden mit „:“ verkettet.

4.5 Anfrageverhalten

Unsere Webservices sind eine Clusterplattform (Multi-Server mit Loadbalancer). Der Loadbalancer entscheidet selbständig, auf welcher Hardware welche Abfrage am effektivsten beantwortet werden kann. Diesbezüglich ist bei der Entwicklung Ihrer Clientabfragen folgendes zu beachten.

Es muss vermieden werden, dass alle bzw. mehrere Abfragen in nur einer Connection zur Webserviceplattform gesendet werden.

Es soll eine Connection so schnell wie möglich wieder geschlossen werden. Unsere Webservices arbeiten „stateless“, auch logisch voneinander abhängige Abfragen können in unterschiedlichen Connections gesendet und somit auch von unterschiedlichen Servern beantwortet werden.

Ein schnelles Schließen der Connection fördert die parallele Abarbeitung der Anfragen, da der Loadbalancer die Abarbeitung Ihrer Anfragen auf einem anderen, möglicherweise weniger ausgelastetem Server fortsetzen kann.

Ein schnelles Schließen der Connection vermeidet zudem Exceptions in Ihrer Anwendung durch von uns durchgeführtes Connection Reset wegen Wartung oder Datenaktualisierung. Für Wartung oder Datenaktualisierungen werden die entsprechenden Server vorher beim Loadbalancer abgemeldet, so dass diese nicht mehr mit neuen Connections belegt werden.