



SACHSEN-ANHALT

Ministerium für
Inneres und Sport



Technische Richtlinie Safe-Server

Länderübergreifendes Glücksspielaufsichtssystem
(LUGAS)

Herausgeber: Dataport AöR
im Auftrag des
Ministeriums für Inneres und Sport des
Landes Sachsen-Anhalt

Version: 1.5.1
21.06.2022

Inhaltsverzeichnis

1	Änderungsverzeichnis	5
2	Einleitung	8
3	Geltungsbereich	9
4	Begriffsdefinitionen	9
4.1	Domainbegriffe	9
4.2	Technische Begriffe	10
5	Spezifikation des Safe-Servers	11
5.1	Belange der Informationssicherheit	11
5.1.1	Integrität der Daten	11
5.1.2	Authentizität & Nichtabstreitbarkeit der Daten	11
5.1.3	Verfahren zur Signierung und Übertragung der Daten	12
5.1.4	Resilienz	12
5.2	Event-Schnittstelle	12
5.2.1	Aufbau	12
5.2.2	Chunks	13
5.2.3	Signatur der Chunks	13
5.2.4	Komprimierung der Chunks	13
5.2.5	Abholen der Events	13
5.2.6	Feedback	17
6	Betriebsphasen	18
6.1	Einrichtung	18
6.2	Normalbetrieb	19
7	Zeichensatz und Datentypen	19
8	Ereignisse	21
8.1	Personendaten Spieler	23
8.1.1	Registrieren	23
8.1.2	Datenänderung	26
8.1.3	Datenbestätigung	28
8.1.4	Bestätigung vorläufiges Spiel	29
8.1.5	Status	30

8.2	Spielersperrung	33
8.2.1	Sperre aktivieren	33
8.2.2	Sperre aufheben	35
8.3	Anbieterbezogene Spielerlimits	36
8.3.1	Limit setzen	36
8.3.2	Limit löschen	39
8.4	Matches	40
8.4.1	Start Einzelspieler-Match	40
8.4.2	Ende Einzelspieler-Match	43
8.4.3	Start Gemeinschaftsspieler-Match Spieler	45
8.4.4	Ende Gemeinschaftsspieler-Match Spieler	47
8.4.5	Ende Gemeinschaftsspieler-Spiel	50
8.5	Zahlungen	51
8.5.1	Zahlungsweg hinzufügen	51
8.5.2	Zahlungsweg entfernen	53
8.5.3	Einzahlung	54
8.5.4	Auszahlung	55
8.5.5	Transaktion Spielkonto	56
8.6	Interaktion zwischen Spieler und Anbieter	58
8.6.1	Log-in	58
8.6.2	Spielererklärung virtuelles Automatenspiel	59
8.6.3	Spieler-Timeout	60
8.6.4	Realitycheck	61
8.6.5	Realitycheck Bestätigung	63
8.6.6	Erinnerung Bereichswechsel	64
8.6.7	Korrekturaufforderung Limitfestsetzung	65
8.6.8	Information statistische Daten	66
8.6.9	Veranlassung Inaktivierung	67
8.7	Nutzung der Zentraldateien	67
8.7.1	Übermittlung Spielerdaten	67
8.7.2	Anbieterübergreifendes Einzahlungslimit	69
8.7.3	Einzahlung	71
8.7.4	Aktivstatus	73
8.7.5	Sperrprüfung	74
8.8	Anbieterdaten	76
8.8.1	Bereich hinzufügen	76
8.8.2	Einzelspieler-Spiel hinzufügen	77
8.8.3	Genehmigung Einzelspieler-Spiel	78

8.8.4	Gemeinschaftsspieler-Spiel hinzufügen	80
8.8.5	Genehmigung Gemeinschaftsspieler-Spiel	82
8.8.6	Genehmigung zur Festsetzung eines abweichenden Limits	84
8.8.7	Zahlungsdienst hinzufügen	86
8.8.8	Zahlungsdienst entfernen	87
8.9	Sonderereignisse Einrichtungsphase	88
8.9.1	Kontostand Spieler	88
8.9.2	Ende Einrichtungsphase	89
9	Beispiele	89
9.1	Initiale Anbindung Safe-Server	90
9.2	Spieler erstellt ein Konto	90
9.3	Durchschnittlicher Spielzyklus	91
9.4	Ablauf der jährlichen Spielerdaten-Überprüfung durch den Anbieter	92
10	Anlagen	92

1 Änderungsverzeichnis

Version 1.5.1 vom 21.06.2022

- Ergänzungen und Klarstellungen in [Safe-Server Spezifikation](#).
 - Zusammenfassung der Zertifikate.
 - Hinweis zur EventId
- Ergänzungen in [Ereignisse](#)
 - Abschlusszeit des letzten Chunks eines Tages.

Version 1.5 vom 25.03.2022

- Neuausrichtung Kapitel [Safe-Server Spezifikation](#).

Version 1.4 vom 14.01.2022

- Anpassungen an [Zeichensatz und Datentypen](#)
 - Ausschließliche Verwendung von UTF-8 als Kodierung der JSON-Dateien.
- Anpassungen an [Ereignisse](#)
 - Änderung der Verzeichnisstruktur zu YYYY/MM/DD.
 - Anpassung der Einschränkungen zur Dateiablage.
 - Anpassung des Sortierschlüssels.
- Anpassungen an [Registrieren](#)
 - Hinweis zum Umgang mit Nachmeldung von Spielern.
 - Hinweis zum Umgang mit Spieler Status in Einrichtungsphase.
- Anpassungen an [Datenänderung](#)
 - Hinweis zur zusätzlichen Datenänderung in den Zentraldateien hinzugefügt.
- Anpassungen an [Datenbestätigung](#)
 - Hinweis zum Spieler Status nach Datenbestätigung.
- Anpassungen an [Status](#)
 - Hinweise auf Umgang mit jährlicher Datenüberprüfung hinzugefügt.
 - “intent” ist nun ein bedingtes Pflichtfeld.
- Anpassungen an [Start Einzelspieler-Match](#)

- “isFreeOfCharge” zu “charge” mit unterschiedlichen Werten geändert. Hinweis zum Umgang mit Bonusgeld eingefügt.
- Hinweis zur Vorlage einer Genehmigung eingefügt.
- Anpassungen an [Start Gemeinschaftsspieler-Match Spieler](#)
 - “isFreeOfCharge” zu “charge” mit unterschiedlichen Werten geändert. Hinweis zum Umgang mit Bonusgeld eingefügt.
 - Hinweis zur Vorlage einer Genehmigung eingefügt.
- Anpassungen an [Genehmigung Einzelspieler-Spiel](#)
 - Hinweis zur Vorlage einer Genehmigung eingefügt.
 - Einleitungstext: “mehr als 5 Sekunden” in “mindestens 5 Sekunden” geändert.
- Anpassungen an [Genehmigung Gemeinschaftsspieler-Spiel](#)
 - Hinweis zur Vorlage einer Genehmigung eingefügt.
- Anpassungen an [Ende Gemeinschaftsspieler-Match Spieler.](#)
 - Neue Felder “bonusStake” und “bonusRevenue” eingefügt. Hinweis zum Umgang mit Bonusgeld eingefügt.
 - Hinweis zum Umgang mit Transaktionen Spielerkonto ergänzt.
 - Beispiele für Bonusgeld und Echt- und Bonusgeld eingeführt.
- Anpassungen an [Ende Einzelspieler-Match.](#)
 - Neue Felder “bonusStake” und “bonusRevenue” eingefügt. Hinweis zum Umgang mit Bonusgeld eingefügt.
 - Hinweis zum Umgang mit Transaktionen Spielerkonto ergänzt.
 - Beispiele für Bonusgeld und Echt- und Bonusgeld eingeführt.
- Anpassungen an [Einzahlung](#) (Spieler).
 - Hinweis zum Zusammenspiel von Abgleich Zentraldateien, Transaktion Spielerkonto und Einzahlungen ergänzt.
 - Hinweis, das Einzahlungen von einem Bonusgeld-Konto nicht mit den Zentraldateien abgeglichen werden müssen ergänzt.
- Anpassungen an [Einzahlung](#) (Zentraldateien).
 - Hinweis zum Zusammenspiel von Abgleich Zentraldateien, Transaktion Spielerkonto und Einzahlungen ergänzt.
 - Hinweis das Einzahlungen von einem Bonusgeld-Konto nicht mit den Zentraldateien abgeglichen werden müssen ergänzt.

- Anpassungen an [Auszahlung](#).
 - Hinweis zum Umgang mit Transaktion Spielerkonto ergänzt
- Anpassungen an [Zahlungsweg hinzufügen](#)
 - Feld *methodName* hinzugefügt, *otherData* entfernt, *idData* umfunktioniert.
 - Beispiele entsprechend angepasst.
- Ergänzungen in [Beispiele](#)
 - Beispiel zur jährlichen Datenüberprüfung hinzugefügt.

Version 1.3 vom 30.09.2021

- Anpassungen an [Start Einzelspieler-Match](#)
- Anpassungen an [Sperrprüfung](#)
- Anpassungen an [Durchschnittlicher Spielzyklus](#)
- Anpassung Überschrift und Verweis [Übermittlung Spielerdaten](#)

Version 1.2 vom 03.08.2021

- Anpassung an [Beispielen](#)

Version 1.1 vom 24.06.2021

- Anpassungen an der [Einleitung](#)
- Anpassungen an der [Spezifikation des Safe-Servers](#)
- Anpassungen an Bedingungen und Einschränkungen des Dateiformats von [Ereignissen](#)
- Ergänzende Hinweise [Betriebsphasen](#)
- Anpassungen und Spezifizierungen in Ereignissen
 - [Auszahlung](#)
 - [Einzahlung](#)
 - [Genehmigung Gemeinschaftsspieler-Spiel](#)
 - [Start Einzelspieler-Match](#)
 - [Start Gemeinschaftsspieler-Match Spieler](#)
 - [Status](#)
 - [Sperrung aufheben](#)
 - [Zahlungsdienst hinzufügen](#)
 - [Zahlungsweg hinzufügen](#)

2 Einleitung

Ab 1. Juli 2021 gilt der Staatsvertrag zur Neuregulierung des Glücksspielwesens in Deutschland vom 29. Oktober 2020 (Glücksspielstaatsvertrag 2021 - GlüStV 2021).

Die Aufsicht über das Glücksspiel im Internet wird zukünftig mit bundesweiter Zuständigkeit zum Großteil durch die Gemeinsame Glücksspielbehörde der Länder mit Sitz in Halle (Saale) wahrgenommen. Während einer Übergangszeit bis zum 31. Dezember 2022 nehmen gemäß § 27p GlüStV 2021 einzelne Länder glücksspielrechtliche Aufgaben mit bundesweiter Zuständigkeit nach dem Glücksspielstaatsvertrag 2021 wahr. Vor diesem Hintergrund wird die zuständige Glücksspielaufsichtsbehörde des Landes Sachsen-Anhalt, das Landesverwaltungsamt, ab dem 1. Juli 2021 insbesondere die Führung der nach dem Glücksspielstaatsvertrag für die Überwachung des Glücksspielmarktes im Internet vorgesehenen informationstechnischen Systeme wahrnehmen.

Der Anschluss an diese Systeme ist für erlaubte Anbieter von Glücksspielen im Internet nach dem Glücksspielstaatsvertrag 2021 verpflichtend.

Das hierfür geschaffene **Länderübergreifende Glücksspielaufsichtssystem (LUGAS)** setzt sich aus den Zentraldateien, bestehend aus

- der Datei zur Einzahlungslimitfestsetzung (Limitdatei),
- der Datei zur Verhinderung parallelen Spiels (Aktivitätendatei)

und dem Auswertesystem zusammen.

Die zuständige Behörde bedient sich des in § 6i Abs. 2 GlüStV 2021 geregelten Auswertesystems als Werkzeug zur Aufsicht über das Glücksspiel im Internet. Es stellt sicher, dass die von den Glücksspielanbietern (im Folgenden kurz Anbieter genannt) selbst erfassten Daten ausgewertet werden und hinsichtlich der Einhaltung des GlüStV geprüft werden.

Die Anbieter sind dazu verpflichtet, alle erforderlichen Daten zutreffend zu erfassen.

Die Erfassung und Bereitstellung der Daten erfolgt selbständig durch die Anbieter auf sogenannten Safe-Servern, die von den Anbietern auf eigene Kosten einzurichten und zu betreiben sind. Die Daten sind auf den Safe-Server digital unveränderlich abzulegen. Das bedeutet, der Anbieter hat eigenverantwortlich sicherzustellen, dass ohne ausdrückliche Aufforderung der Behörde keinerlei Zugriffe auf die automatisch erstellten Dateien möglich sind und Daten nicht aufgrund technischer Defekte verloren gehen oder beschädigt werden. Der Anbieter muss eine jederzeitige Kontrolle einschließlich unmittelbarem Zugriff durch die zuständige Aufsichtsbehörde ermöglichen. Das Auswertesystem greift kontinuierlich auf die Safe-Server zu und holt die bereitgestellten Daten zur Verarbeitung ab.

Das vorliegende Dokument definiert die erforderlichen Daten zur Interaktion mit den Systemen und beschreibt, wie diese von den Anbietern auf den Safe-Servern abzulegen sind.

3 Geltungsbereich

Diese Richtlinie ist für alle Anbieter verpflichtend umzusetzen, welche online Glücksspiele nach Erlaubnis von § 4 GlüStV 2021 im Internet anbieten. Mit dem Erlaubnisantrag stimmt der Anbieter dem aktuellen Umfang dieser Richtlinie zu.

Diese Richtlinie ist mit der Version 1.1 vom 24.06.2021 auf dem Stand, welcher zum 01.07.2021 wirksam wird. Sie unterliegt einer ständigen Überarbeitung und wird nach Wirksamwerden turnusmäßig aktualisiert. Nähere Informationen über den konkreten Modus der Aktualisierung werden durch die zuständige Glücksspielaufsichtsbehörde bekannt gegeben.

4 Begriffsdefinitionen

4.1 Domainbegriffe

Begriff	Definition
Anbieter	Ein Anbieter ist ein Veranstalter von Sportwetten, Online-Casinospielen, Online-Poker oder virtuellen Automaten Spielen im Internet, welcher eine Erlaubnis nach § 4 GlüStV 2021 erhalten hat.
Behörde	Dieser Begriff ist eine Kurzform für die "Gemeinsame Glücksspielbehörde der Länder", welche nach § 27a GlüStV 2021 zum Zweck der Wahrnehmung der Aufgaben der Glücksspielaufsicht errichtet ist.
Match	Ein Match ist eine Partie eines Spiels, bspw. eine Runde an einer virtuellen Slotmaschine.
Spiel	Ein Spiel ist die Ausprägung einer Spielart, bspw. eine konkrete Slotmaschine.
Spielart	Im aktuellen Stand der Richtlinie werden zwei Spielarten abgebildet: Einzelspieler-Spiel und Gemeinschaftsspieler-Spiel.
Spieler	Ein Spieler ist eine natürliche Person, welche das Angebot eines Anbieters nutzt.

Begriff	Definition
Zahlungsdienst	Definiert die Art und Weise einer Zahlung und die Art und Weise, mit der ein Empfänger oder Sender eine Zahlung identifiziert wird.
Zahlungsweg	Alle notwendigen Informationen, um im Rahmen eines Zahlungsdienstes den Empfänger oder Sender einer Zahlung identifizieren zu können.

4.2 Technische Begriffe

Begriff	Definition
Byte	Eine Folge von acht Ja- und Nein-Informationen zur Repräsentation von 256 unterschiedlichen Werten.
URI	Der U niform R esource I dentifier ist ein Identifikator für Ressourcen im Internet.
HTTPS	Das H ypertext T ransfer P rotocol S ecure ist ein verschlüsseltes Kommunikationsprotokoll für das Internet.
API	Ein A plication P rogramming I nterface ist eine Schnittstelle von Computerprogrammen.

5 Spezifikation des Safe-Servers

Jeder Anbieter hält einen öffentlich zugreifbaren Server vor, von dem die anfallenden Daten in Form von **Chunks** abgeholt werden. Der Server muss die im Kapitel **Event-Schnittstelle** beschriebene Schnittstelle zu Verfügung stellen. Es dürfen nur Anfragen von LUGAS akzeptiert werden. Zur Authentifizierung nutzt LUGAS ein Client-Zertifikat. Die benötigten Informationen, werden dem Anbieter im Voraus von der Behörde mitgeteilt.

Die Ablage und Schutz der im Kapitel **Ereignisse** beschriebenen Daten, obliegt der Verantwortung des Anbieters. Der Anbieter muss sich an die, im Kapitel **Belange der Informationssicherheit** beschriebenen Anforderungen halten.

Dabei muss der Anbieter zwei Zertifikate generieren und die Daten mit einen sog. qualifizierten elektronischem Zeitstempel signieren.

Das erste Zertifikat und der qualifizierte elektronische Zeitstempel werden zur Signierung der Daten genutzt. Dabei stellt das Zertifikat die Herkunft der Daten sicher (**Authentizität & Nichtabstreitbarkeit der Daten**). Der qualifizierte elektronische Zeitstempel stellt sicher, dass die Daten ab dem Zeitpunkt des Zeitstempels nicht mehr geändert wurden (**Integrität der Daten**). Das zweite Zertifikat wird zu sicheren Kommunikation auf Basis von mTLS zwischen Safe-Server und LUGAS genutzt (**Resilienz**).

5.1 Belange der Informationssicherheit

5.1.1 Integrität der Daten

Um die Unveränderbarkeit der Daten innerhalb eines Chunks von unabhängiger Stelle sicherzustellen, wird ein sog. qualifizierter elektronischer Zeitstempel genutzt. Der Anbieter ist verpflichtet, die Chunks mit einen qualifizierten elektronischen Zeitstempel von einer anerkannten Zertifizierungsstelle zu signieren. Anerkannte Zertifizierungsstellen können in der EU Trust List¹ gefunden werden.

5.1.2 Authentizität & Nichtabstreitbarkeit der Daten

Um sicherzustellen, dass die Daten von dem Anbieter stammen, wird eine weitere Signatur den Daten eines **Chunks** hinzugefügt. Dafür muss der Anbieter sich ein asymmetrisches Schlüsselpaar erzeugen. Mit dem privaten Schlüssel wird der Chunk signiert. Der Anbieter stellt der Behörde den öffentlichen Schlüssel zur Verfügung, damit diese die Signatur überprüfen kann (siehe **Verfahren zur Signierung und Übertragung der Daten**).

¹<https://esignature.ec.europa.eu/efda/tl-browser/#/screen/home>

5.1.3 Verfahren zur Signierung und Übertragung der Daten

Der zu übertragende Chunk an LUGAS wird signiert. Die geforderten Parameter des kryptografischen Verfahrens zur Signierung und Übertragung der Daten richtet sich nach der TR-02102 des BSI². Dabei wird explizit *DSA ECDSA (NIST 256 oder prime256v1) + SHA-256* eingefordert. Die Laufzeit eines Schlüssels beträgt 3 Jahre. Sollte der Algorithmus oder die Schlüssellänge vorzeitig vom BSI als unsicher eingestuft werden, müssen neue Schlüssel erstellt und ausgetauscht werden. Die Behörde behält sich das Recht vor, die Anforderungen anzupassen, sollten Änderungen nötig werden.

Der öffentliche Teil des Schlüsselpaares muss der Behörde auf sicherem Wege mitgeteilt werden.

5.1.4 Resilienz

Die Kommunikation zwischen LUGAS und den Anbietern erfolgt auf Basis von mTLS (Mutual TLS). Zu diesem Zweck veröffentlicht die Behörde ein entsprechendes Zertifikat. Der Betreiber der Safe-Saver ist angewiesen nur Verbindungen zu akzeptieren, bei dem die Gegenseite sich mit einem solchen gültigen Zertifikat der Behörde ausweisen kann. Um einen reibungslosen Übergang bei neuen Zertifikaten zu ermöglichen, veröffentlicht die Behörde rechtzeitig Nachfolgezertifikate. Neue Zertifikate sind immer unmittelbar gültig, lösen aber dabei nicht den Vorgänger ab. Dies gilt insofern die Behörde nichts anderes kommuniziert.

Neben mTLS muss der Safe-Server sich eindeutig identifizieren lassen. Es ist dabei nicht hinreichend, dass das Zertifikat von einer offiziellen CA unterschrieben ist. Der Behörde muss rechtzeitig das TLS-Zertifikat des Servers zur Verfügung gestellt werden. Nur wenn der Server mit einem der Behörde bekannten Zertifikat antwortet und das Zertifikat entsprechende Eigenschaften aufweist (siehe [Verfahren zur Signierung und Übertragung der Daten](#)) ist die notwendige Bedingung erfüllt.

5.2 Event-Schnittstelle

5.2.1 Aufbau

Die Events werden in sog. Chunks an LUGAS zurückgegeben. Das **eventTime**-Feld aller Events innerhalb eines Chunks muss dasselbe Datum (yyyy-mm-dd) besitzen. Die Uhrzeit kann variieren. Als Zeitzone ist **UTC** zu verwenden. Die Events sind immer in der korrekten, zeitlichen und logischen Reihenfolge zu ordnen.

²<https://www.bsi.bund.de/SharedDocs/Downloads/DE/BSI/Publikationen/TechnischeRichtlinien/TR02102/BSI-TR-02102.pdf>

5.2.2 Chunks

Um die Größe der Antworten zu limitieren werden die Events eines Tages in Chunks unterteilt. Ein Chunk beinhaltet maximal 200.000 Events. Sobald ein Chunk abgeschlossen ist, weil er nicht mehr bearbeitet wird, kann er als Nachfolger abgerufen werden.

5.2.3 Signatur der Chunks

Jeder Chunk muss signiert werden um die Authentizität der Daten sicherzustellen. Dabei kommt die JSON-Web-Signature (JWS), spezifiziert durch RFC 7515³ zum Einsatz.

Die Signatur (siehe [Authentizität & Nichtabstreitbarkeit der Daten](#)) wird entsprechend im Signaturblock des Token abgelegt. Der qualifizierte elektronische Zeitstempel (siehe [Integrität der Daten](#)) wird mit dem Schlüssel `x-qtimestamp` als Header des Token hinzugefügt.

5.2.4 Komprimierung der Chunks

Die Chunks können mittels GZIP⁴ oder Brotli⁵ komprimiert werden. Hierfür wird der **Content-Encoding**-Header bei der Antwort angegeben.

5.2.5 Abholen der Events

Um die Events abzuholen sendet LUGAS eine Anfrage mit dem zuletzt abgeholten Chunk an die Schnittstelle.

Daraufhin wird dann der Nachfolger des über den Pfad spezifizierten Chunks zurückgegeben. Es werden nur Chunks zurückgegeben, die bereits abgeschlossen sind und nicht mehr verändert werden. Mittel des **Accept-Encoding**-Headers gibt LUGAS die unterstützten Komprimierungsalgorithmen an. Die Authentizität des **payload**-Objekts wird durch die JSON Web Signatur (RFC 7515⁶) im **signature**-Feld sichergestellt. Dabei wird die Signatur mit *Detached Content* entsprechend Appendix F⁷ und *Unencoded Content* entsprechend RFC 7797⁸ verwendet.

Path:

1	GET /events/{chunk}/next
---	--------------------------

³<https://www.rfc-editor.org/rfc/rfc7515>

⁴<https://www.gnu.org/software/gzip/>

⁵<https://github.com/google/brotli>

⁶<https://www.rfc-editor.org/rfc/rfc7515>

⁷<https://www.rfc-editor.org/rfc/rfc7515#appendix-F>

⁸<https://www.rfc-editor.org/rfc/rfc7797>

Name	Typ	Beschreibung	Beispiel
chunk	String	Die ID des angefragten Chunks	“20220512-2”

- Für den ersten Chunk muss für **chunk** der Wert **initial** angegeben werden.

Anfrage-Header:

Name	Typ	Beschreibung	Beispiel
Accept-Encoding	array[string]	Liste der von LUGAS unterstützten Komprimierungsalgorithmen (Komprimierung der Chunks)	List[“gzip”, “br”]

Antworten

- 404 (NOT FOUND)** - Es existiert noch kein Nachfolger zum angefragten Chunk.
- 200 (OK)** - Der angefragte Chunk wurde gefunden.

Response-Header:

Name	Datentyp	Beschreibung	Pflicht
Content-Encoding	String	Definiert das verwendete Encoding (wie Kompression) entsprechend der offiziellen Spezifikation ⁹	Ja

Response-Body:

Name	Datentyp	Beschreibung	Pflicht
payload	json	Enthält den angefragten Chunk und weiterführende Informationen (Siehe payload)	Ja
signature	String	Signatur des Payloads mit <i>Detached Content</i> und <i>Unencoded Content</i>	Ja

⁹<https://developer.mozilla.org/en-US/docs/Web/HTTP/Headers/Content-Encoding>

Payload:

Name	Datentyp	Beschreibung	Pflicht
id	String	Eindeutige Kennung des Chunks	Ja
predecessor	json	Enthält die id und signature des vorherigen Chunks	Bedingtes Pflichtfeld
data	array[json]	Enthält Liste der Ereignisse in zeitlich und logisch geordneter Reihenfolge.	Ja

Predecessor:

Name	Datentyp	Beschreibung	Pflicht
id	String	Eindeutige Kennung des vorherigen Chunks	Ja
signature	String	Signatur des vorherigen Chunks	Ja

Hinweise

- Das **predecessor**-Objekt enthält immer die Kennung und die Signatur des vorangegangenen Chunks. Hier wird eine Referenz auf den Chunk aus dem Beispiel erster Chunk dargestellt.
- Der allererste Chunk stellt einen Sonderfall dar. Da hier kein Vorgänger existiert, ist das **predecessor**-Objekt null und wird nicht mit angegeben.

Beispiel Erster Chunk:

```
1 GET /events/initial/next
```

```
1 {
2   "payload": {
3     "id": "3608e046-c713-4864-b559-867018fadb5c",
4     "data": [
5       {
6         "eventName": "PlayerRegistration",
7         "eventTime": "2022-06-08T01:12:59.000000Z",
8         "playerId": "1",
9         "userName": "QuickGamer"
10      },
11      {
12        "eventName": "PlayerRegistration",
13        "eventTime": "2022-06-08T01:14:55.000000Z",
14        "playerId": "77",
15        "userName": "McMoneyMaker"
16      }
17    ]
18  },
19  "signature": "eyJhbGciOiJIUzI1NiIsImI2NCI6ZmFsc2UsImNyaXQiOlsiYjY0Il0sInR5cCI6IkpXVCJ9..7smqrCvRv_-K6toM2K6ze6CxVEFiACWiD5vWDrx8HoetBdzdT3gSI0J9o-aEML-e0nfXBYTQPDg1HTC50pYy0A"
```

Beispiel Normaler Chunk:

```
1 GET /events/3608e046-c713-4864-b559-867018fadb5c/next
```

```
1 {
2   "payload": {
3     "id": "ccc03e9c-2cec-44d2-a165-836813baf87c",
4     "predecessor": {
5       "id": "3608e046-c713-4864-b559-867018fadb5c",
6       "signature": "
7         eyJhbGciOiJFUzI1NiIsImI2NCI6ZmFsc2UsImNyaXQiOlsiYjY0Il0
8         sInR5cCI6IkpXVCJ9..7smqrCvRv_-
9         K6toM2K6ze6CxVEFiACWiD5vWDrx8HoetBd
10        zdT3gSI0J9o-aEML-e0nfXBYTQPDg1HTC50pYy0A"
11      },
12      "data": [
13        {
14          "eventName": "PlayerRegistration",
15          "eventTime": "2022-06-08T01:18:02.000000Z",
16          "playerId": "22",
17          "userName": "LuckyBoi"
18        },
19        {
20          "eventName": "PlayerRegistration",
21          "eventTime": "2022-06-08T01:23:20.000000Z",
22          "playerId": "10993",
23          "userName": "Spieler4"
24        }
25      ]
26    },
27    "signature": "eyJhbGciOiJFUzI1NiIsImI2NCI6ZmFsc2UsImNyaXQiOlsiYjY0Il0
28    sInR5cCI6IkpXVCJ9..sR8Vo0-SigEwW3Vyn_rjnyafE-vVky6kPdl40uu0TDeu
29    QR-H9SxDEo6loZGZuK0izo0mixgj0CXr5lDIfyoSma"
```

5.2.6 Feedback

Feedback von LUGAS zur Verarbeitung der Events.

```
1 POST /feedback
```

Übermittelt für jeden verarbeiteten Chunk das Feedback mit Informationen über den Verlauf der Abholung der Events durch LUGAS.

Das Feedback enthält die Kennung des Chunk und eventuell aufgetretene Fehler. Das **errors**-Array wird nur gesetzt wenn **status** ungleich **success** ist.

Response-Body:

Name	Datentyp	Beschreibung	Pflicht
result	String	Feedback Status. Mögliche Werte "success", "canceled" oder "error"	Ja
chunkId	String	Eindeutige Kennung des verarbeitenden Chunks	Ja
processedEvents	int	Anzahl der verarbeitenden Events innerhalb des Chunks	Ja
errors	array[json]	Enthält die Error Nachrichten	Nein

Errors:

Name	Datentyp	Beschreibung	Pflicht
message	String	Beschreibung des Fehlers	Ja
errorEvent	String	EventId das den Fehler ausgelöst hat	Ja

Hinweise

- Die EventId ergibt sich implizit aus der Position des Events innerhalb des *data*-Arrays des *payload*-Objekts.

Antworten

- 204 (OK)** - Das Feedback wurde erfolgreich entgegengenommen.

6 Betriebsphasen

6.1 Einrichtung

Der Betrieb startet mit der *Einrichtungsphase*. In dieser Phase werden sämtliche benötigten Bestandsdaten des Anbieters als *Ereignisse* in logisch aufeinander aufbauender Reihenfolge auf dem Safe-Server abgelegt. Diese sind:

- Bereiche
- angebotene Spiele mit den jeweiligen Genehmigungsinformationen

- erlaubte Zahlungsdienste
- Spieler inklusive ihres Status, ggf. existierender Sperren oder anbieterbezogenen Limits, Kontostands und hinterlegten Zahlungswegen

Das anhängige Beispiel *Initiale Anbindung Safe-Server* beschreibt exemplarisch, wie solch eine Ablage anhand der konkreten Ereignisse aussehen kann.

Während dieser Phase gilt die Einschränkung, dass nur Bestandsdaten abgelegt werden können, aber noch keine operativen Daten. Daher wird angeraten, den produktiven Spielbetrieb während dieser Phase zu unterbrechen um Inkonsistenzen, insbesondere innerhalb der Spielerkonten, zu vermeiden. Es liegt im Interesse des Anbieters, diese Phase möglichst kurz zu halten. Die Behörde stellt ein Werkzeug bereit ("SafeDemo"), mit dem Abholvorgänge simuliert und die vom Anbieter erzeugten Nachrichten vorab geprüft werden können.

Das zusätzliche Ablegen des Ereignisses *Ende Einrichtungsphase* signalisiert das Ende der Phase aus Anbietersicht. Werden alle Ereignisse vom Auswertesystem ohne Fehler verarbeitet, endet die *Einrichtungsphase* automatisch und geht nahtlos in die *Normalbetriebsphase* über.

6.2 Normalbetrieb

In der *Normalbetriebsphase* können fortan alle operativen Daten abgelegt werden.

7 Zeichensatz und Datentypen

Ereignisse werden als JSON übertragen und richten sich dementsprechend nach dem JSON-Standard¹⁰. Zur weiteren Erklärung einzelner Felder und deren Einschränkungen werden in diesem Dokument folgende Bezeichner verwendet:

Bezeichner	Erklärung	Beispiel
string	Eine JSON-Zeichenkette. Steuerzeichen sowie führende, abschließende oder aufeinanderfolgende Leerzeichen sind nicht erlaubt. Die Länge ist auf 4096 Zeichen begrenzt.	"Hallo Welt!"
smallString	Ein <i>string</i> mit der Einschränkung, dass die Länge auf 256 Zeichen begrenzt ist.	"Hallo Welt!"

¹⁰<https://www.json.org/>

Bezeichner	Erklärung	Beispiel
identifizier	Ein <i>string</i> mit der Einschränkung, dass die Länge auf 36 Zeichen begrenzt ist. Beim Vergleich zweier Daten dieses Typs wird Groß- und Kleinschreibung unterschieden. Er ist geeignet, um jedwede 64-Bit-Zahl aufzunehmen. Er ist eindeutig für seinen jeweiligen Bestimmungszweck wie bspw. Spieler, Zahlungsweg, Einzelspieler-Spiel, Einzelspieler-Match, etc. und wird, so nicht in dieser Richtlinie anders definiert (bspw. für <i>eventName</i> , welcher für jedes Ereignis vordefiniert ist), vom Anbieter vergeben.	"5emh/t5chEyN8S68yG1Rug=="
date	Eine JSON-Zeichenkette, die ein Datum enthält und nach ISO-Standard 8601 ¹¹ formatiert ist.	"2021-01-26"
datetime	Eine JSON-Zeichenkette, die ein Datum, eine Uhrzeit und die entsprechende Zeitzone enthält und nach ISO-Standard 8601 formatiert ist.	"2021-01-26T09:37:37Z"
decimal	Eine JSON-Zahl, für die keine Exponentialschreibweise und für die maximal zwei Nachkommastellen zulässig sind. Der Wertebereich entspricht der Definition des Datentyps decimal in C# unter der Einschränkung zweier Nachkommastellen.	19.47
money	Ein <i>decimal</i> , das einen Geldbetrag ausdrückt. Dieser ist stets in Euro angegeben.	19.47
bool	Gibt einen JSON-Wahrheitswert an.	true
int	Ein <i>decimal</i> mit der Einschränkung, dass Nachkommastellen nicht zulässig sind.	123

8 Ereignisse

Die Ereignisse werden in Form von JSON innerhalb von **Chunks** übertragen. Dabei ist ein Chunk auf 200.000 Events beschränkt. Innerhalb eines Tages dürfen die Events auf mehrere Chunks aufgeteilt werden. Die Events müssen als *eventTime* den jeweiligen Tag haben. Der letzte Chunk des Tages muss spätestens um 2 Uhr nachts (UTC) am Folgetag abgeschlossen sein.

Beispiel

Data innerhalb eines Chunks:

```
1  [
2    {
3      "eventName": "PlayerDataApproved",
4      "eventTime": "2020-01-01T09:42:29.725Z",
5      "playerId": "5emh/t5chEyN8S68yG1Rug=="
6    },
7    {
8      "eventName": "PlayerStatus",
9      "eventTime": "2020-01-01T09:43:29.725Z",
10     "playerId": "5emh/t5chEyN8S68yG1Rug==",
11     "status": "VerificationPending",
12     "intent": "RegularCycle"
13   }
14 ]
```

Es gelten folgende Bedingungen und Einschränkungen:

- Chunks müssen vollständig geschrieben sein, sobald eine neuer Chunk erstellt wurde.
- Eine einmal übertragender Chunk darf ohne Anweisung der Behörde nicht umbenannt, verschoben, gelöscht oder geändert werden.

Hinweis zu nicht verpflichtenden Feldern:

Nicht verpflichtende Felder in den Ereignissen, welche nicht befüllt werden, dürfen nicht in die abgelegten Dateien aufgenommen werden. Dabei ist zu beachten, dass jegliches abgegebene, nicht verpflichtende Feld mit einer konkreten Bedeutung einhergeht und teilweise den gesamten Deutungszweck des Ereignisses maßgeblich beeinflusst. Dies bedeutet im Umkehrschluss, dass diese Felder unter bestimmten Bedingungen verpflichtend anzugeben sind, wenn eben jene Deutung notwendig ist. Generell gilt die Maßgabe, dass nicht verpflichtende Felder gefüllt werden müssen, wenn die einzutragenden Daten für den Zweck des spezifischen Ereignisses existieren. So ist es bspw. nicht zulässig, beim **Registrieren** eines Spielers das Feld *userName* nicht zu übermitteln, wenn für den Spieler ein Nutzernamen vorliegt. Alle weiteren Informationen dazu sind den Hinweisen unter jedem Ereignis zu entnehmen.

¹¹<https://www.iso.org/iso-8601-date-and-time-format.html>

Hinweis zur Ereignisvalidierung:

Beim Verarbeiten der Dateien wird die Plausibilität der Ereignisse geprüft. Wird beispielsweise erkannt, dass ein Ereignis für einen Spieler übermittelt wird, welcher vorher nicht über ein separates Ereignis angelegt wurde, so wird der Import aller Ereignisse pausiert bis das Problem behoben wurde. In diesem Fall nimmt die Behörde Kontakt zu den Anbietern auf.

Zum Zwecke der Validierung teilt der Anbieter in speziellen Ereignissen die Genehmigungen in Bezug auf erlaubte Spiele und Zahlungsdienste mit, die er von der Behörde erhalten hat.

Es folgt eine Übersicht aller Ereignisse, welche von den Anbietern bereitgestellt werden müssen.

8.1 Personendaten Spieler

Hier werden alle Ereignisse aufgeführt, welche Spielerdaten enthalten. Personenbezogene Daten müssen in transliterierter Form abgelegt werden (siehe Anlage Transliterationsvorschrift, LUGAS).

8.1.1 Registrieren

Ein Spieler registriert sich mit seinen Daten beim Anbieter.

Erlaubte Phasen

- Einrichtungsphase
- Normalbetriebsphase

Felder

Name	Datentyp	Beschreibung	Pflicht	Einschränkung
eventName	identifizier	Eindeutiger Bezeichner des Ereignisses	ja	Muss den Wert "PlayerRegistration" enthalten.
eventTime	datetime	Zeitpunkt der Registrierung des Spielers	ja	-
playerId	identifizier	Kennung des Spielers bei dem Anbieter	ja	-
userName	smallString	Der Benutzername des Spielers beim Anbieter	nein	-
fullName	smallString	Vor- und Nachname des Spielers	ja	Wert folgt Anlage Transliterationsvorschrift, LUGAS
postCode	smallString	Die Postleitzahl des Wohnorts des Spielers	ja	Wert folgt Anlage Transliterationsvorschrift, LUGAS
place	smallString	Der Wohnort der Adresse des Spielers	ja	Wert folgt Anlage Transliterationsvorschrift, LUGAS

Name	Datentyp	Beschreibung	Pflicht	Einschränkung
area	smallString	Das Bundesland des Wohnorts des Spielers	ja	Wert folgt Anlage Transliterationsvorschrift, LUGAS
country	smallString	Das Land der angegebenen Adresse	ja	Wert folgt Anlage Transliterationsvorschrift, LUGAS
citizenship	smallString	Die Staatsangehörigkeit des Spielers	ja	Wert folgt Anlage Transliterationsvorschrift, LUGAS
street	smallString	Die Straße der Adresse des Spielers	ja	Wert folgt Anlage Transliterationsvorschrift, LUGAS
number	smallString	Die Hausnummer(n) der Adresse des Spielers	ja	Wert folgt Anlage Transliterationsvorschrift, LUGAS
birthDate	date	Geburtsdatum des Spielers	ja	Wert folgt Anlage Transliterationsvorschrift, LUGAS
birthLocation	smallString	Geburtsort des Spielers	ja	Wert folgt Anlage Transliterationsvorschrift, LUGAS
birthName	smallString	Geburtsname des Spielers	ja	Wert folgt Anlage Transliterationsvorschrift, LUGAS
automaticProfitTransactionmoney		Betragsgrenze, ab welcher Gewinne dem Spielers automatisch ausgezahlt werden.	nein	Muss positiv sein.

Beispiel

```
1 {
2   "eventName": "PlayerRegistration",
3   "eventTime": "2020-11-20T09:42:29.725Z",
4   "playerId": "5emh/t5chEyN8S68yG1Rug==",
5   "userName": "crazyMax123",
6   "fullName": "MAX MUSTERMANN",
7   "postCode": "06128",
8   "place": "HALLE SAALE",
9   "area": "DE-ST",
10  "country": "DE",
11  "citizenship": "DE",
12  "street": "GRONINGERWEG",
13  "number": "15",
14  "birthDate": "1990-11-20",
15  "birthLocation": "MÜNSTER",
16  "birthName": "TEICHLER"
17 }
```

Hinweise

- Um an Glücksspielen teilzunehmen, sind Spieler nach § 6a Abs. 2 GlüStV 2021 verpflichtet, ein Spielkonto einzurichten. Eine Spielteilnahme ohne Spielkonto ist unzulässig.
- Spieler müssen nach § 4 Abs. 3 GlüStV 2021 mindestens 18 Jahre alt sein, um sich registrieren zu dürfen.
- Hat sich der Spieler vor initialer Einrichtung des Safe-Servers registriert, so ist in *eventTime* die Zeit der Ablage der Daten auf dem Safe-Server anzugeben. Gleiches gilt für den Fall, wenn ein Anbieter von nun an meldepflichtige Spiele anbietet und somit Spieler “nachmeldet”.
- Nach der Registrierung erhält der Spieler in der *Normalbetriebsphase* den Status “Verification-Pending” mit *intent* “Registration”, welches ebenfalls als Ereignis hinterlegt werden muss (siehe [Status](#)), da seine Daten durch ein geeignetes und zuverlässiges Verfahren (siehe § 6a Abs. 2 GlüStV 2021) verifiziert werden müssen. In der *Einrichtungsphase* sind alle Spieler mit ihrem aktuellen Status zu übermitteln, die nicht den Status *AccountClosed* haben. Kein Spieler darf den Status “VerificationPending” mit dem *intent* “TemporarilyAccepted” haben.
- Ein in der *Normalbetriebsphase* registrierter Spieler ist berechtigt, vorläufig an Glücksspielen teilzunehmen. Zu diesem Zweck bestätigt er die Kenntnisnahme der Hinweise (siehe [Bestätigung vorläufiges Spiel](#)).
- Wenn der Hauptwohnsitz des Spielers außerhalb Deutschlands liegt, kann der Anbieter einen Nebenwohnsitz in Deutschland statt des Hauptwohnsitzes verwenden. Hierbei kann es sich um einen gemeldeten Nebenwohnsitz oder einen gewöhnlichen Aufenthaltsort handeln. Beim gewöhnlichen Aufenthaltsort handelt es sich um einen Ort, an dem der Spieler nicht nur vorübergehend verweilt, sondern mit gewisser Dauer und Regelmäßigkeit.

- In *citizenship* ist der Staat einzutragen, welcher auf dem Ausweisdokument, mit dem sich der Spieler beim Anbieter registriert hat, angegeben ist.
- *userName* ist nur verpflichtend anzugeben, wenn ein Nutzernamen vergeben wurde.
- Anbieter sind nach § 6b Abs. 3 GlüStV 2021 verpflichtet, eine Funktion zur Verfügung zu stellen, mit der festgelegt werden kann, dass Gewinne über einem bestimmten Betrag automatisch ausgezahlt werden. Dieser Betrag ist in *automaticProfitTransaction* zu hinterlegen, wenn der Spieler diese Funktion nutzt.

8.1.2 Datenänderung

Der Spieler aktualisiert seine Daten beim Anbieter.

Erlaubte Phasen

- Normalbetriebsphase

Felder

Name	Datentyp	Beschreibung	Pflicht	Einschränkung
eventName	identifizier	Eindeutiger Bezeichner des Ereignisses	ja	Muss den Wert "PlayerUpdate" enthalten.
eventTime	datetime	Zeitpunkt der Datenänderung des Spielers	ja	-
playerId	identifizier	Kennung des Spielers bei dem Anbieter	ja	Verweis auf einen existierenden Spieler
userName	smallString	Der Benutzername des Spielers beim Anbieter	nein	-
fullName	smallString	Vor- und Nachname des Spielers	ja	Wert folgt Anlage Transliterationsvorschrift, LUGAS
postCode	smallString	Die Postleitzahl des Wohnorts des Spielers	ja	Wert folgt Anlage Transliterationsvorschrift, LUGAS

Name	Datentyp	Beschreibung	Pflicht	Einschränkung
place	smallString	Der Wohnort der Adresse des Spielers	ja	Wert folgt Anlage Transliterationsvorschrift, LUGAS
area	smallString	Das Bundesland des Wohnorts des Spielers	ja	Wert folgt Anlage Transliterationsvorschrift, LUGAS
country	smallString	Das Land der angegebenen Adresse	ja	Wert folgt Anlage Transliterationsvorschrift, LUGAS
citizenship	smallString	Die Staatsangehörigkeit des Spielers	ja	Wert folgt Anlage Transliterationsvorschrift, LUGAS
street	smallString	Die Straße der Adresse des Spielers	ja	Wert folgt Anlage Transliterationsvorschrift, LUGAS
number	smallString	Die Hausnummer(n) der Adresse des Spielers	ja	Wert folgt Anlage Transliterationsvorschrift, LUGAS
birthDate	date	Geburtsdatum des Spielers	ja	Wert folgt Anlage Transliterationsvorschrift, LUGAS
birthLocation	smallString	Geburtsort des Spielers	ja	Wert folgt Anlage Transliterationsvorschrift, LUGAS
birthName	smallString	Geburtsname des Spielers	ja	Wert folgt Anlage Transliterationsvorschrift, LUGAS
automaticProfitTransactionmoney		Betragsgrenze, ab welcher Gewinne automatisch auf das Bankkonto des Spielers überwiesen werden.	nein	Muss positiv sein.

Beispiel

```
1 {  
2   "eventName": "PlayerUpdate",  
3   "eventTime": "2020-12-20T09:42:29.725Z",  
4   "playerId": "5emh/t5chEyN8S68yG1Rug==",  
5   "userName": "crazyMax123",  
6   "fullName": "MAX MUSTERMANN",  
7   "postCode": "39114",  
8   "place": "MAGDEBURG",  
9   "area": "DE-ST",  
10  "country": "DE",  
11  "citizenship": "DE",  
12  "street": "BERLINERCHAUSSEE",  
13  "number": "15",  
14  "birthDate": "1990-11-20",  
15  "birthLocation": "MÜNSTER",  
16  "birthName": "TEICHLER"  
17 }
```

Hinweise

- Wenn sich die Daten eines Spielers ändern, müssen diese nach § 6a Abs. 5 S. 3 u. 4 GlüStV 2021 erneut verifiziert werden.
- Der Spieler erhält den Status “VerificationPending” mit *intent* “DataChanged”, welches ebenfalls als Ereignis hinterlegt werden muss (siehe [Status](#)). Dieser Status ist ebenfalls zu setzen, wenn der Spieler seine Daten im Zuge der jährlichen Datenprüfung ändert.
- Bei einer Datenänderung müssen ebenfalls die Spielerdaten in den Zentraldateien geändert werden.

8.1.3 Datenbestätigung

Der Spieler bestätigt die Richtigkeit seiner Daten. Dies passiert im Zuge einer regelmäßigen Überprüfung seiner Daten durch den Anbieter, welche nach § 6a Abs. 5 GlüStV 2021 geregelt ist.

Erlaubte Phasen

- Normalbetriebsphase

Felder

Name	Datentyp	Beschreibung	Pflicht	Einschränkung
eventName	identifizier	Eindeutiger Bezeichner des Ereignisses	ja	Muss den Wert "PlayerDataApproved" enthalten.
eventTime	datetime	Zeitpunkt der Bestätigung des Spielers	ja	-
playerId	identifizier	Kennung des Spielers bei dem Anbieter	ja	Verweis auf einen existierenden Spieler

Beispiel

```
1 {  
2   "eventName": "PlayerDataApproved",  
3   "eventTime": "2020-11-20T09:42:29.725Z",  
4   "playerId": "5emh/t5chEyN8S68yG1Rug=="  
5 }
```

Hinweise

- Nach § 6a Abs. 5 S. 5 GlüStV 2021 ist der Anbieter verpflichtet, diese Überprüfung spätestens ein Jahr nach der letzten Bestätigung durchzuführen, bevor der Spieler an einem neuen Match teilnehmen kann.
- Der Spieler erhält den Status "VerificationPending" mit *intent* "RegularCycle", welches ebenfalls als Ereignis hinterlegt werden muss (siehe [Status](#)).
- Sobald der Spieler seine Daten in diesem Zuge bestätigt hat, erhält er den Status "Verified", welches ebenfalls als Ereignis hinterlegt werden muss (siehe [Status](#)).

8.1.4 Bestätigung vorläufiges Spiel

Der Spieler möchte nach § 6a Abs. 4 GlüStV 2021 vorläufig an Glücksspielen teilnehmen und bestätigt die Kenntnisnahme der Hinweise zu den Einschränkungen des vorläufigen Spiels.

Erlaubte Phasen

- Normalbetriebsphase

Felder

Name	Datentyp	Beschreibung	Pflicht	Einschränkung
eventName	identifizier	Eindeutiger Bezeichner des Ereignisses	ja	Muss den Wert "PlayerTemporarily" enthalten.
eventTime	datetime	Zeitpunkt, zu dem der Spieler die Hinweise bestätigt	ja	-
playerId	identifizier	Kennung des Spielers bei dem Anbieter	ja	Verweis auf einen existierenden Spieler

Beispiel

```
1 {  
2   "eventName": "PlayerTemporarily",  
3   "eventTime": "2020-11-20T09:42:29.725Z",  
4   "playerId": "5emh/t5chEyN8S68yG1Rug=="  
5 }
```

Hinweise

- Das vorläufige Spiel ist nur für Spieler möglich, die während der *Normalbetriebsphase* registriert wurden und deren Daten aufgrund dieser Registrierung überprüft werden.
- Der Spieler erhält den Status "VerificationPending" mit *intent* "TemporarilyAccepted", welches ebenfalls als Ereignis hinterlegt werden muss (siehe [Status](#)).

8.1.5 Status

Ein Spieler durchläuft mehrere Zustände, während er ein Spielerkonto beim Anbieter besitzt.

Erlaubte Phasen

- Einrichtungsphase
- Normalbetriebsphase

Felder

Name	Datentyp	Beschreibung	Pflicht	Einschränkung
eventName	identifizier	Eindeutiger Bezeichner des Ereignisses	ja	Muss den Wert "PlayerStatus" enthalten.
eventTime	datetime	Zeitpunkt der Statusänderung des Spielers	ja	-
playerId	identifizier	Kennung des Spielers bei dem Anbieter	ja	Verweis auf einen existierenden Spieler
status	identifizier	Der aktuelle Status des Spielers	ja	Siehe Tabelle unten für die gültigen Werte
intent	identifizier	Eine ggf. weitere erklärende Ursache	bedingtes Pflichtfeld	Siehe Tabelle unten für die gültigen Werte

Folgende Status sind möglich:

Status (<i>status</i>)	Bedeutung
VerificationPending	Die Daten des Spielers werden überprüft. Mit Ausnahme des vorläufigen Spiels darf der Spieler in diesem Status nicht an Glücksspielen teilnehmen.
VerificationFailed	Die Spielerdaten konnten nicht verifiziert werden. Der Spieler darf nicht spielen.
Verified	Die Daten des Spielers wurden verifiziert und der Spieler darf in vollem Umfang spielen.
AccountClosed	Der Spieler hat sein Spielkonto beim Anbieter geschlossen.

Für die Zustände "VerificationPending" und "VerificationFailed" ist eine folgender Ursachen verpflichtend anzugeben. Jede dieser Ursachen ist in beiden Zuständen gültig. Schlägt eine Überprüfung im Status "VerificationPending" fehl, so ist der Status auf "VerificationFailed" zu setzen und die Ursache zu übernehmen.

Ursache (<i>intent</i>)	Bedeutung
Registration	Der Spieler hat sich beim Anbieter registriert.
TemporarilyAccepted	Der Spieler hat zugestimmt, dass bis zur erstmaligen Prüfung seiner Daten nur vorläufiges, beschränktes Spielen nach § 6a Abs. 4 GlüStV 2021 möglich ist.
DataChanged	Der Spieler hat seine Daten nach § 6a Abs. 5 GlüStV 2021 aktualisiert.
ChangedPayment	Der Spieler hat seine Zahlungsinformationen nach § 6a Abs. 6 GlüStV 2021 geändert.
RegularCycle	Der Spieler hat seine Daten im Zuge der regelmäßigen Datenprüfung nach § 6a Abs. 5 GlüStV 2021 bestätigt.

Beispiel

```

1 {
2   "eventName": "PlayerStatus",
3   "eventTime": "2020-11-20T09:42:29.725Z",
4   "playerId": "5emh/t5chEyN8S68yG1Rug==",
5   "status": "VerificationPending",
6   "intent": "RegularCycle"
7 }
```

Hinweise

- Wird der Status des Spielers während der *Einrichtungsphase* übertragen, so ist in *eventTime* die Zeit der Ablage der Daten auf dem Safe-Server anzugeben.
- Enthält *status* den Wert "VerificationPending" oder "VerificationFailed" so ist *intent* verpflichtend zu setzen.
- In der *Einrichtungsphase* darf *intent* nicht "TemporarilyAccepted" sein. Spieler, deren Registrierungsdaten in der *Einrichtungsphase* übermittelt wurden, dürfen auch in der *Normalbetriebsphase* niemals diesen *intent* erhalten.
- Veranlasst der Anbieter die jährliche Überprüfung der Spielerdaten nach § 6a Abs. 5 GlüStV 2021, erhält der Spieler den Status "VerificationPending" mit *intent* "RegularCycle".
- Ändert der Spieler seine Daten aufgrund der jährlichen Überprüfung, so wechselt der *intent* von "RegularCycle" zu "DataChanged".
- Bestätigt der Spieler seine Daten, so erhält er den Status "Verified", welches ebenfalls als Ereignis hinterlegt werden muss (siehe [Datenbestätigung](#)).

8.2 Spielersperrung

8.2.1 Sperre aktivieren

Ein Spieler wurde im System des Anbieters gesperrt.

Erlaubte Phasen

- Einrichtungsphase
- Normalbetriebsphase

Felder

Name	Datentyp	Beschreibung	Pflicht	Einschränkung
eventName	identifizier	Eindeutiger Bezeichner des Ereignisses	ja	Muss den Wert "PlayerSuspension" enthalten.
eventTime	datetime	Zeitpunkt der Aktivierung der Sperre	ja	-
playerId	identifizier	Kennung des Spielers bei dem Anbieter	ja	Verweis auf einen existierenden Spieler
suspensionReason	identifizier	Der Grund der Sperre	ja	Siehe Tabelle unten für gültige Werte
suspensionEndsAt	dateTime	Gibt an, wann die ggf. temporäre Sperre endet	nein	-

Folgende Gründe sind möglich:

Grund (<i>suspensionReason</i>)	Erklärung
Addiction	Ein Spieler wird vom Anbieter aufgrund einer erkannten Suchtproblematik nach § 6i Abs. 1 GlüStV 2021 gesperrt.
Bot	Ein Spieler wird vom Anbieter aufgrund des Einsatzes einer Bot-Software beim Online-Poker, welches einen Verstoß nach § 22b Abs. 3 GlüStV 2021 darstellt, gesperrt.

Grund (<i>suspensionReason</i>)	Erklärung
Legal	Ein Spieler wird beim Anbieter nach einem der in § 6a Abs. 8 GlüStV 2021 aufgeführten Gründe gesperrt. Diese sind Verdacht auf unrechtmäßig erworbene Gewinne, sowie Verstöße gegen gesetzliche Bestimmungen, insbesondere im Bereich der Geldwäsche, gegen den vorliegenden Staatsvertrag oder gegen Bedingungen für das Spielkonto.
Voluntary	Ein Spieler löst selbstständig eine sofortige kurzzeitige Sperre nach § 6i Abs. 3 GlüStV 2021 aus.

Wird der Spieler aus einem anderen Grund vom Anbieter gesperrt, so ist dieses Ereignis nicht abzulegen.

Beispiel

```

1 {
2   "eventName": "PlayerSuspension",
3   "eventTime": "2020-11-20T09:42:29.725Z",
4   "playerId": "5emh/t5chEyN8S68yG1Rug==",
5   "suspensionReason": "Legal"
6 }
```

Hinweise

- Wird der Sperrstatus des Spielers während der *Einrichtungsphase* übertragen, so ist in *eventTime* die Zeit der Ablage der Daten auf dem Safe-Server anzugeben. Sperren mit *suspensionReason* "Voluntary" sind während dieser Phase nicht zulässig.
- Handelt es sich um eine temporäre Sperre, die selbstständig ausläuft, so ist *suspensionEndsAt* anzugeben. Ein explizites Melden der Aufhebung dieser Sperre (siehe [Sperre aufheben](#)) ist dann nicht zwingend nötig.
- Sperren mit *suspensionReason* "Voluntary" sind immer als temporäre Sperren von 24 Stunden anzugeben.
- Ist ein Spieler aus mehreren Gründen gesperrt, so ist dieses Ereignis einmal pro Grund auf dem Safe-Server anzugeben.
- Wurde bereits eine noch nicht ausgelaufene Sperre mit einem bestimmten Grund angegeben, so wird ein erneutes Ereignis mit dem gleichen *suspensionReason* als Aktualisierung der bekannten Sperre angesehen. Es erfolgt keine Differenzierung der Sperren innerhalb eines Sperrgrunds.

- Es ist immer der konkretest mögliche Sperrgrund anzugeben. Wurde bei einem Spieler der Einsatz einer Bot-Software festgestellt, so ist *suspensionReason* mit “Bot” anzugeben, obwohl “Legal” inhaltlich ebenfalls zutreffend wäre.

8.2.2 Sperre aufheben

Eine während des Spiels beim Anbieter ausgelöste Sperre wird aufgehoben.

Erlaubte Phasen

- Normalbetriebsphase

Felder

Name	Datentyp	Beschreibung	Pflicht	Einschränkung
eventName	identifizier	Eindeutiger Bezeichner des Ereignisses	ja	Muss den Wert “PlayerResumption” enthalten.
eventTime	datetime	Zeitpunkt der Aufhebung der Sperre, wenn <i>suspensionEndsAt</i> nicht übermittelt wird.	ja	-
playerId	identifizier	Kennung des Spielers bei dem Anbieter	ja	Verweis auf einen existierenden Spieler
suspensionReason	identifizier	Der Grund der Sperre	ja	Siehe Tabelle oben für gültige Werte
suspensionEndsAt	dateTime	Gibt an, wann die ggf. temporäre Sperre endet	nein	-

Beispiel

```

1 {
2   "eventName": "PlayerResumption",
3   "eventTime": "2020-11-20T09:42:29.725Z",
4   "playerId": "5emh/t5chEyN8S68yG1Rug==",
5   "suspensionReason": "Legal"
6 }

```

Hinweise

- Dieses Ereignis kann zur Verlängerung, Verkürzung und Aufhebung einer temporären Sperre (ggf. auch mehrfach) übermittelt werden.
- Abweichend davon ist es nicht zulässig, Sperren mit *suspensionReason* "Voluntary" zu verlängern.
- Wird *suspensionEndsAt* nicht übermittelt, so gilt die Sperre aus Sicht des Auswertesystems mit dem Zeitpunkt von *eventTime* als aufgehoben.

8.3 Anbieterbezogene Spielerlimits**8.3.1 Limit setzen**

Der Spieler setzt ein Limit nach § 6c Abs. 2 S. 1 GlüStV 2021, welches nur für den Anbieter gilt.

Erlaubte Phasen

- Einrichtungsphase
- Normalbetriebsphase

Felder

Name	Datentyp	Beschreibung	Pflicht	Einschränkung
eventName	identifizier	Eindeutiger Bezeichner des Ereignisses	ja	Muss den Wert "LocalLimitSet" enthalten.
eventTime	datetime	Zeitpunkt der Festsetzung des lokalen Limits beim Anbieter durch den Spieler	ja	-

Name	Datentyp	Beschreibung	Pflicht	Einschränkung
playerId	identifizier	Kennung des Spielers bei dem Anbieter	ja	Verweis auf einen existierenden Spieler
limitScope	identifizier	Art des Limits	ja	Siehe Tabelle unten für die gültigen Werte
timeUnit	identifizier	Intervall des Limits	ja	Siehe Tabelle unten für die gültigen Werte
amount	money	Das gesetzte Limit	ja	Muss positiv sein.

Folgende Limits sind möglich:

Art (<i>limitScope</i>)	Bedeutung
Deposit	Das Limit bezieht sich auf die Summe der Höhe der Einzahlungen, die im Intervall getätigt werden dürfen.
Stake	Das Limit bezieht sich auf die Summe der Einsätze, die im Intervall getätigt werden dürfen.
Loss	Das Limit bezieht sich auf die Summe des Verlusts, der im Intervall gemacht werden darf.

Folgende Intervalle sind möglich:

Intervall (<i>timeUnit</i>)	Bedeutung
Day	Ein tägliches Limit
Week	Ein wöchentliches Limit
Month	Ein monatliches Limit

Beispiel

```
1 {  
2   "eventName": "LocalLimitSet",  
3   "eventTime": "2020-11-20T09:42:29.725Z",  
4   "playerId": "5emh/t5chEyN8S68yG1Rug==",  
5   "limitScope": "Stake",  
6   "timeUnit": "Month",  
7   "amount": 499.47  
8 }
```

Hinweise

- Alle vom Spieler gesetzten Limits sind zu übertragen.
- Wird das Limit des Spielers während der *Einrichtungsphase* übertragen, so ist in *eventTime* die Zeit der Ablage der Daten auf dem Safe-Server anzugeben.
- Es kann nur ein Limit pro *limitScope* und *timeUnit* gesetzt werden.
- Das Setzen eines anbieterbezogenen monatlichen Einzahlungslimits hat keinen Einfluss auf das bestehende anbieterübergreifende monatliche Einzahlungslimit nach § 6c Abs. 1 GlüStV 2021.

8.3.2 Limit löschen

Der Spieler entfernt ein eingetragenes anbieterbezogenes Limit.

Erlaubte Phasen

- Normalbetriebsphase

Felder

Name	Datentyp	Beschreibung	Pflicht	Einschränkung
eventName	identifizier	Eindeutiger Bezeichner des Ereignisses	ja	Muss den Wert "LocalLimitUnset" enthalten.
eventTime	datetime	Zeitpunkt der Entfernung des lokalen Limits beim Anbieter durch den Spieler	ja	-
playerId	identifizier	Kennung des Spielers bei dem Anbieter	ja	Verweis auf einen existierenden Spieler
limitScope	identifizier	Art des Limits	ja	Siehe Tabelle oben für die gültigen Werte
timeUnit	identifizier	Intervall des Limits	ja	Siehe Tabelle oben für die gültigen Werte

Beispiel

```

1 {
2   "eventName": "LocalLimitUnset",
3   "eventTime": "2020-11-20T09:42:29.725Z",
4   "playerId": "5emh/t5chEyN8S68yG1Rug==",
5   "limitScope": "Stake",
6   "timeUnit": "Month"
7 }
```

Hinweise

- Das Löschen eines Limits verhält sich hinsichtlich des Wirksamwerdens wie eine Erhöhung auf den größtmöglichen Wert nach § 6c Abs. 3 S. 2 GlüStV 2021. Somit hat es innerhalb der gesetzlichen Schutzfrist weiterhin Gültigkeit und ggf. auftretende Verstöße werden durch das Auswertesystem erkannt.

8.4 Matches

8.4.1 Start Einzelspieler-Match

Informationen zum Start eines Matches, welches ein Spieler allein durchführt.

Erlaubte Phasen

- Normalbetriebsphase

Felder

Name	Datentyp	Beschreibung	Pflicht	Einschränkung
eventName	identifizier	Eindeutiger Bezeichner des Ereignisses	ja	Muss den Wert "SinglePlayerMatch" enthalten.
eventTime	datetime	Zeitpunkt, zu dem der Spieler das Match gestartet hat	ja	-
playerId	identifizier	Kennung des Spielers bei dem Anbieter	ja	Verweis auf einen existierenden Spieler
gameId	identifizier	Eindeutige Kennung des Spiels	ja	Verweis auf ein vorhandenes Einzelspieler-Spiel
matchId	identifizier	Eindeutige Kennung des laufenden Matches	ja	-

Name	Datentyp	Beschreibung	Pflicht	Einschränkung
profitRatio	decimal	Gibt die durchschnittliche Auszahlungsquote je einen Euro Spieleinsatz gemäß § 22a Abs. 3 Satz 2 GlüStV 2021 an, die sich aus der Überprüfung durch eine Zertifizierungsstelle ergibt. Falls ein solches Zertifikat nicht existiert, ist die durchschnittliche Auszahlungsquote anzugeben, die sich mathematisch nach einer unendlichen Anzahl von Spielen ergeben sollte.	ja	Muss größer als 0 sein.
charge	identifizier	Gibt an, um welche Art von Spiel es sich handelt	ja	Siehe Tabelle unten für gültige Werte.

Folgende Arten sind möglich:

Unterhaltungsangebot (<i>charge</i>)	Bedeutung
Real	Es handelt sich um ein entgeltliches Spiel
Free	Es handelt sich um ein unentgeltliches Unterhaltungsangebot
Bonus	Es handelt sich um ein Spiel mit Bonusgeld
RealAndBonus	Es handelt sich um ein Spiel mit Echt- und Bonusgeld

Beispiel

```
1 {  
2   "eventName": "SinglePlayerMatch",  
3   "eventTime": "2020-11-20T09:51:29.725Z",  
4   "playerId": "5emh/t5chEyN8S68yG1Rug==",  
5   "gameId": "123456",  
6   "matchId": "HD50jhp9b0qD4LBJw7vL1A==",  
7   "profitRatio": 12.47,  
8   "charge": "Real"  
9 }
```

Hinweise

- *gameld* ist ein Verweis zu den von dem Anbieter übermittelten Einzelspieler-Spielen (siehe [Einzelspieler-Spiel hinzufügen](#)).
- Vor dem Start eines Spiels ist es erforderlich, dass für dieses Spiel eine Genehmigung angelegt wurde (siehe [Genehmigung Einzelspieler-Spiel](#)).
- Bei rundenbasierten Einzelspieler-Spielen ist jede Runde nach § 22a Abs. 4 S. 2 GlüStV 2021 als eigenes Match zu behandeln, welches den gesetzlichen Anforderungen wie maximaler Einsatzhöhe und [expliziter Spielererklärung](#) vollumfänglich genügen muss.
- Dieses Ereignis ist sowohl für Promotionen als auch unentgeltliche Angebote nach § 6j GlüStV 2021 zu hinterlegen.
- Enthält *charge* den Wert "Bonus" müssen "bonusStake" und "bonusRevenue" in [Ende Einzelspieler-Match](#) angegeben werden, es muss aber kein [Transaktion Spielerkonto](#) angegeben werden.
- Enthält *charge* den Wert "RealAndBonus" müssen "bonusStake" und "bonusRevenue" in [Ende Einzelspieler-Match](#) angegeben werden, ein [Transaktion Spielerkonto](#) muss nur für den Anteil des Echtgeldes angegeben werden.

8.4.2 Ende Einzelspieler-Match

Informationen zum Ende eines Matches, welches ein Spieler allein durchgeführt hat.

Erlaubte Phasen

- Normalbetriebsphase

Felder

Name	Datentyp	Beschreibung	Pflicht	Einschränkung
eventName	identifizier	Eindeutiger Bezeichner des Ereignisses	ja	Muss den Wert "SinglePlayerMatchResult" enthalten.
eventTime	datetime	Zeitpunkt, an dem das Match beendet wurde	ja	-
matchId	identifizier	Eindeutige Kennung des beendeten Matches	ja	Verweis auf das bestehende Match des Spielers
stake	money	Gesamter Echtgeld Einsatz des Spielers während des Matches	ja	Darf nicht negativ sein.
revenue	money	Gesamter Echtgeld Gewinn oder Verlust des Spielers durch das Match	ja	-
bonusStake	money	Gesamter Bonusgeld Einsatz des Spielers während des Matches	bedingtes Pflichtfeld	Darf nicht negativ sein. Siehe Hinweis.
bonusRevenue	money	Gesamter Bonusgeld Gewinn oder Verlust des Spielers durch das Match	bedingtes Pflichtfeld	Siehe Hinweis.

Beispiel

```
1 {
2   "eventName": "SinglePlayerMatchResult",
3   "eventTime": "2020-11-20T09:51:29.725Z",
4   "matchId": "HD50jhp9b0qD4LBJw7vL1A==",
5   "stake": 19.47,
6   "revenue": -15
7 }
```

Beispiel Bonusgeld

```
1 {
2   "eventName": "SinglePlayerMatchResult",
3   "eventTime": "2020-11-20T09:51:29.725Z",
4   "matchId": "HD50jhp9b0qD4LBJw7vL1A==",
5   "stake": 0,
6   "revenue": 0,
7   "bonusStake": 15,
8   "bonusRevenue": -3
9 }
```

Beispiel Echt- und Bonusgeld

```
1 {
2   "eventName": "SinglePlayerMatchResult",
3   "eventTime": "2020-11-20T09:51:29.725Z",
4   "matchId": "HD50jhp9b0qD4LBJw7vL1A==",
5   "stake": 19.47,
6   "revenue": -15,
7   "bonusStake": 2.47,
8   "bonusRevenue": 2
9 }
```

Hinweise

- *matchId* ist ein Verweis zu einem laufenden Einzelspieler-Match (siehe [Start Einzelspieler-Match](#)).
- Zu jedem Matchabschluss, welcher Gewinn oder Verlust in Echtgeld enthält, muss eine Transaktion des Spielkontos zugeordnet sein (siehe [Transaktion Spielkonto](#)). Zwischen dem Ende des Spiels und der Transaktion dürfen keine weiteren Events zum Spieler liegen.
- Dieses Ereignis ist sowohl für unentgeltliche als auch monetäre Matches zu hinterlegen.
- Bei einem unentgeltlichen Angebot ist *stake* und *revenue* auf 0 zu setzen.
- *bonusStake* und *bonusRevenue* sind zu setzen, sollte es sich um ein Einspieler-Match mit Bonusgeld oder Echt- und Bonusgeld handeln (siehe [Start Einzelspieler-Match](#)).

8.4.3 Start Gemeinschaftsspieler-Match Spieler

Informationen zum Einstieg in ein Match, welches ein Spieler mit anderen Spielern durchführt.

Erlaubte Phasen

- Normalbetriebsphase

Felder

Name	Datentyp	Beschreibung	Pflicht	Einschränkung
eventName	identifizier	Eindeutiger Bezeichner des Ereignisses	ja	Muss den Wert "MultiPlayerMatch" enthalten.
eventTime	datetime	Zeitpunkt, an dem der Spieler dem Gemeinschaftsspieler-Match beigetreten ist	ja	-
playerId	identifizier	Kennung des Spielers bei dem Anbieter	ja	Verweis auf einen existierenden Spieler
gameId	identifizier	Eindeutige Kennung des Spiels	ja	Verweis auf ein vorhandenes Gemeinschaftsspieler-Spiel
matchId	identifizier	Eindeutige Kennung des Matches	ja	-
charge	identifizier	Gibt an, um welche Art von Spiel es sich handelt	ja	Siehe Tabelle unten für gültige Werte.
tableNumber	identifizier	Der Tisch an dem der Spieler spielt.	ja	-
networkHost	smallString	So es sich um ein Netzwerk-Match handelt, Name des Anbieters	nein	-

Name	Datentyp	Beschreibung	Pflicht	Einschränkung
networkGameId	identifizier	So es sich um ein Netzwerk-Match handelt, ID des Netzwerkspiels	nein	-

Folgende Arten sind möglich:

Unterhaltungsangebot (<i>charge</i>)	Bedeutung
Real	Es handelt sich um ein entgeltliches Spiel
Free	Es handelt sich um ein unentgeltliches Unterhaltungsangebot
Bonus	Es handelt sich um ein Spiel mit Bonusgeld
RealAndBonus	Es handelt sich um ein Spiel mit Echt- und Bonusgeld

Beispiel

```

1  {
2    "eventName": "MultiPlayerMatch",
3    "eventTime": "2020-11-20T09:51:29.725Z",
4    "playerId": "5emh/t5chEyN8S68yG1Rug==",
5    "gameId": "789",
6    "matchId": "HD50jhp9b0qD4LBJw7vL1A==",
7    "charge": "Real",
8    "tableNumber": "T4411x1",
9    "networkHost": "ZPOKER",
10   "networkGameId": "MyweoQRQe0KPxR9enkjWbA=="
11  }
```

Hinweise

- *gameId* ist ein Verweis zu den von dem Anbieter übermittelten Gemeinschaftsspieler-Spielen (siehe [Gemeinschaftsspieler-Spiel hinzufügen](#)).
- Vor dem Start eines Spiels ist es erforderlich, dass für dieses Spiel eine Genehmigung angelegt wurde (siehe [Genehmigung Gemeinschaftsspieler-Spiel](#)).
- Ein Gemeinschaftsspieler-Match wird mit der Teilnahme des ersten Spielers gestartet. Eine explizite Deklaration des Match-Starts ist nicht vorgesehen. *eventTime* definiert den Zeitpunkt, wann der Spieler beginnt, sich an dem Match zu beteiligen, und lässt keine Rückschlüsse darauf zu, wann das Match insgesamt gestartet wurde.

- Handelt es sich um ein Netzwerkmatch, so sind *networkHost* und *networkGameld* verpflichtend anzugeben.
- Dieses Ereignis ist sowohl für Promotionen als auch unentgeltliche Angebote nach § 6j GlüStV 2021 zu hinterlegen.
- Enthält *charge* den Wert "Bonus" müssen "bonusStake" und "bonusRevenue" in [Ende Gemeinschaftsspieler-Match Spieler](#) angegeben werden, es muss aber kein [Transaktion Spielerkonto](#) angegeben werden.
- Enthält *charge* den Wert "RealAndBonus" müssen "bonusStake" und "bonusRevenue" in [Ende Gemeinschaftsspieler-Match Spieler](#) angegeben werden, ein [Transaktion Spielerkonto](#) muss nur für den Anteil des Echtgeldes angegeben werden.

8.4.4 Ende Gemeinschaftsspieler-Match Spieler

Informationen zum Ausstieg aus einem Match, welches ein Spieler mit anderen Spielern durchführt.

Erlaubte Phasen

- Normalbetriebsphase

Felder

Name	Datentyp	Beschreibung	Pflicht	Einschränkung
eventName	identifizier	Eindeutiger Bezeichner des Ereignisses	ja	Muss den Wert "Multi-PlayerMatchResult" enthalten.
eventTime	datetime	Zeitpunkt, an dem der Spieler das Match verlassen hat	ja	-
playerId	identifizier	Kennung des Spielers bei dem Anbieter	ja	Verweis auf einen existierenden Spieler
matchId	identifizier	Eindeutige Kennung des aktuell laufenden Matches	ja	Verweis auf das laufende Gemeinschaftsspieler-Match
stake	money	Gesamter Echtgeld Einsatz des Spielers während des Matches	ja	Darf nicht negativ sein.

Name	Datentyp	Beschreibung	Pflicht	Einschränkung
rake	money	Die Gebühr, die der Spieler effektiv für die Teilnahme am Gemeinschaftsspieler-Match zahlen musste.	ja	Darf nicht negativ sein.
totalRake	money	Gesamte Gebühr für die Teilnahme am Gemeinschaftsspieler-Match	ja	Darf nicht negativ sein.
revenue	money	Gesamter Echtgeld Gewinn oder Verlust des Spielers durch das Match	ja	-
bonusStake	money	Gesamter Bonusgeld Einsatz des Spielers während des Matches	bedingtes Pflichtfeld	Darf nicht negativ sein. Siehe Hinweis.
bonusRevenue	money	Gesamter Bonusgeld Gewinn oder Verlust des Spielers durch das Match	bedingtes Pflichtfeld	Siehe Hinweis.

Beispiel

```

1 {
2   "eventName": "MultiPlayerMatchResult",
3   "eventTime": "2020-11-20T09:51:29.725Z",
4   "playerId": "5emh/t5chEyN8S68yG1Rug==",
5   "matchId": "HD50jhp9b0qD4LBJw7vL1A==",
6   "stake": 19.47,
7   "rake": 0.47,
8   "totalRake": 1,
9   "revenue": -15
10 }
```

Beispiel Bonusgeld

```
1 {
2   "eventName": "MultiPlayerMatchResult",
3   "eventTime": "2020-11-20T09:51:29.725Z",
4   "playerId": "5emh/t5chEyN8S68yG1Rug==",
5   "matchId": "HD50jhp9b0qD4LBJw7vL1A==",
6   "stake": 0,
7   "rake": 0.47,
8   "totalRake": 1,
9   "revenue": 0,
10  "bonusStake": 15.3,
11  "bonusRevenue": -6
12 }
```

Beispiel Echt- und Bonusgeld

```
1 {
2   "eventName": "MultiPlayerMatchResult",
3   "eventTime": "2020-11-20T09:51:29.725Z",
4   "playerId": "5emh/t5chEyN8S68yG1Rug==",
5   "matchId": "HD50jhp9b0qD4LBJw7vL1A==",
6   "stake": 4,
7   "rake": 0.47,
8   "totalRake": 1,
9   "revenue": -3,
10  "bonusStake": 15.3,
11  "bonusRevenue": -6
12 }
```

Hinweise

- *matchId* ist ein Verweis zu dem von dem Anbieter übermittelten Gemeinschaftsspieler-Match (siehe [Start Gemeinschaftsspieler-Match Spieler](#)).
- *eventTime* definiert den Zeitpunkt, wann der Spieler aus dem Match aussteigt, und lässt keine Rückschlüsse darauf zu, wann das Match beendet wird. Ein laufendes Gemeinschaftsspieler-Match wird ausschließlich über das Ereignis [Ende Gemeinschaftsspieler-Spiel](#) geschlossen. So ist es auch möglich, dass ein Spieler ein Spiel verlässt und später zufällig wieder an demselben, noch laufenden Spiel erneut teilnimmt. Für jede Teilnahme sind gesonderte Start- und Endereignisse zu übermitteln.
- Zu jedem Matchabschluss, welcher Gewinn oder Verlust in Echtgeld enthält, muss eine Transaktion des Spielkontos zugeordnet sein (siehe [Transaktion Spielkonto](#)). Zwischen dem Ende des Spiels und der Transaktion dürfen keine weiteren Events zum Spieler liegen.
- Bei *totalRake* ist der gesamte eingezogene Betrag zu erfassen, auch wenn der Anbieter nur einen Teil oder nichts von den eingezogenen Gebühren behält.
- Dieses Ereignis ist sowohl für unentgeltliche als auch monetäre Matches zu hinterlegen.

- Bei einem unentgeltlichem Angebot ist *stake*, *rake*, *totalRake* und *revenue* auf 0 zu setzen.
- *bonusStake* und *bonusRevenue* sind zu setzen, sollte es sich um ein Einspieler-Match mit Bonusgeld oder Echt- und Bonusgeld handeln (siehe [Start Gemeinschaftsspieler-Match Spieler](#)).

8.4.5 Ende Gemeinschaftsspieler-Spiel

Der Anbieter teilt mit, dass ein Gemeinschaftsspieler-Spiel aus seiner Sicht beendet und eine Teilnahme von Spielern nicht mehr möglich ist.

Erlaubte Phasen

- Normalbetriebsphase

Felder

Name	Datentyp	Beschreibung	Pflicht	Einschränkung
eventName	identifizier	Eindeutiger Bezeichner des Ereignisses	ja	Muss den Wert "Multi-PlayerMatchClosure" enthalten.
eventTime	datetime	Zeitpunkt, an dem der Anbieter das Match für geschlossen erklärt	ja	-
matchId	identifizier	Eindeutige Kennung des aktuell laufenden Matches	ja	Verweis auf das laufende Gemeinschaftsspieler-Match

Beispiel

```

1 {
2   "eventName": "MultiPlayerMatchClosure",
3   "eventTime": "2020-11-20T09:51:29.725Z",
4   "matchId": "HD50jhp9b0qD4LBJw7vL1A=="
5 }
```

Hinweise

- Dieses Ereignis ist nur nach Verlassen aller Spieler aus dem laufenden Gemeinschaftsspieler-Spiel zulässig.

8.5 Zahlungen

8.5.1 Zahlungsweg hinzufügen

Der Spieler registriert einen Zahlungsweg nach § 6b Abs. 4 GlüStV 2021, über das Ein- und/oder Auszahlungen vorgenommen werden können. Ein Zahlungsweg basiert auf einem Zahlungsdienst, welche nach Anlage Zahlungsdienste, LUGAS definiert sind.

Erlaubte Phasen

- Einrichtungsphase
- Normalbetriebsphase

Felder

Name	Datentyp	Beschreibung	Pflicht	Einschränkung
eventName	identifizier	Eindeutiger Bezeichner des Ereignisses	ja	Muss den Wert "PaymentModalityRegistration" enthalten.
eventTime	datetime	Zeitpunkt, an dem der Spieler seinen Zahlungsweg einträgt	ja	-
paymentModalityId	identifizier	Eindeutige ID des Zahlungswegs des Spielers beim Anbieter	ja	-
playerId	identifizier	Kennung des Spielers bei dem Anbieter	ja	Verweis auf einen existierenden Spieler
methodName	string	Name des verwendeten Zahlungsweges	ja	Die Werte folgen der Anlage Zahlungsdienste, sowie der Anlage Translationsvorschrift LUGAS

Name	Datentyp	Beschreibung	Pflicht	Einschränkung
idData	string	Anzahl der gleichen Zahlungswege pro Spieler	ja	Enthält für jeden Zahlungsweg pro Spieler einen Buchstaben. Alphabetisch aufsteigend nach Anzahl des einzelnen Zahlungsweiges. Die Werte folgen der Anlage Transliterationsvorschrift LUGAS

Beispiel

```

1 {
2   "eventName": "PaymentModalityRegistration",
3   "eventTime": "2020-11-20T09:42:29.725Z",
4   "paymentModalityId": "MyweoQRQe0KPxR9enkjWbA==",
5   "playerId": "5emh/t5chEyN8S68yG1Rug==",
6   "methodName" : "KLARNA",
7   "idData" : "A"
8 }
9 {
10  "eventName": "PaymentModalityRegistration",
11  "eventTime": "2020-11-21T17:33:30.725Z",
12  "paymentModalityId" : "ByxciQrtndaZTkaOPR8nsa==",
13  "playerId": "5emh/t5chEyN8S68yG1Rug==",
14  "methodName" : "KLARNA",
15  "idData" : "B"
16 }
```

Hinweise

- Wird der Zahlungsweg des Spielers während der *Einrichtungsphase* übertragen, so ist in *eventTime* die Zeit der Ablage der Daten auf dem Safe-Server anzugeben.
- Die Anlage Zahlungsdienste wird von der Behörde regelmäßig durch die gemeldeten Daten der Anbieter im Rahmen der Erlaubnisvergabe aktualisiert.
- Die verpflichtende Information im Feld *idData* richtet sich analog dazu nach der Anzahl der gewählten Zahlungsdienste pro Spieler und Anbieter.
- Das Hinzufügen eines Zahlungsweiges in der *Normalbetriebsphase* hat zur Folge, dass die Daten

des Spielers nach § 6a Abs. 6 S. 1 GlüStV 2021 neu geprüft werden. Der **Status des Spielers** wechselt auf “VerificationPending” mit *intent* “ChangedPayment”. Bis zur vollständigen Überprüfung darf der Spieler nach § 6a Abs. 6 S. 2 u. 3 GlüStV 2021 nicht an Spielen teilnehmen und es sind keine Ein- und Auszahlungen mit diesem Zahlungsdienst erlaubt. Dieser Status ist auch zu setzen, wenn der Status bereits auf “VerificationPending” steht. Das Hinzufügen eines Zahlungswegs in der *Einrichtungsphase* zieht keine Folgen nach sich.

- Zahlungswege, welche zum Zeitpunkt der Registrierung noch nicht durch die Anlage Zahlungsdienste, LUGAS beschrieben wurden, müssen vom Anbieter nachträglich gemeldet werden, sobald dies möglich ist.

8.5.2 Zahlungsweg entfernen

Der Spieler entfernt einen Zahlungsweg, über das Ein- und/oder Auszahlungen vorgenommen werden können.

Erlaubte Phasen

- Normalbetriebsphase

Felder

Name	Datentyp	Beschreibung	Pflicht	Einschränkung
eventName	identifizier	Eindeutiger Bezeichner des Ereignisses	ja	Muss den Wert “PaymentModality-Cancelation” enthalten.
eventTime	datetime	Zeitpunkt, an dem der Anbieter den Zahlungsweg als nicht mehr nutzbar deklariert.	ja	-
paymentModalityId	identifizier	Eindeutiger Zahlungsweg des Anbieters	ja	Verweis auf eine bereits vorher angelegten Zahlungsweg

Beispiel

```

1 {
2   "eventName": "PaymentModalityCancelation",
3   "eventTime": "2020-11-20T09:42:29.725Z",
4   "paymentModalityId": "MyweoQRQe0KPxR9enkjWbA=="
5 }

```

Hinweise

- *paymentModalityId* bezieht sich auf einen bereits registrierten Zahlungsweg, siehe [Zahlungsweg hinzufügen](#).
- Jede Änderung an einem bestehenden Zahlungsweg wird vom Auswertesystem als neuer Zahlungsweg interpretiert und benötigt daher eine eigene *paymentModalityId*, welche mit einem neuen [Zahlungsweg hinzufügen](#)-Ereignis zu senden und der bestehende Zahlungsweg über dieses Ereignis zu löschen ist.
- Eine Ein- oder Auszahlung mit diesem Zahlungsweg ist nach dem Entfernen nicht mehr erlaubt. Dabei ist zu beachten, dass ein zu löschender Zahlungsweg aufgrund von § 6a Abs. 6 S. 3 GlüStV 2021 ggf. länger vorgehalten werden muss.

8.5.3 Einzahlung

Der Spieler möchte Geld einzahlen.

Erlaubte Phasen

- Normalbetriebsphase

Felder

Name	Datentyp	Beschreibung	Pflicht	Einschränkung
eventName	identifizier	Eindeutiger Bezeichner des Ereignisses	ja	Muss den Wert "Deposit" enthalten.
eventTime	datetime	Zeitpunkt, zu dem der Spieler die Einzahlung beantragt	ja	-
paymentModalityId	identifizier	Eindeutige Kennung des Zahlungswegs	ja	Verweis auf einen existierenden Zahlungsweg

Name	Datentyp	Beschreibung	Pflicht	Einschränkung
amount	money	Der Betrag den der Spieler einzahlen will.	ja	Der Betrag muss positiv sein.

Beispiel

```

1 {
2   "eventName": "Deposit",
3   "eventTime": "2020-11-20T09:42:29.725Z",
4   "paymentModalityId": "MyweoQRQe0KPxR9enkjWbA==",
5   "amount": 199.47
6 }

```

Hinweise

- Einzahlungen dürfen nur über einen vorher registrierten Zahlungsweg erfolgen, welche über *paymentModalityId* referenziert wird (siehe [Zahlungsweg hinzufügen](#)).
- Jede Einzahlung ist nach § 6c Abs. 6 S. 1 GlüStV 2021 mit den Zentraldateien abzugleichen (siehe [Einzahlung](#)). Außer die Einzahlung erfolgt von einem Bonusgeld-Konto. Zwischen der Einzahlung und dem Abgleich mit den Zentraldateien dürfen keine weiteren Einzahlungs-Events zum Spieler liegen. Auch Zahlungen, welche von den Zentraldateien abgelehnt wurden, sind als Ereignis festzuhalten.
- Zu jeder akzeptierten Einzahlung muss eine Transaktion des Spielkontos zugeordnet sein (siehe [Transaktion Spielkonto](#)). Zwischen der akzeptierten Einzahlung und der Transaktion dürfen keine weiteren Events zum Spieler liegen.

8.5.4 Auszahlung

Der Spieler möchte Geld ausgezahlt haben.

Erlaubte Phasen

- Normalbetriebsphase

Felder

Name	Datentyp	Beschreibung	Pflicht	Einschränkung
eventName	identifizier	Eindeutiger Bezeichner des Ereignisses	ja	Muss den Wert "Payout" enthalten.

Name	Datentyp	Beschreibung	Pflicht	Einschränkung
eventTime	datetime	Zeitpunkt, zu dem die Auszahlung beantragt wurde	ja	-
paymentModalityId	identifizier	Eindeutige Kennung des Zahlungswegs	ja	Verweis auf einen existierenden Zahlungsweg
amount	money	Der Auszahlungsbetrag	ja	Der Betrag muss positiv sein.
automatic	bool	Gibt an, ob die Auszahlung aufgrund einer Betragsgrenze, ab welcher Gewinne dem Spielers automatisch ausgezahlt werden, veranlasst wurde	ja	-

Beispiel

```

1 {
2   "eventName": "Payout",
3   "eventTime": "2020-11-20T09:42:29.725Z",
4   "paymentModalityId": "MyweoQRQe0KPxR9enkjWbA==",
5   "amount": 199.47,
6   "automatic": false
7 }
```

Hinweise

- Auszahlungen dürfen nur über einen vorher registrierten Zahlungsweg erfolgen (siehe [Zahlungsweg hinzufügen](#)).
- Zu jeder Auszahlung muss eine Transaktion des Spielkontos zugeordnet sein (siehe [Transaktion Spielkonto](#)). Zwischen der Auszahlung und der Transaktion dürfen keine weiteren Events zum Spieler liegen.

8.5.5 Transaktion Spielkonto

Der Spielkontostand des Spielers wurde durch eine Transaktion verändert.

Erlaubte Phasen

- Normalbetriebsphase

Felder

Name	Datentyp	Beschreibung	Pflicht	Einschränkung
eventName	identifizier	Eindeutiger Bezeichner des Ereignisses	ja	Muss den Wert "Transaction" enthalten.
eventTime	datetime	Zeitpunkt der Wirksamkeit der Spielkontostandsänderung	ja	-
playerId	identifizier	Kennung des Spielers bei dem Anbieter	ja	Verweis auf einen existierenden Spieler
transactionType	identifizier	Eindeutiger Typ der Transaktion	ja	Siehe Tabelle unten für die gültigen Werte
transactionReference	identifizier	Referenz der Transaktion	ja	Siehe Tabelle unten für den gültigen Verweis
oldBalance	money	Der Kontostand des Spielers vor der Transaktion	ja	-
newBalance	money	Der Kontostand des Spielers nach der Transaktion	ja	-

Als Quelle der Transaktion sind folgende Typen möglich:

Transaktions-Typ (<i>transactionType</i>)	Erklärung	Inhalt des Feldes (<i>transactionReference</i>)
Deposit	Es wurde eine Einzahlung vorgenommen.	ID des Zahlungswegs (<i>paymentModalityId</i>)
Payout	Es wurde eine Auszahlung vorgenommen.	ID des Zahlungswegs (<i>paymentModalityId</i>)

Transaktions-Typ (<i>transactionType</i>)	Erklärung	Inhalt des Feldes (<i>transactionReference</i>)
SinglePlayerMatchResult	Ein Einzelspieler-Match wurde beendet.	ID des Einzelspieler-Matches (<i>matchId</i>)
MultiPlayerMatchResult	Der Spieler hat ein Gemeinschaftsspieler-Match verlassen.	ID des Gemeinschaftsspieler-Matches (<i>matchId</i>)

Beispiel

```

1 {
2   "eventName": "Transaction",
3   "eventTime": "2020-11-20T09:42:29.725Z",
4   "playerId": "5emh/t5chEyN8S68yG1Rug==",
5   "transactionType": "Payout",
6   "transactionReference": "MyweoQRQe0KPxR9enkjWbA==",
7   "oldBalance": 212.47,
8   "newBalance": 12.47
9 }
```

Hinweise

- Dieses Ereignis ist nur bei einer Veränderung des Kontostands anzugeben.
- Bei einer Transaktion von *transactionType* "Deposit" darf sich der Kontostand ausschließlich erhöhen. Ist *transactionType* hingegen "Payout", darf sich der Kontostand ausschließlich verringern.
- Der hier geführte Kontostand bezieht sich ausschließlich auf monetäre Beträge. Ggf. existierende Bonusgeld-Konten für unentgeltliche Unterhaltungsangebote bedürfen keiner Kontoführung im Sinne des Auswertesystems und haben, im Gegensatz zu monetären Matches, keine Ereignisse dieses Typs zur Folge.

8.6 Interaktion zwischen Spieler und Anbieter

Der Anbieter ist verpflichtet, in regelmäßigen Intervallen mit den Spielern zu interagieren und deren Verhalten aufzuzeichnen.

8.6.1 Log-in

Der Spieler loggt sich bei dem Anbieter ein.

Erlaubte Phasen

- Normalbetriebsphase

Felder

Name	Datentyp	Beschreibung	Pflicht	Einschränkung
eventName	identifizier	Eindeutiger Bezeichner des Ereignisses	ja	Muss den Wert "PlayerLogin" enthalten.
eventTime	datetime	Zeitpunkt, zu dem sich der Spieler beim Anbieter eingeloggt hat	ja	-
playerId	identifizier	Kennung des Spielers bei dem Anbieter	ja	Verweis auf einen existierenden Spieler

Beispiel

```

1 {
2   "eventName": "PlayerLogin",
3   "eventTime": "2020-11-20T09:42:29.725Z",
4   "playerId": "5emh/t5chEyN8S68yG1Rug=="
5 }
```

8.6.2 Spielererklärung virtuelles Automatenspiel

Der Spieler gibt eine Erklärung nach § 22a Abs. 4 GlüStV 2021 ab, ein virtuelles Automatenspiel spielen zu wollen.

Erlaubte Phasen

- Normalbetriebsphase

Felder

Name	Datentyp	Beschreibung	Pflicht	Einschränkung
eventName	identifizier	Eindeutiger Bezeichner des Ereignisses	ja	Muss den Wert "SinglePlayerMatchDeclaration" enthalten.

Name	Datentyp	Beschreibung	Pflicht	Einschränkung
eventTime	datetime	Zeitpunkt der Erklärungsabgabe des Spielers	ja	-
playerId	identifizier	Kennung des Spielers bei dem Anbieter	ja	Verweis auf einen existierenden Spieler
gameId	identifizier	Eindeutiger Bezeichner des Spiels beim Anbieter	ja	Verweis auf ein existierendes Spiel

Beispiel

```
1 {  
2   "eventName": "SinglePlayerMatchDeclaration",  
3   "eventTime": "2020-11-20T09:43:02.725Z",  
4   "playerId": "5emh/t5chEyN8S68yG1Rug==",  
5   "gameId": "123456"  
6 }
```

Hinweise

- Ein Match nach § 22a Abs. 6 GlüStV 2021 besteht bei einem virtuellen Automatenspiel aus einer einzelnen Spielrunde, bspw. dem einmaligen betätigen einer Slotmaschine. Jedes erneute Betätigen der Maschine ist ein neues Match und benötigt eine erneute Spielererklärung.

8.6.3 Spieler-Timeout

Meldung darüber, dass ein Spieler 30 Minuten inaktiv war.

Erlaubte Phasen

- Normalbetriebsphase

Felder

Name	Datentyp	Beschreibung	Pflicht	Einschränkung
eventName	identifizier	Eindeutiger Bezeichner des Ereignisses	ja	Muss den Wert "PlayerTimeout" enthalten.
eventTime	datetime	Zeitpunkt, wann der Timeout erkannt wurde	ja	-
playerId	identifizier	Eindeutige Kennung des Spielers bei dem Anbieter	ja	Verweis auf einen existierenden Spieler

Beispiel

```
1 {  
2   "eventName": "PlayerTimeout",  
3   "eventTime": "2020-11-20T09:42:29.725Z",  
4   "playerId": "5emh/t5chEyN8S68yG1Rug=="  
5 }
```

Hinweise

- Ein Timeout hat zur Folge, dass der Spieler nach § 6h Abs. 4 S. 5 GlüStV 2021 in der Aktivitätsdatei auf inaktiv zu setzen ist, sollte er für den Anbieter aktiv geschaltet sein.

8.6.4 Realitycheck

Ein Spieler muss, gemäß § 6h Abs. 7 GlüStV 2021, mindestens nach jeder Stunde Aktivität (laut Zentraldateien) auf die verstrichene Zeit hingewiesen werden.

Erlaubte Phasen

- Normalbetriebsphase

Felder

Name	Datentyp	Beschreibung	Pflicht	Einschränkung
eventName	identifizier	Eindeutiger Bezeichner des Ereignisses	ja	Muss den Wert "RealityCheckProvider" enthalten.
eventTime	datetime	Der Moment, an dem der Anbieter dem Spieler die Meldung erstmalig angezeigt hat oder hätte anzeigen wollen	ja	-
playerId	identifizier	Kennung des Spielers bei dem Anbieter	ja	Verweis auf einen existierenden Spieler

Beispiel

```
1 {  
2   "eventName": "RealityCheckProvider",  
3   "eventTime": "2020-11-20T09:42:40.725Z",  
4   "playerId": "5emh/t5chEyN8S68yG1Rug=="  
5 }
```

Hinweise

- Das Ereignis muss nur einmal angelegt werden, auch wenn der Anbieter den Hinweis dem Spieler mehrfach anzeigen musste (bspw. durch Reload der Website durch den Spieler).

8.6.5 Realitycheck Bestätigung

Ein Spieler bestätigt den Hinweis des Anbieters.

Erlaubte Phasen

- Normalbetriebsphase

Felder

Name	Datentyp	Beschreibung	Pflicht	Einschränkung
eventName	identifizier	Eindeutiger Bezeichner des Ereignisses	ja	Muss den Wert "RealityCheckPlayer" enthalten.
eventTime	datetime	Der Moment, an dem der Spieler die Meldung bestätigt hat oder die Aktivität (lt. Zentraldateien) beendet wurde	ja	-
playerId	identifizier	Kennung des Spielers bei dem Anbieter	ja	Verweis auf einen existierenden Spieler
hasActivityExpired	bool	Gibt an, ob das Ende des Ereignisses durch Ablauf der Aktivität eingetreten ist.	ja	-

Beispiel

```

1 {
2   "eventName": "RealityCheckPlayer",
3   "eventTime": "2020-11-20T09:42:40.725Z",
4   "playerId": "5emh/t5chEyN8S68yG1Rug==",
5   "hasActivityExpired": false
6 }
```

Hinweise

- Dieses Ereignis bezieht sich auf eine Erinnerung des Anbieters zur vergangenen Spielzeit (siehe [Realitycheck](#)).

- Zwischen dem Hinweis des Anbieters und der Bestätigung darf der Spieler nicht weiterspielen.
- Bei virtuellen Automaten Spielen ist insbesondere zu berücksichtigen, dass nach der Bestätigung eine Wartezeit von fünf Minuten einzuhalten ist.
- Bestätigt der Spieler den Hinweis des Anbieters nicht, bspw. durch dauerhaftes Verlassen der Seite oder sonstige Untätigkeit, so greift der **Spieler-Timeout**. In diesem Fall muss der Anbieter den Spieler nach § 6h Abs. 4 S. 5 GlüStV 2021 in der Aktivitätsdatei auf inaktiv setzen. Damit geht einher, dass dieses Ereignis ausgelöst wird. *hasActivityExpired* ist dementsprechend auf "true" zu setzen.

8.6.6 Erinnerung Bereichswechsel

Wenn ein Spieler den Spielbereich des Anbieters wechselt, muss er die Hinweise zu den Gefahren und zur Prävention vor Spielsucht sowie Beratungsangeboten, gemäß § 4 Abs. 5 Nr. 5 S. 4 GlüStV 2021 bestätigen. Mit dem Eintragen dieses Ereignisses meldet der Anbieter, dass der Spieler die Hinweise bestätigt hat.

Erlaubte Phasen

- Normalbetriebsphase

Felder

Name	Datentyp	Beschreibung	Pflicht	Einschränkung
eventName	identifizier	Eindeutiger Bezeichner des Ereignisses	ja	Muss den Wert "PlayerChangedArea" enthalten.
eventTime	datetime	Zeitpunkt, an dem der Spieler die Hinweise bestätigt hat.	ja	-
playerId	identifizier	Kennung des Spielers bei dem Anbieter	ja	Verweis auf einen existierenden Spieler
oldAreald	identifizier	Bereich, den der Spieler verlässt	ja	Verweis auf einen existierenden Bereich
newAreald	identifizier	Bereich, den der Spieler betritt	ja	Verweis auf einen existierenden Bereich

Beispiel

```

1 {
2   "eventName": "PlayerChangedArea",
3   "eventTime": "2020-11-20T09:43:01.725Z",
4   "playerId": "5emh/t5chEyN8S68yG1Rug==",
5   "oldAreaId": "15",
6   "newAreaId": "7"
7 }

```

Hinweise

- *oldAreaId* und *newAreaId* sind Referenzen zu den eingerichteten Bereichen des Anbieters (siehe [Bereich hinzufügen](#)) und müssen sich voneinander unterscheiden.

8.6.7 Korrekturaufforderung Limitfestsetzung

Bildet ab, dass dem Spieler eine Korrekturaufforderung nach § 6c Abs. 1 S. 9 GlüStV 2021 angezeigt wird, wenn er versucht hat ein zu hohes anbieterübergreifendes Limit festzusetzen.

Erlaubte Phasen

- Normalbetriebsphase

Felder

Name	Datentyp	Beschreibung	Pflicht	Einschränkung
eventName	identifizier	Eindeutiger Bezeichner des Ereignisses	ja	Muss den Wert "PlayerLimitCorrection-Request" enthalten.
eventTime	datetime	Zeitpunkt an dem die Aufforderung gesendet wurde	ja	-
playerId	identifizier	Kennung des Spielers bei dem Anbieter	ja	Verweis auf einen existierenden Spieler

Beispiel

```

1 {
2   "eventName": "PlayerLimitCorrectionRequest",
3   "eventTime": "2020-11-20T09:43:01.725Z",
4   "playerId": "5emh/t5chEyN8S68yG1Rug=="
5 }

```

8.6.8 Information statistische Daten

Bildet ab, dass ein Anbieter einem Spieler sämtliche kumulierten Einsätze, Gewinne und Verluste der jeweils vorangegangenen 30 Tage, gemäß § 6d Abs. 2 GlüStV 2021, angezeigt hat.

Erlaubte Phasen

- Normalbetriebsphase

Felder

Name	Datentyp	Beschreibung	Pflicht	Einschränkung
eventName	identifizier	Eindeutiger Bezeichner des Ereignisses	ja	Muss den Wert "PlayerStatistics" enthalten.
eventTime	datetime	Zeitpunkt, zu dem der Anbieter den Spieler die Meldung angezeigt hat.	ja	-
playerId	identifizier	Kennung des Spielers bei dem Anbieter	ja	Verweis auf einen existierenden Spieler
stake	money	Die Summe aller Einsätze in den letzten 30 Tagen	ja	Darf nicht negativ sein.
profit	money	Die Summe aller Gewinne in den letzten 30 Tagen	ja	Darf nicht negativ sein.
loss	money	Die Summe aller Verluste in den letzten 30 Tagen	ja	Darf nicht negativ sein.

Beispiel

```

1 {
2   "eventName": "PlayerStatistics",
3   "eventTime": "2020-11-20T09:43:01.725Z",
4   "playerId": "5emh/t5chEyN8S68yG1Rug==",
5   "stake": 99.47,
6   "profit": 12.47,
7   "loss": 86.47
8 }
```

8.6.9 Veranlassung Inaktivierung

Ein Spieler fordert den Anbieter nach § 6h Abs. 4 S. 2 GlüStV 2021 auf, ihn an der Aktivitätsdatei inaktiv zu schalten.

Erlaubte Phasen

- Normalbetriebsphase

Felder

Name	Datentyp	Beschreibung	Pflicht	Einschränkung
eventName	identifizier	Eindeutiger Bezeichner des Ereignisses	ja	Muss den Wert "PlayerLogout" enthalten.
eventTime	datetime	Zeitpunkt der Aufforderung des Spielers	ja	-
playerId	identifizier	Kennung des Spielers bei dem Anbieter	ja	Verweis auf einen existierenden Spieler

Beispiel

```

1 {
2   "eventName": "PlayerLogout",
3   "eventTime": "2020-11-20T09:43:01.725Z",
4   "playerId": "5emh/t5chEyN8S68yG1Rug=="
5 }
```

8.7 Nutzung der Zentraldateien

Der Anbieter hat die Nutzung der Zentraldateien zu protokollieren.

8.7.1 Übermittlung Spielerdaten

Bildet den Versuch ab, einen Spieler in den Zentraldateien anzulegen (vgl. § 6c Abs. 5 S. 4 bis 6 GlüStV 2021). Die Zentraldateien regeln in ihrer Spezifikation, wie die Antwort zu interpretieren ist.

Erlaubte Phasen

- Einrichtungsphase
- Normalbetriebsphase

Felder

Name	Datentyp	Beschreibung	Pflicht	Einschränkung
eventName	identifizier	Eindeutiger Bezeichner des Ereignisses	ja	Muss den Wert "ApiRequestRegisterPlayer" enthalten.
eventTime	datetime	Zeitpunkt, an dem die Anfrage an die Zentraldateien gesendet wurde	ja	-
playerId	identifizier	Kennung des Spielers bei dem Anbieter	ja	Verweis auf einen existierenden Spieler
depositLimit	money	Das bei der Registrierung gesetzte anbieterübergreifende Einzahlungslimit des Spielers	nein	Muss positiv sein.
registrationResult	identifizier	Das Ergebnis der Anfrage	ja	Siehe Tabelle unten für gültige Werte

Folgende Ergebnisse des API-Aufrufs sind möglich:

Ergebnis (<i>registrationResult</i>)	Erklärung
Successful	Der Spieler wurde erfolgreich angelegt.
Failed	Der Spieler konnte nicht erfolgreich angelegt werden.
Unspecified	Die Antwort der Zentraldateien passt nicht eindeutig in eines der spezifizierten Ergebnisse.
NotAvailable	Der Request konnte nicht an die API gesendet werden. Die Netzwerkverbindung ist nicht verfügbar oder der Webservice meldet einen 5xx-Statuscode.

Beispiel

```

1 {
2   "eventName": "ApiRequestRegisterPlayer",
3   "eventTime": "2020-11-20T09:43:01.725Z",
4   "playerId": "5emh/t5chEyN8S68yG1Rug==",
5   "depositLimit": 200,
6   "registrationResult": "Successful"
7 }
```

Hinweise

- Beim Setzen von *depositLimit* ist zu beachten, dass dies nach § 6c Abs. 3 S. 2 GlüStV 2021 ggf. erst später gültig wird, wenn es bereits ein bestehendes anbieterübergreifendes Einzahlungslimit gibt, welches kleiner ist als das hier gesetzte Limit. Zu der konkreten Mechanik der Wirksamkeit von anbieterübergreifenden Einzahlungslimits ist die Spezifikation der Zentraldateien heranzuziehen.
- Das *depositLimit* wird nicht übertragen, wenn der Spieler ein bereits bestehendes anbieterübergreifendes Einzahlungslimit nach § 6c Abs. 5 S. 4 GlüStV 2021 beibehält.

8.7.2 Anbieterübergreifendes Einzahlungslimit

Bildet den Versuch ab, für einen bereits an den Zentraldateien registrierten Spieler ein neues anbieterübergreifendes Limit nach § 6c Abs. 3 S. 1 GlüStV 2021 in den Zentraldateien zu setzen. Die Zentraldateien regeln in ihrer Spezifikation, wie die Antwort zu interpretieren ist.

Erlaubte Phasen

- Normalbetriebsphase

Felder

Name	Datentyp	Beschreibung	Pflicht	Einschränkung
eventName	identifizier	Eindeutiger Bezeichner des Ereignisses	ja	Muss den Wert "ApiRequestSetDepositLimit" enthalten.
eventTime	datetime	Zeitpunkt, an dem die Anfrage an die Zentraldateien gesendet wurde	ja	-
playerId	identifizier	Kennung des Spielers bei dem Anbieter	ja	Verweis auf einen existierenden Spieler
depositLimit	money	Das neue anbieterübergreifende Einzahlungslimit des Spielers	ja	Muss positiv sein.
depositLimitResult	identifizier	Das Ergebnis der Anfrage	ja	Siehe Tabelle unten für gültige Werte

Folgende Ergebnisse des API-Aufrufs sind möglich:

Ergebnis (<i>depositLimitResult</i>)	Erklärung
Successful	Der Limit wurde erfolgreich gesetzt.
Failed	Das Limit konnte nicht gesetzt werden.
Unspecified	Die Antwort der Zentraldateien passt nicht eindeutig in eines der spezifizierten Ergebnisse.
NotAvailable	Der Request konnte nicht an die API gesendet werden. Die Netzwerkverbindung ist nicht verfügbar oder der Webservice meldet einen 5xx-Statuscode.

Beispiel

```

1 {
2   "eventName": "ApiRequestSetDepositLimit",
3   "eventTime": "2020-11-20T09:43:01.725Z",
4   "playerId": "5emh/t5chEyN8S68yG1Rug==",
5   "depositLimit": 200,
6   "depositLimitResult": "Successful"
7 }

```

Hinweise

- Es ist zu beachten, dass das gesetzte Limit nach § 6c Abs. 3 S. 2 GlüStV 2021 ggf. erst später gültig wird, wenn es bereits ein bestehendes anbieterübergreifendes Einzahlungslimit gibt, welches kleiner ist als das hier gesetzte Limit. Zu der konkreten Mechanik der Wirksamkeit von anbieterübergreifenden Einzahlungslimits ist die Spezifikation der Zentraldateien heranzuziehen.

8.7.3 Einzahlung

Bildet den Versuch ab, eine Einzahlung nach § 6c Abs. 6 S. 1 GlüStV 2021 in den Zentraldateien vorzunehmen. Die Zentraldateien regeln in ihrer Spezifikation, wie die Antwort zu interpretieren ist.

Erlaubte Phasen

- Einrichtungsphase
- Normalbetriebsphase

Felder

Name	Datentyp	Beschreibung	Pflicht	Einschränkung
eventName	identifizier	Eindeutiger Bezeichner des Ereignisses	ja	Muss den Wert "ApiRequestMakeDeposit" enthalten.
eventTime	datetime	Zeitpunkt, an dem die Anfrage an die Zentraldateien gesendet wurde	ja	-
playerId	identifizier	Kennung des Spielers bei dem Anbieter	ja	Verweis auf einen existierenden Spieler

Name	Datentyp	Beschreibung	Pflicht	Einschränkung
amount	money	Die Höhe der Einzahlung	ja	Muss positiv sein.
depositResult	identifizier	Das Ergebnis der Anfrage	ja	Siehe Tabelle unten für gültige Werte

Folgende Ergebnisse des API-Aufrufs sind möglich:

Ergebnis (<i>depositResult</i>)	Erklärung
Successful	Die Einzahlung wurde erfolgreich vorgenommen.
Failed	Die Einzahlung wurde nicht erfolgreich vorgenommen.
Unspecified	Die Antwort der Zentraldateien passt nicht eindeutig in eines der spezifizierten Ergebnisse.
NotAvailable	Der Request konnte nicht an die API gesendet werden. Die Netzwerkverbindung ist nicht verfügbar oder der Webservice meldet einen 5xx-Statuscode.

Beispiel

```

1 {
2   "eventName": "ApiRequestMakeDeposit",
3   "eventTime": "2020-11-20T09:43:01.725Z",
4   "playerId": "5emh/t5chEyN8S68yG1Rug==",
5   "amount": 199.47,
6   "depositResult": "Successful"
7 }
```

Hinweise

- Einzahlungen, welche von den Zentraldateien abgelehnt werden, dürfen dem Spielerkonto nicht gutgeschrieben werden. In diesem Fall wird kein [Transaktionsereignis](#) abgelegt.
- Jede Einzahlung ist nach § 6c Abs. 6 S. 1 GlüStV 2021 mit den Zentraldateien abzugleichen (siehe [Einzahlung](#)). Außer die Einzahlung erfolgt von einem Bonusgeld-Konto. Zwischen der Einzahlung und dem Abgleich mit den Zentraldateien dürfen keine weiteren Einzahlungs-Events zum Spieler liegen. Auch Zahlungen, welche von den Zentraldateien abgelehnt wurden, sind als Ereignis festzuhalten.

- Zu jeder akzeptierten Einzahlung muss eine Transaktion des Spielkontos zugeordnet sein (siehe [Transaktion Spielerkonto](#)). Zwischen der akzeptierten Einzahlung und der Transaktion dürfen keine weiteren Events zum Spieler liegen.

8.7.4 Aktivstatus

Bildet den Versuch ab, einen Spieler in den Zentraldateien nach § 6h Abs. 3 GlüStV 2021 aktiv oder inaktiv zu setzen. Die Zentraldateien regeln in ihrer Spezifikation, wie die Antwort zu interpretieren ist.

Erlaubte Phasen

- Normalbetriebsphase

Felder

Name	Datentyp	Beschreibung	Pflicht	Einschränkung
eventName	identifizier	Eindeutiger Bezeichner des Ereignisses	ja	Muss den Wert "ApiRequestSetActiveState" enthalten.
eventTime	datetime	Zeitpunkt, an dem die Anfrage an die Zentraldateien gesendet wurde	ja	-
playerId	identifizier	Kennung des Spielers bei dem Anbieter	ja	Verweis auf einen existierenden Spieler
active	bool	Gibt den Wunsch des Anbieters an, ob der Spieler in der Datei als aktiv (true) oder inaktiv (false) geführt werden soll	ja	-
activeStateResult	identifizier	Das Ergebnis der Anfrage	ja	Siehe Tabelle unten für gültige Werte

Folgende Ergebnisse des API-Aufrufs sind möglich:

Ergebnis (<i>activeStateResult</i>)	Erklärung
Successful	Der Aktivitätsstatus wurde erfolgreich geändert.
Failed	Der Aktivitätsstatus wurde nicht erfolgreich geändert.
Unspecified	Die Antwort der Zentraldateien passt nicht eindeutig in eines der spezifizierten Ergebnisse.
NotAvailable	Der Request konnte nicht an die API gesendet werden. Die Netzwerkverbindung ist nicht verfügbar oder der Webservice meldet einen 5xx-Statuscode.

Beispiel

```

1 {
2   "eventName": "ApiRequestSetActiveState",
3   "eventTime": "2020-11-20T09:43:01.725Z",
4   "playerId": "5emh/t5chEyN8S68yG1Rug==",
5   "active": true,
6   "activeStateResult": "Successful"
7 }
```

8.7.5 Sperrprüfung

Bildet eine Anfrage an die Sperrdatei nach § 8 Absatz 3 Satz 4 GlüStV 2021 ab, die prüft, ob der Spieler gesperrt ist. Die Sperrdatei regelt in ihrer Spezifikation, wie die Antwort zu interpretieren ist.

Erlaubte Phasen

- Normalbetriebsphase

Felder

Name	Datentyp	Beschreibung	Pflicht	Einschränkung
eventName	identifizier	Eindeutiger Bezeichner des Ereignisses	ja	Muss den Wert "ApiRequestGetSuspendedState" enthalten.

Name	Datentyp	Beschreibung	Pflicht	Einschränkung
eventTime	datetime	Zeitpunkt, an dem die Anfrage an die Zentraldateien gesendet wurde	ja	-
playerId	identifizier	Kennung des Spielers bei dem Anbieter	ja	Verweis auf einen existierenden Spieler
suspensionResult	identifizier	Das Ergebnis der Anfrage	ja	Siehe Tabelle unten für gültige Werte

Folgende Ergebnisse des API-Aufrufs sind möglich:

Ergebnis (<i>suspensionResult</i>)	Erklärung
Permitted	Der Spieler darf an Glücksspielen teilnehmen.
Suspended	Der Spieler darf aufgrund einer Sperre nicht an Glücksspielen teilnehmen.
Unspecified	Die Antwort der Zentraldateien passt nicht eindeutig in eines der spezifizierten Ergebnisse.
NotAvailable	Der Request konnte nicht an die API gesendet werden. Die Netzwerkverbindung ist nicht verfügbar oder der Webservice meldet einen 5xx-Statuscode.

Beispiel

```
1 {  
2   "eventName": "ApiRequestGetSuspendedState",  
3   "eventTime": "2020-11-20T09:43:01.725Z",  
4   "playerId": "5emh/t5chEyN8S68yG1Rug==",  
5   "suspensionResult": "Suspended"  
6 }
```

8.8 Anbieterdaten

8.8.1 Bereich hinzufügen

Der Anbieter informiert über einen Spielbereich nach § 4 Abs. 5 Nr. 5 GlüStV 2021, den er eingerichtet hat.

Erlaubte Phasen

- Einrichtungsphase
- Normalbetriebsphase

Felder

Name	Datentyp	Beschreibung	Pflicht	Einschränkung
eventName	identifizier	Eindeutiger Bezeichner des Ereignisses	ja	Muss den Wert "ProviderAreaRegistration" enthalten.
eventTime	datetime	Zeitpunkt, wann der Anbieter den Bereich angelegt hat	ja	-
areald	identifizier	Eindeutige Kennung des Bereichs	ja	-
name	smallString	Name des Bereichs	ja	-

Beispiel

```
1 {  
2   "eventName": "ProviderAreaRegistration",  
3   "eventTime": "2020-11-20T09:42:29.725Z",  
4   "areaId": "213",  
5   "name": "Slotmaschinen-Paradies"  
6 }
```

Hinweise

- Bei der initialen Einrichtung des Safe-Servers sind alle bestehenden Spielbereiche zu übermitteln.

8.8.2 Einzelspieler-Spiel hinzufügen

Ein Anbieter informiert über ein neues Einzelspieler-Spiel, das er hinzugefügt hat.

Erlaubte Phasen

- Einrichtungsphase
- Normalbetriebsphase

Felder

Name	Datentyp	Beschreibung	Pflicht	Einschränkung
eventName	identifizier	Eindeutiger Bezeichner des Ereignisses	ja	Muss den Wert "SinglePlayerGameRegistration" enthalten.
eventTime	datetime	Zeitpunkt, wann der Anbieter das Spiel angelegt hat	ja	-
gameld	identifizier	Eindeutige Kennung des Spiels	ja	-
name	smallString	Name des Spiels	ja	-
singlePlayerGameCategory	identifizier	Kategorie des Spiels	ja	Siehe Tabelle unten für gültige Werte

Folgende Kategorien sind möglich:

Kategorie (*singlePlayerGameCategory*)

SlotGame

Scratchcard

Videopoker

Other

Beispiel

```
1 {
2   "eventName": "SinglePlayerGameRegistration",
3   "eventTime": "2020-11-20T09:42:29.725Z",
4   "gameId": "123456",
5   "name": "Red Mamba",
6   "singlePlayerGameCategory": "SlotGame"
7 }
```

Hinweise

- Bei der initialen Einrichtung des Safe-Servers sind alle bestehenden aktiven Spiele zu übermitteln.

8.8.3 Genehmigung Einzelspieler-Spiel

Ein Anbieter informiert über die behördliche Erlaubnis, ein Einzelspieler-Spiel anzubieten.

Nach § 22a Abs. 6 GlüStV 2021 muss die Spieldauer eines Einzelspieler-Spiels durchschnittlich mindestens 5 Sekunden betragen. Die Behörde kann in der Erlaubnis weitere Einzelheiten zur Berechnung des Durchschnitts festlegen.

Erlaubte Phasen

- Einrichtungsphase
- Normalbetriebsphase

Felder

Name	Datentyp	Beschreibung	Pflicht	Einschränkung
eventName	identifizier	Eindeutiger Bezeichner des Ereignisses	ja	Muss den Wert "SinglePlayerGamePermission" enthalten.
eventTime	datetime	Zeitpunkt der Aktivierung der Genehmigung, wenn <i>permittedFrom</i> nicht übermittelt wird.	ja	-

Name	Datentyp	Beschreibung	Pflicht	Einschränkung
gameld	identifizier	Eindeutige Kennung des Spiels	ja	Verweis auf ein existierendes Einzelspieler-Spiel
permittedFrom	datetime	Zeitpunkt, ab dem die Genehmigung aktiv ist	nein	Darf, wenn angegeben, nicht kleiner als <i>eventTime</i> sein.
permittedTo	datetime	Zeitpunkt, zu dem die Genehmigung inaktiv wird	nein	-
permittedStake	money	Gibt die maximale Einsatzhöhe an, welche die Behörde für ein Spiel dieses Typs freigegeben hat	ja	Muss positiv sein.
calculationRule	identifizier	Berechnungsvorschrift der durchschnittlichen Spielzeit	nein	Siehe Tabelle unten für gültige Werte.
areald	identifizier	Bereich des Anbieters, in dem dieses Spiel angeboten wird	ja	Verweis auf einen existierenden Bereich des Anbieters

Folgende Berechnungsvorschriften sind möglich:

Vorschrift (*calculationRule*)

Arithmetical

Median

Beispiel

```
1 {
2   "eventName": "SinglePlayerGamePermission",
3   "eventTime": "2020-11-20T09:42:29.725Z",
4   "gameId": "123456",
5   "permittedTo": "2021-11-20T09:42:29.725Z",
6   "permittedStake": 2,
7   "calculationRule": "Arithmetical",
8   "areaId": "14"
9 }
```

Hinweise

- *gameId* ist eine Referenz auf ein registriertes Einzelspieler-Spiel des Anbieters (siehe [Einzelspieler-Spiel hinzufügen](#)).
- *areaId* ist eine Referenz zu dem eingerichteten Bereich des Anbieters (siehe [Bereich hinzufügen](#)).
- Bei unbefristeten Genehmigungen ist *permittedTo* nicht zu übermitteln.
- Wird *permittedFrom* nicht übermittelt, so wird die Genehmigung als sofort aktiv interpretiert.
- Sobald *eventTime* eines beliebigen Ereignisses (ggf. auch des Genehmigungsereignisses selbst) auf oder nach *permittedFrom* fällt, wird die Genehmigung als aktiv interpretiert.
- Eine als aktiv interpretierte Genehmigung wird durch das Übermitteln einer weiteren Genehmigung ggf. früher oder sofort als beendet interpretiert.
- Eine noch nicht als aktiv interpretierte Genehmigung wird durch das Übermitteln einer weiteren Genehmigung hinfällig.
- Eine Genehmigung ist nur für ihren aktiven Zeitraum wirksam.
- Vor dem Start eines Spiels ist es erforderlich, dass für dieses Spiel eine Genehmigung angelegt wurde.

8.8.4 Gemeinschaftsspieler-Spiel hinzufügen

Ein Anbieter informiert über ein neues Gemeinschaftsspieler-Spiel, das er hinzugefügt hat.

Erlaubte Phasen

- Einrichtungsphase
- Normalbetriebsphase

Felder

Name	Datentyp	Beschreibung	Pflicht	Einschränkung
eventName	identifizier	Eindeutiger Bezeichner des Ereignisses	ja	Muss den Wert "MultiPlayerGameRegistration" enthalten.
eventTime	datetime	Zeitpunkt, wann der Anbieter das Spiel angelegt hat	ja	-
gameId	identifizier	Eindeutige Kennung des Spiels	ja	-
name	smallString	Name des Spiels	ja	-
variant	smallString	Die Spielvariante des Spiels	nein	-
multiPlayerGameCategoryidentifizier		Kategorie des Spiels	ja	Siehe Tabelle unten für gültige Werte

Folgende Kategorien sind möglich:

Kategorie (*multiPlayerGameCategory*)

Poker

Sonstige

Beispiel

```

1 {
2   "eventName": "MultiPlayerGameRegistration",
3   "eventTime": "2020-11-20T09:42:29.725Z",
4   "gameId": "789",
5   "name": "Poker",
6   "variant": "Texas hold 'em",
7   "multiPlayerGameCategory": "Poker"
8 }
```

Hinweise

- Bei der initialen Einrichtung des Safe-Servers sind alle bestehenden aktiven Spiele zu übermitteln.
- *variant* ist bei Spielen der *multiPlayerGameCategory* "Poker" verpflichtend anzugeben.

8.8.5 Genehmigung Gemeinschaftsspieler-Spiel

Ein Anbieter informiert über die behördliche Erlaubnis, ein Gemeinschaftsspieler-Spiel anzubieten.

Erlaubte Phasen

- Einrichtungsphase
- Normalbetriebsphase

Felder

Name	Datentyp	Beschreibung	Pflicht	Einschränkung
eventName	identifizier	Eindeutiger Bezeichner des Ereignisses	ja	Muss den Wert "MultiPlayerGamePermission" enthalten.
eventTime	datetime	Zeitpunkt der Aktivierung der Genehmigung, wenn <i>permittedFrom</i> nicht übermittelt wird.	ja	-
gameId	identifizier	Eindeutige Kennung des Spiels	ja	Verweis auf ein existierendes Gemeinschaftsspieler-Spiel
permittedFrom	datetime	Zeitpunkt, ab dem die Genehmigung aktiv ist	nein	Darf, wenn angegeben, nicht kleiner als <i>eventTime</i> sein.
permittedTo	datetime	Zeitpunkt, zu dem die Genehmigung inaktiv wird	nein	-

Name	Datentyp	Beschreibung	Pflicht	Einschränkung
permittedConcurrentMatches	int	Maximale Anzahl gleichzeitiger Matches	nein	Muss mindestens 1 sein
isConcurrencyRestricted-ToVariant	bool	Gibt an, ob die Beschränkung zum gleichzeitigen Spielen exklusiv für diese Spielvariante gilt	nein	-
limitMinStake	money	Höchstgrenzen für die Mindesteinsätze je Spielrunde (bspw. Blinds beim Poker)	nein	Muss positiv sein.
limitTotalAmount	money	Höchstgrenze für den Betrag, den ein Spieler zur Verfügung haben darf	nein	Muss positiv sein.
limitTournamentBuyIn	money	Höchstgrenze für den Betrag zur Teilnahme an einem Turnier	nein	Muss positiv sein.
areald	identifizier	Bereich des Anbieters, in dem dieses Spiel angeboten wird	ja	Verweis auf einen existierenden Bereich des Anbieters

Beispiel

```

1 {
2   "eventName": "MultiPlayerGamePermission",
3   "eventTime": "2020-11-20T09:42:29.725Z",
4   "gameId": "789",
5   "permittedTo": "2020-11-21T09:42:29.725Z",
6   "permittedConcurrentMatches": 4,
7   "isConcurrencyRestrictedToVariant": true,
8   "limitMinStake": 48.47,
9   "limitTotalAmount": 2000,
10  "limitTournamentBuyIn": 19.47,
11  "areaId": "5"
12 }
```

Hinweise

- *gameld* ist eine Referenz auf ein registriertes Gemeinschaftsspieler-Spiel des Anbieters (siehe [Gemeinschaftsspieler-Spiel hinzufügen](#)).
- *areald* ist eine Referenz zu dem eingerichteten Bereich des Anbieters (siehe [Bereich hinzufügen](#)).
- Bei unbefristeten Genehmigungen ist *permittedTo* nicht zu übermitteln.
- Wird *permittedFrom* nicht übermittelt, so wird die Genehmigung als sofort aktiv interpretiert.
- Sobald *eventTime* eines beliebigen Ereignisses (ggf. auch des Genehmigungsereignisses selbst) auf oder nach *permittedFrom* fällt, wird die Genehmigung als aktiv interpretiert.
- Eine als aktiv interpretierte Genehmigung wird durch das Übermitteln einer weiteren Genehmigung ggf. früher oder sofort als beendet interpretiert.
- Eine noch nicht als aktiv interpretierte Genehmigung wird durch das Übermitteln einer weiteren Genehmigung hinfällig.
- Eine Genehmigung ist nur für ihren aktiven Zeitraum wirksam.
- *permittedConcurrentMatches*, *isConcurrencyRestrictedToVariant*, *limitMinStake*, *limitTotalAmount* und *limitTournamentBuyIn* sind bei Pokerspielen verpflichtend anzugeben.
- Vor dem Start eines Spiels ist es erforderlich, dass für dieses Spiel eine Genehmigung angelegt wurde.

8.8.6 Genehmigung zur Festsetzung eines abweichenden Limits

Ein Anbieter informiert über die Genehmigung zur Limiterhöhung gemäß § 6c Abs. 1 S. 3-5 GlüStV 2021.

Erlaubte Phasen

- Einrichtungsphase
- Normalbetriebsphase

Felder

Name	Datentyp	Beschreibung	Pflicht	Einschränkung
eventName	identifizier	Eindeutiger Bezeichner des Ereignisses	ja	Muss den Wert "Special-LimitRegistration" enthalten.

Name	Datentyp	Beschreibung	Pflicht	Einschränkung
eventTime	datetime	Zeitpunkt der Aktivierung der Genehmigung, wenn <i>permittedFrom</i> nicht übermittelt wird.	ja	-
permittedFrom	datetime	Zeitpunkt, ab dem die Genehmigung aktiv ist	nein	-
permittedTo	datetime	Zeitpunkt, zu dem die Genehmigung inaktiv wird	nein	-
limitAmount	money	Maximale Limithöhe, die der Anbieter seinen Spielern erlauben darf	ja	Muss größer als 1000.00 € sein

Beispiel

```

1 {
2   "eventName": "SpecialLimitRegistration",
3   "eventTime": "2020-11-20T09:42:29.725Z",
4   "permittedTo": "2021-11-20T09:42:29.725Z",
5   "limitAmount": 1999.47
6 }
```

Hinweise

- Bei unbefristeten Genehmigungen ist *permittedTo* nicht zu übermitteln.
- Wird *permittedFrom* nicht übermittelt, so wird die Genehmigung sofort aktiv interpretiert.
- Sobald *eventTime* eines beliebigen Ereignisses (ggf. auch des Genehmigungsereignisses selbst) auf oder nach *permittedFrom* fällt, gilt die Genehmigung als aktiv.
- Eine als aktiv interpretierte Genehmigung wird durch das Übermitteln einer weiteren Genehmigung ggf. früher oder sofort als beendet interpretiert.
- Eine noch nicht als aktiv interpretierte Genehmigung wird durch das Übermitteln einer weiteren Genehmigung hinfällig.
- Eine Genehmigung ist nur für ihren aktiven Zeitraum wirksam.

8.8.7 Zahlungsdienst hinzufügen

Ein Anbieter gibt einen behördlich genehmigten Zahlungsdienst nach Anlage Zahlungsdienste, LUGAS an, mit der alle Spieler beim Anbieter Ein- und/oder Auszahlungen vornehmen können.

Erlaubte Phasen

- Einrichtungsphase
- Normalbetriebsphase

Felder

Name	Datentyp	Beschreibung	Pflicht	Einschränkung
eventName	identifizier	Eindeutiger Bezeichner des Ereignisses	ja	Muss den Wert "PaymentServiceRegistration" enthalten.
eventTime	datetime	Zeitpunkt, wann der Anbieter den Zahlungsdienst hinzugefügt hat	ja	-
paymentServiceNumber	identifizier	Die Zahlungsdienstenummer anhand von Anlage Zahlungsdienste, LUGAS	ja	-

Beispiel

```

1 {
2   "eventName": "PaymentServiceRegistration",
3   "eventTime": "2020-11-20T09:42:29.725Z",
4   "paymentServiceNumber": "2"
5 }
```

Hinweise

- Die Anlage Zahlungsdienste, LUGAS wird von der Behörde regelmäßig durch die gemeldeten Daten der Anbieter im Rahmen der Erlaubnisvergabe aktualisiert. Zahlungsdienste, welche zum Zeitpunkt der Registrierung noch nicht in dieser Anlage beschrieben wurden, müssen vom Anbieter nachträglich gemeldet werden, sobald dies möglich ist.

8.8.8 Zahlungsdienst entfernen

Ein Anbieter löscht einen hinterlegten Zahlungsdienst.

Erlaubte Phasen

- Normalbetriebsphase

Felder

Name	Datentyp	Beschreibung	Pflicht	Einschränkung
eventName	identifizier	Eindeutiger Bezeichner des Ereignisses	ja	Muss den Wert "PaymentServiceCancellation" enthalten.
eventTime	datetime	Zeitpunkt, wann der Anbieter den Zahlungsdienst gelöscht hat	ja	-
paymentServiceNumber	identifizier	Die Zahlungsdienstenummer anhand von Anlage Zahlungsdienste, LUGAS	ja	-

Beispiel

```
1 {  
2   "eventName": "PaymentServiceCancellation",  
3   "eventTime": "2020-11-20T09:42:29.725Z",  
4   "paymentServiceNumber": "2"  
5 }
```

Hinweise

- Wenn dieses Ereignis übermittelt wird, darf kein Zahlungsweg an einem Spieler mehr existieren, der auf den zu löschenden Zahlungsdienst aufbaut.

8.9 Sonderereignisse Einrichtungsphase

8.9.1 Kontostand Spieler

Der initiale Spielkontostand des Spielers wird durch den Anbieter bekannt gemacht.

Erlaubte Phasen

- Einrichtungsphase

Felder

Name	Datentyp	Beschreibung	Pflicht	Einschränkung
eventName	identifizier	Eindeutiger Bezeichner des Ereignisses	ja	Muss den Wert "TransactionFirstLoad" enthalten.
eventTime	datetime	Zeitpunkt der Wirksamkeit der Spielkontostandsänderung	ja	-
playerId	identifizier	Kennung des Spielers bei dem Anbieter	ja	Verweis auf einen existierenden Spieler
balance	money	Der Kontostand des Spielers	ja	-

Beispiel

```

1 {
2   "eventName": "TransactionFirstLoad",
3   "eventTime": "2020-11-20T09:42:29.725Z",
4   "playerId": "5emh/t5chEyN8S68yG1Rug==",
5   "balance": 212.47
6 }
```

Hinweise

- Dieses Ereignis dient als Ausgangswert des Spielerkontos. Jede Veränderung dieses Kontos muss durch ein [Transaktionsereignis](#) dokumentiert werden, welches erst in der *Normalbetriebsphase* möglich ist. Es ist daher sinnvoll, dieses Ereignis erst unmittelbar vor Eintritt in die *Normalbetriebsphase* zu übermitteln.

8.9.2 Ende Einrichtungsphase

Der Anbieter gibt bekannt, dass die *Einrichtungsphase* beendet ist und die *Normalbetriebsphase* starten kann.

Erlaubte Phasen

- Einrichtungsphase

Felder

Name	Datentyp	Beschreibung	Pflicht	Einschränkung
eventName	identifizier	Eindeutiger Bezeichner des Ereignisses	ja	Muss den Wert "CompletionFirstLoad" enthalten.
eventTime	datetime	Zeitpunkt des Endes der <i>Einrichtungsphase</i>	ja	-

Beispiel

```
1 {  
2   "eventName": "CompletionFirstLoad",  
3   "eventTime": "2020-11-20T09:42:29.725Z"  
4 }
```

Hinweise

- Nach Ablegen dieses Ereignisses ist ein Ablegen von Ereignissen, die ausschließlich in der *Einrichtungsphase* erlaubt sind, nicht mehr gültig.
- Ereignisse die sowohl in der *Einrichtungsphase* als auch der *Normalbetriebsphase* gültig sind, unterliegen fortan den dokumentierten Verhaltensweisen der *Normalbetriebsphase*.

9 Beispiele

In diesem Kapitel werden einige typische Szenarien skizziert. Anhand dieser Beispiele wird die jeweilige Reihenfolge der dadurch ausgelösten Ereignisse verdeutlicht.

9.1 Initiale Anbindung Safe-Server

Ein Anbieter bindet seinen Safe-Server initial an, er befindet sich somit in der *Einrichtungsphase*.

1. Der Anbieter legt alle seine Bereiche an. Ereignis: [Bereich hinzufügen](#)
2. Der Anbieter legt alle seine Einzelspieler-Spiele an. Ereignis: [Einzelspieler-Spiel hinzufügen](#)
3. Der Anbieter hinterlegt zu jedem seiner Einzelspieler-Spiele die entsprechende Genehmigung. Ereignis: [Genehmigung Einzelspieler-Spiel](#)
4. Der Anbieter legt alle seine Gemeinschaftsspieler-Spiele an. Ereignis: [Gemeinschaftsspieler-Spiel hinzufügen](#)
5. Der Anbieter hinterlegt zu jedem seiner Gemeinschaftsspieler-Spiele die entsprechende Genehmigung. Ereignis: [Genehmigung Gemeinschaftsspieler-Spiel](#)
6. Der Anbieter definiert, welche Zahlungsdienste er unterstützt. Ereignis: [Zahlungsdienst hinzufügen](#)
7. Der Anbieter übermittelt, welche Spieler er im System hat. Ereignis: [Registrieren](#)
8. Zu jedem Spieler wird der aktuelle Status des Spielers hinterlegt. Ereignis: [Status](#)
9. Ist der Spieler vom Anbieter gesperrt, so ist dies ebenfalls anzugeben. Ereignis: [Sperrung aktivieren](#)
10. Setzt der Anbieter bereits das Konzept von anbieterbezogenen Limits um und der Spieler macht davon gebrauch, so sind diese Limits anzugeben. Ereignis: [Limit setzen](#)
11. Zu jedem Spieler werden seine aktuellen Zahlungswege hinterlegt. Ereignis: [Zahlungsweg hinzufügen](#)
12. Zu jedem Spieler wird der aktuelle Kontostand übermittelt. Ereignis: [Kontostand Spieler](#)
13. Der Anbieter beendet die *Einrichtungsphase* und geht in die *Normalbetriebsphase* über. Ereignis: [Ende Einrichtungsphase](#)

9.2 Spieler erstellt ein Konto

Ein Spieler, der bisher noch nicht bei dem Anbieter gemeldet war, registriert sich beim Anbieter. In diesem Beispiel wird von der *Normalbetriebsphase* ausgegangen.

1. Der Anbieter speichert die personenbezogenen Daten des Spielers. Ereignis: [Registrieren](#)
2. Ggf. verpflichtet der Anbieter den Spieler bereits mit der Registrierung einen Zahlungsweg anzugeben. Ereignis: [Zahlungsweg hinzufügen](#)
3. Der Spieler muss den Zentraldateien bekannt gemacht werden und in dem Zuge ein anbieterübergreifendes Limit wählen oder ein bereits bestehendes anbieterübergreifendes Limit bestätigen. Ereignis: [Übermittlung Spielerdaten](#)
4. Der Status des Spielers ist nach der Registrierung im Status "VerificationPending" mit *intent* "Registration". Ereignis: [Status](#)

5. Der Spieler bestätigt die Kenntnisnahme der Hinweise zum vorläufigen Spiel. Ereignis: [Bestätigung vorläufiges Spiel](#)
6. Der Status des Spielers wechselt zu "VerificationPending" mit *intent* "TemporarilyAccepted". Ereignis: [Status](#)
7. Der Spieler veranlasst eine Einzahlung von 100€. Ereignis: [Einzahlung](#)
8. Dies wird zunächst an den Zentraldateien geprüft. Ereignis: [Einzahlung](#)
9. Im Erfolgsfall wird die Einzahlung dem Spielkonto gutgeschrieben. Ereignis: [Transaktion Spielkonto](#)
10. *Die danach vom Spieler durchgeführten Spiele nach der Anmeldung sind analog zum Beispiel [Durchschnittlicher Spielzyklus](#) ab Schritt 3.*
11. Nach Prüfung der Daten durch den Anbieter wechselt der Status des Spielers zu "Verified". Ereignis: [Status](#)

9.3 Durchschnittlicher Spielzyklus

Ein komplett freigeschalteter, nicht gesperrter Spieler meldet sich an, spielt ein paar Partien und meldet sich ab. Dieses Beispiel beleuchtet den idealtypischen Spielzyklus in der *Normalbetriebsphase* ohne Sonderfälle wie z. B., Sperren oder Quereffekte durch konkurrierende Anbieter.

1. Der Spieler meldet sich an. Ereignis: [Log-in](#)
2. Ggf. hat der Anbieter ihm seit mehr als 24 Stunden seine statistischen Umsatzdaten nicht übermittelt und muss dies jetzt nachholen. Ereignis: [Information statistische Daten](#)
3. Der Anbieter hat zu prüfen, dass der Spieler nicht global gesperrt ist. Ereignis: [Sperrprüfung](#)
4. Der Anbieter hat vor Teilnahme des Spielers an einem Spiel, den Spieler in den Zentraldateien auf aktiv zu setzen. Ereignis: [Aktivstatus](#)
5. Der Spieler wählt ein virtuelles AutomatenSpiel aus und bestätigt, dass er dieses spielen möchte (kostenpflichtig oder nicht spielt hierbei keine Rolle). Ereignis: [Spielererklärung virtuelles AutomatenSpiel](#)
6. Das Match beginnt. Ereignis: [Start Einzelspieler-Match](#)
7. Das Match endet. Ereignis: [Ende Einzelspieler-Match](#)
8. Im Falle eines kostenpflichtigen Spiels ist die Veränderung des Spielkontos anhand des gelaufenen Matches zu dokumentieren. Ereignis: [Transaktion Spielkonto](#)
9. Der Spieler spielt noch einige Runden weiter. Die Schritte von Spielererklärung bis Transaktion sind mit Ausnahme des Aktivstatus (da der Spieler für den Anbieter bereits aktiv ist) und der Anfrage an die Sperrdatei entsprechend zu wiederholen.
10. Der Spieler ist so ins Spiel vertieft, dass er die Zeit aus den Augen verliert. Es ist bereits eine Stunde vergangen. Der Anbieter hat den Spieler nun auf die vergangene Zeit hinzuweisen. Ereignis: [Realitycheck](#)

11. Der Spieler bestätigt den Hinweis des Anbieters. Ereignis: [Realitycheck Bestätigung](#)
12. Erschrocken darüber, wie schnell die Zeit vergangen ist, loggt sich der Spieler beim Anbieter aus. Ereignis: [Veranlassung Inaktivierung](#)
13. Durch die Veranlassung des Spielers ist der Anbieter verpflichtet, den Spieler in der Aktivitätsdatei auf inaktiv zu setzen. Ereignis: [Aktivstatus](#)

9.4 Ablauf der jährlichen Spielerdaten-Überprüfung durch den Anbieter

1. Der Anbieter initiiert die Überprüfung der Spielerdaten vor Ablauf eines Jahres seit der letzten Datenbestätigung nach § 6a Abs. 5 GlüStV 2021. Der Status des Spielers wechselt zu "VerificationPending mit intent"RegularCycle". Ereignis: [Status](#)
2. Der Spieler prüft seine Daten auf Richtigkeit und bestätigt oder korrigiert diese:
 - 2a. Der Spieler bestätigt seine Daten. Ereignis: [Datenbestätigung oder](#)
 - 2b. Ändert der Spieler seine Daten (Ereignis: [Datenänderung](#)), so erhält er weiterhin den Status "VerificationPending" allerdings mit dem intent "DataChanged". Ereignis: [Status](#).
3. Der Anbieter verifiziert den Spieler. Der Status des Spielers wechselt zu "Verified". Ereignis: [Status](#).

10 Anlagen

- Anlage Transliterationsvorschrift, LUGAS
- Anlage Zahlungsdienste, LUGAS