



API-referentie

HTTP, websocket en SDK voor Mini Engine 0.4: kamers, rollen, acties, accounts met e-mailbevestiging en het uploaden van eigen games.

Versie 0.4, oktober 2026. Routes onder `/api/v1`.

Inhoud

Zo lees je dit	3
Begrippen	4
Rollen en wat ze zien	4
Hostmodi en runtimes	4
Versies en action-ID's	5
Authenticatie	6
Routeoverzicht	8
HTTP: games	10
HTTP: kamers	12
Maken, deelnemen en lezen	12
Spelverloop	15
Kamerbeheer	16
HTTP: accounts	18
E-mailbevestiging	19
Ontwikkelaars	21
Een versie uploaden	23
Beheer	25
Versies en kamers	27
Websocket	29
Ingebouwde serveracties	31
SDK-client	32
Manifest en instellingen	35
Server en omgeving	37
Grenzen en rate limits	38
Foutformaat en foutcodes	40

Zo lees je dit

Deze referentie beschrijft de publieke interface van Mini Engine 0.4: de HTTP-routes, het websocketprotocol, de TypeScript-SDK, het gamepackage, het uploaden van eigen games en de serverinstellingen. Alles is afgeleid van de broncode en de contractdocumenten van versie 0.4. Nieuw ten opzichte van 0.3: e-mailadressen op accounts met een bevestigingslink, de mailgrens, de beheerroute om een adres te bevestigen, en dev-keys alleen voor accounts met een bevestigd adres.

Basisadres en versies

De engine luistert standaard op `127.0.0.1:3000` (instelbaar met `MINI_ENGINE_BIND`). Alle API-routes staan onder `/api/v1`; dat is de routeversie, los van de engineversie `0.4.0`. Alleen `/api/health` staat daarbuiten. Voorbeelden gebruiken relatieve paden; zet er je eigen basisadres voor.

Verzoeken en antwoorden

- Verzoeken met een body sturen JSON met `Content-Type: application/json`. Onbekende velden worden afgewezen.
- Antwoorden zijn JSON, behalve de HTML-rolpagina's, de gameassets, de releaseassets en routes die `204 No Content` geven.
- Elke fout heeft dezelfde vorm: `{"error":{"code":"...","message":"..."}}`. De `code` is stabiel; de `message` is Nederlands en bedoeld voor mensen.
- Kamercodes zijn niet hoofdlettergevoelig. `tafels` en `TAFELS` zijn dezelfde kamer.

Notatie

Notatie	Betekenis
<code>{code}</code>	Een padparameter, hier de kamercode van zes letters.
<code>HOST_SECRET</code>	Een plaatsvervanger voor een geheim token, zoals ook <code>PLAYER_SECRET</code> , <code>ACCOUNT_SECRET</code> en <code>ADMIN_SECRET</code> . Een dev-key staat er als <code>med_...</code> . Echte tokens staan nooit in dit document.
<code>...</code>	Ingekort voorbeeld. Het echte antwoord bevat meer velden of langere waarden.
Toegang	Welke sleutel de route verwacht: openbaar, een kamertoken van een bepaalde rol, een accounttoken, een dev-key of het admintoken.

Vertrouwen. Lokale gamepackages en geüploade games draaien als vertrouwde code op dezelfde origin als de engine. Ze kunnen in de browser de kamertoken van dat tabblad en een eventuele accountsessie lezen. Installeer alleen packages die je vertrouwt en geef dev-keys alleen aan mensen die je vertrouwt. Een aparte origin voor onbekende makers bestaat in 0.4 nog niet.

Begrippen

Eén kamer, één spel, een handvol apparaten. De server is het middelpunt: hij bewaart de stand, controleert wie wat mag en stuurt iedere rol alleen wat die rol mag zien.

Kamers

Een kamer hoort bij precies één game. Hij heeft een code van zes hoofdletters uit `ABCDEFGHJKLMNPQRSTUVWXYZ` (geen I en O, geen cijfers) en een status die de levenscyclus volgt:

```
lobby → running → finished → lobby
```

Nieuwe spelers komen in de lobby binnen, of tijdens `running` als late joining is toegestaan. Terug naar de lobby geeft een nieuwe `round_id`, zet readiness en spelstate terug en begint een nieuwe deduplicatieperiode. Een kamer verloopt na twaalf uur zonder activiteit; verbonden apparaten houden hem levend. Kamers, accounts en resultaten staan in SQLite en overleven een herstart. Aanwezigheid begint na een herstart als offline.

Ingebouwde en geüploade games

Ingebouwde games staan als map in `games/` op de server. Vanaf 0.3 kunnen uitgenodigde ontwikkelaars ook eigen browserhost-games uploaden met een dev-key. Elke upload is een vaste versie (een *release*); één versie per game is actief en die krijgen nieuwe kamers. Een kamer houdt de versie waarmee hij is gemaakt. Zie *Ontwikkelaars* en *Versies en kamers*.

Rollen en wat ze zien

Elk kamertoken hoort bij één rol. De rol bepaalt welke delen van de state in de snapshot staan.

Rol	Wie	Ziet
<code>host</code>	De maker van de kamer. Blijft <code>host</code> , ook als hij meespeelt.	<code>public_state</code> , <code>host_state</code> , alle <code>player_states</code> ; als hij meespeelt ook zijn eigen <code>private_state</code> .
<code>player</code>	Iemand die met een naam is binnengekomen.	<code>public_state</code> en alleen zijn eigen <code>private_state</code> .
<code>display</code>	Het gedeelde scherm, bijvoorbeeld de tv.	Alleen <code>public_state</code> . Nooit private state.

Namen, aanwezigheid, readiness, eventuele account-ID's, instellingen en de kamerconfig zijn voor alle rollen zichtbaar. De openbare info-route toont geen namen en geen spelstate.

Hostmodi en runtimes

De hostmodus ligt vast bij het maken van de kamer. De game bepaalt welke modi mogen.

Hostmodus	De maker	Spelverloop
<code>moderator</code>	Host zonder spelersstoel; <code>player_id</code> is <code>null</code> .	Een aparte spelleider bedient het spel en heeft zelf geen readiness.
<code>playing</code>	Host met spelersstoel.	De maker speelt mee en houdt zijn beheerrechten.
<code>hostless</code>	Host met spelersstoel.	Deelnemers bedienen het servergame; het tabblad van de maker hoeft niet verbonden te blijven.

De **runtime** zegt waar de spelregels draaien.

- `runtime:"server"` : de regels draaien in Rust op de server. In 0.4 geldt dat alleen voor de ingebouwde games Common Dice (alle drie de modi, standaard `hostless`) en Jeopardy (alleen `moderator`). Een manifestveld alleen installeert geen nieuwe serverruntime.
- `runtime:"host"` : een vertrouwde browserhost verwerkt acties en publiceert de volledige state. Alleen `moderator` is toegestaan en het hosttabblad moet verbonden blijven.

Versies en action-ID's

Elke kamer heeft een `version` . Gameplay, instellingen, config en lifecycle kunnen die verhogen. Aanwezigheid, binnenkomen, vertrekken en readiness geven ook een snapshot, maar verhogen de versie niet. Filter snapshots dus niet weg alleen omdat de versie gelijk bleef.

Wijzigingen die een gelijktijdige schrijver kunnen overschrijven, sturen `expected_version` mee: state publiceren, instellingen vervangen en de config aanpassen. Klopt de versie niet meer, dan volgt `409 version_conflict` . Lees de nieuwste snapshot en beoordeel de handeling opnieuw.

Acties op een servergame mogen een `action_id` meekrijgen: 1–64 tekens zonder controletekens. De server onthoudt per kamer de laatste 256 verwerkte ID's, gekoppeld aan actor en ronde. Een herhaling binnen dat venster slaagt zonder tweede effect en zonder versieverhoging. De SDK maakt standaard een unieke ID per actie en herhaalt opdrachten nooit vanzelf.

Authenticatie

Er zijn vier soorten sleutels die los van elkaar staan. Gebruik ze nooit door elkaar. Alle vier gaan als bearer in de header, nooit in de URL of een logregel.

```
Authorization: Bearer <token>
```

Sleutel	Krijg je van	Gebruik je voor
Kamertoken	Kamer maken (host), deelnemen (speler) of een displaytoken opvragen (scherm)	Alles binnen één kamer
Accounttoken	Registreren of inloggen	<code>/api/v1/accounts/...</code> , en optioneel bij kamer maken en deelnemen
Dev-key (<code>med_...</code>)	De eigenaar van de server	Games uploaden en beheren onder <code>/api/v1/dev</code>
Admintoken	Jezelf, als eigenaar (<code>MINI_ENGINE_ADMIN_TOKEN</code>)	Dev-keys en games beheren onder <code>/api/v1/admin</code>

De engine bewaart van elke sleutel alleen een hash.

Kamertokens

Maken en deelnemen geven een token van 64 tekens terug, met de bijbehorende rol. De server bewaart alleen een SHA-256-hash. Er is één token voor de host, één per speler en één voor het gedeelde scherm.

- Eén token heeft één actieve websocketverbinding. Een nieuwe verbinding met hetzelfde token vervangt de oude; die krijgt `session_replaced`.
- Vertrekken of verwijderd worden trekt het spelerstoken in. Een kamer verwijderen trekt alles in.
- De SDK bewaart kamertokens in `sessionStorage` onder `mini-engine:CODE:ROLE`.

Displaytokens

Het token voor het gedeelde scherm zit in het antwoord van `POST /api/v1/rooms` als `display_token`, tenzij het scherm is uitgeschakeld. Het scherm opent `/room/CODE/display#TOKEN`: de schil bewaart het fragment en haalt het weg voor het gameframe laadt, dat alleen `?room=CODE` krijgt.

De host kan een nieuw displaytoken opvragen; het vorige werkt dan niet meer. Het scherm uitschakelen trekt het token blijvend in. Na opnieuw inschakelen vraagt de host een nieuw token op; het oude komt niet terug.

Accounttokens

Accounts zijn optioneel; gastspel werkt zonder. Registreren en inloggen geven een willekeurig sessietoken van 64 tekens dat dertig dagen geldig is en een herstart overleeft. Elke login maakt een eigen sessie; uitloggen trekt alleen de gebruikte sessie in.

Stuur een accounttoken mee bij het maken of binnenkomen van een kamer om de spelersstoel aan het account te koppelen. De server controleert het token; een ongeldig token geeft `401`. De body kan geen `account_id` opgeven. Eén account bezet hoogstens één actieve stoel per kamer.

Dev-keys

Een dev-key hoort bij een gewoon account met een bevestigd e-mailadres en geeft toegang tot de routes onder `/api/v1/dev`. De eigenaar maakt hem met `POST /api/v1/admin/dev-tokens`. De key is `med_` gevolgd door 64 kleine hexadecimale tekens. Hij staat alleen in dat ene antwoord; de server bewaart een SHA-256-hash.

- Een key kan een verloopdatum hebben (1–3650 dagen) of onbeperkt geldig zijn.
- Intrekken werkt meteen. Games die al zijn geüpload blijven staan; de beheerder kan ze blokkeren.
- Een ongeldige, verlopen of ingetrokken key geeft `401 unauthorized`. Elk gebruik werkt `last_used_at` bij.

Admintoken

Het admintoken is een geheim dat de eigenaar zelf kiest en als `MINI_ENGINE_ADMIN_TOKEN` in de omgeving van de engine zet, bijvoorbeeld in `.env` naast `compose.yaml`. Het moet minstens 32 tekens lang zijn; met een korter token start de engine niet.

```
MINI_ENGINE_ADMIN_TOKEN=$(openssl rand -hex 32)
```

Zonder deze variabele staan alle routes onder `/api/v1/admin` uit en geven ze `404 admin_disabled`. Een verkeerd token geeft `401 unauthorized`. Bewaar het token als een wachtwoord: wie het heeft, kan dev-keys maken.

Origin

Verzoeken met een `Origin`-header worden alleen geaccepteerd als die origin gelijk is aan de host van het verzoek of in `MINI_ENGINE_ORIGINS` staat (kommagescheiden). Anders volgt `403 origin_forbidden`. Elk antwoord krijgt `no-store`, `nosniff`, `no-referrer` en een strikte content security policy die alleen `'self'` toestaat.

Routeoverzicht

Alle routes van Mini Engine 0.4 op één plek. Nieuw in 0.4 zijn `verify-email` en `email` onder `/api/v1/accounts` en de beheerroute die een adres met de hand bevestigt. De details staan per groep in de hoofdstukken hierna.

Methode	Pad	Toegang	Resultaat
GET	<code>/api/health</code>	openbaar	Status en engineversie
GET	<code>/api/v1/games</code>	openbaar	Alle speelbare games
GET	<code>/api/v1/games/{id}</code>	openbaar	Metadata van één game
GET	<code>/api/v1/games/{id}/{role}</code>	openbaar	HTML voor host, player of display; met <code>?room=</code> de versie van die kamer
GET	<code>/games/{id}/assets/{path}</code>	openbaar	Publieke assets van de actieve versie
GET	<code>/releases/{id}/{version}/assets/{path}</code>	openbaar	Assets van de actieve versie, of van een versie die een lopende kamer speelt
POST	<code>/api/v1/rooms</code>	gast of account	201 met hosttoken
POST	<code>/api/v1/rooms/{code}/join</code>	gast of account	201 met spelerstoken
GET	<code>/api/v1/rooms/{code}/info</code>	openbaar	Openbare kamerinfo
GET	<code>/api/v1/rooms/{code}</code>	kamertoken	Snapshot voor jouw rol
GET	<code>/api/v1/rooms/{code}/game-data</code>	host	Private data.json
POST	<code>/api/v1/rooms/{code}/ready</code>	deelnemer, lobby	Readiness zetten
POST	<code>/api/v1/rooms/{code}/actions</code>	speler; host bij servergame	Actie sturen
PUT	<code>/api/v1/rooms/{code}/state</code>	host, hostruntime	Volledige state vervangen
PUT	<code>/api/v1/rooms/{code}/settings</code>	host	Instellingen vervangen
POST	<code>/api/v1/rooms/{code}/lifecycle</code>	host	Status doorzetten
PUT	<code>/api/v1/rooms/{code}/config</code>	host, lobby	Kameropties aanpassen
POST	<code>/api/v1/rooms/{code}/display-token</code>	host, scherm aan	Nieuw displaytoken
POST	<code>/api/v1/rooms/{code}/leave</code>	speler	204, token ingetrokken
DELETE	<code>/api/v1/rooms/{code}/players/{player_id}</code>	host	204, speler verwijderd
DELETE	<code>/api/v1/rooms/{code}</code>	host	204, kamer verwijderd
GET	<code>/api/v1/rooms/{code}/ws</code>	kamertoken in eerste bericht	Websocket
POST	<code>/api/v1/accounts/register</code>	openbaar	201 met account en token; bevestigingsmail
POST	<code>/api/v1/accounts/login</code>	openbaar	200 met account en token
GET	<code>/api/v1/accounts/me</code>	account	Accountobject
POST	<code>/api/v1/accounts/logout</code>	account	204, sessie ingetrokken
GET	<code>/api/v1/accounts/stats</code>	account	Statistieken
POST	<code>/api/v1/accounts/verify-email</code>	account van de link	E-mailadres bevestigen

Methode	Pad	Toegang	Resultaat
PUT	/api/v1/accounts/email	account	Adres zetten of link opnieuw sturen
GET	/api/v1/dev/me	dev-key	Account en key
GET	/api/v1/dev/games	dev-key	Jouw games
GET	/api/v1/dev/games/{id}	dev-key	Eén game
PUT	/api/v1/dev/games/{id}	dev-key	Game aan- of uitzetten
DELETE	/api/v1/dev/games/{id}	dev-key	204, game met alle versies weg
POST	/api/v1/dev/games/{id}/versions	dev-key	201, nieuwe versie geüpload
POST	/api/v1/dev/games/{id}/versions/{version}/activate	dev-key	Versie actief maken
DELETE	/api/v1/dev/games/{id}/versions/{version}	dev-key	204, versie verwijderd
POST	/api/v1/admin/dev-tokens	admintoken	201 met nieuwe dev-key
GET	/api/v1/admin/dev-tokens	admintoken	Alle dev-keys, zonder geheimen
DELETE	/api/v1/admin/dev-tokens/{id}	admintoken	204, key ingetrokken
POST	/api/v1/admin/accounts/{username}/verify-email	admintoken	E-mailadres met de hand bevestigen
GET	/api/v1/admin/games	admintoken	Alle geüploade games met eigenaar
PUT	/api/v1/admin/games/{id}	admintoken	Game blokkeren of deblokkeren

Daarnaast serveert de engine de pagina's `/demo`, `/room/{code}/{role}` en de SDK onder `/sdk/`; zie *HTTP: games*.

HTTP: games

Openbare routes die de geïnstalleerde games en hun bestanden beschrijven. Geen token nodig.

GET /api/health

Toegang openbaar

Status en de gebouwde engineversie.

Antwoord 200

```
{"status": "ok", "version": "0.4.0"}
```

GET /api/v1/games

Toegang openbaar

Alle speelbare games met metadata en instellingenschema: de ingebouwde games en de aangezette uploads, elk met de metadata van de actieve versie. Uitgezette en geblokkeerde uploads staan er niet in. De velden staan bij *Manifest en instellingen*.

Antwoord 200

```
{
  "games": [
    {
      "id": "common-dice",
      "title": "Common Dice",
      "description": "Eén worp voor de hele tafel. ...",
      "version": "2.0.0",
      "min_players": 1,
      "max_players": 100,
      "has_display": true,
      "supports_late_join": true,
      "default_settings": {
        "dice_count": 5, "sides": 6, "roll_permission": "everyone", "theme": "ivory"
      },
      "settings_schema": {
        "type": "object", "properties": {...}, "required": [...], "additionalProperties": false
      },
      "runtime": "server",
      "host_modes": ["moderator", "playing", "hostless"],
      "default_host_mode": "hostless"
    }
  ]
}
```

GET /api/v1/games/{id}

Toegang openbaar

De metadata van één game, in dezelfde vorm als een item uit de lijst.

Fouten: **404** `game_not_found` (ook voor een uitgezette of geblokkeerde upload)

GET /api/v1/games/{id}/{role}

Toegang openbaar

De HTML-pagina van de game voor `host`, `player` of `display`, als `text/html`. De launcher laadt deze pagina in het gameframe.

Met `?room=CODE` serveert de engine de HTML van de versie waarmee die kamer is gemaakt. Bij een geüploade game verwijst hij daarin `/games/{id}/assets/` naar `/releases/{id}/{version}/assets/`. Zonder `?room=` krijg je de actieve versie van een speelbare game.

Fouten: `404 game_not_found` (onbekend, geblokkeerd, of uitgezet en zonder kamer), `404 game_page_not_found`, `404 release_not_found`

GET /games/{id}/assets/{path}

Toegang openbaar

Een publiek bestand uit de map `assets/` van de actieve versie van de game, met een MIME-type op basis van de extensie. Paden met lege delen, `.`, `..` of backslashes worden geweigerd. `data.json` is nooit een publiek asset.

Fouten: `404 game_not_found` (onbekend, uitgezet of geblokkeerd), `404 asset_not_found`

GET /releases/{id}/{version}/assets/{path}

Toegang openbaar

Een publiek bestand uit één bewaarde versie van een geüploade game, met dezelfde padregels. Alleen de actieve versie van een game die aan staat, of een versie die een lopende kamer nog speelt, wordt geserveerd. Zo laden de pagina's van een kamer hun assets uit hun eigen versie, ook als de ontwikkelaar de game uitzet.

Fouten: `404 game_not_found` (onbekend of geblokkeerd), `404 release_not_found`, `404 asset_not_found` (ook voor een ongebruikte, niet-actieve versie)

Pagina's en SDK buiten de API

Pad	Inhoud
<code>/</code>	Tijdelijke doorverwijzing naar <code>/demo</code> , de startpagina.
<code>/demo</code> , <code>/demo/join</code> , <code>/demo/account</code> , <code>/demo/new/{id}</code>	Werkbankpagina's om games te kiezen, kamers te maken en binnen te komen.
<code>/room/{code}/{role}</code>	De kamerschil voor <code>host</code> , <code>player</code> of <code>display</code> , met het gameframe erin.
<code>/sdk/mini-engine.js</code>	De door de engine geserveerde SDK-build als ES-module.
<code>/sdk/mini-engine.d.ts</code>	De bijbehorende TypeScript-declaraties, als platte tekst.

HTTP: kamers

Routes om kamers te maken, binnen te komen en het spel te bedienen. De meeste routes geven de bijgewerkte snapshot van jouw rol terug, zodat je meteen kunt tekenen.

Maken, deelnemen en lezen

POST /api/v1/rooms

Toegang gast, of een accounttoken om de stoel van de maker te koppelen

Maakt een kamer en geeft de hostcredentials terug. Alleen `game_id` is verplicht.

- `max_players` : standaard het maximum van de game, afgetopt op 24 en niet lager dan het minimum.
- `settings` : wordt samengevoegd met de standaardinstellingen van de game en daarna tegen het schema gecontroleerd.
- `config` : weggelaten velden worden de standaardmodus van de game, `display_enabled` volgens `has_display` en `allow_late_join:false` .
- `name` : naam van een meespelende maker; standaard `Spelleider` .

De kamer onthoudt de actieve versie van de game als `game_version` .

Verzoek

```
{
  "game_id": "common-dice",
  "name": "Dani",
  "max_players": 12,
  "settings": {"dice_count": 5, "sides": 6},
  "config": {"host_mode": "hostless", "display_enabled": true, "allow_late_join": false}
}
```

Antwoord 201

```
{
  "code": "TAFELS",
  "token": "HOST_SECRET",
  "role": "host",
  "player_id": "8c0e6c2a-...",
  "display_token": "DISPLAY_SECRET"
}
```

Een moderator krijgt `"player_id": null` . Bij een uitgeschakeld scherm ontbreekt `display_token` .

Fouten: `400 invalid_request` (spelersaantal, instellingen, modus of late joining die de game niet ondersteunt; browsergames vereisen `moderator`), `400 invalid_json` , `401 unauthorized` (ongeldig accounttoken), `404 game_not_found` (ook een uitgezette of geblokkeerde upload), `429 rate_limit` , `503 room_limit`

POST /api/v1/rooms/{code}/join

Toegang gast, of een accounttoken om de stoel te koppelen

Komt de kamer binnen als speler. Namen zijn 1–24 tekens zonder controletekens en uniek binnen de kamer, ongeacht hoofdletters.

Verzoek

```
{"name": "Sander"}
```

Antwoord 201

```
{"code": "TAFELS", "token": "PLAYER_SECRET", "role": "player", "player_id": "3f2c9d71-..."}
```

Fouten: 400 invalid_request (naam), 401 unauthorized, 404 room_not_found, 404 game_not_found (game geblokkeerd), 409 room_started, 409 room_full, 409 name_taken, 409 already_joined (dit account zit al in de kamer; hervat de bestaande sessie), 429 rate_limit

GET /api/v1/rooms/{code}/info

Toegang openbaar

Wat je moet weten voordat je binnenkomt. Geen namen, geen spelstate.

Antwoord 200

```
{
  "code": "TAFELS",
  "game_id": "common-dice",
  "game_version": "2.0.0",
  "status": "lobby",
  "player_count": 3,
  "max_players": 12,
  "config": {"host_mode": "hostless", "display_enabled": true, "allow_late_join": false},
  "runtime": "server"
}
```

Fouten: 404 room_not_found

GET /api/v1/rooms/{code}

Toegang elk kamertoken

De snapshot voor jouw rol. Dezelfde vorm komt over de websocket als `{"type": "snapshot", "room": ...}`. Hieronder de snapshot van een speler in Common Dice.

Antwoord 200

```
{
  "code": "TAFELS",
  "game_id": "common-dice",
  "game_version": "2.0.0",
  "status": "running",
  "version": 7,
  "max_players": 12,
  "settings": {
    "dice_count": 5, "sides": 6, "roll_permission": "everyone", "theme": "ivory"
  },
  "host_connected": true,
  "players": [
    {"id": "8c0e6c2a-...", "name": "Dani", "connected": true, "ready": true,
      "account_id": null},
    {"id": "3f2c9d71-...", "name": "Sander", "connected": true, "ready": true,
      "account_id": "b71e0c55-..."}
  ],
  "public_state": {
    "results": [5, 3, 6, 2, 6], "roller": "Sander", "roller_id": "3f2c9d71-...",
    "sequence": 4, "configuration": "5d6", "held": [], "rolled_at": 1792656000000,
    "history": [...]
  },
  "you": {"role": "player", "player_id": "3f2c9d71-..."},
  "runtime": "server",
  "config": {"host_mode": "hostless", "display_enabled": true, "allow_late_join": false},
  "round_id": "9b1d4f0e-...",
  "private_state": null
}
```

De host krijgt daarnaast `host_state` en `player_states`; een display krijgt geen `private_state`. `host_connected` is alleen aanwezigheid: serveracties hangen er niet van af. `game_version` is de versie waarmee de kamer is gemaakt; bij kamers van voor 0.3 is het `null`.

Fouten: `401 unauthorized`, `404 room_not_found`

GET /api/v1/rooms/{code}/game-data

Toegang hosttoken

De private `data.json` van de versie van de kamer, bijvoorbeeld vragen en antwoorden. Alleen de host kan dit lezen.

Fouten: `401 unauthorized`, `403 forbidden`, `404 room_not_found`, `404 game_not_found` (game geblokkeerd)

Spelverloop

POST /api/v1/rooms/{code}/ready

Toegang deelnemer met een stoel, alleen in de lobby

Zet jouw readiness. Starten kan pas als iedere deelnemer klaarstaat. Verhoogt de versie niet.

Verzoek

```
{"ready": true}
```

Antwoord 200

De bijgewerkte snapshot.

Fouten: **403 forbidden** (geen spelersstoel, zoals een moderator of display), **409 room_started**

POST /api/v1/rooms/{code}/actions

Toegang speler; bij een servergame ook de host als de regels dat toestaan

Stuurt een spelactie. De naam is 1–64 letters, cijfers, `_`, `.` of `-`; `data` is hoogstens 8 KiB en standaard `null`. Bij een servergame verwerkt Rust de actie en slaat het resultaat op voordat het zichtbaar wordt. Bij een hostruntime moet de host verbonden zijn; de actie gaat dan naar zijn wachtrij.

Verzoek

```
{"name": "roll", "data": {"held": [0, 2]}, "action_id": "c0a8e1f4-..."}
```

Antwoord 200

De bijgewerkte snapshot.

Fouten: **400 invalid_request** (naam, grootte of `action_id`), **403 forbidden** (display, of een rol die deze actie niet mag), **409 game_not_running**, **409 host_offline**, **409 invalid_game_action**, **409 buzzer_closed**

PUT /api/v1/rooms/{code}/state

Toegang hosttoken, alleen bij `runtime:"host"`

Vervangt de volledige state. Weggelaten `host_state` wordt `null`, weggelaten `player_states` wordt `{}`. Sleutels in `player_states` moeten bestaande speler-ID's zijn. De state wordt opgeslagen voordat de nieuwe snapshot uitgaat. Servergames weigeren deze route, ook voor de host.

Verzoek

```
{
  "expected_version": 4,
  "public_state": {"round": 1},
  "host_state": {"answer": "Venus"},
  "player_states": {"3f2c9d71-...": {"cards": [1, 2]}}
}
```

Fouten: **400 invalid_request** (onbekende speler), **403 forbidden**, **409 version_conflict**, **413 state_too_large** (meer dan 64 KiB)

PUT /api/v1/rooms/{code}/settings

Toegang hosttoken

Vervangt het hele instellingenobject. De server controleert het schema van de game en daarna de spelregels; bij Common Dice wist een ander aantal of andere zijden de worpen, een ander thema niet.

Verzoek

```
{
  "expected_version": 7,
  "settings": {
    "dice_count": 3, "sides": 20, "roll_permission": "everyone", "theme": "obsidian"
  }
}
```

Fouten: 400 invalid_request, 403 forbidden, 409 version_conflict, 409 teams_locked (Jeopardy)

POST /api/v1/rooms/{code}/lifecycle

Toegang hosttoken

Zet de status door: lobby → running → finished → lobby. Naar running vereist dat alle deelnemers klaarstaan; Jeopardy vraagt daarnaast een teamkeuze van iedere speler. Afronden van een servergame slaat de accountresultaten op in dezelfde transactie als de kamer.

Verzoek

```
{"status": "running"}
```

Fouten: 403 forbidden, 409 invalid_transition, 409 players_not_ready, 409 teams_required

Kamerbeheer

PUT /api/v1/rooms/{code}/config

Toegang hosttoken, alleen in de lobby

Past display_enabled en allow_late_join aan. De hostmodus ligt vast; onbekende velden worden afgewezen. Late joining kan alleen aan als de game supports_late_join:true heeft.

Verzoek

```
{"expected_version": 0, "config": {"display_enabled": false, "allow_late_join": true}}
```

Fouten: 400 invalid_request, 403 forbidden, 409 room_started, 409 version_conflict

POST /api/v1/rooms/{code}/display-token

Toegang hosttoken, scherm ingeschakeld

Maakt een nieuw displaytoken. Het vorige werkt meteen niet meer.

Antwoord 200

```
{"token": "DISPLAY_SECRET"}
```

Fouten: 403 forbidden (geen host, of scherm uitgeschakeld)

POST /api/v1/rooms/{code}/leave

Toegang spelerstoken

Verlaat de kamer en trekt het eigen token in. Antwoord `204` zonder body. De host kan deze route niet gebruiken.

Fouten: `401 unauthorized`, `403 forbidden`

DELETE /api/v1/rooms/{code}/players/{player_id}

Toegang hosttoken

Verwijdert een speler, zijn private state en zijn token. Antwoord `204`. De maker kan zijn eigen stoel niet verwijderen.

Fouten: `400 invalid_request` (eigen stoel), `403 forbidden`, `404 player_not_found`

DELETE /api/v1/rooms/{code}

Toegang hosttoken

Verwijdert de kamer en sluit alle verbindingen met `room_deleted`. Antwoord `204`.

Fouten: `401 unauthorized`, `403 forbidden`, `404 room_not_found`

HTTP: accounts

Een lokaal account dat alle games op deze engine delen, met een e-mailadres dat de speler via een link bevestigt. Geen externe inlog, geen wachtwoordherstel. Gastspel blijft altijd mogelijk.

Velden en regels

Veld	Regel
<code>username</code>	Wordt getrimd en naar kleine letters gezet; daarna 3–24 tekens uit a–z, 0–9 en <code>_</code> . Uniek.
<code>display_name</code>	Getrimd, 1–24 Unicode-tekens, geen controletekens.
<code>password</code>	Precies zoals ingevoerd, 10–128 UTF-8-bytes. Opgeslagen als Argon2id-hash.
<code>email</code>	Verplicht bij registreren. Getrimd en naar kleine letters; een geldig adres van hoogstens 254 tekens. Een bevestigd adres hoort bij één account; een onbevestigd adres reserveert niets.

Het accountobject is `{id, username, display_name, created_at, email, email_verified}`. `email` is `null` bij een account zonder adres, zoals een account van vóór 0.4.

POST /api/v1/accounts/register

Toegang openbaar

Verzoek

```
{"username": "sander", "password": "dobbelsteen-42", "display_name": "Sander",  
  "email": "sander@example.com"}
```

Antwoord 201

```
{  
  "account": {  
    "id": "b71e0c55-...", "username": "sander", "display_name": "Sander",  
    "created_at": 1792656000, "email": "sander@example.com", "email_verified": false  
  },  
  "token": "ACCOUNT_SECRET"  
}
```

`created_at` is in Unix-seconden. Het token wordt niet automatisch bewaard. De engine stuurt een bevestigingsmail; het account kan meteen inloggen en spelen.

Fouten: `400 invalid_request`, `409 username_taken`, `409 email_taken` (adres al bevestigd door een ander account), `429 rate_limit`

POST /api/v1/accounts/login

Toegang openbaar

Geeft dezelfde vorm als registreren, met status 200. Een onbekende naam en een fout wachtwoord geven dezelfde fout.

Verzoek

```
{"username": "sander", "password": "dobbelsteen-42"}
```

Fouten: `401 unauthorized`, `429 rate_limit`

GET /api/v1/accounts/me

Toegang accounttoken

Het accountobject: {id, username, display_name, created_at, email, email_verified}.

POST /api/v1/accounts/logout

Toegang accounttoken

Trekt deze sessie in. Antwoord 204. Kamertokens die al zijn uitgegeven blijven geldig.

GET /api/v1/accounts/stats

Toegang accounttoken

Statistieken uit afgeronde serverrondes, eenmaal per account en ronde. Er is geen route om zelf scores te posten; eigen browsergames tellen niet mee.

Antwoord 200

```
{
  "totals": {"games_played": 12, "wins": 3},
  "games": [
    {"game_id": "jeopardy", "games_played": 5, "wins": 3,
      "total_score": 4200, "best_score": 1400, "counters": {"correct": 17, "wrong": 6}},
    {"game_id": "common-dice", "games_played": 7, "wins": 0,
      "total_score": 0, "best_score": 0,
      "counters": {"rolls": 41, "dice_rolled": 180, "total": 652}}
  ],
  "recent": [
    {"round_id": "9b1d4f0e-...", "game_id": "jeopardy", "room_code": "TAFELS",
      "player_id": "3f2c9d71-...", "account_id": "b71e0c55-...", "display_name": "Sander",
      "score": 1400, "won": true, "counters": {"correct": 5, "wrong": 1},
      "finished_at": 1792659600}
  ]
}
```

recent bevat de laatste twintig resultaten. Tellers tellen alleen gehele getallen op; scores kunnen negatief zijn. Common Dice kent geen winnaars toe.

E-mailbevestiging

De bevestigingsmail bevat een link naar `MINI_ENGINE_PUBLIC_URL/demo/account?verify=<token>` en noemt het account (`@gebruikersnaam`). De accountpagina laat de speler eerst inloggen en stuurt het token daarna naar `verify-email`.

- Een link werkt alleen voor het ingelogde account waarvoor hij is verstuurd. Wie zich met andermans adres registreert, krijgt het niet bevestigd doordat de eigenaar van het adres op de link klikt.
- Linktokens zijn willekeurig (256 bits) en worden als SHA-256-hash bewaard. Een link verloopt na 48 uur, werkt één keer en vervalt zodra het adres verandert.
- Een onbevestigd adres reserveert niets. Het account dat als eerste bevestigt, houdt het; bij andere accounts met hetzelfde onbevestigde adres wordt `email` `null`.
- Hoogstens één bevestigingsmail per account per minuut. Zonder `MINI_ENGINE_SMTP_URL` komt de mail met de link alleen in de serverlog.
- Zonder bevestigd adres kun je inloggen en spelen; alleen een dev-key vraagt een bevestigd adres.

POST /api/v1/accounts/verify-email

Toegang accounttoken van het account waarvoor de link is

Verzoek

```
{"token": "TOKEN UIT DE LINK"}
```

Antwoord 200: {"account": Account} met email_verified: true.

Fouten: 400 invalid_token (onbekend, verlopen, gebruikt, voor een ander adres of account), 401 unauthorized, 409 email_taken, 429 rate_limit

PUT /api/v1/accounts/email

Toegang accounttoken

Voegt een adres toe, wijzigt het (weer onbevestigd) of stuurt de link opnieuw als het adres gelijk blijft. Er gaat altijd een nieuwe mail uit.

Verzoek

```
{"email": "sander@example.com"}
```

Antwoord 200

```
{
  "account": {
    "id": "b71e0c55-...", "username": "sander", "display_name": "Sander",
    "created_at": 1792656000, "email": "sander@example.com", "email_verified": false
  },
  "verification_sent": true
}
```

Fouten: 400 invalid_request, 401 unauthorized, 409 email_already_verified, 409 email_taken, 429 rate_limit (al een mail in de afgelopen minuut, of de mailgrens per IP-adres)

Alle accountroutes met een token geven 401 unauthorized bij een ontbrekend, ongeldig of verlopen token.

Ontwikkelaars

Routes onder `/api/v1/dev` waarmee een ontwikkelaar met een dev-key zijn eigen browserhost-games uploadt, versies beheert en games aan- of uitzet. De `me`-CLI gebruikt precies deze routes.

Stuur de dev-key mee als `Authorization: Bearer med_...`. Een ongeldige, verlopen of ingetrokken key geeft `401 unauthorized`. Je ziet en beheert alleen je eigen games; een game van iemand anders geeft `404 game_not_found`.

Objecten

Een **DevGame** beschrijft één geüploade game met al zijn bewaarde versies, nieuwste eerst. `title` komt uit het manifest van de actieve versie.

```
{
  "id": "quiz-night",
  "title": "Quiz Night",
  "enabled": true,
  "blocked": false,
  "active_version": "1.1.0",
  "created_at": 1792660000,
  "updated_at": 1792663600,
  "versions": [Release, ...]
}
```

Een **Release** is één bewaarde versie. `size` is in bytes, tijden zijn Unix-seconden. `active` is de versie die nieuwe kamers krijgen; `in_use` is `true` zolang een lopende kamer die versie speelt.

```
{
  "version": "1.1.0",
  "title": "Quiz Night",
  "content_hash": "9f2c...",
  "size": 48213,
  "file_count": 7,
  "created_at": 1792663600,
  "active": true,
  "in_use": false
}
```

Account en games

GET `/api/v1/dev/me`

Toegang dev-key

Het account achter de key en de key zelf, zonder geheim.

Antwoord 200

```
{
  "account": {"id": "b71e0c55-...", "username": "sander", "display_name": "Sander"},
  "token": {"id": "4d7a2c19-...", "label": "laptop", "expires_at": null}
}
```

GET /api/v1/dev/games

Toegang dev-key

Jouw games: {"games": [DevGame, ...]}, gesorteerd op id.

GET /api/v1/dev/games/{id}

Toegang dev-key

Eén DevGame.

Fouten: 404 game_not_found (onbekend, of niet van jou)

PUT /api/v1/dev/games/{id}

Toegang dev-key

Zet de game uit of aan en geeft de DevGame terug. Een uitgezette game verdwijnt uit GET /api/v1/games en /demo, er kunnen geen nieuwe kamers mee worden gemaakt en de HTML en assets zonder kamer geven 404. Lopende kamers spelen door via hun eigen versie.

Verzoek

```
{"enabled": false}
```

Fouten: 403 game_blocked (aanzetten terwijl de beheerder de game blokkeerde), 404 game_not_found

DELETE /api/v1/dev/games/{id}

Toegang dev-key

Verwijdert de game met alle versies. Antwoord 204. Het id is daarna weer vrij.

Fouten: 404 game_not_found, 409 game_in_use (een kamer speelt de game nog), 403 game_blocked (een geblokkeerde game kan alleen de beheerder vrijgeven)

Een versie uploaden

POST /api/v1/dev/games/{id}/versions

Toegang dev-key; body hoogstens 16 MiB

Uploadt een nieuwe versie: alle bestanden van het pakket als base64, met het pad als sleutel. De versie is het `version` uit `manifest.json`. Bestaat de game nog niet, dan wordt hij van jou en is deze eerste versie meteen actief. Latere versies worden alleen actief met `"activate": true`.

Verzoek

```
{
  "files": {
    "manifest.json": "eyJpZCI6InF1aXotbmhnaHQiLC4uLn0=",
    "host.html": "PCFkb2N0eXBliGh0bWw+...",
    "player.html": "PCFkb2N0eXBliGh0bWw+...",
    "display.html": "PCFkb2N0eXBliGh0bWw+...",
    "assets/app.js": "aW1wb3J0IHsgTWluaUVuZ2luZSB9..."
  },
  "activate": true
}
```

Antwoord 201

```
{
  "game": {"id": "quiz-night", "title": "Quiz Night", "enabled": true, "blocked": false,
    "active_version": "1.1.0", "created_at": 1792660000, "updated_at": 1792663600,
    "versions": [{"version": "1.1.0", ...}, {"version": "1.0.0", ...}],
    "release": {"version": "1.1.0", "title": "Quiz Night", "content_hash": "9f2c...",
      "size": 48213, "file_count": 7, "created_at": 1792663600,
      "active": true, "in_use": false}
}
```

Regels voor een upload

Een upload volgt dezelfde regels als een game op schijf: manifest, rolpagina's, instellingschema, hoogstens 4096 bestanden, 2 MiB per bestand en 10 MiB samen. Daarbij komt:

- Het `id` in het manifest is gelijk aan het id in de URL.
- `version` is 1–64 tekens uit `0–9 A–Z a–z . + _ -`, begint met een letter of cijfer en is nieuw voor deze game. Een versie overschrijven kan niet.
- Alleen `runtime:"host"` met een aparte spelleider (`host_modes: ["moderator"]`). Serverregels zitten in de engine.
- Paden zijn relatief, met `/`, hoogstens 255 tekens en 32 delen, zonder lege delen, `.`, `..`, backslashes of controletekens.
- Hoogstens 10 games per account en 10 bewaarde versies per game.
- Een upload is één databasetransactie: een mislukte upload laat niets achter. De engine bewaart een contenthash (SHA-256 over paden en inhoud) en legt de upload vast in het auditlog.

Fouten bij uploaden

Status en code	Wanneer
400 <code>invalid_request</code>	Ongeldig pakket: manifest, rolpagina's, instellingenschema, pad, base64, versienaam, een <code>runtime</code> anders dan <code>host</code> , of te groot.
400 <code>invalid_json</code>	Geen geldige JSON of onbekende velden in de body.
403 <code>forbidden</code>	Het id hoort bij een game van een andere ontwikkelaar.
409 <code>game_id_taken</code>	Het id hoort bij een ingebouwde game.
409 <code>version_exists</code>	Deze versie bestaat al voor deze game. Verhoog <code>version</code> in <code>manifest.json</code> .
409 <code>quota_games</code>	Je hebt al 10 games. Verwijder er eerst een.
409 <code>quota_releases</code>	Deze game heeft al 10 versies. Verwijder eerst oude versies.
413 <code>invalid_json</code>	De body is groter dan 16 MiB.

Versies beheren

POST `/api/v1/dev/games/{id}/versions/{version}/activate`

Toegang dev-key

Maakt een bewaarde versie actief, ook om terug te gaan naar een oudere. De engine controleert de versie eerst opnieuw, zodat een beschadigde versie nooit actief wordt. Geeft de DevGame terug. Lopende kamers houden hun eigen versie.

Fouten: `404 game_not_found`, `404 release_not_found`

DELETE `/api/v1/dev/games/{id}/versions/{version}`

Toegang dev-key

Verwijdert één bewaarde versie. Antwoord `204`.

Fouten: `404 game_not_found`, `404 release_not_found`, `409 release_active` (activeer eerst een andere versie), `409 release_in_use` (een lopende kamer speelt deze versie nog)

Beheer

Routes onder `/api/v1/admin` voor de eigenaar van de server: dev-keys maken en intrekken, e-mailadressen bevestigen, en geüploade games bekijken en blokkeren.

Beheer staat alleen aan als de engine draait met `MINI_ENGINE_ADMIN_TOKEN` (minstens 32 tekens); anders geeft elke route hieronder `404 admin_disabled`. Stuur het token mee als `Authorization: Bearer ADMIN_SECRET`. Een verkeerd of ontbrekend token geeft `401 unauthorized`.

Dev-keys

POST `/api/v1/admin/dev-tokens`

Toegang admintoken

Maakt een dev-key voor een bestaand account met een bevestigd e-mailadres. Alleen `username` is verplicht; de ontwikkelaar maakt eerst zelf een gewoon account op `/demo/account` en bevestigt het adres.

- `label`: 1–80 tekens zonder controletekens, standaard `cli`.
- `expires_in_days`: 1–3650. Weglaten betekent geen verlooptdatum.

Verzoek

```
{"username": "sander", "label": "laptop", "expires_in_days": 365}
```

Antwoord 201

```
{"id": "4d7a2c19-...", "token": "med_...", "username": "sander", "display_name": "Sander",  
  "label": "laptop", "created_at": 1792660000, "expires_at": 1824196000}
```

Het `token` staat alleen in dit antwoord. Geef het via een veilig kanaal door.

Fouten: `400 invalid_request` (label of geldigheid), `404 account_not_found`, `409 email_not_verified` (nog geen bevestigd adres)

GET `/api/v1/admin/dev-tokens`

Toegang admintoken

Alle dev-keys, nieuwste eerst, zonder geheimen. Ook ingetrokken keys staan erin, met `revoked_at`.

Antwoord 200

```
{"tokens": [{"id": "4d7a2c19-...", "username": "sander", "display_name": "Sander",  
  "label": "laptop", "created_at": 1792660000, "expires_at": null,  
  "last_used_at": 1792661234, "revoked_at": null}]}
```

DELETE `/api/v1/admin/dev-tokens/{id}`

Toegang admintoken

Trekt de key in. Antwoord `204`; de key werkt meteen niet meer. Games die al zijn geüpload blijven staan. Wil je ze weghalen, blokkeer ze dan ook.

Fouten: `404 token_not_found` (onbekend of al ingetrokken)

Accounts

POST /api/v1/admin/accounts/{username}/verify-email

Toegang admintoken

Bevestigt het e-mailadres van een account met de hand, voor wie de mail niet krijgt. Geen body. Een adres dat al bevestigd was, blijft bevestigd. De `me` -CLI doet dit met `me admin account verify <gebruikersnaam>`.

Antwoord 200

```
{"username": "sander", "email": "sander@example.com", "email_verified": true}
```

Fouten: `404 account_not_found`, `409 email_missing` (het account heeft geen adres)

Games

GET /api/v1/admin/games

Toegang admintoken

Alle geüploade games van alle ontwikkelaars, als DevGame met daarbij de eigenaar.

Antwoord 200

```
{
  "games": [
    {
      "id": "quiz-night", "title": "Quiz Night", "enabled": true, "blocked": false,
      "active_version": "1.1.0", "created_at": 1792660000, "updated_at": 1792663600,
      "versions": [...],
      "owner": {"id": "b71e0c55-...", "username": "sander", "display_name": "Sander"}}
  ]
}
```

PUT /api/v1/admin/games/{id}

Toegang admintoken

`false` blokkeert de game (`blocked: true`). Dat is een noodstop: de engine serveert er niets meer van, ook niet aan lopende kamers. HTML, assets, releaseassets en `game-data` geven 404, en meedoen en verbinden geven `404 game_not_found`. De ontwikkelaar kan de game niet zelf weer aanzetten. `true` heft de blokkade op en zet de game aan. Geeft de DevGame met `owner` terug.

Verzoek

```
{"enabled": false}
```

Fouten: `404 game_not_found` (geen geüploade game met dit id)

Auditlog

De tabel `audit_log` in de SQLite-database bewaart met tijd, maker (gebruikersnaam of `admin`) en onderwerp: `release.upload`, `release.activate`, `release.delete`, `game.enable`, `game.disable`, `game.delete`, `game.block`, `game.unblock`, `dev_token.create`, `dev_token.revoke` en `account.verify_email`. Er is geen route om het log te lezen.

Versies en kamers

Elke upload is een vaste versie. Een kamer speelt de versie die actief was toen hij werd gemaakt, ook als de ontwikkelaar daarna een nieuwe activeert, iemand de pagina ververst of later aansluit.

De versie van een kamer

Bij het maken van een kamer bewaart de engine de actieve versie als `game_version`. Het veld staat in `GET /api/v1/rooms/{code}/info` en in elke snapshot. Instellingen controleren, `game-data`, meedoen en verbinden gebruiken die versie. Bij kamers van voor 0.3 is `game_version` `null`.

Ingebouwde games krijgen het `version` uit hun manifest, maar worden bij een upgrade van de engine op hun plaats vervangen. Voor hen is `game_version` informatief: de engine serveert altijd de geïnstalleerde versie.

Vaste HTML en assets

De kamerschil laadt de rolpagina met `?room=CODE`. De engine zoekt de versie van die kamer op en serveert de HTML van die versie. Bij een geüploade game vervangt hij in die HTML elk voorkomen van `/games/{id}/assets/` door het pad van de vaste versie:

```
GET /api/v1/games/quiz-night/player?room=TAFELS
/games/quiz-night/assets/app.js → /releases/quiz-night/1.0.0/assets/app.js
```

Hoort de kamer bij een andere game, of bestaat hij niet, dan negeert de engine `?room=` en serveert hij de actieve versie.

Relatieve imports

De engine past alleen de HTML aan, niet je JavaScript. Schrijf imports tussen je eigen modules daarom relatief, zodat ze uit dezelfde versie komen:

```
import { score } from './logic.js'; // goed: dezelfde versie
import { score } from '/games/quiz-night/assets/logic.js'; // fout: altijd de actieve versie
```

Aan, uit en geblokkeerd

Toestand	Catalogus en nieuwe kamers	Zonder kamer	Lopende kamers
Aan	In <code>GET /api/v1/games /demo</code> en	HTML en assets van de actieve versie	Spelen hun eigen versie
Uitgezet door de ontwikkelaar	Weg; nieuwe kamers geven <code>404 game_not_found</code>	HTML zonder <code>?room=</code> en <code>/games/{id}/assets/...</code> geven <code>404 game_not_found</code>	Spelen door via <code>?room=</code> en <code>/releases/{id}/{version}/...</code>
Geblokkeerd door de beheerder	Weg; nieuwe kamers geven <code>404 game_not_found</code>	<code>404 game_not_found</code>	Stoppen: HTML, assets, releaseassets en <code>game-data</code> geven 404, meedoen geeft <code>404 game_not_found</code>

Opruimen

Een game bewaart hoogstens 10 versies. Verwijderen kan niet bij de actieve versie (`409 release_active`) en niet bij een versie die een lopende kamer nog speelt (`409 release_in_use`). Een hele game verwijderen kan pas als geen enkele kamer hem meer speelt (`409 game_in_use`). Een kamer telt mee tot hij verloopt, twaalf uur na zijn

laatste activiteit.

Websocket

Eén verbinding per kamertoken. De server stuurt na elke wijziging een verse snapshot voor jouw rol; jij stuurt opdrachten en krijgt per opdracht een bevestiging of een fout.

Verbinden

```
GET /api/v1/rooms/{code}/ws (upgrade naar websocket)
```

Stuur binnen vijf seconden als eerste tekstframe het token. Daarna volgt de eerste snapshot.

```
→ {"type": "auth", "token": "PLAYER_SECRET"}  
← {"type": "snapshot", "room": {...}}
```

Alleen tekstframes met JSON; binaire frames geven een fout. Na vijf seconden zonder auth volgt `auth_timeout` en sluit de server. Bestaat de kamer of de game niet (meer), of is de game geblokkeerd, dan faalt de upgrade met een gewone HTTP-fout.

Van client naar server

Elk bericht heeft een `type` en mag een `request_id` van hoogstens 64 tekens meesturen. De server zet die terug in de `ack` of `error`.

type	Velden	Gelijk aan
action	name, data, optioneel action_id	POST .../actions
ready	ready	POST .../ready
configure	expected_version, gedeeltelijke config	PUT .../config
set_state	expected_version, public_state, optioneel host_state, player_states	PUT .../state
settings	expected_version, volledig settings	PUT .../settings
lifecycle	status	POST .../lifecycle
ping	geen	alleen een ack

Van server naar client

type	Wanneer	Vorm
snapshot	Na auth, na elke wijziging in de kamer en na een eigen mutatie.	<code>{"type": "snapshot", "room": {...}}</code>
ack	Een opdracht is gelukt.	<code>{"type": "ack", "request_id": "1"}</code>
error	Een opdracht is mislukt, of er ging iets mis buiten een opdracht.	<code>{"type": "error", "request_id": "1", "error": {"code": "...", "message": "..."} }</code>
action	Alleen voor de host van een hostruntime: een speler stuurde een actie.	<code>{"type": "action", "player_id": "...", "name": "vote", "data": {...}, "request_id": "3"}</code>
closed	De server sluit de verbinding.	<code>{"type": "closed", "reason": "room_deleted"}</code>

Na een mutatie komt eerst de bijgewerkte snapshot en dan de `ack`; de `ack` bevat zelf geen state. Bij een hostruntime bevestigt de `ack` van een speleractie alleen dat de actie in de wachtrij van de host staat, niet dat het spel haar heeft verwerkt.

Een ronde in berichten

```
→ {"type": "ready", "ready": true, "request_id": "1"}
← {"type": "snapshot", "room": {"status": "lobby", "version": 3, ...}}
← {"type": "ack", "request_id": "1"}
→ {"type": "action", "name": "roll", "data": {"held": []}, "action_id": "c0a8e1f4-...", "request_id": "2"}
← {"type": "snapshot", "room": {"status": "running", "version": 5, ...}}
← {"type": "ack", "request_id": "2"}
```

Sluitredenen

reason	Betekenis	Wat de SDK doet
<code>session_replaced</code>	Hetzelfde token verbond opnieuw, bijvoorbeeld in een ander tabblad.	Stopt; credentials blijven bewaard voor expliciet hervatten.
<code>session_revoked</code>	Het token is ingetrokken: vertrokken, verwijderd of het scherm is uitgezet.	Stopt.
<code>room_deleted</code>	De host heeft de kamer verwijderd.	Stopt.
<code>room_expired</code>	Twaalf uur geen activiteit.	Stopt.
<code>host_lagged</code>	De host liep te ver achter op de gebeurtenissen.	Verbindt opnieuw met backoff.
<code>heartbeat_timeout</code>	45 seconden geen verkeer van de client.	Verbindt opnieuw met backoff.
<code>rate_limit</code>	Te veel berichten per seconde.	Verbindt opnieuw na een extra pauze.
<code>server_shutdown</code>	De engine stopt.	Verbindt opnieuw met backoff.

De server stuurt elke vijftien seconden een websocket-ping. Spelers en schermen die achterlopen krijgen gewoon een verse snapshot. Onzekere opdrachten worden nooit automatisch herhaald.

Ingebouwde serveracties

Common Dice en Jeopardy draaien in Rust. Hun pagina's sturen alleen acties en tekenen snapshots. De server kiest de uitkomst; animaties laten die alleen zien.

Common Dice

Instellingen: `dice_count` 1–20, `sides` 4, 6, 8, 10, 12 of 20, `roll_permission` `everyone` of `owner`, `theme` `ivory`, `obsidian` of `candy`. Standaard `{dice_count:5, sides:6, roll_permission:"everyone", theme:"ivory"}`.

Actie	Wie	Wat er gebeurt
<code>roll {held?: [0, 2]}</code>	Iedere deelnemer; bij <code>owner</code> alleen de host.	Alleen tijdens <code>running</code> . Gooit de vrije stenen opnieuw. <code>held</code> verwijst naar de huidige worp; minstens één steen blijft vrij.
<code>clear_history {}</code>	Host	Wist de zichtbare worpgeschiedenis.
<code>start {}, finish {}</code>	In <code>hostless</code> iedere deelnemer, anders de host.	Start of rondt de ronde af.

`roll_permission:"owner"` gaat niet samen met `hostless`. De publieke state bevat `results`, `roller`, `roller_id`, `sequence`, `configuration` (bijvoorbeeld `"5d6"`), `held`, `rolled_at` in epochmilliseconden en hoogstens twintig items in `history`.

Jeopardy

Instelling: `team_count` 2–6, standaard 3. Alleen `moderator`; geen late joining. `team_count` kan alleen in de lobby veranderen, zolang nog niemand een team koos.

Actie	Wie	Wat er gebeurt
<code>choose_team {team_id}</code>	Speler, in de lobby	Kiest een team. Starten vereist dat iedere speler er een heeft.
<code>buzz {}</code>	Speler	De eerste geldige buzzer bij een open vraag wint. Anders <code>409 buzzer_closed</code> .
<code>select_clue {clue_id}</code>	Host	Opent een vraag. Het antwoord staat in <code>host_state.active_answer</code> .
<code>judge {correct}</code>	Host	Kent de score één keer toe; bij fout gaat de vraag open voor de andere teams.
<code>reveal, reopen, close_clue, finish</code>	Host	Onthult het antwoord, opent opnieuw, sluit de vraag of rondt het spel af.

Lege teams kunnen niet winnen. Ongeldige zetten geven `409 invalid_game_action`.

SDK-client

`@casb02/mini-engine-sdk` versie 0.3.1: een getypeerde client voor browsers en Node.js 22 of nieuwer. De broncode staat openbaar op GitHub (`Casb02/mini-engine-sdk`, MIT); het pakket is niet op npm gepubliceerd.

Nieuw in 0.3.1 voor engine 0.4: `register` vraagt een `email`, de methoden `verifyEmail` en `setEmail`, en `Account` heeft `email` en `email_verified`. De engine serveert deze versie onder `/sdk/`. De SDK heeft geen methoden voor `/api/v1/dev` en `/api/v1/admin`; daarvoor is de `me` -CLI.

Laden

In een gamepagina importeer je de build die de engine zelf serveert. Buiten de engine installeer je de tarball uit `npm pack`.

```
import { ApiClient, MiniEngine, loadSession } from '/sdk/mini-engine.js';

const code = new URL(location.href).searchParams.get('room') || '';
const credentials = loadSession(code, 'player');
if (!credentials) throw new Error('Open deze pagina via de kamer.');

const client = new MiniEngine(credentials).connect();
client.on('snapshot', room => render(room));
client.on('error', error => showError(error.message));
await client.waitForSnapshot();
await client.ready(true);
// Zodra het spel loopt:
await client.action('roll', { held: [] });
```

ApiClient

`new ApiClient({ baseUrl?, accountToken?, fetch? })`. Zonder `baseUrl` gebruikt hij de huidige origin. Methoden geven een Promise en gooien een `MiniEngineError` bij een HTTP-fout.

Methode	Route	Geeft
<code>getGames()</code>	<code>GET /games</code>	<code>{ games: Manifest[] }</code>
<code>getGame(id)</code>	<code>GET /games/{id}</code>	<code>Manifest</code>
<code>createRoom(options)</code>	<code>POST /rooms</code>	<code>RoomCredentials</code>
<code>joinRoom(code, name)</code>	<code>POST /rooms/{code}/join</code>	<code>RoomCredentials</code>
<code>roomInfo(code)</code>	<code>GET /rooms/{code}/info</code>	openbare kamerinfo
<code>getRoom(code, token)</code>	<code>GET /rooms/{code}</code>	<code>Snapshot</code>
<code>gameData(code, token)</code>	<code>GET /rooms/{code}/game-data</code>	de private speldata
<code>displayToken(code, token)</code>	<code>POST /rooms/{code}/display-token</code>	<code>{ token }</code>
<code>register(input)</code>	<code>POST /accounts/register</code>	<code>AccountSession</code> ; input met <code>email</code>
<code>verifyEmail(token)</code>	<code>POST /accounts/verify-email</code>	<code>{ account }</code> ; met <code>accountToken</code>
<code>setEmail(email)</code>	<code>PUT /accounts/email</code>	<code>{ account, verification_sent }</code>
<code>login(input)</code>	<code>POST /accounts/login</code>	<code>AccountSession</code>
<code>me()</code>	<code>GET /accounts/me</code>	<code>Account</code>

Methode	Route	Geeft
<code>stats()</code>	GET <code>/accounts/stats</code>	<code>AccountStats</code>
<code>logout()</code>	POST <code>/accounts/logout</code>	<code>null</code> ; wist <code>accountToken</code>

`createRoom` en `joinRoom` sturen het `accountToken` mee als dat is ingesteld. `register` en `login` bewaren het token niet; zet het zelf in `accountToken`. `verifyEmail` werkt alleen met het `accountToken` van het account waarvoor de link is. Losse helpers voor de huidige origin: `getGames()`, `createRoom(options)`, `joinRoom(code, name)` en `http(path, options)`, die JSON ophaalt met een optioneel `token`.

MiniEngine

`new MiniEngine({ code, token, baseUrl?, reconnectDelay?, requestTimeout?, WebSocket? })`. Een `RoomCredentials`-object past direct. Standaard: `reconnectDelay` 500 ms, verdubbeld tot hoogstens 15 seconden met wat spreiding; `requestTimeout` 10 seconden.

Lid	Wat het doet
<code>connect()</code>	Opent de verbinding en authenticceert. Geeft de client terug.
<code>resume()</code>	Verbindt opnieuw na een stop, bijvoorbeeld na <code>session_replaced</code> . De laatste snapshot blijft staan tot er een nieuwe is.
<code>close()</code>	Stopt, wijst open opdrachten af en verbindt niet opnieuw.
<code>on(type, fn)</code> , <code>once(type, fn)</code>	Luistert naar een event. Geeft een functie terug die het luisteren stopt.
<code>waitForSnapshot(predicate?, { timeout?, signal? })</code>	Wacht op een snapshot die aan de voorwaarde voldoet.
<code>action(name, data?, { actionId? })</code>	Stuurt een actie, standaard met een nieuwe unieke <code>action_id</code> .
<code>ready(ready = true)</code>	Zet readiness.
<code>configure(config)</code>	Past <code>display_enabled</code> of <code>allow_late_join</code> aan met de huidige versie.
<code>setSettings(settings)</code>	Vervangt alle instellingen met de huidige versie.
<code>setState(update)</code>	Publiceert de volledige state; alleen voor een hostruntime.
<code>lifecycle(status)</code>	Zet de status door.
<code>request(type, payload)</code>	Stuurt een willekeurig bericht met een eigen <code>request_id</code> en wacht op de <code>ack</code> .
<code>room</code> , <code>connected</code> , <code>code</code> , <code>token</code>	De laatste snapshot, de verbindingstatus en de credentials.

Alle opdrachten geven een Promise die oplost met de `ack`. Ze worden afgewezen bij een `error`, bij een time-out (`request_timeout`), bij verbreken (`disconnected`) of als er nog geen verbinding is (`not_connected`).

Events

Event	Waarde
<code>snapshot</code>	De nieuwe <code>Snapshot</code> . De eerste snapshot markeert de client als verbonden.
<code>action</code>	Een <code>ActionEnvelope</code> ; alleen voor de host van een hostruntime.
<code>status</code>	<code>connecting</code> , <code>connected</code> , <code>reconnecting</code> of <code>disconnected</code> .
<code>error</code>	Een <code>MiniEngineError</code> met <code>code</code> en <code>message</code> .
<code>closed</code>	De sluitreden als de server definitief sloot.

Sessies

Functie	Wat het doet
<code>sessionKey(code, role)</code>	Geeft <code>mini-engine:CODE:ROLE</code> .
<code>saveSession(credentials, storage?)</code>	Bewaart credentials, standaard in <code>sessionStorage</code> .
<code>loadSession(code, role, storage?)</code>	Leest ze terug, of <code>null</code> als ze ontbreken of niet kloppen.
<code>removeSession(code, role, storage?)</code>	Verwijdert ze.

Fouten en typen

`MiniEngineError` breidt `Error` uit met `code` en, bij HTTP, `status`. Bij serverfouten is `code` de foutcode van de engine; daarnaast bestaan clientcodes als `not_connected`, `request_timeout`, `snapshot_timeout`, `disconnected`, `closed` en `missing_base_url`.

De belangrijkste typen: `Json`, `Role`, `HostMode`, `Lifecycle`, `ConnectionStatus`, `RoomConfig`, `Player`, `Snapshot<PublicState, Settings, PrivateState>`, `Manifest`, `SettingsProperty`, `RoomCredentials`, `CreateRoomOptions`, `Account`, `AccountSession`, `AccountStats`, `Ack`, `ActionEnvelope`, `StateUpdate` en `ActionPayloads` voor de ingebouwde acties. `MiniEngine<P, S, V>` geeft de typen van publieke state, instellingen en private state door aan `room` en de events.

Een eigen browserhost

Bij `runtime:"host"` verwerkt de hostpagina de acties en publiceert de hele volgende state. Valideer speler, payload, status en spelregels; vertrouw nooit een identiteit of score uit de clientdata. Zet alle mutaties in één wachtrij en wacht op snapshot en ack voordat de volgende een versie gebruikt.

```
client.on('action', async ({ player_id, name, data }) => {
  if (name !== 'vote' || client.room?.status !== 'running') return;
  const next = tally(client.room.public_state, player_id, data);
  await client.setState({
    expected_version: client.room.version,
    public_state: next,
  });
});
```

De scaffolder maakt een werkend voorbeeld: `create-mini-game <slug> [--directory pad]`.

Manifest en instellingen

Een game is een map in `games/` met een manifest, drie rolpagina's en publieke assets. De mapnaam is de game-ID. Een upload heeft dezelfde opbouw; daar is het id in de URL de game-ID.

```
games/my-game/  
  manifest.json  
  host.html  
  player.html  
  display.html  
  assets/  
  data.json
```

verplicht bij `has_display: true`
publiek, onder `/games/my-game/assets/...`
optioneel en privé; alleen de host leest dit

Manifestvelden

Veld	Type	Betekenis
<code>id</code>	string	Gelijk aan de mapnaam.
<code>title</code> , <code>description</code> , <code>version</code>	string	Naam, korte omschrijving en eigen versie van de game. Bij een upload is <code>version</code> de naam van de release.
<code>min_players</code> , <code>max_players</code>	getal	Grenzen voor <code>max_players</code> van een kamer.
<code>has_display</code>	boolean	Of er een gedeeld scherm is; dan is <code>display.html</code> verplicht.
<code>default_settings</code>	object	Standaardinstellingen; moeten aan het schema voldoen.
<code>settings_schema</code>	object	Het instellingenschema, zie hieronder.
<code>runtime</code>	<code>host</code> <code>server</code>	Standaard <code>host</code> . Een upload moet <code>host</code> zijn.
<code>host_modes</code>	lijst	Toegestane hostmodi. Standaard <code>["moderator"]</code> .
<code>default_host_mode</code>	string	Standaard <code>moderator</code> .
<code>supports_late_join</code>	boolean	Standaard <code>true</code> voor oudere manifests. Kies het expliciet voor nieuwe games.

```
{
  "id": "my-game",
  "title": "My Game",
  "description": "Een spel voor vrienden.",
  "version": "1.0.0",
  "runtime": "host",
  "host_modes": ["moderator"],
  "default_host_mode": "moderator",
  "supports_late_join": false,
  "min_players": 1,
  "max_players": 12,
  "has_display": true,
  "default_settings": {"rounds": 5, "mode": "normal"},
  "settings_schema": {
    "type": "object",
    "properties": {
      "rounds": {"type": "integer", "minimum": 1, "maximum": 20, "title": "Rondes"},
      "mode": {"type": "string", "enum": ["normal", "quick"]}
    },
    "required": ["rounds", "mode"],
    "additionalProperties": false
  }
}
```

Instellingenschema

Een klein deel van JSON Schema. De catalogus controleert schema en standaardwaarden bij het opstarten en weigert alles wat hij niet kent.

- Het schema is een `object` met `properties`, optioneel `required` en verplicht `additionalProperties:false`. Annotaties `title`, `description` en `default` mogen.
- Een property heeft `type` `integer`, `number`, `string` of `boolean`, en optioneel `enum`, `minimum`, `maximum`, `maxLength`, `title`, `description` en `default`.
- Arrays, geneste objecten en `$ref` worden niet ondersteund.

Gebruik absolute paden in de HTML, bijvoorbeeld `/games/my-game/assets/app.js` en `/sdk/mini-engine.js`. Bij een geüploade game zet de engine `/games/my-game/assets/` in de HTML van een kamer om naar de vaste versie. Imports tussen je eigen JS-modules schrijf je relatief (`./logic.js`), want daar verandert de engine niets aan.

Server en omgeving

De engine leest zijn instellingen uit omgevingsvariabelen. Paden zijn relatief aan de werkmap; buiten Docker Compose wordt `.env` niet automatisch gelezen.

Variabele	Standaard	Betekenis
<code>MINI_ENGINE_BIND</code>	<code>127.0.0.1:3000</code>	Adres en poort
<code>MINI_ENGINE_DB</code>	<code>data/mini-engine.sqlite</code>	SQLite voor kamers, accounts, uploads en resultaten
<code>MINI_ENGINE_GAMES</code> , <code>MINI_ENGINE_WEB</code>	<code>games</code> , <code>web</code>	Ingebouwde games; demo en SDK-bundle
<code>MINI_ENGINE_ORIGINS</code>	leeg	Extra toegestane browser-origins, kommagescheiden
<code>MINI_ENGINE_ADMIN_TOKEN</code>	leeg	Zet <code>/api/v1/admin</code> aan; minstens 32 tekens
<code>MINI_ENGINE_PUBLIC_URL</code>	<code>http://localhost:3000</code>	Basis voor links in mails, bijvoorbeeld <code>https://miniengine.nl</code>
<code>MINI_ENGINE_SMTP_URL</code>	leeg	SMTP voor bevestigingsmails, zoals <code>smtps://gebruiker:wachtwoord@smtp.example.com:465</code> . Zonder: de mail komt in de log.
<code>MINI_ENGINE_MAIL_FROM</code>	Mini Engine <noreply@localhost>	Afzender van de mails
<code>MINI_ENGINE_TRUST_PROXY</code>	leeg	Achter een reverse proxy: <code>x-forwarded-for</code> (het laatste adres dat de proxy toevoegde) of een headernaam zoals <code>cf-connecting-ip</code> . Alleen zetten als de engine uitsluitend via die proxy bereikbaar is.

Met een ongeldige `MINI_ENGINE_PUBLIC_URL` (die begint met `https://` of `http://`), `MINI_ENGINE_SMTP_URL` of `MINI_ENGINE_TRUST_PROXY` start de engine niet; `MINI_ENGINE_MAIL_FROM` wordt gecontroleerd zodra SMTP is ingesteld. Het compose-bestand voor Coolify (`compose.coolify.yaml`) maakt `MINI_ENGINE_ADMIN_TOKEN` en `MINI_ENGINE_SMTP_URL` verplicht en zet `MINI_ENGINE_PUBLIC_URL` standaard op `https://miniengine.nl` en `MINI_ENGINE_TRUST_PROXY` op `x-forwarded-for` .

Hulpcommando's

Commando	Wat het doet
<code>mini-engine</code>	Start de server.
<code>mini-engine healthcheck</code>	Vraagt <code>/api/health</code> op via <code>127.0.0.1</code> en de poort uit <code>MINI_ENGINE_BIND</code> ; exitcode 0 bij een 200. De Docker-image gebruikt dit elke 30 seconden.
<code>mini-engine backup <bestand></code>	Schrijft een consistente kopie van de database (<code>VACUUM INTO</code>), ook terwijl de server draait. Het bestand mag nog niet bestaan.

Grenzen en rate limits

Vaste grenzen in 0.4. Overschrijden geeft een fout; niets wordt stiltejes ingekort.

Wat	Grens	Bij overschrijden
HTTP-body	96 KiB	413 invalid_json
Uploadbody (POST .../versions)	16 MiB	413 invalid_json
Websocketbericht	96 KiB	verbinding sluit
Gepubliceerde state (publiek, host en spelers samen)	64 KiB	413 state_too_large
Actiedata en instellingen	8 KiB	400 invalid_request
Actienaam	1–64 tekens: letters, cijfers, _ . -	400 invalid_request
action_id , request_id	hoogstens 64 tekens	400 invalid_request
Deduplicatievenster	laatste 256 action-ID's per kamer	oudere ID's tellen weer als nieuw
Spelersnaam, weergavenaam	1–24 tekens	400 invalid_request
Standaard max_players	maximum van de game, hoogstens 24	hoger mag tot het gamemaximum
Kamers op de server	500	503 room_limit
Websocketverbindingen	1024	503 connection_limit
Kamer zonder activiteit	12 uur	sluit met room_expired
Websocket: auth en stilte	auth binnen 5 s; hoogstens 45 s stil (ping elke 15 s)	auth_timeout , heartbeat_timeout
Accountsessie	30 dagen	401 unauthorized
E-mailadres	hoogstens 254 tekens	400 invalid_request
Bevestigingslink	48 uur, één keer	400 invalid_token
Bevestigingsmails	één per account per minuut	429 rate_limit
Gamepackage	2 MiB per bestand, 10 MiB en 4096 bestanden per game	engine start niet; bij een upload 400 invalid_request
Geüploade games	10 per account	409 quota_games
Bewaarde versies	10 per game	409 quota_releases
Versienaam van een upload	1–64 tekens: letters, cijfers, . + _ -	400 invalid_request
Bestandspad in een upload	hoogstens 255 tekens en 32 delen	400 invalid_request
Dev-key	label 1–80 tekens; geldig 1–3650 dagen of onbeperkt	400 invalid_request
MINI_ENGINE_ADMIN_TOKEN	minstens 32 tekens	engine start niet

Rate limits

HTTP-tellers lopen per IP-adres in vaste vensters van zestig seconden, of tien minuten waar dat staat. IPv6-adressen tellen per /64-blok. Daarnaast wacht een gebruikersnaam een kwartier na tien mislukte inlogpogingen, van welk adres ook; een geslaagde login zet de teller terug. Achter een reverse proxy delen alle bezoekers het adres van de proxy, tenzij `MINI_ENGINE_TRUST_PROXY` is gezet. De SDK wacht na `rate_limit` minstens 1,2 seconden voordat hij opnieuw verbindt.

Wat	Limiet	Fout
<code>POST /api/v1/rooms</code> en <code>POST .../join</code>	60 per minuut per IP-adres	429 <code>rate_limit</code>
<code>POST /api/v1/accounts/register</code> , <code>.../login</code> en <code>.../verify-email</code>	10 per minuut per IP-adres	429 <code>rate_limit</code>
Mails: <code>POST .../register</code> en <code>PUT /api/v1/accounts/email</code>	5 per tien minuten per IP-adres	429 <code>rate_limit</code>
Alle routes onder <code>/api/v1/admin/</code>	30 per minuut per IP-adres	429 <code>rate_limit</code>
<code>POST</code> , <code>PUT</code> , <code>DELETE</code> onder <code>/api/v1/dev/</code>	60 per minuut per IP-adres	429 <code>rate_limit</code>
Uploads (<code>POST .../versions</code>)	20 per tien minuten per IP-adres	429 <code>rate_limit</code>
Verkeerde admin- of dev-sleutel	na 10 keer 401 binnen tien minuten: tien minuten geweigerd, ook met de goede sleutel	429 <code>rate_limit</code>
Websocketberichten	per verbinding 240 per seconde voor de host, 60 voor andere rollen	<code>error</code> en sluiten met <code>rate_limit</code>

Foutformaat en foutcodes

Elke fout heeft dezelfde vorm, over HTTP en over de websocket. Reageer op de `code`; toon de `message` aan mensen.

```
{"error": {"code": "room_full", "message": "Deze room is vol."}}
```

Over de websocket staat dezelfde fout in `{"type": "error", "request_id": "...", "error": {...}}`.

Codes

Status	Code	Wanneer
400	<code>invalid_request</code>	Een veld klopt niet: naam, e-mailadres, instellingen, config, actie of payload.
400	<code>invalid_token</code>	Bevestigingslink onbekend, verlopen, al gebruikt of voor een ander account.
400	<code>invalid_json</code>	Geen geldige JSON, onbekende velden of verkeerd <code>Content-Type</code> . Bij een te grote body met status 413.
401	<code>unauthorized</code> , <code>auth_timeout</code>	Token ontbreekt, is ongeldig, ingetrokken of verlopen; of geen websocket-auth binnen vijf seconden.
403	<code>forbidden</code> , <code>origin_forbidden</code>	Deze rol mag dit niet, de game is van een andere ontwikkelaar, of de browser-origin is niet toegestaan.
404	<code>room_not_found</code>	De kamer bestaat niet of is verlopen.
403	<code>game_blocked</code>	De beheerder heeft deze game geblokkeerd; de ontwikkelaar kan hem niet aanzetten.
404	<code>game_not_found</code>	De game bestaat niet, is uitgezet (zonder kamer) of is geblokkeerd. Onder <code>/api/v1/dev</code> ook: niet van jou.
404	<code>game_page_not_found</code> , <code>asset_not_found</code> , <code>player_not_found</code> , <code>not_found</code>	Rolpagina, asset, speler of route bestaat niet.
404	<code>release_not_found</code>	Deze versie van de game bestaat niet (meer).
404	<code>admin_disabled</code>	Beheer staat uit: <code>MINI_ENGINE_ADMIN_TOKEN</code> is niet gezet.
404	<code>account_not_found</code> , <code>token_not_found</code>	Geen account met deze gebruikersnaam, of de dev-key bestaat niet of is al ingetrokken.
405	<code>method_not_allowed</code>	De route bestaat, de methode niet.
409	<code>version_conflict</code>	De versie is intussen veranderd. Lees de nieuwste snapshot.
409	<code>room_started</code> , <code>room_full</code> , <code>name_taken</code> , <code>already_joined</code> , <code>username_taken</code>	Binnenkomen, klaarzetten of configureren kan niet (meer), de naam of gebruikersnaam is bezet, of dit account heeft al een stoel.
409	<code>email_taken</code> , <code>email_already_verified</code>	Het adres is al bevestigd door een ander account, of al bevestigd voor dit account.
409	<code>email_not_verified</code> , <code>email_missing</code>	Beheer: een dev-key vraagt een bevestigd adres; met de hand bevestigen kan niet zonder adres.
409	<code>invalid_transition</code> , <code>players_not_ready</code>	Lifecycle in de verkeerde volgorde, of niet iedereen staat klaar.
409	<code>game_not_running</code> , <code>host_offline</code>	Een spelactie buiten <code>running</code> , of bij een hostruntime zonder verbonden host.
409	<code>session_replaced</code>	Een nieuwere verbinding gebruikt dit token.

Status	Code	Wanneer
409	<code>invalid_game_action</code> , <code>buzzer_closed</code> , <code>teams_locked</code> , <code>teams_required</code>	Spelregels van een servergame.
409	<code>game_id_taken</code> , <code>version_exists</code>	Upload met het id van een ingebouwde game, of een versie die al bestaat.
409	<code>quota_games</code> , <code>quota_releases</code>	Al 10 games op dit account, of al 10 versies van deze game.
409	<code>release_active</code> , <code>release_in_use</code> , <code>game_in_use</code>	Verwijderen kan niet: de actieve versie, of een versie of game die een lopende kamer nog speelt.
413	<code>state_too_large</code>	Gepubliceerde state groter dan 64 KiB.
429	<code>rate_limit</code>	Te veel verzoeken, berichten of mails.
500	<code>internal_error</code>	Opslaan mislukte. De oude state blijft staan.
503	<code>room_limit</code> , <code>connection_limit</code>	De server zit vol. Probeer het later opnieuw.