



LehrPuls-Austauschformat (Format-Version 2)

Dieses Dokument beschreibt vollständig, wie Inhalte in LehrPuls hinein- und wieder herauskommen: Themenbäume, Szenarien, alle elf Fragetypen einschließlich Lückentext, Sprachfassungen, Bilder, einzelne Fragen ohne Themenbaum, Kurse mit Teilnehmenden und Playlists mit Fragenblöcken. Es richtet sich an alle, die Importdateien nicht im Frageneditor erzeugen, sondern mit einem KI-Dienst, einem Skript oder aus einem anderen System heraus.

Es gibt **ein** Format. Was hier über Felder gesagt wird, gilt überall — ob die Inhalte als reine JSON-Datei kommen, als `.lpx`-Archiv oder eingefügt in den Dialog „Fragen mit KI erstellen“. Der einzige Unterschied zwischen diesen Wegen sind Bilder; das erklärt Abschnitt 2.

Inhalt

1. Womit fange ich an?

2. Die zwei Verpackungen: nacktes JSON und der .lpx-Container

3. Der Kopf der Datei

4. Themen

5. Szenarien

6. Fragen

7. Die elf Fragetypen im Einzelnen

`mc_single` — Multiple Choice mit genau einer richtigen Antwort

`mc_multi` — Multiple Choice mit mehreren richtigen Antworten

`poll` — Abstimmung ohne richtige Antwort

`poll_multi` — Abstimmung+ mit Mehrfachauswahl

`open` — Freitextfrage

`numeric` — Zahlenwert mit Toleranz

`estimate` — Schätzfrage ohne Wertung

`order` — Elemente in die richtige Reihenfolge bringen

`matching` — Elemente Kategorien zuordnen

`info` — Mitteilung ohne Frage

8. Lückentext

9. Sprachfassungen

10. Bilder

11. Fragen ohne Themenbaum

12. Kurse und Teilnehmende

13. Playlists

14. Was der Import prüft

15. Vollständiges Beispiel

16. Schritt für Schritt: eine .lpx-Datei von Hand bauen

Prüfliste

1. Womit fange ich an?

Kurz gesagt: Geben Sie dieses Dokument weiter — mehr braucht es nicht.

Wer Fragen von einem KI-Dienst erstellen lassen will, lädt zwei Dinge hoch: dieses Dokument und die eigene Quelle (etwa das Vorlesungsskript als PDF). Die KI erzeugt daraus den Inhalt einer JSON-Datei. Diesen Text fügen Sie in LehrPuls unter **Themen & Fragen** in den Dialog „Fragen mit KI erstellen“ ein — oder Sie speichern ihn als Datei mit der Endung `.json` und nehmen **Importieren (.lpx, .json)**.

Sie müssen dabei nichts auswählen oder weglassen. Eine KI liefert Text, und Text deckt dieses Format vollständig ab — bis auf Bilddateien, die kein KI-Dienst mitliefern kann. Fragen, Themen, Szenarien, die englische Fassung, Kurse und ganze Playlists gehen alle als reiner JSON-Text.

LehrPuls prüft den Text, bevor etwas angelegt wird, und meldet jeden Verstoß im Klartext. Der Dialog „Fragen mit KI erstellen“ prüft dabei jede Frage einzeln und zeigt alle Beanstandungen auf einmal — diese Liste können Sie der KI zur Korrektur zurückgeben. Importiert wird ganz oder gar nicht; ein fehlgeschlagener Versuch lässt Ihren Bestand unverändert.

Diesen Absatz können Sie der KI mitgeben: Erzeuge als Ergebnis ausschließlich den Inhalt einer einzigen JSON-Datei nach dieser Beschreibung — ohne erklärenden Text davor oder danach. Halte dich exakt an die genannten Feldnamen und Regeln und erfinde keine zusätzlichen Felder. Verwende den Fragetyp `cloze` nicht; er hat eine eigene Lücken-Syntax im Fragentext und ist für maschinell erzeugte Dateien nicht vorgesehen. Referenziere keine Bilder.

Unbekannte Zusatzfelder außerhalb von `options_json` und `i18n` werden stillschweigend übergangen — alle Angaben müssen deshalb in den hier beschriebenen Feldern stehen. Innerhalb von `options_json` und `i18n` führt ein unbekanntes Feld dagegen zur Ablehnung, damit ein Tippfehler in einer Wertungsangabe oder einer Übersetzung nicht unbemerkt verschwindet.

2. Die zwei Verpackungen: nacktes JSON und der .lpx-Container

Beide Wege tragen dieselben Inhalte in derselben Struktur. Sie unterscheiden sich in genau einem Punkt: **Bilder brauchen den Container.**

	nackte JSON-Datei	.lpx -Container
Themen, Szenarien, Fragen, Sprachfassungen	ja	ja
Kurse mit Teilnehmenden, Playlists	ja	ja
Bilddateien	nein	ja
Einfügen in „Fragen mit KI erstellen“	ja	nein (nur als Datei)
Von einem KI-Dienst erzeugbar	ja	nein

Eine `.lpx`-Datei ist ein gewöhnliches ZIP-Archiv mit fester Endung. Innen liegen genau zwei Dinge:

```
Sammlung.lpx (ZIP)
├─ export.json      – alle Inhalte, menschenlesbar (UTF-8, ohne Byte-Order-Mark)
└─ media/
   ├─ 3f9ab2...c1.png – Bilddateien unter ihrem Inhalts-Hash (siehe Abschnitt 10)
   └─ 91c07d...e4.svg
```

Die nackte JSON-Datei ist nichts anderes als der Inhalt dieser `export.json`, für sich allein gespeichert. Alles, was dieses Dokument über Felder sagt, beschreibt deshalb beide Wege zugleich.

Die Datei `export.json` muss auf der obersten Ebene des Archivs liegen, sonst meldet LehrPuls „export.json fehlt“. Bilder liegen im Unterordner `media`; andere Einträge im Archiv werden übergangen. Die Kompressionsmethode ist gleichgültig – LehrPuls selbst schreibt `export.json` komprimiert und legt Bilder unkomprimiert ab, weil PNG und JPEG bereits komprimiert sind.

So entsteht das Archiv von Hand. Legen Sie einen Ordner mit `export.json` und dem Unterordner `media` an. Am Mac und unter Linux packen Sie im Terminal aus diesem Ordner heraus mit `zip -r ../Sammlung.lpx export.json media`. Unter Windows markieren Sie im Explorer die Datei `export.json` und den Ordner `media`, wählen im Kontextmenü „Senden an → ZIP-komprimierter Ordner“ und benennen das Ergebnis in `Sammlung.lpx` um. Packen Sie nicht den übergeordneten Ordner ein – dann läge `export.json` eine Ebene zu tief.

Ohne Bilder reicht die nackte JSON-Datei. Speichern Sie den Inhalt von `export.json` direkt als Datei mit der Endung `.json` oder `.lpx`. LehrPuls prüft den Inhalt, nicht die Endung: Beginnt die Datei mit den ZIP-Kennbytes, wird sie als Archiv gelesen; sonst liest LehrPuls das erste JSON-Objekt, das in ihr steht. Text davor und danach – eine Einleitung, ein Code-Zaun, eine Rückfrage der KI – wird dabei übergangen. Eine Datei ganz ohne geschweifte Klammer wird mit „weder ZIP noch JSON“ abgewiesen. Auch LehrPuls selbst schreibt diese Form: Wer beim Exportieren im Speichern-Dialog den Dateityp **JSON (ohne Bilder)** wählt oder der Zieldatei die Endung `.json` gibt, erhält den Inhalt von `export.json` als Textdatei; Bildverweise bleiben im Text stehen, die Bilddateien fehlen.

Beide Dateien kommen über denselben Menüpunkt **Importieren (.lpx, .json)** herein. Der Dialog „Fragen mit KI erstellen“ nimmt eingefügten JSON-Text; für eine Datei mit Bildern hat er den Knopf **.lpx importieren**.

3. Der Kopf der Datei

`export.json` ist ein einzelnes JSON-Objekt. Der Kopf besteht aus diesen Feldern:

Feld	Pflicht	Bedeutung
format	ja	wörtlich "lehrpuls-export" — sonst „keine LehrPuls-Exportdatei“
version	ja	Format-Version, derzeit 2 ; Dateien der Version 1 werden unverändert gelesen (Abschnitt 9), eine höhere Zahl wird als „aus einer neueren LehrPuls-Version“ abgewiesen
exported_at	nein	Zeitstempel des Exports, nur zur Anzeige
app_version	nein	LehrPuls-Version des Exports, nur zur Anzeige
source_path	nein	Themenpfad der Herkunft als Liste von Namen; der Import-Dialog zeigt ihn unter „Herkunft“, und beim Import loser Fragen ohne Ziel-Thema heißt das neu angelegte Thema nach seinem letzten Eintrag

Darunter stehen die Inhaltsblöcke. Jeder ist optional, sie lassen sich beliebig kombinieren, und mindestens einer muss Inhalt haben — sonst meldet LehrPuls „Die Datei enthält keine Inhalte“:

Block	Inhalt	Abschnitt
topics	Themenbaum mit Szenarien, Fragen und Unterthemen	4 bis 10
scenarios und questions	Szenarien und Fragen ohne Themenbaum	11
courses	Kurse mit Teilnehmenden	12
playlists	Playlists mit Abschnitten und Fragenblöcken	13

Unbekannte Felder werden stillschweigend übergangen — mit zwei Ausnahmen: In options_json einer Frage und in i18n (Abschnitt 9) führt jedes fremde Feld zur Ablehnung der Datei, in i18n ebenso ein Schlüssel, der keine Sprachkennung ist. Alle Angaben müssen deshalb in den hier beschriebenen Feldern stehen. Ein Block scenarios allein genügt nicht als Inhalt: Szenarien ohne Themenbaum kommen nur zusammen mit questions.

Die kleinste vollständige Datei:

```
{
  "format": "lehrpuls-export",
  "version": 2,
  "topics": [
    {
      "name": "Probethema",
      "questions": [
        { "qtype": "open", "title": "Erste Frage", "body_md": "Was ist LehrPuls?",
"options_json": {} }
      ]
    }
  ]
}
```

Was beim Import immer gilt. Der Import läuft ganz oder gar nicht: Ist irgendetwas in der Datei nicht in Ordnung, bleibt der Bestand unverändert, und die Meldung nennt den Grund. Der Import legt grundsätzlich neu an — nichts wird zusammengeführt oder überschrieben; ein Thema mit gleichem Namen entsteht ein

zweites Mal, und Sie entscheiden selbst, was Sie zusammenlegen. Nur Bilder gleichen Inhalts werden nicht doppelt gespeichert. Die Reihenfolge in der Datei ist die Reihenfolge im Bestand. Importierte Fragen tragen in der Spalte **Quelle** den Namen der Datei.

4. Themen

Ein Thema ist ein Knoten des Baums unter **Themen & Fragen**. Unterthemen stehen in `children`, beliebig tief verschachtelt:

Feld	Pflicht	Bedeutung
<code>name</code>	ja	Name des Themas
<code>i18n</code>	nein	Sprachfassungen des Namens, z. B. <code>{ "de": { "name" }, "en": { "name" } }</code> — siehe Abschnitt 9
<code>archived</code>	nein	<code>true</code> legt das Thema archiviert an; Vorgabe <code>false</code>
<code>scenarios</code>	nein	Szenarien dieses Themas (Abschnitt 5)
<code>questions</code>	nein	Fragen dieses Themas (Abschnitt 6)
<code>children</code>	nein	Unterthemen gleichen Aufbaus

Beim Import unter **Themen & Fragen** wählen Sie ein **Ziel-Thema**, unter das der Baum aus der Datei gehängt wird; — **Oberste Ebene** — legt die Themen der Datei als neue Hauptthemen an. Importieren Sie dieselbe Datei auf der Seite **Kurse & Teilnehmende** oder **Playlists**, landen enthaltene Themen immer auf der obersten Ebene.

5. Szenarien

Ein Szenario ist eine gemeinsame Ausgangslage, auf die sich mehrere Fragen beziehen — die Fallbeschreibung vor einer Reihe von Aufgaben. In der Datei steht es im Thema unter `scenarios`, und Fragen verweisen mit ihrem Feld `scenario` auf seinen Schlüssel:

Feld	Pflicht	Bedeutung
<code>key</code>	ja	dateilokaler Schlüssel, frei wählbare Zeichenkette, in der ganzen Datei eindeutig — LehrPuls vergibt beim Export <code>s1</code> , <code>s2</code> ...
<code>title</code>	ja	Titel, nicht leer
<code>body_md</code>	nein	der Szenario-Text (Markdown, Formeln, Bilder)
<code>i18n</code>	nein	weitere Sprachfassungen von Titel und Text, z. B. <code>{ "en": { "title", "body_md" } }</code> — siehe Abschnitt 9
<code>archived</code>	nein	<code>true</code> legt es archiviert an

Die Schlüssel haben außerhalb der Datei keine Bedeutung; beim Import werden sie durch die neuen Datenbank-Einträge ersetzt. Ein doppelter Schlüssel wird mit „Doppelter Szenario-Schlüssel in der Datei“ abgewiesen, ein Verweis auf einen nicht definierten Schlüssel mit „Szenario-Schlüssel ist in der Datei nicht definiert“. Die Reihenfolge in der Datei spielt für Verweise keine Rolle: Alle Szenarien werden vor allen Fragen angelegt.

Szenarien gelten in LehrPuls für ihr Thema und alle Unterthemen. Definieren Sie ein Szenario deshalb in dem Thema, in dem die Fragen stehen, oder in einem übergeordneten; technisch nimmt der Import auch Verweise über Themengrenzen hinweg an, aber in der Szenarien-Leiste der Fragenliste erscheint das Szenario dann nicht als geerbt. Wer eine Frage im Fragenblock einer Playlist mit einem Szenario spielen will, verweist von der Playlist aus auf denselben Schlüssel (Abschnitt 13).

6. Fragen

Jede Frage ist ein Objekt in `questions` — im Thema oder, ohne Themenbaum, auf oberster Ebene (Abschnitt 11):

Feld	Pflicht	Bedeutung
<code>qtype</code>	ja	Fragetyp, einer der elf Werte aus der Tabelle unten
<code>title</code>	ja	kurzer, nicht leerer Titel für die Fragenliste
<code>body_md</code>	nein	der Fragentext (Markdown, Formeln, Bilder); beim Lückentext trägt er die Lücken
<code>options_json</code>	ja	JSON-Objekt mit den typabhängigen Feldern; notfalls <code>{}</code>
<code>solution_md</code>	nein	Lösungshinweis, wird beim Auflösen gezeigt
<code>i18n</code>	nein	weitere Sprachfassungen der Texte, z. B. <code>{ "en": { "title", "body_md", "solution_md", "options_json" } }</code> — siehe Abschnitt 9; <code>options_json</code> trägt dort nur die Textfelder an ihren Kennungen
<code>difficulty</code>	nein	Schwierigkeitsgrad, ganze Zahl von 1 (leicht) bis 5 (schwer)
<code>tags</code>	nein	Stichworte, kommagetrennt in einer Zeichenkette, z. B. "Kapitel 3, Grundlagen"
<code>time_budget_s</code>	nein	vorgesehene Bearbeitungszeit in Sekunden, positive ganze Zahl
<code>scenario</code>	nein	Schlüssel eines Szenarios aus der Datei (Abschnitt 5)
<code>archived</code>	nein	<code>true</code> legt die Frage archiviert an
<code>key</code>	nein	dateilokaler Fragen-Schlüssel, in der ganzen Datei eindeutig; nur nötig, wenn eine Playlist auf die Frage verweist (Abschnitt 13)

In allen Markdown-Feldern — `body_md`, `solution_md`, den Texten der Antwortoptionen, dem Szenario-Text und dem Arbeitsauftrag einer Playlist — gilt: Markdown ist erlaubt (Listen, **fett**, Code, Tabellen), mathematische Formeln in KaTeX-Schreibweise zwischen `$...$` im Text oder `$$...$$` auf eigener Zeile; ein Dollarzeichen direkt hinter einem Buchstaben oder einer Ziffer (`US$`, `99$`) bleibt ein Währungszeichen. Wie überall in JSON steht jeder Backslash verdoppelt: `"$\frac{a}{b}$"` ergibt die Formel $\frac{a}{b}$. Einen einfachen Backslash vor einem Formelnamen (`\Omega`) korrigiert LehrPuls beim Einlesen. Jeder Zeilenumbruch wird als

neue Zeile angezeigt, eine Leerzeile trennt Absätze. Kein HTML — HTML-Tags erscheinen als Text. Bilder werden als Markdown-Bild mit einer Referenz in den Medienordner eingebunden (Abschnitt 10).

Die elf Fragetypen. Die Regeln zu `options_json` beschreibt Abschnitt 7 im Detail; das vollständige Beispiel in Abschnitt 15 zeigt jeden Typ einmal. Die Spalte „Name in LehrPuls“ nennt den Typ so, wie ihn die Oberfläche und die Beispiel-Prompts des Dialogs „Fragen mit KI erstellen“ nennen — in der Datei steht immer der Wert aus der Spalte `qtype`. Hier die Übersicht:

qtype	Name in LehrPuls	Bedeutung	Felder in <code>options_json</code>
<code>open</code>	Offene Frage	Freitextfrage	<code>max_len</code> optional, 1 bis 10000, Vorgabe 500; <code>sketch</code> optional (<code>true</code> erlaubt zusätzlich eine Skizze)
<code>mc_single</code>	Multiple Choice (eine richtig)	Multiple Choice, genau eine richtige Antwort	<code>choices</code> (mindestens zwei, je <code>id</code> und <code>text_md</code>), <code>correct</code> mit genau einer ID
<code>mc_multi</code>	Multiple Choice (mehrere richtig)	Multiple Choice, mehrere richtige Antworten	wie <code>mc_single</code> , <code>correct</code> mit mindestens einer ID
<code>poll</code>	Abstimmung	Abstimmung ohne Wertung, eine Stimme	<code>choices</code> wie oben, <code>correct</code> leer oder weggelassen
<code>poll_multi</code>	Abstimmung+ (Mehrfachauswahl)	Abstimmung+ mit Mehrfachauswahl	wie <code>poll</code>
<code>numeric</code>	Zahlenantwort	Zahlenwert mit Toleranz	<code>answer</code> Pflicht, <code>tolerance</code> optional (nicht negativ, Vorgabe 0), <code>unit</code> optional
<code>estimate</code>	Schätzung	Schätzfrage ohne Wertung	<code>unit</code> und <code>reference</code> optional
<code>order</code>	Reihenfolge	Reihenfolge herstellen	<code>items</code> (mindestens zwei, je <code>id</code> und <code>text_md</code>), <code>correct_order</code> mit genau allen IDs
<code>matching</code>	Zuordnung	Elemente Kategorien zuordnen	<code>categories</code> und <code>items</code> (je mindestens zwei); jedes Element nennt in <code>category</code> seine richtige Kategorie
<code>info</code>	Mitteilung	Mitteilung ohne Frage	<code>url</code> optional (<code>http://</code> , <code>https://</code> oder <code>mailto:</code>), <code>ack</code> "none", "ok" (Vorgabe) oder "yesno", Knopf-Beschriftungen <code>ack_yes</code> und <code>ack_no</code> mit höchstens 40 Zeichen
<code>cloze</code>	Lückentext	Lückentext	leer: <code>{}</code> — die Lücken stehen im Fragentext (Abschnitt 8)

IDs innerhalb einer Frage sind frei wählbare, nicht leere Zeichenketten und müssen eindeutig sein; Optionstexte dürfen nicht leer sein.

7. Die elf Fragetypen im Einzelnen

mc_single — Multiple Choice mit genau einer richtigen Antwort

```
{
  "choices": [
    { "id": "a", "text_md": "Antwort A" },
    { "id": "b", "text_md": "Antwort B" },
    { "id": "c", "text_md": "Antwort C" }
  ],
  "correct": ["b"]
}
```

Mindestens zwei Antwortoptionen; die IDs sind frei wählbare Zeichenketten (auch "1" steht in Anführungszeichen), müssen aber eindeutig und nicht leer sein, die Optionstexte ebenfalls nicht leer; `correct` enthält genau eine dieser IDs.

mc_multi — Multiple Choice mit mehreren richtigen Antworten

Gleicher Aufbau wie `mc_single`; `correct` enthält mindestens eine ID, es dürfen mehrere sein.

poll — Abstimmung ohne richtige Antwort

Gleicher Aufbau wie `mc_single`, aber ohne Wertung: `correct` bleibt leer oder wird weggelassen. Teilnehmende wählen genau eine Option.

poll_multi — Abstimmung+ mit Mehrfachauswahl

Wie `poll`, Teilnehmende dürfen aber mehrere Optionen ankreuzen. `correct` bleibt ebenfalls leer.

open — Freitextfrage

```
{ "max_len": 500 }
```

`max_len` ist optional (Vorgabe 500) und begrenzt die Antwortlänge in Zeichen (1 bis 10000). `sketch` ist optional: Mit `"sketch": true` bekommen die Teilnehmenden unter dem Textfeld eine Zeichenfläche für Stift, Finger oder Maus; fehlt die Angabe, gibt es nur das Textfeld.

numeric — Zahlenwert mit Toleranz

```
{ "answer": 42.0, "tolerance": 0.5, "unit": "m" }
```

Eine Antwort gilt als richtig, wenn ihr Abstand zu `answer` höchstens `tolerance` beträgt. `answer` ist Pflicht. `tolerance` ist optional (Vorgabe 0 — exakter Wert) und darf nicht negativ sein, `unit` ist eine optionale Einheit zur Anzeige.

estimate — Schätzfrage ohne Wertung

```
{ "unit": "km", "reference": 384400 }
```

Beide Felder sind optional; `reference` ist der Referenzwert, der beim Auflösen gezeigt wird.

order — Elemente in die richtige Reihenfolge bringen

```
{
  "items": [
    { "id": "1", "text_md": "Wasser aufsetzen" },
    { "id": "2", "text_md": "Kaffee mahlen" },
    { "id": "3", "text_md": "Aufbrühen" }
  ],
  "correct_order": ["1", "2", "3"]
}
```

Mindestens zwei Elemente mit eindeutigen, nicht leeren IDs und nicht leeren Texten; `correct_order` enthält genau alle IDs in der richtigen Reihenfolge.

matching — Elemente Kategorien zuordnen

```
{
  "categories": [
    { "id": "k1", "text_md": "Säugetier" },
    { "id": "k2", "text_md": "Vogel" }
  ],
  "items": [
    { "id": "i1", "text_md": "Delfin", "category": "k1" },
    { "id": "i2", "text_md": "Pinguin", "category": "k2" }
  ]
}
```

Mindestens zwei Kategorien und zwei Elemente, IDs jeweils eindeutig und nicht leer, Texte nicht leer; `category` nennt die richtige Kategorie des Elements und ist die Wertung.

info — Mitteilung ohne Frage

```
{ "ack": "ok" }
```

Zeigt den `body_md` als Mitteilung. `ack` ist optional (Vorgabe "ok") und steuert die Rückmeldung der Teilnehmenden: "none" (nur lesen), "ok" (mit einem Knopf bestätigen) oder "yesno" (zwischen zwei Knöpfen wählen); bei "yesno" beschriften die optionalen Felder `ack_yes` und `ack_no` die beiden Knöpfe, bei "ok" darf allein `ack_yes` den einzelnen Knopf beschriften, bei "none" bleiben beide leer; `ack_no` ist nur zusammen mit `ack_yes` erlaubt, und eine Beschriftung hat höchstens 40 Zeichen. Optional trägt `url` einen Link, der mit `http://` oder `https://` beginnen muss und keine Leerzeichen enthalten darf — oder eine E-Mail-Adresse als `mailto:name@hochschule.example` (auf Wunsch mit `?subject=...`).

Der elfte Typ, `cloze` (Lückentext), verwendet eine eigene Lücken-Syntax im Fragentext und steht deshalb in einem eigenen Abschnitt (8). Für KI-erzeugte Dateien ist er nicht vorgesehen — ein KI-Dienst lässt diesen Typ weg.

8. Lückentext

Beim Fragetyp `cloze` bleibt `options_json` leer, und die Lücken stehen samt Lösungen direkt im Fragentext: `{{Lösung}}` markiert eine Lücke, ein senkrechter Strich trennt gleichwertige Alternativen — `{{Wien|Vienna}}` nimmt beides an. Mindestens eine Lücke muss im Text stehen, und jede Lücke braucht mindestens eine nicht leere Lösung. Lücken funktionieren auch in Tabellenzellen und Listen. Eine Schreibweise, um `{{` wörtlich zu zeigen, gibt es nicht.

```
{
  "qtype": "cloze",
  "title": "Hauptstädte",
  "body_md": "Die Hauptstadt von Österreich ist {{Wien|Vienna}}, die von Frankreich {{Paris}}.\nDer Siedepunkt von Wasser liegt bei {{100}} °C.",
  "options_json": {}
}
```

Beim Vergleich der Antworten schneidet LehrPuls Leerzeichen an den Rändern ab, fasst mehrfache Leerzeichen zusammen und ignoriert Groß- und Kleinschreibung; sind Lösung und Eingabe Zahlen, zählt der Zahlenwert, ein Komma gilt als Dezimalpunkt (22, 22.0 und 22,0 sind gleich). Die Breite des Eingabefelds richtet sich nach der längsten Lösung. Die Lösungen verlassen den Rechner des Lehrenden nicht: Teilnehmende bekommen nur neutrale Platzhalter, bis aufgelöst wird.

9. Sprachfassungen

Eine Session läuft durchgängig in einer Sprache. Deshalb tragen nicht nur die Fragen mehrere Sprachfassungen, sondern auch die Texte um sie herum — der Szenario-Text vor einer Fragenreihe, der Arbeitsauftrag eines Fragenblocks und die Namen von Abschnitten und Themen.

Vier Objektarten haben dafür ein Feld `i18n`. Es ist überall gleich gebaut: ein Objekt mit der Sprachkennung als Schlüssel, darunter dieselben Feldnamen wie in den Hauptfeldern. Als Sprachkennung gilt jedes Kürzel aus zwei oder drei Kleinbuchstaben, wahlweise mit Zusatz — `de`, `en`, `fr`, `pt-br`. Alle Sprachen sind gleichrangig; welche davon eine Lehrperson pflegt, legt sie im Cockpit unter **Einstellungen** → **Sprachen der Inhalte** fest.

Objekt	Felder je Sprache
Frage (Abschnitt 6)	<code>title</code> , <code>body_md</code> , <code>solution_md</code> , <code>options_json</code>
Szenario (Abschnitt 5)	<code>title</code> , <code>body_md</code>
Playlist-Knoten (Abschnitt 13)	<code>name</code> , <code>preamble_md</code>
Thema (Abschnitt 4)	<code>name</code>

Die Hauptfelder (`title`, `body_md`, `name` ...) tragen die Fassung der ersten vorhandenen Sprache. Sie sind der Rückfall: Fehlt ein Text in der gewünschten Sprache, wird er von dort angezeigt — halb übersetzte Sammlungen bleiben spielbar. Für selbst geschriebene Dateien gilt eine einfache Regel:

- Stehen die Hauptfelder auf **Deutsch**, brauchen Sie nichts weiter zu tun. LehrPuls übernimmt sie beim Import als deutsche Fassung; weitere Sprachen stehen in `i18n`. So waren auch Dateien der Format-Version 1 gebaut, und so werden sie weiterhin gelesen.
- Stehen die Hauptfelder in einer **anderen Sprache** — etwa bei einer rein englischen Sammlung —, wiederholen Sie dieselben Texte in `i18n` unter ihrer Kennung (`"en"`). Daran erkennt LehrPuls, dass es keine deutsche Fassung gibt.

Exportdateien aus LehrPuls tragen in `i18n` stets alle Fassungen, auch `de`.

Bei Fragen stehen in `options_json` einer Sprachfassung **nur die Textfelder**, festgemacht an den Kennungen aus `options_json` der Frage: `choices`, `items` und `categories` als Listen aus `{ "id", "text_md" }`, dazu `unit`, `ack_yes` und `ack_no`, wo der Typ sie kennt. Eine Kennung, die es in `options_json` der Frage nicht gibt, führt zur Ablehnung der Datei. Alle übrigen Felder gehören nicht in eine Sprachfassung, denn sie gelten für alle Sprachen: die Wertungsfelder (`correct`, `correct_order`, `answer`, `tolerance`, `reference`, `category`) ebenso wie die Einstellungen `max_len`, `url` und `ack`. Wer sie dennoch dorthin kopiert, dessen Datei wird abgelehnt. Hat ein Fragetyp keine Textfelder in `options_json` — etwa die Freitextfrage `open` oder der Lückentext `cloze` —, fehlt `options_json` in der Sprachfassung ganz; die englische Fassung einer Freitextfrage ist also nur `{ "title", "body_md", "solution_md" }`. Beim Lückentext muss jede Fassung gleich viele Lücken in derselben Reihenfolge haben: Die erste Lücke des englischen Textes entspricht der ersten des deutschen. Gewertet wird je Lücke gegen die Lösungen aller Fassungen — „sun“ zählt ebenso wie „Sonne“.

```
{
  "qtype": "mc_single",
  "title": "Ort der Fotosynthese",
  "body_md": "In welchem Zellorganell findet die Fotosynthese statt?",
  "options_json": {
    "choices": [
      { "id": "a", "text_md": "Chloroplast" },
      { "id": "b", "text_md": "Mitochondrium" }
    ],
    "correct": ["a"]
  },
  "solution_md": "Die Fotosynthese läuft in den Chloroplasten ab.",
  "i18n": {
    "en": {
      "title": "Site of photosynthesis",
      "body_md": "In which organelle does photosynthesis take place?",
      "options_json": {
        "choices": [
          { "id": "a", "text_md": "Chloroplast" },
          { "id": "b", "text_md": "Mitochondrion" }
        ]
      },
      "solution_md": "Photosynthesis takes place in the chloroplasts."
    }
  }
}
```

```
}  
}
```

Beim Szenario sind es nur Titel und Text:

```
{  
  "key": "s1",  
  "title": "Werksplanung",  
  "body_md": "Ein Werk wird geplant ...",  
  "i18n": { "en": { "title": "Plant planning", "body_md": "A plant is being planned ..." } }  
}
```

Beim Playlist-Knoten sind es der Name des Knotens und sein Arbeitsauftrag:

```
{  
  "name": "Kostenblock",  
  "kind": "complex",  
  "preamble_md": "Bearbeiten Sie die Fragen in Zweiergruppen.",  
  "i18n": { "en": { "name": "Cost block", "preamble_md": "Work on the questions in pairs." } }  
}
```

Läuft eine Session auf Englisch, erscheinen Szenario-Text, Arbeitsauftrag und Blockname in dieser Sprache, genau wie die Fragen. Ein Text ohne englische Fassung bleibt dabei stehen, wie er ist — die Session bricht nicht ab, sie zeigt an dieser Stelle die erste vorhandene Sprache.

Die Lehrperson wählt beim Start einer Session die Fragensprache; die Teilnehmenden können auf ihrem Gerät zwischen den vorhandenen Sprachen wechseln. Antworten und Statistik hängen an den Kennungen, nicht am Text — sie sind für alle Sprachen dieselben.

Nicht übersetzbar sind Kurs- und Teilnehmernamen: Sie erscheinen nur in der Verwaltung, nie beim Abspielen. Auch der Name der Playlist-Wurzel bleibt einsprachig — er ist ein Verwaltungsname; übersetzt werden die Abschnitte und Blöcke darunter. Themennamen erscheinen ebenfalls nie beim Abspielen; ihre Sprachfassungen dienen allein der Arbeit im Cockpit, dessen Themenbaum der Inhaltssprache folgt.

10. Bilder

Bilder liegen in LehrPuls inhaltsadressiert: Der Dateiname ist der SHA-256-Hash des Dateiinhalts als 64 Hexadezimalzeichen in Kleinbuchstaben, gefolgt von der Endung. Im Markdown wird ein Bild mit genau diesem Namen referenziert, und im Archiv liegt es unter demselben Namen im Ordner `media`:

```
![Schaltbild]  
(media/9f86d081884c7d659a2feaa0c55ad015a3bf4f1b2b0b822cd15d6c15b0f00a08.png)
```

Zulässig sind die Endungen `png`, `jpg`, `gif`, `webp` und `svg`; JPEG-Dateien müssen `jpg` heißen, nicht `jpeg`. Eine Datei darf höchstens 10 MB groß sein. Beim Import berechnet LehrPuls den Hash jeder Bilddatei neu und vergleicht ihn mit dem Namen — stimmt er nicht, wird der gesamte Import mit „Bilddatei passt nicht

zu ihrem Namen" abgebrochen, bevor etwas in die Datenbank gelangt. Ein Bild, das im Zielbestand schon liegt, wird nicht ein zweites Mal gespeichert; ein Bild, das mehrfach referenziert wird, gehört nur einmal ins Archiv.

Den Hash berechnen. Am Mac und unter Linux liefert `shasum -a 256 bild.png` den Wert, unter Windows `certutil -hashfile bild.png SHA256` oder in PowerShell `Get-FileHash bild.png`. Benennen Sie die Datei in `<hash>.png` um und legen Sie sie unter `media/` ab. Ein Skript, das die Datei erzeugt, berechnet den Hash direkt beim Kopieren.

Wo Bilder stehen dürfen. In jedem Markdown-Feld: im Fragentext, im Lösungshinweis, in den Texten der Antwortoptionen, im Szenario-Text und im Arbeitsauftrag einer Playlist. Der Export sucht in allen diesen Feldern nach Referenzen und packt die gefundenen Dateien ein.

Bildgröße. Hinter der Referenz darf eine Größenstufe stehen, ohne Anführungszeichen: `![Bild] (media/...png klein)`. Die Stufen heißen `klein`, `mittel`, `groß`, `sehr groß` und `original` (englisch `small`, `medium`, `large`, `extra large`); ohne Angabe gilt `groß`. Der Beamer bemisst die Stufe nach der Höhe, Teilnehmergeräte und Druck nach der Breite.

Fehlende Dateien. Eine Referenz ohne zugehörige Datei ist kein Fehler: Die Frage wird angelegt, das Bild bleibt leer. Das gilt auch für nacktes JSON, das nie Bilder trägt. Die Vorschau des Imports nennt solche Referenzen als fehlende Bilder. Die Referenz bleibt dabei gültig — sobald die Datei über einen anderen Weg im Medienordner liegt, etwa durch den Import einer `.lpx`-Datei, die dasselbe Bild enthält, erscheint es.

11. Fragen ohne Themenbaum

Einzelne Fragen können ohne umgebendes Thema in der Datei stehen — so exportiert LehrPuls markierte Fragen aus der Fragenliste. Sie liegen dann im Block `questions` auf oberster Ebene, mitreferenzierte Szenarien im Block `scenarios` daneben:

```
{
  "format": "lehrpuls-export",
  "version": 2,
  "source_path": ["Medizin", "Innere"],
  "scenarios": [
    { "key": "s1", "title": "Fall Meier", "body_md": "Ein 54-jähriger Patient stellt sich vor mit ..." }
  ],
  "questions": [
    {
      "qtype": "open",
      "title": "Erstdiagnose",
      "body_md": "Welche Verdachtsdiagnose stellen Sie?",
      "options_json": { "max_len": 500, "sketch": true },
      "scenario": "s1"
    }
  ]
}
```

Beim Import unter **Themen & Fragen** landen diese Fragen und Szenarien im gewählten Ziel-Thema. Ist kein Ziel gewählt, legt LehrPuls dafür ein neues Hauptthema an — benannt nach dem letzten Eintrag von `source_path`, ersatzweise nach dem Dateinamen ohne Endung; bei eingefügtem Text ohne Datei heißt es „Import“.

12. Kurse und Teilnehmende

Ein Kurs wird als Anlage exportiert: Name, Semester, Beschreibung, Löschfrist und Teilnehmerliste — ohne Sessions und Antworten.

Feld	Pflicht	Bedeutung
<code>name</code>	ja	Kursname, nicht leer
<code>semester</code>	nein	Semester als Text, z. B. "WS 2026/27"
<code>description</code>	nein	Beschreibung
<code>purge_after</code>	nein	Löschfrist als Datum <code>JJJJ-MM-TT</code>
<code>archived</code>	nein	<code>true</code> legt den Kurs archiviert an; Vorgabe <code>false</code>
<code>participants</code>	nein	die Teilnehmerliste

Jeder Eintrag der Teilnehmerliste:

Feld	Pflicht	Bedeutung
<code>display_name</code>	ja	Anzeigenname, nicht leer und innerhalb des Kurses eindeutig
<code>import_ref</code>	nein	Matrikelnummer oder andere Kennung
<code>email</code>	nein	E-Mail-Adresse
<code>note</code>	nein	Anmerkung
<code>active</code>	nein	<code>false</code> setzt den Eintrag inaktiv; Vorgabe <code>true</code>
<code>short_code</code>	nein	Kennung der Person für den Gerätewechsel: ein Großbuchstabe ohne <code>I</code> und <code>O</code> , gefolgt von drei Ziffern, z. B. "K417"; fehlt sie, vergibt LehrPuls eine

```
{
  "format": "lehrpuls-export",
  "version": 2,
  "courses": [
    {
      "name": "Grundlagen der Elektrotechnik 1",
      "semester": "WS 2026/27",
      "description": "Pflichtfach im 1. Semester",
      "purge_after": "2027-09-30",
      "participants": [
        {
          "display_name": "Anna Muster",
```

```

    "import_ref": "12345678",
    "email": "anna.muster@hochschule.example",
    "short_code": "K417"
  },
  { "display_name": "Ben Beispiel", "note": "Wiederholer", "active": false }
]
}
}
}

```

Der Import läuft über die Seite **Kurse & Teilnehmende**; enthält eine Datei neben Themen oder Playlists auch Kurse, werden diese auf jeder Seite mit importiert und in der Vorschau angekündigt. Eine solche Datei enthält personenbezogene Daten und verlässt die geschützte Ablage von LehrPuls — geben Sie sie nur mit Rechtsgrundlage weiter. Ein archivierter Kurs behält seinen Archivstand: LehrPuls schreibt dann `"archived": true`, und der Import legt ihn wieder archiviert an. Ältere LehrPuls-Versionen übergehen das Feld und legen ihn als gewöhnlichen Kurs an.

Die Kennung einer Person reist mit: LehrPuls schreibt sie beim Export als `short_code`, und der Import übernimmt sie, damit der Gerätewechsel per Kennung nach einem Umzug des Kurses weiter gilt. Hat eine Angabe nicht die Form einer Kennung oder kommt sie im selben Kurs ein zweites Mal vor, ist das kein Fehler — die Person bekommt dann eine neue Kennung, ebenso wenn das Feld fehlt. In selbst geschriebenen Dateien lassen Sie das Feld am besten weg. Ältere LehrPuls-Versionen (bis 1.14.0) übergehen das Feld und vergeben neue Kennungen.

13. Playlists

Eine Playlist ist ein Baum: die Wurzel, darunter Abschnitte, darin Fragen-Einträge und Fragenblöcke. In der Datei verweisen die Einträge über dateilokale Schlüssel auf Fragen, die im selben Export stehen — im Block `topics` oder `questions`, jeweils mit gesetztem Feld `key`. So bleibt die Datei selbsttragend; einen Kurs- oder Semester-Bezug trägt sie nicht.

Feld	Pflicht	Bedeutung
<code>name</code>	ja	Name des Knotens, nicht leer
<code>kind</code>	nein	<code>"section"</code> für einen Abschnitt (Vorgabe) oder <code>"complex"</code> für einen Fragenblock; der oberste Knoten wird immer als Wurzel angelegt, egal was hier steht — Exporte aus LehrPuls schreiben dort <code>"root"</code>
<code>preamble_md</code>	nein	der Arbeitsauftrag eines Fragenblocks (Markdown, Formeln, Bilder)
<code>i18n</code>	nein	weitere Sprachfassungen von Name und Arbeitsauftrag, z. B. <code>{ "en": { "name", "preamble_md" } }</code> — siehe Abschnitt 9
<code>scenario</code>	nein	Schlüssel des Szenarios, das der Fragenblock als Präambel zeigt
<code>skipped</code>	nein	<code>true</code> markiert den Knoten als übersprungen
<code>archived</code>	nein	<code>true</code> legt die Playlist archiviert an; gilt nur am obersten Knoten, Vorgabe <code>false</code>
<code>items</code>	nein	die Fragen-Einträge dieses Knotens, in Reihenfolge
<code>children</code>	nein	Unterknoten gleichen Aufbaus
<code>after_items</code>	nein	nur an Unterknoten: so viele Fragen-Einträge des übergeordneten Knotens stehen vor diesem Unterknoten; fehlt das Feld, folgt er auf alle Fragen

Fragen und Unterknoten eines Knotens können sich abwechseln, etwa erst ein Übungsabschnitt und dann die Einzelfragen. Die Stellung eines Unterknotens gibt `after_items` an: `0` stellt ihn vor die erste Frage, `2` hinter die zweite. Mehrere Unterknoten mit gleichem Wert folgen in ihrer Reihenfolge in `children`. Ältere LehrPuls-Versionen übergehen das Feld und stellen alle Unterknoten hinter die Fragen.

Ein Fragen-Eintrag hat drei Felder: `question` mit dem Schlüssel der Frage (Pflicht), `peer_instruction` (optional, `true` schaltet Peer Instruction ein; nur für `mc_single` und `mc_multi` sinnvoll) und `priority` (optional, `1` = Muss, `2` = Soll, `3` = Kann; fehlt das Feld oder ist der Wert ungültig, gilt „Soll“). Ein Fragenblock enthält seine Teilfragen als `items`; Fragenblöcke ineinander sind nicht vorgesehen. Ein Verweis auf einen Schlüssel, den keine Frage der Datei trägt, wird mit „Fragen-Schlüssel ist in der Datei nicht definiert“ abgewiesen, ein doppelt vergebener Schlüssel mit „Doppelter Fragen-Schlüssel in der Datei“.

```
{
  "format": "lehrpuls-export",
  "version": 2,
  "topics": [
    {
      "name": "Museum",
      "scenarios": [
        {
          "key": "s1",
          "title": "Sonderausstellung",
          "body_md": "Das Stadtmuseum plant für den Herbst eine Sonderausstellung ..."
        }
      ],
      "questions": [
        {
          "key": "q1",
```

```

    "qtype": "open",
    "title": "Konzept",
    "body_md": "Skizzieren Sie das Ausstellungskonzept.",
    "options_json": { "max_len": 800 },
    "scenario": "s1",
    "time_budget_s": 600
  },
  {
    "key": "q2",
    "qtype": "open",
    "title": "Budget",
    "body_md": "Stellen Sie einen groben Kostenplan auf.",
    "options_json": { "max_len": 800 },
    "scenario": "s1",
    "time_budget_s": 600
  },
  {
    "key": "q3",
    "qtype": "mc_single",
    "title": "Aufwärmfrage",
    "body_md": "Wie viele Museen hat die Stadt?",
    "options_json": {
      "choices": [ { "id": "a", "text_md": "3" }, { "id": "b", "text_md": "7" } ],
      "correct": ["b"]
    }
  }
]
}
],
"playlists": [
  {
    "name": "Projektwoche",
    "children": [
      {
        "name": "Einstieg",
        "items": [ { "question": "q3", "peer_instruction": true } ]
      },
      {
        "name": "Station 1 · Konzept und Budget",
        "kind": "complex",
        "scenario": "s1",
        "preamble_md": "Bearbeiten Sie die Aufgaben aus Sicht der Museumsleitung.",
        "items": [ { "question": "q1" }, { "question": "q2" } ]
      }
    ]
  }
]
}

```

Der Import legt zuerst die Themen, Szenarien und Fragen an und danach die Playlist mit Verweisen auf die neuen Fragen. Die Playlist erscheint auf der Seite **Playlists** ganz oben in der Liste. Importieren Sie die Datei

unter **Themen & Fragen**, kommen die Playlists ebenso mit; der Themenbaum landet dann im gewählten Ziel-Thema. Eine archivierte Playlist behält ihren Archivstand: LehrPuls schreibt dann am obersten Knoten `"archived": true`, und der Import legt sie wieder archiviert an. Ältere LehrPuls-Versionen übergehen das Feld und legen sie als gewöhnliche Playlist an.

14. Was der Import prüft

Der Import-Dialog zeigt vor dem Anlegen die Herkunft (`source_path`) und den Umfang der Datei — Themen, Fragen, Szenarien, Bilder, Kurse, Teilnehmende, Playlists und Fragen-Einträge. Beim Anlegen selbst wird die Datei als Fremd-Eingabe geprüft; die häufigsten Meldungen und ihre Ursachen:

Meldung	Ursache
Das ist keine LehrPuls-Exportdatei (weder ZIP noch JSON)	Datei beginnt nicht mit den ZIP-Kennbytes und enthält nirgends ein { – etwa reiner Text oder eine leere Datei
Das ist keine LehrPuls-Exportdatei (export.json fehlt)	export.json liegt nicht auf der obersten Ebene des Archivs
Das ist keine LehrPuls-Exportdatei	format ist nicht "lehrpuls-export"
Exportdatei-Fehler: ... in Zeile ..., Spalte ... (...)	JSON-Syntaxfehler („kein gültiges JSON“), fehlendes Pflichtfeld oder falscher Datentyp („unerwarteter Inhalt“); die Meldung nennt Zeile und Spalte, in Klammern steht der Wortlaut des Lesers, etwa missing field mit dem Namen des fehlenden Felds
Die Datei enthält keine Inhalte	alle vier Inhaltsblöcke sind leer
Unbekannter Fragetyp	qtype ist keiner der elf Werte
Optionen ungültig: unknown field ...	options_json enthält ein Feld, das der Fragetyp nicht kennt
i18n.en.options_json: unbekanntes Feld „max_len“	eine Sprachfassung enthält in options_json ein Feld, das kein Textfeld ist – max_len, url, ack und die Wertungsfelder stehen nur im options_json der Frage (Abschnitt 9)
Genau eine korrekte Option markieren, Mindestens zwei Antwortoptionen angeben ...	Regel des Fragetyps verletzt (siehe Abschnitt 6 und Abschnitt 7)
Mindestens eine Lücke {...} in den Fragetext schreiben	Lückentext ohne Lücke
Antwortoptionen brauchen einen Text oder ein Bild	eine Antwortoption hat kein text_md mit Inhalt
Szenario-Schlüssel ist in der Datei nicht definiert	scenario verweist auf einen Schlüssel, den kein Szenario der Datei trägt
Fragen-Schlüssel ist in der Datei nicht definiert	Playlist-Eintrag verweist auf einen key, den keine Frage trägt
Bilddatei passt nicht zu ihrem Namen	Hash der Datei im Archiv stimmt nicht mit dem Dateinamen überein
Nicht unterstützter Bildtyp	Endung ist nicht png, jpg, gif, webp oder svg
„...“ steht schon auf der Teilnehmerliste	zwei Teilnehmende mit demselben Anzeigenamen im selben Kurs

Bei nacktem JSON im Dialog „Fragen mit KI erstellen“ prüft LehrPuls jede Frage einzeln und zeigt alle Beanstandungen auf einmal; der Datei-Import bricht bei der ersten ab. In beiden Fällen bleibt der Bestand unverändert.

15. Vollständiges Beispiel

Ein Archiv `Physik.lpx` mit einem Bild:

```

Physik.lpx
├─ export.json
└─ media/
    └─ 9f86d081884c7d659a2feaa0c55ad015a3bf4f1b2b0b822cd15d6c15b0f00a08.png

```

Die zugehörige `export.json` zeigt jeden Fragetyp einmal, ein Szenario mit Bild, ein Unterthema, eine Playlist mit Fragenblock und einen Kurs:

```

{
  "format": "lehrpuls-export",
  "version": 2,
  "exported_at": "2026-09-15T09:00:00Z",
  "source_path": ["Physik"],
  "topics": [
    {
      "name": "Physik",
      "scenarios": [
        {
          "key": "s1",
          "title": "Versuchsaufbau Pendel",
          "body_md": "Ein Fadenpendel der Länge  $l = 1\text{ m}$  wird um  $5^\circ$ 
ausgelenkt.\n\n! [Versuchsaufbau]
(media/9f86d081884c7d659a2feaa0c55ad015a3bf4f1b2b0b822cd15d6c15b0f00a08.png mittel)"
        },
      ],
      "questions": [
        {
          "key": "q1",
          "qtype": "mc_single",
          "title": "Periodendauer",
          "body_md": "Wovon hängt die Periodendauer des Pendels bei kleiner Auslenkung
ab?",
          "options_json": {
            "choices": [
              { "id": "a", "text_md": "von der Masse" },
              { "id": "b", "text_md": "von der Fadenlänge" },
              { "id": "c", "text_md": "von der Auslenkung" }
            ],
            "correct": ["b"]
          },
          "solution_md": " $T = 2\pi\sqrt{l/g}$  – die Masse kommt nicht vor.",
          "difficulty": 2,
          "tags": "Mechanik, Schwingungen",
          "time_budget_s": 60,
          "scenario": "s1"
        },
        {
          "key": "q2",
          "qtype": "numeric",
          "title": "Periodendauer berechnen",

```

```

        "body_md": "Berechnen Sie die Periodendauer in Sekunden ( $g = 9.81 \frac{m}{s^2}$ ).",
        "options_json": { "answer": 2.01, "tolerance": 0.05, "unit": "s" },
        "solution_md": "$T = 2\pi\sqrt{\frac{1}{9.81}} \approx 2.01 \text{ s}$",
        "difficulty": 3,
        "time_budget_s": 120,
        "scenario": "s1"
    },
    {
        "key": "q3",
        "qtype": "mc_multi",
        "title": "Energieformen",
        "body_md": "Welche Energieformen wandelt das Pendel ineinander um?",
        "options_json": {
            "choices": [
                { "id": "a", "text_md": "kinetische Energie" },
                { "id": "b", "text_md": "potenzielle Energie" },
                { "id": "c", "text_md": "chemische Energie" }
            ],
            "correct": ["a", "b"]
        },
        "difficulty": 2
    },
    {
        "key": "q4",
        "qtype": "poll",
        "title": "Vorwissen",
        "body_md": "Hatten Sie Physik in der Oberstufe?",
        "options_json": {
            "choices": [ { "id": "a", "text_md": "Ja" }, { "id": "b", "text_md": "Nein" } ]
        }
    },
    {
        "key": "q5",
        "qtype": "poll_multi",
        "title": "Interessen",
        "body_md": "Welche Themen interessieren Sie? (Mehrfachauswahl)",
        "options_json": {
            "choices": [
                { "id": "a", "text_md": "Mechanik" },
                { "id": "b", "text_md": "Optik" },
                { "id": "c", "text_md": "Thermodynamik" }
            ]
        }
    },
    {
        "key": "q6",
        "qtype": "open",
        "title": "Fehlerquellen",

```

```

    "body_md": "Nennen Sie zwei Fehlerquellen bei der Messung der Periodendauer.",
    "options_json": { "max_len": 400 },
    "solution_md": "Reaktionszeit beim Stoppen, Luftreibung, zu große
Auslenkung.",
    "time_budget_s": 180
  },
  {
    "key": "q7",
    "qtype": "estimate",
    "title": "Fallzeit",
    "body_md": "Schätzen Sie: Wie lange fällt ein Stein aus 20 m Höhe?",
    "options_json": { "unit": "s", "reference": 2.02 }
  },
  {
    "key": "q8",
    "qtype": "order",
    "title": "Messablauf",
    "body_md": "Bringen Sie die Schritte in die richtige Reihenfolge.",
    "options_json": {
      "items": [
        { "id": "1", "text_md": "Pendel auslenken" },
        { "id": "2", "text_md": "Zehn Schwingungen stoppen" },
        { "id": "3", "text_md": "Zeit durch zehn teilen" }
      ],
      "correct_order": ["1", "2", "3"]
    }
  },
  {
    "key": "q9",
    "qtype": "matching",
    "title": "Größen und Einheiten",
    "body_md": "Ordnen Sie jeder Größe ihre Einheit zu.",
    "options_json": {
      "categories": [ { "id": "k1", "text_md": "Sekunde" }, { "id": "k2",
"text_md": "Meter" } ],
      "items": [
        { "id": "i1", "text_md": "Periodendauer", "category": "k1" },
        { "id": "i2", "text_md": "Fadenlänge", "category": "k2" }
      ]
    }
  },
  {
    "key": "q10",
    "qtype": "cloze",
    "title": "Formel ergänzen",
    "body_md": "Die Periodendauer wächst mit der {{Wurzel|Quadratwurzel}} der
Fadenlänge; verdoppelt man die Länge, wächst sie um den Faktor {{1,41|1.41|√2}}.",
    "options_json": {}
  },
  {
    "key": "q11",

```

```

    "qtype": "info",
    "title": "Versuchsanleitung",
    "body_md": "Die vollständige Versuchsanleitung finden Sie im Skript, Kapitel
4.",
    "options_json": { "url": "https://hochschule.example/physik/skript.pdf",
"ack": "yesno", "ack_yes": "Gelesen", "ack_no": "Später" }
    }
  ],
  "children": [
    {
      "name": "Übungsaufgaben",
      "questions": [
        {
          "qtype": "numeric",
          "title": "Pendel auf dem Mond",
          "body_md": "Welche Periodendauer hätte dasselbe Pendel auf dem Mond ( $g = 1{,}62\frac{\text{m}}{\text{s}^2}$ )?",
          "options_json": { "answer": 4.94, "tolerance": 0.1, "unit": "s" },
          "difficulty": 4,
          "scenario": "s1"
        }
      ]
    }
  ]
}
],
"playlists": [
  {
    "name": "Vorlesung 3 · Schwingungen",
    "children": [
      {
        "name": "Einstieg",
        "items": [ { "question": "q4" }, { "question": "q11" } ]
      },
      {
        "name": "Pendelversuch",
        "kind": "complex",
        "scenario": "s1",
        "preamble_md": "Werten Sie den Versuch in Zweiergruppen aus.",
        "items": [ { "question": "q1", "peer_instruction": true, "priority": 1 }, {
"question": "q2" }, { "question": "q6", "priority": 3 } ]
      }
    ]
  }
],
"courses": [
  {
    "name": "Physik 1",
    "semester": "WS 2026/27",
    "participants": [ { "display_name": "Anna Muster" }, { "display_name": "Ben
Beispiel" } ]
  }
]

```

```
}  
]  
}
```

16. Schritt für Schritt: eine .lpx-Datei von Hand bauen

1. Legen Sie einen Ordner an, darin die Datei `export.json` und den Unterordner `media`.
2. Schreiben Sie `export.json` nach den Abschnitten 3 bis 13 — am einfachsten, indem Sie das vollständige Beispiel aus Abschnitt 15 kopieren und kürzen. Speichern Sie als UTF-8 ohne Byte-Order-Mark.
3. Berechnen Sie für jedes Bild den SHA-256-Hash, benennen Sie die Datei in `<hash>.<endung>` um und legen Sie sie unter `media/` ab; referenzieren Sie sie im Markdown als `media/<hash>.<endung>`.
4. Packen Sie `export.json` und `media` — nicht den Ordner darüber — in ein ZIP-Archiv und geben Sie ihm die Endung `.lpx`.
5. Importieren Sie die Datei in LehrPuls über **Importieren (.lpx, .json)** und prüfen Sie in der Vorschau, ob Umfang und Herkunft stimmen. Schlägt der Import fehl, nennt die Meldung den Grund (Abschnitt 14); die Datenbank bleibt unverändert.
6. Am schnellsten lernt man das Format an einer echten Datei: Exportieren Sie ein kleines Thema mit einem Bild aus LehrPuls, benennen Sie die `.lpx`-Datei in `.zip` um und sehen Sie hinein.

Prüfliste

- `format` ist `"lehrpuls-export"`, `version` ist `2`; mindestens einer der Blöcke `topics`, `questions`, `courses`, `playlists` ist gefüllt.
- Jede Frage hat `qtype`, `title` und ein zum Typ passendes `options_json` ohne fremde Felder; Lückentexte haben mindestens eine Lücke im Text.
- Szenario-Schlüssel und Fragen-Schlüssel sind in der Datei eindeutig, und jeder Verweis zeigt auf einen definierten Schlüssel.
- Jedes Bild liegt als `media/<sha256>.<endung>` im Archiv, der Hash stimmt, die Endung ist `png`, `jpg`, `gif`, `webp` oder `svg`.
- `export.json` liegt auf der obersten Ebene des Archivs und ist UTF-8 ohne Byte-Order-Mark.
- Kurse haben einen Namen, Teilnehmende einen im Kurs eindeutigen Anzeigenamen; Dateien mit Teilnehmerlisten nur mit Rechtsgrundlage weitergeben.

Zusätzlich für maschinell erzeugte Dateien: Steht wirklich nur der JSON-Text im Ergebnis, ohne Einleitung, Nachwort oder Code-Zaun? Sind `format` und `version` wörtlich gesetzt? Trägt jede Frage einen nicht leeren `title` und ein `options_json` (notfalls `{}`)? Ist der Fragetyp einer der elf zulässigen Werte, und ist `cloze` vermieden? Sind alle Kennungen innerhalb einer Frage eindeutig und nicht leer? Enthält `options_json` in `i18n` nur Textfelder (`choices`, `items`, `categories`, `unit`, `ack_yes`, `ack_no`) — und fehlt es bei Freitextfragen dort ganz, also kein `max_len`? Steht keine Bildreferenz im Text?