

Ein bestehendes Projekt mit der neuesten Kit-Fassung abgleichen

Kit de documentation

Projekt — Aktualisierungsleitfaden — DE

1 Warum das Kit vollständig eingespielt wird

Dieses Dokument ist eine sinngemäße Übersetzung. Maßgeblich ist das Originaldokument auf Französisch; Erweiterungen und Änderungen werden stets dort eingepflegt.

Dieser Leitfaden ist für Sie, nicht für die *KI*. Er beschreibt, wie eine aktualisierte Fassung des Kits in ein bestehendes Projekt eingespielt wird, ohne die laufende Arbeit zu zerbrechen.

Das Kit entwickelt sich als Ganzes. Eine neue Fassung bringt nicht nur ein korrigiertes *Stylesheet* oder ein umformuliertes Dokument: sie führt eine vollständige Abhängigkeitskette mit sich. Ein *Stylesheet* wird stets mit seiner zugehörigen Reference geliefert, mit demselben *Zeitstempel*. Das *Registry* nennt die neue aktive Version. Der technische *Prompt* kann neue Regeln enthalten, die sich auf die neuen Funktionen stützen.

Eine Aktualisierung nach Belieben aufzunehmen — ein *Stylesheet* ohne seine Reference ersetzen, das *Registry* vergessen, einen alten *Prompt* behalten, der die neuen Regeln nicht kennt — erzeugt stille Unstimmigkeiten. Die *KI* liest dann Dokumente, die einander widersprechen, wendet Regeln an, die nicht mehr zu den verfügbaren Funktionen passen, und der Aufbau der erzeugten Dokumente verfällt ohne Warnung.

Einfache Regel: Eine Kit-Aktualisierung wird als Archiv in einem einzigen Schritt bezogen, unter der im *Registry* in `projectZip.downloadUrl` erklärten Adresse. Alle Dateien des Loses ersetzen ihre *Vorgänger* in einem Zug. Keine Auswahl nach Belieben, kein teilweises Ersetzen.

Diese Regel ist bewusst weiter gefasst als das strikt Nötige. Datei für Datei zu entscheiden, was nützlich ist, setzt Kenntnis der Abhängigkeiten voraus — und genau das soll Ihnen nicht abverlangt werden. Ein *paar* erklärende Dokumente zu viel im *Kontext* kosten nichts; eine fehlende Datei erzeugt eine Unstimmigkeit, die sich erst bei der nächsten *Erzeugung* zeigt.

💡 *Eine Aktualisierung ersetzt die Dateien des Kits und lässt Ihre stehen. Ihre laufende Arbeit bleibt erhalten, solange Sie keine Datei des Kits selbst geändert haben.*

2 Regel der Nicht-Verträglichkeit

Das Kit sichert keine Verträglichkeit zu, weder aufwärts noch abwärts. Jede neue Lieferung geht vom aktiven *Stylesheet* und seiner Reference aus, Skript from scratch. Die Skripte früherer Sitzungen werden weder herangezogen noch angepasst — für die laufende Sitzung bestehen sie nicht.

Das ist auch der Grund, weshalb die Kit-Dokumente keine Versionsnummer im Text tragen: es sind nie zwei Fassungen im Umlauf. Ein Dokument beschreibt den gegenwärtigen Stand, und die Geschichte lebt in den Changelog-Blöcken am Anfang der Code-Dateien.

3 Betroffene Dateien

Zwei Arten von Dateien bestehen in Ihrem Projekt nebeneinander: die Kit-Dateien, bei jeder Aktualisierung vollständig ersetzt, und die projekteigenen Dateien, nie angetastet.

- **Kit-Dateien — alles, gemeinsam:** eine Kit-Aktualisierung wird vollständig eingespielt, so wie sie geliefert wird. Die vollständige Liste steht in Einrichtungsleitfaden §3 — sie unterscheidet sich zwischen Einrichtung und Aktualisierung nicht. Keine Prüfung Datei für Datei nötig.
- **Dateien Ihres Projekts — unangetastet:** keine Datei mit Ihrem Präfix wird von einer Kit-Aktualisierung berührt. Sie bleiben genau so, wie Sie sie hinterlassen haben. Einzige Ausnahme: in Schritt 4 unten, wenn eine von der Aktualisierung eingeführte strukturelle Änderung eine Migration verlangt — und nur mit Ihrer ausdrücklichen Bestätigung, Migration für Migration.

4 Vorgehen

Die Aktualisierung erfolgt in fünf Schritten. Bei laufenden Aktualisierungen — Korrekturen, kleinere neue Funktionen — erfordert das Ersetzen der Dateien keine sofortige Sitzung: die Versionserkennung geschieht selbsttätig zu Beginn der nächsten Sitzung.

Ausnahme: bringt die Aktualisierung eine größere strukturelle Änderung mit — neuer Vertrag eines Datenmoduls, neues *Registry*-Flag, umgebauter Export —, ist eine Prüfsitzung vor der Wiederaufnahme der Dokumentarbeit verbindlich. Dann gilt das Vorgehen aus §4.4.

4.1 Schritt 1 — Die neuen Dateien beschaffen

Holen Sie die aktualisierten Kit-Dateien von Ihrer Bezugsquelle. Bewahren Sie die alten Dateien bis zum Ende des Vorgehens auf Ihrem Rechner auf.

 *Bewahren Sie stets die vorige Fassung des Stylesheets auf, bis die neue Fassung in einer Sitzung vollständig geprüft ist.*

4.2 Schritt 2 — Die Versionen prüfen

Prüfen Sie vor dem Ersetzen, dass das *Registry* zum heruntergeladenen Los gehört und dass die angegebenen Versionsnummern den tatsächlichen Versionen der neuen Dateien entsprechen.

Prüfung	Wo die Angabe steht
Version der neuen <i>Stylesheet</i>-Datei	Erste Zeilen der .js-Datei — der Kommentar "Version : x.xx".
Im <i>Registry</i> angegebene Version	Kit - <i>Registry.js</i> , Abschnitt stylesheets, Feld version jedes Stylesheets.
<i>Zeitstempel</i> des Paares	Kit - <i>Registry.js</i> , Abschnitt stylesheets, Feld ts. Muss für .js und Reference .docx gleich sein.

Eine *Stylesheet*-Datei trägt ihre Versionsnummer an zwei Stellen: dem Kopfkomentar, auf einen Blick lesbar, und der exportierten Konstante, die tatsächlich in jedes erzeugte Dokument eingetragen wird. Eine Abweichung zwischen beiden ist zur Laufzeit unsichtbar, verfälscht aber jede manuelle Prüfung. Der zugehörige Audit ist in Qualitätssicherung §4.2 beschrieben.

 *Das Registry ist die einzige Wahrheitsquelle für die aktiven Versionen. Eine Version wird nie aus dem Gedächtnis behalten: sie wird zu jedem Sitzungsbeginn im Registry nachgelesen.*

4.3 Schritt 3 — Im Projekt ersetzen

Löschen Sie in Ihrem Projekt die alten Kit-Dateien und laden Sie die neuen hoch, das *Registry* eingeschlossen. Die Reihenfolge spielt für diesen Schritt keine Rolle.

Zu Beginn der nächsten Sitzung wird das *Registry* gelesen und jede Abweichung zwischen den angegebenen und den tatsächlich im *Kontext* vorhandenen Versionen wird selbsttätig erkannt. Wurde das *Registry* nicht zusammen mit den Stylesheets hochgeladen, wird die Unstimmigkeit gemeldet und seine Ablage vor jeder *Erzeugung* verlangt.

4.4 Schritt 4 — Die Projektdateien anpassen

Eine Kit-Aktualisierung kann Änderungen mit sich bringen, die den Aufbau der *Projektdateien* berühren: neues Pflichtfeld in einem *Datenmodul*, neues Flag, umgebauter Export. In der ersten Arbeitssitzung nach der Aktualisierung gilt das folgende Vorgehen ausnahmslos vor jeder anderen Aufgabe.

- **Abweichungsdiagnose:** der Aufbau der bestehenden *Projektdateien* wird mit dem verbindlichen Aufbau von Gemeinsame Struktur verglichen. Jede Abweichung wird in der Form “Abweichung erkannt in [Datei]: [Beschreibung]. Vorgeschlagene Migration: [Handlung]” gemeldet und wartet auf Ihre ausdrückliche Bestätigung. Keine stille Änderung.
- **Datenmodule:** die modulspezifische Prüfung — Pflichtfelder, Konformität der Exporte, mehrsprachiges Format — folgt dem Vorgehen aus Gemeinsame Struktur §6.7.
- **Fehlende optionale Dateien:** für jede in Gemeinsame Struktur beschriebene und in Ihrem Projekt fehlende optionale Datei wird Ihnen die *Erzeugung* ausdrücklich angeboten. Nie eine stille *Erzeugung*.
- **Vorhandene optionale Dateien:** eine bereits vorhandene optionale Datei wird nie gelöscht, auch wenn ihr Flag false ist. Das Löschen einer bestehenden Projektdatei geschieht nur auf Ihre ausdrückliche *Anweisung*.

Hinweis: Keine Projektdatei wird jemals eigenmächtig gelöscht, aus welchem Grund auch

immer — abgeschaltetes Flag, für überholt gehaltene Datei oder anderes. Das Löschen ist immer Ihre Entscheidung, im Chat formuliert.

4.5 Schritt 5 — Die Darstellungsschwelle prüfen

Die meisten Aktualisierungen berühren das Aussehen bereits erzeugter Dokumente nicht. Manche schon: ein geänderter Abstand, eine geänderte Breite, eine ersetzte Farbe. Ihre alten Dokumente unterscheiden sich dann von Ihren neuen, und der *Ordner* wird zu einer Mischung aus zwei Layouts.

Das Kit zeigt diese Fälle über eine Schwelle an, `minDocumentVersion`, angegeben in Kit - *Registry.js* — *HTML-Pipeline* §3.1. Sie trägt die letzte Version, die die Darstellung geändert hat. Jedes erzeugte Dokument behält die Version, die es erzeugt hat.

- **Die Schwelle hat sich nicht bewegt:** nichts zu tun. Ihre Dokumente bleiben gültig, gleich welche Version sie erzeugt hat.
- **Die Schwelle ist gestiegen:** jedes von einer früheren Version erzeugte Dokument wird neu erzeugt. Der Inhalt ändert sich nicht — er wird aus der bestehenden Quelldatei gewonnen und von der aktiven Version dargestellt. Der Website-Generator weigert sich, ein Dokument unterhalb der Schwelle zu veröffentlichen, und nennt dabei den Dateinamen.

Hinweis: Eine Neuerzeugungskampagne wird Dokument für Dokument geprüft: *kit_check_markup.py* vergleicht die Auszeichnung zwischen der Quelldatei und dem neu erzeugten Dokument. Über einen ganzen Bestand hat diese Prüfung bereits ein einwandfrei gültiges Dokument erwischt, dessen Inhalt der eines anderen war.

5 Aufnahmesitzung

Öffnen Sie nach dem Ersetzen der Kit-Dateien im *Arbeitsbaum* ein neues *Gespräch* in diesem Projekt und fügen Sie den folgenden Text als erste Nachricht ein.

Dieser Text löst die geordnete Aufnahme der Aktualisierung aus: erneutes Lesen aller Wahrheitsquellen, Erkennen von Versionsabweichungen zwischen *Registry* und tatsächlichen Dateien, Anwendung des Vorgehens aus §4.4 auf Ihre bestehenden *Projektdateien* und Vorschlag der nötigen Migrationen, eine nach der anderen. Keine Ihrer Dateien wird ohne Ihre Bestätigung geändert.

```
AUFNAHME EINER KIT-AKTUALISIERUNG
=====
```

```
KONTEXT :
```

```
-----
```

```
Die aktualisierten Kit-Dateien wurden in den Arbeitsbaum
hochgeladen, das Registry eingeschlossen. Keine projekteigene
Datei darf ohne ausdrückliche Bestätigung geändert werden.
```

```
PFLICHTLEKTÜRE — STRENGE REIHENFOLGE :
```

```
-----
```

Vor jeder anderen Handlung in dieser Reihenfolge lesen :

1. Kit - Registry.js
 - Wahrheitsquelle der aktiven Versionen.
2. Kit - Projet - Prompt (TS).docx
 - dauerhafte technische Anweisungen ; kann neue Regeln enthalten.
3. Kit - Projet - Guide de mise à jour (TS).docx
 - dieser Leitfaden, insbesondere §4.4 Anpassungsvorgehen.
4. Kit - Projet - Structure commune (TS).docx
 - verbindlicher Vertrag der Projektdateien, Bezug der Abweichungsdiagnose.
5. Kit - Stylesheet - General - Code.js + Reference.docx
 - Hauptpaar.
6. Die übrigen Stylesheet-Paare (Glossary, YAML, HTML)
 - jeweils Code.js + Reference.docx.

VERSIONSERKENNUNG — PFLICHTBERICHT :

Für jedes Stylesheet vergleichen :

- in Kit - Registry.js angegebene Version
- am Kopf der .js-Datei gelesene Version
- von der .js-Datei exportierte Versionskonstante
- Zeitstempel des Paares (.js und Reference .docx)

Erwarteter Bericht je Stylesheet :

- Registry-Version : x.xx
- Version .js-Kopf : x.xx
- exportierte Konstante : x.xx
- Zeitstempel des Paares : JJJJ-MM-TT - HHhMM
- stimmig : ja / nein

ABWEICHUNGSDIAGNOSE DER PROJEKTDATEIEN :

Nach der vollständigen Kit-Lektüre den Aufbau der bestehenden Projektdateien mit dem verbindlichen Aufbau von Kit - Projet - Structure commune vergleichen. Jede Abweichung melden in der Form :

« Abweichung erkannt in [Datei] : [Beschreibung].
Vorgeschlagene Migration : [Handlung]. »

Alle Abweichungen auflisten, bevor irgendeine Migration angewandt wird. Nichts ändern, solange der vollständige Bericht nicht geliefert ist.

ABBRUCHBEDINGUNGEN :

Ist eine der folgenden Bedingungen erfüllt, vollständiger Halt und ausdrückliche Meldung :

- Kit - Registry.js fehlt oder ist nicht aktuell
- eine Kit-.js-Datei ohne ihre Reference .docx vorhanden,

```

oder umgekehrt, oder mit unstimmen Zeitstempeln
- Registry-Version und Version der .js-Datei weichen ab
- eine erwartete Kit-Datei fehlt im Arbeitsbaum
- eine bestehende Projektdatei widerspricht einem Vertrag
  von Structure commune auf mehrdeutige Weise

```

NACH VOLLSTÄNDIGEM BERICHT — AUF DAS GO WARTEN :

Keine Änderung an einer bestehenden Projektdatei ohne ausdrückliche Bestätigung, Migration für Migration. Keine Projektdatei wird eigenmächtig gelöscht, aus welchem Grund auch immer.

Bei fehlenden optionalen Dateien ausdrücklich fragen, ob sie zu erzeugen sind. Nie eine stille Erzeugung.

Antworten Sie nach dem Lesebericht und der Abweichungsdiagnose “go” für jede Migration, die Sie freigeben, oder “skip” für die, die Sie verschieben möchten. Nur freigegebene Migrationen werden angewandt.

6 Sonderfall — Sprachwechsel

Möchten Sie die Dokumentationssprache Ihres Projekts wechseln, ist nur eine Handlung nötig: das Sprachfeld in Ihrem *Registry* ändern, das dessen einzige Wahrheitsquelle ist, und die aktualisierte Datei in den *Arbeitsbaum* legen.

Alle ab dieser Sitzung erzeugten Dokumente verwenden die neue Sprache. Bestehende Dokumente werden nicht geändert. Das Umbenennen Ihrer Dateien entsprechend den Kategorien und Themen der neuen Sprache ist in Namenskonvention §5.4 beschrieben.

Nicht zu verwechseln mit der Veröffentlichung eines Dokuments in mehreren Sprachen, die eine Entscheidung je Dokument ist und die Projektsprache nicht berührt — Namenskonvention §2.4.

7 Was sich nie ändert

Einige Bestandteile Ihres Projekts sind von Kit-Aktualisierungen vollständig abgeschottet.

Hinweis: “Kit-X - Registry.js” und “Kit - Registry.js” ähneln sich in einer Dateiliste, und nur die zweite wird ersetzt. Lesen Sie nach, was Sie gerade überschreiben wollen, bevor Sie bestätigen: eine zu weite Auswahl nimmt Ihre Erweiterung mit dem Kit mit. Ihre Kit-X-Dateien vor jeder Aktualisierung zu sichern kostet eine Minute.

Bestandteil	Auswirkung einer Kit-Aktualisierung
Ihre <i>Projektdateien</i>	Keine — nie angetastet, nur auf Ihre ausdrückliche <i>Anweisung</i> geändert.
Ihre bestehenden Generatorskripte	Nicht wiederverwendbar — das Skript wird bei jeder Sitzung from scratch neu geschrieben.

Bestandteil	Auswirkung einer Kit-Aktualisierung
Ihre bereits erzeugten Inhaltsdokumente	Keine — es sind statische Dateien.
Ihr Glossar und Ihre <i>Datenmodule</i>	Keine — <i>Projektdateien</i> , ihr Vorhandensein hängt von Ihren Flags ab.
Ihre projekteigenen Konventionen	Keine — sie leben im eigenen Abschnitt Ihres Projekt-Prompts.
Ihre Kit-X-Dateien	Keine — das Präfix Kit-X ist Ihres. Es sind Ihre Erweiterungen des Kits: zusätzliche Stylesheets, Farbpalette, Werkzeuge. Sie gelten für alle Ihre Projekte und folgen Ihnen von einem zum anderen.