Stud.App für Ihren Stud.IP-Standort

Diese Seite richtet sich an Stud.IP-Administration, Rechenzentrum, Datenschutz und IT-Sicherheit. Für den Anschluss brauchen Sie genau eines: das Stud.IP-Plugin `app-connect` in Ihrer Instanz. Es verändert den Stud.IP-Kern nicht und legt den OAuth-2.0-Client für die Apps selbst an.

> **Auf einen Blick:** reiner Client · Stud.IP 6.0 bis 6.99 · ein Plugin, keine Kernänderung · OAuth 2.0 mit PKCE · keine Server des Herausgebers im Datenweg · quelloffen unter GPL 3.0 · keine Kosten, kein Vertrag.

Was die App tut, und was nicht

Die App …	Die App … nicht
zeigt Kurse, Termine, Dateien, Nachrichten, Ankündigungen, Blubber und Courseware	sendet keine Daten an Server des Herausgebers
spricht direkt mit der JSON:API Ihrer Instanz	ist kein Proxy und repliziert nichts
blendet Funktionen aus, die Ihre Instanz nicht anbietet	umgeht keine Berechtigungen, das entscheidet Ihr Server
speichert Tokens nur im Schlüsselbund des Geräts	sieht keine Passwörter, der Login läuft auf Ihrer Seite
nutzt Texterkennung und Ähnliches nur auf dem Gerät	enthält keine Tracker, Werbe- oder Crash-SDKs
ist kostenlos und kennt keinen Kauf	verlangt kein Konto beim Herausgeber

Voraussetzungen

Anforderung	Warum
Stud.IP 6.0 oder neuer (6.x), PHP ab 8.1	Spanne des Plugins; die App nutzt ausschließlich die JSON:API, 6.2 bietet den größten Umfang.
JSON:API aktiv unter `/jsonapi.php/v1/`	Einziger Datenkanal der App.
Plugin `app-connect` installiert und aktiviert	Liefert Instanzdaten und den OAuth-2.0-Client. Zur Laufzeit ohne Composer-Pakete.
HTTPS für die ganze Instanz, gültiges Zertifikat, TLS 1.3	Die App verweigert schwächere Verbindungen.
`$ABSOLUTE_URI_STUDIP` in `config/config_local.inc.php` gesetzt	Sonst leitet Stud.IP seine Adresse aus dem `Host`-Header ab; ein gefälschter Header käme in erzeugte Links.
Hinter Proxy oder Load-Balancer: echte Client-Adresse in `REMOTE_ADDR`	Die Kopplung zählt Fehlversuche je Adresse. Kommt alles von der Proxy-Adresse, sperrt sie für alle.
Laufender Cron-Worker	Der Aufräum-Job des Plugins löscht abgelaufene Codes, Links und Gerätezeilen.
Aus dem Internet erreichbar	Geräte sind überwiegend im Mobilfunk unterwegs.
Ansprechperson mit `root`-Zugang	Die Sektion „Stud.App“ in der Konfiguration ist `root`-Sache.

Einrichtung in drei Schritten

1

Plugin installieren

Admin → System → Plugins, Paket hochladen und aktivieren. Die Migrationen laufen bei Installation und Update (`cli/studip plugin:migrate app-connect`).

2

App-Verbindung einschalten

Als `root` Admin → System → Konfiguration, Sektion „Stud.App“, `APP_CONNECT_ENABLED` einschalten. Das Plugin legt dabei einen öffentlichen OAuth-2.0-Client ohne

3

Angaben setzen und testen

In derselben Sektion lassen sich Name und Kurzname, Logo und Akzentfarbe, Datenschutz-Adresse, Links, Support-Adresse, Mensa-Quelle und die einzelnen Funktionen

	Secret an (PKCE, `S256`) oder verwendet den bisherigen wieder. Bitte nicht in der OAuth-2.0-Verwaltung des Kerns bearbeiten, das Plugin verwaltet ihn. Es gibt nichts einzutippen: Adresse, Endpunkte und Client-ID holen die Apps aus der Discovery des Plugins. Alle Installationen nutzen dieselbe Redirect-Adresse. `de.studapp2.open:/oauth2`	setzen. Danach ein kurzer Test mit einigen Konten. Der Editor des Kerns nimmt jeden Wert an; das Plugin prüft nach dem Speichern und verwendet ungültige Werte nicht, sondern zeigt einen Hinweis.

> Ausschalten widerruft den Client: Angemeldete Apps verlieren sofort den Zugang. Erneutes Einschalten verwendet denselben Client wieder. Dasselbe gilt beim Deaktivieren oder Deinstallieren des Plugins. Serverseitig bleibt nichts zurück, das Sie aufräumen müssten.

Wie Angehörige ihr Gerät verbinden

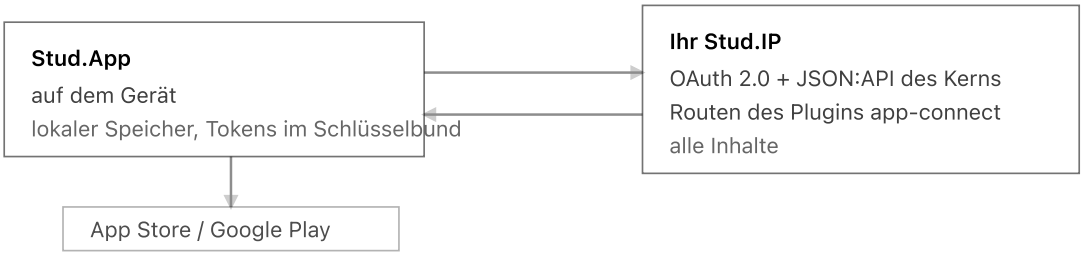
Es gibt kein zentrales Verzeichnis und keine Freischaltung beim Herausgeber. Die App fragt nach der Adresse Ihrer Instanz, liest deren Discovery und zeigt Name und Host zur Bestätigung. Danach stehen zwei Wege offen:

- ✓ **QR-Code** aus Profil → Einstellungen → „Stud.App Verbindung“ (ab Status `autor`): einmaliger Kopplungscode, fünf Minuten gültig, der erste Scan gewinnt. Gespeichert wird nur dessen Hash. Ist die App nicht installiert, führt der Code auf die öffentliche Seite „App holen“.
- ✓ **Anmeldung im Systembrowser** auf Ihrer eigenen Loginseite, einschließlich Single Sign-on, wenn Sie eines anbieten. Zurück kommt ein Code, den die App gegen Tokens tauscht (Authorization Code mit PKCE, `S256`).

Jede neue Verbindung schickt eine Mail an das Konto, mit Gerätename, Plattform, Zeitpunkt und einem Link zur Verwaltung. Dort sehen und entfernen Angehörige ihre Geräte selbst; Entfernen beendet den Zugang sofort, auch für später erneuerte Tokens. Fehlgeschlagene Kopplungsversuche sind auf 10 je Adresse in 10 Minuten und 100 je Minute insgesamt begrenzt. Der QR-Code ist ab Werk aus und wird eingeschaltet, sobald die Instanz eingerichtet ist.

Architektur und Datenflüsse

Alle Inhalte fließen zwischen Gerät und Ihrer Instanz. Ein Server des Herausgebers steht nicht im Datenweg: Die Apps rufen keinen auf. Die Stores liefern nur die App selbst aus.



Inhalte ausschließlich von Ihrer Instanz. Aus den Stores kommt nur das Programm.

Was das Plugin liefert

Alle Routen liegen unter `/jsonapi.php/v1/` und antworten `404`, solange die App-Verbindung aus ist. Die Discovery kündigt an, welche Funktionen eingeschaltet sind; die Apps nutzen eine Route nur dann. Diese Liste steuert, was die App anzeigt, und ist kein Zugriffsschutz: Die Routen des Kerns bleiben für angemeldete Personen ohnehin erreichbar.

Route	Anmeldung	Funktion
`app-connect/discovery`	keine	Öffentliche Instanzdaten aus einer festen Feldliste: Name, Adressen, Client-ID und Endpunkte, eingeschaltete Funktionen, Gestaltung, Links, Mensa-Quelle. Enthält nie Geheimnisse, `ETag` und fünf Minuten cachebar.
`app-connect/token`	Client-ID der App	Token-Endpunkt des App-Clients: Browser-Anmeldung mit PKCE, Refresh mit Gerätebindung, Kopplung per QR-Code. Andere Clients werden abgewiesen.
`app-connect/devices`	App-Token	Registriert das Gerät nach einer Browser-Anmeldung und schickt die Verbindungsmail.
`app-connect/course-dates/places`	ja	Wo bis zu 50 Termine stattfinden: gebuchte Räume mit Koordinaten und Raumfinder-Link, sonst die Freitext-Angabe. Termine ohne Leseberechtigung fehlen.
`app-connect/user-events`	ja	Der Kalender der Person für zwei Wochen wie im Web, mit der Farbgruppe je Veranstaltung und mit Ausfällen.
`app-connect/user-language`	ja	Die Sprache, in der die Person Stud.IP nutzt.
`app-connect/web-login`	App-Token	Einmaliger Link für die Anmeldung im Web, siehe unten.
`app-connect/me/ical-key`	App-Token	Kalender-Schlüssel der Person, anlegen und zurücksetzen, siehe unten.

Alles Weitere kommt aus der JSON:API des Kerns mit dem Token der Person, ohne Kernänderung: Veranstaltungen, Semester, Mitgliedschaften mit Farbgruppe, Termine, Stundenplan, Ordner und Dateien, Ankündigungen, Nachrichten, Blubber, Courseware sowie Bilder von Personen und Veranstaltungen.

> Mit eingeschaltetem `display_errors` landen PHP-Warnungen der Kern-Route `file-refs/{id}/content` in der heruntergeladenen Datei. Im Betrieb bitte ausgeschaltet lassen.

Anmeldung im Web aus der App

Öffnet die App eine Seite der Weboberfläche, etwa zum Eintragen in eine Veranstaltung oder über „In Stud.IP öffnen“, holt sie einen einmaligen Link (60 Sekunden gültig, nur als Hash

gespeichert, höchstens 10 je Person in 5 Minuten). Der Browser fragt „Als <Name> anmelden?“ und meldet erst nach Bestätigung an; ist dort jemand anderes angemeldet, sagt die Seite das vorher. Ein zweiter Faktor Ihres Identity Providers wird dabei nicht abgefragt, die Zwei-Faktor-Authentisierung von Stud.IP schon. Schalten Sie `APP_CONNECT_FEATURE_WEB_LOGIN` aus, wenn jede Web-Sitzung über Ihren Identity Provider laufen muss.

Kalender-Schlüssel

Stud.IP liefert den persönlichen Kalender ohne Sitzung aus; der Schlüssel ist die ganze Berechtigung. Die App holt ihn über das Plugin, damit ein Abo in einem Schritt im Gerätekalender landet, und kann ihn zurücksetzen. Neue Schlüssel erzeugt das Plugin selbst: 32 Zeichen aus einem kryptografischen Zufallsgenerator statt der acht Zeichen aus `rand()`, die der Kern auf der Kalender-Seite vergibt. Höchstens fünf neue Schlüssel je Person und Stunde. Ausgeschaltet (`APP_CONNECT_FEATURE_ICAL_KEY`) bleibt der Weg über die Kalender-Seite im Browser.

Was gespeichert wird, und wie lange

Alles liegt in Ihrer Datenbank. Ein Cronjob des Plugins („app-connect: Aufräumen“, täglich 03:17) löscht Kopplungscodes einen Tag nach Ablauf, Web-Login-Links einen Tag nach Ausstellung und Gerätezeilen einen Monat nach Ablauf des Refresh-Tokens. Wird ein Konto gelöscht, verschwinden seine Geräte, Codes und Links sofort.

Daten	Form
Geräte	in der App gewählter Gerätename, Plattform, Art der Verbindung, IDs der Tokens (nicht die Tokens), Zeitpunkte
Kopplungscodes	SHA-256-Hash, Person, Geräte-ID, Zeitpunkte
Web-Login-Links	SHA-256-Hash, Person, Ziel, Verweis auf Token und Gerät, Zeitpunkte
Kalender-Schlüssel	wie vom Kern verwaltet; das Plugin legt ihn an oder löscht ihn
Fehlversuche beim Koppeln	Zähler je gehashter Adresse, 10 Minuten im Cache
Einstellungen	Ihre Werte, Client-ID, gespeicherte Adresse

Die Personendaten-Auskunft von Stud.IP (Benutzerverwaltung → Aktionen) enthält Geräte, Kopplungscodes und Links einer Person, Codes und Links nur als Hash. Ein- und Ausschalten, Clientanlage, geänderte Einstellungen, Web-Anmeldungen und Kalender- Schlüssel stehen im Ereignisprotokoll (`APPCONNECT_*`), mit der Person, nie mit Link oder Schlüssel.

Datenschutz

Die Nutzung der App ist eine Verarbeitung **Ihrer Hochschule**, nicht des Herausgebers der App, wie beim Zugang über den Browser. Der Herausgeber erhält keine Daten: kein Konto, keine Kennung, keine Inhalte, keine Nutzungsereignisse. Ein Auftragsverarbeitungsvertrag ist deshalb nicht nötig.

Das Betreiberhandbuch des Plugins enthält einen fertigen Textbaustein für Ihre Datenschutzerklärung, mit Platzhaltern für Speicherdauer und Rechtsgrundlage. In der App sind Datenexport und vollständige Löschung der lokalen Daten jederzeit erreichbar. Was die App selbst auf dem Gerät speichert, steht in unserer Datenschutzerklärung.

> Nutzungsstatistik: Das Plugin kann der App ein eigenes Matomo Ihrer Hochschule ankündigen. Die ausgelieferten Apps nutzen das noch nicht, der Schalter bleibt verborgen und es wird nichts gesendet. Externe Tracker gibt es in keiner Fassung.

Betrieb und Support

Nach der Einrichtung entsteht für Sie kein laufender Betrieb. Beim Update des Plugins laufen die Migrationen mit. Bitte behalten Sie im Blick: eine neue Domain oder geänderte Pfade, eine Umstellung des Logins und größere Stud.IP-Upgrades. Möchten Sie Stud.App wieder entfernen, schalten Sie `APP_CONNECT_ENABLED` aus oder deinstallieren das Plugin; der Client wird dabei widerrufen.

Fragen zu Konto, Rechten und Kursinhalten beantwortet Ihr Support. Setzen Sie dafür `APP_CONNECT_SUPPORT_EMAIL`, dann bieten die Apps Ihre Adresse an. Fragen zur App selbst und Barrieren in der App gehen an uns.

Lizenz und Quellcode

Apps und Plugin stehen unter der **GPL 3.0 oder neuer**, mit Zusatzbedingungen nach Abschnitt 7, die den Hinweis auf Autor, Quellcode und Lizenz sichtbar halten. Der Quellcode wird mit der Veröffentlichung in den Stores zugänglich; bis dahin schicken wir Ihnen Plugin-Paket und Handbuch auf Anfrage zu.

Häufige Fragen

Kostet der Anschluss etwas? ×

Nein, und auch die App ist kostenlos. Es gibt keinen Kauf, kein Abo, keine Freischaltung und keinen Vertrag.

Ändert das Plugin etwas am Stud.IP-Kern?

Nein. Es bringt eigene Routen und Seiten mit, legt eigene Tabellen an und verwaltet einen OAuth-2.0-Client. Alles Weitere liest die App aus der JSON:API des Kerns. Den Kalender-Schlüssel legt es dort ab, wo der Kern ihn erwartet.

Ist das eine offizielle Stud.IP-App?

Nein. Stud.App ist ein unabhängiger Client, entwickelt im Austausch mit Mitgliedern des Stud.IP e. V. und der Community.

Funktioniert unser Single Sign-on?

Ja, sofern Ihre Loginseite es anbietet. Die Anmeldung läuft im Systembrowser auf Ihrer eigenen Seite.

Brauchen wir eine Firewall-Freigabe?

Nein. Die Geräte greifen wie ein Browser auf Ihre öffentlichen Stud.IP-Adressen zu.

Können wir den Zugang auf bestimmte Gruppen beschränken?

Ja, über Ihre Stud.IP-Rechte. Die App kann nur das, was das jeweilige Konto auch im Web darf. Den QR-Code finden Personen ab Status `autor`.

Was passiert bei einer Stud.IP-Wartung?

Die App zeigt zwischengespeicherte Inhalte weiter an und gleicht ab, sobald Ihre Instanz wieder erreichbar ist.

Können Angehörige die App ohne unser Zutun nutzen?

Nein. Ohne eingeschaltete App-Verbindung antworten die Routen des Plugins mit `404`, und ohne Plugin gibt es keinen Client für die Apps.

Anschluss anfragen

Schreiben Sie uns kurz, wer bei Ihnen die Stud.IP-Administration übernimmt und welche Version Sie betreiben. Wir melden uns mit Plugin-Paket, Betreiberhandbuch und den nächsten Schritten.

Stand: 3. November 2026