Easy!Appointments

Consolidated Official Documentation Companion

User Manual + Administration + Integrations + Operations + API Reference

Scope

A single, organized reference built from the current Easy!Appointments documentation center, the official user and administrator articles, and selected official 1.6.x release information. It is written as a consolidated companion rather than a verbatim copy of the website.

Prepared September 17, 2026

Official documentation: easyappointments.org/documentation

CHAPTER 0

About This Consolidated Manual

The official Easy!Appointments Help Center is split into separate user and administrator articles. The documentation center currently reports 13 articles in its Admin Manual collection and 6 articles in its User Manual collection. This companion brings the core material into one continuous reference, with added cross-links, checklists, workflow summaries, and a source index.

Copyright and accuracy note

This is not an official Easy!Appointments publication and does not reproduce the website word-for-word. It paraphrases and reorganizes the official documentation, preserves important commands and settings, and points back to the original pages for verification. Product behavior can vary by Easy!Appointments release and by local customization.

Current release context

As of this compilation date, the project home page offers Easy!Appointments v1.6.0 as the production download. Easy!Appointments 1.6.1-beta.1 was published on September 7, 2026 as a pre-release for testing and is explicitly not recommended for production use.

- Use this manual primarily for v1.6.0 production installations unless you intentionally run a pre-release.
- For older installations, consult the legacy versioned documentation archive before applying a command or configuration change.
- Before upgrades or structural changes, back up both application files and database data.

How this manual is organized

Section	Purpose
Part I	Daily use: configuration, roles, booking, calendar, settings, notifications.
Part II	Administration: installation, upgrades, translations, integrations, troubleshooting.
Part III	Technical operations: console, backups, API, operational checklists.
Part IV	Quick reference: decision trees, glossary, complete official source index.

Table of contents

PART I

User and Daily Operations

The material most business owners, receptionists, providers, and staff use every day.

CHAPTER 1

User Manual Overview

Official source: User Manual - Introduction

Easy!Appointments is an open-source web application for appointment scheduling. The official user manual is aimed at administrators, staff, technical users, and new users who need to configure and operate the platform. Its core topics are installation and configuration, services and providers, appointments, calendar and availability, customer information, notifications, and customization.

Typical adoption flow

Configure business -> **Add services** -> **Add providers** -> **Set availability** -> **Open booking page** -> **Manage appointments**

Who should use which parts of this manual?

Role	Recommended chapters
Business owner / administrator	Chapters 2-6, 7-18, and the operating checklists.
Receptionist / secretary	Chapters 3-6, daily workflow, provider calendar and customer management.
Provider	Roles, calendar, availability, appointments, calendar synchronization.
Technical administrator	Installation, update, console, backups, integrations, API, troubleshooting.

CHAPTER 2

Initial Configuration

Official source: Official Initial Configuration

After installation, configure the business before accepting real bookings. The official sequence is to log in to the backend, set global company information, define services, create provider accounts, set working plans and breaks, adjust booking rules, and run a test booking from the public page.

2.1 Access the backend

1. Open the Easy!Appointments installation in a browser.
2. Use the Backend link on the booking page, or navigate directly to the calendar backend route.

```
https://your-installation/index.php/calendar
```

3. Sign in with the administrator account created during installation.

2.2 Global business settings

Setting	Purpose
Company Name	Displayed in customer-facing communications.
Company Email	Default sender identity for system notifications.
Time Zone	Critical for accurate appointment times and external calendar synchronization.
Date / Time Format	Controls how dates and times are displayed.
Language	Default interface language.
Currency	Used where pricing or related features display monetary values.

2.3 Add services

Create one service record for each bookable offering. The official configuration article calls out the service name, duration, optional price, and calendar color. Service categories can be used to organize larger catalogs.

2.4 Add providers

Create each staff/provider record, assign the services that person can perform, define the weekly working plan, and add scheduled breaks. Providers receive their own calendar and credentials.

2.5 Scheduling rules

- Minimum notice: how soon before a time slot a customer is allowed to book.
- Maximum advance booking: how far into the future customers can schedule.
- Appointment statuses: define the status list used by your installation.

2.6 Validate before launch

4. Log out of the backend.
5. Open the public booking page as if you were a customer.
6. Book a test service and provider at an available time.
7. Confirm that the booking appears in the backend calendar and that notifications behave as expected.

Best practice

Do not start with real customers until you have tested every service, each provider schedule, at least one cancellation/reschedule scenario, and outbound email.

CHAPTER 3

Understanding User Roles

Official source: Official User Roles

Easy!Appointments separates access by role. The official documentation identifies Administrator, Provider, Customer, and Secretary behavior. Roles are assigned to backend users and should reflect the minimum access each person needs.

Role	What it can do
Administrator	Full system access. Manages users, settings, services, providers, integrations, and all appointments.
Provider	Manages personal availability and own appointments. Can access customer information tied to assigned appointments and can synchronize a calendar when enabled.
Secretary	Manages one or more assigned providers, their appointments, and relevant customer information, but cannot change system-wide settings.
Customer	Books through the public interface, manages their own appointment actions, and has no backend access.

Access-control rule

The official user-role article states that a single user cannot hold multiple roles. Assign carefully and avoid giving administrator access to staff who only need calendar or provider-management functions.

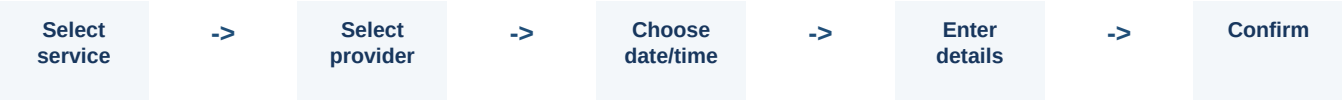
CHAPTER 4

Booking and Managing Appointments

Official source: Official Booking and Managing Appointments

4.1 Customer booking flow

Public booking workflow



Customers can book from the public page without registering a normal backend account. They choose a service, optionally choose a provider, select an available date and time, enter configured contact fields, and confirm the appointment.

4.2 Backend calendar

- Calendar views include day, week, and month.
- Services can be color-coded for faster scanning.
- Appointments can be filtered by provider or service.
- Selecting an appointment exposes its details and editing controls.

4.3 Create an appointment manually

1. Open Calendar.
2. Select an empty time slot.
3. Choose or create the customer.
4. Choose the service and provider.
5. Set date/time; service duration can populate the end time.
6. Choose an appointment status and save.

4.4 Edit or cancel

Open the appointment from the calendar. To edit, change fields and save. To cancel, change the status to the configured cancellation status and save. Customer/provider notifications depend on system settings and the action being performed.

4.5 Statuses

Typical examples in the official guide include Pending, Confirmed, Cancelled, and Completed, but the application lets the administrator configure the actual status list.

CHAPTER 5

Application Settings

Official source: Official Application Settings

Application settings control global behavior, booking rules, customer-facing fields, integrations, and privacy-related options. Administrators reach them from the backend Settings area.

5.1 General settings

Setting	Effect
Company Name	Displayed on booking pages and notifications.
Company Email	Default sender address used by the application.
Company Link	Business URL included in customer-facing communications.
Default Language	Default language for the booking page and users.
Default Timezone	Reference timezone for scheduling and synchronization.
First Day of Week	Controls calendar week layout.

5.2 Business logic

Setting	Effect
Working Plan	Defines days and hours during which bookings can occur.
Blocked Periods	Closes defined periods for all providers.
Reschedule / cancellation limit	Controls how near to the appointment a customer can still change it.
Future Booking Limit	Limits how many days ahead public bookings can be made.
Appointment Statuses	Defines the status choices used on the calendar.

5.3 Booking and privacy settings

- Toggle standard customer fields and decide whether they are required or optional.
- Add up to five custom customer fields.
- Enable or disable customer notifications.
- Limit provider/secretary access to customers they have appointments with.
- Enable CAPTCHA on public booking/update actions.
- Allow the customer to select Any Provider.
- Show or hide the login button on the booking page.
- Allow personal-information deletion where supported.
- Disable the public booking page temporarily.

5.4 Integrations

The settings area exposes integration controls for Google Calendar, analytics, the REST API, and LDAP. CalDAV integration is handled through the provider calendar flow and newer 1.6.1 pre-release builds add further CalDAV settings.

CHAPTER 6

Appointment Notifications

Official source: Official Appointment Notifications

Easy!Appointments can send automated email notifications when appointments are created, updated, or cancelled. The usual recipients include the customer, assigned provider, secretaries, and administrators, subject to configuration and the action being performed.

Notification	Typical purpose
Confirmation	Sent after a successful booking. Typically includes date/time, service, provider, and reschedule/cancel information when enabled.
Update	Sent when appointment data changes, such as time or service.
Cancellation	Communicates that an appointment was cancelled and can include the original schedule and cancellation context.

If email is missing

The official notification guide says to verify that the server can send mail, check error logs, and test SMTP configuration. Email transport is configured outside the normal Admin Panel in the application email configuration file.

PART II

Administration and Integrations

Install, maintain, integrate, and troubleshoot the Easy!Appointments server.

CHAPTER 7

Administrator Overview

Official source: Official Admin Manual Introduction

The Admin Manual is intended for people responsible for installing, configuring, managing, and maintaining Easy! Appointments. Its official scope includes upgrades, global settings, languages/time zones, users, services, availability, locations, notifications, integrations, backups, and system security.

Administrator lifecycle

Install	->	Configure	->	Secure	->	Integrate	->	Back up	->	Monitor / troubleshoot

CHAPTER 8

Installation

Official source: Official Installation Guide

The official installation article describes a traditional web-server deployment. It currently lists Apache 2.4, PHP 7.2+, and MySQL 5.7 as minimum components, with php_curl needed for Google Calendar integration. For a new 1.6.x deployment, confirm requirements against the current release package because supported versions continue to evolve.

8.1 Installation sequence

1. Prepare a compatible web server, PHP runtime, and MySQL database server.
2. Create a database and obtain its connection credentials.
3. Upload the Easy!Appointments source files to the intended web directory.
4. Ensure the storage directory is writable by the web application.
5. Configure the root config.php with the installation URL and database connection values. Add calendar integration credentials if needed.
6. Open the installation URL in a browser, complete the installation form, and create the administrator account.

8.2 Initial business configuration after install

The installation guide recommends setting the business working plan, creating services, adding providers and their service assignments, optionally creating secretary users, then validating the public booking page with a test appointment.

Deployment note

If you customize code, themes, or templates, keep a written inventory of modified files. That inventory becomes important during upgrades, because official update instructions replace application files.

CHAPTER 9

Updating an Installation

Official source: Official Update Guide

Before anything else

Back up the database and application files before updating. This is an explicit step in the official update guidance.

9.1 Standard update flow

1. Back up database and files.
2. Replace application files/directories while preserving the root config.php and any known custom work that must be reapplied.
3. Run database migrations.

```
php index.php console migrate
```

The official documentation also describes a browser-based update route. Exact routes have changed across releases; for 1.6.x, prefer the release notes and CLI migration command when you have shell access.

9.2 If an update produces HTTP 500

Check the Easy!Appointments storage/logs directory first, then the web-server error log. A failed migration, permissions issue, incompatible PHP runtime, or local custom code can all surface as a 500 response.

CHAPTER 10

Managing Translations

Official source: Official Manage Translations

Easy!Appointments supports multiple languages and custom translations. The current application has a modern localization system, while the official translation article also documents the underlying language-file approach used by earlier versions.

10.1 Operational guidance

- Prefer the translation management features provided by your installed release before editing core files manually.
- When editing translation files, copy the base language and preserve all required keys.
- After upgrades, compare custom language files against the new default files so new strings are not missing.
- If you make custom UI translations, keep them under version control or in your deployment backup.

Version caution

The translation article contains paths and configuration examples that originated in older Easy!Appointments releases. Use the principles, but confirm file locations in your installed version before editing.

CHAPTER 11

Google Calendar Synchronization

Official source: Official Google Calendar Sync

Easy!Appointments supports two-way synchronization with Google Calendar. A provider can connect a Google Calendar so changes in Easy!Appointments and Google Calendar remain coordinated.

11.1 Prerequisites

- Working Easy!Appointments installation.
- At least one configured service and provider.
- Google account and Google Cloud project with Calendar API enabled.

11.2 Configure Google credentials

1. Create a Google Cloud project.
2. Enable the Google Calendar API.
3. Configure the OAuth consent screen.
4. Create a Web Application OAuth client.
5. Set the authorized origin to your Easy!Appointments domain and the redirect URI to the OAuth callback used by your installation.
6. Place the Google client ID and secret in the Easy!Appointments configuration required by your release.

11.3 Connect a provider

7. Open Calendar in the Easy!Appointments backend.
8. Select the provider.
9. Choose Enable Sync and complete Google authorization.

Important limits from the official guide

Each provider connects to one Google Calendar. Recurring events are not fully supported. Synchronization can be triggered by application changes, manually in the backend, and automatically through scheduled console synchronization.

CHAPTER 12

CalDAV Synchronization

Official source: Official CalDAV Calendar Sync

Since Easy!Appointments 1.5, providers can use two-way CalDAV synchronization with compatible calendar servers. Unlike Google Calendar, CalDAV normally uses the target server URL and account credentials directly rather than a separate third-party API project.

12.1 Required connection information

- Final Calendar URL
- Owner username
- Owner password

12.2 Connect

1. Open Calendar and select the provider.
2. Choose Enable Sync and select CalDAV when prompted.
3. Enter the calendar URL and credentials, then save.

Official limitations

One CalDAV calendar can be associated with a provider. Recurring events are not fully supported. Synchronization is triggered from the backend and by appointment-plan changes; scheduled sync can be automated with the console.

CHAPTER 13

Webhooks

Official source: Official Configuring Webhooks

Webhooks send HTTP POST messages to another system when supported records are saved or deleted. This lets Easy!Appointments notify CRMs, messaging systems, analytics services, and custom automation without modifying the core application.

Event	Meaning
Save	Triggered when a supported record is created or updated.
Delete	Triggered when a supported record is deleted.

13.1 Configure a webhook

1. Sign in as an administrator and open Settings > Integrations > Configure Webhooks.
2. Create a webhook record with a clear name.
3. Select the event(s) to send.
4. Enter the destination URL.
5. Configure a secret header or token for request validation.

Security

Use HTTPS and validate a secret/token on the receiving system. Treat webhook payloads as potentially sensitive because they can contain appointment-related data.

CHAPTER 14

LDAP Integration

Official source: Official LDAP Integration

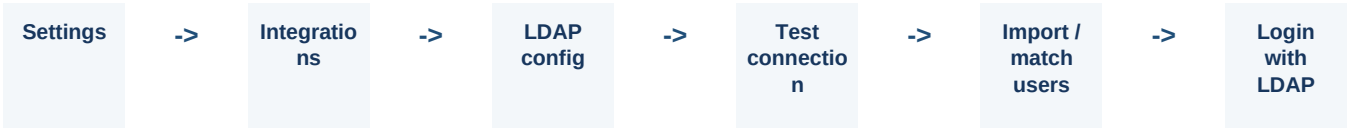
LDAP integration lets staff and administrator users authenticate against a central directory such as Active Directory or OpenLDAP. Customers continue to use the public booking workflow and do not require LDAP accounts.

14.1 Prerequisites

- Reachable LDAP or Active Directory server.
- Correct server/network/firewall path from Easy!Appointments to LDAP.
- Host, base DN, bind information, and user search details.

14.2 Setup flow

LDAP setup



Security recommendations

Use LDAPS or StartTLS, use a read-only service account for directory binding where possible, and rotate the bind password regularly.

CHAPTER 15

Docker Development Environment

Official source: Official Docker Guide

The official Docker article describes the project development environment. It is explicitly intended to make development easier rather than serve as production guidance.

15.1 Development startup

```
docker compose up
```

The documented development stack includes the Easy!Appointments application, MySQL, phpMyAdmin, Mailpit, Baikal for CalDAV testing, OpenLDAP, and phpLDAPAdmin. The sample config points the database host to the MySQL service name inside the Compose network.

> **Production warning**
>
> Do not treat the development Compose configuration as a hardened production deployment. The official documentation directs production Docker users to the dedicated Easy!Appointments Docker image/project.

CHAPTER 16

Troubleshooting

Official source: Official Troubleshooting Guide

The official troubleshooting strategy starts with the environment, then debug information, then application and web-server logs. This order avoids wasting time on application code when the underlying hosting environment changed.

16.1 Troubleshooting order

1. Verify the server environment and recent hosting/runtime changes.
2. Temporarily enable Easy!Appointments debug mode when additional diagnostics are required. Disable it again after troubleshooting so sensitive details are not exposed.
3. Inspect application logs under storage/logs.
4. Inspect the web server error log, commonly under /var/log/nginx/ or /var/log/apache2/ on Linux.
5. Reproduce the problem once while watching logs so timestamps and stack traces line up with the failing action.

Production safety

Debug mode can expose detailed internal information. Use it temporarily and never leave verbose debugging enabled on an Internet-facing production installation.

PART III

Technical Operations

CLI, backups, REST API, email, automation, and security.

CHAPTER 17

Console / CLI

Official source: Official Console Guide

Easy!Appointments introduced command-line support in v1.4.0. Commands are invoked through the root index.php file and are useful for migrations, testing, installation, backups, synchronization, and automation.

Command	Purpose
php index.php console migrate	Bring the database schema to the latest migration state.
php index.php console migrate fresh	Reset migration changes and rebuild from the beginning. Destructive for normal production data; use only when you intentionally want a fresh state.
php index.php console seed	Create test records such as an admin, provider, customer, and service.
php index.php console install	Perform a command-line installation after config.php is prepared.
php index.php console backup	Create an application-data backup under storage/backups.
php index.php console backup /path/to/folder	Write backup output to a specified directory.
php index.php console sync	Trigger calendar synchronization for all providers.
php index.php console help	Show available console functionality.

Production caution

The "migrate fresh" and seed commands are development/test-oriented operations. Do not run destructive reset commands on a production database unless you explicitly intend to rebuild it.

CHAPTER 18

Backup and Recovery Operations

Official source: Official Console guide

Official source: Official automated-backup article

The console backup command exports Easy!Appointments application data to a backup location. The official backup article describes the default output as a gzipped SQL backup under storage/backups. For real disaster recovery, protect both the database backup and the application/configuration files.

18.1 Recommended backup set

- Database backup generated by the Easy!Appointments console or an equivalent tested database backup process.
- Root config.php and any environment-specific configuration.
- Email configuration and integration credentials stored outside the database.
- Custom themes, templates, translation files, extensions, or local code modifications.
- Any web-server/reverse-proxy configuration required to make the site reachable.

18.2 Backup command

```
php index.php console backup
```

18.3 Recovery principle

A backup is only useful if it can be restored. Periodically restore a copy into a non-production environment, verify database migrations, sign in, view the calendar, and create a test booking. Document the exact restore procedure for your hosting stack.

CHAPTER 19

REST API

Official source: Official REST API Guide

The Easy!Appointments v1 REST API exchanges JSON and supports standard REST operations. The official documentation supports Basic Authentication with administrator credentials and also describes a configurable API key that can be sent as a Bearer token. Production API access should always use HTTPS/TLS.

19.1 HTTP operations

Method	Typical operation
GET	Read resources.
POST	Create resources.
PUT	Update existing resources.
DELETE	Remove resources.

19.2 Query helpers

- q - search within a resource.
- page and length - paginate results.
- fields - return only selected fields.
- with - include related resources where supported.

19.3 Authentication example

```
curl https://your-installation/index.php/api/v1/appointments --user username:password
```

Security

Do not send Basic Authentication over plain HTTP. Use TLS, restrict administrative credentials, and prefer a dedicated API key/token workflow where it fits your deployment.

CHAPTER 20

FAQ and Common Problems

Official source: Official FAQ

The official FAQ collects practical setup questions such as verifying Apache/PHP/MySQL prerequisites, creating Google Calendar credentials, and diagnosing failed installation pages.

20.1 Environment verification

The FAQ calls out Apache, PHP, MySQL, the PHP curl extension, and Apache mod_rewrite as common prerequisites for the traditional stack. On managed hosting, confirm these with the provider or inspect the PHP environment.

20.2 Google OAuth problems

If Google Calendar authorization fails, verify that the Calendar API is enabled, the OAuth client is the correct application type, and the redirect URI exactly matches the callback URL used by the Easy!Appointments installation.

20.3 Installation page fails

Verify the installation base URL, database credentials, rewrite configuration, browser developer-console errors, and web-server logs. A failed AJAX request during installation often points to server or rewrite configuration rather than the browser page itself.

CHAPTER 21

Email Delivery and SMTP

Official source: Official Appointment Notifications

Official source: Official email-delivery troubleshooting article

Appointment notifications only help if the server can actually deliver mail. Easy!Appointments supports custom SMTP configuration for environments where the default PHP mail path is missing, blocked, or unreliable.

21.1 When mail does not arrive

1. Confirm the appointment was successfully created/updated and that notifications are enabled.
2. Check Easy!Appointments logs and the web-server/PHP mail errors.
3. Verify the sender address and SMTP server settings.
4. Confirm SMTP host, port, encryption mode (TLS/SSL as appropriate), authentication username/password, and network/firewall access.
5. Send a controlled test booking and inspect both application logs and the receiving mailbox spam/quarantine.

CHAPTER 22

Scheduled Tasks and Cron

Official source: Official cron-jobs article

Scheduled tasks are useful for recurring calendar synchronization and database backups. On Linux/macOS, the official article demonstrates cron; on Windows, the equivalent is Task Scheduler.

22.1 Hourly examples from the official article

```
0 * * * * /usr/bin/php /path/to/easyappointments/index.php console backup
0 * * * * /usr/bin/php /path/to/easyappointments/index.php console sync
```

Operational note

Choose a frequency that matches your business needs and calendar provider limits. Confirm the scheduled job runs as the intended OS user and has permission to read the application and write backup/log files.

CHAPTER 23

Security Baseline for 1.6.x

Official source: Official Easy!Appointments 1.6.0 release

Official source: Official 1.6 security overview

Easy!Appointments 1.6.0 places greater emphasis on request validation, booking protection, calendar-integration safety, password recovery, privacy, and automated cleanup. A secure deployment still depends on the surrounding server configuration and maintenance practices.

23.1 Practical baseline

- Run the current production release and apply maintenance/security updates after testing.
- Use HTTPS for booking, backend login, API calls, webhooks, and external calendar integrations.
- Use CAPTCHA/ALTCHA protection where appropriate for public forms.
- Limit administrator accounts and assign providers/secretaries only the access they need.
- Protect configuration files and secrets from web exposure and source-control leaks.
- Use LDAPS/StartTLS for LDAP and secret validation for webhooks.
- Monitor application and web-server logs, and do not leave debug mode enabled in production.
- Back up regularly and test recovery.

1.6.1 status

The official 1.6.1 beta published September 7, 2026 contains additional security and synchronization fixes, but the project explicitly labels it a pre-release and says not to use it in production. Test it separately if you want to evaluate the fixes.

PART IV

Operating Playbooks and Reference

Practical checklists built from the official documentation.

CHAPTER 24

First-Day Setup Checklist

- [ ] Confirm production Easy!Appointments version and server requirements.
- [ ] Create/verify database and application backup.
- [ ] Configure Company Name, Company Email, timezone, date/time format, language, and company link.
- [ ] Create every service with correct duration, price (if used), and calendar color.
- [ ] Create each provider; assign services, weekly working plan, breaks, and days off/exceptions.
- [ ] Create secretary accounts only for staff who need multi-provider management.
- [ ] Configure booking limits, statuses, custom fields, CAPTCHA, Any Provider, and customer privacy options.
- [ ] Configure SMTP and verify confirmation/update/cancellation emails.
- [ ] Configure Google Calendar or CalDAV only after provider schedules are correct.
- [ ] Create a full test booking, edit it, reschedule it, cancel it, and confirm the expected emails.
- [ ] Test public booking on desktop and mobile.
- [ ] Record the exact backup and update procedure for this installation.

CHAPTER 25

Daily Front Desk Workflow

Daily front-desk loop

Opening routine

- Review today and tomorrow in Calendar.
- Check provider absences, blocked periods, and special working-plan exceptions.
- Confirm any externally synchronized calendars are current if you rely on them for conflicts.

During the day

- Create walk-in/phone appointments directly in Calendar.
- Use the correct service, provider, start time, and status.
- When rescheduling, save the change and confirm the intended customer notification behavior.
- When cancelling, use the configured cancellation status rather than deleting records unless deletion is truly intended.

Closing routine

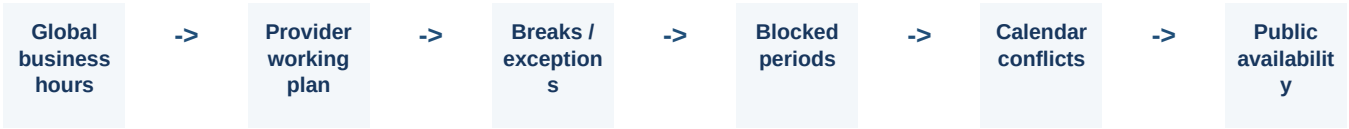
- Review unresolved/pending appointments.
- Confirm next-day schedule and provider availability.
- Investigate failed email or calendar sync warnings while logs are recent.

CHAPTER 26

Provider and Availability Management

Provider availability is a combination of assigned services, normal working hours, breaks, days off/exceptions, global blocked periods, booking rules, and any external calendar data that participates in synchronization.

How availability is built



Change-control rule

When changing provider working hours, immediately test the public booking page for at least one affected date. If calendar synchronization is enabled, run or wait for synchronization and confirm the change did not create an unexpected conflict.

CHAPTER 27

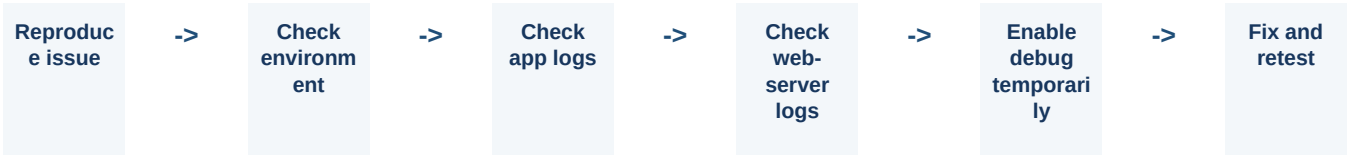
Integration Health Checklist

Integration	Health check
Google Calendar	OAuth token valid; correct redirect URI; provider sync enabled; test create/update/delete in both directions.
CalDAV	Calendar URL reachable; credentials valid; provider connected to correct calendar; large calendar sync tested.
Webhooks	HTTPS endpoint; secret validation; save/delete events received; retries/logging handled by receiver.
LDAP	Encrypted connection; bind account restricted; user DN mapping correct; login tested with non-admin account.
REST API	TLS enforced; API key/admin credentials protected; only required clients allowed; error responses monitored.
Email	SMTP reachable; sender accepted; spam/DNS policy checked; confirmation/update/cancel tests successful.

CHAPTER 28

Troubleshooting Decision Tree

General troubleshooting



Booking page problem

Check provider/service assignments, working plan, blocked periods, booking limits, timezone, and any synchronized external calendar conflicts.

Email problem

Check notification settings, SMTP transport, application logs, PHP/web-server logs, and the receiving mailbox spam/quarantine.

Calendar sync problem

Check provider sync status, OAuth/CalDAV credentials, remote service availability, clock/timezone correctness, network/TLS errors, and sync logs. Run console sync manually to reproduce under observation.

HTTP 500 after update

Check storage/logs, the web-server error log, database migration state, PHP compatibility/extensions, file permissions, and any locally modified code that was reapplied.

CHAPTER 29

Quick Reference

29.1 Key backend areas

Area	Primary purpose
Calendar	View, create, edit, reschedule, cancel, and filter appointments.
Services	Define bookable offerings, duration, price, category/color.
Providers / Users	Assign services, credentials, schedules, breaks, and role access.
Customers	View customer records created through bookings or backend entry.
Settings	Global business logic, booking behavior, customer fields, privacy, integrations.
Integrations	Google Calendar, webhooks, LDAP, API and related connection settings depending on release.

29.2 CLI commands

```
php index.php console migrate
php index.php console seed
php index.php console install
php index.php console backup
php index.php console sync
php index.php console help
```

29.3 Log locations

- Easy!Appointments application logs: storage/logs/*
- Typical Nginx error log: /var/log/nginx/error.log
- Typical Apache error log: /var/log/apache2/error.log

CHAPTER 30

Glossary

Term	Meaning
Administrator	Backend user with full system control.
Provider	Staff member who performs one or more services and owns an availability/calendar schedule.
Secretary	Backend user who manages appointments for assigned providers.
Customer	Person who books a service through the public page; no normal backend access.
Service	Bookable offering with a configured duration and optional price/category/color.
Working Plan	Recurring weekly schedule that defines normal working hours.
Break	Non-bookable time within a provider working period.
Blocked Period	Global closure interval that can prevent bookings across providers.
Working Plan Exception	Date-specific override to a normal provider schedule.
Google Calendar Sync	Two-way provider-calendar synchronization through Google Calendar API.
CalDAV	Open calendar protocol used for two-way synchronization with compatible calendar servers.
Webhook	HTTP POST notification from Easy!Appointments to an external system when configured events occur.
LDAP	Directory protocol used for centralized staff/admin authentication and user import.
REST API	HTTP/JSON interface for programmatic access to Easy! Appointments resources.
Migration	Database schema update applied when moving to a newer application version.
Cron job	Scheduled command on Unix-like systems, commonly used for sync and backup automation.

CHAPTER 31

Official Source Index

The following official pages were used as the basis for this companion. The original pages remain authoritative for wording, screenshots, release-specific changes, and corrections.

- **Documentation Center:** https://easyappointments.org/documentation/
- **Admin Manual collection:** https://easyappointments.org/documentation_category/admin-manual/
- **User Manual collection:** https://easyappointments.org/documentation_category/user-manual/
- **Admin Manual - Introduction:** https://easyappointments.org/documentation/introduction/
- **Installation:** https://easyappointments.org/documentation/installation/
- **Update:** https://easyappointments.org/documentation/update/
- **Manage Translations:** https://easyappointments.org/documentation/manage-translations/
- **Google Calendar Sync:** https://easyappointments.org/documentation/google-calendar-sync/
- **CalDAV Calendar Sync:** https://easyappointments.org/documentation/caldav-calendar-sync/
- **Configuring Webhooks:** https://easyappointments.org/documentation/configuring-webhooks/
- **LDAP Integration:** https://easyappointments.org/documentation/ldap-integration/
- **Docker:** https://easyappointments.org/documentation/docker/
- **Troubleshooting:** https://easyappointments.org/documentation/troubleshooting/
- **Console:** https://easyappointments.org/documentation/console/
- **REST API:** https://easyappointments.org/documentation/rest-api/
- **FAQ:** https://easyappointments.org/documentation/faq/
- **User Manual - Introduction:** https://easyappointments.org/documentation/introduction-2/
- **Initial Configuration:** https://easyappointments.org/documentation/initial-configuration/
- **Understanding User Roles:** https://easyappointments.org/documentation/understanding-user-roles/
- **Booking and Managing Appointments:** https://easyappointments.org/documentation/booking-and-managing-appointments/
- **Application Settings:** https://easyappointments.org/documentation/application-settings/
- **Appointment Notifications:** https://easyappointments.org/documentation/appointment-notifications/
- **Current product home / download page:** https://easyappointments.org/
- **Easy!Appointments 1.6.0 release:** https://easyappointments.org/blog/easyappointments-v1-6-0-release/
- **Easy!Appointments 1.6.1 beta release:** https://easyappointments.org/blog/easyappointments-1-6-1-beta-release/
- **Legacy versioned documentation archive:** https://easyappointments.org/docs.html

CHAPTER A

Appendix A - Easy!Appointments 1.6.0 Highlights

Official source: Official 1.6.0 Release

Version 1.6.0 is the production download presented on the Easy!Appointments home page at the time this manual was compiled. Major additions and improvements described by the project include:

- Built-in Jitsi integration and Google Meet support for online meetings.
- ALTCHA/CAPTCHA options and broader security hardening.
- Provider-controlled synchronization options and improved Google/CalDAV reliability.
- A meeting-link field on appointments.
- More deliberate customer-notification prompts around create/update/delete actions.
- Expanded GDPR/privacy controls.
- Multi-date working-plan exceptions.
- Improved password recovery and authentication flows.
- Automatic cleanup of old logs, cache, and sessions.
- UI consistency, dark-theme fixes, and booking/calendar improvements.

Upgrade discipline

After moving to 1.6.0, test calendar integrations, online-meeting behavior, email notifications, booking privacy, and custom themes before declaring the upgrade complete.

CHAPTER B

Appendix B - Easy!Appointments 1.6.1 Beta Notes

Official source: Official 1.6.1 Beta Release

Easy!Appointments 1.6.1-beta.1 was released September 7, 2026. The project describes it as a maintenance and security-focused pre-release and explicitly instructs users not to run it in production. It includes additional synchronization isolation, validation improvements, security fixes, CalDAV configuration changes, and clearer error handling.

- Google Calendar and CalDAV synchronization failures are handled more independently.
- Validation is tightened for service values and request parameters.
- CalDAV receives a dedicated settings area and improved connection diagnostics.
- Additional CAPTCHA/authentication and appointment-authorization security fixes are included.

Production status

Keep production on the current stable release unless you intentionally operate a test environment. Evaluate the beta separately and wait for the project to publish a stable 1.6.1 before normal production adoption.

End of consolidated reference

easyappointments.org/documentation