

Readiness report — OpenEMS Energy Management System

Checked 2026-09-03 15:46 · 10 searches, 18 pages fetched

Readiness: OpenEMS Energy Management System

Verdict: Ready with caveats · 2026-09-03 · est. parts cost — (no parts specified)

Cybersecurity — software & network: Do not place on an open network

This is the cold check for this lab — it establishes the baseline for future drift comparisons. OpenEMS is a pure software platform: the materials contain no schematics, no PCB or Gerber files, and no bill of materials, so there is nothing to source and no part can go obsolete, out of stock, or overpriced between now and the next check. The verdict is not a flat "Ready" for two reasons. First, the lab defines no hardware at all — the instructor must supply a host machine, and while the requirement is genuinely easy (the project's build sets `java_source` and `java_target` to 21, and the Getting Started guide states "OpenEMS requires Java 21 to run", so any Java 21-capable PC or single-board computer works), it is still an undefined decision the instructor has to make. Second, one of the lab's own hosted resources has already rotted: the OpenEMS API Javadoc link returns HTTP 404. Separately, and more urgently than anything in the parts picture, the Cybersecurity section below should be read before this lab is connected to any network.

Action required

Part	Problem	Recommended action
—	No parts are specified in this lab; the bill of materials is empty by design.	None — all sourcing is out of scope for this lab.

Non-part items needing attention this run:

Item	Problem	Recommended action
OpenEMS API Javadoc link	Returns HTTP 404 — dead link in the lab materials.	Find the current Javadoc entry point from the live documentation site, or generate Javadoc locally from the checked-out source, and update the materials.
Member Security Advisory Category (forum)	Returns HTTP 404 to anonymous visitors; reachable only with OpenEMS Association membership, if at all.	Do not rely on it in the lab procedure unless the instructor holds membership and has confirmed access.

Item	Problem	Recommended action
Gitpod live demo · Codecov dashboard	Hosts answer but serve JavaScript-only pages; contents could not be verified by fetch.	Open both in a browser before class rather than assuming the live demo still launches.

Cybersecurity — software & network

Do not place on an open network — The lab's documented setup ships default admin/admin credentials on the Apache Felix configuration console and a Backend mode that accepts any password, so as built it has no working authentication on a network-facing service.

Component	Version	How known	Support status
OpenEMS Edge	not determined	not determined (No version string is declared for Edge in the fetched build files (gradle.properties declares only java_source/java_target).)	—
OpenEMS Backend	not determined	not determined (No version string is declared for Backend in the fetched build files (gradle.properties declares only java_source/java_target).)	—
OpenEMS UI (openems-ui)	2026.10.0-SNAPSHOT	read from project (ui/package.json 'version' field (fetched/01-package.json); same value is the lockfile root version in ui/package-lock.json.)	—
Java runtime (Eclipse Temurin)	21	read from project (gradle.properties java_source/java_target = 21 (fetched/05-gradle.properties); the Getting Started guide states 'OpenEMS requires Java 21 to run'.)	supported until 2029-12-31
TypeScript	6.0.3	read from project (ui/package.json devDependencies (fetched/01-package.json).)	—
Angular framework (@angular/core and the pinned @angular/* set)	22.1.3	read from project (ui/package.json dependencies (fetched/01-package.json).)	—
@types/node	26.3.0	read from project (ui/package.json declares the range ^26.3.0; ui/package-lock.json resolves it to exactly 26.3.0 (fetched/02-package-lock.json).)	—
@ionic/angular	8.8.18	read from project (ui/package.json declares ^8.8.18; ui/package-lock.json resolves it to exactly 8.8.18 (fetched/02-package-lock.json).)	—
@capacitor/core	8.5.0	read from project (ui/package.json declares ^8.5.0; ui/package-lock.json resolves it to exactly 8.5.0 (fetched/02-package-lock.json).)	—

Component	Version	How known	Support status
ajv (transitive, under @ngx-formly/schematics)	8.9.0	read from project (ui/package-lock.json, node_modules/@ngx-formly/schematics/node_modules/ajv (fetched/02-package-lock.json).)	—
JaCoCo coverage toolchain (org.jacoco:*)	0.8.14	read from project (gradle.lockfile (fetched/03-gradle.lockfile).)	—
OW2 ASM bytecode library (org.ow2.asm:*)	9.9	read from project (gradle.lockfile (fetched/03-gradle.lockfile).)	—

Component	Finding	Severity	Fix available?
ajv (transitive, under @ngx-formly/schematics)	The copy of ajv 8.9.0 that the UI build pulls in under @ngx-formly/schematics can be forced into a very slow pattern match when it validates a schema that uses the \$data option, which stalls the build tool rather than anything students reach over the network. — GHSA-2g4f-4pwh-qvx6	moderate	No — @angular-devkit/core nested under @ngx-formly/schematics pins ajv 8.9.0 exactly, so the top-level project cannot raise it to the fixed 8.18.0
OpenEMS Backend (Metadata.Dummy)	The Backend configuration the guide walks through uses Metadata.Dummy, which the documentation states accepts any user and password combination, so the Backend has no working authentication as the lab builds it.	posture	Mitigation only — switch to a real Metadata implementation, as the documentation instructs, before the Backend is reachable from anywhere but the bench
OpenEMS Edge / OpenEMS Backend (Apache Felix Web Console)	The lab's own setup guide has students log in to the Apache Felix Web Console — the console that configures every OpenEMS component — with the username admin and the password admin, on port 8080 for Edge and port 8079 for Backend.	posture	Mitigation only — change the Felix Web Console password during setup and keep ports 8080 and 8079 off any shared or campus network
OpenEMS Edge to Backend link	The Edge-to-Backend connection the guide configures is an unencrypted websocket (ws://localhost:8081), so telemetry and control traffic between the two halves of the lab is sent in the clear.	posture	Mitigation only — keep both halves on the same isolated machine or segment for the lab, and use an encrypted wss:// endpoint for anything beyond it

Component	Finding	Severity	Fix available?
OpenEMS UI	In the Edge-only setup the guide describes, anyone who opens the UI can log in as guest by leaving the password blank, or gain extended permissions simply by typing admin into the password field.	posture	Mitigation only — configure real user accounts before the UI is reachable by anyone outside the bench, and keep it on an isolated network segment
OpenEMS UI dependency manifest	Dependencies unpinned: 95 of 114 in ui/package.json, and the lab materials shipped no package-lock.json, so an instructor installing from the lab's copy gets whatever npm resolves on the day rather than the versions checked here.	posture	Mitigation only — install from the repository's ui/package-lock.json (npm ci) instead of resolving ui/package.json fresh

What this means for your lab

Built the way its own Getting Started guide describes, this system is not safe to put on a shared network. The administrative console that configures every part of OpenEMS is protected by the username admin and the password admin, and the guide tells students to log in that way — anyone who can reach port 8080 (Edge) or 8079 (Backend) can therefore reconfigure the whole installation, including the control logic that would drive real batteries, inverters and chargers. The web interface is no better: a visitor can sign in as a guest with no password at all, or get elevated permissions by typing admin into the password box, and the Backend setup in the guide accepts any username and password you give it. Traffic between the Edge and Backend halves is unencrypted, so anyone on the same network can read the telemetry going past. None of this is a defect the project is hiding — the documentation itself says this setup is for testing only — but it does mean the lab must be run on an isolated bench network or a single offline machine, never on campus Wi-Fi or a building network, and never wired to real energy equipment while in that state. Before the lab runs, change the console password, set up real user accounts, and talk to campus IT if the machine needs to be reachable from anywhere else. One moderate-severity issue was found in a build-time library used to generate UI code; it affects the developer's build, not anything a student reaches over the network, and it cannot be upgraded from this project because a dependency pins the affected version. Note also that a large part of the UI's locked dependency list was beyond this run's automatic check limit, so a clean result here is not a statement that everything in it is clean.

Not checked

- OpenEMS Edge: No version is declared in the project's build files, so no advisory database could be queried for it.
- OpenEMS Backend: No version is declared in the project's build files, so no advisory database could be queried for it.

- openems-ui 2026.10.0-SNAPSHOT: Not published to npm — the registry has no package by that name, so no advisory database covers the application's own release.
- 709 of the 1217 locked npm dependencies in ui/package-lock.json: Beyond the 500-row automatic OSV check cap for one manifest this run.
- Java runtime (Eclipse Temurin 21) vulnerabilities: Only the support lifecycle was looked up; no CVE database was queried for the JDK build itself.
- Listening ports of a running installation: Ports were read from the setup documentation, not observed on a running system; a real deployment may differ.
- openems-ui 2026.10.0-SNAPSHOT: no package by this name in the registry
- 709 further dependencies in package-lock.json: beyond the 500-dependency automatic check cap (e.g. @typescript-eslint/tsconfig-utils, @typescript-eslint/type-utils, @typescript-eslint/typescript-estree, @typescript-eslint/utils, @typescript-eslint/visitor-keys)
- java 21: support status: endoflife.date has no such product

Parts

No parts. The snapshot states: "No PCB layout files, Gerber packages, schematic captures, or physical hardware bills of materials are provided in the project files. OpenEMS is a pure software platform." There are no BOM entries to table, and no lookups were dispatched.

Item	Spec/Package	Status	Stock	Price	Source
(no bill of materials)	—	not applicable	—	—	—
Host computing platform (industrial Linux gateway, single-board computer, or PC)	Not specified by the lab; must run a Java 21 runtime	not checked — no part specified	—	—	—
Integrated field equipment (energy storage inverters, PV arrays, meters, charging controllers)	Not specified by the lab; connected over field communication networks	not checked — no part specified	—	—	—

Substitutes

None needed.

No substitute was proposed this run, so there is nothing to verify against a datasheet. Where substitutes do appear in future reports, check each one against its datasheet before ordering.

Other dependencies

Every external link and hosted resource the snapshot flags was fetched this run.

- **OpenEMS Official Documentation** — live. [Introduction](#) retrieved (4,239 characters), current and describing the Edge / UI / Backend stack the lab uses.
- **OpenEMS Getting Started Guide** — live. [Getting Started](#) retrieved (8,000 characters). This is the page the lab procedure leans on most heavily.
- **OpenEMS API Javadoc** — **dead, HTTP 404**: <https://openems.github.io/openems.io/javadoc/> no longer resolves. The single piece of link rot found this run.
- **OpenEMS Community Forum** — live and active. community.openems.io shows topics dated 2–3 September 2026, including "Release 2026.9.0 · OpenEMS/openems". The support channel students are pointed to is healthy.
- **Member Security Advisory Category** — **not reachable anonymously, HTTP 404**: <https://community.openems.io/c/openems-association/security-advisories-members-only/23>. Consistent with a members-only category hidden from unauthenticated visitors, but this check cannot distinguish hidden from deleted.
- **Public Security Advisories** — live. [The repository's advisory page](#) states "There aren't any published security advisories", with a repository advisory count of 0. Recorded as the baseline: anything appearing here by the next check is new.
- **GitHub Actions Build Workflow** — live and running. [build.yml](#) lists recent "Build OpenEMS" runs up to #7928 on the develop branch, so upstream CI is healthy and pre-built releases should keep appearing.
- **Zenodo Archive / Scientific Reference DOI** — live. [10.5281/zenodo.4440883](https://doi.org/10.5281/zenodo.4440883) resolves to the archived record, latest deposited version 2026.9.0, published 31 August 2026. The citation the lab offers works.
- **Code Coverage Dashboard** — reachable but not verifiable. codecov.io/gh/openems/openems returned only "You need to enable JavaScript to run this app." The host answers; whether it still holds OpenEMS coverage data needs a browser to confirm.
- **Gitpod Live Demo Environment** — reachable but not verifiable, same result: "You need to enable JavaScript to run this app." Whether a workspace still launches for this repository could not be established. Gitpod's free-tier terms have changed repeatedly — try it before class.
- **Security Reporting Contact (security@openems.io)** — not checked. An email address is not a fetchable resource, and no tool here can confirm a mailbox is monitored without sending mail to it.
- **Source repository** — live. github.com/OpenEMS/openems was listed successfully on the develop branch this run and five project files were fetched from it; the code the lab downloads is where the materials say it is.

Fabrication — none. The snapshot confirms no custom PCB or physical fabrication data exists in the repository: no board dimensions, layer counts, copper weights, or drill specifications.

Watch list

- **The Javadoc URL** — already 404 this run and will stay broken until the materials are updated. The highest-value fix available to the instructor.
- **The Gitpod live demo** — unverifiable by fetch and dependent on a third-party free tier that has changed before; the live link most likely to fail outright next.
- **The members-only advisory category** — 404 anonymously; if the lab expects students to read it, that expectation is already unmet.
- **Upstream release drift** — the archived release moved to 2026.9.0 on 31 August 2026 while the materials track a 2026.10.0-SNAPSHOT development build. Expect the lab's pinned dependency versions to drift from upstream within a release or two.
- **No parts to watch** — no obsolescence or stock risk exists for this lab, and none will appear unless a future revision adds hardware.