

AuroraStack Calendar Web Lite User Manual

AuroraStack Calendar Web Lite User Manual

AuroraStack Calendar Web Lite is the standalone events calendar for websites that do not use WordPress. It is a browser-based Web Component: you upload the CSS, JavaScript, and a JSON event file, then place one custom HTML tag where the calendar should appear.

Web Lite is suitable for static sites, ordinary HTML pages, React/Vue/Angular applications, and website builders that allow custom HTML and script files.

Table Of Contents

1. [What Web Lite Includes](#what-web-lite-includes)
2. [Package Contents](#package-content)
3. [Install On A Website](#install-on-a-website)
4. [Create Your First Calendar](#create-your-first-calendar)
5. [Event Data Format](#event-data-format)
6. [Add, Edit, Or Delete Events](#add-edit-or-delete-events)
7. [Import Events From CSV](#import-events-from-csv)
8. [Calendar View](#calendar-view)
9. [List View](#list-view)
10. [Image And Calendar View](#image-and-calendar-view)
11. [Customise The Look And Feel](#customise-the-look-and-feel)
12. [Event Icons And Downloads](#event-icons-and-downloads)
13. [Seasons And Month Navigation](#seasons-and-month-navigation)
14. [Mobile And Tablet Behaviour](#mobile-and-tablet-behaviour)
15. [Accessibility And Security](#accessibility-and-security)
16. [Troubleshooting](#troubleshooting)

What Web Lite Includes

Web Lite includes:

- Month calendar view.
- Upcoming-events list view.
- Image plus calendar view with image top, left, or right.
- JSON event loading from a file.
- Inline JSON event loading for simple pages.
- CSV parsing helper for customer-side import tools.

- Event icons from image URLs.
- Linked details pages.
- Post-event download links from public URLs.
- Optional manual WhatsApp share buttons.
- Multiple season definitions.
- Responsive phone, tablet, and desktop layouts.
- Configurable width, height, cell size, colours, fonts, button colours, event icons, list-card sizing, and alignment.

Web Lite does not include the Web Manager, licence checks, Hall of Fame, registrations, payments, protected downloads, WhatsApp Cloud API messaging, backend reminders, or server-side spreadsheet/DOCX imports. Those are Web Pro features.

Package Contents

The Lite ZIP contains the files needed to embed the component:

- `aurorastack-calendar.css`: visual styles.
- `aurorastack-calendar.js`: the Web Component.
- `edition.js`: marks the package as Lite.
- `edition.json`: package metadata.
- `examples/index.html`: a working example page.
- `examples/events.json`: sample data.
- `docs/LITE-DOCUMENTATION.md`: this manual.
- `docs/AuroraStack-Calendar-Web-Lite-User-Manual.pdf`: PDF copy of this manual.
- `docs/screenshots/`: Web Lite screenshots used by the manual.

Install On A Website

1. Unzip the package.
2. Upload `aurorastack-calendar.css`, `edition.js`, `aurorastack-calendar.js`, and your `events.json` file to the same public folder.
3. Add the stylesheet in the page `<head>`.
4. Add the two script files before the closing `</body>` tag, or use `defer`.
5. Add `<aurorastack-calendar>` where the calendar should display.

Basic install:

```
<link rel="stylesheet" href="/calendar/aurorastack-calendar.css">
<script src="/calendar/edition.js"></script>
<script src="/calendar/aurorastack-calendar.js" defer></script>

<aurorastack-calendar src="/calendar/events.json" view="calendar"></aurorastack-calendar>
```

If your files are in the same folder as the page:

```
<link rel="stylesheet" href="aurorastack-calendar.css">
<script src="edition.js"></script>
<script src="aurorastack-calendar.js" defer></script>
```

```
<aurorastack-calendar src="events.json"></aurorastack-calendar>
```

Create Your First Calendar

1. Copy `examples/events.json`.
2. Rename it to `events.json`.
3. Edit the `settings.title` value.
4. Replace the sample events with your own events.
5. Upload the file beside the Web Component files.
6. Open the page in a browser and confirm the calendar loads.

Minimal `events.json`:

```
{
  "settings": {
    "title": "Community Events",
    "defaultView": "calendar",
    "showWhatsApp": false,
    "width": "1100px",
    "cellWidth": "150px",
    "cellHeight": "170px"
  },
  "events": [
    {
      "id": "community-meeting",
      "title": "Community Meeting",
      "start": "2026-09-10T19:00:00",
      "end": "2026-09-10T20:30:00",
      "location": "Community Room",
      "description": "Monthly planning meeting for members.",
      "detailsUrl": "https://example.com/events/community-meeting"
    }
  ]
}
```

Event Data Format

Each event can use these fields:

- `id`: stable event ID. Use one unique value per event.
- `title`: event title.
- `start`: start date and time in ISO format, such as `2026-09-10T19:00:00`.
- `end`: end date and time.
- `location`: optional location.
- `description`: event description.
- `iconUrl`: optional JPG or PNG icon URL.
- `detailsUrl`: optional page URL for more event details.
- `registerUrl`: optional registration page URL.
- `paymentUrl`: optional payment page URL.
- `resources`: optional post-event files.
- `calendar`: optional named calendar value. In Lite this is stored in the event data but advanced multi-calendar filtering is a Pro feature.

Post-event resources:

```
"resources": [  
  {  
    "label": "Programme PDF",  
    "url": "https://example.com/downloads/programme.pdf"  
  },  
  {  
    "label": "Recording",  
    "url": "https://example.com/downloads/recording.mp4"  
  }  
]
```

Multi-day events use one start and one end. If an event starts on Friday and ends on Sunday, Web Lite shows it on Friday, Saturday, and Sunday in both calendar and list views. The same event icon is used on every displayed day.

Add, Edit, Or Delete Events

Web Lite does not include a browser manager. You manage events by editing the JSON file.

To add an event:

1. Open `events.json`.
2. Add a new object inside the `events` array.
3. Give it a unique `id`.
4. Set `title`, `start`, and `end`.
5. Add optional `location`, `description`, `iconUrl`, `detailsUrl`, and `resources`.
6. Save the file and upload it to the website.

To edit an event:

1. Find the event by `id` or `title`.
2. Change the required fields.
3. Save and upload `events.json`.
4. Refresh the page.

To remove an event:

1. Delete that event object from the `events` array.
2. Make sure the JSON still has valid commas.
3. Save and upload the file.

For non-technical site owners, create a spreadsheet that exports JSON through your website build workflow, or upgrade to Web Pro and use the Web Manager.

Import Events From CSV

Web Lite includes helper functions that can parse CSV in browser-based tools. It does not include a hosted import screen.

Recommended CSV columns:

title,start,end,location,description,icon_url,details_url
Community Meeting,2026-09-10 19:00,2026-09-10 20:30,Community Room,Monthly planning meeting,https://example.com

Accepted event icon columns are:

- icon_url
- event_icon_url
- event_icon
- image_url
- event_image
- image
- icon

Accepted details page columns are:

- details_url
- post_url
- url

If you need an import screen, spreadsheet upload, DOCX parsing, or stored imports, use Web Pro.

Calendar View

Calendar view shows one month at a time.



Web Lite calendar view

Use:

```
<aurorastack-calendar src="events.json" view="calendar"></aurorastack-calendar>
```

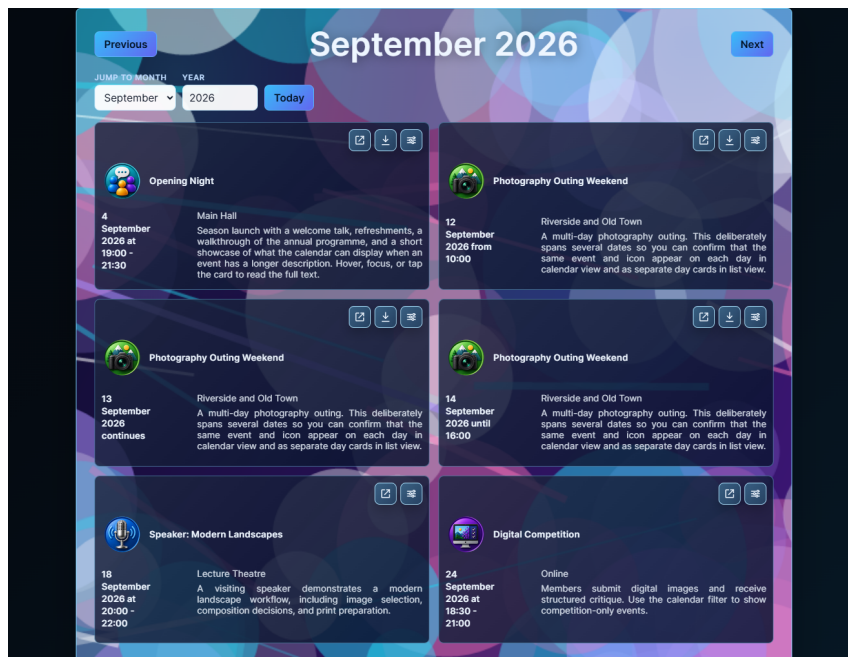
Useful calendar settings:

- width: overall calendar width, for example "1180px".
- height: optional calendar height.
- cellWidth: minimum day-cell width.
- cellHeight: minimum day-cell height.
- forceCenter: centre the calendar in the host page.
- weekStart: monday or sunday.
- showSync: show or hide sync controls.
- showWhatsApp: show or hide manual WhatsApp share buttons.

The calendar expands to fit the configured cell width where the screen has room. On narrow screens it switches to a phone-friendly stacked layout.

List View

List view shows one navigable month at a time. It groups the visible events under the selected month and gives previous/next controls plus month/year jump controls.



Web Lite list view

Use:

```
<aurorastack-calendar src="events.json" view="list"></aurorastack-calendar>
```

Useful list settings:

- listCardWidth: card width in pixels.
- listCardHeight: card height in pixels, or 0 for automatic height.
- listColumns: 1, 2, 3, or auto.
- fitListWidthToCards: sizes the list container around the configured card layout on wider screens.

When the list uses one column, cards can be wider, up to around 1200px. Multi-column layouts keep each card smaller so two or three columns can fit cleanly.

List cards use a compact action area and show the full description in a popover or mobile bottom sheet when the visible description is clipped.

Image And Calendar View

Image and calendar view displays a hero image with the month grid.

Use:

```
<aurorastack-calendar
  src="events.json"
  view="image_month"
  image-position="top">
</aurorastack-calendar>
```

Image placement options:

- top: image above the grid.
- left: image to the left of the grid on wide screens.
- right: image to the right of the grid on wide screens.

Relevant settings:

- heroImage: JPG or PNG URL.
- imagePosition: top, left, or right.
- backgroundImage: background image for the calendar.
- backgroundSize: cover, contain, stretch, repeat, or auto.
- backgroundPosition: for example center center.

Customise The Look And Feel

Add settings inside the settings object.

Example:

```
{
  "settings": {
    "title": "Galaxy Season Calendar",
    "defaultView": "calendar",
    "width": "1180px",
    "cellWidth": "150px",
    "cellHeight": "170px",
    "fontFamily": "Georgia, serif",
    "accent": "#7dd3fc",
    "textColor": "#f8fafc",
    "mutedColor": "#cbd5e1",
    "cellBackground": "rgba(15, 23, 42, 0.55)",
    "emptyCellBackground": "rgba(15, 23, 42, 0.22)",
    "eventBackground": "rgba(8, 47, 73, 0.88)",
    "eventTextColor": "ffffff",
    "buttonBackground": "#38bdf8",
    "buttonText": "#061018",
    "iconSize": 64,
    "iconBackground": "transparent",
    "forceCenter": true
  }
}
```

Common settings:

- title: calendar heading.
- listTitle: list-view heading.
- defaultView: default view if the embed does not set view.
- fontFamily: general font.
- numberFontFamily: day-number font.
- eventFontFamily: event-card font.
- buttonFontFamily: button font.
- accent: main accent colour.
- textColor: main text colour.
- mutedColor: quieter text colour.
- cellBackground: day-cell background.
- emptyCellBackground: empty-day background.
- eventBackground: event-card background.
- eventTextColor: event-card text.
- buttonBackground: primary button background.
- buttonBackground2: secondary button background.
- buttonText: button text.
- arrowBackground: previous/next arrow button background.
- arrowText: previous/next arrow button text.
- borderImage: optional border image URL.
- borderWidth: border width.

Useful embed attributes:

```
<aurorastack-calendar
  src="events.json"
  view="list"
  width="1200px"
  cell-width="160"
  cell-height="180"
  list-card-width="560"
  list-card-height="280"
  list-columns="2"
  force-center="true"
  show-whatsapp="false">
</aurorastack-calendar>
```

Attributes override the matching JSON setting for that embed only.

Event Icons And Downloads

Use square transparent PNG icons where possible. Web Lite supports 32x32, 64x64, 128x128, and 256x256 icons. The component keeps the displayed icon close to the configured iconSize.

Set a global icon size:

```
"settings": {
```

```

    "iconSize": 64,
    "iconBackground": "transparent"
  }

```

Set an event icon:

```

{
  "title": "Photography Outing",
  "iconUrl": "https://example.com/icons/photography-outing-128.png"
}

```

Post-event downloads appear after the event has completed. Only link files you are allowed to publish.

Seasons And Month Navigation

Use seasons when your organisation works around academic years, club seasons, or programme cycles.

Single season:

```

"settings": {
  "seasonStart": "09-01",
  "seasonEnd": "06-30"
}

```

Multiple seasons:

```

"settings": {
  "seasons": [
    {
      "label": "2026-2027 Season",
      "start": "2026-09-01",
      "end": "2027-06-30"
    },
    {
      "label": "Summer 2027",
      "start": "2027-07-01",
      "end": "2027-08-31"
    }
  ]
}

```

If a season is in progress, the calendar opens around the current month in that season. If the last season has ended and a future season exists, it opens on the next upcoming season start month.

Visitors can still jump to another month and year.

Mobile And Tablet Behaviour

Web Lite adapts to phones and tablets:

- Month grids become readable stacked day cards on portrait phones.
- List cards collapse to one column.
- Full descriptions open as a bottom sheet with a close button on small screens.
- Jump-to-month controls keep the year field wide enough for all digits.
- Large event icons are contained so they do not stretch.
- Buttons and action icons use touch-friendly sizes.

Always test both phone portrait and phone landscape. In desktop Chrome or Edge, open developer tools and use the device toolbar to emulate common phone sizes.

Accessibility And Security

Accessibility:

- Use high-contrast text and event backgrounds.
- Avoid placing light text over bright images without a dark overlay.
- Keep event titles concise.
- Use meaningful details page links.
- Test keyboard focus on event cards and action buttons.

Security:

- The Web Lite package runs in the visitor browser.
- Do not put secrets, API keys, payment keys, or private file URLs in `events.json`.
- Use HTTPS URLs for images, details links, and downloads.
- Only publish files that visitors are allowed to download.

Troubleshooting

The calendar does not appear:

- Confirm `aurorastack-calendar.js` is loading without a 404 error.
- Confirm `edition.js` loads before `aurorastack-calendar.js`.
- Confirm the HTML tag is `<aurorastack-calendar>`.

Events do not appear:

- Open `events.json` directly in the browser and check it loads.
- Validate the JSON syntax.
- Confirm dates use a supported format such as `2026-09-10T19:00:00`.
- Check that events are inside the `events` array.

Images or icons do not show:

- Confirm the image URL is public.
- Use JPG or PNG.
- Avoid hotlink-protected image hosts.
- Use transparent square PNGs for event icons.

The layout is too wide:

- Reduce `cellWidth`.
- Reduce `width`.

- Use fewer `listColumns`.
- Keep `forceCenter` enabled.

Text is hard to read:

- Darken `cellBackground` or `eventBackground`.
- Increase contrast between `eventTextColor` and `eventBackground`.
- Avoid busy images behind small text.