

## events.js — Simple Events Manager (JavaScript)

This PDF contains a single-file JavaScript module 'events.js' implementing a small Events manager with add, remove, update, list, find-by-date, import/export JSON and an example usage section. Copy the code into a .js file to use.

```
// events.js
// Simple Events manager in plain JavaScript
// - Add, remove, list events
// - Find events by date
// - Export/import JSON
// - Example usage at the bottom

class Events {
  constructor() {
    // Store events as objects: { id, title, start, end, location, notes }
    this.events = [];
    this._nextId = 1;
  }

  _generateId() {
    return this._nextId++;
  }

  addEvent({ title, start, end = null, location = '', notes = '' }) {
    if (!title || !start) throw new Error('Event must have a title and start date.');
    const event = {
      id: this._generateId(),
      title,
      start: new Date(start).toISOString(),
      end: end ? new Date(end).toISOString() : null,
      location,
      notes
    };
    this.events.push(event);
    return event;
  }

  removeEvent(id) {
    const idx = this.events.findIndex(e => e.id === id);
    if (idx === -1) return false;
    this.events.splice(idx, 1);
    return true;
  }

  updateEvent(id, updates) {
    const event = this.events.find(e => e.id === id);
    if (!event) return null;
    Object.keys(updates).forEach(k => {
      if (k === 'start' || k === 'end') {
        event[k] = updates[k] ? new Date(updates[k]).toISOString() : null;
      } else if (k !== 'id') {
        event[k] = updates[k];
      }
    });
    return event;
  }

  listEvents() {
    // return a copy sorted by start date
    return [...this.events].sort((a, b) => new Date(a.start) - new Date(b.start));
  }
}
```

```

findByDate(date) {
  // find events that start on the same calendar date (local)
  const target = new Date(date);
  return this.events.filter(e => {
    const s = new Date(e.start);
    return s.getFullYear() === target.getFullYear()
      && s.getMonth() === target.getMonth()
      && s.getDate() === target.getDate();
  }).sort((a,b) => new Date(a.start) - new Date(b.start));
}

exportJSON() {
  return JSON.stringify({ events: this.events }, null, 2);
}

importJSON(jsonString) {
  try {
    const parsed = JSON.parse(jsonString);
    if (!Array.isArray(parsed.events)) throw new Error('Invalid format');
    this.events = parsed.events.map(ev => ({
      ...ev,
      start: new Date(ev.start).toISOString(),
      end: ev.end ? new Date(ev.end).toISOString() : null
    }));
    // recompute next id
    const maxId = this.events.reduce((m, e) => Math.max(m, e.id || 0), 0);
    this._nextId = maxId + 1;
  } catch (err) {
    throw new Error('Failed to import JSON: ' + err.message);
  }
}

/* Example usage */
if (typeof module !== 'undefined' && require) {
  // Node environment demo
  const ev = new Events();
  ev.addEvent({ title: 'Team Meeting', start: '2025-09-15T10:00:00', end:
'2025-09-15T11:00:00', location: 'Zoom', notes: 'Discuss Q4' });
  ev.addEvent({ title: 'Doctor', start: '2025-09-20T09:00:00', location: 'Clinic' });
  console.log('All events:', ev.listEvents());
  console.log('Events on 2025-09-15:', ev.findByDate('2025-09-15'));
  const exported = ev.exportJSON();
  console.log('Exported JSON:', exported);
}

export default Events;

```