

eMS Event Chapter View Implementation Guide

Generic template and page usage for event results, preview pages, and direct links from character pages

This guide explains how to add the Chapter View feature to an existing eFed Management Suite template. The feature adds a collapsible chapter list, participant filtering, direct anchor links to matches and segments, optional feature-toggle display control, and PDF download buttons.

Contents

What this feature does	1
Recommended eMS assumptions	2
Implementation overview	2
1. Add the PDF download URL	2
2. Load matches and segments in display order	2
3. Build the Chapter View data array	3
4. Add the Chapter View markup	4
5. Add anchors to the visible results	6
6. Add PDF download buttons	7
7. Add the JavaScript filter logic.....	7
8. Add the CSS.....	9
9. Link directly from character pages	12
10. Public page and preview page usage	13
11. Suggested implementation checklist.....	13
Common pitfalls.....	13
Final notes	13

What this feature does

- Adds a collapsed-by-default Chapter View above the event results.
- Lists matches and segments in sort order.
- Lets users click a chapter entry to jump directly to that result block.
- Lets users filter the visible chapter list and visible event results by Matches, Segments, and Participant.
- Uses main participants only, so interference appearances are not exposed as spoilers.
- Supports direct deep links such as `/event/123#event-item-456` from other eMS pages.
- Can be shown or hidden visually using `feature_settings` while leaving the logic intact.

Recommended eMS assumptions

This implementation assumes a standard eMS install with the following tables or equivalent logic already in place:

- event_bookings
- events
- venues
- event_matches
- match_characters
- segment_characters
- event_match_writers
- segment_credits
- feature_settings

The example implementation also assumes that segment participants are tied to `segment_characters.match_id` and that the public event PDF endpoint is available at `/content/download_event_pdf.php?id={booking_id}`.

Implementation overview

At a high level, the feature is added in six parts:

- Load the event booking record and normal event items.
- Build an event item array that includes a stable anchor id for each match or segment.
- Build a participant list for matches and segments using main participants only.
- Optionally read `feature_settings` to decide whether the Chapter View box should be rendered.
- Render the Chapter View markup and give each result block a matching HTML id.
- Add the CSS and JavaScript needed to style and filter the chapter list and visible results.

1. Add the PDF download URL

After the booking has been loaded, create a PDF URL variable for reuse in the page.

```
$pdfUrl = '/content/download_event_pdf.php?id=' . $booking_id;
```

This can be used for a button near the event header and again near the bottom of the page.

2. Load matches and segments in display order

Use the `event_matches` table ordered by `sort_order` and `id`. This gives the Chapter View the same order as the visible results on the page.

```
$sql2 = "
SELECT
    m.id                AS match_id,
    m.match_name       AS title,
    m.match_content    AS content,
    m.is_segment       AS is_segment,
    m.reserved_segment_id AS reserved_segment_id,
    m.sort_order       AS sort_order
FROM event_matches m
```

```

WHERE m.booking_id = ?
ORDER BY m.sort_order ASC, m.id ASC
";

```

3. Build the Chapter View data array

For each event item, create a stable anchor id. The recommended format is event-item-{match_id}. This is what makes direct links from character pages work.

```

$credits = [];
$eventItems = [];
$participantOptions = [];

while ($row = $res2->fetch_assoc()) {
    $matchId = (int)$row['match_id'];
    $isSegment = (int)$row['is_segment'] === 1;
    $segmentId = (int)$row['reserved_segment_id'];
    $title = trim((string)($row['title'] ?? ''));
    $content = (string)($row['content'] ?? '');
    $anchorId = 'event-item-' . $matchId;
    $participants = [];

    if ($isSegment) {
        $stmtP = $db->prepare("
            SELECT DISTINCT c.name
            FROM segment_characters sc
            INNER JOIN characters c ON sc.character_id = c.id
            WHERE sc.match_id = ?
            ORDER BY c.name ASC
        ");

        if ($stmtP) {
            $stmtP->bind_param('i', $matchId);
            $stmtP->execute();
            $rp = $stmtP->get_result();

            while ($p = $rp->fetch_assoc()) {
                $name = trim((string)($p['name'] ?? ''));

                if ($name !== '') {
                    $participants[] = $name;
                    $participantOptions[$name] = $name;
                }
            }

            $stmtP->close();
        }
    } else {
        $stmtP = $db->prepare("
            SELECT DISTINCT c.name

```

```

        FROM match_characters mc
        INNER JOIN characters c ON mc.character_id = c.id
        WHERE mc.match_id = ?
        ORDER BY c.name ASC
    ");

    if ($stmtP) {
        $stmtP->bind_param('i', $matchId);
        $stmtP->execute();
        $rp = $stmtP->get_result();

        while ($p = $rp->fetch_assoc()) {
            $name = trim((string)($p['name'] ?? ''));

            if ($name !== '') {
                $participants[] = $name;
                $participantOptions[$name] = $name;
            }
        }

        $stmtP->close();
    }

    sort($participants, SORT_NATURAL | SORT_FLAG_CASE);

    $eventItems[] = [
        'match_id'          => $matchId,
        'segment_id'       => $segmentId,
        'title'            => $title,
        'content'          => $content,
        'is_segment'       => $isSegment,
        'anchor_id'        => $anchorId,
        'participants'     => $participants,
        'participant_string' => implode('||', $participants),
    ];
}

ksort($participantOptions, SORT_NATURAL | SORT_FLAG_CASE);

```

Important: this version intentionally does not load interference participants. That prevents spoilers in the public Chapter View and participant filter.

4. Add the Chapter View markup

Render the Chapter View between the event meta area and the event results. Wrap the whole block in the feature toggle if you are using one.

```

<?php if ($showChapterView && !empty($eventItems)): ?>
    <details class="uta-event-preview-chapters">
        <summary>

```

```

    <span class="uta-event-preview-chapters__title">Chapter View</span>
    <span class="uta-event-preview-chapters__subtitle">Click to expand</span>
</summary>

<div class="uta-event-preview-chapters__inner">
  <div class="uta-event-preview-filters">
    <div class="uta-event-preview-filter-row">
      <label for="chapter-type-filter">Show</label>
      <select id="chapter-type-filter">
        <option value="all">All</option>
        <option value="matches">Matches</option>
        <option value="segments">Segments</option>
      </select>
    </div>

    <div class="uta-event-preview-filter-row">
      <label for="chapter-participant-filter">Participant</label>
      <select id="chapter-participant-filter">
        <option value="">All Participants</option>
        <?php foreach ($participantOptions as $participantName): ?>
          <option value="<?= eventPreviewEscape($participantName) ?>">
            <?= eventPreviewEscape($participantName) ?>
          </option>
        <?php endforeach; ?>
      </select>
    </div>

    <div class="uta-event-preview-filter-row uta-event-preview-filter-row--button">
      <button type="button" id="chapter-filter-reset" class="uta-event-preview-reset-
btn">
        Reset Filters
      </button>
    </div>
  </div>

  <div class="uta-event-preview-results-note" id="chapter-results-note">
    Showing all results.
  </div>

  <ol class="uta-event-preview-chapter-list" id="event-chapter-list">
    <?php foreach ($eventItems as $index => $item): ?>
      <li
        class="uta-event-preview-chapter-item event-chapter-item"
        data-type="<?= $item['is_segment'] ? 'segment' : 'match' ?>"
        data-participants="<?= eventPreviewEscape($item['participant_string']) ?>"
      >
        <a href="#"<?= eventPreviewEscape($item['anchor_id']) ?>">
          <span class="uta-event-preview-chapter-number"><?= $index + 1 ?>.</span>
          <span class="uta-event-preview-chapter-label"><?= $item['is_segment'] ?
'Segment' : 'Match' ?>:</span>

```

```

        <span class="uta-event-preview-chapter-title-text"><?=
eventPreviewEscape($item['title']) ?></span>
    </a>

    <?php if (!empty($item['participants']): ?>
        <div class="uta-event-preview-chapter-participants">
            <?= eventPreviewEscape(implode(', ', $item['participants'])) ?>
        </div>
    <?php endif; ?>
</li>
<?php endforeach; ?>
</ol>

<div class="uta-event-preview-empty" id="event-chapter-no-results"
style="display:none;">
    No matches or segments matched your filters.
</div>
</div>
</details>
<?php endif; ?>

```

The specific class names can be changed to fit any template. Keep the data-type and data-participants attributes, because the JavaScript filter logic depends on them.

5. Add anchors to the visible results

Each visible result block must use the same anchor id that was created when building the event item array.

```

<?php foreach ($eventItems as $item): ?>
    <article
        id="<?= eventPreviewEscape($item['anchor_id']) ?>"
        class="uta-event-preview-item event-result-item"
        data-type="<?= $item['is_segment'] ? 'segment' : 'match' ?>"
        data-participants="<?= eventPreviewEscape($item['participant_string']) ?>"
    >
        <div class="uta-event-preview-item__header">
            <span><?= $item['is_segment'] ? 'Segment' : 'Match' ?></span>
            <h2><?= eventPreviewEscape($item['title']) ?></h2>
        </div>



        <div class="uta-event-preview-item__content">
            <?= $item['content'] ?>
        </div>
    </article>
<?php endforeach; ?>

```

The shared id and anchor format is what allows jump links from the Chapter View and from other pages such as character.php.

6. Add PDF download buttons

A generic eMS implementation usually works best with one PDF button near the top and one near the bottom.

```
<div class="uta-event-preview-actions">
  <a href="<? = eventPreviewEscape($pdfUrl) ?>" class="uta-event-preview-btn" target="_blank">
    <span aria-hidden="true">  </span> Download as PDF
  </a>
</div>
<div class="uta-event-preview-bottom-actions">
  <a href="<? = eventPreviewEscape($pdfUrl) ?>" class="uta-event-preview-btn" target="_blank">
    <span aria-hidden="true">  </span> Download Results PDF
  </a>
</div>
```

7. Add the JavaScript filter logic

The JavaScript below filters both the Chapter View list and the visible event results. If the Chapter View is not rendered because the feature is turned off, the script exits safely.

```
<script>
document.addEventListener('DOMContentLoaded', function () {
  const typeFilter = document.getElementById('chapter-type-filter');
  const participantFilter = document.getElementById('chapter-participant-filter');
  const resetBtn = document.getElementById('chapter-filter-reset');
  const chapterItems = document.querySelectorAll('.event-chapter-item');
  const resultItems = document.querySelectorAll('.event-result-item');
  const noResultsBox = document.getElementById('event-chapter-no-results');
  const resultsNote = document.getElementById('chapter-results-note');

  if (!typeFilter || !participantFilter) {
    return;
  }

  function typeMatches(itemType, selectedType) {
    if (selectedType === 'all') return true;
    if (selectedType === 'matches') return itemType === 'match';
    if (selectedType === 'segments') return itemType === 'segment';
    return true;
  }

  function participantMatches(participantString, selectedParticipant) {
    if (!selectedParticipant) return true;
    const participants = participantString ? participantString.split('||') : [];
    return participants.includes(selectedParticipant);
  }

  function updateResultsNote(typeValue, participantValue, visibleCount) {
    let parts = [];
```

```

    if (typeValue === 'matches') {
        parts.push('matches only');
    } else if (typeValue === 'segments') {
        parts.push('segments only');
    } else {
        parts.push('all results');
    }
}

if (participantValue) {
    parts.push('for ' + participantValue);
}

if (resultsNote) {
    resultsNote.textContent = visibleCount > 0
        ? 'Showing ' + parts.join(' ') + '.'
        : 'No results matched your filters.';
}
}

function applyFilters() {
    const selectedType = typeFilter.value;
    const selectedParticipant = participantFilter.value;
    let visibleCount = 0;

    chapterItems.forEach(function (item) {
        const itemType = item.getAttribute('data-type') || '';
        const itemParticipants = item.getAttribute('data-participants') || '';
        const visible = typeMatches(itemType, selectedType) &&
participantMatches(itemParticipants, selectedParticipant);

        item.style.display = visible ? '' : 'none';

        if (visible) {
            visibleCount++;
        }
    });

    resultItems.forEach(function (item) {
        const itemType = item.getAttribute('data-type') || '';
        const itemParticipants = item.getAttribute('data-participants') || '';
        const visible = typeMatches(itemType, selectedType) &&
participantMatches(itemParticipants, selectedParticipant);

        item.style.display = visible ? '' : 'none';
    });

    if (noResultsBox) {
        noResultsBox.style.display = visibleCount === 0 ? 'block' : 'none';
    }
}

```

```

        updateResultsNote(selectedType, selectedParticipant, visibleCount);
    }

    typeFilter.addEventListener('change', applyFilters);
    participantFilter.addEventListener('change', applyFilters);

    if (resetBtn) {
        resetBtn.addEventListener('click', function () {
            typeFilter.value = 'all';
            participantFilter.value = '';
            applyFilters();
        });
    }

    applyFilters();
});
</script>

```

8. Add the CSS

Below is a generic dark-theme CSS block that can be adapted to most eMS templates using red, gold, and charcoal styling. Rename the classes if needed, but keep the structural pieces aligned with the markup.

```

.uta-event-preview-chapters {
    margin: 18px 0 15px;
    border: 1px solid #444;
    border-radius: 8px;
    background: #2b2b2b;
    overflow: hidden;
    box-shadow: 0 0 5px rgba(0,0,0,0.25);
}

.uta-event-preview-chapters summary {
    cursor: pointer;
    list-style: none;
    padding: 14px 16px;
    font-weight: 700;
    user-select: none;
    display: flex;
    align-items: center;
    justify-content: space-between;
    gap: 12px;
    background: #8b0000;
    color: gold;
    border-bottom: 1px solid #5a0000;
}

.uta-event-preview-chapters summary::-webkit-details-marker {
    display: none;
}

```

```
.uta-event-preview-chapters summary::after {
  content: "▼";
  font-size: 14px;
  transition: transform 0.2s ease;
  color: gold;
}

.uta-event-preview-chapters[open] summary::after {
  transform: rotate(180deg);
}

.uta-event-preview-chapters__title {
  font-size: 16px;
  color: gold;
}

.uta-event-preview-chapters__subtitle {
  font-size: 12px;
  font-weight: 400;
  color: #ddd;
  margin-left: auto;
  margin-right: 10px;
}

.uta-event-preview-chapters__inner {
  padding: 0 16px 16px;
  border-top: 1px solid #333;
  background: #2b2b2b;
  color: #ddd;
}

.uta-event-preview-filters {
  display: flex;
  flex-wrap: wrap;
  gap: 10px 14px;
  align-items: end;
  margin: 15px 0 12px;
}

.uta-event-preview-filter-row {
  display: flex;
  flex-direction: column;
  gap: 6px;
  min-width: 180px;
}

.uta-event-preview-filter-row label {
  font-size: 13px;
  color: gold;
}
```

```
}

.uta-event-preview-filter-row select {
  padding: 8px 10px;
  border: 1px solid #555;
  border-radius: 6px;
  background: #1f1f1f;
  color: #f5f5f5;
}

.uta-event-preview-filter-row select:focus {
  outline: none;
  border-color: gold;
  box-shadow: 0 0 0 1px rgba(255, 215, 0, 0.25);
}

.uta-event-preview-reset-btn {
  padding: 8px 12px;
  border: 1px solid #5a0000;
  border-radius: 6px;
  background: #b22222;
  color: #fff;
  cursor: pointer;
  font-weight: 600;
  transition: background 0.2s ease, border-color 0.2s ease;
}

.uta-event-preview-reset-btn:hover {
  background: #8b0000;
  border-color: #8b0000;
}

.uta-event-preview-results-note {
  font-size: 13px;
  color: #ccc;
  margin-bottom: 10px;
}

.uta-event-preview-chapter-list {
  margin: 0;
  padding-left: 18px;
}

.uta-event-preview-chapter-item {
  margin-bottom: 12px;
  color: #ddd;
}

.uta-event-preview-chapter-item a {
  text-decoration: none;
}
```

```

    font-weight: 600;
    color: gold;
}

.uta-event-preview-chapter-item a:hover,
.uta-event-preview-chapter-item a:focus {
    color: #ffd966;
    text-decoration: underline;
}

.uta-event-preview-chapter-participants {
    font-size: 12px;
    color: #bbb;
    margin-top: 3px;
    margin-left: 2px;
}

.event-result-item {
    scroll-margin-top: 20px;
}

@media (max-width: 700px) {
    .uta-event-preview-chapters summary {
        align-items: flex-start;
        flex-direction: column;
    }

    .uta-event-preview-chapters__subtitle {
        margin-left: 0;
        margin-right: 0;
    }

    .uta-event-preview-filter-row {
        width: 100%;
        min-width: 100%;
    }
}

```

If your template already has event card, panel, or button styles, you can fold these rules into those existing classes instead of adding a new visual system.

9. Link directly from character pages

Once the event page uses ids in the format event-item-{event_match_id}, any other eMS page can link directly to that result.

```

<a href=" ../event/<?=$evt['booking_id'] ?>#event-item-<?=(int)$evt['event_match_id'] ?>">
    <?=$htmlspecialchars($evt['event_name']) ?>:
    <?=$htmlspecialchars($evt['booking_label']) ?>
</a>

```

For character.php, update the Events tab query to also select em.id AS event_match_id. Then use that value in the event link or in the match name link. This gives a clean jump from the character history table to the exact match or segment on the event page.

10. Public page and preview page usage

The same structure works for both public event pages and booking preview pages.

- Public results page: load only event_bookings rows where booking_status = 'Public'.
- Preview page: load booking rows where booking_status = 'Booking' or whatever status your preview page uses.
- If the preview page uses a different PDF endpoint, only the pdfUrl line needs to change.
- All Chapter View, anchor, filter, and CSS logic can remain the same between both versions.

11. Suggested implementation checklist

- Add the PDF URL variable.
- Load event_matches in sort order.
- Build \$eventItems and \$participantOptions.
- Use main participants only for spoiler-safe filtering.
- Add the feature_settings toggle lookup if desired.
- Insert the Chapter View details block above the results or above the PDF button, depending on your layout.
- Give each result block an id in the format event-item-{match_id}.
- Add the filter JavaScript.
- Add the CSS to your template stylesheet or page-specific stylesheet.
- Update character.php or other pages to link to /event/{booking_id}#event-item-{event_match_id}.

Common pitfalls

- Using the wrong column for segment_characters. In many eMS installs, segment_characters uses match_id rather than segment_id.
- Showing interference participants in the public Chapter View. This can spoil surprises.
- Forgetting to include the same anchor id on the visible result block. The chapter links will render but not jump correctly.
- Changing the data-type or data-participants attributes without updating the JavaScript selectors.
- Using a feature_settings key in code that does not exist in the database.
- Pointing preview pages at the public-only booking status or vice versa.

Final notes

This feature is designed to be generic enough for nearly any eMS front-end template. The exact class names, markup wrappers, or button styling can change to match the theme, but the anchor system, participant collection, data attributes, and JavaScript filter behavior should remain consistent. That consistency is what makes the Chapter View reusable across public event pages, preview pages, and direct links from character profiles or other site areas.